A Discourse-Aware Graph-Based Content-Selection Framework

Seniz Demir Sandra Carberry Kathleen F. McCoy
Department of Computer Science
University of Delaware
Newark, DE 19716
{demir,carberry,mccoy}@cis.udel.edu

Abstract

This paper presents an easy-to-adapt, discourse-aware framework that can be utilized as the content selection component of a generation system whose goal is to deliver descriptive texts in several turns. Our framework involves a novel use of a graph-based ranking algorithm, to iteratively determine what content to convey to a given request while taking into account various considerations such as capturing a priori importance of information, conveying related information, avoiding redundancy, and incorporating the effects of discourse history. We illustrate and evaluate this framework in an accessibility system for sight-impaired individuals.

1 Introduction

Content selection is the task responsible for determining what to convey in the output of a generation system at the current exchange (Reiter and Dale, 1997). This very domain dependent task is extremely important from the perspective of users (Sripada et al., 2001) who have been observed to be tolerant of realization problems as long as the appropriate content is expressed. The NLG community has proposed various content selection approaches since early systems (Moore and Paris, 1993; McKeown, 1985) which placed emphasis on text structure and adapted planning techniques or schemas to meet discourse goals.

This paper proposes a domain-independent framework which can be incorporated as a content selection component in a system whose goal is to deliver descriptive or explanatory texts, such as the ILEX (O'Donnell et al., 2001), KNIGHT (Lester and Porter, 1997), and POLIBOX (Chiarcos and Stede, 2004) systems. At the core of our framework lies a novel use of a graph-based ranking algorithm, which exploits discourse related considerations in determining what content to convey in response to a request for information. This framework provides the ability to generate successive history-aware texts and the flexibility to generate different texts with different parameter settings.

One discourse consideration is the tenet that the propositions selected for inclusion in a text should be in some way related to one another. Thus, the selection process should be influenced by the relevance of information to what has already been selected for inclusion. Moreover, we argue that if the information given in a proposition can be deduced from the information provided by any other proposition in the text, this would introduce redundancy and should be avoided.

Many systems (such as MATCH (Walker et al., 2004) and GEA (Carenini and Moore, 2006)) contain a user model which is employed to adapt content selection to the user's preferences (Reiter and Dale, 1997). Our framework provides a facility to model a stereotypical user by incorporating the a priori importance of propositions. This facility can also be used to capture the preferences of a particular user.

In a dialogue system, utterances that are generated without exploiting the previous discourse seem awkward and unnatural (Moore, 1993). Our framework takes the previous discourse into account so as to omit recently communicated propositions and to determine when repetition of a previously communicated proposition is appropriate.

To our knowledge, our work is the first effort utilizing a graph-based ranking algorithm for content selection, while taking into account what information preferably should and shouldn't be conveyed together, the a priori importance of information, and the discourse history. Our framework is a domain-independent methodology containing domain-dependent features that must be instantiated when applying the methodology to a domain.

Section 2 describes our domain-independent methodology for determining the content of a response. Section 3 illustrates its application in an accessibility system for sight-impaired individuals and shows the generation flexibility provided by this framework. Finally, Section 4 discusses the results of user studies conducted to evaluate the effectiveness of our methodology.

2 A Graph-based Content Selection Framework

Our domain-independent framework can be applied to any domain where there is a set of propositions that *might* be conveyed and where a bottom-up strategy for content selection is appropriate. It is particularly useful when the set of propositions should be delivered a little at a time. For example, the ILEX system (O'Donnell et al., 2001) uses multiple descriptions to convey the available information about a museum artifact, since the length of the text that can be displayed on a page is limited. In order to use our framework, an application developer should identify the set of propositions that might be conveyed in the domain, specify the relations between these propositions, and optionally assess a priori importance of the propositions.

Our framework uses a weighted undirected graph **(relation_graph)**, where the propositions are captured as vertices of the graph and the edges represent relations between these propositions. While the number and kinds of relations represented is up to the developer, the framework does require the use of one specific relation (**Redundancy_Relation**) that is generalizable to any descriptive domain. Redundancy_Relation must be specified between two propositions if they provide similar kinds of information or the information provided by one of the propositions can be deduced from the information provided by the other. For example, consider applying the framework to the ILEX domain. Since the proposition that "this jewelry is produced by a single craftsman" can be deduced from the proposition that "this jewelry is made by a British designer", these propositions should be connected with a Redundancy_Relation in the relation_graph.

