Hierarchical Summarization: Scaling Up Multi-Document Summarization

Janara Christensen **Stephen Soderland**
Computer Science & Engineering
University of Washington
Seattle, USA
janara@cs.washington.edu
soderlan@cs.washington.edu

Gagan Bansal **Mausam**
Computer Science & Engineering
Indian Institute of Technology
Delhi, India
gaganbansal1993@gmail.com
mausam@cse.iitd.ac.in

Abstract

Multi-document summarization (MDS) systems have been designed for short, unstructured summaries of 10-15 documents, and are inadequate for larger document collections. We propose a new approach to scaling up summarization called *hierarchical summarization*, and present the first implemented system, SUMMA.

SUMMA produces a hierarchy of relatively short summaries, in which the top level provides a general overview and users can navigate the hierarchy to drill down for more details on topics of interest. SUMMA optimizes for coherence as well as coverage of salient information. In an Amazon Mechanical Turk evaluation, users preferred SUMMA ten times as often as flat MDS and three times as often as timelines.

1 Introduction

The explosion in the number of documents on the Web necessitates automated approaches that organize and summarize large document collections on a complex topic. Existing methods for multi-document summarization (MDS) are designed to produce short summaries of 10-15 documents.[1] MDS systems do not scale to data sets ten times larger and proportionately longer summaries: they either cannot run on large input or produce a disorganized summary that is difficult to understand.

We present a novel MDS paradigm, *hierarchical summarization*, which operates on large document collections, creating summaries that organize the information coherently. It mimics how someone with a general interest in a complex topic would learn about it from an expert – first, the expert would provide an overview, and then more specific information about various aspects. Hierarchical summarization has the following novel characteristics:

- The summary is hierarchically organized along one or more organizational principles such as time, location, entities, or events.
- Each non-leaf summary is associated with a set of child summaries where each gives details of an element (e.g. sentence) in the parent summary.
- A user can navigate within the hierarchical summary by clicking on an element of a parent summary to view the associated child summary.

For example, given the topic, "1998 embassy bombings," the first summary (Figure 1) might

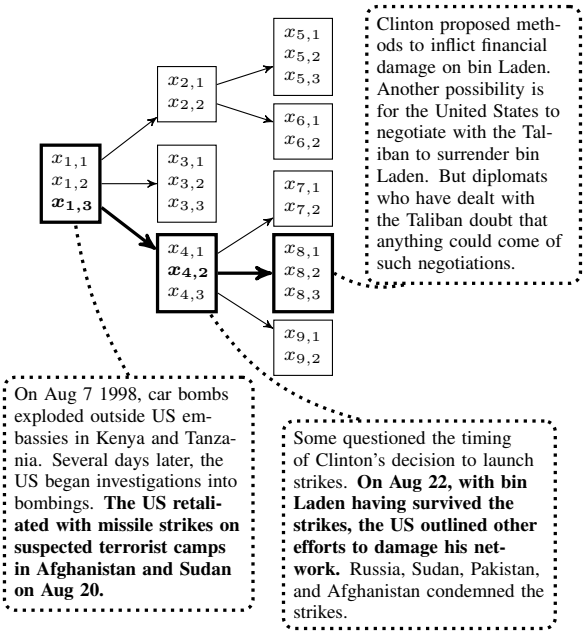

Figure 1: A hierarchical summary of the 1998 embassy bombings. Each rectangle represents a summary and each $x_{i,j}$ is a sentence within a summary. The root summary provides an overview of the events of August 1998. When the third sentence is selected, a more detailed summary of the missile strikes is displayed. Selecting the second sentence of that summary produces a more detailed summary of the US' options.

[1] In the DUC evaluations, summaries have a budget of 665 bytes and cover 10 documents.

mention that the US retaliated by striking Afghanistan and Sudan. The user can click on this information to learn more about these attacks. In this way, the system can present large amounts of information without overwhelming the user, and the user can tailor the output to their interests.

In this paper, we describe SUMMA, the first hierarchical summarization system for multi-document summarization.[2] It operates on a corpus of related news articles. SUMMA hierarchically clusters the sentences by time, and then summarizes the clusters using an objective function that optimizes salience and coherence.

We conducted an Amazon Mechanical Turk (AMT) evaluation where AMT workers compared the output of SUMMA to that of timelines and flat summaries. SUMMA output was judged superior more than three times as often as timelines, and users learned more in twice as many cases. Users overwhelmingly preferred hierarchical summaries to flat summaries (92%) and learned just as much.

Our main contributions are as follows:

- We introduce and formalize the novel task of hierarchical summarization.
- We present SUMMA, the first hierarchical summarization system, which operates on news corpora and summarizes over an order of magnitude more documents than traditional MDS systems, producing summaries an order of magnitude larger.
- We present a user study which demonstrates the value of hierarchical summarization over timelines and flat multi-document summaries in learning about a complex topic.

