Automatically Generating Wikipedia Articles: A Structure-Aware Approach

Christina Sauper and Regina Barzilay
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
{csauper,regina}@csail.mit.edu

Abstract

In this paper, we investigate an approach for creating a comprehensive textual overview of a subject composed of information drawn from the Internet. We use the high-level structure of human-authored texts to automatically induce a domain-specific template for the topic structure of a new overview. The algorithmic innovation of our work is a method to learn topic-specific extractors for content selection jointly for the entire template. We augment the standard perceptron algorithm with a global integer linear programming formulation to optimize both local fit of information into each topic and global coherence across the entire overview. The results of our evaluation confirm the benefits of incorporating structural information into the content selection process.

1 Introduction

In this paper, we consider the task of automatically creating a multi-paragraph overview article that provides a comprehensive summary of a subject of interest. Examples of such overviews include actor biographies from IMDB and disease synopses from Wikipedia. Producing these texts by hand is a labor-intensive task, especially when relevant information is scattered throughout a wide range of Internet sources. Our goal is to automate this process. We aim to create an overview of a subject – e.g., *3-M Syndrome* – by intelligently combining relevant excerpts from across the Internet.

As a starting point, we can employ methods developed for multi-document summarization. However, our task poses additional technical challenges with respect to content planning. Generating a well-rounded overview article requires proactive strategies to gather relevant material, such as searching the Internet. Moreover, the challenge of maintaining output readability is magnified when creating a longer document that discusses multiple topics.

In our approach, we explore how the high-level structure of human-authored documents can be used to produce well-formed comprehensive overview articles. We select relevant material for an article using a domain-specific automatically generated content template. For example, a template for articles about diseases might contain *diagnosis*, *causes*, *symptoms*, and *treatment*. Our system induces these templates by analyzing patterns in the structure of human-authored documents in the domain of interest. Then, it produces a new article by selecting content from the Internet for each part of this template. An example of our system's output[1] is shown in Figure 1.

The algorithmic innovation of our work is a method for learning topic-specific extractors for content selection jointly across the entire template. Learning a single topic-specific extractor can be easily achieved in a standard classification framework. However, the choices for different topics in a template are mutually dependent; for example, in a multi-topic article, there is potential for redundancy across topics. Simultaneously learning content selection for all topics enables us to explicitly model these inter-topic connections.

We formulate this task as a *structured classification* problem. We estimate the parameters of our model using the perceptron algorithm augmented with an integer linear programming (ILP) formulation, run over a training set of example articles in the given domain.

The key features of this structure-aware approach are twofold:

[1] This system output was added to Wikipedia at http://en.wikipedia.org/wiki/3-M_syndrome on June 26, 2008. The page's history provides examples of changes performed by human editors to articles created by our system.

Diagnosis ...No laboratories offering molecular genetic testing for prenatal diagnosis of 3-M syndrome are listed in the GeneTests Laboratory Directory. However, prenatal testing may be available for families in which the disease-causing mutations have been identified in an affected family member in a research or clinical laboratory.

Causes Three M syndrome is thought to be inherited as an autosomal recessive genetic trait. Human traits, including the classic genetic diseases, are the product of the interaction of two genes, one received from the father and one from the mother. In recessive disorders, the condition does not occur unless an individual inherits the same defective gene for the same trait from each parent. ...

Symptoms ...Many of the symptoms and physical features associated with the disorder are apparent at birth (congenital). In some cases, individuals who carry a single copy of the disease gene (heterozygotes) may exhibit mild symptoms associated with Three M syndrome.

Treatment ...Genetic counseling will be of benefit for affected individuals and their families. Family members of affected individuals should also receive regular clinical evaluations to detect any symptoms and physical characteristics that may be potentially associated with Three M syndrome or heterozygosity for the disorder. Other treatment for Three M syndrome is symptomatic and supportive.

Figure 1: A fragment from the automatically created article for 3-M Syndrome.

- **Automatic template creation:** Templates are automatically induced from human-authored documents. This ensures that the overview article will have the breadth expected in a comprehensive summary, with content drawn from a wide variety of Internet sources.
- **Joint parameter estimation for content selection:** Parameters are learned jointly for all topics in the template. This procedure optimizes both local relevance of information for each topic and global coherence across the entire article.