There is at most one edge between any two vertices and the weight of that edge represents how important it is to convey the corresponding propositions in the same text (which we refer to as the strength of the relation between these propositions). For example, suppose that once a museum artifact is introduced in ILEX, it is more important to convey its design style in the same description as opposed to where it is produced. In this case, the weight of the edge between the propositions introducing the artifact and its style should be higher than the weight of the edge between the propositions introducing the artifact and its production place.

The framework incorporates a stereotypical user model via an additional vertex (**priority_vertex**) in the relation_graph. The priority_vertex is connected to all other vertices in the graph. The weight of the edge between a vertex and the priority_vertex represents the a priori importance of that vertex, which in turn specifies the importance of the corresponding proposition. For example, suppose that in the ILEX domain an artifact has two features that are connected to the proposition introducing the artifact by the "feature-of" relation. The a priori importance of one of these features over the other can be specified by giving a higher weight to the edge connecting this proposition to the priority_vertex than is given to the edge between the other feature and the priority_vertex. This captures a priori importance and makes it more likely that the important feature will be included in the artifact's description.

2.1 Our Ranking Algorithm

With this graph-based setting, the most important thing to say is the proposition which is most central. Several centrality algorithms have been proposed in the literature (Freeman, 1979; Navigli and Lapata, 2007) for calculating the importance scores of vertices in a graph. The well-known PageRank centrality (Brin and Page, 1998) calculates the importance of a vertex by taking into account the importance of all other vertices and the relation of vertices to one another. This metric has been applied to various tasks such as word sense disambiguation (Sinha and Mihalcea, 2007) and text summarization (Erkan and Radev, 2004). We adopted the weighted PageRank metric (Sinha and Mihalcea, 2007) for our framework and therefore compute the importance score of a vertex (V_x) as:

$$PR(V_x) = (1-d) + d * \sum_{(V_x,V_y)\in E} \frac{w_{yx}}{\sum\limits_{(V_z,V_y)\in E} w_{yz}} PR(V_y)$$

where w_{xy} is the weight associated with the edge between vertices (V_x) and (V_y), E is the set of all edges, and d is the damping factor, set to 0.85, which is its usual setting.

Once the propositions in a domain are captured in a relation_graph with weights assigned to the edges between them, the straightforward way of identifying the propositions to be conveyed in the generated text would be to calculate the importance of each vertex via the formula above and then select the k vertices with the highest scores. However, this straightforward application would fail to address the discourse issues cited earlier. Thus we select propositions incrementally, where with each proposition selected, weights in the graph are adjusted causing related propositions to be highlighted and redundant information to be repelled. Because our responses are delivered over several turns, we also adjust weights between responses to reflect that discourse situation.

Our algorithm, shown in Figure 1, is run each time a response text is to be generated. For each new response, the algorithm begins by adjusting the importance of the priority_vertex (making it high) and clearing the list of selected propositions. Step **2** is the heart of the algorithm for generating a single response. It incrementally selects propositions to include in the current response, and adjusts weights to reflect what has been selected. In particular, in order to select a proposition, importance scores are computed using the weighted PageRank metric for all vertices corresponding to propositions that have not yet been selected for inclusion in this response (Step **2-a**), and only the proposition that receives the highest score is selected (Step **2-b**). Then, adjustments are made to achieve four goals toward taking discourse information into account (Steps **2-c** thru **2-g**) before the PageRank algorithm is run again to select the next proposition. Steps **3** and **4** adjust weights to reflect the completed response and to prepare for generating the next response.

Our first goal is to reflect the a priori importance of propositions in the selection process. For this purpose, we always assign the highest (or one of the highest) importance scores to the priority_vertex among the other vertices (Steps **1** and **2-g**). This will make the priority_vertex as influential as any other neighbor of a vertex when calculating its importance.