In the next section, we formalize hierarchical summarization. We then describe our methodology to implement the SUMMA hierarchical summarization system: hierarchical clustering in Section 3 and creating summaries based on that clustering in Section 4. We discuss our experiments in Section 5, related work in Section 6, and conclusions in Section 7.

2 Hierarchical Summarization

We propose a new task for large-scale summarization called *hierarchical summarization*. Input to a hierarchical summarization system is a set of related documents D and a budget b for each summary within the hierarchy (in bytes, words, or sentences). The output is the hierarchical summary H, which we define formally as follows.

[2] http://knowitall.cs.washington.edu/summa/

Definition A *hierarchical summary* H of a document collection D is a set of summaries X organized into a hierarchy. The top of the hierarchy is a summary X_1 representing all of D, and each summary X_i consists of summary units $x_{i,j}$ (*e.g.* the jth sentence of summary i) that point to a child summary, except at the leaf nodes of the hierarchy.

A child summary adds more detail to the information in its parent summary unit. The child summary may include sub-events or background and reactions to the event or topic in the parent.

We define several metrics in Section 4 for a well-constructed hierarchical summary. Each summary should maximize coverage of *salient* information; it should minimize *redundancy*; and it should have *intra-cluster coherence* as well as *parent-to-child coherence*.

Hierarchical summarization has two important strengths in the context of large-scale summarization. First, the information presented at the start is small and grows only as the user directs it, so as not to overwhelm the user. Second, each user directs his or her own experience, so a user interested in one aspect need only explore that section of the data without having to view or understand the entire summary. The parent-to-child links provide a means for a user to navigate, drilling down for more details on topics of interest.

There are several possible organizing principles for the hierarchy – by date, by entities, by locations, or by events. Some organizing principles will fit the data in a document collection better than others. A system may select different organization for different portions of the hierarchy, for example, organizing first by location or prominent entity and then by date for the next level.

3 Hierarchical Clustering

Having defined the task, we now describe the methodology behind our implementation, SUMMA. In future work we intend to design a system that dynamically selects the best organizing principle for each level of the hierarchy. In this first implementation, we have opted for temporal organization, since this is generally the most appropriate for news events.

The problem of hierarchical summarization as described in Section 2 has all of the requirements of MDS, and additional complexities of inducing a hierarchical structure, processing an order of magnitude bigger input, generating a much larger output, and enforcing coherence between parent and

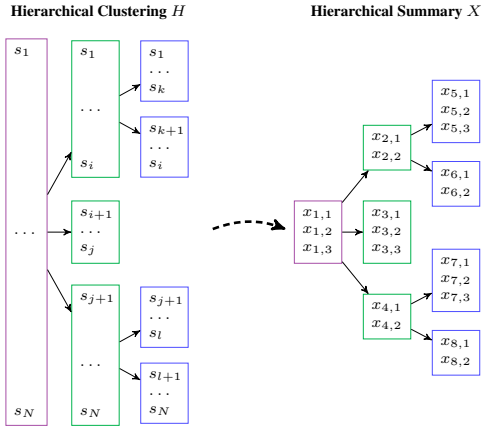

Figure 2: Examples of input and output to hierarchical summarization. The input sentences are $s \in S$, the number of input sentences is N, and the summary sentences are $x \in X$.

child summaries.

We simplify the problem by decomposing it into two steps: hierarchical clustering and summarizing over the clustering (see Figure 2 for an example). A hierarchical clustering is a tree in which if a cluster g_p is the parent of cluster g_c, then each sentence in g_c is also in g_p. This organizes the information into manageable, semantically-related sections and induces a hierarchical structure over the input.

The hierarchical clustering serves as input to the second step – summarizing given the hierarchy. The hierarchical summary follows the hierarchical structure of the clustering. Each node in the hierarchy has an associated flat summary, which summarizes the sentences in that cluster. Moreover, the number of sentences in a flat summary is exactly equal to the number of child clusters of the node, since the user will click a sentence to get to the child summary. See Figure 2 for an illustration of this correspondence.

Because we are interested in *temporal* hierarchical summarization, we hierarchically cluster all the sentences in the input documents by time. Unfortunately, neither agglomerative nor divisive clustering is suitable, since both assume a binary split at each node (Berkhin, 2006). The number of clusters at each split should be what is most natural for the input data. We design a recursive clustering algorithm that automatically chooses the appropriate number of clusters at each split.

Before clustering, we timestamp all sentences. We use SUTime (Chang and Manning, 2012) to normalize temporal references, and we parse the sentences with the Stanford parser (Klein and Manning, 2003) and use a set of simple heuristics to determine if the timestamps in the sentence refer to the root verb. If no timestamp is given, we use the article date.

3.1 Temporal Clustering

After acquiring the timestamps, we must hierarchically cluster the sentences into sets that make sense to summarize together. Since we wish to partition along the temporal dimension, our problem reduces to identifying the best dates at which to split a cluster into subclusters. We identify these dates by looking for bursts of activity.

