Update Summary Generation based on Semantically Adapted Vector Space Model

A.Kogilavani
Kongu Engineering College
Perundurai, Erode
Tamilnadu, India

P.Balasubramanie
Kongu Engineering College
Perundurai, Erode
Tamilnadu, India

ABSTRACT

This paper proposes an approach of personalizing the vector space model with dependency parse relations and applying Latent Semantic Analysis on it to generate update summary from multiple documents. The purpose of the update summary is to inform the reader of new information about the topic. The main task was to produce two concise summaries from two related sets of documents, where the second summary was an update summary of the first one. In the proposed system individual word weight is calculated using tsf-isf and dependency parse of the document has been used to modify the tsf-isf weight of words by incorporating the dependency between each pair of words. To preserve important semantic information in the text LSA is performed and to select relevant sentences basic features, advanced features and update specific features are used. The experiment result shows that low overlap between initial summary and its update summary.

General Terms

Document Summarization, Natural Language Processing

Keywords

Preprocessing, PoS Tagging, Similarity Matrix, Dependency Parsing, Semantic Similarity Matrix, Feature Specific Sentence Ranking Strategy, Initial Summary, Update Summary.

1. INTRODUCTION

Electronic document information is exponentially growing and where time is a critical resource in this epoch, it has become practically impossible for any user to read large numbers of individual documents. It is therefore important to discover methods of allowing users to extract the main idea from collection of documents. Automatic document summarization of multiple documents would thus be immensely useful to fulfill such information seeking goals by providing an approach for the user to quickly view highlights or relevant portions of documents. Multi-document summarization is the process of generating a generic or topic-focused summary by reducing documents in size while retaining the main characteristics of the original documents. Since one of the problems of data overload is caused by the fact that many documents share the same or similar topics, automatic multi-document summarization has attracted much attention in recent years. A number of researchers have done good work in multi-document summarization. Unfortunately, much of the existing system has focused on the specified static document collection, without attempting to capture the changes over time. Furthermore, the difficulty of constructing an adequate model for dynamically changing information itself is not fully recognized. Thus the update summarization task is valuable in periodically monitoring the important changes for the new relevant information over a given time period. Document updating technique is also very helpful for people to acquire new information or knowledge by eliminating out-of-date or redundant information. It aims to produce a summary by describing the majority of information content from a set of documents under the assumption that the user has already read a given set of earlier documents. This type of summarization has been proved extremely useful in tracing news stories, only new and update contents should be summarized if users have already known something about the documents. Update summarization task is to produce first initial summary which is a compressed summary of a set of newswire articles about a particular topic. Next is to produce update summary which is a compressed summary of a subsequent set of newswire articles for the same topic, under the assumption that the reader has already read the first set of documents. The purpose of the update summary is to inform the reader of new information about the topic. The task is based on a scenario in which a user has a standing question that gets asked of an information retrieval or summarization system at two different times. The first time, the system retrieves a number of relevant newswire articles, which the user reads completely. Later perhaps the next day, or even weeks later, the user has time to return to the system to see if there are any updates concerning his question of interest. New articles have arrived, and the system must generate an update summary of the new articles, under the assumption that the user has already read the initial articles. The issues to be considered for multi-document update summarization are as follows: First, simple word-matching measure is not able to completely capture the content similarity because news articles consist of different words to describe the same events. Traditional vector space model assumes a bag-of-words model of the document where the words within a document are independent of each other. Therefore, effort has to be taken to find the dependency between the words which is used to select semantically important sentences from the document collection. Second issue will be generating well organized fluent summary by selecting more relevant keywords from multiple documents. This can be done with the help of Latent Semantic Analysis (LSA) which is also used to reduce dimensions. Third, the information enclosed in different documents frequently overlaps with each other, therefore, it is necessary to find an effective way to select the relevant sentences while removing redundancy. This can be done with the help of feature specific sentence ranking algorithm. Finally, to select novel sentences from the given document set, there is a need to introduce novelty measure. In order to address the first issue, an automatic document summarization method that uses semantics of information in order to form efficient and relevant summary is proposed. The proposed work retrieves the semantics of information through semantic analysis. Initially documents are mapped into vector space model and similarity matrix is constructed by tsf-isf which utilizes WordNet. WordNet is an online lexical database in which English nouns, verbs, adjectives and adverbs are organized into synonym sets or synsets. The proposed system retrieves list of synonyms of each word from WordNet which is used for calculating term frequency of a word because document sentences may contain different vocabularies to express the same meaning. Then similarity matrix is modified into semantic similarity matrix by identifying grammatical relationship between the words. In order to extract one of the most fundamental semantic units from natural language text Stanford Tagger and Parser is used. The context is intuitively extracted from typed dependency structures basically depicting dependency relations. The dependency relations imply deep, fine grained, labeled dependencies that encode long distance relations and passive information. To address the second issue LSA is performed to reduce semantic similarity matrix and this reduction has the effect of preserving the most important semantic information in the text while reducing noise and other undesirable artifacts. To address the third issue, feature specific sentence ranking strategy is applied to all the sentences and high ranking sentences are selected as candidate sentences for summary. To address the fourth issue, the proposed system utilizes Novel and Topic Relevance measure in order to extract novel sentences from the second set of documents.

