Latent Dirichlet Allocation Based Multi-Document Summarization *

Rachit Arora
Department of Computer Science and Engineering
Indian Institute of Technology Madras
Chennai - 600 036, India.
rachitar@cse.iitm.ernet.in

Balaraman Ravindran
Department of Computer Science and Engineering
Indian Institute of Technology Madras
Chennai - 600 036, India.
ravi@cse.iitm.ernet.in

ABSTRACT

Extraction based Multi-Document Summarization Algorithms consist of choosing sentences from the documents using some weighting mechanism and combining them into a summary. In this article we use Latent Dirichlet Allocation to capture the events being covered by the documents and form the summary with sentences representing these different events. Our approach is distinguished from existing approaches in that we use mixture models to capture the topics and pick up the sentences without paying attention to the details of grammar and structure of the documents. Finally we present the evaluation of the algorithms on the DUC 2002 Corpus multi-document summarization tasks using the ROUGE evaluator to evaluate the summaries. Compared to DUC 2002 winners, our algorithms gave significantly better ROUGE-1 recall measures.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis,Multi-Document Summarization*

Keywords

Latent Dirichlet Allocation, Multi-Document Summarization

1. INTRODUCTION

Multi-Document Summarization consists of computing the summary of a set of related documents such that they give the user a general view of the events in the documents. Let us assume we have a set of M related documents together in a corpus. These documents have a central theme or event which is the property of all the documents or the corpus. The documents have other sub-events which support or revolve around this central event. Thus together this central theme and the sub-events form the *topics* for the set of documents.

One way of approaching the task of multi-document summarization is to break the documents into these topics and then describe or represent these topics adequately in the summary. Thus we can view the documents as being composed of topics, which we have to infer, and the visible variables which are the words of documents are just means of expressing these topics, which is the data that we have. A sentence can be viewed as describing a single event, describing multiple events or maybe connecting different events of the document. Here we are dealing with extraction based summarization i.e. we are extracting the entire sentence from the document without any modification to it like removal or adding of words and combining sentences together in the summary.

Latent Dirichlet Allocation (LDA) [1] is a generative three-level hierarchical Bayesian probabilistic model for collections of discrete data such as text documents. The documents are modeled as a finite mixture over an underlying set of topics which, in turn, are modeled as an infinite mixture over an underlying set of topic probabilities. Thus in the context of text modeling, the topic probabilities provide an explicit representation of the documents.

We can view the LDA model as breaking down the collection of documents into *topics* by representing the document as a mixture of topics with a probability distribution representing the importance of the topic for that document. The topics in turn are represented as a mixture of words with a probability representing the importance of the word for that topic. Another way of looking at it is that LDA soft-clusters the words of the documents into these topics i.e. instead of doing hard clustering and assigning a word to one topic, it gives a probability of the word belonging to the topic. Thus in a way we can view the these documents as a three level hierarchical Bayesian model with the topics, their distribution and the Dirichlet parameters as latent variables and the words and the documents that they belong to as the only visible variables.

Another way of looking at LDA is by viewing the documents as a weighted mixture of topics i.e. the probability distribution of topics for each document are the weights of the

AND '08, July 24, 2008 Singapore

Symbol	Meaning
C	Corpus
D_k	The k^{th} Document
S_r	The r^{th} Sentence
T_j	The j^{th} Topic
W_i	The i^{th} Word
M	The number of documents
R	The number of sentences
K	The number of Topics
V	The vocabulary size

Table 1: List of Symbols

topics in that document such that the sum of the weights is 1. Similarly, the topics can be viewed as a weighted mixture of words with the probability distribution of the topic over the words as the weights. We will use this notion of LDA as a three level hierarchical Bayesian model and as a weight distribution inter-changeably to come up with an algorithm for multi-document summarization.

