Sentence Ordering in Multidocument Summarization

Regina Barzilay
Computer Science Department
1214 Amsterdam Ave
New York, 10027, NY, USA
regina@cs.columbia.edu

Noemie Elhadad
Computer Science Department
1214 Amsterdam Ave
New York, 10027, NY, USA
noemie@cs.columbia.edu

Kathleen R. McKeown
Computer Science Department
1214 Amsterdam Ave
New York, 10027, NY, USA
kathy@cs.columbia.edu

ABSTRACT

The problem of organizing information for multidocument summarization so that the generated summary is coherent has received relatively little attention. In this paper, we describe two naive ordering techniques and show that they do not perform well. We present an integrated strategy for ordering information, combining constraints from chronological order of events and cohesion. This strategy was derived from empirical observations based on experiments asking humans to order information. Evaluation of our augmented algorithm shows a significant improvement of the ordering over the two naive techniques we used as baseline.

1. INTRODUCTION

Multidocument summarization poses a number of new challenges over single document summarization. Researchers have already investigated issues such as identifying repetitions or contradictions across input documents and determining which information is salient enough to include in the summary [1, 3, 6, 11, 15, 19]. One issue that has received little attention is how to organize the selected information so that the output summary is coherent. Once all the relevant pieces of information have been selected across the input documents, the summarizer has to decide in which order to present them so that the whole text makes sense. In single document summarization, one possible ordering of the extracted information is provided by the input document itself. However, [10] observed that, in single document summaries written by professional summarizers, extracted sentences do not retain their precedence orders in the summary. Moreover, in the case of multiple input documents, this does not provide a useful solution: information may be drawn from different documents and therefore, no one document can provide an ordering. Furthermore, the order between two pieces of information can change significantly from one document to another.

We investigate constraints on ordering in the context of multidocument summarization. We first describe two naive ordering algorithms, used in several systems and show that they do not yield satisfactory results. The first, Majority Ordering, is critically linked to the level of similarity of the information ordering across the input texts. But many times input texts have different structure, and therefore, this algorithm is not acceptable. The second, Chronological Ordering, can produce good results when the information is event-based and can, therefore, be ordered based on temporal occurence. However, texts do not always refer to events. We have conducted experiments to identify additional constraints using a manually built collection of multiple orderings of texts. These experiments show that cohesion as an important constraint. While it is recognized in the generation community that cohesion is a necessary feature for a generated text, we provide an operational way to automatically ensure cohesion when ordering sentences in an output summary. We augment the Chronological Ordering algorithm with a cohesion constraint, and compare it to the naive algorithms.

Our framework is the MultiGen system [15], a domain independent multidocument summarizer which has been trained and tested on news articles. In the following sections, we first give an overview of MultiGen. We then describe the two naive ordering algorithms and evaluate them. We follow this with a study of multiple orderings produced by humans. This allows us to determine how to improve the Chronological Ordering algorithm using cohesion as an additional constraint. The last section describes the augmented algorithm along with its evaluation.

2. MULTIGEN OVERVIEW

MultiGen operates on a set of news articles describing the same event. It creates a summary which synthesizes common information across documents. In the case of multidocument summarization of articles about the same event, source articles can contain both repetitions and contradictions. Extracting all the similar sentences would produce a verbose and repetitive summary, while extracting only some of the similar sentences would produce a summary biased towards some sources. MultiGen uses a comparison of extracted similar sentences to select the appropriate phrases to include in the summary and reformulates them as a new text.

MultiGen consists of an analysis and a generation component. The analysis component [7] identifies units of text which convey similar information across the input documents using statistical techniques and shallow text analysis. Once similar text units are identified, we cluster them

into *themes*. Themes are sets of sentences from different documents that contain repeated information and do not necessarily contain sentences from all the documents. For each theme, the generation component [1] identifies phrases which are in the intersection of the theme sentences, and selects them as part of the summary. The intersection sentences are then ordered to produce a coherent text.