News tends to be *bursty* – many articles on a topic appear at once and then taper out (Kleinberg, 2002). For example, Figure 3 shows the number of articles per day related to the 1998 embassy bombings published in the New York Times (identified using a key word search). There were two main events – on the 7th, the embassies were bombed and on the 20th, the US retaliated through missile strikes. The figure shows a correspondence between these events and news spikes.

Ideal splits for this example would occur just before each spike in coverage. However, when there is little differentiation in news coverage, we prefer clusters evenly spaced across time. We thus choose clusters $C = \{c_1, \ldots, c_k\}$ as follows:

$$\underset{C}{\text{maximize}} \quad B(C) + \alpha E(C) \tag{1}$$

where C is a clustering, $B(C)$ is the burstiness of the set of clusters, $E(C)$ is the evenness of the clusters, and α is the tradeoff parameter.

$$B(C) = \sum_{c \in C} burst(c) \tag{2}$$

$burst(c)$ is the difference in the number of sentences published the day before the first date in c and the average number of sentences published on the first and second date of c:

$$burst(c) = \frac{pub(d_i) + pub(d_{i+1})}{2} - pub(d_{i-1}) \tag{3}$$

where d is a date indexed over time, such that d_j is a day before d_{j+1}, and d_i is the first date in c. $pub(d_i)$ is the number of sentences published on d_i. The evenness of the split is measured by:

$$E(C) = \min_{c \in C} size(c) \tag{4}$$

where $size(c)$ is the number of dates in cluster c.

We perform hierarchical clustering top-down, at each point solving for Equation 1. α was set using a grid-search over a development set.

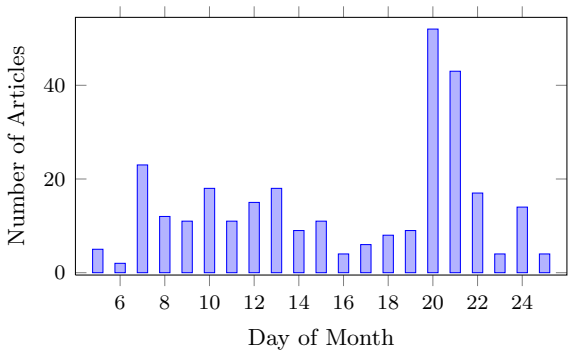

Figure 3: News coverage by date for the embassy bombings in Tanzania and Kenya. There are spikes in the number of articles published at the two major events.

3.2 Choosing the number of clusters

We cannot know *a priori* the number of clusters for a given topic. However, when the number of clusters is too large for the given summary budget, the sentences will have to be too short, and when the number of clusters is too small, we will not use enough of the budget. We set the maximum number of clusters k_{max} and minimum number of clusters k_{min} to be a function of the budget b and the average sentence length in the cluster s_{avg}, such that $k_{max} \cdot s_{avg} \leq b$ and $k_{min} \cdot s_{avg} \geq b/2$.

Given a maximum and minimum number of clusters, we must determine the appropriate number of clusters. At each level, we cluster the sentences by the method described above and choose the number of clusters k according to the gap statistic (Tibshirani et al., 2000). Specifically, for each level, the algorithm will cluster repeatedly with k varying from the minimum to the maximum. The algorithm will return the k that maximizes the gap statistic:

$$Gap_n(k) = E_n^*\{\log(W_k)\} - log(W_k) \qquad (5)$$

where W_k is the score for the clusters computed with Equation 1, and E_n^* is the expectation under a sample of size n from a reference distribution.

Ideally, the maximum depth of the clustering would be a function of the number of sentences in each cluster, but in our implementation, we set the maximum depth to three, which works well for the size of the datasets we use (300 articles).

4 Summarizing within the Hierarchy

After the sentences are clustered, we have a structure for the hierarchical summary that dictates the number of summaries and the number of sentences in each summary. We also have the set of sentences from which each summary is drawn.

Intuitively, each cluster summary in the hierarchical summary should convey the most **salient** information in that cluster. Furthermore, the hierarchical summary should not include **redundant** sentences. A hierarchical summary that is only salient and nonredundant may still not be suitable if the sentences within a cluster summary are disconnected or if the parent sentence for a summary does not relate to the child summary. Thus, a hierarchical summary must also have **intra-cluster coherence** and **parent-to-child coherence**.

4.1 Salience

Salience is the value of each sentence to the topic from which the documents are drawn. We measure salience of a summary ($Sal(X)$) as the sum of the saliences of individual sentences ($\sum_i Sal(x_i)$). Following previous research in MDS, we computed individual saliences using a linear regression classifier trained on ROUGE scores over the DUC'03 dataset (Lin, 2004; Christensen et al., 2013). This method finds those sentences more salient that mention nouns or verbs that occur frequently in the cluster.