In the past multi-document summarization algorithms have been mostly about word alignment on the summaries, by using the term frequency and inverse-document frequency or some combination of other weighting mechanisms of the words in the documents. The sentences are then given measures using the weights of the words in some combination or other and using some similarity and anti-redundancy techniques [4] [12]. These weighting mechanisms give limited choice in terms of giving weights to the words and the sentences. Other multi-document summarizer's have taken a probabilistic approach by using mixture models [10]. On the other hand LDA, even though it was meant as a text-topic model, has found wide application in the field of image and video processing to capture objects and actions [5] [13]. LDA has found limited application in the fields of Information Retrieval and Statistical Natural Language Processing. In this we propose a novel approach of using LDA to capture the topics that the documents are based on. Using LDA we represent these documents as being composed of topics and use that as a central idea in forming multi-document summaries. Whereas other summarization algorithms weight the sentences *without capturing* the events, we weight the sentences by capturing these events that the documents are based on by using LDA. Also LDA and the summarization algorithms assume the documents to be "bag-of-words" and we don't involve the grammar. Thus the approach is purely statistical and the summarization algorithms don't involve the structure of the documents or of the sentences in terms of grammar and the meanings conveyed by the words.

In Section 2 we look at some of the algorithms and their derivations we have proposed for multi-document summarization and Section 3 talks about further modifications we have made to these algorithms. Section 4 talks about some of the problems and their work-arounds we faced for the multi-document summarization algorithms due to the limitations within the framework of LDA. Section 5 talks about the algorithm for computing the multi-document summary. Section 6 gives the Evaluation of our algorithms and Section 7 talks about the future work.

2. MULTI-DOCUMENT SUMMARIZATION ALGORITHMS

Using LDA we break down the set of documents into topics. LDA uses a Dirichlet distribution to represent these topics and under the Dirichlet distribution these variables are independent of each other. We make the assumption that the number of topics in LDA is the same as the number of topics or the number of events that will describe the corpus. We assume that the complete sentence of a document belongs to only one topic and thus calculate the probability that the sentence belongs to the topic.

For all the sentences S_r, r $\in$ {1,...,R} in the documents and all the Topics T_j, j $\in$ {1,...,K} we calculate the Probability of the Sentence S_r given the Topic T_j i.e. $P(S_r|T_j)$.Thus we are calculating the probability that the sentence S_r belongs or represents the topic T_j. Let the words of the sentence S_r be {W_1,W_2, ... W_q}.

2.1 Algorithm I - Simple Inference Based

In the Simple Inference Based Multi-Document Summarization algorithm we assume that a sentence S_r of document D_B represents a topic T_j if all the words of the sentence S_r belong to the topic T_j and that the topic belongs to the document D_B. Thus our assumption that a sentence can belong to only one topic is restricting us that even words of the sentence S_r that don't belong to the Topic T_j i.e. have a low probability value for $P(W_i|T_j)$ are forced to represent this topic.

$$P(S_r|T_j) = \prod_{W_i \in S_r} P(W_i|T_j) * P(T_j|D_B) * P(D_B) \qquad (1)$$

Where

- $P(S_r|T_j)$ stands for *probability that a sentence S_r represents Topic T_j.*
- $\prod_{W_i \in S_r} P(W_i|T_j)$ stands for *probability that the words of S_r belongs to Topic T_j.*
- $P(T_j|D_B)$ stands for *probability that Topic T_j belongs to Document D_B.*
- $P(D_B)$ stands for *probability of the document D_B.*

2.2 Algorithm II - Partial Generative

In this algorithm we assume that a sentence S_r of document D_B represents a topic T_j if all the words of the sentence S_r belong to the topic T_j and that the Document D_B *generates* the topic T_j.

$$P(S_r|T_j) = \prod_{W_i \in S_r} P(W_i|T_j) * P(D_B|T_j) * P(D_B) \qquad (2)$$

Where

- $P(S_r|T_j)$ stands for *probability that sentence S_r represents Topic T_j.*

- $\prod_{W_i \in S_r} P(W_i|T_j)$ stands for *probability that words of S_r belongs to Topic T_j.*
- P($D_B|T_j$) stands for *probability that the document D_B generates the Topic T_j.*
- P(D_B) stands for *probability of the document D_B.*

Using the Bayes rule we have

$$P(D_B|T_j) * P(T_j) = P(T_j|D_B) * P(D_B) \quad (3)$$

To calculate the P(T_j) we sum P($T_j|D_k$) over all the documents k ∈ {1,...,M}.