3. NAIVE ORDERING ALGORITHMS ARE NOT SUFFICIENT

When producing a summary, any multidocument summarization system has to choose in which order to present the output sentences. In this section, we describe two algorithms for ordering sentences suitable for domain independent multidocument summarization. The first algorithm, Majority Ordering (MO), relies only on the original orders of sentences in the input documents. It is the first solution one can think of when addressing the ordering problem. The second one, Chronological Ordering (CO) uses time related features to order sentences. We analyze this strategy because it was originally implemented in MultiGen and followed by other summarization systems [18]. In the MultiGen framework, ordering sentences is equivalent to ordering themes and we describe the algorithms in terms of themes, but the concepts can be adapted to other summarization systems such as [3]. Our evaluation shows that these methods alone do not provide an adequate strategy for ordering.

3.1 Majority Ordering

3.1.1 The Algorithm

Typically, in single document summarization, the order of sentences in the output summary is determined by their order in the input text. This strategy can be adapted to multidocument summarization. Consider two themes, Th_1 and Th_2; if sentences from Th_1 preceed sentences from Th_2 in all input texts, then presenting Th_1 before Th_2 is an acceptable order. But, when the order between sentences from Th_1 and Th_2 varies from one text to another, this strategy is not valid anymore. One way to define the order between Th_1 and Th_2 is to adopt the order occuring in the majority of the texts where Th_1 and Th_2 occur. This strategy defines a pairwise order between themes. However, this pairwise relation is not transitive; for example, given the themes Th_1 and Th_2 occuring in a text, Th_2 and Th_3 occuring in another text, and Th_3 and Th_1 occuring in a third text, there is a conflict between the orders (Th_1, Th_2, Th_3) and (Th_3, Th_1). Since transitivity is a necessary condition for a relation to be called an order, this relation does not form a global order.

We, therefore, have to expand this pairwise relation to a global order. In other words, we have to find a linear order between themes which maximizes the agreement between the orderings imposed by the input texts. For each pair of themes, Th_i and Th_j, we keep two counts, $C_{i,j}$ and $C_{j,i}$ — $C_{i,j}$ is the number of input texts in which sentences from Th_i occur before sentences from Th_j and $C_{j,i}$ is the same for the opposite order. The weight of a linear order $(Th_{i_1}, \ldots, Th_{i_k})$ is defined as the sum of the counts for every pair C_{i_l,i_m}, such that $i_l \leq i_m$ and $l, m \in \{1 \ldots k\}$. Stating this problem in terms of a directed graph where nodes are themes, and a vertex from Th_i to Th_j has for weight $C_{i,j}$, we are looking for a path with maximal weight which traverses each node exactly once. Unfortunately this problem is NP-complete; this can be shown by reducing the *traveling salesman problem* to this problem. Despite this fact, we still can apply this ordering, because typically the length of the output summary is limited to a small number of sentences. For longer summaries, the approximation algorithm described in [4] can be applied. Figures 1 and 2 show examples of produced summaries.

The main problem with this strategy is that it can produce several orderings with the same weight. This happens when there is a tie between two opposite orderings. In this situation, this strategy does not provide enough constraints to determine one optimal ordering; one order is chosen randomly among the orders with maximal weight.

The man accused of firebombing two Manhattan subways in 1994 was convicted Thursday after the jury rejected the notion that the drug Prozac led him to commit the crimes.
He was found guilty of two counts of attempted murder, 14 counts of first-degree assault and two counts of criminal possession of a weapon.
In December 1994, Leary ignited firebombs on two Manhattan subway trains. The second blast injured 50 people – 16 seriously, including Leary.
Leary wanted to extort money from the Transit Authority.
The defense argued that Leary was not responsible for his actions because of "toxic psychosis" caused by the Prozac.

Figure 1: A summary produced using the Majority Ordering algorithm, graded as Good.

A man armed with a handgun has surrendered to Spanish authorities, peacefully ending a hijacking of a Moroccan jet.
Officials in Spain say a person commandeered the plane.
After the plane was directed to Spain, the hijacker said he wanted to be taken to Germany.
After several hours of negotiations, authorities convinced the person to surrender early today.
Police said the man had a pistol, but a Moroccan security source in Rabat said the gun was likely a "toy".
There were no reported injuries.
Officials in Spain say the Boeing 737 left Casablanca, Morocco, Wednesday night with 83 passengers and a nine- person crew headed for Tunis, Tunisia.
Spanish authorities directed the plane to an isolated section of El Prat Airport and officials began negotiations.

Figure 2: A summary produced using the Majority Ordering algorithm, graded as Poor.

