Integrating Rhetorical-Semantic Relation Models for Query-Focused Summarization

Sasha Blair-Goldensohn and Kathleen McKeown
Department of Computer Science
Columbia University

May, 2006

Abstract

We present our recent work on query-focused summarization, focusing on our efforts in building and applying models of rhetorical-semantic relations (RSRs) such as contrast and causality. We overview ongoing work in extracting and evaluating RSR models. We describe our system for query-focused summarization, focusing on an enhanced, feature-based framework. We present results of experiments to measure the impact of both RSR and other features on selection and ordering of summary content. We conclude with an overview of results from the official DUC06 evaluation.

1 Introduction

In DUC06, our participation served to propel two different yet complimentary avenues of research. First, we used the opportunity to add a novel, semantically-motivated aspect to our work by integrating models of rhetorical-semantic phenomena such as causality and contrast. Second, given data available from previous DUCs, we retrained our system to take best advantage of all features, properly weighting both both new rhetorical-semantic features, as well as existing features. In this paper, we will focus on both aspects of our work, with particular emphasis on our novel use of rhetorical semantic relations and experimental results.

Before delving into more detail, we touch on our motivation for exploring rhetorical/semantic notions in the DUC context, using the following example, adapted from a biographically-focused document set about Sonia Gandhi in the DUC04 data. Consider the following two sentences, which were taken from separate original documents and placed together in a summary as follows:

> The BJP had shrugged off the influence of the 51-year-old Sonia Gandhi when she stepped into politics **early** this year, dismissing her as a **foreigner**. Sonia Gandhi is **now** an Indian **citizen**.

Clearly, these sentences, placed together, are well-chosen. And in large part their felicitousness stems from the implicit contrast being made between (*early, now*) and (*foreigner, citizen*).

We believe that the ability to actively create such combinations using rhetorical-semantic models is attractive for at least two complementary reasons. First, *content selection* can be improved by using these models to that certain information is closely linked to the topic being summarized, when it might otherwise appear marginal based on purely lexical criteria. Second, *content ordering* can be improved by using these new criteria to arrange summary content so as to form multi-sentence spans of strongly coherent text.

In the remainder of this paper, we describe our progress in this new direction, explain how we have integrated this idea into our existing system and present experimental results of our ongoing experiments and evaluations as well as the official DUC06 results.

2 Related Work

Work on rhetorical and discourse theory has a long tradition in computational linguistics (see Moore et al.[22] for a comprehensive overview). Areas of research include informational and semantic relationships[10, 14, 18]; processing models in discourse, including attentional and intentional structure [8]; as well as work that takes models of both information structure and processing into account for modeling [23, 1] or generating [19, 21] text.

In rhetorical structure theory (RST)[14], the precise number and taxonomy of relations have been a point of contention [11, 23], but its basic tenets have formed the basis for approaches that aim to model the informative content of text, such as text summarization [16] and essay-grading [6].

Marcu and Echihabi [17] describe a system which recognizes RST-derived relationships between text spans using a model extracted automatically from large corpora using cue phrases. Their method first extracts from a corpus all instances of cue phrase patterns associated with a given relation, e.g. the pattern "*A*. However, *B*.", which is associated with a Contrast relation (where *A* is a full sentence and *B* is the remainder of a sentence after *However*). Then, for any pair of text spans *X,Y*, whose relationship is unknown, its "Contrast-ness" can be estimated according to how closely it matches the aggregate properties of *A,B* pairs seen in this and other Contrast-associated patterns over a large corpus.

This work can in turn be seen as a variant of earlier data-driven approaches like Latent Semantic Analysis (LSA)[7]. However, a crucual difference is that LSA does not distinguish a semantics of word relationship; it can offer the information that two words are "related," but the manner of that relation is unspecified.

Several recent systems in question answering and summarization attempt to apply LSA and other kinds of semantic resources such as WordNet[20, 9, 12] and Marcu's experiments with RST-informed summarization[16] shows the importance of using rhetorical links identified via cue-phrase patterns in forming extracts. However, to our knowledge, there have not been other systems in these areas which attempt to apply the type of automatically-derived rhetorical models built by Marcu and Echihabi [17].

3 Rhetorical-Semantic Relations

3.1 Description

Our concept of Rhetorical-Semantic Relations derives most directly from the work in rhetorical structure discussed in the previous psection. As noted there, while much of this work differs on issues of relation taxonomy, there are also basic recurring themes; for instance, that information relationships can include both an intentional and informational dimension, and that certain relationships such as causality and contrast, are of recurring importance even if their precise descriptions vary [11, 23]. We choose to model three distinct relations, Cause, Contrast and Adjacent. We describe the intuition behind these relations in this section, and provide a procedural definition in the next.