2. RELATED WORK

Su Jian Li et al [1] adopted the feature-based sentence extractive framework and introduced a new filter feature to adapt to the update task. The design of features is the important part. The principle of designing filtering features is to distinguish the current documents from the previous documents, and reflect the main idea of current documents. But due to limited factors to design the filtering feature, they couldn’t claim any statistical significance on their findings. Bysani et al [2] build a sentence extractive summarizer that extracts and ranks sentences before finally generating summaries. For sentence scoring, a machine learning algorithm, support vector regression is used to predict sentence rank using various features. It proposes a feature that can effectively capture the novelty along with relevancy of a sentence in a topic. But the new feature incorporated is query independent. So query focused novelty factor is need to be considered. Josef Steinberger et al [3] described the development of update summarizer that must solve the novelty versus redundancy problem which is based on Iterative Residual Rescaling that creates the latent semantic space of a set of documents under considerations. Eric Wehrli et al [4] developed a multi-document topic driven update summarizer, News Symbolic Summarizer, proposed by RALI. The most distinctive feature of this summarization system is to relay on the syntactical parser FIPS to extract linguistic knowledge from the source documents. News Symbolic Summarizer selects sentences based on linguistic metrics, especially TF-IDF score. The result of FIPS parse tree is not so accurate. Praveen Bysani et al [5] build a sentence extractive summarizer that extracts and ranks sentences before finally generating summaries. For sentence scoring, a machine learning algorithm, support vector regression is used to predict sentence rank using various features. It proposes a feature that can effectively capture the novelty along with relevancy of a sentence in a topic. But the new feature incorporated is query independent. So query focused novelty factor is need to be considered. Ravindranath Chowdary et al [6] utilized MMR approach to filter the information which is already present in the summaries that were generated on set A and then select that set of sentences from set B that are both informative and non repetitive. But MMR approach alone is not giving satisfactory results.