$$P(T_j) = \sum_{k=1}^{M} P(T_j|D_k) * P(D_k) \quad (4)$$

Using Equations 3 and 4 in Equation 2 we get

$$P(S_r|T_j) = \frac{P(T_j|D_B)P(D_B)}{\sum_{k=1}^{M} P(T_j|D_k) * P(D_k)} * \prod_{W_i \in S_r} P(W_i|T_j) \quad (5)$$

2.3 Algorithm III - Full Generative

In this algorithm we assume that the document D_B *generates* the topic T_j and the topic T_j *generates* the sentence S_r.

$$P(S_r|T_j) = P(D_B) * P(D_B|T_j) * P(T_j|S_r) \quad (6)$$

Where

- P($S_r|T_j$) stands for *probability that a sentence S_r represents Topic T_j.*
- P(D_B) stands for *probability of the Document D_B.*
- P($D_B|T_j$) stands for *probability that Document D_B generates Topic T_j.*
- P($T_j|S_r$) stands for *probability that Topic T_j generates words of sentence S_r.*

Using the Bayes Rule we have

$$P(T_j|S_r) * P(S_r) = P(S_r|T_j) * P(T_j) \quad (7)$$
$$P(D_B|T_j) * P(T_j) = P(T_j|D_B) * P(D_B) \quad (8)$$

Using Equations 7 and 8 in Equation 6, we get

$$P(S_r|T_j) = P(D_B) * \frac{P(T_j|D_B) * P(D_B)}{P(T_j)} * \frac{P(S_r|T_j) * P(T_j)}{P(S_r)} \quad (9)$$

Now the sentence S_r consists of the words {W_1,W_2, ... W_q}, thus the topic T_j *generates* the sentence S_r only if the Topic T_j generates all the words the {W_1,W_2, ... W_q}

$$P(S_r|T_j) = P^2(D_B) * P(T_j|D_B) * \frac{P(\prod_{W_i \in S_r} W_i|T_j)}{P(\prod_{W_i \in S_r} W_i)} \quad (10)$$

$$P(S_r|T_j) = P^2(D_B) * P(T_j|D_B) * \frac{\prod_{W_i \in S_r} P(W_i|T_j)}{\prod_{W_i \in S_r} P(W_i)} \quad (11)$$

To calculate P(W_i) we sum over all the possible Topics j ∈ {1,...,K} and Documents k ∈ {1,...,M}.

$$P(W_i) = \sum_{k=1}^{M} \sum_{j=1}^{K} P(W_i|T_j) * P(T_j|D_k) * P(D_k) \quad (12)$$

Thus we get

$$P(S_r|T_j) = \frac{P^2(D_B) * P(T_j|D_B) * \prod_{W_i \in S_r} P(W_i|T_j)}{\prod_{W_i \in S_r} \sum_{k=1}^{M} \sum_{j=1}^{K} P(W_i|T_j)P(T_j|D_k)P(D_k)} \quad (13)$$

3. MODIFIED ALGORITHMS

Since P($W_i|T_j$) < 1, using the product of the *probability of word i belonging to sentence* i.e. $\prod_{W_i \in S_r} P(W_i|T_j)$ will penalize longer sentences. Let us say we have two sentences S_1 and S_2 of such that they have **n** words in common. Let the length of S_1 be n and that of S_2 be n+1. Thus we have

$$\begin{aligned} \prod_{i=1}^{n} P(W_i|T_j) &> \prod_{i=1}^{n} P(W_i|T_j) * P(W_{n+1}|T_j) \\ &OR \\ P(S_1|T_j) &> P(S_2|T_j) \end{aligned} \quad (14)$$

This will hold for all sentences and all topics, as a result the summary will consist of the shortest sentences in the documents.

Recall that another way of looking at LDA is by viewing the probability measures as a weight mixture, whose weights sum up to 1. The probability distribution of topics for each document are the weights of the topics in that document. Similarly, the topics can be viewed as a weighted mixture of words with the probability distribution of the topic over the words as the weights. Thus higher the probability, higher is the weight of the term in that item.