First, we should note that we are indebted to the work of Marcu and Echihabi [17] (M&E) both in our intuitive conception of these relations, as well as our implentation of a model for them. M&E make the persuasive case that many of the apparent many of the differences in RST taxonomies can be abstracted away by using a coarser-grained model. For instance, we follow their lead in choosing to model a single Cause relation, thus avoiding problematic distinctions between, e.g., "volitional" vs "non-volitional" causality made in Mann and Thompson's RST taxonomy [14]; similarly we follow M&E in modeling a Contrast relation which subsumes "antithesis," "concession," and "contrast," among others.

While we see our Cause and Contrast as drawing on abstractions of these informational-semantic relations of RST, our Adjacent relation is conceived with a less specific semantics. Instead, it is meant as a sort of catch-all for relations which pertain over information presented in literally adjacent sentences (and as such can in some instances subsume more semantically specific relations, including Cause and Contrast). The idea of Adjacent as a relation can perhaps be viewed as a shallow version of Grosz and Sidner's [8] "satisfaction-precedes" intentional constraint, and follows on Martin's observation that "... simply putting clauses next to each other suggests some logical connection between them, whether or not [it] is made explicit ..."[18, p.165]. As compared to the more semantically specific, informational relations Cause and Contrast, the Adjacent relation serves as a complimentary mechanism to model the intentional tendencies implicit in ordering, without selecting for specific informational semantics.

Lastly, we note an important departure in our work from previous models of rhetorical and discourse structure, as well as the more recent work of Marcu and Echihabi [17]. Namely, we consider these RSR models as having potential value beyond analysis of the information structure within a single, human-authored document. We attempt instead to actively "create" RSR-like relationships within summary responses by including sentence pairings which resemble our acquired models. The "Sonia Gandhi" example from the Introduction section demonstrates the type of result we hope to achieve with this process.

In summary, we take the inspiration from earlier work about the *kinds* of relations which are important within documents, yet we aim to consider possibilities for understanding and using the models in a broader context. For this reason, we depart from the "Structure" terminology and instead coin the term Rhetorical-Semantic *Relations*.

3.2 Definition

We have so far described the basic intuition of these relations; here we define them procedurally. To this end, we rely on a set of patterns similar to the "However, ..." pattern used by Marcu and Echihabi [17] mentioned above. These patterns are based on cue words such as *however* and *therefore*, which we have refined from published lists (Marcu's thesis [15] is an especially good resource). In all, we assemble approximately forty such patterns for each of the Contrast and Cause relations.

In this sense, the definition of these relations is quite simple: namely, any text span pair which matches one of our cue phrase patterns is an instance of the associated relation. However, the primary objection to such a definition is that cue phrase patterns can sometimes match against spans which do not, in fact, express the associated rhetorical semantic function. For instance, for the Contrast pattern, "Although A , B .", the alignment of A and B will be incorrect if the comma in the pattern matches the first comma in "Although he studied, studied and studied, he still failed."

Here, we view the issue as a lack of precision in the pattern, and expect that by improving pattern precision (in this case, syntactic analysis or the presence of the additional cue word "still" might be used) we can derive instances which result in better models.

3.3 Extraction and Modeling

The framework we choose for extracting and modeling our RSRs very much adopts the approach of Marcu and Echihabi [17] (henceforth, M&E). For brevity, we refer the reader to their work for details of their framework, and describe here only the ways in which we have extended or diverged from their work.

As in the M&E model, we mine a large corpus for our relations of interest, using cue phrase patterns to find relation instances. We find Contrast and Cause instances as in M&E, as well as deriving instances of a NoRelSame relation, which consists simply of sentence pairs taken from the same document

which are separated by at least four intervening sentences. (The purpose of NoRelSame is to provide a baseline of topically related spans across which more specific relations like Cause are assumed not to pertain.) In addition to these relations, we add the Adjacent relation described above, which is simply extracted by randomly choosing adjacent sentences from a document. The corpus we mine is the English Gigaword Corpus[1], of nearly 5 million newswire documents. We extract approximately two million instances matches for Cause, six million of Contrast, and we choose to extract four million instances each for Adjacent and NoRelSame.