3. PROPOSED SYSTEM OVERVIEW

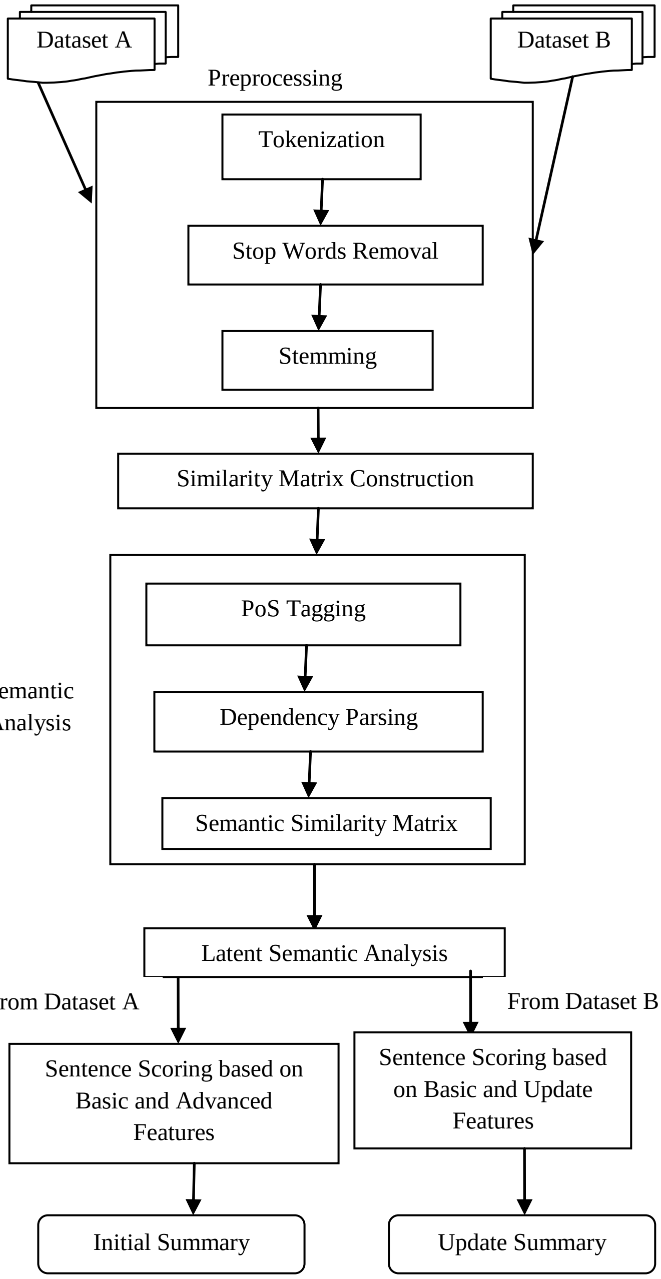

Fig 1: Overview of the Proposed Approach

Figure 1 illustrates an overview of the proposed approach to generate initial and update summary from multiple documents. The input to the system is a collection of topic related two sets of documents. The output is a concise set of two summaries providing the condensed information of the input documents. The main aim is to simulate a user who is interested in learning about the latest developments on a specific topic, and who wishes to read a brief summary of the latest news. After preprocessing, semantic analysis is performed to identify the semantic importance of the sentences. Semantic similarity matrix is reduced by applying LSA in order to identify most important sentences. Then for each sentence, score is calculated based on generic feature for dataset A articles and based on generic as well as update features for dataset B articles. Based on the ranking of sentences, sentences are selected and ordered in a way in which the sentences are included in the original documents and finally initial and update summary is generated. The proposed approach can be decomposed into the following sub processes: 1. Preprocessing 2. Similarity Matrix Construction 3.Semantic Analysis 4. Latent Semantic Analysis 5.Initial summary generation 6.Update Summary generation.

3.1. Preprocessing

During preprocessing, split the documents into sentences and then into words. Tokenization is the very basic ability of splitting text in the documents into meaningful units like words, punctuation, etc. From words list remove frequently occurring insignificant words called stop words because they do not contribute to the meaning of the sentence. Stemming refers to the ability to recognize variants of words as pertaining to the same words. For example women and woman are two variants of the same word. Get the stem of each word by applying enhanced Porter Stemmer algorithm.

3.2. Similarity Matrix Construction

Let D = {d_1,d_2,d_3……d_k) be the collection of documents where k is the total number of documents in D. Let N = {s_1,s_2,s_3……s_n} be the number of sentences in document collection D which can be calculated during preprocessing. Let M = {w_1,w_2,w_3……w_m} be the number of words in each sentence after removing stop words. Let d_j be the i^{th} document in document collection D, $S_{i,k}$ be the i^{th} sentence in any document d_k, w_m be a word in a sentence $S_{i,k}$. The Term Synonym Frequency (TSF) of each word is calculated by

$$TSF(w_m) = \sum_{w_m \in \{\{w\} \cup synonym(w)\}} \alpha.TF(w_m) \qquad (1)$$

In TSF calculation to incorporate word synonym into account the Tern Frequency(TF) of each word and it's synonym is multiplied by α where α = 1 for the word and α = 0.5 for synonym of word. Synonym is retrieved from WordNet which is a lexical database for English language. The Term Frequency *(TF)* of each word is calculated by

$$TF(w_m) = \frac{n_m}{\sum_k n_k} \qquad (2)$$

where n_m is the number of times the m^{th} word appears in the document collection D. For example if word 'cargo' is occurred 10 times in document collection D, then n_m value is 10. The denominator in eqn.(2) is the number of occurrences of all words in all sentences of the document collection D. Inverse Sentence Frequency is calculated as,

$$ISF(w_m) = \log \frac{N}{S}$$

where S is the number of sentences that contain m^{th} word.

Table 1. Sample TSF-ISF Values

Article ID	Sentence ID	Words	TSF-ISF
1	5	December 19 2000 Airbus **Launches** Plane **Calling** A380	0.00624 0.01339 0.00624 0.00092 **0.00233** 0.00098 **0.00446** 0.00018
1	6	January 2001 US Freight company Federal Express **announces** cargo version reserving 10	0.00446 0.00892 0.00446 0.00312 0.00277 0.00312 0.00000 **0.01401** 0.00000 0.00178 0.00446 0.00233

Table 1 represents sample TSF-ISF values for two sample sentences in the same article. These TSF-ISF values are used to calculate similarity matrix.