3.1.2 Evaluation

We asked three human judges to evaluate the order of information in 20 summaries produced using the MO algorithm into three categories— Poor, Fair and Good. We define a Poor summary, in an operational way, as a text whose readability would be significantly improved by reordering its sentences. A Fair summary is a text which makes sense but reordering of some sentences can yield a better readability. Finally, a summary which cannot be further improved by any sentence reordering is considered a Good summary.

The judges were asked to grade the summaries taking only into account the order in which the information is presented. To help them focus on this aspect of the texts, we resolved dangling references beforehand. Figure 8 shows the grades assigned to the summaries using majority to combine the

judges grades. In our experiments, judges had strong agreement; they never gave three different grades to a summary.

The MO algorithm produces a small number of Good summaries, but most of the summaries were graded as Fair. For instance, the summary graded Good shown in Figure 1 orders the information in a natural way; the text starts with a sentence summary of the event, then the outcome of the trial is given, a reminder of the facts that caused the trial and a possible explanation of the facts. Looking at the Good summaries produced by MO, we found that it performs well when the input articles follow the same order when presenting the information. In other words, the algorithm produces a good ordering if the input articles orderings have high agreement.

On the other hand, when analyzing Poor summaries, as in Figure 2, we observe that the input texts have very different orderings. By trying to maximize the agreement of the input texts orderings, MO produces a new ordering that doesn't occur in any input text. The ordering is, therefore, not guaranteed anymore to be acceptable. An example of a new produced ordering is given in Figure 2. The summary would be more readable if several sentences were moved around (the last sentence would be better placed before the fourth sentence because they both talk about the Spanish authorities handling the hijacking).

This algorithm can be used to order sentences accurately if we are certain that the input texts follow similar organizations. This assumption may hold in limited domains. However, in our case, the input texts we are processing do not have such regularities. MO's performance critically depends on the quality of the input texts, therefore, we should design an ordering strategy which better fits our input data. From here on, we will focus only on the Chronological Ordering algorithm and ways to improve it.

3.2 Chronological Ordering

3.2.1 The Algorithm

Multidocument summarization of news typically deals with articles published on different dates, and articles themselves cover events occurring over a wide range in time. Using chronological order in the summary to describe the main events helps the user understand what has happened. It seems like a natural and appropriate strategy. As mentioned earlier, in our framework, we are ordering themes; in this strategy, we therefore need to assign a date to themes. To identify the date an event occured requires a detailed interpretation of temporal references in articles. While there have been recent developments in disambiguating temporal expressions and event ordering [12], correlating events with the date on which they occurred is a hard task. In our case, we approximate the theme time by its first publication date; that is, the first time the theme has been reported in our set of input articles. It is an acceptable approximation for news events; the first publication date of an event usually corresponds to its occurrence in real life. For instance, in a terrorist attack story, the theme conveying the attack itself will have a date previous to the date of the theme describing a trial following the attack.

Articles released by news agencies are marked with a publication date, consisting of a date and a time with three fields (hour, minutes and seconds). Articles from the same news agency are, then, guaranteed to have different publication dates. This also holds for articles coming from different news agencies. We never encountered two articles with the same publication date during the development of MultiGen. Thus, the publication date serves as a unique identifier over articles. As a result, when two themes have the same publication date, it means that they both are reported for the first time in the same article.

Our Chronological Ordering (CO) algorithm takes as input a set of themes and orders them chronologically whenever possible. Each theme is assigned a date corresponding to its first publication. This establishes a partial order over the themes. When two themes have the same date (that is, they are reported for the first time in the same article) we sort them according to their order of presentation in this article. We have now a complete order over the input themes.

To implement this algorithm in MultiGen, we select for each theme the sentence that has the earliest publication date. We call it the time stamp sentence and assign its publication date as the time stamp of the theme. Figures 3 and 4 show examples of produced summaries using CO.

> One of four people accused along with former Pakistani Prime Minister Nawaz Sharif has agreed to testify against him in a case involving possible hijacking and kidnapping charges, a prosecutor said Wednesday.
> Raja Quereshi, the attorney general, said that the former Civil Aviation Authority chairman has already given a statement to police.
> Sharif's lawyer dismissed the news when speaking to reporters after Sharif made an appearance before a judicial magistrate to hear witnesses give statements against him. Sharif has said he is innocent.
> The allegations stem from an alleged attempt to divert a plane bringing army chief General Pervez Musharraf to Karachi from Sri Lanka on October 12.