As in M&E, we extract from these instances a single feature, namely word-pair frequencies of the individual lexical pairs found across two related text spans. That is, if we find, "He studied, therefore he passed.", we consider it to be positive evidence of a Cause relation between all word pairs across the spans "he studied" and "he passed," i.e. (*he* , *studied*), (*he*, *passed*), (*studied*, *he*), (*studied*, *passed*). These frequencies are tallied over the entire corpus, and used as maximum likelihood estimates in a Bayesian classifier.

After building the model, we first verified that, as expected, it achieved comparable classification accuracy to the Marcu model on which it was based in the task of classifying unknown text-span pairs as to the actual relation type which was used to extract that pair (e.g., Cause versus Contrast). At that point we examined several possible methods to improve this accuracy. Recall that our overall goal is to use the constructed models in the applied task of summarization; nonetheless, we believe that increased classifiaction accuracy is our best heuristic for the quality of these models. For instance, as mentioned in our procedural definition of RSRs, an improvement to the cue phrase extraction patterns would be expected to filter through to the RSR models, and in turn make, e.g. the Cause versus Contrast classification performance improve. This improvement would be due to the relevant model representing more accurately the tendencies of, e.g. Contrast relationships (in the case of our example, the fact that "studied" and "failed" are in contrast), and having a more accurate model would in turn improve the possibility contribution to our applied task of summarization.

We have thus far implemented several experiments to attempt to improve classification performance. These experiments included:

- Examination of smoothing parameters and methods
- Comparison of tradeoffs in training set, vocabulary and stoplist sizes
- Investigation of the impact of using automated topical segmentation to reduce one source of noise in relation extraction; namely, improving pattern precision by preventing (or, in the case of NoRelSame, requiring) matches which cross a topic boundary.

While we do not present detailed results here for the purposes of brevity, we have found that classifaction accuracy can be improved using the results of each type of experiment, and that in several cases the effect is significant. Accordingly, the RSR models used in the following section use models which have been extracted and parameterized to produce the best classification accuracy.

[1] available from LDC at `http://www.ldc.upenn.edu`

4 System Implementation and Integration of RSRs

4.1 Overview

DefScriber [5] is a system which creates dynamic, multi-sentence answers to definitional, biographical and open-form questions using a hybrid of goal- and data-driven methods. We used DefScriber successfully in DUC04 [4] to create biographical summaries, and adapted it in DUC05 [3] to create summaries which responded to broad-topic questions. Our cited previous papers describe the basic concepts of our hybrid method and DefScriber's system components; we focus here on two updates to the system: (1) an improved framework for using all features via a retraining against DUC data and implementation of a post-content-selection reordering step (2) the incorporation of RSRs into this framework to improve answer content and coherency.

In order to make these two extensions of the system, we decided to refactor our system's sentence selection and ordering algorithms to consider all of the features described in Table1 in a common framework. This combined feature set includes features which had proven successful in previous system versions, as well as a new feature derived from our RSR models. Using this new framework and feature set, we were able to carry out several experiments to train and test our system using previous DUC data sets.

To get the clearest picture of the specific contribution of RSRs, we decided to carry out this training in two phases. In the first phase, we did not include the RSR-derived feature, and focused on estimating optimal parameters for the other features only. Then, in the second phase, we set the non-RSR feature parameters to their optimal learned values, and explored various ways to use the additional information offered by our RSR models.

Before presenting the details and results of the experiments, we outline the algorithm used for sentence selection by DefScriber. It is an iterative algorithm which, at each iteration, greedily chooses a "next" sentence for the summary by maximizing a weighted sum of feature values. DefScriber's answer summaries are created by repeatedly choosing a "next" sentence until a length criterion has been met (or no input sentences remain).[2]

Note several important points in DefScriber's operation which precede this algorithm:

- The set S of "Non-Specific Definitional" (NSD) sentences, i.e. sentences which have any broad relevance to the topic being described/defined, have been identified and separated

[2] In practice, the algorithm is run within a beam search framework, where the most promising n summaries are kept after each iteration in order to counteract the possibility of local maxima which can result from greedy choices in the presence of several order-dependent features such as LexicalCohesion.

