Content Models for Survey Generation: A Factoid-Based Evaluation

Rahul Jha[⋆], Catherine Finegan-Dollak[⋆], Reed Coke[⋆], Ben King[⋆], Dragomir Radev[⋆†]

[⋆] Department of EECS, University of Michigan, USA
[†] School of Information, University of Michigan, USA
{rahuljha,cfdollak,reedcoke,benking,radev}@umich.edu

Abstract

We present a new factoid-annotated dataset for evaluating content models for scientific survey article generation containing 3,425 sentences from 7 topics in *natural language processing*. We also introduce a novel HITS-based content model for automated survey article generation called HitSum that exploits the lexical network structure between sentences from citing and cited papers. Using the factoid-annotated data, we conduct a pyramid evaluation and compare HitSum with two previous state-of-the-art content models: C-Lexrank, a network based content model, and TopicSum, a Bayesian content model. Our experiments show that our new content model captures useful survey-worthy information and outperforms C-Lexrank by 4% and TopicSum by 7% in pyramid evaluation.

1 Introduction

Survey article generation is the task of automatically building informative surveys for scientific topics. Given the rapid growth of publications in scientific fields, the development of such systems is crucial as human-written surveys exist for a limited number of topics and get outdated quickly. In this paper, we investigate content models for extracting survey-worthy information from scientific papers. Such models are an essential component of any system for automatic survey article generation. Earlier work in the area of survey article generation has investigated content models based on lexical networks (Mohammad et al., 2009; Qazvinian and Radev, 2008). These models take as input citing sentences that describe important papers on the topic and assign them a salience score based on centrality in a lexical network formed by the input citing sentences. In this paper, we propose a new content model based on network structure previously unexplored for this task that exploits the lexical relationship between citing sentences and the sentences from the original papers that they cite. Our new formulation of the lexical network structure fits nicely with the hubs and authorities model for identifying important nodes in a network (Kleinberg, 1999), leading to a new content model called HitSum. In addition to this new content model, we also describe how Bayesian content models previously explored in the news domain can be adapted for the content modeling task for survey generation.

Factoid	Weight
Question Answering	
answer extraction	6
question classification	6
definition of question answering	5
TREC QA track	5
information retrieval	5
Dependency Parsing	
non-projective dependency structures / trees	6
projectivity / projective dependency trees	6
deterministic parsing approaches: Nivre's algorithm	5
terminology: head - dependent	4
grammar driven approaches for dependency parsing	4

Table 1: Sample factoids from the topics of *question answering* and *dependency parsing* along with their factoid weights.

For the task of evaluating various content models discussed in this paper, we have annotated a total of 3,425 sentences across 7 topics in the field of *natural language processing* with factoids from each of the topics. The factoids we use were extracted from existing survey articles and tutorials on each topic (Jha et al., 2013), and thus represent information that must be captured by a survey article on the corresponding topic. Each of the factoids is assigned a weight based on its frequency in the surveys/tutorials, which allows us to do pyra-

Topic	# Sentences
dependency parsing	487
named entity recognition	383
question answering	452
semantic role labeling	466
sentiment analysis	613
summarization	507
word sense disambiguation	425

Table 2: List of seven NLP topics used in our experiments along with input size.

mid evaluation of our content models. Some sample factoids are shown in Table 1. Evaluation using factoids extracted from existing survey articles can help us understand the limits of automated survey article generation and how well these systems can be expected to perform. For example, if certain kinds of factoids are missing consistently from our input sentences, improvements in content models are unlikely to get us closer to the goal of generating survey articles that match those generated by humans, and effort must be directed to extracting text from other sources that will contain the missing information. On the other hand, if most of the factoids exist in the input sentences but important factoids are not found by the content models, we can think of strategies for improving these models by doing error analysis.

The main contributions of this paper are:

- HITSUM, a new HITS-based content model for automatic survey generation for scientific topics.
- A new dataset of 3,425 factoid-annotated sentences for scientific articles in 7 topics.
- Experimental results for pyramid evaluation comparing three existing content models (Lexrank, C-Lexrank, TOPICSUM) with HITSUM.