Figure 3: A summary produced using the Chronological Ordering algorithm graded as Good.

> Thousands of people have attended a ceremony in Nairobi commemorating the first anniversary of the deadly bombings attacks against U.S. Embassies in Kenya and Tanzania.
> Saudi dissident Osama bin Laden, accused of masterminding the attacks, and nine others are still at large.
> President Clinton said, "The intended victims of this vicious crime stood for everything that is right about our country and the world".
> U.S. federal prosecutors have charged 17 people in the bombings.
> Albright said that the mourning continues.
> Kenyans are observing a national day of mourning in honor of the 215 people who died there.

Figure 4: A summary produced using the Chronological Ordering algorithm graded as Poor.

3.2.2 Evaluation

Following the same methodology we used for the MO algorithm evaluation, we asked three human judges to grade 20 summaries generated by the system using the CO algorithm applied to the same collection of input texts. The results are shown in Figure 8.

Our first suspicion was that our approximation deviates too much from the real chronological order of events, and,

therefore, lowers the quality of sentence ordering. To verify this hypothesis, we identified sentences that broke the original chronological order and restored the ordering manually. Interestingly, the displaced sentences were mainly background information. The evaluation of the modified summaries shows a slight but not visible improvement.

When comparing Good (Figure 3) and Poor (Figure 4) summaries, we notice two phenomena: first, many of the badly placed sentences cannot be ordered based on their temporal occurence. For instance, in Figure 4, the sentence quoting Clinton is not one event in the sequence of events being described, but rather a reaction to the main events. This is also true for the sentence reporting Albright's reaction. Assigning a date to a reaction, or more generally to any sentence conveying background information, and placing it into the chronological stream of the main events does not produce a logical ordering. The ordering of these themes is therefore not covered by the CO algorithm.

The second phenomenon we observed is that Poor summaries typically contain abrupt switches of topics and general incoherences. For instance, in Figure 4, quotes from US officials (third and fifth sentences) are split and sentences about the mourning (first and sixth sentences) appear too far apart in the summary. Grouping them together would increase the readability of the summary. At this point, we need to find additional constraints to improve the ordering.

4. IMPROVING THE ORDERING: EXPERIMENTS AND ANALYSIS

In the previous section, we showed that using naive ordering algorithms does not produce satisfactory orderings. In this section, we investigate through experiments with humans, how to identify patterns of orderings that can improve the algorithm.

Sentences in a text can be ordered in a number of ways, and the text as a whole will still convey the same meaning. But undoubtedly, some orders are definitely unacceptable because they break conventions of information presentation. One way to identify these conventions is to find commonalities between different acceptable orderings of the same information. Extracting regularities in several acceptable orderings can help us specify the main ordering constraints for a given input type. Since a collection of multiple summaries over the same set of articles doesn't exist, we created our own collection of multiple orderings produced by different humans. Using this collection, we studied common behaviors and mapped them to strategies for ordering.

Our collection of multiple orderings is available at `http://www.cs.columbia.edu/~noemie/ordering/`. It was built in the following way. We collected ten sets of articles. Each set consisted of two to three news articles reporting the same event. For each set, we manually selected the intersection sentences, simulating MultiGen[1]. On average, each set contained 8.8 intersection sentences. The sentences were cleaned of explicit references (for instance, occurrences of "the President" were resolved to "President Clinton") and connectives, so that participants wouldn't use them as clues for ordering. Ten subjects participated in the experiment and they each built one ordering per set of intersection sentences. Each subject was asked to order the intersection sentences of a set so that they form a readable text. Overall, we obtained 100 orderings, ten alternative orderings per set. Figure 5 shows the ten alternative orderings collected for one set.

[1]We performed a manual simulation to ensure that ideal data was provided to the subjects of the experiments

We first observe that a surprising majority of orderings are different. Out of the ten sets, only two sets had some identical orderings (in one set, one pair of orderings were identical while in the other set, two pairs of orderings were identical). In other words, there are many acceptable orderings given one set of sentences. This confirms the intuition that we do not need to look for a single ideal global ordering but rather construct an acceptable one.