Centroid Relevance to overall topic.	IDF-weighted cosine distance between s and S.
Coverage Coverage of sub-topics.	One if c has least (or tied for least) representatives already in P as compared with other clusters in C; zero otherwise.
LexicalCohesion Lexical cohesion with previous sentence.	IDF-weighted word-stem vector cosine distance between s and p.
Query Overlap with query terms.	Word-stem vector cosine distance between s and q. Term weighting uses IDF but penalizes terms which have already been covered in P.
DocumentCohesion Original document cohesion.	Score is non-zero iff s and p are from the same input document, and p preceeded s in original ordering; score is inversely proportional to number of sentences between p and s. If p immediately preceeded s and was joined by a discourse cue-phrase, an additional bonus is given.
RSR RSR-derived score.	Calculated by comparing by "classifying" whether s and p appear to be related by our models for Cause, Contrast and Adjacent RSR relations. Calculation parameters are discussed in Subsection 4.3.

Table 1: Features used in DefScriber's sentence selection algorithm. Features are evaluated for a candidate sentence s to be added to an under-construction summary P with last sentence p; s is a member of a cluster c in a set of clusters C which parition the full set of relevant input sentences S.

out from the (possibly) larger set of all input document sentences.

- The sentences in S have been clustered into a set of clusters C, in an effort to segment the sub-topics of the topic being described in the answer/summary.
- Any "Genus-Species" (GS) sentences which provide a category-differentiator (or "is-a") statement for the topic being described, are identified from among S using lexical-syntactic patterns.
- The first sentence in the summary is chosen as the highest-ranked (by Centroid feature) GS sentence, or simply highest-ranked setnence in S if no GS sentences were found. This sentences is passed into the algorithm as the initial, single-sentence summary P.
- The set of features F used in sentence selection generally includes all features in Table 1, but the RSR feature was excluded when running DefScriber for the first phase of experiments described in Section 4.2.

Following these initial steps, DefScriber repeatedly selects the next sentence for the summary following the algorithm below.

Algorithm 1 ChooseNextSentence(P, C, F, W)

INPUT:
P current summary with ¿ 0 sentences
C set of candidate sentence clusters
F set of features to evaluate for candidate sentences
W set of weight parameters defined over all $f \in F$

OUTPUT:
the best next-sentence candidate n

$B \leftarrow GetBestUnused(C, P)$ {for each cluster in C, extract its highest-ranked (by Centroid feature) sentence not in P}
for all $b \in B$ **do**
  $Score[b] \leftarrow 0$
  for all $f \in F$ **do**
    $Score[b] \leftarrow CalcFeatScore(f, b, P) * W[f]$
  end for
end for
return $b \in B$ with maximum $Score[b]$

4.2 Experiments with Non-RSR Features

4.2.1 Sentence Selection

In this first set of experiments, the goal was to estimate the parameters for all features other than RSR to be used in sentence selection. While these features had been included in earlier system versions, we decided a retraining was important for several reasons: (a) The feature calculations had in some cases been refined (b) We had not trained specifically on DUC data in the past (c) We wanted to determine the best values before examining the RSR features, so that any improvement achieved via these features would be in addition to an already-optimized score.

In these experiments, we had five distinct feature weights to train (Centroid, Coverage, LexicalCohesion, Query and DocumentCohesion). Even using a single, linear function of the features as we do in the ChooseNextSentence() algorithm, testing over five possible weights for each feature would involved 5^5 permutations. Since our system takes several minutes to summarize a single DUC input set, an exhaustive search was not a feasible approach. Thus, we decided to use a hill-climbing search with randomized restarts in order to explore the search space, using macro-averaged ROUGE scores (specifically the SU4 recall measure, one of the official measures used in DUC05) over a training set of topics as our heuristic for evaluating the results at each weighting.

We trained the system separately on DUC04 (task 5) and DUC05 topics, using in each case a randomly selected 80 percent of the data for training, and 20 percent for testing. Our intuition for examining the data sets separately was that these sets had somewhat different properties; DUC04 summaries used biographically-focused topics and 100-word models, whereas DUC05 summaries (like those of DUC06) are on broader topics with 250-word models.

Table 2 shows results for the experiment broken down by training set and system parameter setting. We show in the table only

Setting	Training	Test
DUC05 best hill-climb	0.1351	0.1300
DUC05 median hill-climb	0.1325	0.1285
DUC05 baseline	0.1307	0.1267
DUC05 original	0.1285	0.1236
DUC05 peer best	0.1290	0.1337
DUC04 best hill-climb	0.1233	0.1268
DUC04 median hill-climb	0.1167	0.1165
DUC04 baseline	0.1152	0.1223
DUC04 original	0.1177	0.1209

Table 2: ROUGE SU-4 recall macro-averaged scores from experiments in Section 4.2.