The rest of this paper is organized as follows. Section 2 describes the dataset used in our experiment and the factoid annotation process. Section 3 describes each of the content models used in our experiments including HITSUM. Section 4 describes our experiments and Section 5 summarizes the results. We summarize the related work in Section 6 and conclude in Section 7.

2 Data

Prior research in automatic survey generation has explored using text from different parts of scientific papers. Some of the recent work has treated survey generation as a direct extension of single paper summarization (Qazvinian and Radev, 2008) and used citing sentences to a set of relevant papers as the input for the summarizer (Mohammad et al., 2009; Qazvinian et al., 2013). However, in our prior work, we have observed that it's difficult to generate coherent and readable summaries using just citing sentences and have proposed the use of sentences from introductory texts of papers that cite a number of important papers on a topic (Jha et al., 2015). The use of full text allows for the use of discourse structure of these documents in framing coherent and readable surveys. Since the content models we explore are meant to be part of a larger system that should be able to generate coherent and readable survey articles, we use the introduction sentences for our experiments as well.

The corpus we used for extracting our experimental data was the ACL Anthology Network, a comprehensive bibliographic dataset that contains full text and citations for papers in most of the important venues in *natural language processing* (Radev et al., 2013). An oracle method is used for selecting the initial set of papers for each topic. For each topic, the bibliographies of at least three human-written surveys were extracted, and any papers that appeared in more than one survey were added to the target document set for the topic.

The text for summarization is extracted from introductory sections of papers that cite papers in the target document set. The intuition behind this is that the introductory sections of papers that cite these target document summarize the research in papers from the target document set as well as the relationships between these papers. Thus, these introductions can be thought of as mini-surveys for specific aspects of the topic; combining text from these introductory sections should allow us to generate good comprehensive survey articles for the topic[1]. For our experiments, we sort the citing papers based on the number of papers they cite

[1] Other sections of papers might have such information, e.g. related work. Initial data analysis showed, however, that not all papers in our corpus had related work sections. Thus for consistency, we decided to use introduction sections. The perfect system for this task would be able to extract ”related work style” text segments from an entire paper.

Input sentence	**Factoids**
According to [1] , the corpus based supervised machine learning methods are the most successful approaches to WSD where contextual features have been used mainly to distinguish ambiguous words in these methods.	supervised wsd, corpus based wsd
Compared with supervised methods, unsupervised methods do not require tagged corpus, but the precision is usually lower than that of the supervised methods.	supervised wsd, unsupervised wsd
Word sense disambiguation (WSD) has been a hot topic in natural language processing, which is to determine the sense of an ambiguous word in a specific context.	definition of word sense disambiguation
Improvement in the accuracy of identifying the correct word sense will result in better machine translation systems, information retrieval systems, etc.	wsd for machine translation, wsd for information retrieval
The SENSEVAL evaluation framework (Kilgarriff 1998) was a DARPA-style competition designed to bring some conformity to the field of WSD, although it has yet to achieve that aim completely.	senseval

Table 3: Sample input sentences from the topic of *word sense disambiguation* annotated with factoids.

in the target document set, pick the top 20 papers, and extract sentences from their introductions to form the input text for the summarizer. The seven topics used in our experiments and input size for each topic are shown in Table 2.

Once the input text for each topic has been extracted, we annotate the sentences in the input text with factoids for that topic. Some annotated sentences in the topic of *word sense disambiguation* are shown in Table 3. Given this new annotated data, we can compare how the factoids are distributed across different citing sentences (as annotated by Jha et al. (2013)) and introduction sentences that we have annotated. For this, we divide the factoids into five categories: definitions, venue, resources, methodology, and applications. The fractional distribution of factoids in these categories is shown in Table 4. We can see that the distribution of factoids relating to venues, methodology and applications is similar for the two datasets. However, factoids related to definitional sentences are almost completely missing in the citing sentences data. This lack of background information in citing sentences is one of the motivations for using introduction sentences for survey article generation as opposed to previous work.

The complete set of factoids as well as annotated sentences for all the topics is available for download at `http://clair.si.umich.edu/corpora/Surveyor_CM_Data.tar.gz`.