We also notice that, within the multiple orderings of a set, some sentences always appear together. They do not appear in the same order from one ordering to another, but they share an adjacency relation. From now on, we refer to them as blocks. For each set, we identify blocks by clustering sentences. We use as a distance metric between two sentences the average number of sentences that separate them over all orderings. In Figure 5, for instance, the distance between the sentences D and G is 2. The blocks identified by clustering are: sentences B, D, G and I; sentences A and J; sentences C and F; and sentences E and H.

Participant 1	D B G I H F C J A E
Participant 2	D G B I C F A J E H
Participant 3	D B I G F J A E H C
Participant 4	D C F G I B J A H E
Participant 5	D G B I H F J A C E
Participant 6	D G I B F C E H J A
Participant 7	D B G I F C H E J A
Participant 8	D B C F G I E H A J
Participant 9	D G I B E H F A J C
Participant 10	D B G I C F A J E H

Figure 5: Multiple orderings for one set in our collection.

We observed that all the blocks in the experiment correspond to clusters of topically related sentences. These blocks form units of text dealing with the same subject, and exhibit cohesive properties. For ordering, we can use this to opportunistically group sentences together that all refer to the same topic.

Collecting a set of multiple orderings is an expensive task; it is difficult and time consuming for a human to order sentences from scratch. Furthermore, to discover significant commonalities across orderings, many multiple orderings of the same set are necessary. We plan to extend our collection and we are confident that it will provide more insights on ordering. Still, the existing collection enables us to identify cohesion as an important factor for ordering. We describe next how we integrate the cohesion constraint in the CO algorithm.

5. THE AUGMENTED ALGORITHM

In the output of the CO algorithm, disfluencies arise when topics are distributed over the whole text, violating cohesion properties [13]. A typical scenario is illustrated in Figure 6. The inputs are texts T_1, T_2, T_3 (in order of publication). A_1, A_2 and A_3 belong to the same theme whose intersection sentence is A and similarly for B and C. The themes A and

B are topically related, but C is not related. Summary S_1, based only on chronological clues, contains two topical shifts; from A to C and back from C to B. A better summary would be S_2 which keeps A and B together.

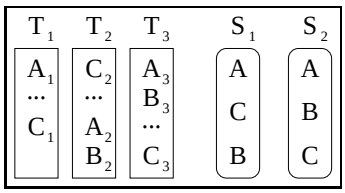

Figure 6: Input texts $T_1T_2T_3$ are summarized by the Chronological Ordering (S_1) or by the Augmented algorithm (S_2).

5.1 The Algorithm

Our goal is to remove disfluencies from the summary by grouping together topically related themes. This can be achieved by integrating cohesion as an additional constraint to the CO algorithm. The main technical difficulty in incorporating cohesion in our ordering algorithm is to identify and to group topically related themes across multiple documents. In other words, given two themes, we need to determine if they belong to the same cohesion block. For a single document, segmentation [8] could be used to identify blocks, but we cannot use such a technique to identify cohesion between sentences across multiple documents. The main reason is that segmentation algorithms exploit the linear structure of an input text; in our case, we want to group together sentences belonging to different texts.

Our solution consists of the following steps. In a preprocessing stage, we segment each input text, so that given two sentences within the same text, we can determine if they are topically related. Assume the themes A and B, where A contains sentences $(A_1 \dots A_n)$, and B contains sentences $(B_1 \dots B_m)$. Recall that a theme is a set of sentences conveying similar information drawn from different input texts. We denote $\#AB$ to be the number of pairs of sentences (A_i, B_j) which appear in the same text, and $\#AB^+$ to be the number of sentence pairs which appear in the same text and are in the same segment.

In a first stage, for each pair of themes A and B, we compute the ratio $\#AB^+/\#AB$ to measure the relatedness of two themes. This measure takes into account both positive and negative evidence. If most of the sentences in A and B that appear together in the same texts are also in the same segments, it means that A and B are highly topically related. In this case, the ratio is close to 1. On the other hand, if among the texts containing sentences from A and B, only a few pairs are in the same segments, then A and B are not topically related. Accordingly the ratio is close to 0. A and B are considered related if this ratio is higher than a predetermined threshold. In our experiments, we set it to 0.6.