3.3. Semantic Analysis

The similarity matrix constructed using tsf-isf is unable to define the context between the words. To identify context between the words, modify the similarity matrix by identifying verb-object pair. Merely acquiring the verbs from sentences doesn't provide the semantic to the similarity matrix. There is a need to find whether these words are really important. The importance of the word can be estimated by the application of the word in the document. Objects corresponding to the verbs have to be identified in order to determine the extent of relevancy. The objects are Nouns/Adjectives for the verbs. Weight for verbs in similarity matrix is updated based on weight of objects affected by it. If more than object is there for the same verb then maximum weight amongst all the objects corresponding to the given verb is obtained and this weight is added to the weight of the corresponding verb in similarity matrix which yields semantic similarity matrix. The vector space model has been adapted to incorporate semantic information by increasing the tsf-isf value if a semantic relation holds between the words. After performing semantic analysis, initial summary or general summary is generated for dataset A documents and then update summary is generated for dataset B documents based on various features.

3.3.1 Semantic analysis algorithm

- Get the similarity matrix of document collection D.
- Assign Part-of-Speech (PoS) to each word in the document to get tagged documents.

- Apply tagged documents to parser to find the dependencies between words in a sentence.
- Identify verbs in each sentence of the document and objects that are affected by it.
- Find contextual object. If more than object is there for the same verb then maximum weight amongst all the objects corresponding to the given verb is selected as contextual object.
- Add the contextual object weight with its similarity matrix weight to form semantic similarity matrix.

3.3.2 PoS tagging

In corpus linguistics, part-of-speech tagging is the process of marking up the words in a corpus as corresponding to a particular part of speech, based on both its definition, as well as its context —i.e. relationship with adjacent and related words in a sentence.

Table 2. Sample sentences and their tagged output

Article ID	Sentence ID	Sentence	Word	Tag
1	5	December 19, 2000: Airbus officially launches the plane, calling it the A380.	December	NNP
			19	CD
				,
			2000	CD
				:
			Airbus	NNP
			Officially	RB
			Launches	VBZ
			The	DT
			Plane	NN
				,
			Calling	VBG
			It	PRP
			The	DT
			A380	NN
1	6	January 2001: The US freight company Federal Express announces the first order of the cargo version of the A380, reserving 10.	January	NNP
			2001	CD
				:
			The	DT
			US	NNP
			freight	NN
			company	NN
			Federal	NNP
			Express	NNP
			announces	VBZ
			the	DT
			first	JJ
			order	NN
			of	IN
			the	DT
			cargo	NN
			version	NN
			of	IN
			the	DT
			A380	NN
				,
			reserving	VBG
			10	CD

Sample sentences and their corresponding tagged outputs are shown in Table 2. Traditional grammar classifies words based on eight parts of speech: the verb, the noun, the adjective, the pronoun, the adverb, the preposition, the conjunction and the interjection. In fact the same word can be a noun in one sentence and a verb or adjective in another. In order to assigns parts of speech to each word the proposed system utilizes Stanford Log-Linear Part-of-speech Tagging which produces tagged documents. Then the tagged documents are passed through Stanford parser to extract grammatical relationships in a sentence and the output is represented using Stanford typed dependencies. This helps in identifying the verbs in each sentence of the document and the nouns and adjectives called as objects affected by it.

3.3.3 Dependency parsing

A dependency parse represents dependencies between individual words. A typed dependency parse additionally labels dependencies with grammatical relations such as subject and indirect object. Each word in the sentence is the dependent of one other word. Stanford dependencies generated for each of the above parsed sentences carry word-position numbers along with their arguments.

3.3.4 Semantic similarity matrix construction

In order to construct the semantic similarity matrix, first Verb-Object pair is identified and then contextual words are identified if more than one object is there for the same verb. Contextual words are verbs which are applied to the important object. The object with maximum weight is added with the original verb weight to modify the weight of the corresponding verb and based on this value semantic similarity matrix is constructed and represented in Table 3.

Table 3. Semantic similarity matrix construction

Article ID	Sentence ID	Words	S_(TSF-ISF)
1	5	December	0.00624
		19	0.01339
		2000	0.00624
		Airbus	0.00092
		Launches	**0.00857**
		Plane	0.00098
		Calling	**0.00464**
		A380	0.00018
1	6	January	0.00446
		2001	0.00892
		US	0.00446
		Freight	0.00312
		company	0.00277
		Federal	0.00312
		Express	0.00000
		announces	**0.01713**
		cargo	0.00000
		version	0.00178
		reserving	0.00446
		10	0.00233