3 Content Models

We now describe each of the content models used in our experiments.

Factoid category	**% Citing**	**% Intro**
definitions	0	4
venue	6	6
resources	18	2
methodology	70	83
applications	6	5

Table 4: Fractional distribution of factoids across various categories in citing sentences vs introduction sentences.

3.1 Lexrank

Lexrank is a network-based content selection algorithm that serves as a baseline for our experiments. Given an input set of sentences, it first creates a network using these sentences where each node represents a sentence and each edge represents the tf-idf cosine similarity between the sentences. Two methods for creating the network are possible. First, we can remove all edges that are lower than a certain threshold of similarity (generally set to 0.1). The Lexrank value for a node $p(u)$ in this case is calculated as:

$$\frac{1-d}{N} + d \sum_{v \in adj[u]} \frac{p(v)}{deg(v)}$$

Where N is the total number of sentences, d is the damping factor that controls the probability of a random jump (usually set to 0.85), $deg(v)$ is the degree of the node v, and $adj[u]$ is the set of nodes connected to the node u. A different way of creating the network is to treat the sentence similarities as edge weights and use the adjacency matrix as a transition matrix after normalizing the rows; the formula then becomes:

A dictionary such as the LDOCE has broad coverage of word senses, useful for WSD .

This paper describes a program that disambiguates English word senses in unrestricted text using statistical models of the major Roget's Thesaurus categories.

Our technique offers benefits both for online semantic processing and for the challenging task of mapping word senses across multiple MRDs in creating a merged lexical database.

The words in the sentences may be any of the 28,000 headwords in Longman's Dictionary of Contemporary English (LDOCE) and are disambiguated relative to the senses given in LDOCE.

This paper describes a heuristic approach to automatically identifying which senses of a machine-readable dictionary (MRD) headword are semantically related versus those which correspond to fundamentally different senses of the word.

Figure 1: A sentence from P_{citing} with a high hub score (bolded) and some of sentences from P_{cited} that it links to (italicised). The sentence from P_{citing} obtain a high hub score by being connected to the sentences with high authority scores.

$$\frac{1-d}{N} + d \sum_{v \in adj[u]} \frac{cos(u,v)}{TotalCos_v} p(v)$$

Where $cos(u,v)$ gives the tf-idf cosine similarity between sentence u and v and $TotalCos_v = \sum_{z \in adj[v]} cos(z,v)$. In our experiments, we employ this second formulation. The above equation can be solved efficiently using the power method (Newman, 2010) to obtain $p(u)$ for each node, which is then used as the score for ordering the sentences. The final Lexrank values $p(u)$ for a node represent the stationary distribution of the Markov chain represented by the transition matrix. Lexrank has been shown to perform well in summarization experiments (Erkan and Radev, 2004).

3.2 C-Lexrank

C-Lexrank is a clustering-based summarization system that was proposed by Qazvinian and Radev (2008) to summarize different perspectives in citing sentences that reference a paper or a topic. To create summaries, C-LexRank constructs a fully connected network in which vertices are sentences, and edges are cosine similarities calculated using the tf-idf vectors of citation sentences. It then employs a hierarchical agglomeration clustering algorithm proposed by Clauset et al. (2004) to find communities of sentences that discuss the same scientific contributions. Once the graph is clustered and communities are formed, the method extracts sentences from different clusters to build a summary. It iterates through the clusters from largest to smallest, choosing the most salient sentence of each cluster, until the summary length limit is reached. The salience of a sentence in its cluster is defined as its Lexrank value in the lexical network formed by sentences in the cluster.

3.3 HitSum

The input set of sentences in our data come from introductory sections of papers that cite important papers on a topic. We'll refer to the set of citing papers that provide the input text for the summarizer as P_{citing} and the set of important papers that represent the research we are trying to summarize as P_{cited}. Both Lexrank and C-Lexrank work by finding central sentences in a network formed by the input sentences and thus, only use the lexical information present in P_{citing}, while ignoring additional lexical information from the papers in P_{cited}. We now present a formulation that uses the network structure that exists between the sentences in the two sets of papers to incorporate additional lexical information into the summarization system. This system is based on the hubs and authorities or the HITS model (Kleinberg, 1999) and hence is called HitSum.