Setting	Cent	Covg	LexC	Query	DocC
DUC05 best h-c	2	2	1	4	4
DUC05 median h-c	0	2	0	1	0
DUC05 baseline	1	1	1	1	1
DUC04 best h-c	2	3	0	1	0
DUC04 median h-c	2	4	0	3	3
DUC04 baseline	1	1	1	1	1

Table 3: Weightings found by hill-climbing (h-c) experiments described in Section 4.2. Feature descriptions are given in Table 1.

selected points from the parameter search carried out by our hill-climbing algorithm, namely the best and median settings visited. We also show an evenly-weighted baseline, as well as the scores for the official test run submitted in the given year's DUC competition for our system and the best-scoring peer.

- For DUC05, our best learned setting outperforms both the baseline and our original system's scores on the test set using ROUGE's built-in 95 percent intervals, and confirmed using the sign test ($P < 0.05$). Given that our system had the fourth-highest ROUGE-SU4 Recall score among the 32 scored systems in DUC05, this significant imrpovement is important. Versus the best peer from DUC05, we do slightly better on the training set and slightly worse on the test, but there is no significant difference in either case.

- For DUC04, our best learned setting outperforms our original system and the baseline on the test set scores using both ROUGE and sign test ($P < 0.05$). (Our original system was the best-performing peer in DUC04.)

- Table 3 shows the learned weights for the hill-climbing settings whose scores are shown in Table 2. While the hill-climbing algorithm we used for learning the weights purposely did not explore the entire feature space, we nonetheless find the differences in the "best" settings found for DUC04 versus DUC05 to be of interest. For instance, the Query feature appears more significant for the broad queries of DUC05 than for DUC04, where it provides more specific information about relevance.

  Another difference is that Centroid and Coverage features dominate the DUC04 best settings. We theorize this to be due also to shorter summary length, since in shorter summaries, the Lexical/DocumentCohesion features can cause too much summary content to be devoted to a relatively narrow sub-topic. In a limited experiment with longer-length summaries for DUC04 document sets, we indeed found that these coherence-based features appeared to be more useful.

- While the training and test scores vary, their relative order is mostly constant, indicating that overfitting is not a critical issue.

4.2.2 Sentence Reordering

After performing these experiments, we could see that the learned parameters achieved a high level of performance according to the ROUGE automated evaluation. However, after a manual examination of some output summaries, we found that the ordering of content within the summaries, an aspect not evaluated by ROUGE, was not always optimal. We thus decided to implement a post-sentence-selection step for cohesion improvement in order to (a) achieve more cohesive summaries and better scores in this year's DUC and (b) make sure that, before running our RSR-based experiments, summary cohesion would be in as strong a state as summary content selection.

To this end, we implemented an algorithm that examines the set of selected, ordered sentences produced by the iterative calls to ChooseNextSentence(), and looks for opportunities to reorder which will increase coherence. While the DocumentCoherence and LexicalCoherence features, when used during ChooseNextSentence(), can enhance local coherence of successive sentence pairs, they are also being weighted with other features which considers other aspects of content "goodness." In some sense, in the ChooseNextSentence() content selection phase, the role of the Lexical/DocumentCoherence features is to inform the system about what sentences are "close" to sentences which are already good, on the intuition that this can itself be a clue to content "goodness." By contrast, when our reordering algorithm is called, content has already been selected, and these features can be used used purely to determine if, within the selected sentences, we can achieve a better order to increase coherence.

The reordering algorithm takes as input the N sentences selected by ChooseNextSentence() in the order of their selection (which is equivalent to the final summary which was being produced by the system during our hill-climbing experiments, modulo possible truncation of the final sentence to conform to DUC word-length limit). The algorithm then proceeds, for sentences 2 through N, to move a sentence "upward" in the summary order iff:

- Overall coherence of the newly adjacent sentences is increased, where coherence is measured by the DocumentCoherence and LexicalCoherence features. We experimented briefly with weightings which are used for these features here, and found DocumentCoherence should be considered in preference to LexicalCoherence, which is intuitive given that a human author can be assumed to have written coherently, whereas lexical similarity is a weaker heuristic. (Note

Setting	Kendall's Tau
DUC05 best h-c, no reorder	0.069
DUC05 best h-c, reorder	0.079
DUC04 best h-c, no reorder	0.228
DUC04 best h-c, reorder	0.242

Table 4: Kendall Tau scores for non-reordered versus reordered summaries.

that these features are considered separately in the reordering algorithm than in sentence selection, i.e. the weights used in sentence slection have no effect here.)

