Exploring Content Models for Multi-Document Summarization

Aria Haghighi
UC Berkeley, CS Division
aria42@cs.berkeley.edu

Lucy Vanderwende
Microsoft Research
Lucy.Vanderwende@microsoft.com

Abstract

We present an exploration of generative probabilistic models for multi-document summarization. Beginning with a simple word frequency based model (Nenkova and Vanderwende, 2005), we construct a sequence of models each injecting more structure into the representation of document set content and exhibiting ROUGE gains along the way. Our final model, HIERSUM, utilizes a hierarchical LDA-style model (Blei et al., 2004) to represent content specificity as a hierarchy of topic vocabulary distributions. At the task of producing generic DUC-style summaries, HIERSUM yields state-of-the-art ROUGE performance and in pairwise user evaluation strongly outperforms Toutanova et al. (2007)'s state-of-the-art discriminative system. We also explore HIERSUM's capacity to produce multiple 'topical summaries' in order to facilitate content discovery and navigation.

1 Introduction

Over the past several years, there has been much interest in the task of multi-document summarization. In the common Document Understanding Conference (DUC) formulation of the task, a system takes as input a document set as well as a short description of desired summary focus and outputs a word length limited summary.[1] To avoid the problem of generating cogent sentences, many systems opt for an extractive approach, selecting sentences from the document set which best reflect its core content.[2]

There are several approaches to modeling document content: simple word frequency-based methods (Luhn, 1958; Nenkova and Vanderwende, 2005), graph-based approaches (Radev, 2004; Wan and Yang, 2006), as well as more linguistically motivated techniques (Mckeown et al., 1999; Leskovec et al., 2005; Harabagiu et al., 2007). Another strand of work (Barzilay and Lee, 2004; Daumé III and Marcu, 2006; Eisenstein and Barzilay, 2008), has explored the use of structured probabilistic topic models to represent document content. However, little has been done to directly compare the benefit of complex content models to simpler surface ones for generic multi-document summarization.

In this work we examine a series of content models for multi-document summarization and argue that LDA-style probabilistic topic models (Blei et al., 2003) can offer state-of-the-art summarization quality as measured by automatic metrics (see section 5.1) and manual user evaluation (see section 5.2). We also contend that they provide convenient building blocks for adding more structure to a summarization model. In particular, we utilize a variation of the hierarchical LDA topic model (Blei et al., 2004) to discover multiple specific 'subtopics' within a document set. The resulting model, HIERSUM (see section 3.4), can produce general summaries as well as summaries for any of the learned sub-topics.

2 Experimental Setup

The task we will consider is extractive multi-document summarization. In this task we assume a document collection $\mathcal{D}$ consisting of documents $D_1, \ldots, D_n$ describing the same (or closely related)

[1] In this work, we ignore the summary focus. Here, the word *topic* will refer to elements of our statistical model rather than summary focus.

[2] Note that sentence extraction does not solve the problem of selecting and ordering summary sentences to form a coherent narrative (Lapata, 2003).

Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL, pages 362–370,
Boulder, Colorado, June 2009.

set of events. Our task will be to propose a summary $\mathbf{S}$ consisting of sentences in $\mathcal{D}$ totaling at most L words.[3] Here as in much extractive summarization, we will view each sentence as a bag-of-words or more generally a bag-of-ngrams (see section 5.1). The most prevalent example of this data setting is document clusters found on news aggregator sites.

2.1 Automated Evaluation

For model development we will utilize the DUC 2006 evaluation set[4] consisting of 50 document sets each with 25 documents; final evaluation will utilize the DUC 2007 evaluation set (section 5).

Automated evaluation will utilize the standard DUC evaluation metric ROUGE (Lin, 2004) which represents recall over various n-grams statistics from a system-generated summary against a set of human-generated peer summaries.[5] We compute ROUGE scores with and without stop words removed from peer and proposed summaries. In particular, we utilize R-1 (recall against unigrams), R-2 (recall against bigrams), and R-SU4 (recall against skip-4 bigrams)[6]. We present R-2 without stop words in the running text, but full development results are presented in table 1. Official DUC scoring utilizes the jackknife procedure and assesses significance using bootstrapping resampling (Lin, 2004). In addition to presenting automated results, we also present a user evaluation in section 5.2.