We evaluate our approach by creating articles in two domains: Actors and Diseases. For a data set, we use Wikipedia, which contains articles similar to those we wish to produce in terms of length and breadth. An advantage of this data set is that Wikipedia articles explicitly delineate topical sections, facilitating structural analysis. The results of our evaluation confirm the benefits of structure-aware content selection over approaches that do not explicitly model topical structure.

2 Related Work

Concept-to-text generation and text-to-text generation take very different approaches to content selection. In traditional concept-to-text generation, a content planner provides a detailed template for what information should be included in the output and how this information should be organized (Reiter and Dale, 2000). In text-to-text generation, such templates for information organization are not available; sentences are selected based on their salience properties (Mani and Maybury, 1999). While this strategy is robust and portable across domains, output summaries often suffer from coherence and coverage problems.

In between these two approaches is work on domain-specific text-to-text generation. Instances of these tasks are biography generation in summarization and answering definition requests in question-answering. In contrast to a generic summarizer, these applications aim to characterize the types of information that are essential in a given domain. This characterization varies greatly in granularity. For instance, some approaches coarsely discriminate between biographical and non-biographical information (Zhou et al., 2004; Biadsy et al., 2008), while others go beyond binary distinction by identifying atomic events – e.g., occupation and marital status – that are typically included in a biography (Weischedel et al., 2004; Filatova and Prager, 2005; Filatova et al., 2006). Commonly, such templates are specified manually and are hard-coded for a particular domain (Fujii and Ishikawa, 2004; Weischedel et al., 2004).

Our work is related to these approaches; however, content selection in our work is driven by domain-specific automatically induced templates. As our experiments demonstrate, patterns observed in domain-specific training data provide sufficient constraints for topic organization, which is crucial for a comprehensive text.

Our work also relates to a large body of recent work that uses Wikipedia material. Instances of this work include information extraction, ontology induction and resource acquisition (Wu and Weld, 2007; Biadsy et al., 2008; Nastase, 2008; Nastase and Strube, 2008). Our focus is on a different task — generation of new overview articles that follow the structure of Wikipedia articles.

3 Method

The goal of our system is to produce a comprehensive overview article given a title – e.g., *Cancer*. We assume that relevant information on the subject is available on the Internet but scattered among several pages interspersed with noise.

We are provided with a training corpus consisting of n documents $d_1 \ldots d_n$ in the same domain – e.g., *Diseases*. Each document d_i has a title and a set of delineated sections[2] $s_{i1} \ldots s_{im}$. The number of sections m varies between documents. Each section s_{ij} also has a corresponding heading h_{ij} – e.g., *Treatment*.

Our overview article creation process consists of three parts. First, a preprocessing step creates a template and searches for a number of candidate excerpts from the Internet. Next, parameters must be trained for the content selection algorithm using our training data set. Finally, a complete article may be created by combining a selection of candidate excerpts.

1. **Preprocessing** (Section 3.1) Our preprocessing step leverages previous work in topic segmentation and query reformulation to prepare a template and a set of candidate excerpts for content selection. Template generation must occur once per domain, whereas search occurs every time an article is generated in both learning and application.
   (a) **Template Induction** To create a content template, we cluster all section headings $h_{i1} \ldots h_{im}$ for all documents d_i. Each cluster is labeled with the most common heading h_{ij} within the cluster. The largest k clusters are selected to become topics $t_1 \ldots t_k$, which form the domain-specific content template.
   (b) **Search** For each document that we wish to create, we retrieve from the Internet a set of r excerpts $e_{j1} \ldots e_{jr}$ for each topic t_j from the template. We define appropriate search queries using the requested document title and topics t_j.
2. **Learning Content Selection** (Section 3.2) For each topic t_j, we learn the corresponding topic-specific parameters $\mathbf{w}_j$ to determine the quality of a given excerpt. Using the perceptron framework augmented with an ILP formulation for global optimization, the system is trained to select the best excerpt for each document d_i and each topic t_j. For training, we assume the best excerpt is the original human-authored text s_{ij}.
3. **Application** (Section 3.2) Given the title of a requested document, we select several excerpts from the candidate vectors returned by the search procedure (1b) to create a comprehensive overview article. We perform the decoding procedure jointly using learned parameters $\mathbf{w}_1 \ldots \mathbf{w}_k$ and the same ILP formulation for global optimization as in training. The result is a new document with k excerpts, one for each topic.