- Ordering heuristics are maintained (for example, extremely short sentences or sentences starting with a discourse connective like "but" can only appear in the summary coupled with the sentence they appeared with in their source document).

In order to verify that this reordering algorithm was indeed improving our output summary order, we carried out the following experiment: For each training set, we prepared four augmented versions of its input document set, containing an additional "document" which is actually a model summary for that set. Then, we implemented a constrained mode of DefScriber which would be restricted to output only sentences from the model summary, but without using any information about the original order of its sentences. In this way, we were able to ensure that our system would be tested not on content selection (which would trivially be ideal), but rather on its ordering of these sentences from the model. As the DocumentCoherence feature is disabled here, the ordering is determined by Centroid ordering (recall that the entire document set is input and only the ouput is constrained, so that the Centroid will reflect sentences both from the model and the other documents), Coverage, LexicalCohesion and Query features.

Using this experimental setup, our goal was to compare the outputs from the system with and without reordering. The evaluation metric we chose for this task was Kendall's Tau measure. In addition to having been used in other document-ordering evaluations [13, 2], a desirable property of this measure is its ease of interpretation. Values range from -1 to 1, reflecting inverse to perfect ordering, respectively. Interpretively, one can view a score as proportional to the probability that a given pair of sentences within a summary is ordered as in the original model; a score of 0, for instance, reflects an even probability that any pair will be in correct versus inverse order; 0.5 means that 75 percent of the sentence pairs are correctly ordered with respect to the model.

Table 4 shows the results of this experiment with post-selection reordering. We compare mean Tau scores of our best-performing weighting (as learned in the previous experiment) with and without the reordering component, and observe that in both cases, the mean score increases. Carrying out a one-way ANOVA, we find that the effect of the reordering is significant at $P < 0.05$ on the DUC04 data, but not the DUC05. Here, we show results for the test set only.

4.3 Experiments with RSR Features

4.3.1 RSR Feature Overview

After running the above learning experiments to improve and optimize DefScriber, we were curious to see if features derived from our Rhetorical-Semantic Relation models for Cause, Contrast and/or Coherence could further improve either the content or coherence of our summaries.

In particular, we decided to implement the classifier alluded to in Section 3 as a feature. The concept here is that, when assessing sentence "goodness," either for content or coherence, we had already seen lexical similarity to be a useful feature; here, our goal was to experiment with the usefulness of a feature which measures whether a candidate sentence s relates to an already-chosen sentence p in a way that resembles, e.g., our Cause, Contrast or Adjacent models. In brief, this classifier allows us to estimate this by providing probability estimates for whether sentences appear to be related, e.g., by Cause ($P(p, s|Cause)$) or more closely resemble a baseline model ("NoRelSame") of sentences which are from the same document but bear no particular discourse relation ($P(p, s|NoRelSame)$). We then use the difference in these estimated probabilities to compute a normalized value from zero to one which expresses how Cause, Contrast or Adjacent-like a given sentence pair is.

In order to experiment with these features, we considered several ways to compute the RSR feature from Table 1. In particular, we experimented with a hill-climbing methodology similar to the previous section, but one in which the weights for the other features were set at their best-learned values, while the parameters being optimized were all with regard to the usage of the RSR-feature, namely:

RSR weight Parameter used to weight overall RSR feature as calculated according to the remaining parameters when combining with five non-RSR parameters (in sentence selection) or two cohesion parameters (in sentence reordering)

Cause, Contrast, Adjacent weights Three separate parameters which multiply the base zero-to-one value calculated by comparing ($P(p, s|R_k)$ and $P(p, s|NoRelSame)$ where R_k is one of the three RSR relations. We use separate parameters to enable learning how to interpret the probabilities of each model independently.

Combine mode Parameter which determines how to combine the three individual Cause, Contrast and Adjacent scores; can take on two settings, to either take the mean or maximum of scores; meant to learn relative importance of a single RSR model being very strongly matched, versus several RSR models all having a medium-strength match.

4.3.2 Experiments and Results

As in the non-RSR experiments, we ran an initial set of hill-climbing experiments focused on maximizing ROUGE scores. These results are summarized in Table5. It appears that adding the RSR features results in a small but positive increase on

Setting	Training	Test
DUC05, non-RSR	0.1351	0.1300
DUC05, best RSR h-c	0.1362	0.1326
DUC04, non-RSR	0.1233	0.1268
DUC04, best RSR h-c	0.1282	0.1273

Table 5: ROUGE SU-4 recall macro-averaged scores before/after integrating RSR features with best ROUGE-focused hill-climbing (h-c) weight.