HitSum, in addition to the sentences from the introductory sections of papers in P_{citing}, also uses sentences from the abstracts of P_{cited}. It starts by computing the tf-idf cosine similarity between the sentences of each paper $p_i \in P_{citing}$ with the sentences in the abstracts of each paper $p_j \in P_{cited}$ that is directly cited by p_i. A directed edge is created between every sentence s_i in p_i and s_j in p_j if $sim(s_i, s_j) > s_{min}$, where s_{min} is a similarity threshold (set to 0.1 for our experiments). Once this process has been completed for all papers in P_{citing}, we end up with a bipartite graph between sentences from P_{citing} and P_{cited}.

In this bipartite graph, sentences in P_{cited} that

ϕ_B		$\phi_{C/QA}$		$\phi_{D/J07-1005}$		$\phi_{C/NER}$		$\phi_{D/I08-1071}$	
the	0.066	question	0.044	metathesaurus	0.00032	ne	0.028	wikipedia	0.0087
of	0.040	questions	0.038	umls	0.00032	entity	0.022	pages	0.0053
and	0.034	answer	0.028	biomedical	0.00024	named	0.022	million	0.0018
a	0.029	answering	0.022	relevance	0.00024	entities	0.017	extracting	0.0018
in	0.027	qa	0.021	citation	0.00024	ner	0.014	articles	0.0018
to	0.027	answers	0.017	wykoff	0.00024	names	0.009	contributors	0.0018
is	0.017	2001	0.016	bringing	0.00016	location	0.008	version	0.0009
for	0.014	system	0.011	appropriately	0.00016	tagging	0.007	dakka	0.0009
that	0.012	trec	0.008	organized	0.00016	recognition	0.007	service	0.0009
we	0.011	factoid	0.008	foundation	0.00016	classes	0.007	academic	0.0009

Figure 2: Top words from different word distributions learned by TopicSum on our input document set of 15 topics. ϕ_B is the background word distribution that captures stop words. $\phi_{C/QA}$ and $\phi_{C/NER}$ are the word distributions for the topics of *question answering* and *named entity recognition* respectively. $\phi_{D/J07-1005}$ is the document-specific word distribution for a single paper in *question answering* that focuses on clinical question answering. $\phi_{D/I08-1071}$ is the document-specific word distribution for a single paper in *named entity recognition* that focuses on named entity recognition in Wikipedia articles.

have a lot of incoming edges represent sentences that presented important contributions in the field. Similarly, sentences in P_{citing} that have a lot of outgoing edges represent sentences that summarize a number of important contributions in the field. This suggests using the HITS algorithm, which, given a network, assigns hubs and authorities scores to each node in the network in a mutually reinforcing way. Thus, nodes with high authority scores are those that are pointed to by a number of good hubs, and nodes with high hub scores are those that point to a number of good authorities. This can be formalized with the following equation for the hub score of a node:

$$h(v) \quad = \sum_{u \in successors(v)} a(u)$$

Where $h(v)$ is the hub score for node v, $successors(v)$ is the set of all nodes that v has an edge to, and $a(u)$ is the authority score for node u. Similarly, the authority score for each node is computed as:

$$a(v) \quad = \sum_{u \in predecessors(v)} h(u)$$

Where $predecessors(v)$ is the set of all nodes that have an edge to v. The hub and authority score for each node can be computed using the power method that starts with an initial value and iteratively updates the scores for each node based on the above equations until the hub and authority scores for each node converge to within a tolerance value (set to 1E-08 for our experiments).

In our bipartite lexical network, we expect sentences in P_{cited} receiving high authority scores to be the ones reporting important contributions and sentences in P_{citing} that receive high hub scores to be sentences summarizing important contributions. Figure 1 shows an example of a sentence with a high hub score from the topic of *word sense disambiguation*, along with some of the sentences that it points to. HitSum computes the hub and authority score for each sentence in the lexical network and then uses the hub scores for sentences in P_{citing} as their relevance score. Sentences from P_{cited} are part of the lexical network, but are not used in the output summary.