3 Summarization Models

We present a progression of models for multi-document summarization. Inference details are given in section 4.

3.1 SumBasic

The SUMBASIC algorithm, introduced in Nenkova and Vanderwende (2005), is a simple effective procedure for multi-document extractive summarization. Its design is motivated by the observation that the relative frequency of a non-stop word in a document set is a good predictor of a word appearing in a human summary. In SUMBASIC, each sentence S is assigned a score reflecting how many high-frequency words it contains,

$$Score(S) = \sum_{w \in S} \frac{1}{|S|} P_{\mathcal{D}}(w) \quad (1)$$

where $P_{\mathcal{D}}(\cdot)$ initially reflects the observed unigram probabilities obtained from the document collection $\mathcal{D}$. A summary $\mathbf{S}$ is progressively built by adding the highest scoring sentence according to (1).[7]

In order to discourage redundancy, the words in the selected sentence are updated $P_{\mathcal{D}}^{new}(w) \propto P_{\mathcal{D}}^{old}(w)^2$. Sentences are selected in this manner until the summary word limit has been reached.

Despite its simplicity, SUMBASIC yields 5.3 R-2 without stop words on DUC 2006 (see table 1).[8] By comparison, the highest-performing ROUGE system at the DUC 2006 evaluation, SUMFOCUS, was built on top of SUMBASIC and yielded a 6.0, which is not a statistically significant improvement (Vanderwende et al., 2007).[9]

Intuitively, SUMBASIC is trying to select a summary which has sentences where most words have high likelihood under the document set unigram distribution. One conceptual problem with this objective is that it inherently favors repetition of frequent non-stop words despite the 'squaring' update. Ideally, a summarization criterion should be more recall oriented, penalizing summaries which omit moderately frequent document set words and quickly diminishing the reward for repeated use of word.

Another more subtle shortcoming is the use of the raw empirical unigram distribution to represent content significance. For instance, there is no distinction between a word which occurs many times in the same document or the same number of times across several documents. Intuitively, the latter word is more indicative of significant document set content.

3.2 KLSum

The KLSUM algorithm introduces a criterion for selecting a summary $\mathbf{S}$ given document collection $\mathcal{D}$,

$$\mathbf{S}^* = \min_{\mathbf{S}: words(\mathbf{S}) \leq L} KL(P_{\mathcal{D}} \| P_{\mathbf{S}}) \quad (2)$$

[3] For DUC summarization tasks, L is typically 250.

[4] http://www-nlpir.nist.gov/projects/duc/data.html

[5] All words from peer and proposed summaries are lowercased and stemmed.

[6] Bigrams formed by skipping at most two words.

[7] Note that sentence order is determined by the order in which sentences are selected according to (1).

[8] This result is presented as 0.053 with the official ROUGE scorer (Lin, 2004). Results here are scaled by 1,000.

[9] To be fair obtaining statistical significance in ROUGE scores is quite difficult.

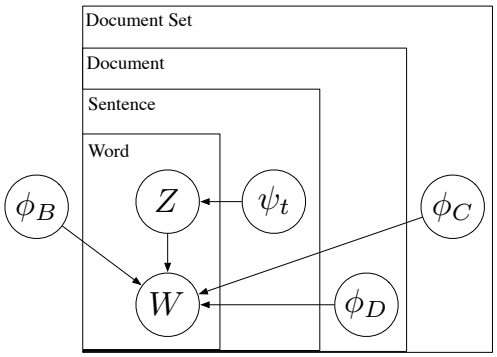


Figure 1: Graphical model depiction of TOPICSUM model (see section 3.3). Note that many hyperparameter dependencies are omitted for compactness.

where $P_{\mathbf{S}}$ is the empirical unigram distribution of the candidate summary $\mathbf{S}$ and $KL(P\|Q)$ represents the Kullback-Lieber (KL) divergence given by $\sum_w P(w) \log \frac{P(w)}{Q(w)}$.[10] This quantity represents the divergence between the true distribution P (here the document set unigram distribution) and the approximating distribution Q (the summary distribution). This criterion casts summarization as finding a set of summary sentences which closely match the document set unigram distribution. Lin et al. (2006) propose a related criterion for robust summarization evaluation, but to our knowledge this criteria has been unexplored in summarization systems. We address optimizing equation (2) as well as summary sentence ordering in section 4.