In preliminary experiments, we noticed that many sentences that were *reaction* sentences were given a higher salience than *action* sentences. For example, the reaction sentence, "President Clinton vowed to track down the perpetrators behind the bombs that exploded outside the embassies in Tanzania and Kenya on Friday," would have a higher score than the *action* sentence, "Bombs exploded outside the embassies in Tanzania and Kenya on Friday." This problem occurs because the first sentence has a higher ROUGE score (it covers more important words than the second sentence). To adjust for this problem, we use only words identified in the main clause (heuristically identified via the parse tree) to compute our salience scores.

4.2 Redundancy

We identify redundant sentences using a linear regression classifier trained on a manually labeled subset of the DUC'03 sentences. The features include shared noun counts, sentence length, TF*IDF cosine similarity, timestamp difference, and features drawn from information extraction such as number of shared tuples in Open IE (Mausam et al., 2012).

4.3 Summary Coherence

We require two types of coherence: coherence between the parent and child summaries and coherence within each summary X_i.

We rely on the approximate discourse graph (ADG) that was proposed in (Christensen et al., 2013) as the basis for measuring coherence. Each node in the ADG is a sentence from the dataset. An edge from sentence s_i to s_j with positive weight indicates that s_j may follow s_i in a coherent summary, *e.g.* continued mention of an event or entity, or coreference link between s_i and s_j. A negative edge indicates an unfulfilled discourse cue or co-reference mention.

Parent-to-Child Coherence: Users navigate the hierarchical summary from parent sentence to child summary, so if the parent sentence bears no relation to the child summary, the user will be understandably confused. The parent sentence must have positive evidence of coherence with the sentences in its child summary.

We estimate parent to child coherence as the coherence between a parent sentence and each sentence in its child summary as:

$$PCoh(X) = \sum_{c \in C} \sum_{i=1..|X_c|} w_{G+}(x_c^p, x_{c,i})) \quad (6)$$

where x_c^p is the parent sentence for cluster c and $w_{G+}(x_c^p, x_{c,i})$ is the sum of the positive edge weights from x_c^p to $x_{c,i}$ in the ADG G.

Intra-cluster Coherence: In traditional MDS, the documents are usually quite focused, allowing for highly focused summaries. In hierarchical summarization, however, a cluster summary may span hundreds of documents and a wide range of information. For this reason, we may consider a summary acceptable even if it has limited positive evidence of coherence in the ADG, as long as there is no negative evidence in the form of negative edges. For example, the following is a reasonable summary for events spanning two weeks:

s_1 *Bombs exploded at two US embassies.*

s_2 *US missiles struck in Afghanistan and Sudan.*

Our measure of intra-cluster coherence minimizes the number of *missing references*. These are coreference mentions or discourse cues where none of the sentences read before (either in an ancestor summary or in the current summary) contain an antecedent:

$$CCoh(X) = -\sum_{c \in C} \sum_{i=1..|X_c|} \#missingRef(x_{c,i}) \quad (7)$$

4.4 Objective Function

Having estimated salience, redundancy, and two forms of coherence, we can now put this information together into a single objective function that measures the quality of a candidate hierarchical summary.

Intuitively, the objective function should balance salience and coherence. Furthermore, the summary should not contain redundant information and each cluster summary should honor the given budget, i.e., maximum summary length b. We treat redundancy and budget as hard constraints and coherence and salience as soft constraints. Lastly, we require that sentences are drawn from the cluster that they represent and that the number of sentences in the summary corresponding to each non-leaf cluster c is equivalent to the number of child clusters of c. We optimize:

$$\begin{aligned} \textbf{maximize:} \quad & F(x) \triangleq Sal(X) + \beta PCoh(X) + \gamma CCoh(X) \\ s.t. \quad & \forall c \in C : \textstyle\sum_{i=1..|X_c|} len(x_{c,i}) < b \\ & \forall x_i, x_j \in X : \text{redundant}(x_i, x_j) = 0 \\ & \forall c \in C, \forall x_c \in X_c : x_c \in c \\ & \forall c \in C : |X_c| = \#children(c) \end{aligned}$$

The tradeoff parameters β and γ were set based on a development set.

4.5 Algorithm

Optimizing this objective function is NP-hard, so we approximate a solution by using beam search over the space of partial hierarchical summaries. Notice the contribution from a sentence depends on individual salience, coherence ($CCoh$) based on sentences visible on the user path down the hierarchy to this sentence, and coherence ($PCoh$) based on its parent sentence and its child summary. Since most of the sentence contributions depend on the path from the root to the sentence, we build our partial summary by incrementally adding a sentence top-down in the hierarchy and from first sentence to last within a cluster summary.

To account for $PCoh$, we estimate the contribution of the sentence by jointly identifying its best child summary. However, we do not fix the child summary at this time – we simply use it to estimate $PCoh$ when using that sentence. Since computing the best child summary is also intractable we approximate a solution by a local search algorithm over the child cluster.

Overall, our algorithm is a two level nested search algorithm – beam search in the outer loop to

search through the space of partial summaries and local search (hill climbing with random restarts) in the inner loop to pick the best sentence to add to the existing partial summary. We use a beam of size ten in our implementation.

