A Hybrid Hierarchical Model for Multi-Document Summarization

Asli Celikyilmaz
Computer Science Department
University of California, Berkeley
asli@eecs.berkeley.edu

Dilek Hakkani-Tur
International Computer Science Institute
Berkeley, CA
dilek@icsi.berkeley.edu

Abstract

Scoring sentences in documents given abstract summaries created by humans is important in extractive multi-document summarization. In this paper, we formulate extractive summarization as a two step learning problem building a generative model for pattern discovery and a regression model for inference. We calculate scores for sentences in document clusters based on their latent characteristics using a hierarchical topic model. Then, using these scores, we train a regression model based on the lexical and structural characteristics of the sentences, and use the model to score sentences of new documents to form a summary. Our system advances current state-of-the-art improving ROUGE scores by ∼7%. Generated summaries are less redundant and more coherent based upon manual quality evaluations.

1 Introduction

Extractive approach to multi-document summarization (MDS) produces a summary by selecting sentences from original documents. Document Understanding Conferences (DUC), now TAC, fosters the effort on building MDS systems, which take document clusters (documents on a same topic) and description of the desired summary focus as input and output a word length limited summary. Human summaries are provided for training summarization models and measuring the performance of machine generated summaries.

Extractive summarization methods can be classified into two groups: supervised methods that rely on provided document-summary pairs, and unsupervised methods based upon properties derived from document clusters. Supervised methods treat the summarization task as a classification/regression problem, e.g., (Shen et al., 2007; Yeh et al., 2005). Each candidate sentence is classified as summary or non-summary based on the features that they pose and those with highest scores are selected. Unsupervised methods aim to score sentences based on semantic groupings extracted from documents, e.g., (Daumé III and Marcu, 2006; Titov and McDonald, 2008; Tang et al., 2009; Haghighi and Vanderwende, 2009; Radev et al., 2004; Branavan et al., 2009), etc. Such models can yield comparable or better performance on DUC and other evaluations, since representing documents as topic distributions rather than bags of words diminishes the effect of lexical variability. To the best of our knowledge, there is no previous research which utilizes the best features of both approaches for MDS as presented in this paper.

In this paper, we present a novel approach that formulates MDS as a prediction problem based on a two-step hybrid model: a generative model for hierarchical topic discovery and a regression model for inference. We investigate if a hierarchical model can be adopted to discover salient characteristics of sentences organized into hierarchies utilizing human generated summary text.

We present a probabilistic topic model on sentence level building on hierarchical Latent Dirichlet Allocation (hLDA) (Blei et al., 2003a), which is a generalization of LDA (Blei et al., 2003b). We construct a hybrid learning algorithm by extracting salient features to characterize summary sentences, and implement a regression model for inference (Fig.3). Contributions of this work are:
− construction of hierarchical probabilistic model designed to discover the topic structures of all sentences. Our focus is on identifying similarities of candidate sentences to summary sentences using a novel tree based sentence scoring algorithm, concerning topic distributions at different levels of the discovered hierarchy as described in § 3 and § 4,
− representation of sentences by meta-features to

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 815–824,
Uppsala, Sweden, 11-16 July 2010.

characterize their candidacy for inclusion in summary text. Our aim is to find features that can best represent summary sentences as described in § 5,
– implementation of a feasible inference method based on a regression model to enable scoring of sentences in test document clusters without retraining, (which has not been investigated in generative summarization models) described in § 5.2.

We show in § 6 that our hybrid summarizer achieves comparable (if not better) ROUGE score on the challenging task of extracting the summaries of multiple newswire documents. The human evaluations confirm that our hybrid model can produce coherent and non-redundant summaries.

2 Background and Motivation

There are many studies on the principles governing multi-document summarization to produce coherent and semantically relevant summaries. Previous work (Nenkova and Vanderwende, 2005; Conroy et al., 2006), focused on the fact that frequency of words plays an important factor. While, earlier work on summarization depend on a word score function, which is used to measure sentence rank scores based on (semi-)supervised learning methods, recent trend of purely data-driven methods, (Barzilay and Lee, 2004; Daumé III and Marcu, 2006; Tang et al., 2009; Haghighi and Vanderwende, 2009), have shown remarkable improvements. Our work builds on both methods by constructing a hybrid approach to summarization.

Our objective is to discover from document clusters, the latent topics that are organized into hierarchies following (Haghighi and Vanderwende, 2009). A hierarchical model is particularly appealing to summarization than a "flat" model, e.g. LDA (Blei et al., 2003b), in that one can discover "abstract" and "specific" topics. For instance, discovering that "baseball" and "football" are both contained in an abstract class "sports" can help to identify summary sentences. It follows that summary topics are commonly shared by many documents, while specific topics are more likely to be mentioned in rather a small subset of documents.

Feature based learning approaches to summarization methods discover salient features by measuring similarity between candidate sentences and summary sentences (Nenkova and Vanderwende, 2005; Conroy et al., 2006). While such methods are effective in extractive summarization, the fact that some of these methods are based on greedy algorithms can limit the application areas. Moreover, using information on the hidden semantic structure of document clusters would improve the performance of these methods.