This strategy defines pairwise relations between themes. A transitive closure of this relation builds groups of related themes and as a result ensures that themes that do not appear together in any article but are both related to a third theme will still be linked. This creates an even higher degree of relatedness among themes. Because we use a threshold to establish pairwise relations, the transitive closure does not produce elongated chains that could link together unrelated themes. We are now able to identify topically related themes. At the end of the first stage, they are grouped into blocks.

In a second stage, we assign a time stamp to each block of related themes, as the earliest time stamp of the themes it contains. We adapt the CO algorithm described in *3.2.1* to work at the level of the blocks. The blocks and the themes correspond to, respectively, themes and sentences in the CO algorithm. By analogy, we can easily show that the adapted algorithm produces a complete order of the blocks. This yields a macro-ordering of the summary. We still need to order the themes inside each block.

In the last stage of the augmented algorithm, for each block, we order the themes it contains by applying the CO algorithm to them. Figure 7 shows an example of a summary produced by the augmented algorithm.

This algorithm ensures that cohesively related themes will not be spread over the text, and decreases the number of abrupt switches of topics. Figure 7 shows how the Augmented algorithm improves the sentence order compared with the order in the summary produced by the CO algorithm in Figure 4; sentences quoting US officials are now grouped together and so are descriptions of the mourning.

Thousands of people have attended a ceremony in Nairobi commemorating the first anniversary of the deadly bombings attacks against U.S. Embassies in Kenya and Tanzania. Kenyans are observing a national day of mourning in honor of the 215 people who died there.

Saudi dissident Osama bin Laden, accused of masterminding the attacks, and nine others are still at large. U.S. federal prosecutors have charged 17 people in the bombings.

President Clinton said, "The intended victims of this vicious crime stood for everything that is right about our country and the world". Albright said that the mourning continues.

Figure 7: A Summary produced using the Augmented algorithm. Related sentences are grouped into paragraphs.

5.2 Evaluation

Following the same methodology used to evaluate the MO and the CO algorithms, we asked the judges to grade 20 summaries produced by the Augmented algorithm. Results are shown in Figure 8.

The manual effort needed to compare and judge system output is extensive; consider that each human judge had to read three summaries for each input set as well as skim the input texts to verify that no misleading order was introduced in the summaries. Consequently, the evaluation that we performed to date is limited. Still, this evaluation shows a significant improvement in the quality of the orderings from the CO algorithm to the augmented algorithm. To assess the significance of the improvement, we used the Fisher exact test, conflating Poor and Fair summaries into one category. This test is adapted to our case because of the reduced size of our test set. We obtained a p value of 0.014 [20].

6. RELATED WORK

Finding an acceptable ordering has not been studied before in summarization. In single document summarization,

	Poor	Fair	Good
Majority Ordering	2	12	6
Chronological Ordering	7	7	6
Augmented Ordering	2	7	11

Figure 8: Evaluation of the the Majority Ordering, the Chronological Ordering and the Augmented Ordering.

summary sentences are typically arranged in the same order that they were found in the full document (although [10] reports that human summarizers do sometimes change the original order). In multidocument summarization, the summary consists of fragments of text or sentences that were selected from different texts. Thus, there is no complete ordering of summary sentences that can be found in the original documents.

The ordering task has been extensively investigated in the generation community [14, 17, 9, 2, 16]. One approach is top-down, using schemas [14] or plans [5] to determine the organizational structure of the text. This appproach postulates a rhetorical structure which can be used to select information from an underlying knowledge base. Because the domain is limited, an encoding can be developed of the kinds of propositional content that match rhetorical elements of the schema or plan, thereby allowing content to be selected and ordered. Rhetorical Structure Theory (RST) allows for more flexibility in ordering content. The relations occur between pairs of propositions. Constraints based on intention (e.g., [17]), plan-like conventions [9], or stylistic constraints [2] are used as preconditions on the plan operators containing RST relations to determine when a relation is used and how it is ordered with respect to other relations.

MultiGen generates summaries of news on any topic. In an unconstrained domain like this, it would be impossible to enumerate the semantics for all possible types of sentences which could match the elements of a schema, a plan or rhetorical relations. Furthermore, it would be difficult to specify a generic rhetorical plan for a summary of news. Instead, content determination in MultiGen is opportunistic, depending on the kinds of similarities that happen to exist between a set of news documents. Similarly, we describe here an ordering scheme that is opportunistic and bottom-up, depending on the coherence and temporal connections that happen to exist between selected text. Our approach is similar to the use of basic blocks [16] where a bottom-up technique is used to group together stretches of text in a long, generated document by finding propositions that are related by a common focus. Since this approach was developed for a generation system, it finds related propositions by comparisons of proposition arguments at the semantic level. In our case, we are dealing with a surface representation, so we find alternative methods for grouping text fragments.