Setting	Train	Test
DUC05 non-RSR	0.081	0.079
DUC05 best RSR h-c	0.084	0.087
DUC04 non-RSR	0.240	0.242
DUC04 best RSR h-c	0.267	0.259

Table 6: Mean Kendall's Tau scores with/without RSR-derived feature. Weighting of RSR-derived feature here uses learned settings from order-focused hill climbing.

ROUGE scores. However, in neither the DUC04 or DUC05 case does the improved score on the test set achieve a statistically significant improvement over the non-RSR results (in terms of intrinsic ROUGE confidence measures or the sign test). However, given that our scores were already nearly tied with (and not significantly below) the best peer from DUC05, and significantly better than the best peer from DUC04, we are pleased by the incremental improvement.

In a second set of experiments, we examine the effect of the RSR features on sentence ordering, using essentially the same model-constrained experimental setup described in the previous section. In addition to the effect on ordering which can result from the use of the RSR feature during original sentence selection, we also added the RSR feature to the LexicalCohesion and DocumentCohesion features considered in the reordering step.

As distinct from those features, the weightings used in sentence selection *do* carry over to the way the RSR feature is computed during reordering. This means that unlike in the ordering experiments in the previous section, which simply compare a single setting of the reordering ("on" versus "off"), in this case we conducted another hill-climbing experiment which attempted to maximimize the Kendall's Tau score. The reason for this was that in the previous experiment, we were comfortable manually estimating the weights used in reordering because of our clear understanding of the scores for LexicalCoherence and DocumentCoherence. However, in this case, the RSR probabilities returned by the different models for Cause, Contrast and Adjacent are not as well understood, and moreover they can be weighted and combined differently to compute the RSR feature. Thus, in this experiment, we perform a hill climbing which adjusts the RSR feature parameters with the goal of improving the Kendall's Tau score.

The results of this ordering-focused hill-climbing experiment are shown in Table 6; for brevity we only show the results at the best learned setting. Interestingly, the results are not only higher overall for DUC04, btut there is also a larger overall improvement on the test set when using the RSRs. We theorize that, at least in part, this may be due to the fact that biographical summaries, which describe a single entity and tend to follow a clearer more-to-less importance ordering, are closer to DefScriber's original purpose of creating descriptive definitions. In both DUC04 and DUC05 results, the effect does not rise to statistical significance; however, using a one-way ANOVA, the improvement for the DUC04 test set is significant at $P < 0.08$ for DUC04 data, so it approaches significance.

5 DUC 2006 Result Overview

As some of the experiments mentioned in the previous section were not yet complete at the time we produced our official DUC06 run, our system had not yet been fully optimized and refined. Nonetheless, the system as run for the test data included both of the main improvements mentioned in the previous section, including the reordering algorithm and best learned weights for the non-RSR features. In addition, we used an early implementation of the RSR feature, but since the RSR-specific experiments were not complete at that time our use of the RSR feature parameters was not entirely optimal.

As we focused our main effort on the above-mentioned experiments, which were predicated on the data sets for DUC04 and DUC05 available at that time, we have not yet performed an extensive analysis of this year's results. However, our preliminary examination shows our system performing above the median in all of the evaluated scoring categories, including manual and automatic measures. In particular, among 35 automatic peers evaluated, our scores were in the top 10 for ROUGE recall, mean linguistic quality, mean responsiveness.

Based on this initial analysis, we feel that our system remains a robust and competitive one in this task. While our new research did not vault us to the top of the rankings, we observe that many other participants appear to have been hard at work as well; for instance, our ROUGE scores this year would have been significantly "winners" last year. Thus, we are especially looking forward to hearing about the innovations from other participants.

6 Conclusion

For this year's DUC, we chose to follow an experimental course of research, with the hope of exploring an interesting new direction while at the same time improving our applied system. We are pleased because we were able to make significant accomplishments in both of these avenues.

In our work to integrate Rhetorical-Semantic Relation models, we took advantage of DUC to leverage our research on increasing the accuracy of such models. At the same time, we made a significant overhaul of our core DefScriber system to use a trainable, feature-based framework and sentence reordering algorithm. Using these enhancements, we achieved significant improvements in both content and ordering experiments using the features from our existing system. Then, in a second set of experiments, we added in the RSR-derived features and found additional improvements in both content and ordering scores. Finally, we are encouraged to see that this enhanced version of our

system continues to be strongly competitive among its peers, according to our preliminary analysis of this year's official results.