Recent studies focused on the discovery of latent topics of document sets in extracting summaries. In these models, the challenges of inferring topics of test documents are not addressed in detail. One of the challenges of using a previously trained topic model is that the new document might have a totally new vocabulary or may include many other specific topics, which may or may not exist in the trained model. A common method is to re-build a topic model for new sets of documents (Haghighi and Vanderwende, 2009), which has proven to produce coherent summaries. An alternative yet feasible solution, presented in this work, is building a model that can summarize new document clusters using characteristics of topic distributions of training documents. Our approach differs from the early work, in that, we combine a generative hierarchical model and regression model to score sentences in new documents, eliminating the need for building a generative model for new document clusters.

3 Summary-Focused Hierarchical Model

Our MDS system, hybrid hierarchical summarizer, **HybHSum**, is based on an hybrid learning approach to extract sentences for generating summary. We discover hidden topic distributions of sentences in a given document cluster along with provided summary sentences based on hLDA described in (Blei et al., 2003a)[1]. We build a summary-focused hierarchical probabilistic topic model, sumHLDA, for each document cluster at sentence level, because it enables capturing expected topic distributions in given sentences directly from the model. Besides, document clusters contain a relatively small number of documents, which may limit the variability of topics if they are evaluated on the document level. As described in § 4, we present a new method for scoring candidate sentences from this hierarchical structure.

Let a given document cluster D be represented with sentences O=$\{o_m\}_{m=1}^{|O|}$ and its corresponding human summary be represented with sentences S=$\{s_n\}_{n=1}^{|S|}$. All sentences are comprised of words $V = \{w_1, w_2, ..w_{|V|}\}$ in $\{O \cup S\}$.

[1]Please refer to (Blei et al., 2003b) and (Blei et al., 2003a) for details and demonstrations of topic models.

Summary hLDA (sumHLDA): The hLDA represents distribution of topics in sentences by organizing topics into a tree of a fixed depth L (Fig.1.a). Each candidate sentence o_m is assigned to a path c_{o_m} in the tree and each word w_i in a given sentence is assigned to a hidden topic z_{o_m} at a level l of c_{o_m}. Each node is associated with a topic distribution over words. The sampler method alternates between choosing a new path for each sentence through the tree and assigning each word in each sentence to a topic along that path. The structure of tree is learnt along with the topics using a nested Chinese restaurant process (nCRP) (Blei et al., 2003a), which is used as a prior.

The nCRP is a stochastic process, which assigns probability distributions to infinitely branching and infinitely deep trees. In our model, nCRP specifies a distribution of words into paths in an L-level tree. The assignments of sentences to paths are sampled sequentially: The first sentence takes the initial L-level path, starting with a single branch tree. Later, mth subsequent sentence is assigned to a path drawn from the distribution:

$$\begin{aligned} p(path_{old}, c|m, m_c) &= \frac{m_c}{\gamma+m-1} \\ p(path_{new}, c|m, m_c) &= \frac{\gamma}{\gamma+m-1} \end{aligned} \quad (1)$$

$path_{old}$ and $path_{new}$ represent an existing and novel (branch) path consecutively, m_c is the number of previous sentences assigned to path c, m is the total number of sentences seen so far, and γ is a hyper-parameter which controls the probability of creating new paths. Based on this probability each node can branch out a different number of child nodes proportional to γ. Small values of γ suppress the number of branches.

Summary sentences generally comprise abstract concepts of the content. With sumHLDA we want to capture these abstract concepts in candidate sentences. The idea is to represent each path shared by similar candidate sentences with representative summary sentence(s). We let summary sentences share existing paths generated by similar candidate sentences instead of sampling new paths and influence the tree structure by introducing two separate hyper-parameters for nCRP prior:

- if a summary sentence is sampled, use $\gamma = \gamma_s$,
- if a candidate sentence is sampled, use $\gamma = \gamma_o$.

At each node, we let summary sentences sample a path by choosing only from the existing children of that node with a probability proportional to the number of other sentences assigned to that child. This can be achieved by using a small value for γ_s ($0 < \gamma_s \lll 1$). We only let candidate sentences to have an option of creating a new child node with a probability proportional to γ_o. By choosing $\gamma_s \lll \gamma_o$ we suppress the generation of new branches for summary sentences and modify the γ of nCRP prior in Eq.(1) using γ_s and γ_o hyper-parameters for different sentence types. In the experiments, we discuss the effects of this modification on the hierarchical topic tree.

The following is the generative process for sumHLDA used in our `HybHSum` :

(1) For each topic $k \in T$, sample a distribution $\beta_k \backsim$ Dirichlet(η).

(2) For each sentence $d \in \{O \cup S\}$,

(a) if $d \in O$, draw a path $c_d \backsim$ nCRP(γ_o), else if $d \in S$, draw a path $c_d \backsim$ nCRP(γ_s).

(b) Sample L-vector θ_d mixing weights from Dirichlet distribution $\theta_d \sim Dir(\alpha)$.

(c) For each word n, choose: (*i*) level $z_{d,n}|\theta_d$ and (*ii*) word $w_{d,n}|\{z_{d,n}, c_d, \beta\}$

Given sentence d, θ_d is a vector of topic proportions from L dimensional Dirichlet parameterized by α (distribution over levels in the tree.) The nth word of d is sampled by first choosing a level $z_{d,n} = l$ from the discrete distribution θ_d with probability $\theta_{d,l}$. Dirichlet parameter η and γ_o control the size of tree effecting the number of topics. (Small values of γ_s do not effect the tree.) Large values of η favor more topics (Blei et al., 2003a).