KLSUM yields 6.0 R-2 without stop words, beating SUMBASIC but not with statistical significance. It is worth noting however that KLSUM's performance matches SUMFOCUS (Vanderwende et al., 2007), the highest R-2 performing system at DUC 2006.

3.3 TopicSum

As mentioned in section 3.2, the raw unigram distribution $P_{\mathcal{D}}(\cdot)$ may not best reflect the content of $\mathcal{D}$ for the purpose of summary extraction. We propose TOPICSUM, which uses a simple LDA-like topic model (Blei et al., 2003) similar to Daumé III and Marcu (2006) to estimate a content distribution for summary extraction.[11] We extract summary sentences as before using the KLSUM criterion (see equation (2)), plugging in a learned content distribution in place of the raw unigram distribution.

System	ROUGE -stop			ROUGE all		
	R-1	R-2	R-SU4	R-1	R-2	R-SU4
SUMBASIC	29.6	5.3	8.6	36.1	7.1	12.3
KLSUM	30.6	6.0	8.9	38.9	8.3	13.7
TOPICSUM	**31.7**	**6.3**	9.1	39.2	8.4	13.6
HIERSUM	30.5	**6.4**	9.2	40.1	8.6	**14.3**

Table 1: ROUGE results on DUC2006 for models presented in section 3. Results in bold represent results statistically significantly different from SUMBASIC in the appropriate metric.

First, we describe our topic model (see figure 1) which generates a collection of document sets. We assume a fixed vocabulary V:[12]

1. Draw a background vocabulary distribution ϕ_B from DIRICHLET(V,λ_B) shared across document collections[13] representing the background distribution over vocabulary words. This distribution is meant to flexibly model stop words which do not contribute content. We will refer to this topic as BACKGROUND.

2. For each document set $\mathcal{D}$, we draw a content distribution ϕ_C from DIRICHLET(V,λ_C) representing the significant content of $\mathcal{D}$ that we wish to summarize. We will refer to this topic as CONTENT.

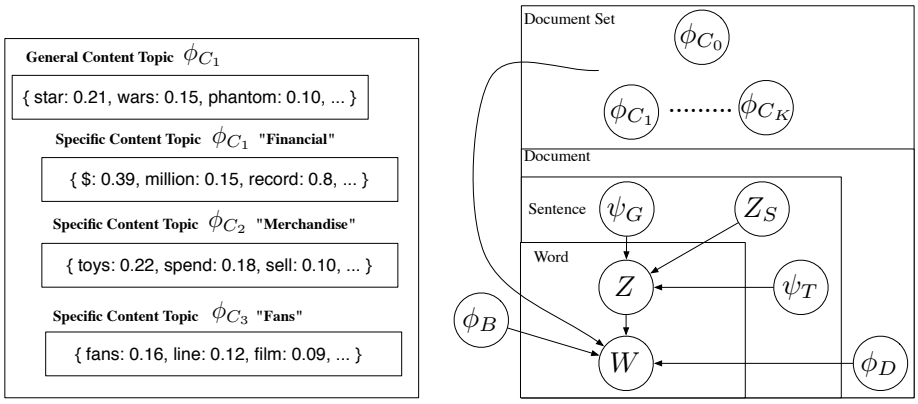
3. For each document D in $\mathcal{D}$, we draw a document-specific vocabulary distribution ϕ_D from DIRICHLET(V,λ_D) representing words which are local to a single document, but do not appear across several documents. We will refer to this topic as DOCSPECIFIC.

[10] In order to ensure finite values of KL-divergence we smoothe $P_{\mathbf{S}}(\cdot)$ so that it has a small amount of mass on all document set words.

[11] A topic model is a probabilistic generative process that generates a collection of documents using a mixture of topic vocabulary distributions (Steyvers and Griffiths, 2007). Note this usage of topic is unrelated to the summary focus given for document collections; this information is ignored by our models.