5 Experiments

Our experiments are designed to evaluate how effective hierarchical summarization is in summarizing a large, complex topic and how well this helps users learn about the topic. Our evaluation addresses the following questions:

- Do users prefer hierarchical summaries for topic exploration? (Section 5.1)
- Are hierarchical summaries more effective than other methods for learning about complex events? (Section 5.2)
- How informative are the hierarchical summaries compared to the other methods? (Section 5.3)
- How coherent is the hierarchical structure in the summaries? (Section 5.4)

We compared SUMMA against two baseline systems which represent the main NLP methods for large-scale summarization: an algorithm for creating timelines over sentences (Chieu and Lee, 2004),[3] and a state-of-the-art flat MDS system (Lin and Bilmes, 2011).[4] Each system was given the same budget (over 10 times the traditional MDS budget, which is 665 bytes).

We evaluated the questions on ten news topics, representing a range of tasks: (1) Pope John Paul II's death and the 2005 Papal Conclave, (2) Bush v. Gore, (3) the Tulip Revolution, (4) Daniel Pearl's kidnapping, (5) the Lockerbie bombing handover of suspects, (6) the Kargil War, (7) NATO's bombing of Yugoslavia in 1999, (8) Pinochet's arrest in London, (9) the 2005 London bombings, and (10) the crash and investigation of SwissAir Flight 111. We chose topics containing a set of related events that unfolded over several months and were prominent enough to be reported in at least 300 articles.

We drew our articles from the Gigaword corpus, which contains articles from the New York Times and other major newspapers. For each topic, we used the 300 documents that best matched a key word search. We selected topics which were between five and fifteen years old so that evaluators would have relatively less pre-existing knowledge about the topic.

5.1 User Preference

In our first experiment, we simply wished to evaluate which system users most prefer. We hired Amazon Mechanical Turk (AMT) workers and assigned two topics to each worker. We paired up workers such that one worker would see output from SUMMA for the first topic and a competing system for the second and the other worker would see the reverse. For quality control, we asked workers to complete a qualification task first, in which they were required to write a short summary of a news article. We also manually removed spam from our results. Previous work has used AMT workers for summary evaluations and has shown high correlations with expert ratings (Christensen et al., 2013). Five workers were hired to view each topic-system pair.

We asked the workers to choose which format they preferred and to explain why. The results are as follows:

SUMMA	**76%**	TIMELINE	24%
SUMMA	**92%**	FLAT-MDS	8%

Users preferred the hierarchical summaries three times more often than timelines and over ten times more often than flat summaries. When we examined the reasons given by the users, we found that the people who preferred the hierarchical summaries liked that they gave a big picture overview and were then allowed to drill down deeper. Some also explained that it was easier to remember information when presented with the overview first. Typical responses included, "Could gather and absorb the information at my own pace," and, "Easier to follow and understand." When users preferred the timelines, they usually remarked that it was more familiar, i.e. "I liked the familiarity of the format. I am used to these timelines and they feel comfortable." Users complained that the flat summaries were disjointed, confusing, and very frustrating to read.

5.2 Knowledge Acquisition

Evaluating how much a user learned is inherently difficult, more so when the goal is to allow the user the freedom to explore information based on individual interest. For this reason, instead of asking a set of predefined questions, we assess the knowl-

[3] Unfortunately, we were unable to obtain more recent timeline systems from authors of the systems.

[4] (Christensen et al., 2013) is a state-of-the-art coherent MDS system, but does not scale to 300 documents.

edge gain by following the methodology of (Shahaf et al., 2012) – asking users to write a paragraph summarizing the information learned.

Using the same setup as in the previous experiment, for each topic, five AMT workers spent three minutes reading through a timeline or summary and were then asked to write a description of what they had learned. Workers were not allowed to see the timeline or summary while writing. We collected five descriptions for each topic-system combination. We then asked other AMT workers to read and compare the descriptions written by the first set of workers. Each evaluator was presented with a corresponding Wikipedia article and descriptions from a pair of users (timeline vs. SUMMA or flat MDS vs. SUMMA). The descriptions were randomly ordered to remove bias. The workers were asked which user appeared to have learned more and why. For each pair of descriptions, four workers evaluated the pair. Standard checks such as approval rating, location filtering, etc. were used for removing spam. The results of this experiment are as follows:

Prefer		Indiff.	Prefer	
SUMMA	**58%**	17%	TIMELINE	25%
SUMMA	**40%**	22%	FLAT-MDS	38%

Descriptions written by workers using SUMMA were preferred over twice as often as those from timelines. We looked more closely at those cases where the participants either preferred the timelines or were indifferent and found that this preference was most common when the topic was not dominated by a few major events, but was instead a series of similarly important events. For example, in the kidnapping and beheading of Daniel Pearl there were two or three obviously major events, whereas in the Kargil War there were many smaller important events. In latter cases, the hierarchical summaries provided little advantage over the timelines because it was more difficult to arrange the sentences hierarchically.