3.4 Latent Semantic Analysis

Latent Semantic Analysis (LSA) is a mathematical technique for extracting and inferring relations of expected contextual usage of words in passages of discourse. It utilizes a mathematical technique called Singular Value Decomposition to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text. Sentences that are closely related, but contain no common words, will be recognized as similar under LSA analysis. Queries, or concept searches, against a set of documents that have undergone LSI will return results that are conceptually similar in meaning to the search criteria even if the results don't share a specific word or words with the search criteria. LSA overcomes two of the most problematic constraints of Boolean keyword queries: multiple words that have similar meanings (synonymy) and words that have more than one meaning (polysemy). Synonymy and polysemy are often the cause of mismatches in the vocabulary used by the authors of documents and the users of information retrieval systems. As a result, Boolean keyword queries often return irrelevant results and miss information that is relevant. In general, the process involves constructing a weighted term-document matrix, performing a Singular Value Decomposition on the matrix by transforming the single term-frequency matrix, *A*, into three other matrices— a term-keyword vector matrix U, a singular values matrix ∑, and a document-keyword vector matrix V^T, which satisfy the following relations:

$$A = U \sum V^T$$

In the formula, A, is the supplied *m* by *n* weighted tsf-isf matrix in a collection of documents where *m* is the number of unique terms, and *n* is the number of documents. U is a computed *m* by *r* matrix of term vectors where *r* is the rank of A—a measure of its unique dimensions ≤ min(*m*,*n*). ∑ is a computed *r* by *r* diagonal matrix of decreasing singular values which represents the strength of each keyword, and V^T is a computed *n* by *r* matrix of document vectors. Dimensionality reduction is done by setting the smallest singular values to zero and by excluding out the negative metrics. After selecting the relevant document, the words with highest weights are selected as keywords which are of high relevance in summary. Thus SVD is used to reduce the rank or truncate the singular value matrix and this reduction has the effect of preserving the most important semantic information in the text while reducing noise and other undesirable artifacts of the original space of A. JAMA (Java Matrix Package) from Joe Hicklin el al [7] is used to perform all the matrix operations.

3.5 Initial Summary Generation

To generate initial summary or general summary, there is a need to capture the relevant sentences from multiple documents. Relevant sentences are selected based on different features. The proposed work combines six features from Kogilavani et al [8] which is referred to as Basic Features with new additional features referred to as Advanced Features like word similarity between sentence and topic, sentence frequency score and document frequency score. During initial summary generation, a subset of ranked sentences is selected to generate summary. A redundancy check is done between a sentence, and summary generated so far, before selecting it into the summary. Sentences are adjusted on their order of occurrence in original documents to improve readability.

Basic Feature 1. *Word feature*

The Word_Feature (*W_F*) is defined as

$$W\_F(s_{i,k}) = \sum Word\_Score(s_{i,k}).f(w_m, s_{i,k})$$

where $f(w_m,s_{i,k})$ is the frequency of each word *w* in sentence $s_{i,k..}$Word score is calculated according to the following formula and is used to determine the coverage as well as anti-redundancy of a summary.

$$Word\_Score(s_{i,k}) = \sum_{i=1}^{m} S\_(TSF(w_i).ISF(w_i))$$

where S_(TSF(w_i).ISF(w_i)) is the Semantic Term SynonymFrequency and Inverse Sentence Frequency whose values are available in Table 7. Word score of each sentence is calculated by adding S_(TSF.ISF) values of all the words in the sentence.

Basic Feature 2.*Position feature*

Always the first sentence of the document is most important. From Pierre-Etienne Genest et al [9] the position feature is defined by considering maximum positions of 3. For example, the first sentence in a document has a score value of 3/3, the second sentence has a score 2/3 and third sentence has a score value of 1/3. The Position Feature (*P_F*) is calculated by following the equation

$$P\_F(s_{i,k}) = \frac{Position(s_{i.k})}{3}$$

Basic Feature 3.*Sentence length feature*

To find out the importance of the sentence in a particular document the length of the sentence plays a vital role. The length feature of the sentence is defined as,

$$L\_F(s_{i,k}) = \frac{N * length(s_{i,k})}{length(d_k)}$$

Here *length($s_{i,k}$)* is the number of words in a sentence $s_{i,k}$ *and length(d_k)* is the number of words in a document d_k.