3.4 TopicSum

TopicSum is a probabilistic content model presented in Haghighi and Vanderwende (2009) and is very similar to an earlier model called BayesSum proposed by Daumé and Marcu (2006). It is a hierarchical, LDA (Latent Dirichlet Allocation) style model that is based on the following generative story:[2] words in any sentence in the corpus can come from one of three word distributions: a background word distribution ϕ_B that flexibly models stop words, a content word distribution ϕ_C for each document set that models content relevant to the entire document set, and a document-specific word distribution ϕ_D. The word distributions are learned using Gibbs sampling. Given n document sets each with k doc-

[2] To avoid confusion in use of the term "topic," in this paper we refer to topics in the LDA sense as "word distributions." "Topics" in this paper refer to the natural language processing topics such as *question answering*, *word sense disambiguation*, etc.

Topic	Lexrank	C-Lexrank	TOPICSUM	HITSUM
dependency parsing	0.47	0.76	0.62	1.00*
named entity recognition	0.80	0.89	0.90*	0.80
question answering	0.65	0.67	0.65	0.76*
sentiment analysis	0.64	0.62	0.75*	0.63
semantic role labeling	0.75*	0.67	0.65	0.69
summarization	0.52	0.75*	0.57	0.68
word sense disambiguation	0.78	0.66	0.67	0.79*
Average	**0.66**	**0.72**	**0.69**	**0.76***

Table 5: Pyramid scores obtained by different content models for each topic along with average scores for each model across all topics. For each topic as well as the average, the best performing method has been highlighted with a *.

uments, we get n content word distributions and $n * k$ document-specific distributions leading to a total of $1 + n + n * k$ word distributions.

To illustrate the kind of distributions TOPICSUM learns in our dataset, Figure 2 shows the top words along with their probabilities from the background word distribution, two content distributions and two document-specific word distributions. We see that the model effectively captures general content words for each topic. $\phi_{C/QA}$ is the word distribution for the topic of *question answering*, while $\phi_{D/J07-1005}$ is the document-specific word distribution for a specific paper in the document set for *question answering*[3] that focuses on clinical question answering. The word distribution $\phi_{D/J07-1005}$ contains words that are relevant to the specific subtopic in the paper, while $\phi_{C/QA}$ contains content words relevant to the general topic of *question answering*. Similar results can be seen in the word distributions for *named entity recognition* $\phi_{C/NER}$ and the document-specific word distribution for a specific paper in the topic $\phi_{D/I08-1071}$[4] that focuses on comparable entity mining.

These topics, learned using Gibbs sampling, can be used to select sentences for a summary in the following way. To summarize a document set, we greedily select sentences that minimize the KL-divergence of our summary to the document-set-specific topic. Thus, the score for each sentence s is $KL(\phi_C || P_s)$ where P_s is the sentence word distribution with add-one smoothing applied to both distributions. Using this objective, sentences that contain words from the content word distribution with high probability are more likely to be selected in the generated summary.

We implemented TOPICSUM in Python using Numpy and then optimized it using Scipy Weave. This code is available for use at `https://github.com/rahuljha/content-models`. The repository also contains Python code for HITSUM.

4 Experiments

For evaluating our content models, we generated 2,000-character-long summaries using each of the systems (Lexrank, C-Lexrank, HITSUM, and TOPICSUM) for each of the topics. The summaries are generated by ranking the input sentences using each content model and picking the top sentences till the budget of 2,000 characters is reached. Each of these summaries is then given a pyramid score (Nenkova and Passonneau, 2004) computed using the factoids assigned to each sentence.

For the pyramid evaluation, the factoids are organized in a pyramid of order n. The top tier in this pyramid contains the highest weighted factoids, the next tier contains the second highest weighted factoids, and so on. The score assigned to a summary is the ratio of the sum of the weights of the factoids it contains to the sum of weights of an optimal summary with the same number of factoids. Pyramid evaluation allows us to capture how each content model performs in terms of selecting sentences with the most highly weighted factoids. Since the factoids have been extracted from human-written surveys and tutorials on each of the topics, the pyramid score gives us an idea of the survey-worthiness of the sentences selected by

[3] Dina Demner-Fushman and Jimmy Lin. 2007. *Answering Clinical Questions with Knowledge-Based and Statistical Techniques.* Computational Linguistics.