Our second goal is to select propositions that are relevant to previously selected propositions, or in terms of the graph-based notation, to **attract** the selection of vertices that are connected to the selected vertices. To achieve this, we increase the importance of the vertices corresponding to selected propositions so that the propositions related to them have a higher probability of being chosen as the next proposition to include (Step **2-g**).

Our third goal is to avoid selecting propositions that preferably shouldn't be communicated with previously selected propositions if other related propositions are available. To accomplish this, we introduce the term **repellers** to refer to the kinds of relations between propositions that are dispreferred over other relations once one of the propositions is selected for inclusion. Once a proposition is selected, we penalize the weights on the edges between the corresponding vertex and other vertices that are connected by a repeller (Step **2-d**). We don't provide any general repellers in the framework, but rather this is left for the developer familiar with the domain; any number (zero or more) and kinds of relations could be identified as repellers for a particular application domain. For example, suppose that in the ILEX domain, some artifacts (such as necklaces) have as features both a set of design characteristics and the person who found the artifact. Once the artifact is introduced, it becomes more important to present the design characteristics rather than the person who found that artifact. This preference might be captured by classifying the relation connecting the proposition conveying the person who found it to the proposition introducing the artifact as arepeller.

Our fourth goal is to avoid redundancy by discouraging the selection of propositions connected by a Redundancy_Relation to previously selected propositions. Once a proposition is selected, we identify the vertices **(redundant_to_selected vertices)** which are connected to the selected vertex by the Redundancy_Relation (Step **2-e**). For each redundant_to_selected vertex, we penalize the weights on the edges of the vertex except the edge connected to the priority_vertex (Step **2-f**) and hence decrease the probability of that vertex being chosen for inclusion in the same response.

We have so far described how the content of a single response is constructed in our framework. To capture a situation where the system is engaged in a dialogue with the user and must generate additional responses for each subsequent user request, we need to ensure that discourse flows naturally. Thus, the ranking algorithm must take the previ-

1. Set an importance score to the ***priority_vertex*** and empty the ***selected_vertices*** set
2. Repeat steps 2-a to 2-g until the stopping criteria is met:

- **(a)** use the weighted PageRank metric to calculate the importance scores of all vertices excluding the ***priority_vertex*** and the vertices in the ***selected_vertices*** set
- **(b)** mark the vertex that received the highest score as ***selected*** and add it to the ***selected_vertices*** set
- **(c)** decrease the weight of the edge between the ***selected_vertex*** and the ***priority_vertex***
- **(d)** penalize the weights of the edges between the ***selected_vertex*** and the vertices which are connected to it by a repeller via the penalty factor, if they weren't already adjusted
- **(e)** find the vertices that are connected by a Redundancy_Relation to the ***selected_vertex*** (if any) and mark them as ***redundant_to_selected***
- **(f)** penalize the weights of all edges of ***redundant_to_selected*** vertices via the redundancy penalty factor except the edges connected to the **priority_vertex**
- **(g)** set the importance scores of the ***priority_vertex*** and the vertices in the ***selected_vertices*** set to the highest importance score calculated so far

3. Penalize the weights of the edges of the vertices in the ***selected_vertices*** set via the penalty factor, if they weren't adjusted in Step 2
4. Increase the weights of the edges of all other vertices (excluding the vertices in the ***selected_vertices*** set and ***redundant_to_selected*** vertices) via the boost factor

Figure 1: Our Ranking Algorithm for Content Selection.

ous discourse into account in order to identify and preferably select propositions that have not been conveyed before and to determine when repetition of a previously communicated proposition is appropriate. So once a proposition is included in a response, we have to reduce its ability to compete for inclusion in subsequent responses. Thus once a proposition is conveyed in a response, the weight of the edge connecting the corresponding vertex to the priority_vertex is reduced (Step **2-c** in Figure 1). Once a response is completed, we penalize the weights of the edges of each vertex that has been selected for inclusion in the current response via a penalty factor (if they aren't already adjusted) (Step **3** in Figure 1). We use the same penalty factor (which is used in Step **2-d** in Figure 1) on each edge so that all edges connected to a selected vertex are penalized equally. However, it isn't enough just to penalize the edges of the vertices corresponding to the communicated propositions. Even after the penalties are applied, a proposition that has just been communicated might receive a higher importance score than an uncommunicated proposition[1]. In order to allow all propositions to become important enough to be said at some point, the algorithm increases the weights of the edges of all other vertices in the graph if they haven't already been decreased (Step **4** in Figure 1), thereby increasing their ability to compete in subsequent responses. In the current implementation, the weight of an edge is increased via a boost factor after a response if it is not connected to a proposition included in that response. The boost factor ensures that all propositions will eventually become important enough for inclusion.