Since SUMMA was judged to be so much superior to flat MDS systems in Section 5.1, it is surprising that users descriptions from flat MDS were preferred nearly as often as those from SUMMA. While the flat summaries were disjointed, they were good at including salient information, with the most salient tending to be near the start of the summary. Thus, descriptions from both SUMMA and flat MDS generally covered the most salient information.

5.3 Informativeness

In this experiment, we assess the salience of the information captured by the different systems, and the ability of SUMMA to organize the information so that more important information is placed at higher levels.

ROUGE Evaluation: We first automatically assessed informativeness by calculating the ROUGE-1 scores of the output of each of the systems. For the gold standard comparison summary, we use the Wikipedia articles for the topics.[5] Note that there is no good translation of ROUGE for hierarchical summarization. Thus, we simply use the traditional ROUGE metric, which will not capture any of the hierarchical format. This score will essentially serve as a rough measure of coverage of the entire summary to the Wikipedia article. The scores for each of the systems are as follows:

	P	R	F1
SUMMA	0.25	**0.67**	0.31
TIMELINE	0.28	0.65	0.33
FLAT-MDS	**0.30**	0.64	**0.34**

None of the differences are significant. From this evaluation, one can gather that the systems have similar coverage of the Wikipedia articles.

Manual Evaluation: While ROUGE serves as a rough measure of coverage, we were interested in gathering more fine-grained information on the informativeness of each system. We performed an additional manual evaluation that assesses the recall of important events for each system.

We first identified which events were most important in a news story. Because reading 300 articles per topic is impractical, we asked AMT workers to read a Wikipedia article on the same topic and then identify the three most important events and the five most important secondary events. We aggregated responses from ten workers per topic and chose the three most common primary and five most common secondary events.

One of the authors then manually identified the presence of these events in the hierarchical summaries, the timelines and the flat MDS summaries. Below we show event recall (the percentage of the events that were mentioned).

[5] We excluded one topic (the handover of the Lockerbie bombing suspects) because the corresponding Wikipedia article had insufficient information.

Events	SUMMA	TIMELINE	FLAT-MDS
Prim.	**96%**	74%	93%
Sec.	**76%**	53%	64%

The difference in recall between SUMMA and TIMELINE was significant in both cases, and the difference between SUMMA and FLAT-MDS was not. In general, the flat summaries were quite redundant, which contributed to the slightly lower event recall. The timelines, on the other hand, were both incoherent and at the same time reported less important facts.

We also evaluated at what level in the hierarchy the events were identified for the hierarchical summaries. The event recall shows the percentage of events mentioned at that level or above in the hierarchical summary:

Events	Level 1	Level 2	Level 3
Prim.	63%	81%	96%
Sec.	27%	51%	76%

81% of the primary events are present in the first or second level, and 76% of the secondary events are mentioned by the third level. While recognizing primary events is relatively simple because they are repeated frequently, identification of important secondary events often requires external knowledge.

5.4 Parent-to-Child Coherence

We next tested the hierarchical coherence. One of the authors graded how much each non-leaf sentence in a summary was coherent with its child summary on a scale of one to five, with one being incoherent and five being perfectly coherent. We used the coherence scale from DUC'04.[6]

	Level 1	Level 2
Coherence	3.8	3.4

We found that for the top level of the summary, the parent sentence generally represented the most important event in the cluster and the child summary usually expressed details or reactions of the event. The lower coherence scores were often the result of too few lexical connections or lack of a theme or story. While the facts of the sentences made sense together, the summaries sometimes did not read as if they were written by a human, but as a series of disparate sentences.

For the second level, the problems were more basic. The parent sentence occasionally expressed a less important fact that the child summary did not then expand on or, more commonly, the child summary was not focused enough. This result stems from two problems in our algorithm. First, summarizing sentences are rare, making good choices for parent sentences difficult to find. The second problem relates to the difficulty in identifying whether two sentences are on the same topic. For example, suppose the parent sentence is, "A Swissair plane Wednesday night crashed off Nova Scotia, Canada." A very good child sentence is, "The airline confirmed that all passengers died." However, based on their surface features, the sentence, "A plane made an unscheduled landing after a Swissair plane crashed off the coast of Canada," appears to be a better choice.

Even though there is scope for improvement, we find these coherence scores encouraging for a first algorithm for the task.

6 Related Work

Traditional approaches to large-scale summarization have included flat summaries and timelines. There are two primary shortcomings to these approaches: first, they require the user to sort through large amounts of potentially overwhelming information, and second, the output is static – users with different interests will see the same information. Below we describe related work on traditional MDS, structured summaries, timelines, discovering threads of documents and the uses of hierarchies in generating summaries.