Model Learning: Gibbs sampling is a common method to fit the hLDA models. The aim is to obtain the following samples from the posterior of: (*i*) the latent tree T, (*ii*) the level assignment **z** for all words, (*iii*) the path assignments **c** for all sentences conditioned on the observed words **w**.

Given the assignment of words **w** to levels **z** and assignments of sentences to paths **c**, the expected posterior probability of a particular word w at a given topic **z**=l of a path **c**=c is proportional to the number of times w was generated by that topic:

$$p(w|\mathbf{z}, \mathbf{c}, \mathbf{w}, \eta) \propto n_{(\mathbf{z}=l, \mathbf{c}=c, \mathbf{w}=w)} + \eta \quad (2)$$

Similarly, posterior probability of a particular topic z in a given sentence d is proportional to number of times z was generated by that sentence:

$$p(z|\mathbf{z}, \mathbf{c}, \alpha) \propto n_{(\mathbf{c}=c_d, \mathbf{z}=l)} + \alpha \quad (3)$$

$n_{(.)}$ is the count of elements of an array satisfying the condition. Note from Eq.(3) that two sentences d_1 and d_2 on the same path **c** would have

different words, and hence different posterior topic probabilities. Posterior probabilities are normalized with total counts and their hyperparameters.

4 Tree-Based Sentence Scoring

The sumHLDA constructs a hierarchical tree structure of candidate sentences (per document cluster) by positioning summary sentences on the tree. Each sentence is represented by a path in the tree, and each path can be shared by many sentences. The assumption is that sentences sharing the same path should be more similar to each other because they share the same topics. Moreover, if a path includes a summary sentence, then candidate sentences on that path are more likely to be selected for summary text. In particular, the similarity of a candidate sentence o_m to a summary sentence s_n sharing the same path is a measure of strength, indicating how likely o_m is to be included in the generated summary (Algorithm 1):

Let c_{o_m} be the path for a given o_m. We find summary sentences that share the same path with o_m via: $M = \{s_n \in S | c_{s_n} = c_{o_m}\}$. The score of each sentence is calculated by similarity to the best matching summary sentence in M:

$$score(o_m) = \max_{s_n \in M} sim(o_m, s_n) \quad (4)$$

If *M*=∅, then *score(*o_m*)*=∅. The efficiency of our similarity measure in identifying the best matching summary sentence, is tied to how expressive the extracted topics of our sumHLDA models are. Given path c_{o_m}, we calculate the similarity of o_m to each s_n, *n*=1..$|M|$ by measuring similarities on:

⋆ **sparse unigram distributions** (sim_1) at each topic l on c_{o_m}: similarity between $p(\mathbf{w}_{o_m,l}|\mathbf{z}_{o_m} = l, c_{o_m}, v_l)$ and $p(\mathbf{w}_{s_n,l}|\mathbf{z}_{s_n} = l, c_{o_m}, v_l)$

⋆⋆ **distributions of topic proportions** (sim_2); similarity between $p(\mathbf{z}_{o_m}|c_{o_m})$ and $p(\mathbf{z}_{s_n}|c_{o_m})$.

− ***sim***$_1$***:*** We define two sparse (discrete) unigram distributions for candidate o_m and summary s_n at each node l on a vocabulary identified with words generated by the topic at that node, $v_l \subset V$. Given $\mathbf{w}_{o_m} = \{w_1, ..., w_{|o_m|}\}$, let $\mathbf{w}_{o_m,l} \subset \mathbf{w}_{o_m}$ be the set of words in o_m that are generated from topic $\mathbf{z}_{o_m}$ at level l on path c_{o_m}. The discrete unigram distribution $p_{o_m l} = p(\mathbf{w}_{o_m,l}|\mathbf{z}_{o_m} = l, c_{o_m}, v_l)$ represents the probability over all words v_l assigned to topic $\mathbf{z}_{o_m}$ at level l, by sampling only for words in $\mathbf{w}_{o_m,l}$. Similarly, $p_{s_n,l} = p(\mathbf{w}_{s_n,l}|\mathbf{z}_{s_n}, c_{o_m}, v_l)$ is the probability of words $\mathbf{w}_{s_n}$ in s_n of the same topic. The probability of each word in $p_{o_m,l}$ and $p_{s_n,l}$ are obtained using Eq. (2) and then normalized (see Fig.1.b).

Algorithm 1 Tree-Based Sentence Scoring

```
Given tree T from sumHLDA, candidate and summary
  sentences:  O = {o_1, ..., o_m} , S = {s_1, ..., s_n}
for sentences m ← 1, ..., |O| do
    - Find path c_{o_m} on tree T and summary sentences
    on path c_{o_m}: M = {s_n ∈ S | c_{s_n} = c_{o_m}}
    for summary sentences n ← 1, ..., |M| do
        - Find score(o_m)=max_{s_n} sim(o_m, s_n),
        where sim(o_m, s_n) = sim_1 * sim_2
        using Eq.(7) and Eq.(8)
    end for
end for
Obtain scores Y = {score(o_m)}_{m=1}^{|O|}
```