[2] In data sets where such mark-up is not available, one can employ topical segmentation algorithms as an additional preprocessing step.

3.1 Preprocessing

Template Induction A content template specifies the topical structure of documents in one domain. For instance, the template for articles about actors consists of four topics $t_1 \ldots t_4$: *biography*, *early life*, *career*, and *personal life*. Using this template to create the biography of a new actor will ensure that its information coverage is consistent with existing human-authored documents.

We aim to derive these templates by discovering common patterns in the organization of documents in a domain of interest. There has been a sizable amount of research on structure induction ranging from linear segmentation (Hearst, 1994) to content modeling (Barzilay and Lee, 2004). At the core of these methods is the assumption that fragments of text conveying similar information have similar word distribution patterns. Therefore, often a simple segment clustering across domain texts can identify strong patterns in content structure (Barzilay and Elhadad, 2003). Clusters containing fragments from many documents are indicative of topics that are essential for a comprehensive summary. Given the simplicity and robustness of this approach, we utilize it for template induction.

We cluster all section headings $h_{i1} \ldots h_{im}$ from all documents d_i using a repeated bisectioning algorithm (Zhao et al., 2005). As a similarity function, we use cosine similarity weighted with TF*IDF. We eliminate any clusters with low internal similarity (i.e., smaller than 0.5), as we assume these are "miscellaneous" clusters that will not yield unified topics.

We determine the average number of sections k over all documents in our training set, then select the k largest section clusters as topics. We order these topics as $t_1 \ldots t_k$ using a majority ordering algorithm (Cohen et al., 1998). This algorithm finds a total order among clusters that is consistent with a maximal number of pairwise relationships observed in our data set.

Each topic t_j is identified by the most frequent heading found within the cluster – e.g., *Causes*. This set of topics forms the content template for a domain.

Search To retrieve relevant excerpts, we must define appropriate search queries for each topic $t_1 \ldots t_k$. Query reformulation is an active area of research (Agichtein et al., 2001). We have experimented with several of these methods for drawing search queries from representative words in the body text of each topic; however, we find that the best performance is provided by deriving queries from a conjunction of the document title and topic – e.g., *"3-M syndrome" diagnosis*.

Using these queries, we search using Yahoo! and retrieve the first ten result pages for each topic. From each of these pages, we extract all possible excerpts consisting of chunks of text between standardized boundary indicators (such as <p> tags). In our experiments, there are an average of 6 excerpts taken from each page. For each topic t_j of each document we wish to create, the total number of excerpts r found on the Internet may differ. We label the excerpts $e_{j1} \ldots e_{jr}$.

3.2 Selection Model

Our selection model takes the content template $t_1 \ldots t_k$ and the candidate excerpts $e_{j1} \ldots e_{jr}$ for each topic t_j produced in the previous steps. It then selects a series of k excerpts, one from each topic, to create a coherent summary.

One possible approach is to perform individual selections from each set of excerpts $e_{j1} \ldots e_{jr}$ and then combine the results. This strategy is commonly used in multi-document summarization (Barzilay et al., 1999; Goldstein et al., 2000; Radev et al., 2000), where the combination step eliminates the redundancy across selected excerpts. However, separating the two steps may not be optimal for this task — the balance between coverage and redundancy is harder to achieve when a multi-paragraph summary is generated. In addition, a more discriminative selection strategy is needed when candidate excerpts are drawn directly from the web, as they may be contaminated with noise.

We propose a novel joint training algorithm that learns selection criteria for all the topics simultaneously. This approach enables us to maximize both local fit and global coherence. We implement this algorithm using the perceptron framework, as it can be easily modified for structured prediction while preserving convergence guarantees (Daumé III and Marcu, 2005; Snyder and Barzilay, 2007).

In this section, we first describe the structure and decoding procedure of our model. We then present an algorithm to jointly learn the parameters of all topic models.