[12] In contrast to previous models, stop words are *not* removed in pre-processing.

[13] DIRICHLET(V,λ) represents the symmetric Dirichlet prior distribution over V each with a pseudo-count of λ. Concrete pseudo-count values will be given in section 4.



(a) Content Distributions (b) HIERSUM Graphical Model

Figure 2: (a): Examples of general versus specific content distributions utilized by HIERSUM (see section 3.4). The general content distribution ϕ_{C_0} will be used throughout a document collection and represents core concepts in a story. The specific content distributions represent topical 'sub-stories' with vocabulary tightly clustered together but consistently used across documents. Quoted names of specific topics are given manually to facilitate interpretation. (b) Graphical model depiction of the HIERSUM model (see section 3.4). Similar to the TOPICSUM model (see section 3.3) except for adding complexity in the content hierarchy as well as sentence-specific prior distributions between general and specific content topics (early sentences should have more general content words). Several dependencies are missing from this depiction; crucially, each sentence's specific topic Z_S depends on the last sentence's Z_S.

4. For each sentence S of each document D, draw a distribution ψ_T over topics (CONTENT, DOCSPECIFIC, BACKGROUND) from a Dirichlet prior with pseudo-counts $(1.0, 5.0, 10.0)$.[14] For each word position in the sentence, we draw a topic Z from ψ_T, and a word W from the topic distribution Z indicates.

Our intent is that ϕ_C represents the core content of a document set. Intuitively, ϕ_C does not include words which are common amongst several document collections (modeled with the BACKGROUND topic), or words which don't appear across many documents (modeled with the DOCSPECIFIC topic). Also, because topics are tied together at the sentence level, words which frequently occur with other content words are more likely to be considered content words.

We ran our topic model over the DUC 2006 document collections and estimated the distribution $\phi_C(\cdot)$ for each document set.[15] Then we extracted a summary using the KLSUM criterion with our estimated ϕ_C in place of the the raw unigram distribution. Doing so yielded 6.3 R-2 without stop words (see TOPICSUM in table 1); while not a statistically significant improvement over KLSUM, it is our first model which outperforms SUMBASIC with statistical significance.

Daumé III and Marcu (2006) explore a topic model similar to ours for query-focused multi-document summarization.[16] Crucially however, Daumé III and Marcu (2006) selected sentences with the highest expected number of CONTENT words.[17] We found that in our model using this extraction criterion yielded 5.3 R-2 without stop words, significantly underperforming our TOPICSUM model. One reason for this may be that Daumé III and Marcu (2006)'s criterion encourages selecting sentences which have words that are confidently generated by the CONTENT distribution, but not necessarily sentences which contain a plurality of it's mass.

[14] The different pseudo-counts reflect the intuition that most of the words in a document come from the BACKGROUND and DOCSPECIFIC topics.

[15] While possible to obtain the predictive posterior CONTENT distribution by analytically integrating over ϕ_C (Blei et al., 2003), doing so gave no benefit.

[16] Daumé III and Marcu (2006) note their model could be used outside of query-focused summarization.

[17] This is phrased as selecting the sentence which has the highest posterior probability of emitting CONTENT topic words, but this is equivalent.

(a) HIERSUM output

The French government Saturday announced several emergency measures to support the jobless people, including sending an additional 500 million franc (84 million U.S. dollars) unemployment aid package. The unemployment rate in France dropped by 0.3 percent to stand at 12.4 percent in November, said the Ministry of Employment Tuesday.

(b) PYTHY output

Several hundred people took part in the demonstration here today against the policies of the world's most developed nations. The 12.5 percent unemployment rate is haunting the Christmas season in France as militants and unionists staged several protests over the past week against unemployment.

(c) Ref output

High unemployment is France's main economic problem, despite recent improvements. A top worry of French people, it is a factor affecting France's high suicide rate. Long-term unemployment causes social exclusion and threatens France's social cohesion.