[4] Wisam Dakka and Silviu Cucerzan. 2008. *Augmenting wikipedia with named entity tags.* In Proceedings of IJCNLP.

Question classification is a crucial component of modern question answering system.

A what-type question is defined as the one whose question word is 'what', 'which', 'name' or 'list'.

This metaclassifier beats all published numbers on standard question classification benchmarks [4.4].

Due to its challenge, this paper focuses on what-type question classification.

In this paper, we focus on fine-category classification.

The promise of a machine learning approach is that the QA system builder can now focus on designing features and providing labeled data, rather than coding and maintaining complex heuristic rule bases.

Figure 3: Part of the summary generated by HitSum for the topic of *question answering*.

each content model.

5 Results and Discussion

The results of pyramid evaluation are summarized in Table 5. It shows the pyramid score obtained by each system on each of the topics as well as the average score. The highest performing system on average is HitSum with an average performance of 76%. HitSum does especially well for the topics of *dependency parsing*, *question answering*, and *word sense disambiguation*. The second best performing system is C-Lexrank, which is not surprising because it was developed specifically for the task of scientific paper summarization. However, HitSum outperforms C-Lexrank on several topics and by 4% on average.

Figure 3 shows part of the summary generated by HitSum for the topic of question answering. The summary contains mostly informative sentences about different aspects of question answering. One obvious drawback of this summary is that it's not very coherent and readable. However, previous work has shown how network based content models can be combined with discourse models to generate informative yet readable summaries (Jha et al., 2015). We looked at some of the network statistics of the lexical networks used by HitSum. One of the things we noticed is that the lexical networks for topics where HitSum performs well seem to have higher degree assortativity compared to the topics for which it doesn't perform well. High degree assortativity in lexical networks means sentences with high degree tend to be linked to other sentences with high degree. This suggests that HITS performs well for topics where a set of important factoids are mentioned in many citing and source sentences. A larger evaluation dataset is needed for a more thorough analysis of how the network properties of these lexical networks correlate with the performance of various content models.

TopicSum does well on the topics of *named entity recognition* and *sentiment analysis*, but does not do well on average. This can be attributed to the fact that it was developed as a content model for the domain of news summarization and does not translate well to our domain. All systems outperform Lexrank, which achieves the lowest average score. This result is also intuitive, because every other system in our evaluation uses additional information not used by Lexrank: C-Lexrank exploits the community structure in the input set of sentences, HitSum exploits the lexical information from cited sentences, and TopicSum exploits information about global word distribution across all topics.

The different systems we tried in our evaluation depend on using different lexical information and seem to perform well for different topics. This suggests that further gains can be made by combining these systems. For example, C-Lexrank and HitSum can be combined by utilizing both the network formed by citing sentences and the network between the citing sentences and the cited sentences into a larger lexical network. TopicSum scores can be combined with these network-based system by using the TopicSum scores as a prior for each node, and then running either Pagerank or HITS on top of it. We leave exploration of such hybrid systems to future work.

6 Related Work

The goal of content models in the context of summarization is to extract a representation from input text that can help in identifying important sentences that should be in the output summary. Our work is related to two main classes of content models: network-based methods and probabilistic methods. We summarize related work for each of these classes of content models, followed by a short summary of the related work in the domain of scientific summarization.

Network-based content models: Network-based content models (Erkan and Radev, 2004; Mihalcea and Tarau, 2004) work by converting the input sentences into a network. Each sentence is represented by a node in the network, and the edges between sentences are given weight based on the similarities of sentences. They then run Pagerank on this network, and sentences are selected based on their Pagerank score in the network. For computing Pagerank, the network can either be pruned by removing edges that have weights less than a certain threshold, or a weighted version of Pagerank can be run on the network. The method can also be modified for query-focused summarization (Otterbacher et al., 2009). C-Lexrank (Qazvinian and Radev, 2008) modifies Lexrank by first running a clustering algorithm on the network to partition the network into different communities and then selecting sentences from each community by running Lexrank on the sub-network within each community. C-Lexrank was also used in the task of automated survey generation with encouraging results (Mohammad et al., 2009).