The similarity between $p_{o_m,l}$ and $p_{s_n,l}$ is obtained by first calculating the divergence with *information radius- IR* based on Kullback-Liebler(KL) divergence, *p*=$p_{o_m,l}$, *q*=$p_{s_n,l}$:

$$IR_{c_{o_m},l}(p_{o_m,l}, p_{s_n,l}) = KL\left(p||\tfrac{p+q}{2}\right) + KL\left(q||\tfrac{p+q}{2}\right) \quad (5)$$

where, $KL(p||q) = \sum_i p_i \log \frac{p_i}{q_i}$. Then the divergence is transformed into a similarity measure (Manning and Schuetze, 1999):

$$W_{c_{o_m},l}(p_{o_m,l}, p_{s_n,l}) = 10^{-IR_{c_{o_m},l}(p_{o_m,l}, p_{s_n,l})} \quad (6)$$

IR is a measure of total divergence from the average, representing how much information is lost when two distributions p and q are described in terms of average distributions. We opted for *IR* instead of the commonly used *KL* because with *IR* there is no problem with infinite values since $\frac{p_i+q_i}{2} \neq 0$ if either $p_i \neq 0$ or $q_i \neq 0$. Moreover, unlike *KL*, *IR* is symmetric, i.e., *KL(p,q)≠KL(q,p)*.

Finally sim_1 is obtained by average similarity of sentences using Eq.(6) at each level of c_{o_m} by:

$$sim_1(o_m, s_n) = \frac{1}{L} \sum_{l=1}^{L} W_{c_{o_m},l}(p_{o_m,l}, p_{s_n,l}) * l \quad (7)$$

The similarity between $p_{o_m,l}$ and $p_{s_n,l}$ at each level is weighted proportional to the level l because the similarity between sentences should be rewarded if there is a specific word overlap at child nodes.

−***sim***$_2$***:*** We introduce another measure based on sentence-topic mixing proportions to calculate the concept-based similarities between o_m and s_n. We calculate the topic proportions of o_m and s_n, represented by $p_{z_{o_m}} = p(\mathbf{z}_{o_m}|c_{o_m})$ and $p_{z_{s_n}} = p(\mathbf{z}_{s_n}|c_{o_m})$ via Eq.(3). The similarity between the distributions is then measured with transformed IR

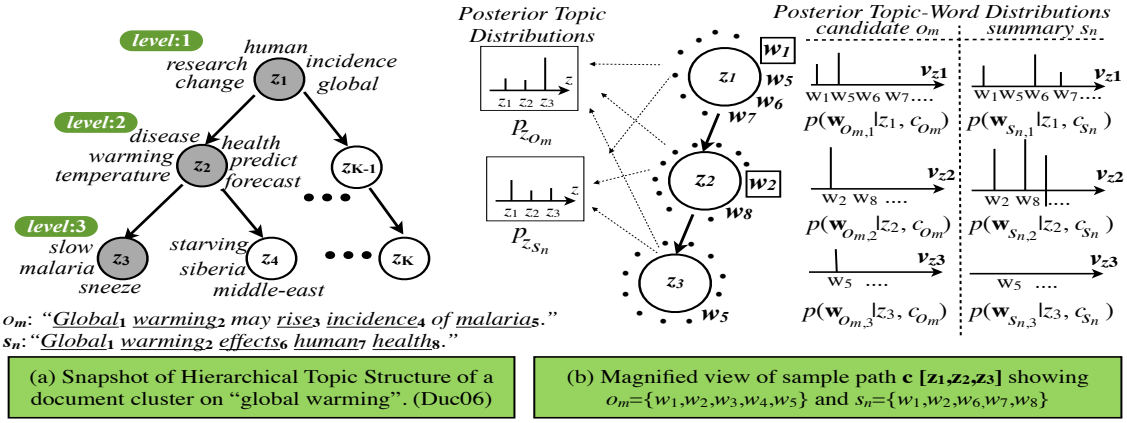

Figure 1: (a) A sample 3-level tree using sumHLDA. Each sentence is associated with a path c through the hierarchy, where each node $z_{l,c}$ is associated with a distribution over terms (Most probable terms are illustrated). (b) magnified view of a path (darker nodes) in (a). Distribution of words in given two sentences, a candidate (o_m) and a summary (s_n) using sub-vocabulary of words at each topic v_{z_l}. Discrete distributions on the left are topic mixtures for each sentence, $p_{z_{o_m}}$ and $p_{z_{s_n}}$.

as in Eq.(6) by:

$$sim_2(o_m, s_n) = 10^{-IR_{c_{o_m}}(p_{z_{o_m}}, p_{z_{s_n}})} \quad (8)$$

sim_1 provides information about the similarity between two sentences, o_m and s_n based on topic-word distributions. Similarly, sim_2 provides information on the similarity between the weights of the topics in each sentence. They jointly effect the sentence score and are combined in one measure:

$$sim(o_m, s_n) = sim_1(o_m, s_n) * sim_2(o_m, s_n) \quad (9)$$

The final score for a given o_m is calculated from Eq.(4). Fig.1.b depicts a sample path illustrating sparse unigram distributions of o_m and s_m at each level as well as their topic proportions, $p_{z_{o_m}}$, and $p_{z_{s_n}}$. In experiment 3, we discuss the effect of our tree-based scoring on summarization performance in comparison to a classical scoring method presented as our *baseline* model.