(d) Reference Unigram Coverage

word	Ref	PYTHY	HIERSUM
unemployment	8	9	10
france's	6	1	4
francs	4	0	1
high	4	1	2
economic	2	0	1
french	2	1	3
problem	2	0	1
benefits	2	0	0
social	2	0	2
jobless	2	1	2

Table 2: Example summarization output for systems compared in section 5.2. (a), (b), and (c) represent the first two sentences output from PYTHY, HIERSUM, and reference summary respectively. In (d), we present the most frequent non-stop unigrams appearing in the reference summary and their counts in the PYTHY and HIERSUM summaries. Note that many content words in the reference summary absent from PYTHY's proposal are present in HIERSUM's.

3.4 HIERSUM

Previous sections have treated the content of a document set as a single (perhaps learned) unigram distribution. However, as Barzilay and Lee (2004) observe, the content of document collections is highly structured, consisting of several topical themes, each with its own vocabulary and ordering preferences. For concreteness consider the DUC 2006 document collection describing the opening of *Star Wars: Episode 1* (see figure 2(a)).

While there are words which indicate the general content of this document collection (e.g. *star*, *wars*), there are several sub-stories with their own specific vocabulary. For instance, several documents in this collection spend a paragraph or two talking about the financial aspect of the film's opening and use a specific vocabulary there (e.g. *$*, *million*, *record*). A user may be interested in general content of a document collection or, depending on his or her interests, one or more of the sub-stories. We choose to adapt our topic modeling approach to allow modeling this aspect of document set content.

Rather than drawing a single CONTENT distribution ϕ_C for a document collection, we now draw a general content distribution ϕ_{C_0} from DIRICHLET(V,λ_G) as well as specific content distributions ϕ_{C_i} for $i = 1, \ldots, K$ each from DIRICHLET(V,λ_S).[18] Our intent is that ϕ_{C_0} represents the general content of the document collection and each ϕ_{C_i} represents specific sub-stories.

As with TOPICSUM, each sentence has a distribution ψ_T over topics (BACKGROUND, DOCSPECIFIC, CONTENT). When BACKGROUND or DOCSPECIFIC topics are chosen, the model works exactly as in TOPICSUM. However when the CONTENT topic is drawn, we must decide whether to emit a general content word (from ϕ_{C_0}) or from one of the specific content distributions (from one of ϕ_{C_i} for $i = 1, \ldots, K$). The generative story of TOPICSUM is altered as follows in this case:

- **General or Specific?** We must first decide whether to use a general or specific content word. Each sentence draws a binomial distribution ψ_G determining whether a CONTENT word in the sentence will be drawn from the general or a specific topic distribution. Reflecting the intuition that the earlier sentences in a document[19] describe the general content of a story, we bias ψ_G to be drawn from BETA(5,2), preferring general content words, and every later sentence from BETA(1,2).[20]

- **What Specific Topic?** If ψ_G decides we are

[18] We choose K=3 in our experiments, but one could flexibly choose K as Blei et al. (2004) does.

[19] In our experiments, the first 5 sentences.

[20] BETA(a,b) represents the beta prior over binomial random variables with a and b being pseudo-counts for the first and second outcomes respectively.

emitting a topic specific content word, we must decide which of $\phi_{C_1}, \ldots, \phi_{C_K}$ to use. In order to ensure tight lexical cohesion amongst the specific topics, we assume that each sentence draws a single specific topic Z_S used for every specific content word in that sentence. Reflecting intuition that adjacent sentences are likely to share specific content vocabulary, we utilize a 'sticky' HMM as in Barzilay and Lee (2004) over the each sentences' Z_S. Concretely, Z_S for the first sentence in a document is drawn uniformly from $1, \ldots, K$, and each subsequent sentence's Z_S will be identical to the previous sentence with probability σ, and with probability $1 - \sigma$ we select a successor topic from a learned transition distribution amongst $1, \ldots, K$.[21]

Our intent is that the general content distribution ϕ_{C_0} now prefers words which not only appear in many documents, but also words which appear consistently throughout a document rather than being concentrated in a small number of sentences. Each specific content distribution ϕ_{C_i} is meant to model topics which are used in several documents but tend to be used in concentrated locations.