Probabilistic content models: One of the first probabilistic content models seems to be BAYESSUM (Daumé and Marcu, 2006), designed for query-focused summarization. BAYESSUM models a set of document collections using a hierarchical LDA style model. Each word in a sentence can be generated using one of three language models: 1) a general English language model that captures English filler or background knowledge, 2) a document-specific language model, and 3) a query language model. These language models are inferred using expectation propagation, and then sentences are ranked based on their likelihood of being generated from the query language model. A similar model for general multidocument summarization called TOPICSUM was proposed by Haghighi and Vanderwende (2009), where the query language model is replaced by a document-collection-specific language model; thus sentences are selected based on how likely they are to contain information that summarizes the entire document collection instead of information pertaining to individual documents or background knowledge.

Barzilay and Lee (2004) present a Hidden Markov Model (HMM) based content model where the hidden states of the HMM represent the topics in the text. The transition probabilities are learned through Viterbi decoding. They show that the HMM model can be used for both reordering of sentences for coherence and discriminative scoring of sentences for extractive summarization. Fung and Ngai (2006) present a similar HMM-based model for multi-document summarization. Jiang and Zhai (2005) proposed an HMM-based model for the problem of extracting coherent passages relevant to a query from a relevant document. They learn an HMM with two background states (B_1 and B_2) and a query-relevant state (R), each associated with a language model. The HMM starts in background state B_1, switches to relevant state R and then switches to the next background state B_2. The sentences that the HMM emits while in R constitute the query-relevant passage from the document.

Scientific summarization: Early work in scientific summarization used abstracts of scientific articles to produce summaries of specific scientific papers (Kupiec et al., 1995). However, later work (Elkiss et al., 2008) showed that citation sentences are as important in understanding the main contributions of a paper.

Nanba and Okumura (1999) explored using reference information to build a system for supporting writing survey articles. Their system extracts citing sentences that describe a referred paper and identify the type of reference relationships. The type of references can be one of the three: 1) type B that base on other researcher's theory, 2) type C that compare with related works, or 3) type O representing relationships other than B or C. They posit that type C sentences are the most important for survey generation and can help show the similarities and differences among cited papers.

Teufel and Moens (2002) propose a method for summarizing scientific articles based on rhetorical status of sentences in scientific articles. They annotate sentences in a corpus of 80 scientific articles with rhetorical status, where the rhetorical status can be one of aim (specific research goal), textual (section structure), own (neutral description of own work), background (generally accepted background), contrast (comparison with other work), basis (agreement with or continuation of other work), and other (neutral description of other's work). They describe classifiers for tagging the rhetorical status of sentences automatically and present a method for using this to assign relevance score to sentences.

In other work, Kan et al. (2002) use a corpus of 2000 annotated bibliographies for scientific papers as a first step towards a supervised summarization system. They found that summaries in their corpus were mostly single-document abstractive summaries that were both indicative and informative and were organized around a "theme," making them ideal for query-based summarization. Mei and Zhai (2008) presented an impact-based summarization method for single-paper summarization that assigns relevance scores to sentences in a paper based on their similarity to the set of citing sentences that reference the paper.

More recently, Hoang and Kan (2010) present a method for automated related work generation. Their system takes as input a set of keywords arranged in a hierarchical fashion that describes a target paper's topic. They hypothesize that sentences in a related work provide either background information or specific contributions. They use two different models to extract these two kinds of sentences using the input tree and combines them to create the final output summary. Zhang et al. (2013) explore methods for biomedical summarization by identifying cliques in a network of semantic predications extracted from citations. These cliques are then clustered and labeled to identify different points of view represented in the summary.