3.2.1 Model Structure

The model inputs are as follows:

- The title of the desired document
- $t_1 \ldots t_k$ — topics from the content template
- $e_{j1} \ldots e_{jr}$ — candidate excerpts for each topic t_j

In addition, we define feature and parameter vectors:

- $\phi(e_{jl})$ — feature vector for the lth candidate excerpt for topic t_j
- $\mathbf{w}_1 \ldots \mathbf{w}_k$ — parameter vectors, one for each of the topics $t_1 \ldots t_k$

Our model constructs a new article by following these two steps:

Ranking First, we attempt to rank candidate excerpts based on how representative they are of each individual topic. For each topic t_j, we induce a ranking of the excerpts $e_{j1} \ldots e_{jr}$ by mapping each excerpt e_{jl} to a score:

$$score_j(e_{jl}) = \phi(e_{jl}) \cdot \mathbf{w}_j$$

Candidates for each topic are ranked from highest to lowest score. After this procedure, the position l of excerpt e_{jl} within the topic-specific candidate vector is the excerpt's rank.

Optimizing the Global Objective To avoid redundancy between topics, we formulate an optimization problem using excerpt rankings to create the final article. Given k topics, we would like to select one excerpt e_{jl} for each topic t_j, such that the rank is minimized; that is, $score_j(e_{jl})$ is high.

To select the optimal excerpts, we employ integer linear programming (ILP). This framework is

commonly used in generation and summarization applications where the selection process is driven by multiple constraints (Marciniak and Strube, 2005; Clarke and Lapata, 2007).

We represent excerpts included in the output using a set of indicator variables, x_{jl}. For each excerpt e_{jl}, the corresponding indicator variable $x_{jl} = 1$ if the excerpt is included in the final document, and $x_{jl} = 0$ otherwise.

Our objective is to minimize the ranks of the excerpts selected for the final document:

$$\min \sum_{j=1}^{k} \sum_{l=1}^{r} l \cdot x_{jl}$$

We augment this formulation with two types of constraints.

Exclusivity Constraints We want to ensure that exactly one indicator x_{jl} is nonzero for each topic t_j. These constraints are formulated as follows:

$$\sum_{l=1}^{r} x_{jl} = 1 \qquad \forall j \in \{1 \dots k\}$$

Redundancy Constraints We also want to prevent redundancy across topics. We define $sim(e_{jl}, e_{j'l'})$ as the cosine similarity between excerpts e_{jl} from topic t_j and $e_{j'l'}$ from topic $t_{j'}$. We introduce constraints that ensure no pair of excerpts has similarity above 0.5:

$$(x_{jl} + x_{j'l'}) \cdot sim(e_{jl}, e_{j'l'}) \leq 1$$

$$\forall j, j' = 1 \dots k \qquad \forall l, l' = 1 \dots r$$

If excerpts e_{jl} and $e_{j'l'}$ have cosine similarity $sim(e_{jl}, e_{j'l'}) > 0.5$, only one excerpt may be selected for the final document – i.e., either x_{jl} or $x_{j'l'}$ may be 1, but not both. Conversely, if $sim(e_{jl}, e_{j'l'}) \leq 0.5$, both excerpts may be selected.

Solving the ILP Solving an integer linear program is NP-hard (Cormen et al., 1992); however, in practice there exist several strategies for solving certain ILPs efficiently. In our study, we employed *lp_solve*,[3] an efficient mixed integer programming solver which implements the Branch-and-Bound algorithm. On a larger scale, there are several alternatives to approximate the ILP results, such as a dynamic programming approximation to the knapsack problem (McDonald, 2007).

[3] `http://lpsolve.sourceforge.net/5.5/`

Feature	Value
UNI_$word_i$	count of word occurrences
POS_$word_i$	first position of word in excerpt
BI_$word_i$_$word_{i+1}$	count of bigram occurrences
SENT	count of all sentences
EXCL	count of exclamations
QUES	count of questions
WORD	count of all words
NAME	count of title mentions
DATE	count of dates
PROP	count of proper nouns
PRON	count of pronouns
NUM	count of numbers
FIRST_$word_1$	1*
FIRST_$word_1$_$word_2$	1†
SIMS	count of similar excerpts‡

Table 1: Features employed in the ranking model.
* Defined as the first unigram in the excerpt.
† Defined as the first bigram in the excerpt.
‡ Defined as excerpts with cosine similarity > 0.5

Features As shown in Table 1, most of the features we select in our model have been employed in previous work on summarization (Mani and Maybury, 1999). All features except the SIMS feature are defined for individual excerpts in isolation. For each excerpt e_{jl}, the value of the SIMS feature is the count of excerpts $e_{jl'}$ in the same topic t_j for which $sim(e_{jl}, e_{jl'}) > 0.5$. This feature quantifies the degree of repetition within a topic, often indicative of an excerpt's accuracy and relevance.