With this notion of LDA of interpreting the probability distributions given by LDA as a weight mixture, we replace the *product of probability of words* or the *product of weights of words* by *sum of probability of words* or *sum of weights of words.*

Thus for the simple inference based algorithm, we get the equation

$$\hat{P}(S_r|T_j) = \sum_{W_i \in S_r} P(W_i|T_j) * P(T_j|D_B) * P(D_B) \quad (15)$$

Longer sentences will have an advantage due to this, but the longer a sentence is, the more is its information content since we are removing all the function words and stemming the words. Thus the sentence contains only the "informative" words wrt to the corpus and thus longer the sentences, more is the number of informative words in the sentence and greater is its probability measure according to Equation 1.

However it is not always the case that the longer sentences will always have higher probability measure since each word has a probability measure associated with it, thus a shorter sentence containing words with higher probability measure might win over a longer sentence containing words with a lower probability measure.
We normalize the probability measure by dividing the measure by the length of the sentence S_r, we get the equation for the modified **simple inference based multi-document summarization algorithm** as

$$\hat{P}(S_r|T_j) = \frac{\sum_{W_i \in S_r} P(W_i|T_j) * P(T_j|D_B) * P(D_B)}{length(S_r)} \quad (16)$$

In this case S_2 is the longer sentence. Thus

$$\begin{aligned} \hat{P}(S_2|T_j) &> \hat{P}(S_1|T_j) \\ \frac{\sum_{i=1}^{n}(W_i|T_j) + P(W_{n+1}|T_j)}{n+1} &> \frac{\sum_{i=1}^{n} P(W_i|T_j)}{n} \\ P(W_{n+1}|T_j) &> \frac{\sum_{i=1}^{n} P(W_i|T_j)}{n} \end{aligned} \quad (17)$$

Thus in no way it is guaranteed that a longer sentence will be always chosen. Thus the sentence will only be chosen if the extra word in this sentence increases the current overall representation of the sentence towards the topic T_j.
Similarly the equation for the modified **partial generative multi-document summarization** algorithm is

$$\hat{P}(S_r|T_j) = \frac{P(T_j|D_B)P(D_B)}{\sum_{k=1}^{M} P(T_j|D_k) * P(D_k)} * \frac{\sum_{W_i \in S_r} P(W_i|T_j)}{length(S_r)} \quad (18)$$

We don't need to modify the full generative algorithm Equation 13 since the product term in the numerator is normalized by the product term in the denominator. Thus the equation for the **full generative multi-document summarization** algorithm is

$$P(S_r|T_j) = \frac{P^2(D_B) * P(T_j|D_B) * \prod_{W_i \in S_r} P(W_i|T_j)}{\prod_{W_i \in S_r} \sum_{k=1}^{M} \sum_{j=1}^{K} P(W_i|T_j)P(T_j|D_k)P(D_k)} \quad (19)$$

4. LIMITATIONS AND ENHANCEMENTS

The LDA model has some limitations that we have looked into and tried to find a work-around within the LDA framework in order to be used in the task of Multi-Document Summarization. Also we have made some enhancement to the LDA estimation that we describe over here.

4.1 Topic Probabilities

In LDA all the topics are not equi-probable. According to the mixture components and the α values, some topics might have a higher absolute probability value than others. Thus we calculate the probability of the topic by summing the mixture components over the documents.

$$P(T_j) = \sum_{k=1}^{M} P(T_j|D_k) * P(D_k) \quad (20)$$

4.2 Document Probabilities

In our algorithms we have considered the probability of the document D_B. One way of computing these probabilities is as given in LDA [1].

$$P(\mathbf{D}_m|\alpha,\beta) = \frac{\Gamma(\sum_{j=1}^{K}\alpha_j)}{\prod_{j=1}^{K}\Gamma(\alpha_j)} \int (\prod_{j=1}^{K} P(T_j|D_m)^{\alpha_j - 1}) (\prod_{n=1}^{N_m} \sum_{j=1}^{K} \prod_{i=1}^{V} (P(W_i|T_j)P(T_j|D_m))^{w_n^i}) d\theta_m \quad (21)$$

Alternatively if we know the probabilities of the documents a priori, we can use that in our calculations instead of inferring them from the LDA model.
In our case, for the sake of simplicity, we have assumed that all documents are equi-probable. Thus the *probability of the document* values don't make any difference during the calculation of the $P(S_r|T_j)$.