Basic Feature 4.*Sentence centrality feature*

Sentence centrality defines words overlap between the given sentence and other sentences in the document. It is calculated as,

$$C\_F(s_{i,k}) = \frac{words(s_{i,k}) \cap words(others)}{words(s_{i,k}) \cup words(others)}$$

Basic Feature 5.*Sentence with proper noun feature*

Any sentence that contains more proper nouns is an important one. In order to identify proper nouns, the following equation is used.

$$PN\_F(s_{i,k}) = \frac{PN\_Count(s_{i,k})}{Length(s_{i,k})}$$

Here *PN-Count($s_{i,k}$)* denotes number of proper nouns in sentence $s_{i,k..}$

Basic Feature 6.*Sentence with numerical data feature*(5)

Any sentence that contains numerical data is an important one. Hence it must be extracted and to be included in the summary. This feature is calculated as, Here *ND_Count($s_{i,k}$)* denotes number of numerical data in sentence $s_{i,k.}$

Advanced Feature 1.*Word similarity between sentence and topic feature*

Any sentence that contains words similar to the given topic is an important one. To identify the similarity between the term and the topic, the following equation is used.

$$WSim\_F(s_{i,k}) = \sum_{w_i \varepsilon T, w_j \varepsilon S} sim(w_i, w_j)$$

where *sim(w_i,w_j)*=1 if both word and the topic are same, 0 otherwise. Here *T* represents topic sentence.

Advanced Feature 2.*Sentence frequency score feature*

To determine the number of sentences in which particular word occurred in a document set among the total number of sentences in the document set, the following equation is used. To calculate the importance of individual word in a sentence, the following equation is used.

$$SFS(s_{i,k}) = \sum_{i \in S} \frac{SFS(w_i)}{|S|}$$

$$SFS(w) = \frac{\{|s| : w \in s\}}{|N|}$$

Advanced Feature 3.*Document frequency score feature*

To determine the number of documents in which particular word occurred in a document set among the total number of sentences in the document set, the following equation is used. To calculate the importance of individual word in a sentence, equation (16) is used.

$$DFS(w) = \frac{\{|d| : w \in d\}}{|D|}$$

$$DFS(s_{i,k}) = \sum_{i \in d} \frac{DFS(w_i)}{|d|}$$

Sentence score is calculated for all sentences in different feature combinations. High scored sentences are selected for summary and those sentences are arranged in decreasing order of score. Highest ranking sentences are selected and summary is generated by arranging the selected sentences in the order in which they appeared in original documents.

3.6 Update Summary Generation

To capture the relevant sentences from dataset B articles, the proposed work combines six Basic Features with two Update specific features. These features are defined as follows

Update Feature1. *Novel and Topic Relevance measure (NTRM)*

This new update feature is defined as the combination of topic relevance and novelty. The NTRM of a word is defined as follows

$$NTRM(w_i) = NM(w_i) + TRM(w_i)$$

The Novelty factor is defined as

$$NM(w_i) = \frac{|D|_B}{|D|_A + |D|}$$

where

$|D|_B$ = No. of document in dataset B that contains word$_i$ or relevance of word.

$|D|_A$ = No. of document in dataset A that contains word$_i$ or novelty of word.

|D| = No. of documents in dataset B.

The Topic relevance between sentences and topic is defined as follows

$$TRM(w_i) = \sum_{w_i \in S, w_j \in T} sim(w_i, w_j)$$

where *sim(w_i,w_j)* - Topic relevance of word, *T* - Topic, *S* - Sentence

The NTRM of a sentence is defined as follows

$$NTRM(S) = \frac{\sum_{i \in S} NTRM(w_i)}{|S|}$$

where |*S*| - Length of sentences.

Update Feature2. *Simple Update Summary Measure(SIM)*

This new update feature is used to identify the similarity between word in dataset A and word in dataset B. If both the words are similar then similarity between words are set to 1, otherwise set to 0.

4. EXPERIMENTS AND EVALUATION

(15)

In this section, our summarization method will be evaluated on the TAC 2008 dataset. Dataset consists of 48 topics, 20 documents per topic in chronological order. The entire dataset is arranged into two clusters of articles, referred to as dataset A and dataset B in which dataset B articles were more recent than dataset A articles, and the summary of the second cluster had to provide only an update about the topic, avoiding any repetition of information from the first cluster. Main task is to produce initial summary from a set of A articles. Update task is to produce update summary from a set of B articles with the assumption that the information in the first set is already known to the reader. According to Yihong Gong et al [10] we evaluated our method by comparing the generated summaries to human summaries under three different measures like Precision, Recall and ROUGE-1 measure.

(16)

4.1 Precision

Precision can be calculated based on machine generated summary and the human summary.

$$P = \frac{N_o}{N_m}$$

where N_o = Number of common terms or words in both human and machine summary, N_m= Number of terms in machine summary.

4.2 Recall

Recall (R) is defined as

$$R = \frac{N_o}{N_h}$$

where N_o = Number of common terms in both human and machine summary, N_h= Number of terms in human summary.