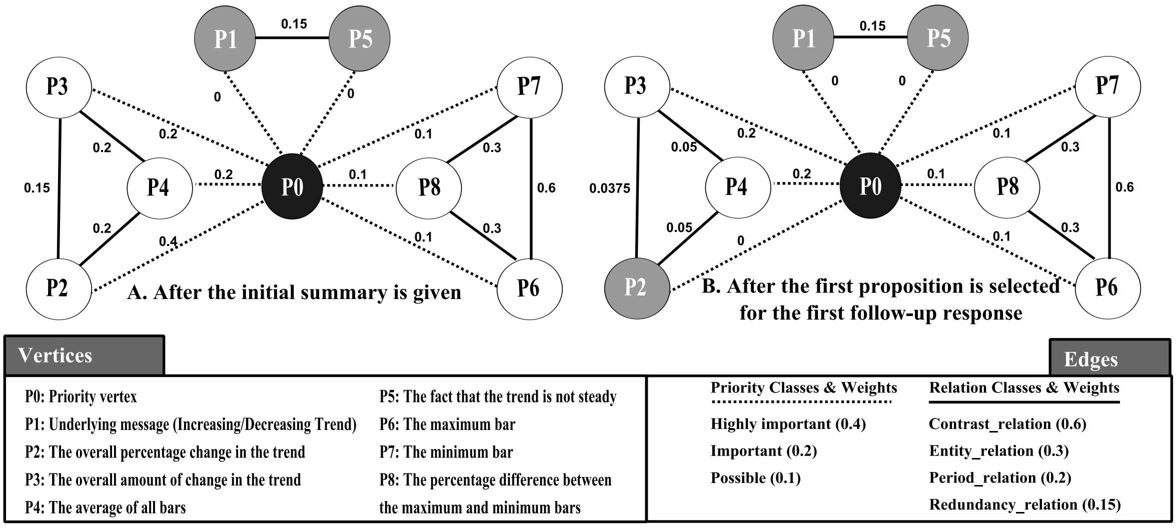
3 Application in a Particular Domain

This section illustrates the application of our framework to a particular domain and how our framework facilitates flexible content selection. Our example is content selection in the SIGHT system (Elzer et al., 2007), whose goal is to provide visually impaired users with the knowledge that one would gain from viewing information graphics (such as bar charts) that appear in popular media. In the current implementation, SIGHT constructs a brief initial summary (Demir et al., 2008) that conveys the primary message of a bar chart along with its salient features. We enhanced the current SIGHT system to respond to user's follow-up requests for more information about the graphic, where the request does not specify the kind of information that is desired.

The first step in using our framework is determining the set of propositions that might be conveyed in this domain. In our earlier work (Demir et al., 2008), we identified a set of propositions that capture information that could be determined by looking at a bar chart, and for each message type defined in SIGHT, specified a subset of these propositions that are related to this message type. In our example, we use these propositions as candidates for inclusion in follow-up responses. Figure 2 presents a portion of the relation_graph, where some of the identified propositions are represented as vertices.

The second step is optionally assessing the a priori importance of each proposition. In user

[1] We observed that it might happen if a vertex is connected only to the priority_vertex.