7. CONCLUSION AND FUTURE WORK

In this paper we investigated information ordering constraints in multidocument summarization. We analyzed two naive ordering algorithms, the Majority Ordering (MO) and the Chronological Ordering (CO). We show that the MO algorithm performs well only when all input texts follow similar presentation of the information. The CO algorithm can provide an acceptable solution for many cases, but is not sufficient when summaries contain information that is not event based. We report on the experiments we conducted to identify other constraints contributing to ordering. We show that cohesion is an important factor, and describe an operational way to incorporate it in the CO algorithm. This results in a definite improvement of the overall quality of automatically generated summaries.

In future work, we first plan to extend our collection of multiple orderings, so that we can extract more regularities and understand better how human order information to produce a readable and fluent text. Even though we did not encounter any misleading inferences introduced by reordering MultiGen output, we plan to do an extended study of the side effects caused by reorderings. We also plan to investigate whether the MO algorithm can be improved by applying it on cohesive blocks of themes, rather than themes.

8. ACKNOWLEDGMENT

This work was partially supported by DARPA grant N66001-00-1-8919, a Louis Morin scholarship and a Viros scholarship. We thank Eli Barzilay for providing help with the experiments interface, Michael Elhadad for the useful discussions and comments, and all the voluntary participants in the experiments.

9. REFERENCES

[1] R. Barzilay, K. McKeown, and M. Elhadad. Information fusion in the context of multi-document summarization. In *Proc. of the 37th Annual Meeting of the Assoc. of Computational Linguistics*, 1999.

[2] N. Bouayad-Agha, R. Power, and D. Scott. Can text structure be incompatible with rhetorical structure? In *Proceedings of the First International Conference on Natural Language Generation (INLG'2000)*, Mitzpe Ramon, Israel, 2000.

[3] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998.

[4] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. The MIT Press, 1990.

[5] R. Dale. *Generating Referring Expressions: Constructing Descriptions in a Domain of Objects and Processes*. MIT Press, Cambridge, MA, 1992.

[6] N. Elhadad and K. McKeown. Generating patient specific summaries of medical articles. Submitted, 2001.

[7] V. Hatzivassiloglou, J. Klavans, and E. Eskin. Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

[8] M. Hearst. Multi-paragraph segmentation of expository text. In *Proceedings of the 32th Annual Meeting of the Association for Computational Linguistics*, 1994.

[9] E. Hovy. Automated discourse generation using discourse structure relations. *Artificial Intelligence*, 63, 1993. Special Issue on NLP.

[10] H. Jing. Summary generation through intelligent cutting and pasting of the input document. Technical report, Columbia University, 1998.

[11] I. Mani and E. Bloedorn. Multi-document summarization by graph search and matching. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 1997.

[12] I. Mani and G. Wilson. Robust temporal processing of news. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, 2000.

[13] K. McCoy and J. Cheng. Focus of attention: Constraining what can be said next. In C. Paris, W. Swartout, and W. Mann, editors, *Natural Language Generation in Artificial Intelligence and Computational Linguistics*. Kluwer Academic Publishers, 1991.

[14] K. McKeown. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press, England, 1985.

[15] K. McKeown, J. Klavans, V. Hatzivassiloglou, R. Barzilay, and E. Eskin. Towards multidocument summarization by reformulatin: Progress and prospects. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, 1999.

[16] D. Mooney, S. Carberry, and K. McCoy. The generation of high-level structure for extended explanations. In *Proceedings of the International Conference on Computational Linguistics (COLING–90)*, pages 276–281, Helsinki, 1990.

[17] J. Moore and C. Paris. Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Journal of Computational Linguistics*, 19(4), 1993.

[18] D. Radev, H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the ANLP/NAACL 2000 Workshop on Automatic Summarization*, 2000.

[19] D. Radev and K. McKeown. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, **24**(3):469–500, September 1998.

[20] S. Siegal and N. J. Castellan. *Non-Parametric statistics for the behavioural sciences*. McGraw Hill, 1988.