4.3 Number of Topics

To the best of our knowledge there is no known method of estimating a priori the number of topics in LDA to best represent the mixture components that represent the set of documents. Thus we estimate the LDA model for a range of topics and compute the summary for the corpus using this LDA model. Now we have two options

- Calculate the Log-Likelihood of the vocabulary using the estimated LDA Model and choose the LDA model and hence the summary that gives the best value for the Log-Likelihood.
- Use the LDA model to run inference on the summary and estimate its mixture components. Then calculate the Log-Likelihood of the summary using the summary's mixture components and choose the summary with the best value for the Log-Likelihood

We tried out both these methods and got very similar results, thus in the computation of the summary we can use either of the methods.
However this has a flaw, we are assuming very weak prior constraints on the number of topics. P(C|K), C being the corpus and K the topics, is just the likelihood term in the inference to P(K|C) and the prior P(K) might overwhelm this likelihood if we had a particularly strong preference for a smaller number of topics [3]. Models based on HDP [15] don't have this limitation.

4.4 LDA With Non Symmetric Dirichlet Prior

For the estimation of the LDA model we use Gibbs Sampling algorithm. However we have used a non-symmetric Dirichlet Prior α whose components $\{\alpha_1,...,\alpha_K\}$ we have to estimate.

We have done this estimation by Moment Matching which is a Maximum Likelihood Estimation algorithm which estimates the Dirichlet parameters by finding the density which matches the moments of the data.
The first two moments for the Dirichlet distribution are

$$E[p_k] = \frac{\alpha_k}{\sum_{i=1}^{K} \alpha_i} \tag{22}$$

$$E[p_k^2] = E[p_k] \frac{1 + \alpha_k}{1 + \sum_{i=1}^{K} \alpha_i} \tag{23}$$

Thus we get the summation of all the components of α as

$$\sum_{i=1}^{K} \alpha_i = \frac{E[p_k] - E[p_k^2]}{E[p_k^2] - E[p_k]^2} \tag{24}$$

By using a modified version of using all the parameters as suggested by Ronning [11] wherein we use the moments for all the components, we get

$$\sum_{k=1}^{K} \alpha_k = \exp(\frac{1}{K} \sum_{i=1}^{K} \log(\frac{E[p_i] - E[p_i^2]}{E[p_i^2] - E[p_i]^2})) \tag{25}$$

Using Equation 22 in Equation 25, we get

$$\alpha_k = E[p_k] \exp(\frac{1}{K} \sum_{i=1}^{K} \log(\frac{E[p_k] - E[p_k^2]}{E[p_k^2] - E[p_k]^2})) \tag{26}$$

p_k is a probability distribution containing M samples. Let
n^d - Number of words in the document d
n_k^d - Number of words assigned to topic k in document d

We get the moments for the data which is our set of documents as

$$E[p_k] = \frac{1}{M} \sum_{d=1}^{M} (\frac{n_k^d}{n^d}) \tag{27}$$

$$E[p_k^2] = \frac{1}{M} \sum_{d=1}^{M} (\frac{n_k^d}{n^d})^2 \tag{28}$$

Thus using Equations 27 and 28 which represent the moments for the data i.e. the documents in Equation 26 which represents the estimation of the components of Dirichlet parameter α we can estimate the $\{\alpha_1,\alpha_2,...,\alpha_K\}$ components of the non-symmetric Dirichlet parameter α

5. ALGORITHM

Here we present our LDA based Multi-Document Summarization Algorithms which are based on the Equations 16, 18 and 19.

1 Run the LDA Model for a fixed number of Topics K. We use Gibbs Sampling with a non-symmetric Dirichlet parameter α whose components $\{\alpha_1,\alpha_2,...,\alpha_K\}$ are estimated using Moment Matching. The α components are estimated after every iteration of Gibbs Sampling. We ran the 5 Markov Chains for 12000 iterations with 2000 iterations as BURNIN which were discarded and thereafter collected 100 samples at a lag of 100 iterations. To combine the samples from the 5 Markov chains we mapped the topics between the chains.

2 Thus we get the probability distributions $P(T_j|D_k)$ - Probability of Topic T_j given Document D_k and $P(W_i|T_j)$ - Probability of Word W_i given Topic T_j. Using these Probability Distributions and the **Equation 16 for Simple Inference Based Summarizer**, **Equation 18 for Partial Generative Summarizer** and **Equation 19 for Full Generative Summarizer**, we calculate the Probability of choosing the sentence S_r given the Topic T_j for all the sentences r $\in$ {1,...,R} and all the topic j $\in$ {1,...,K}.