HIERSUM can be used to extract several kinds of summaries. It can extract a general summary by plugging ϕ_{C_0} into the KLSUM criterion. It can also produce topical summaries for the learned specific topics by extracting a summary over each ϕ_{C_i} distribution; this might be appropriate for a user who wants to know more about a particular substory. While we found the general content distribution (from ϕ_{C_0}) to produce the best single summary, we experimented with utilizing topical summaries for other summarization tasks (see section 6.1). The resulting system, HIERSUM yielded 6.4 R-2 without stop words. While not a statistically significant improvement in ROUGE over TOPICSUM, we found the summaries to be noticeably improved.

4 Inference and Model Details

Since globally optimizing the KLSUM criterion in equation (equation (2)) is exponential in the total number of sentences in a document collection, we opted instead for a simple approximation where sentences are greedily added to a summary so long as they decrease KL-divergence. We attempted more complex inference procedures such as McDonald (2007), but these attempts only yielded negligible performance gains. All summary sentence ordering was determined as follows: each sentence in the proposed summary was assigned a number in $[0, 1]$ reflecting its relative sentence position in its source document, and sorted by this quantity.

All topic models utilize Gibbs sampling for inference (Griffiths, 2002; Blei et al., 2004). In general for concentration parameters, the more specific a distribution is meant to be, the smaller its concentration parameter. Accordingly for TOPICSUM, $\lambda_G = \lambda_D = 1$ and $\lambda_C = 0.1$. For HIERSUM we used $\lambda_G = 0.1$ and $\lambda_S = 0.01$. These parameters were minimally tuned (without reference to ROUGE results) in order to ensure that all topic distribution behaved as intended.

5 Formal Experiments

We present formal experiments on the DUC 2007 data main summarization task, proposing a general summary of at most 250 words[22] which will be evaluated automatically and manually in order to simulate as much as possible the DUC evaluation environment.[23] DUC 2007 consists of 45 document sets, each consisting of 25 documents and 4 human reference summaries.

We primarily evaluate the HIERSUM model, extracting a single summary from the general content distribution using the KLSUM criterion (see section 3.2). Although the differences in ROUGE between HIERSUM and TOPICSUM were minimal, we found HIERSUM summary quality to be stronger.

In order to provide a reference for ROUGE and manual evaluation results, we compare against PYTHY, a state-of-the-art supervised sentence extraction summarization system. PYTHY uses human-generated summaries in order to train a sentence ranking system which discriminatively maximizes

[21] We choose $\sigma = 0.75$ in our experiments.

[22] Since the ROUGE evaluation metric is recall-oriented, it is always advantageous - with respect to ROUGE - to use all 250 words.

[23] Although the DUC 2007 main summarization task provides an indication of user intent through topic focus queries, we ignore this aspect of the data.

System	ROUGE w/o stop			ROUGE w/ stop		
	R-1	R-2	R-SU4	R-1	R-2	R-SU4
HIERSUM unigram	34.6	7.3	10.4	43.1	9.7	15.3
HIERSUM bigram	33.8	9.3	11.6	42.4	11.8	16.7
PYTHY w/o simp	34.7	8.7	11.8	42.7	11.4	16.5
PYTHY w/ simp	35.7	8.9	12.1	42.6	11.9	16.8

Table 3: Formal ROUGE experiment results on DUC 2007 document set collection (see section 5.1). While HIERSUM unigram underperforms both PYTHY systems in statistical significance (for R-2 and RU-4 with and without stop words), HIERSUM bigram's performance is comparable and statistically no worse.

ROUGE scores. PYTHY uses several features to rank sentences including several variations of the SUMBASIC score (see section 3.1). At DUC 2007, PYTHY was ranked first overall in automatic ROUGE evaluation and fifth in manual content judgments. As PYTHY utilizes a sentence simplification component, which we do not, we also compare against PYTHY without sentence simplification.

5.1 ROUGE Evaluation

ROUGE results comparing variants of HIERSUM and PYTHY are given in table 3. The HIERSUM system as described in section 3.4 yields 7.3 R-2 without stop words, falling significantly short of the 8.7 that PYTHY without simplification yields. Note that R-2 is a measure of bigram recall and HIERSUM does not represent bigrams whereas PYTHY includes several bigram and higher order n-gram statistics.

