Detecting Discrepancies and Improving Intelligibility: Two Preliminary Evaluations of RIPTIDES

Michael White,[1] Claire Cardie,[2]
Vincent Ng,[2] Kiri Wagstaff,[2] and Daryl McCullough[1]

[1]CoGenTex, Inc.
840 Hanshaw Road
Ithaca, NY 14850, USA
mike,daryl@cogentex.com

[2]Department of Computer Science
Cornell University
Ithaca, NY 14850, USA
cardie,yung,wkiri@cs.cornell.edu

Abstract

We report on two preliminary evaluations of RIPTIDES, a system that combines information extraction (IE), extraction-based summarization, and natural language generation to support user-directed multidocument summarization. We report first on a case study of the system's ability to detect discrepancies in numerical estimates appearing in different news articles at different time points in the evolution of a story, using a corpus of more than 100 articles from multiple sources about an earthquake in Central America in January 2001. We then report on how our domain-independent, extraction-based summarizer fared on the DUC multidocument task, discussing the extent to which we were able to improve cohesion and organization over the baselines, without unduly sacrificing content relevance.

1 Introduction

We report on two preliminary evaluations of RIPTIDES, a prototype system that combines information extraction (IE), extraction-based summarization,

and natural language generation to support user-directed multidocument summarization.

The RIPTIDES system works as follows (cf. [13] for a more detailed system description). First, the system requires that the user select (1) a set of documents in which to search for information, and (2) one or more scenario templates (extraction domains) to activate. RIPTIDES next applies its Information Extraction subsystem to generate a database of extracted events for the selected domain and then invokes the Summarizer to generate a natural language summary of the extracted information subject to the user's constraints. The RIPTIDES system currently operates in the domain of natural disasters. Below we describe the IE system and Summarizer in turn.

The RIPTIDES system for the most part employs a traditional IE architecture [2]. In addition, we use an in-house implementation of the TIPSTER architecture [5] to manage all linguistic annotations. A preprocessor first finds sentences and tokens. Syntactic analysis is accomplished via the Charniak [3] parser, which creates Penn Treebank-style parses [7] rather than the partial parses used in most IE systems. Output from the parser is converted automatically into TIPSTER *parse* and *part-of-speech* annotations, which are added to the set of linguistic annotations for the document. BBN's Identifinder [1] locates dates, times, and other named entities.

The extraction phase of the system identifies domain-specific relations among relevant entities in the text using syntactico-semantic extraction patterns acquired via Autoslog-XML, an XSLT implementation of the weakly supervised Autoslog-TS pattern-learning system [10]. Unlike Autoslog-TS, however, Autoslog-XML proposes patterns for the extraction of constituents other than noun phrases. The current system, for example, also learns patterns to extract verb groups, adjectives, adverbs, and single-noun modifiers. Selectional restrictions on allowable slot fillers are implemented via WordNet-based [4] heuristics; heuristics also identify numeric modifying phrases in the extracted slot fillers. Finally, a simple clustering algorithm organizes the extracted concepts into output templates, which are provided as input to the summarization component along with all linguistic annotations accrued in the IE phase.

The Summarizer works in three main stages. In the first stage, the IE output templates are merged into an event-oriented structure where comparable facts are grouped. Towards the same objective, more surface-oriented clustering is used to group sentences from different documents into clusters that are likely to report similar content. To date we have experimented with both a simple word overlap clustering method as well as Columbia's

SimFinder tool [6]. In the second stage, importance scores are assigned to the slots/sentences, based on a combination of document position, document recency and group/cluster membership as well as further heuristics. In the third and final stage, the summary is generated from the resulting content pool using a combination of top-down, schema-like text building rules and surface-oriented revisions. The extracted sentences are simply listed in document order, grouped into blocks of adjacent sentences. The Summarizer is implemented using the Apache implementation of XSLT [8] and CoGenTex's Exemplars Framework [12].

Two unique aspects of the summarizer are its handling of numeric estimates and its method of selecting adjacent extracted sentences to improve intelligibility, without unduly sacrificing content relevance. Discrepancies in numeric estimates from different sources are highlighted, taking into account varying degrees of specificity (e.g. *thousands* vs. *3,000*). To improve the intelligibility of the extracted sentences, we have experimented with using a "coherence boost" to favor the inclusion of adjacent extracted sentences, especially when connected by an initial pronoun or a strong rhetorical marker (e.g. *however*), and using a randomized local search procedure to choose the highest ranked set of sentences.

To explore the current status of these two capabilities, we report first on a case study of the system's ability to detect discrepancies in numerical estimates appearing in different news articles at different time points in the evolution of a story, using a corpus of more than 100 articles from multiple sources about an earthquake in Central America in January 2001. We then report on how our domain-independent, extraction-based summarizer fared on the DUC multidocument task, discussing the extent to which we were able to improve cohesion and organization over the baselines, without unduly sacrificing content relevance.

2 Detecting Discrepancies: A Case Study

2.1 Motivation

In [9], Radev and McKeown provide anecdotal evidence of the need for multidocument summarizers to identify differences in the information that is reported by different news sources. However, from their article it remains unclear how often such differences actually represent significant discrepancies in the available information, vs. simple updates in what is known. Thus one is left to wonder to what extent one may find a complete and accurate picture of the available information by simply looking in the latest article.