3 Compute the Probability of the Topics as in Section 4.1. With this probability distribution over the topics choose a topic T_j by doing a multinomial sampling on the probability distribution.

4 For the topic T_j thus sampled, pick the sentence S_r with the highest probability given the Topic T_j i.e. the highest value of $P(S_r|T_j)$ and include it in the summary. If it is already included in the summary then pick the sentence with the next highest probability for this Topic T_j.

5 If the target Summary has reached the size then terminate the operation, else continue from Step 3.

6 After the summary has been computed, run the LDA Inference using Gibbs Sampling on the summary using the LDA Model estimated for the Corpus to get the mixture components for the summary. Using the mixture components of the summary calculate the Log-Likelihood of the Summary.

7 Choose the number of topics K and hence the summary with the best Log-Likelihood value for the summary.

6. EVALUATION

For the purpose of evaluation of our multi-document summarization algorithms we used the DUC 2002 Corpus dataset. The data was made up of 59 sets of Documents each containing on an average 10 documents and the candidate algorithms were required to make multi-document summary for each document set. The length of the summary was limited to 200 and 400 words. The candidate algorithms were supposed to be extraction based i.e. the sentences in the summary were supposed to be as they were in the documents, without any sort of modification. In addition for each document set we are given 2 model summaries against which the extracted candidate summary could be compared against.
We used the ROUGE Evaluator [7] which evaluates the summary using Ngram Co-Occurrence Statistics [8] and using Longest Common Subsequence and Skip-Bigram Statistics [9]. For evaluation using ROUGE we have used two methods, keeping stop-words and removing stop-words from both model summaries and candidate summary. In both the cases Porter Stemmer was used to stem the words to their root

ROUGE Setting	GISTEXTER	WSRSE	Inference Based	Partial Generative	Full Generative
Length = 200	0.48671	0.48694	0.55317	0.55613	0.51993
StopWords Kept	0.46198 - 0.51153	0.46000 - 0.51294	0.53955 - 0.56810	0.54065 - 0.57279	0.50508 - 0.53587
Length = 200	0.39526	0.40060	0.45299	0.45625	0.40549
StopWords Removed	0.36437 - 0.42729	0.37202 - 0.42910	0.43247 - 0.47433	0.43533 - 0.47703	0.38382 - 0.42837
Length = 400	0.56327	0.58006	0.61048	0.60775	0.58037
StopWords Kept	0.54737 - 0.57991	0.55579 - 0.60231	0.59823 - 0.62310	0.59683 - 0.61868	0.57036 - 0.59153
Length = 400	0.46669	0.48511	0.50674	0.50198	0.46232
StopWords Removed	0.44647 - 0.48815	0.45952 - 0.51206	0.49015 - 0.52405	0.48571 - 0.51752	0.44650 - 0.47856

Table 2: Evaluation on the DUC 2002 Corpus. The top value is the ROUGE-1 Recall Measure and the below value is its 95% Confidence Interval.

form. We have calculated the ROUGE scores separately for 200 and 400 length summary as we want to even see the effect of the length of the summary on the quality of the summary. We are mainly interested in the ROUGE-1 Recall score, which uses unigram statistics, since the precision scores can be manipulated by adjusting the length of the candidate summary [8]. Also since there were 2 model summaries for each document set, we have used the average score for each document set.

In the ROUGE settings we have used Porter Stemmer to stem the words to their root form in our computed summary and the model summaries given. We have evaluated the summary with removing stop-words and without removing stop-words.

We compare the results of our algorithms against the top two algorithms of the DUC2002 Multi-Document Summarization task, "Generating Single and Multi-Document Summaries with GISTEXTER" (GISTEXTER) [4] and "Writing Style Recognition and Sentence Extraction" (WSRSE) [12] in terms of ROUGE-1 Recall Measures. We also take a look at the 95% Confidence Interval.