P0: Priority vertex
P1: Underlying message (Increasing/Decreasing Trend)
P2: The overall percentage change in the trend
P3: The overall amount of change in the trend
P4: The average of all bars
P5: The fact that the trend is not steady
P6: The maximum bar
P7: The minimum bar
P8: The percentage difference between the maximum and minimum bars

Priority Classes & Weights	Relation Classes & Weights
Highly important (0.4)	Contrast_relation (0.6)
Important (0.2)	Entity_relation (0.3)
Possible (0.1)	Period_relation (0.2)
	Redundancy_relation (0.15)

Figure 2: Subgraph of the Relation_graph for Increasing and Decreasing Trend Message Types.

studies (Demir et al., 2008), we asked subjects to classify the propositions given for a message type into one of three classes according to their importance for inclusion in the initial summary: **essential**, **possible**, and **not important**. We leverage this information as the a priori importance of vertices in our graph representation. We define three priority classes. For the propositions that were not selected as *essential* by any participant, we classify the edges connecting these propositions to the priority_vertex into **Possible** class. For the propositions which were selected as *essential* by a single participant, we classify the edges connecting them to the priority_vertex into **Important** class. The edges of the remaining propositions are classified into **Highly Important** class. In this example instantiation, we assigned different numeric scores to these classes where Highly_Important and Possible received the highest and lowest scores respectively.

The third step requires specifying the relations between every pair of related propositions and determining the weights associated with these relations in the relation_graph. First, we identified propositions which we decided should be connected by the Redundancy_Relation (such as the propositions conveying “the overall amount of change in the trend” and “the range of the trend”). Next, we had to determine other relations and assign relative weights. Instead of defining a unique relation for each related pair, we defined three relation classes, and assigned the relations between related propositions to one of these classes:

- **Period_Relation:** expresses a relation between two propositions that span the same time period
- **Entity_Relation:** expresses a relation between two propositions if the entities involved in the propositions overlap
- **Contrast_Relation:** expresses a relation between two propositions if the information provided by one of the propositions contrasts with the information provided by the other

We determined that it was very common in this domain to deliver contrasting propositions together (similar to other domains (Marcu, 1998)) and therefore we assigned the highest score to the Contrast_Relation class. For local focusing purposes, it is desirable that propositions involving common entities be delivered in the same response and thus the Entity_Relation class was given the second highest score. On the other hand, two propositions which only share the same period are not very related and conveying such propositions in the same response could cause the text to appear “choppy”. We thus identified the Period_Relation class as a repeller and assigned the second lowest score to relations in that class. Since we don’t want redundancy in the generated text, the lowest score was assigned to the Redundancy_Relation class. The next section shows how associating particular weights with the priority and relation classes changes the behavior of the framework.

In the domain of graphics, a collection of descriptions of the targeted kind which would facilitate a learning based model isn’t available. However, the accessibility of a corpus in a new domain would allow the identification of the propositions along with their relations to each other and the determination of what weighting scheme and adjustment policy will produce the corpus within reasonable bounds.

3.1 Generating Flexible Responses

The behavior of our framework is dependent on a number of design parameters such as the weights associated with various relations, the identification of repellers, the a priori importance of information (if applicable), and the extent to which conveying redundant information should be avoided. The framework allows the application developer to adjust these factors resulting in the selection of different content and the generation of different responses. For instance, in a very straightforward setting where the same numeric score is assigned to all relations, the a priori importance of information would be the major determining factor in the selection process. In this section, we will illustrate our framework's behavior in SIGHT with three different scenarios. In each case, the user is assumed to post two consecutive requests for additional information about the graphic in Figure 3 after receiving its initial summary.

In our first scenario (which we refer to as "base-setting"), the following values have been given to various design parameters that must be specified in order to run the ranking algorithm. 1) The weights of the relations are set to the numeric scores shown in the text labelled **Edges** at the bottom (right side) of Figure 2. 2) The stopping criteria which specifies the number of propositions selected for inclusion in a follow-up response (Step **2** in Figure 1) is set to four. 3) The amount of decrease in the weight of the edge between the priority_vertex and the vertex selected for inclusion (Step **2-c** in Figure 1) is set to that edge's original weight. Thus, in our example, the weight of that edge is set to 0 once a proposition has been selected for inclusion. 4) The penalty and the redundancy penalty factors which are used to penalize the edges of a selected vertex and the vertices redundant to the selected vertex (Steps **2-d** and **3**, and **2-f** in Figure 1) are set to the quotient of the highest numeric score initially assigned to a relation class divided by the lowest numeric score initially assigned to a relation class. A penalized score for a relation class is computed by dividing its initial score by the penalty factor. The edges of a vertex are penalized by assigning the penalized scores to these edges based on the relations that they represent. This setting guarantees that the weight of an edge which represents the strongest relation cannot be penalized to be lower than the score initially assigned to the weakest relation. 5) The boost factor which is used to favor the selection of previously unconveyed propositions for inclusion in subsequent responses (Step **4** in Figure 1) is set to the square root of the penalty factor. Thus, the weights of the edges connected to vertices of previously communicated propositions are restored to their initial scores slowly.