In order to put HIERSUM and PYTHY on equal-footing with respect to R-2, we instead ran HIERSUM with each sentence consisting of a bag of bigrams instead of unigrams.[24] All the details of the model remain the same. Once a general content distribution over bigrams has been determined by hierarchical topic modeling, the KLSUM criterion is used as before to extract a summary. This system, labeled HIERSUM bigram in table 3, yields 9.3 R-2 without stop words, significantly outperforming HIERSUM unigram. This model outperforms PYTHY with and without sentence simplification, but not with statistical significance. We conclude that both PYTHY variants and HIERSUM bigram are comparable with respect to ROUGE performance.

[24]Note that by doing topic modeling in this way over bigrams, our model becomes degenerate as it can generate inconsistent bags of bigrams. Future work may look at topic models over n-grams as suggested by Wang et al. (2007).

Question	PYTHY	HIERSUM
Overall	20	49
Non-Redundancy	21	48
Coherence	15	54
Focus	28	41

Table 4: Results of manual user evaluation (see section 5.2). 15 participants expressed 69 pairwise preferences between HIERSUM and PYTHY. For all attributes, HIERSUM outperforms PYTHY; all results are statistically significant as determined by pairwise t-test.

5.2 Manual Evaluation

In order to obtain a more accurate measure of summary quality, we performed a simple user study. For each document set in the DUC 2007 collection, a user was given a reference summary, a PYTHY summary, and a HIERSUM summary;[25] note that the original documents in the set were *not* provided to the user, only a reference summary. For this experiment we use the bigram variant of HIERSUM and compare it to PYTHY without simplification so both systems have the same set of possible output summaries.

The reference summary for each document set was selected according to highest R-2 without stop words against the remaining peer summaries. Users were presented with 4 questions drawn from the DUC manual evaluation guidelines:[26] (1) Overall quality: Which summary was better overall? (2) Non-Redundancy: Which summary was less redundant? (3) Coherence: Which summary was more coherent? (4) Focus: Which summary was more

[25]The system identifier was of course not visible to the user. The order of automatic summaries was determined randomly.

[26]http://www-nlpir.nist.gov/projects/duc/duc2007/quality-questions.txt

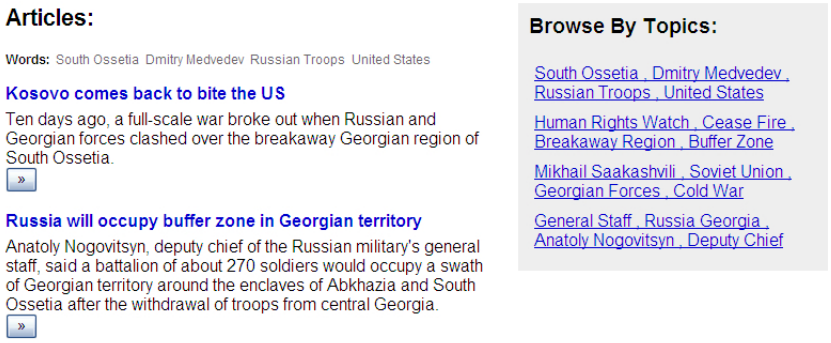


Figure 3: Using HIERSUM to organize content of document set into topics (see section 6.1). The sidebar gives key phrases salient in each of the specific content topics in HIERSUM (see section 3.4). When a topic is clicked in the right sidebar, the main frame displays an extractive 'topical summary' with links into document set articles. Ideally, a user could use this interface to quickly find content in a document collection that matches their interest.

focused in its content, not conveying irrelevant details? The study had 16 users and each was asked to compare five summary pairs, although some did fewer. A total of 69 preferences were solicited. Document collections presented to users were randomly selected from those evaluated fewest.

As seen in table 5.2, HIERSUM outperforms PYTHY under all questions. All results are statistically significant as judged by a simple pairwise t-test with 95% confidence. It is safe to conclude that users in this study strongly preferred the HIERSUM summaries over the PYTHY summaries.