5 Regression Model

Each candidate sentence o_m, $m = 1..|O|$ is represented with a multi-dimensional vector of q features $\mathbf{f}_m = \{f_{m1}, ..., f_{mq}\}$. We build a regression model using sentence scores as output and selected salient features as input variables described below:

5.1 Feature Extraction

We compile our training dataset using sentences from different document clusters, which do not necessarily share vocabularies. Thus, we create n-gram *meta*-features to represent sentences instead of word n-gram frequencies:

(I) nGram Meta-Features (NMF): For each document cluster D, we identify most frequent (non-stop word) unigrams, i.e., $v_{freq} = \{w_i\}_{i=1}^{r} \subset V$, where r is a model parameter of number of most frequent unigram features. We measure observed unigram probabilities for each $w_i \in v_{freq}$ with $p_D(w_i) = n_D(w_i)/\sum_{j=1}^{|V|} n_D(w_j)$, where $n_D(w_i)$ is the number of times w_i appears in D and $|V|$ is the total number of unigrams. For any ith feature, the value is $f_{mi} = 0$, if given sentence does not contain w_i, otherwise $f_{mi} = p_D(w_i)$. These features can be extended for any n-grams. We similarly include bigram features in the experiments.

(II) Document Word Frequency Meta-Features (DMF): The characteristics of sentences at the document level can be important in summary generation. DMF identify whether a word in a given sentence is specific to the document in consideration or it is commonly used in the document cluster. This is important because summary sentences usually contain abstract terms rather than specific terms.

To characterize this feature, we re-use the r most frequent unigrams, i.e., $w_i \in v_{freq}$. Given sentence o_m, let d be the document that o_m belongs to, i.e., $o_m \in d$. We measure unigram probabilities for each w_i by $p(w_i \in o_m) = n_d(w_i \in o_m)/n_D(w_i)$, where $n_d(w_i \in o_m)$ is the number of times w_i appears in d and $n_D(w_i)$ is the number of times w_i appears in D. For any ith feature, the value is $f_{mi} = 0$, if given sentence does not contain w_i, otherwise $f_{mi} = p(w_i \in o_m)$. We also include bigram extensions of DMF features.

(III) Other Features (OF): Term frequency of sentences such as SUMBASIC are proven to be good predictors in sentence scoring (Nenkova and Vanderwende, 2005). We measure the average unigram probability of a sentence by: $p(o_m) = \sum_{w \in o_m} \frac{1}{|o_m|} P_D(w)$, where $P_D(w)$ is the observed unigram probability in the document collection D and $|o_m|$ is the total number of words in o_m. We use sentence bigram frequency, sentence rank in a document, and sentence size as additional features.

5.2 Predicting Scores for New Sentences

Due to the large feature space to explore, we chose to work with support vector regression (SVR) (Drucker et al., 1997) as the learning algorithm to predict sentence scores. Given training sentences $\{\mathbf{f}_m, y_m\}_{m=1}^{|O|}$, where $f_m = \{f_{m1}, ..., f_{mq}\}$ is a multi-dimensional vector of features and y_m=*score*$(o_m) \in \mathbb{R}$ are their scores obtained via Eq.(4), we train a regression model. In experiments we use non-linear Gaussian kernel for SVR. Once the SVR model is trained, we use it to predict the scores of n_{test} number of sentences in test (*unseen*) document clusters, $O_{test} = \{o_1, ...o_{|O_{test}|}\}$.

Our HybHSum captures the sentence characteristics with a regression model using sentences in different document clusters. At test time, this valuable information is used to score testing sentences.

Redundancy Elimination: To eliminate redundant sentences in the generated summary, we incrementally add onto the summary the highest ranked sentence o_m and check if o_m significantly repeats the information already included in the summary until the algorithm reaches word count limit. We use a word overlap measure between sentences normalized to sentence length. A o_m is discarded if its similarity to any of the previously selected sentences is greater than a threshold identified by a greedy search on the training dataset.

6 Experiments and Discussions

In this section we describe a number of experiments using our hybrid model on 100 document clusters each containing 25 news articles from DUC2005-2006 tasks. We evaluate the performance of HybHSum using 45 document clusters each containing 25 news articles from DUC2007 task. From these sets, we collected $\backsim$80K and $\backsim$25K sentences to compile training and testing data respectively. The task is to create max. 250 word long summary for each document cluster.

We use Gibbs sampling for inference in hLDA and sumHLDA. The hLDA is used to capture abstraction and specificity of words in documents (Blei et al., 2009). Contrary to typical hLDA models, to efficiently represent sentences in summarization task, we set ascending values for Dirichlet hyper-parameter η as the level increases, encouraging mid to low level distributions to generate as many words as in higher levels, e.g., for a tree of depth=3, $\eta = \{0.125, 0.5, 1\}$. This causes sentences share paths only when they include similar concepts, starting higher level topics of the tree. For SVR, we set $\epsilon = 0.1$ using the default choice, which is the inverse of the average of $\phi(\mathbf{f})^T \phi(\mathbf{f})$ (Joachims, 1999), dot product of kernelized input vectors. We use greedy optimization during training based on ROUGE scores to find best regularizer $C = \{10^{-1}..10^2\}$ using the Gaussian kernel.