7 Acknowledgments

We would like to acknowledge the support of this work by the ARDA AQUAINT program (contract MDA908-02-C-0008). In addition, we are thankful for the support of our colleagues in the Natural Language Processing group at Columbia University.

We would also like to emphasize our gratitude to NIST and the DUC community for creating and making available its rich data sets, without which the experiments described in this paper would not have been possible.

References

[1] N. Asher and A. Lascarides. *The logic of conversation*. Cambridge University Press, 2003.

[2] Regina Barzilay and Lillian Lee. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *NAACL-HLT*, 2004.

[3] Sasha Blair-Goldensohn. From definitions to complex topics: Columbia university at DUC 2005. In *5th Document Understanding Conference (DUC 2004) at HLT/NAACL 2004*, 2005.

[4] Sasha Blair-Goldensohn, David Evans, Vasileios Hatzivassiloglou, Kathleen McKeown, Ani Nenkova, Rebecca Passonneau, Barry Schiffman, Andrew Schlaikjer, Advaith Siddharthan, and Sergey Siegelman. Columbia University at DUC 2004. In *4th Document Understanding Conference (DUC 2004) at HLT/NAACL 2004*, 2004.

[5] Sasha Blair-Goldensohn, Kathy McKeown, and Andrew Schlaikjer. Answering definitional questions: A hybrid approach. In Mark Maybury, editor, *New Directions In Question Answering*, chapter 4. AAAI Press, 2004.

[6] Jill Burstein, Daniel Marcu, Slava Andreyev, and Martin Chodorow. Towards automatic classification of discourse elements in essays. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-2001)*, 2001.

[7] Peter W. Foltz, Walter Kintsch, and Thomas K. Landauer. The measurement of textual coherence with latent semantic analysis. *Discourse Pricesses*, 25(2-3):285–307, 1998.

[8] B.J. Grosz and C.L. Sidner. Attention, intention and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.

[9] B. Hachey, G. Murray, and D. Reitter. The embra system at DUC 2005: Query-oriented multi-document summarization with a very large latent semantic space. In *Document Understanding Conference*, 2005.

[10] J.R. Hobbs. Coherence and coreference. *Cognitive Science*, 3(1):67–90, 1979.

[11] Eduard Hovy and Elisabeth Maier. Parsimonious or profligate: How many and which discourse structure relations? Unpublished Manuscript, 1993.

[12] J. Jagadeesh, P. Pingali, and V. Varma. A relevance-based language modeling approach to duc 2005. In *Document Understanding Conference*, 2005.

[13] M. Lapata. Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of ACL2003*, 2003.

[14] W.C. Mann and S.A. Thompson. Rhetorical structure theory: Towards a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[15] Daniel Marcu. *The Rhetorical Parsing, Summarization and Generation of Natural Language Texts*. PhD thesis, University of Toronto, Department of Computer Science, 1997.

[16] Daniel Marcu. The rhetorical parsing of unrestricted texts: A surface-based approach. *Computational Linguistics*, 26(3), 2000.

[17] Daniel Marcu and Abdessamad Echihabi. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, 2002.

[18] James Martin. *English Text: System and Structure*. John Benjamins, 1992.

[19] Kathleen R. McKeown. *Text generation: Using discourse strategies and focus constraints to generate natural language text*. Cambridge University Press, 1985.

[20] Tristan Miller. Latent semantic analysis and the construction of coherent extracts. In Galia Angelova, Kalina Bontcheva, Ruslan Mitkov, Nicolas Nicolov, and Nikolai Nikolov, editors, *Proceedings of the International Conference RANLP-2003 (Recent Advances in Natural Language Processing)*, pages 270–277, September 2003.

[21] J.D. Moore and C. Paris. Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics*, 19(4):651–695, 1993.

[22] Johanna D. Moore and Peter Wiemer-Hastings. Discourse in computational linguistics and artificial intelligence. In M. A. Gernbacher A. G. Graesser and S. R. Goldman, editors, *Handbook of Discourse Processes*, pages 439–487. Lawrence Erlbaum Associates, 2003.

[23] Megan G. Moser and Johanna D. Moore. Toward a synthesis of two accounts of discourse structure. *Computational Linguistics*, 22(3):409–420, 1996.