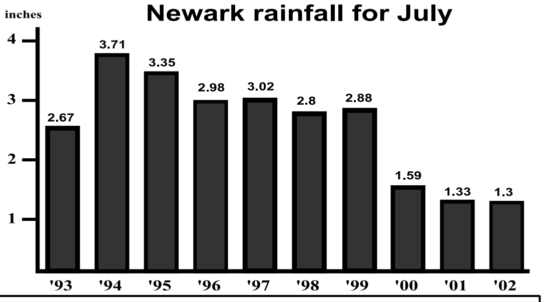
Since in our example, the initial summary has already been presented, we treat the propositions conveyed in that summary (P1 and P5 in Figure 2) as if they had been conveyed in a follow-up response and penalize the edges of their corresponding vertices (Steps **2-c** and **3** in Figure 1). Thus, before we invoke the algorithm to construct the first follow-up response, the weights of edges of the graph are as shown in Figure 2-A. Within this base-setting, SIGHT generates the set of follow-up responses shown in Figure 3A.

In our first scenario (base-setting), we assumed that the user is capable of making mathematical deductions such as inferring "the overall amount of change in the trend" from "the range of the trend"; thus we identified such propositions as sharing a Redundancy_Relation. Young readers (such as fourth graders) might not find these propositions as redundant because they are lacking in mathematical skills. In our second scenario, we address this issue by setting the redundancy penalty factor to 1 (**Step 2-f** in Figure 1) and thus eliminate the penalty on the Redundancy_Relation. Now, for the same graphic, SIGHT generates, in turn, the second alternative set of responses shown in Figure 3B. The responses for the two scenarios differ in the second follow-up response. In the first scenario, a description of the smallest drop was included. However, in the second scenario, this proposition is replaced with the overall amount of change in the trend. This proposition was excluded in the first scenario because the redundancy penalty factor made it drop in importance.

Our third scenario shows how altering the weights assigned to relations may change the responses. Consider a situation where the Contrast_Relation is given even higher importance by doubling its score; this might occur in a university course domain where courses on the same general topic are contrasted. SIGHT would then generate the third alternative set of follow-up responses shown in Figure 3C. The algorithm is more strongly forced to group propositions that

Initial Summary

Following a moderate rise between the year 1993 and the year 1994, the graphic shows a decreasing trend in the amount of newark rainfall for july over the period from the year 1994 to the year 2002. The amount of newark rainfall for july shows the largest drop of about 1.29 inches between the year 1999 and the year 2000. With the exception of a few rises, slight decreases are observed almost every year over the period from the year 1994 to the year 2002.



First follow-up response:
The amount of newark rainfall for july ranges from 3.71 to 1.3 inches and shows a decrease of nearly 65 percent over the period from the year 1994 to the year 2002. The amount of this rainfall for july averages 2.55 inches.

Second follow-up response:
Recall that there is a decreasing trend in the amount of newark rainfall for july over the period from the year 1994 to the year 2002. The amount of newark rainfall for july shows the smallest drop of about 0.03 inches between the year 2001 and the year 2002. The year 1994 at 3.71 inches has the highest amount of rainfall for july and the year 2002 at 1.3 inches has the lowest amount of rainfall for july.

A. First alternative set of responses is shown above (base-setting)

First follow-up response:
The amount of newark rainfall for july ranges from 3.71 to 1.3 inches and shows a decrease of nearly 65 percent over the period from the year 1994 to the year 2002. The amount of this rainfall for july averages 2.55 inches.