7 Conclusion and Future Work

We have presented a new factoid-annotated dataset for evaluating content models for scientific survey article generation by annotating sentences from seven topics in *natural language processing*. We also introduce a new HITS-based content model called HITSUM for survey article generation that exploits the lexical information from cited papers along with citing papers to rank input sentences for survey-worthiness. We conduct pyramid evaluation using our factoid dataset to compare HITSUM with existing network-based methods (Lexrank, C-Lexrank) as well as methods based on Bayesian content modeling (TOPICSUM). On average, HITSUM outperforms C-Lexrank by 4% and TOPICSUM by 7%. Since the different content models use different kinds of lexical information, further gains might be obtained by combining some of these models into a joint model. We plan to explore this in future work.

References

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 113–120, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Aaron Clauset, Mark E. J. Newman, and Cristopher Moore. 2004. Finding community structure in very large networks. *Phys. Rev. E*, 70(6):066111, Dec.

Hal Daumé, III and Daniel Marcu. 2006. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 305–312, Stroudsburg, PA, USA. Association for Computational Linguistics.

Aaron Elkiss, Siwei Shen, Anthony Fader, Güneş Erkan, David States, and Dragomir R. Radev. 2008. Blind men and elephants: What do citation summaries tell us about a research article? *Journal of the American Society for Information Science and Technology*, 59(1):51–62.

Güneş Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR)*.

Pascale Fung and Grace Ngai. 2006. One story, one flow: Hidden markov story models for multilingual multidocument summarization. *ACM Trans. Speech Lang. Process.*, 3(2):1–16, July.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 362–370, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cong Duy Vu Hoang and Min-Yen Kan. 2010. Towards automated related work summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 427–435, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rahul Jha, Amjad Abu-Jbara, and Dragomir R. Radev. 2013. A system for summarizing scientific topics starting from keywords. In *Proceedings of The Association for Computational Linguistics (short paper)*.

Rahul Jha, Reed Coke, and Dragomir R. Radev. 2015. Surveyor: A system for generating coherent survey articles for scientific topics. In *Proceedings of the Twenty-Ninth AAAI Conference*.

Jing Jiang and ChengXiang Zhai. 2005. Accurately extracting coherent relevant passages using hidden Markov models. pages 289–290.

Min-Yen Kan, Judith L. Klavans, and Kathleen R. McKeown. 2002. Using the Annotated Bibliography as a Resource for Indicative Summarization. In *The International Conference on Language Resources and Evaluation (LREC)*, Las Palmas, Spain.

Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *J. ACM*, 46:604–632, September.

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-95)*, pages 68–73.

Qiaozhu Mei and ChengXiang Zhai. 2008. Generating impact-based summaries for scientific literature. In *Proceedings of the 46th Annual Conference of the Association for Computational Linguistics (ACL-08)*, pages 816–824.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, July.

Saif Mohammad, Bonnie Dorr, Melissa Egan, Ahmed Hassan, Pradeep Muthukrishan, Vahed Qazvinian, Dragomir Radev, and David Zajic. 2009. Using citations to generate surveys of scientific paradigms. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 584–592, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hidetsugu Nanba and Manabu Okumura. 1999. Towards multi-paper summarization using reference information. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 926–931.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (HLT-NAACL '04)*.

Mark E. J. Newman. 2010. *Networks: An Introduction*. Oxford University Press.

Jahna Otterbacher, Gunes Erkan, and Dragomir R. Radev. 2009. Biased lexrank: Passage retrieval using random walks with question-based priors. *Inf. Process. Manage.*, 45(1):42–54, January.

Vahed Qazvinian and Dragomir R. Radev. 2008. Scientific paper summarization using citation summary networks. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING-08)*, Manchester, UK.

Vahed Qazvinian, Dragomir R. Radev, Saif M. Mohammad, Bonnie Dorr, David Zajic, Michael Whidby, and Taesun Moon. 2013. Generating extractive summaries of scientific paradigms. *J. Artif. Int. Res.*, 46(1):165–201, January.

Dragomir R. Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. 2013. The acl anthology network corpus. *Language Resources and Evaluation*, pages 1–26.

Simone Teufel and Marc Moens. 2002. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445.

Han Zhang, Marcelo Fiszman, Dongwook Shin, Bartlomiej Wilkowski, and Thomas C. Rindflesch. 2013. Clustering cliques for graph-based summarization of the biomedical research literature. *BMC Bioinformatics*, 14:182.