Figure 2 & 3 represents performance measure based on precision and recall for all six Basic Features(BF), Six Basic Features combined with Advanced Feature1(BF+AF1), Six Basic Features combined with Advanced Feature2(BF+AF2), Six Basic Features combined with Advanced Feature3(BF+AF3), Six Basic Features combined with All Advanced Features(BF + All AF). The chart shows that when Basic Features are combined with Advanced Feature2, i.e Sentence Frequency Score, the precision is high compared to all other feature combinations. By incorporating sentence specific features along with S_(TSF-ISF), the precision is improved which implies that the coverage in machine summary is improved.

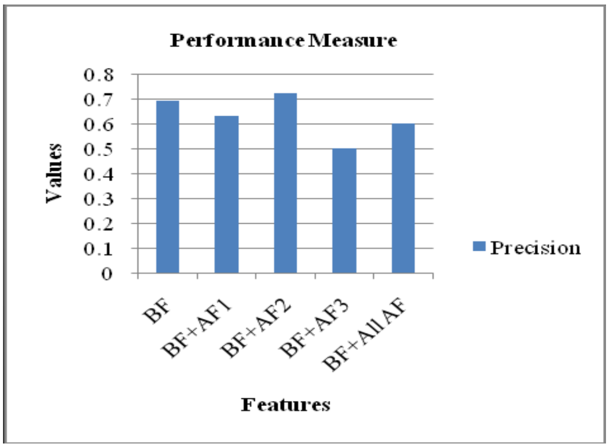

Fig 2: Initial Summary Performance Measure Based on Precision

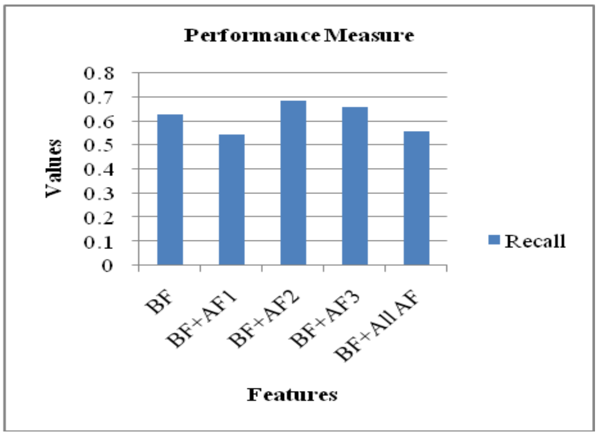

Fig 3: Initial Summary Performance Measure Based on Recall

Figure 4 & 5 represents performance measure based on precision and recall for all six Basic Features(BF) combined with Update Feature1(BF+UF1), Six Basic Features combined with Update Feature2(BF+UF2), Six Basic Features combined with all two Update Features(BF+UF1+UF2). The chart shows that when considering both Update Features, the precision is high compared to all other feature combinations.

(23)

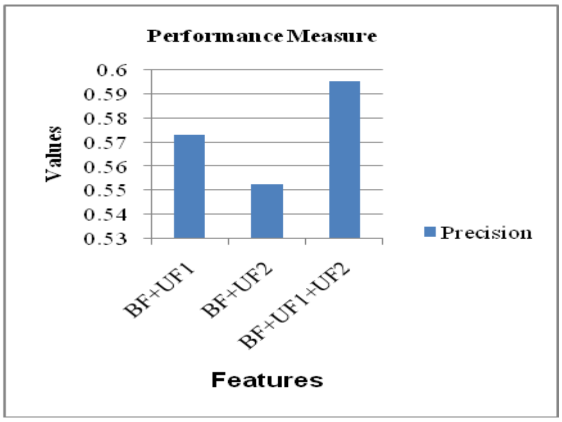

Fig 4:Update Summary Performance Measure Based on Precision

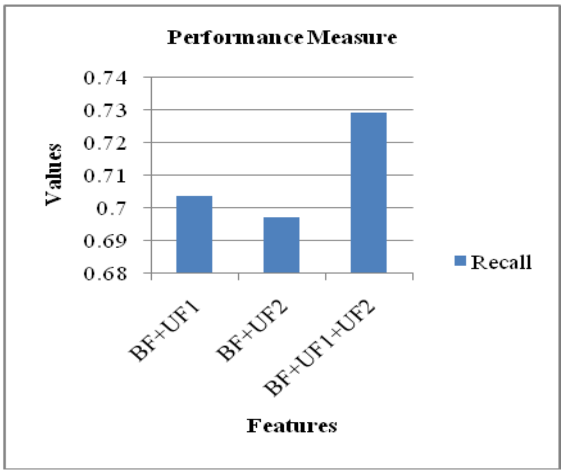

Fig 5: Update Summary Performance Measure Based on Recall

4.3 ROUGE-1 measure

According to Lin [11] ROUGE, or Recall-Oriented Understudy for Gisting Evaluation, is a set of metrics and a software package used for evaluating automatic summarization and machine translation software in natural language processing. *ROUGE* measures the quality of a summary by counting the overlapping units such as the *n*-gram, word sequences and word pairs between the generated summary and the reference summary. We use ROUGE-1 as the evaluation metric.