Second follow-up response:
Recall that there is a decreasing trend in the amount of newark rainfall for july over the period from the year 1994 to the year 2002. The year 1994 at 3.71 inches has the highest amount of rainfall for july and the year 2002 at 1.3 inches has the lowest amount of rainfall for july.***The difference between the amount of newark rainfall for july in the year 1994 and that in the year 2002 is 2.41 inches.***

B. Second alternative set of responses is shown above (the Redundancy_Relation is not penalized)

First follow-up response:
The amount of newark rainfall for july ranges from 3.71 to 1.3 inches and shows a decrease of nearly 65 percent over the period from the year 1994 to the year 2002. **The year 1994 at 3.71 inches has the highest amount of rainfall for july and the year 2002 at 1.3 inches has the lowest amount of rainfall for july.**

Second follow-up response:
Recall that there is a decreasing trend over the period from the year 1994 to the year 2002 in the amount of newark rainfall for july, which **shows the largest drop of 1.29 inches between the year 1999 and the year 2000.** At the year 1997 and the year 1999, unusual rises are observed in the amount of this rainfall for july, which **shows the smallest drop of 0.03 inches between the year 2001 and the year 2002.**

C. Third alternative set of responses is shown above (the numeric score of the Contrast_Relation is doubled)

Figure 3: Initial Summary and Follow-up Responses.

are in a contrast relation (shown in bold), which changes the ranking of these propositions.

4 Evaluation

To determine whether our framework selects appropriate content within the context of an application, and to assess the contribution of the discourse related considerations to the selected content and their impact on readers' satisfaction, we conducted two user studies. In both studies, the participants were told that the initial summary should include the most important information about the graphic and that the remaining pieces of information should be conveyed via follow-up responses. The participants were also told that the information in the first response should be more important than the information in subsequent responses.

Our goal in the first study was to evaluate the effectiveness of our framework (base-setting) in determining the content of follow-up responses in SIGHT. To our knowledge, no one else has generated high-level descriptions of information graphics, and therefore evaluation using implementations of existing content selection modules in the domain of graphics as a baseline is not feasible. Thus, we evaluated our framework by comparing the content that it selects for inclusion in a follow-up response for a particular graphic with the content chosen by human subjects for the same response. Twenty one university students participated in the first study and each participant was presented with the same four graphics. For each graphic, the participants were first presented with its initial summary and the set of propositions (18 different propositions) that were used to construct the relation_graph in our framework. The participants were then asked to select the four propositions that they thought were most important to convey in the first follow-up response.

For each graphic, we ranked the propositions with respect to the number of times that they were selected by the participants and determined the position of each proposition selected by our framework for inclusion in the first follow-up response with respect to this ranking. The propositions selected by our framework were ranked by the participants as the *1st, 2nd, 3rd, and 5th* in the first graphic, as the *1st, 3rd, 4th, and 5th* in the second graphic, as the *1st, 2nd, 3rd, and 6th* in the third graphic, and as the *2nd, 3rd, 4th, and 6th* in the fourth graphic. Thus for every graph, three of the four propositions selected by our framework were also in the top four highly-rated propositions selected by the participants. Therefore, this study demonstrated that our content selection framework selects the most important information for inclusion in a response at the current exchange.

We argued that simply running PageRank to select the highly-rated propositions is likely to lead to text that does not cohere because it may contain unrelated or redundant propositions, or fail to communicate related propositions. Thus, our approach iteratively runs PageRank and includes discourse related factors in order to allow what has been selected to influence the future selections and consequently improve text coherence. To verify this argument, we conducted a second study with four graphics and two different sets of follow-up responses (each consisting of two consecutive responses) generated for each graphic. We constructed the first set of responses **(baseline)** by running PageRank to completion and selecting the top eight highly-rated propositions, where the top four propositions form the first response. The content of the second set of responses was identified by our approach. Twelve university students (who did not participate in the first study) were presented with these four graphics along with their initial summaries. Each participant was also presented with the set of responses generated by our approach in two graphics and the set of responses generated by the baseline in other cases; the participants were unaware of how the follow-up responses were generated. Overall, each set of responses was presented to six participants.