6.1 Traditional MDS

Traditionally, MDS systems have focused on three to six sentence summaries covering 10-15 documents. Most extractive summarization research aims to maximize coverage while reducing redundancy (e.g. (Carbonell and Goldstein, 1998; Saggion and Gaizauskas, 2004; Radev et al., 2004)). Lin and Bilmes (2011) proposed a state-of-the-art system that uses submodularity in sentence selection to accomplish these goals. Christensen et al. (2013) presented an algorithm for coherent MDS, but it does not scale to larger output.

Structured Summaries: Some research has explored generating structured summaries. These approaches attempt to identify major aspects of a topic, but do not compile content to describe those aspects. Rather, they rely on pre-existing, labeled paragraphs (for example, a paragraph titled, "Symptoms of Meningitis"). Aspects are identified either by a training corpus of articles in the

[6] http://duc.nist.gov/duc2004/quality.questions.txt

same domain (Sauper and Barzilay, 2009), by an entity-aspect LDA model (Li et al., 2010), or by Wikipedia templates of related topics (Yao et al., 2011). These methods assume a common structure for all topics in a category, and do not allow for more than two levels in the structure.

Timeline Generation: Recent papers in timeline generation have emphasized the relationship with summarization. Yan et al. (2011b) balanced coherence and diversity to create timelines, Yan et al. (2011a) used inter-date and intra-date sentence dependencies, and Chieu and Lee (2004) used sentence similarity. Others have emphasized identifying important dates, primarily by bursts of news (Swan and Allen, 2000; Akcora et al., 2010; Hu et al., 2011; Kessler et al., 2012). While timelines can be useful for understanding events, they do not generalize to other domains. Additionally, long timelines can be overwhelming, short timelines have low information content, and there is no method for personalized exploration.

Document Threads: A related track of research investigates discovering threads of documents. While we aim to summarize collections of information, this track seeks to identify relationships between documents. This research operates on the document level, while ours operates on the sentence level. Shahaf and Guestrin (2010) formalized the characteristics of a good chain of articles and proposed an algorithm to connect two specified articles. Gillenwater et al. (2012) proposed a probabilistic technique for extracting a diverse set of threads from a given collection. Shahaf et al. (2012) extended work on coherent threads to finding coherent maps of documents, where a map is set of intersecting threads representing how the threads interact and relate.

Summarization and Hierarchies: A few papers have examined the relationship between summarization and hierarchies. Some focused on creating a hierarchical summary of a single document (Buyukkokten et al., 2001; Otterbacher et al., 2006), relying on the structure inherent in single documents. Others investigated creating hierarchies of words or phrases to organize documents (Lawrie et al., 2001; Lawrie, 2003; Takahashi et al., 2007; Haghighi and Vanderwende, 2009).

Other research identifies the hierarchical structure of the documents and generates a summary that prioritizes more general information according to the structure (Ouyang et al., 2009; Celikyilmaz and Hakkani-Tur, 2010), or gains coverage by drawing sentences from different parts of the hierarchy (Yang and Wang, 2003; Wang et al., 2006).

7 Conclusions

We have introduced a new paradigm for large-scale summarization called hierarchical summarization, which allows a user to navigate a hierarchy of relatively short summaries. We present SUMMA, an implemented hierarchical news summarization system,[7] and demonstrate its effectiveness in a user study that compares SUMMA with a timeline system and a flat MDS system. When compared to timelines, users learned more with SUMMA in twice as many cases, and SUMMA was preferred more than three times as often. When compared to flat summaries, users overwhelming preferred SUMMA and learned just as much.

This first implementation performs temporal clustering – in future work, we will investigate dynamically selecting an organizing principle that is best suited to the data at each level of the hierarchy: by entity, by location, by event, or by date. We also intend to scale the system to even larger document collections, and explore joint clustering and summarization. Lastly, we plan to research hierarchical summarization in other domains.

Acknowledgments

We thank Amitabha Bagchi, Niranjan Balasubramanian, Danish Contractor, Oren Etzioni, Tony Fader, Carlos Guestrin, Prachi Jain, Lucy Vanderwende, Luke Zettlemoyer, and the anonymous reviewers for their helpful suggestions and feedback. We thank Hui Lin and Jeff Bilmes for providing us with their code. This research was supported in part by ARO contract W911NF-13-1-0246, DARPA Air Force Research Laboratory (AFRL) contract FA8750-13-2-0019, UW-IITD subcontract RP02815, and the Yahoo! Faculty Research and Engagement Award. This paper is also supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via AFRL contract number FA8650-10-C-7058. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, AFRL, or the U.S. Government.

[7] http://knowitall.cs.washington.edu/summa/

References

C. G. Akcora, M. A. Bayir, M. Demirbas, and H. Ferhatosmanoglu. 2010. Identifying breakpoints in public opinion. In *1st KDD Workshop on Social Media Analytics*.

Berkhin Berkhin. 2006. A survey of clustering data mining techniques. *Grouping Multidimensional Data*, pages 25–71.

Orkut Buyukkokten, Hector Garcia-Molina, and Andreas Paepcke. 2001. Seeing the whole in parts: Text summarization for web browsing on handheld devices. In *Proceedings of WWW 2001*, pages 652–662.

Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of SIGIR 1998*, pages 335–336.

Asli Celikyilmaz and Dilek Hakkani-Tur. 2010. A hybrid hierarchical model for multi-document summarization. In *Proceedings of ACL 2010*, pages 815–824.

Angel Chang and Christopher Manning. 2012. SUTime: A library for recognizing and normalizing time expressions. In *Proceedings of LREC 2012*.

Hai Leong Chieu and Yoong Keok Lee. 2004. Query based event extraction along a timeline. In *Proceedings of SIGIR 2004*, pages 425–432.

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2013. Towards coherent multi-document summarization. In *Proceedings of NAACL 2013*.

Jennifer Gillenwater, Alex Kulesza, and Ben Taskar. 2012. Discovering diverse and salient threads in document collections. In *Proceedings of EMNLP-CoNLL 2012*, pages 710–720.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. *Proceedings of NAACL 2009*, pages 362–370.

Po Hu, Minlie Huang, Peng Xu, Weichang Li, Adam K. Usadi, and Xiaoyan Zhu. 2011. Generating breakpoint-based timeline overview for news topic retrospection. In *Proceedings of ICDM 2011*.

Remy Kessler, Xavier Tannier, Caroline Hagège, Véronique Moriceau, and André Bittar. 2012. Finding salient dates for building thematic timelines. In *Proceedings of ACL 2012*, pages 730–739.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430.

Jon Kleinberg. 2002. Bursty and hierarchical structure in streams. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 91–101.

Dawn Lawrie, W. Bruce Croft, and Arnold Rosenberg. 2001. Finding topic words for hierarchical summarization. In *Proceedings of SIGIR '01*, pages 349–357.

Dawn J. Lawrie. 2003. *Language models for hierarchical summarization*. Ph.D. thesis, University of Massachusetts Amherst.

Peng Li, Jing Jiang, and Yinglin Wang. 2010. Generating templates of entity summaries with an entity-aspect model and pattern mining. In *Proceedings of ACL 2010*, pages 640–649.

Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of ACL 2011*, pages 510–520.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of EMNLP 2012*, pages 523–534.

Jahna Otterbacher, Dragomir Radev, and Omer Kareem. 2006. News to go: Hierarchical text summarization for mobile devices. In *Proceedings of SIGIR 2006*, pages 589–596.

You Ouyang, Wenji Li, and Qin Lu. 2009. An integrated multi-document summarization approach based on word hierarchical representation. In *Proceedings of the ACLShort 2009*, pages 113–116.

Dragomir R. Radev, Hongyan Jing, Malgorzata Stys, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40(6):919–938.

Horacio Saggion and Robert Gaizauskas. 2004. Multi-document summarization by cluster/profile relevance and redundancy removal. In *Proceedings of DUC 2004*.

Christina Sauper and Regina Barzilay. 2009. Automatically generating Wikipedia articles: A structure-aware approach. In *Proceedings of ACL 2009*, pages 208–216.

Dafna Shahaf and Carlos Guestrin. 2010. Connecting the dots between news articles. In *Proceedings of KDD 2010*, pages 623–632.

Dafna Shahaf, Carlos Guestrin, and Eric Horvitz. 2012. Trains of thought: Generating information maps. In *Proceedings of WWW 2012*.

Russell Swan and James Allen. 2000. Automatic generation of overview timelines. In *Proceedings of SIGIR 2000*, pages 49–56.

Kou Takahashi, Takao Miura, and Isamu Shioya. 2007. Hierarchical summarizing and evaluating for web pages. In *Proceedings of the 1st workshop on emerging research opportunities for Web Data Management (EROW 2007)*.

Robert Tibshirani, Guenther Walther, and Trevor Hastie. 2000. Estimating the number of clusters in a dataset via the gap statistic. *Journal of the Royal Statistical Society, Series B*, 32(2):411–423.

Fu Lee Wang, Christopher C. Yang, and Xiaodong Shi. 2006. Multi-document summarization for terrorism information extraction. In *Proceedings of ISI'06*.

Rui Yan, Liang Kong, Congrui Huang, Xiaojun Wan, Xiaoming Li, and Yan Zhang. 2011a. Timeline generation through evolutionary trans-temporal summarization. In *Proceedings of EMNLP 2011*, pages 433–443.

Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. 2011b. Evolutionary timeline summarization: A balanced optimization framework via iterative substitution. In *Proceeding of SIGIR 2011*, pages 745–754.

Christopher C. Yang and Fu Lee Wang. 2003. Fractal summarization: summarization based on fractal theory. In *Proceedings of SIGIR 2003*, pages 391–392.

Conglei Yao, Xu Jia, Sicong Shou, Shicong Feng, Feng Zhou, and Hongyan Liu. 2011. Autopedia: Automatic domain-independent wikipedia article generation. In *Proceedings of WWW 2011*, pages 161–162.