We applied feature extraction of § 5.1 to compile the training and testing datasets. ROUGE is used for performance measure (Lin and Hovy, 2003; Lin, 2004), which evaluates summaries based on the maxium number of overlapping units between generated summary text and a set of human summaries. We use R-1 (recall against unigrams), R-2 (recall against bigrams), and R-SU4 (recall against skip-4 bigrams).

Experiment 1: sumHLDA Parameter Analysis: In sumHLDA we introduce a prior different than the standard nested CRP (nCRP). Here, we illustrate that this prior is practical in learning hierarchical topics for summarization task.

We use sentences from the human generated summaries during the discovery of hierarchical topics of sentences in document clusters. Since summary sentences generally contain abstract words, they are indicative of sentences in documents and should produce minimal amount of new topics (if not none). To implement this, in nCRP prior of sumHLDA, we use dual hyper-parameters and choose a very small value for summary sentences, $\gamma_s = 10e^{-4} \ll \gamma_o$. We compare the results to hLDA (Blei et al., 2003a) with nCRP prior which uses only one free parameter, γ. To analyze this prior, we generate a corpus of $\backsim$1300 sentences of a document cluster in DUC2005. We repeated the experiment for 9 other clusters of similar size and averaged the total number of generated topics. We show results for different values of γ and γ_o hyper-parameters and tree depths.

$\gamma = \gamma_o$	**0.1**			**1**			**10**		
depth	**3**	**5**	**8**	**3**	**5**	**8**	**3**	**5**	**8**
hLDA	3	5	8	41	267	1509	1522	4080	8015
sumHLDA	3	5	8	27	162	671	1207	3598	7050

Table 1: Average # of topics per document cluster from sumHLDA and hLDA for different γ and γ_o and tree depths. $\gamma_s = 10e^{-4}$ is used for sumHLDA for each depth.

Features	Baseline			HybHSum		
	R-1	R-2	R-SU4	R-1	R-2	R-SU4
NMF (1)	40.3	7.8	13.7	41.6	8.4	12.3
DMF (2)	41.3	7.5	14.3	41.3	8.0	13.9
OF (3)	40.3	7.4	13.7	**42.4**	8.0	14.4
(1+2)	41.5	7.9	14.0	41.8	8.5	14.5
(1+3)	40.8	7.5	13.8	41.6	8.2	14.1
(2+3)	40.7	7.4	13.8	**42.7**	**8.7**	**14.9**
(1+2+3)	41.4	8.1	13.7	**43.0**	9.1	**15.1**

Table 2: ROUGE results (with stop-words) on DUC2006 for different features and methods. Results in bold show statistical significance over baseline in corresponding metric.

As shown in Table 1, the nCRP prior for sumHLDA is more effective than hLDA prior in the summarization task. Less number of topics(nodes) in sumHLDA suggests that summary sentences share pre-existing paths and no new paths or nodes are sampled for them. We also observe that using $\gamma_o = 0.1$ causes the model to generate minimum number of topics (# of topics=depth), while setting $\gamma_o = 10$ creates excessive amount of topics. $\gamma_0 = 1$ gives reasonable number of topics, thus we use this value for the rest of the experiments. In experiment 3, we use both nCRP priors in HybHSum to analyze whether there is any performance gain with the new prior.

Experiment 2: Feature Selection Analysis

Here we test individual contribution of each set of features on our HybHSum (using sumHLDA). We use a **Baseline** by replacing the scoring algorithm of HybHSum with a simple cosine distance measure. The score of a candidate sentence is the cosine similarity to the maximum matching summary sentence. Later, we build a regression model with the same features as our HybHSum to create a summary. We train models with DUC2005 and evaluate performance on DUC2006 documents for different parameter values as shown in Table 2.

As presented in § 5, NMF is the bundle of frequency based meta-features on document cluster level, DMF is a bundle of frequency based meta-features on individual document level and OF represents sentence term frequency, location, and size features. In comparison to the baseline, OF has a significant effect on the ROUGE scores. In addition, DMF together with OF has shown to improve all scores, in comparison to baseline, on average by 10%. Although the NMF have minimal individual improvement, all these features can statistically improve R-2 without stop words by 12% (significance is measured by t-test statistics).

Experiment 3: ROUGE Evaluations

We use the following multi-document summarization models along with the Baseline presented in Experiment 2 to evaluate HybSumm.

⋆ **PYTHY :** (Toutanova et al., 2007) A state-of-the-art supervised summarization system that ranked first in overall ROUGE evaluations in DUC2007. Similar to HybHSum, human generated summaries are used to train a sentence ranking system using a classifier model.

⋆ **HIERSUM :** (Haghighi and Vanderwende, 2009) A generative summarization method based on topic models, which uses sentences as an additional level. Using an approximation for inference, sentences are greedily added to a summary so long as they decrease KL-divergence.

⋆ **HybFSum (Hybrid Flat Summarizer):** To investigate the performance of hierarchical topic model, we build another hybrid model using flat LDA (Blei et al., 2003b). In LDA each sentence is a superposition of all K topics with sentence specific weights, there is no hierarchical relation between topics. We keep the parameters and the features of the regression model of hierarchical HybHSum intact for consistency. We only change the sentence scoring method. Instead of the new tree-based sentence scoring (§ 4), we present a similar method using topics from LDA on sentence level. Note that in LDA the topic-word distributions ϕ are over entire vocabulary, and topic mixing proportions for sentences θ are over all the topics discovered from sentences in a document cluster. Hence, we define sim_1 and sim_2 measures for LDA using topic-word proportions ϕ (in place of discrete topic-word distributions from each level in Eq.2) and topic mixing weights θ in sentences (in place of topic proportions in Eq.3) respectively. Maximum matching score is calculated as same as in HybHSum.