6 Discussion

While it is difficult to qualitatively compare one summarization system over another, we can broadly characterize HIERSUM summaries compared to some of the other systems discussed. For example output from HIERSUM and PYTHY see table 2. On the whole, HIERSUM summaries appear to be significantly less redundant than PYTHY and moderately less redundant than SUMBASIC. The reason for this might be that PYTHY is discriminatively trained to maximize ROUGE which does not directly penalize redundancy. Another tendency is for HIERSUM to select longer sentences typically chosen from an early sentence in a document. As discussed in section 3.4, HIERSUM is biased to consider early sentences in documents have a higher proportion of general content words and so this tendency is to be expected.

6.1 Content Navigation

A common concern in multi-document summarization is that without any indication of user interest or intent providing a single satisfactory summary to a user may not be feasible. While many variants of the general summarization task have been proposed which utilize such information (Vanderwende et al., 2007; Nastase, 2008), this presupposes that a user knows enough of the content of a document collection in order to propose a query.

As Leuski et al. (2003) and Branavan et al. (2007) suggest, a document summarization system should facilitate content discovery and yield summaries relevant to a user's interests. We may use HIERSUM in order to facilitate content discovery via presenting a user with salient words or phrases from the specific content topics parametrized by $\phi_{C_1}, \ldots, \phi_{C_K}$ (for an example see figure 3). While these topics are not adaptive to user interest, they typically reflect lexically coherent vocabularies.

Conclusion

In this paper we have presented an exploration of content models for multi-document summarization and demonstrated that the use of structured topic models can benefit summarization quality as measured by automatic and manual metrics.

Acknowledgements The authors would like to thank Bob Moore, Chris Brockett, Chris Quirk, and Kristina Toutanova for their useful discussions as well as the reviewers for their helpful comments.

References

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *HLT-NAACL*.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *JMLR*.

David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. 2004. Hierarchical topic models and the nested chinese restaurant process. In *NIPS*.

S.R.K. Branavan, Pawan Deshpande, and Regina Barzilay. 2007. Generating a table-of-contents. In *ACL*.

Hal Daumé III and Daniel Marcu. 2006. Bayesian query-focused summarization. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*.

Jacob Eisenstein and Regina Barzilay. 2008. Bayesian unsupervised topic segmentation. In *EMNLP-SIGDAT*.

Thomas Griffiths. 2002. Gibbs sampling in the generative model of latent dirichlet allocation.

Sanda Harabagiu, Andrew Hickl, and Finley Lacatusu. 2007. Satisfying information needs with multi-document summaries. *Inf. Process. Manage.*, 43(6).

Mirella Lapata. 2003. Probabilistic text structuring: Experiments with sentence ordering. In *ACL*.

Jurij Leskovec, Natasa Milic-frayling, and Marko Grobelnik. 2005. Impact of linguistic analysis on the semantic graph coverage and learning of document extracts. In *In AAAI 05*.

Anton Leuski, Chin-Yew Lin, and Eduard Hovy. 2003. ineats: Interactive multi-document summarization. In *ACL*.

Chin-Yew Lin, Guihong Cao, Jianfeng Gao, and Jian-Yun Nie. 2006. An information-theoretic approach to automatic evaluation of summaries. In *HLT-NAACL*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proc. ACL workshop on Text Summarization Branches Out*.

H.P. Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal*.

Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *ECIR*.

Kathleen R. Mckeown, Judith L. Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. 1999. Towards multidocument summarization by reformulation: Progress and prospects. In *In Proceedings of AAAI-99*.

Vivi Nastase. 2008. Topic-driven multi-document summarization with encyclopedic knowledge and spreading activation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*.

A. Nenkova and L. Vanderwende. 2005. The impact of frequency on summarization. Technical report, Microsoft Research.

Dragomir R. Radev. 2004. Lexrank: graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR*.

M. Steyvers and T. Griffiths, 2007. *Probabilistic Topic Models*.

Kristina Toutanova, Chris Brockett, Michael Gamon Jagadeesh Jagarlamudi, Hisami Suzuki, and Lucy Vanderwende. 2007. The pythy summarization system: Microsoft research at duc 2007. In *DUC*.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. volume 43.

Xiaojun Wan and Jianwu Yang. 2006. Improved affinity graph based multi-document summarization. In *HLT-NAACL*.

Xuerui Wang, Andrew McCallum, and Xing Wei. 2007. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *ICDM*.