$$ROUGE\_1\ Score = \frac{X}{Y} \tag{24}$$

where X is count of number of unigrams that occur in machine and manual summary and Y is total number of unigrams. The following figure 6 compares *ROUGE-1 Score* of Initial Summary with Update Summary, Initial Summary with Initial Manual Summary, Update Summary with Update Manual Summary. The result shows that the overlap between Initial Summary and Update Summary is low.

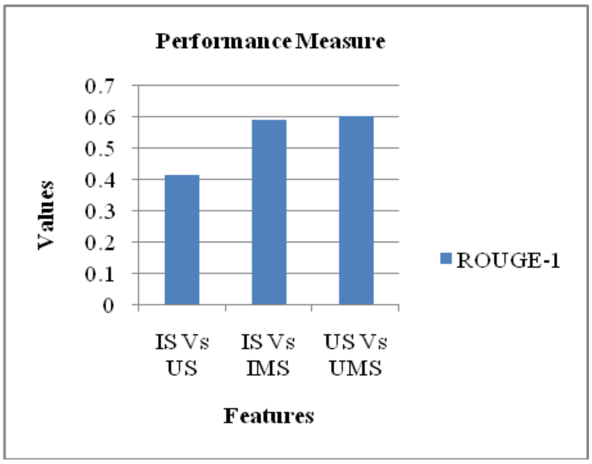

Fig 6: Performance Measure Based on ROUGE-1 Score

5. CONCLUSION

The proposed system generates initial and update summary from multiple documents based on semantic analysis and relevant sentences are selected by applying LSA and different combinations of features. Relevancy is improved by employing S_(TSF-ISF) measure. The update summary generated using the proposed novel and topic relevance measure is compared with manual summary as well as with its initial summary and the result shows that the summary generated by the proposed system is efficient.

6. REFERENCES

[1] Su Jian Li, Wei Wang, Chen Wang, TAC 2008 Update Summarization Task, In Proceedings of Text Analysis Conference, NIST, Maryland, USA, November 2008.

[2] Bysani P., Bharat V., Varma V., Modeling Novelty and Feature combination using Support Vector Regression for Update Summarization, In Proceedings of the 7th International Conference on Natural Language Processing, India, 2007.

[3] Josef Steinberger, Karel Jezek, Update Summarization Based on Novel Topic Distribution, In Proceedings of the 9th ACM symposium on Document engineering, USA, 2009.

[4] Eric Wehrli , Pierre-Etienne Genest, Guy Lapalme Luka Nerima, A Symbolic Summarizer for the Update Task of TAC 2008, In Proceedings of the First Text Analysis Conference, Gaithersburg, Maryland, USA, 2008.

[5] Praveen Bysani, Vijay Bharat, Vasudeva Varma, Modeling Novelty and Feature Combination using Support Vector Regression for Update summarization, In Proceedings of 7th International Conference on Natural Language Processing, 2009.

[6] Ravindranath Chowdary C., Sreenivasa Kumar P., Update Summarizer using MMR Approach, In Proceedings of Text Analysis Conference, NIST, Maryland, USA, November 2008.

[7] Joe Hicklin, Cleve Moler, Peter Webb, JAMA: A Java Matrix Package, http://math.nist.gov/javanumerics/jama/

[8] Kogilavani A., Balasubramanie P., Clustering and Feature Specific Sentence Extraction Based Summarization of Multiple Documents, International Journal of Computer Science and Information Technology, Vol.2, No.4, August 2010.

[9] Pierre-Etienne Genest, Guy Lapalme, Luka Nerima, Eric Wehrli, A Symbolic Summarizer for the Update task of TAC 2008, In Proceedings of TAC, NIST, USA, 2008.

[10] Yihong Gong, Xin Liu, Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis, SIGIR 2001. USA.

[11] Lin C.Y., ROUGE: A package for automatic evaluation of summaries. In proceedings of the workshop on Text Summarization, Barcelona. ACL, 2004.