3.2.2 Model Training

Generating Training Data For training, we are given n original documents $d_1 \dots d_n$, a content template consisting of topics $t_1 \dots t_k$, and a set of candidate excerpts $e_{ij1} \dots e_{ijr}$ for each document d_i and topic t_j. For each section of each document, we add the gold excerpt s_{ij} to the corresponding vector of candidate excerpts $e_{ij1} \dots e_{ijr}$. This excerpt represents the target for our training algorithm. Note that the algorithm does not require annotated ranking data; only knowledge of this "optimal" excerpt is required. However, if the excerpts provided in the training data have low quality, noise is introduced into the system.

Training Procedure Our algorithm is a modification of the perceptron ranking algorithm (Collins, 2002), which allows for joint learning across several ranking problems (Daumé III and Marcu, 2005; Snyder and Barzilay, 2007). Pseudocode for this algorithm is provided in Figure 2.

First, we define $Rank(e_{ij1} \dots e_{ijr}, \mathbf{w}_j)$, which

ranks all excerpts from the candidate excerpt vector $e_{ij1} \dots e_{ijr}$ for document d_i and topic t_j. Excerpts are ordered by $score_j(e_{jl})$ using the current parameter values. We also define $Optimize(e_{ij1} \dots e_{ijr})$, which finds the optimal selection of excerpts (one per topic) given ranked lists of excerpts $e_{ij1} \dots e_{ijr}$ for each document d_i and topic t_j. These functions follow the ranking and optimization procedures described in Section 3.2.1. The algorithm maintains k parameter vectors $\mathbf{w}_1 \dots \mathbf{w}_k$, one associated with each topic t_j desired in the final article. During initialization, all parameter vectors are set to zeros (line 2).

To learn the optimal parameters, this algorithm iterates over the training set until the parameters converge or a maximum number of iterations is reached (line 3). For each document in the training set (line 4), the following steps occur: First, candidate excerpts for each topic are ranked (lines 5-6). Next, decoding through ILP optimization is performed over all ranked lists of candidate excerpts, selecting one excerpt for each topic (line 7). Finally, the parameters are updated in a joint fashion. For each topic (line 8), if the selected excerpt is not similar enough to the gold excerpt (line 9), the parameters for that topic are updated using a standard perceptron update rule (line 10). When convergence is reached or the maximum iteration count is exceeded, the learned parameter values are returned (line 12).

The use of ILP during each step of training sets this algorithm apart from previous work. In prior research, ILP was used as a postprocessing step to remove redundancy and make other global decisions about parameters (McDonald, 2007; Marciniak and Strube, 2005; Clarke and Lapata, 2007). However, in our training, we intertwine the complete decoding procedure with the parameter updates. Our joint learning approach finds per-topic parameter values that are maximally suited for the global decoding procedure for content selection.

Input:
$d_1 \dots d_n$: A set of n documents, each containing k sections $s_{i1} \dots s_{ik}$
$e_{ij1} \dots e_{ijr}$: Sets of candidate excerpts for each topic t_j and document d_i
Define:
$Rank(e_{ij1} \dots e_{ijr}, \mathbf{w}_j)$:
As described in Section 3.2.1:
Calculates $score_j(e_{ijl})$ for all excerpts for document d_i and topic t_j, using parameters $\mathbf{w}_j$.
Orders the list of excerpts by $score_j(e_{ijl})$ from highest to lowest.
$Optimize(e_{i11} \dots e_{ikr})$:
As described in Section 3.2.1:
Finds the optimal selection of excerpts to form a final article, given ranked lists of excerpts for each topic $t_1 \dots t_k$.
Returns a list of k excerpts, one for each topic.
$\phi(e_{ijl})$:
Returns the feature vector representing excerpt e_{ijl}
Initialization:
1 **For** $j = 1 \dots k$
2 Set parameters $\mathbf{w}_j = 0$
Training:
3 Repeat until convergence or while $iter < iter_{max}$:
4 **For** $i = 1 \dots n$
5 **For** $j = 1 \dots k$
6 $Rank(e_{ij1} \dots e_{ijr}, \mathbf{w}_j)$
7 $x_1 \dots x_k = Optimize(e_{i11} \dots e_{ikr})$
8 **For** $j = 1 \dots k$
9 **If** $sim(x_j, s_{ij}) < 0.8$
10 $\mathbf{w}_j = \mathbf{w}_j + \phi(s_{ij}) - \phi(x_i)$
11 $iter = iter + 1$
12 Return parameters $\mathbf{w}_1 \dots \mathbf{w}_k$