⋆ **HybHSum**$_1$ and **HybHSum**$_2$: To analyze the effect of the new nCRP prior of sumHLDA on sum-

ROUGE	w/o stop words			w/ stop words		
	R-1	R-2	R-4	R-1	R-2	R-4
Baseline	32.4	7.4	10.6	41.0	9.3	15.2
PYTHY	35.7	8.9	**12.1**	42.6	11.9	16.8
HIERSUM	33.8	**9.3**	11.6	42.4	11.8	16.7
HybFSum	34.5	8.6	10.9	43.6	9.5	15.7
HybHSum_1	34.0	7.9	11.5	44.8	11.0	16.7
HybHSum_2	35.1	8.3	11.8	**45.6**	11.4	**17.2**

Table 3: ROUGE results of the best systems on DUC2007 dataset (best results are **bolded**.)

marization model performance, we build two different versions of our hybrid model: $\textbf{HybHSum}_1$ using standard hLDA (Blei et al., 2003a) and $\textbf{HybHSum}_2$ using our sumHLDA.

The ROUGE results are shown in Table 3. The HybHSum_2 achieves the best performance on R-1 and R-4 and comparable on R-2. When stop words are used the HybHSum_2 outperforms state-of-the-art by 2.5-7% except R-2 (with statistical significance). Note that R-2 is a measure of bigram recall and sumHLDA of HybHSum_2 is built on unigrams rather than bigrams. Compared to the HybFSum built on LDA, both $\text{HybHSum}_{1\&2}$ yield better performance indicating the effectiveness of using hierarchical topic model in summarization task. HybHSum_2 appear to be less redundant than HybFSum capturing not only common terms but also specific words in Fig. 2, due to the new hierarchical tree-based sentence scoring which characterizes sentences on deeper level. Similarly, $\text{HybHSum}_{1\&2}$ far exceeds baseline built on simple classifier. The results justify the performance gain by using our novel tree-based scoring method. Although the ROUGE scores for HybHSum_1 and HybHSum_2 are not significantly different, the sumHLDA is more suitable for summarization tasks than hLDA.

HybHSum_2 is comparable to (if not better than) fully generative HIERSUM. This indicates that with our regression model built on training data, summaries can be efficiently generated for test documents (suitable for online systems).

Experiment 4: Manual Evaluations

Here, we manually evaluate quality of summaries, a common DUC task. Human annotators are given two sets of summary text for each document set, generated from two approaches: best hierarchical hybrid HybHSum_2 and flat hybrid HybFSum models, and are asked to mark the better summary

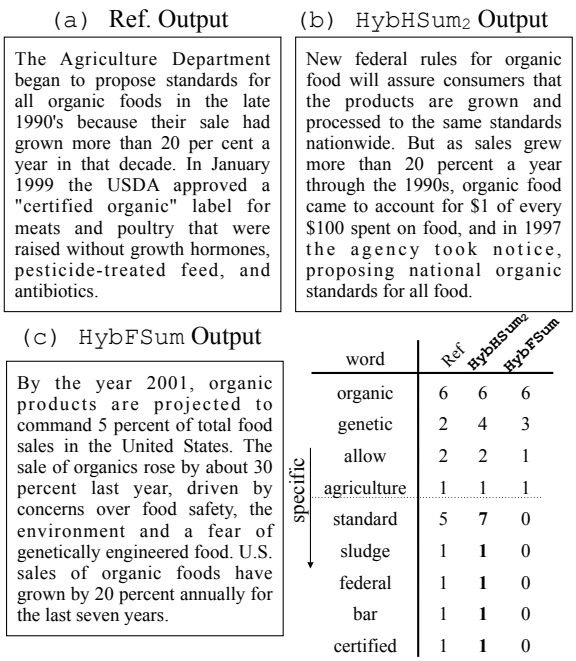

Figure 2: Example summary text generated by systems compared in Experiment 3. (Id:D0744 in DUC2007). Ref. is the human generated summary.

Criteria	HybFSum	HybHSum_2	Tie
Non-redundancy	26	44	22
Coherence	24	56	12
Focus	24	56	12
Responsiveness	30	50	12
Overall	24	66	2

Table 4: Frequency results of manual quality evaluations. Results are statistically significant based on t-test. Tie indicates evaluations where two summaries are rated equal.

according to five criteria: *non-redundancy* (which summary is less redundant), *coherence* (which summary is more coherent), *focus and readability* (content and not include unnecessary details), *responsiveness* and *overall* performance.

We asked 4 annotators to rate DUC2007 predicted summaries (45 summary pairs per annotator). A total of 92 pairs are judged and evaluation results in frequencies are shown in Table 4. The participants rated HybHSum_2 generated summaries more coherent and focused compared to HybFSum. All results in Table 4 are statistically significant (based on t-test on 95% confidence level.) indicating that HybHSum_2 summaries are rated significantly better.