We see that all the 3 LDA based algorithms perform better than the highest submissions of DUC 2002 multi-document summarization task (GISTEXTER and WSRSE). This holds for the summaries of length 200 and 400 words, thus showing that the algorithms work irrespective of the size of the summary to be computed. Also the lower bound of the 95% Confidence Interval for the Recall-1 measure for **Inference Based** and **Partial Generative** is higher than the upper bound of the 95% Confidence Interval of both GISTEXTER and WSRSE, thus showing that the two LDA based algorithms are statistically better.

Among the 3 LDA based algorithms, the **Inference Based** and **Partial Generative** outperform the **Full Generative** algorithm. There is little to choose between **Inference Based** and **Partial Generative** algorithms, however the latter performs better when summary size is small and the former performs better when the summary size is big.

7. CONCLUSION AND FUTURE WORK

In this we have shown a novel way of approaching the task of multi-document summarization by using LDA. We can use the mixture-models to explicitly represent the documents as topics or events and use these topics as the basis of choosing the sentences from the documents to form the summary. The performance of this approach on the DUC 2002 Multi-Document Summarization tasks shows statistically significant improvement over other summarization algorithms in terms of the ROUGE-1 recall measures.

We can extend the basic idea to other topic models like "Pachinko allocation: DAG-structured mixture models of topic correlations" (PAM) [6] and "Mixtures of hierarchical topics with Pachinko allocation" (HPAM) [2], in which we can estimate the number of topics as given in Section 4.3 here, or use it with "Hierarchical Dirichlet Processes" (HDP) [15] and "Nonparametric Bayes Pachinko Allocation" (NBP) [14] in which the number of topics is inferred within the model.

PAM captures the correlations among the topics which reflects which events in the documents are related to each other and which events are independent. This can be used in multi-document summarization to reduce redundancy by picking only one topic from a list of similar topics. On the other hand we can use it in the matters of coherence and cohesion by keeping sentences of correlated topics together in the summary.

HPAM on the other hand captures correlations among topics and also builds a hierarchy of topics. The topics higher up in the hierarchy represent topics that are more general in context to the documents whereas topics that are lower represent a more specific view of the events. Thus we can use this hierarchy to form the summary by using sentences that represent higher, more general topics and going into its specific topics only if the situation needs.

8. REFERENCES

[1] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of machine Learning Research 3*, pages 993–1022, 2003.

[2] W. L. David Mimno and A. McCallum. Mixtures of hierarchical topics with pachinko allocation. *Proceedings of the 24th international conference on Machine learning*, 24:633–640, 2007.

[3] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(1):5228–5235, 2004.

[4] S. M. Harabagiu and F. Lacatusu. Generating single and multi-document summaries with gistexter. *In Proceedings of the DUC 2002*, pages 30–38, 2002.

[5] F.-F. Li and P. Perona. A bayesian hierarchical model for learning natural scene categories. *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2:524–531, 2005.

[6] W. Li and A. McCallum. Pachinko allocation:

Dag-structured mixture models of topic correlations. *Proceedings of the 23rd international conference on Machine learning*, 23:577–584, 2006.

[7] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. *Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004*, 2004.

[8] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. *Proceedings of the Human Technology Conference 2003 (HLT-NAACL-2003)*, 2003.

[9] C.-Y. Lin and F. J. Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, 2004.

[10] S. R. M. Saravanan and B. Ravindran. A probabilistic approach to multi-document summarization for generating a tiled summary. *Proceedings of the Sixth International Conference on Computational Intelligence and Multimedia Applications*, 6:167 – 172, 2005.

[11] G. Ronning. Maximum-likelihood estimation of dirichlet distribution. *Journal of Statistical Computation and Simulation*, 32(4):215–221, 1989.

[12] H. van Halteren. Writing style recognition and sentence extraction. *In U. Hahn and D. Harman (Eds.), Proceedings of the workshop on automatic summarization*, pages 66–70, 2002.

[13] X. Wang and E. Grimson. Spatial latent dirichlet allocation. *Proceedings of Neural Information Processing Systems Conference (NIPS) 2007*, 2007.

[14] A. M. Wei Li and D. Blei. Nonparametric bayes pachinko allocation. *Proceedings of Conference on Uncertainty in Artificial Intelligence*, 2007.

[15] M. B. Y.W. Teh, M.I. Jordan and D. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association,*, 101(476):1566–1581, 2006.