Figure 2: An algorithm for learning several ranking problems with a joint decoding mechanism.

4 Experimental Setup

We evaluate our method by observing the quality of automatically created articles in different domains. We compute the similarity of a large number of articles produced by our system and several baselines to the original human-authored articles using ROUGE, a standard metric for summary quality. In addition, we perform an analysis of editor reaction to system-produced articles submitted to Wikipedia.

Data For evaluation, we consider two domains: American Film Actors and Diseases. These domains have been commonly used in prior work on summarization (Weischedel et al., 2004; Zhou et al., 2004; Filatova and Prager, 2005; Demner-Fushman and Lin, 2007; Biadsy et al., 2008). Our text corpus consists of articles drawn from the corresponding categories in Wikipedia. There are 2,150 articles in American Film Actors and 523 articles in Diseases. For each domain, we randomly select 90% of articles for training and test on the remaining 10%. Human-authored articles in both domains contain an average of four topics, and each topic contains an average of 193 words. In order to model the real-world scenario where Wikipedia articles are not always available (as for new or specialized topics), we specifically exclude Wikipedia sources during our search pro-

	Avg. Excerpts	Avg. Sources
Amer. Film Actors		
Search	2.3	1
No Template	4	4.0
Disjoint	4	2.1
Full Model	**4**	**3.4**
Oracle	4.3	4.3
Diseases		
Search	3.1	1
No Template	4	2.5
Disjoint	4	3.0
Full Model	**4**	**3.2**
Oracle	5.8	3.9

Table 2: Average number of excerpts selected and sources used in article creation for test articles.

cedure (Section 3.1) for evaluation.

Baselines Our first baseline, *Search*, relies solely on search engine ranking for content selection. Using the article title as a query – e.g., *Bacillary Angiomatosis*, this method selects the web page that is ranked first by the search engine. From this page we select the first k paragraphs where k is defined in the same way as in our full model. If there are less than k paragraphs on the page, all paragraphs are selected, but no other sources are used. This yields a document of comparable size with the output of our system. Despite its simplicity, this baseline is not naive: extracting material from a single document guarantees that the output is coherent, and a page highly ranked by a search engine may readily contain a comprehensive overview of the subject.

Our second baseline, *No Template*, does not use a template to specify desired topics; therefore, there are no constraints on content selection. Instead, we follow a simplified form of previous work on biography creation, where a classifier is trained to distinguish biographical text (Zhou et al., 2004; Biadsy et al., 2008).

In this case, we train a classifier to distinguish domain-specific text. Positive training data is drawn from all topics in the given domain corpus. To find negative training data, we perform the search procedure as in our full model (see Section 3.1) using only the article titles as search queries. Any excerpts which have very low similarity to the original articles are used as negative examples. During the decoding procedure, we use the same search procedure. We then classify each excerpt as relevant or irrelevant and select the k non-redundant excerpts with the highest relevance confidence scores.

Our third baseline, *Disjoint*, uses the ranking perceptron framework as in our full system; however, rather than perform an optimization step during training and decoding, we simply select the highest-ranked excerpt for each topic. This equates to standard linear classification for each section individually.

In addition to these baselines, we compare against an *Oracle* system. For each topic present in the human-authored article, the *Oracle* selects the excerpt from our full model's candidate excerpts with the highest cosine similarity to the human-authored text. This excerpt is the optimal automatic selection from the results available, and therefore represents an upper bound on our excerpt selection task. Some articles contain additional topics beyond those in the template; in these cases, the *Oracle* system produces a longer article than our algorithm.

Table 2 shows the average number of excerpts selected and sources used in articles created by our full model and each baseline.