We asked the participants to evaluate the set of responses in terms of their quality in conveying additional information (from 1 to 5 with 5 being the best). We also asked each participant to choose which set of responses (from among the four sets of responses presented to them) best provides further information about the corresponding graphic. The participants gave the set of responses generated by our approach an average rating of **4.33**. The average participant rating for the set of responses generated by the baseline was **3.96**. In addition, the lowest score given to the set of responses generated by our approach was 3, whereas the lowest score that the baseline received was 2. We also observed that the set of responses generated by our approach was selected as the best set by eight of the twelve participants. Three of the remaining four participants selected the set of responses generated by the baseline as best (although they gave the same score to a set of responses generated by our approach). In these cases, the participants emphasized the wording of the responses as the reason for their selection. Thus this study demonstrated that the inclusion of discourse related factors in our approach, in addition to the use of PageRank (which utilizes the a priori importance of the propositions and their relations to each other), contributes to text coherence and improves readers' satisfaction.

5 Conclusion

This paper has presented our implemented domain-independent content selection framework, which contains domain-dependent features that must be instantiated when applying it to a particular domain. To our knowledge, our work is the first to select appropriate content by using an incremental graph-based ranking algorithm that takes into account the tendency for some information to seem related or redundant to other information, the a priori importance of information, and what has already been said in the previous discourse. Although our framework requires a knowledge engineering phase to port it to a new domain, it handles discourse issues without requiring that the developer write code to address them. We have demonstrated how our framework was incorporated in an accessibility system whose goal is the generation of texts to describe information graphics. The evaluation studies of our framework within that accessibility system show its effectiveness in determining the content of follow-up responses.

6 Acknowledgements

The authors would like to thank Debra Yarrington and the members of the NLP-AI Lab at UD for their help throughout the evaluation of this work. This material is based upon work supported by the National Institute on Disability and Rehabilitation Research under Grant No. H133G080047.

References

S. Brin and L. Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117.

G. Carenini and J. Moore. 2006. Generating and evaluating evaluative arguments. *Artificial Intelligence*, 170(11):925–452.

C. Chiarcos and M. Stede. 2004. Salience-Driven Text Planning. *In Proc. of INLG'04*.

S. Demir, S. Carberry, and K. F. McCoy. 2008. Generating Textual Summaries of Bar Charts. *In Proc. of INLG'08*.

S. Elzer, E. Schwartz, S. Carberry, D. Chester, S. Demir, and P. Wu. 2007. A browser extension for providing visually impaired users access to the content of bar charts on the web. In *Proc. of WEBIST'2007*.

G. Erkan and D. Radev. 2004. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

L. C. Freeman. 1979. Centrality in Social Networks: I. Conceptual Clarification. *Social Networks*, 1:215–239.

J. Lester and B. Porter. 1997. Developing and empirically evaluating robust explanation generators: the KNIGHT experiments. *Computational Linguistics*, 23(1):65–101.

D. Marcu. 1998. The rhetorical parsing, summarization, and generation of natural language texts. *PhD. Thesis, Department of Computer Science, University of Toronto*.

K. McKeown. 1985. Discourse strategies for generating natural-language text. *Artificial Intelligence*, 27(1):1–41.

J. Moore and C. Paris. 1993. Planning text for advisory dialogues: capturing intentional and rhetorical information. *Computational Linguistics*, 19(4):651–694.

J. Moore. 1993. Indexing and exploiting a discourse history to generate context-sensitive explanations. *In Proc. of HLT'93*, 165–170.

R. Navigli and M. Lapata. 2007. Graph Connectivity Measures for Unsupervised Word Sense Disambiguation. *In Proc. of IJCAI'07*, 1683–1688.

M. O'Donnell, C. Mellish, J. Oberlander, and A. Knott. 2001. ILEX: an architecture for a dynamic hypertext generation system. In *Natural Language Engineering*, 7(3):225–250.

E. Reiter and R. Dale. 1997. Building applied natural language generation systems. In *Natural Language Engineering*, 3(1):57–87.

R. Sinha and R. Mihalcea. 2007. Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity. *In Proc. of ICSC'07*.

S. Sripada, E. Reiter, J. Hunter, and J. Yu. 2001. A Two-Stage Model for Content Determination. *In Proc. of ENLGW'01*.

M. Walker, S.J. Whittaker, A. Stent, P. Maloor, J. Moore, M. Johnston, and G. Vasireddy. 2004. Generation and evaluation of user tailored responses in multimodal dialogue. In *Cognitive Science*, 28(5):811–840.