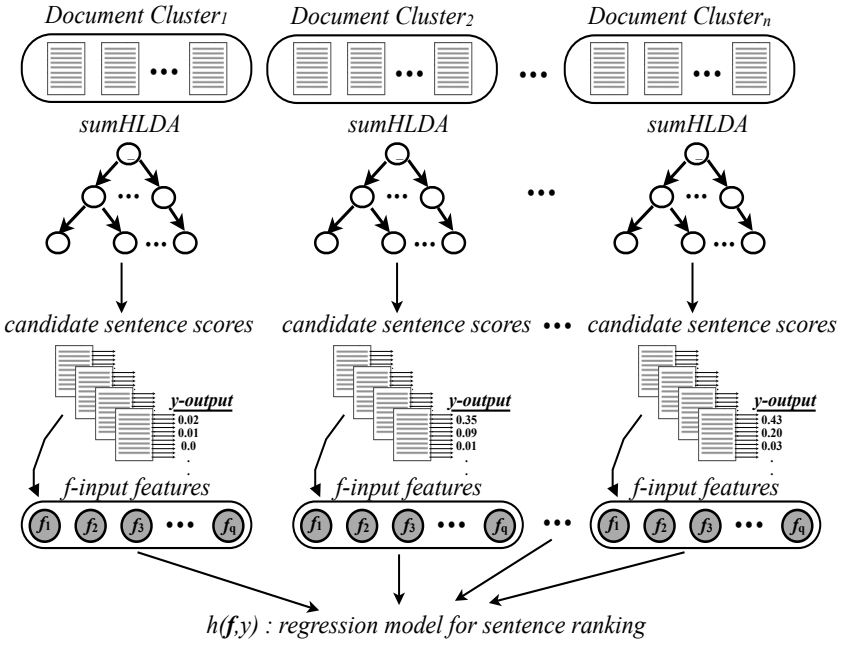

Figure 3: Flow diagram for Hybrid Learning Algorithm for Multi-Document Summarization.

7 Conclusion

In this paper, we presented a hybrid model for multi-document summarization. We demonstrated that implementation of a summary focused hierarchical topic model to discover sentence structures as well as construction of a discriminative method for inference can benefit summarization quality on manual and automatic evaluation metrics.

Acknowledgement

Research supported in part by ONR N00014-02-1-0294, BT Grant CT1080028046, Azerbaijan Ministry of Communications and Information Technology Grant, Azerbaijan University of Azerbaijan Republic and the BISC Program of UC Berkeley.

References

R. Barzilay and L. Lee. Catching the drift: Probabilistic content models with applications to generation and summarization. In *In Proc. HLT-NAACL'04*, 2004.

D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *In Neural Information Processing Systems [NIPS]*, 2003a.

D. Blei, T. Griffiths, and M. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. In *Journal of ACM*, 2009.

D. M. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. In *Jrnl. Machine Learning Research, 3:993-1022*, 2003b.

S.R.K. Branavan, H. Chen, J. Eisenstein, and R. Barzilay. Learning document-level semantic properties from free-text annotations. In *Journal of Artificial Intelligence Research*, volume 34, 2009.

J.M. Conroy, J.D. Schlesinger, and D.P. O'Leary. Topic focused multi-cument summarization using an approximate oracle score. In *In Proc. ACL'06*, 2006.

H. Daumé III and D. Marcu. Bayesian query focused summarization. In *Proc. ACL-06*, 2006.

H. Drucker, C.J.C. Burger, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. In *NIPS 9*, 1997.

A. Haghighi and L. Vanderwende. Exploring content models for multi-document summarization. In *NAACL HLT-09*, 2009.

T. Joachims. Making large-scale svm learning practical. In *In Advances in Kernel Methods - Support Vector Learning. MIT Press.*, 1999.

C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *In Proc. ACL Workshop on Text Summarization Branches Out*, 2004.

C.-Y. Lin and E.H. Hovy. Automatic evaluation of summaries using n-gram co-occurance statistics. In *Proc. HLT-NAACL, Edmonton, Canada*, 2003.

C. Manning and H. Schuetze. Foundations of statistical natural language processing. In *MIT Press. Cambridge, MA*, 1999.

A. Nenkova and L. Vanderwende. The impact of frequency on summarization. In *Tech. Report MSR-TR-2005-101, Microsoft Research, Redwood, Washington*, 2005.

D.R. Radev, H. Jing, M. Stys, and D. Tam. Centroid-based summarization for multiple documents. In *In Int. Jrnl. Information Processing and Management*, 2004.

D. Shen, J.T. Sun, H. Li, Q. Yang, and Z. Chen. Document summarization using conditional random fields. In *Proc. IJCAI'07*, 2007.

J. Tang, L. Yao, and D. Chens. Multi-topic based query-oriented summarization. In *SIAM International Conference Data Mining*, 2009.

I. Titov and R. McDonald. A joint model of text and aspect ratings for sentiment summarization. In *ACL-08:HLT*, 2008.

K. Toutanova, C. Brockett, M. Gamon, J. Jagarlamudi, H. Suzuki, and L. Vanderwende. The phthy summarization system: Microsoft research at duc 2007. In *Proc. DUC*, 2007.

J.Y. Yeh, H.-R. Ke, W.P. Yang, and I-H. Meng. Text summarization using a trainable summarizer and latent semantic analysis. In *Information Processing and Management*, 2005.