Automatic Evaluation To assess the quality of the resulting overview articles, we compare them with the original human-authored articles. We use ROUGE, an evaluation metric employed at the Document Understanding Conferences (DUC), which assumes that proximity to human-authored text is an indicator of summary quality. We use the publicly available ROUGE toolkit (Lin, 2004) to compute recall, precision, and F-score for ROUGE-1. We use the Wilcoxon Signed Rank Test to determine statistical significance.

Analysis of Human Edits In addition to our automatic evaluation, we perform a study of reactions to system-produced articles by the general public. To achieve this goal, we insert automatically created articles[4] into Wikipedia itself and examine the feedback of Wikipedia editors. Selection of specific articles is constrained by the need to find topics which are currently of "stub" status that have enough information available on the Internet to construct a valid article. After a period of time, we analyzed the edits made to the articles to determine the overall editor reaction. We report results on 15 articles in the Diseases category[5].

[4] In addition to the summary itself, we also include proper citations to the sources from which the material is extracted.

[5] We are continually submitting new articles; however, we report results on those that have at least a 6 month history at time of writing.

	Recall	Precision	F-score
Amer. Film Actors			
Search	0.09	0.37	0.13 *
No Template	0.33	0.50	0.39 *
Disjoint	0.45	0.32	0.36 *
Full Model	**0.46**	**0.40**	**0.41**
Oracle	0.48	0.64	0.54 *
Diseases			
Search	0.31	0.37	0.32 †
No Template	0.32	0.27	0.28 *
Disjoint	0.33	0.40	0.35 *
Full Model	**0.36**	**0.39**	**0.37**
Oracle	0.59	0.37	0.44 *

Table 3: Results of ROUGE-1 evaluation.
* Significant with respect to our full model for $p \leq 0.05$.
† Significant with respect to our full model for $p \leq 0.10$.

Type	Count
Total articles	15
Promoted articles	10
Edit types	
Intra-wiki links	36
Formatting	25
Grammar	20
Minor topic edits	2
Major topic changes	1
Total edits	85

Table 4: Distribution of edits on Wikipedia.

Since Wikipedia is a live resource, we do not repeat this procedure for our baseline systems. Adding articles from systems which have previously demonstrated poor quality would be improper, especially in Diseases. Therefore, we present this analysis as an additional observation rather than a rigorous technical study.

5 Results

Automatic Evaluation The results of this evaluation are shown in Table 3. Our full model outperforms all of the baselines. By surpassing the *Disjoint* baseline, we demonstrate the benefits of joint classification. Furthermore, the high performance of both our full model and the *Disjoint* baseline relative to the other baselines shows the importance of structure-aware content selection. The *Oracle* system, which represents an upper bound on our system's capabilities, performs well.

The remaining baselines have different flaws: Articles produced by the *No Template* baseline tend to focus on a single topic extensively at the expense of breadth, because there are no constraints to ensure diverse topic selection. On the other hand, performance of the *Search* baseline varies dramatically. This is expected; this baseline relies heavily on both the search engine and individual web pages. The search engine must correctly rank relevant pages, and the web pages must provide the important material first.

Analysis of Human Edits The results of our observation of editing patterns are shown in Table 4. These articles have resided on Wikipedia for a period of time ranging from 5-11 months. All of them have been edited, and no articles were removed due to lack of quality. Moreover, ten automatically created articles have been promoted by human editors from stubs to regular Wikipedia entries based on the quality and coverage of the material. Information was removed in three cases for being irrelevant, one entire section and two smaller pieces. The most common changes were small edits to formatting and introduction of links to other Wikipedia articles in the body text.

6 Conclusion

In this paper, we investigated an approach for creating a multi-paragraph overview article by selecting relevant material from the web and organizing it into a single coherent text. Our algorithm yields significant gains over a structure-agnostic approach. Moreover, our results demonstrate the benefits of structured classification, which outperforms independently trained topical classifiers. Overall, the results of our evaluation combined with our analysis of human edits confirm that the proposed method can effectively produce comprehensive overview articles.

This work opens several directions for future research. Diseases and American Film Actors exhibit fairly consistent article structures, which are successfully captured by a simple template creation process. However, with categories that exhibit structural variability, more sophisticated statistical approaches may be required to produce accurate templates. Moreover, a promising direction is to consider hierarchical discourse formalisms such as RST (Mann and Thompson, 1988) to supplement our template-based approach.

Acknowledgments

The authors acknowledge the support of the NSF (CAREER grant IIS-0448168, grant IIS-0835445, and grant IIS-0835652) and NIH (grant V54LM008748). Thanks to Mike Collins, Julia Hirschberg, and members of the MIT NLP group for their helpful suggestions and comments. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors, and do not necessarily reflect the views of the funding organizations.

References

Eugene Agichtein, Steve Lawrence, and Luis Gravano. 2001. Learning search engine specific query transformations for question answering. In *Proceedings of WWW*, pages 169–178.

Regina Barzilay and Noemie Elhadad. 2003. Sentence alignment for monolingual comparable corpora. In *Proceedings of EMNLP*, pages 25–32.

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of HLT-NAACL*, pages 113–120.

Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of ACL*, pages 550–557.

Fadi Biadsy, Julia Hirschberg, and Elena Filatova. 2008. An unsupervised approach to biography production using wikipedia. In *Proceedings of ACL/HLT*, pages 807–815.

James Clarke and Mirella Lapata. 2007. Modelling compression with discourse constraints. In *Proceedings of EMNLP-CoNLL*, pages 1–11.

William W. Cohen, Robert E. Schapire, and Yoram Singer. 1998. Learning to order things. In *Proceedings of NIPS*, pages 451–457.

Michael Collins. 2002. Ranking algorithms for named-entity extraction: Boosting and the voted perceptron. In *Proceedings of ACL*, pages 489–496.

Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. 1992. *Intoduction to Algorithms*. The MIT Press.

Hal Daumé III and Daniel Marcu. 2005. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *Proceedings of HLT/EMNLP*, pages 97–104.

Dina Demner-Fushman and Jimmy Lin. 2007. Answering clinical questions with knowledge-based and statistical techniques. *Computational Linguistics*, 33(1):63–103.

Elena Filatova and John M. Prager. 2005. Tell me what you do and I'll tell you what you are: Learning occupation-related activities for biographies. In *Proceedings of HLT/EMNLP*, pages 113–120.

Elena Filatova, Vasileios Hatzivassiloglou, and Kathleen McKeown. 2006. Automatic creation of domain templates. In *Proceedings of ACL*, pages 207–214.

Atsushi Fujii and Tetsuya Ishikawa. 2004. Summarizing encyclopedic term descriptions on the web. In *Proceedings of COLING*, page 645.

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *Proceedings of NAACL-ANLP*, pages 40–48.

Marti A. Hearst. 1994. Multi-paragraph segmentation of expository text. In *Proceedings of ACL*, pages 9–16.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of ACL*, pages 74–81.

Inderjeet Mani and Mark T. Maybury. 1999. *Advances in Automatic Text Summarization*. The MIT Press.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Tomasz Marciniak and Michael Strube. 2005. Beyond the pipeline: Discrete optimization in NLP. In *Proceedings of CoNLL*, pages 136–143.

Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *Proceedings of EICR*, pages 557–564.

Vivi Nastase and Michael Strube. 2008. Decoding wikipedia categories for knowledge acquisition. In *Proceedings of AAAI*, pages 1219–1224.

Vivi Nastase. 2008. Topic-driven multi-document summarization with encyclopedic knowledge and spreading activation. In *Proceedings of EMNLP*, pages 763–772.

Dragomir R. Radev, Hongyan Jing, and Malgorzata Budzikowska. 2000. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *Proceedings of ANLP/NAACL*, pages 21–29.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge.

Benjamin Snyder and Regina Barzilay. 2007. Multiple aspect ranking using the good grief algorithm. In *Proceedings of HLT-NAACL*, pages 300–307.

Ralph M. Weischedel, Jinxi Xu, and Ana Licuanan. 2004. A hybrid approach to answering biographical questions. In *New Directions in Question Answering*, pages 59–70.

Fei Wu and Daniel S. Weld. 2007. Autonomously semantifying wikipedia. In *Proceedings of CIKM*, pages 41–50.

Ying Zhao, George Karypis, and Usama Fayyad. 2005. Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, 10(2):141–168.

L. Zhou, M. Ticrea, and Eduard Hovy. 2004. Multi-document biography summarization. In *Proceedings of EMNLP*, pages 434–441.