Since this question has not been systematically investigated to our knowledge, we set out to perform a case study of the need to detect such discrepancies across news sources. For this study, we wanted to use a fairly complete set of news articles about an event, rather than a selected subset of articles as in the DUC and TDT document sets. Consequently, when two earthquakes took place in January, in Central America and India, we collected as many articles from the web as we could find (in a reasonably short period of time) during the first week after each quake, over 150 articles in each case. In perusing these collections, we ended up with the impression that trying to manually extricate the latest damage estimates at various time points from the various news sources would be very tedious. Although the estimates do usually converge, they often change rapidly at first, and then are gradually dropped from later articles, and thus simply looking at the latest article is often not satisfactory.

In the rest of this section, we examine some of the discrepancies in the Central America quake corpus in greater detail, along with the extent to which the RIPTIDES system is able to help identify these discrepancies.

2.2 Central America Quake Corpus

To simplify the automatic processing of this corpus, we wrote Perl scripts to extract the text of the article for five of the twelve available news sources, removing all the site navigation bars, banner ads, etc. This reduced the original set of 164 articles to 132 articles, from AP, Reuters, CNN, BBC and the Washington Post. During this process, we also sorted the articles according to their manually identified date and time of publication, normalized to EST, and assuming midnight for the Washington Post articles, which did not provide a time of publication. Additionally, for ease of browsing, we created a hyperlinked index to the articles, and concatenated the leading two paragraphs from all the articles into one file.

In looking at the leading two paragraphs of the articles, we[1] found significant variation in the facts reported, which suggests that there is considerable opportunity for a multidocument summarizer to surface key facts that may be missing from the beginning of the most recent article. In particular, in looking at the most frequently reported numerical damage estimates, we found that while the death toll is mentioned in the first two paragraphs in 72% of the articles (95/132, or 72%), the number missing appears in only 41% of the leading two paragraphs (54/132), and the number injured in only

[1]The first author determined the numbers reported in this subsection.

10% (13/132). In contrast, when we examined the whole articles, we found that the death toll was mentioned in about 92% of the articles, and both the number missing and injured in around 60%.

To get a more detailed picture of the discrepancies across news sources, we then examined the articles from the first four days after the quake to see how often the most recent article gave a significantly different picture of the death toll than one would get from reading all the articles up to that point. In so doing, we found that 20% (22/107) of the articles failed to provide a complete and accurate picture of what was known about the death toll at the time.[2] Of these 22 articles, half consisted of articles that included no mention of the overall death toll, focusing instead solely on the progress of relief efforts or on reports from a specific locale.

As an example of the discrepancies we found, the second CNN article on the quake reported at least two dead, whereas the latest AP article (posted four minutes earlier) reported at least twelve dead; the first CNN article (from the previous hour) reported more than ten killed; and the latest BBC and Reuters articles reported at least seven and at least five, respectively. As for the number injured, the CNN article reported at least 10 injured, whereas the latest AP article quoted the Salvadoran President as stating that 100 were reported injured. The CNN article did however interview a journalist on the scene who reported that 100-200 people had been buried in a landslide in a suburb of the capital, and thus were presumably injured or killed. As another example, a BBC article on the second day reported a death toll of at least 80, which was consistent with the latest confirmed estimates from CNN and Reuters, but conflicted with another BBC article (posted one minute earlier) that gave an estimate of hundreds, as well as with the latest AP estimate of at least 122, and a quote from a police agency in the same article of 234.

2.3 System Specifics

For this evaluation, the IE system for natural disasters was trained on 23 texts from topic 89 of the TDT2 corpus, a set of newswires that describe the 1998 earthquake in Afganistan. More specifically, Autoslog-XML proposed over 1400 extraction patterns based on the topic 89 texts, of which 317 were accepted and labeled w.r.t. concept type (i.e. disaster-event-

[2]Note that in one case, we found a BBC article that advanced the death toll well ahead of the other news sources, so we counted it as the outlier; had we instead counted all of the subsequent articles that continued to report the previous consensus, our tally would have been considerably higher.

type, disaster-event-location, disaster-event-date, disaster-event-time, victim, damage-object, confidence, epicenter, magnitude, magnitude-confidence-marker, person, organization, group, relief-agent, reporting-agent, damage-outcome-object, outcome-victim).

Prior to the evaluation, we had begun the implementation of one IE system capability not typically included in existing IE systems. In particular, we had noted in our earlier investigations [13] that multi-document summarization would require that the IE system distinguish different reports or views of the same event from multiple sources. With this in mind, we proposed to handle within-document event merging as a clustering task: given a set of extractions and the context in which each occurred, the system partitions the extracted material into equivalence classes, one for each reporting agent in the text. To date, we have implemented only a few very simple heuristics to constrain the partitioning process (e.g. extracted material from the same sentence should be in the same event partition; each identified reporting agent can be associated with only one partition). Our research plans include the investigation of learning methods that could acquire automatically the hard and soft constraints that allow accurate event partitioning.

Since we considered the evaluation for this case study to be exploratory, we decided to do a quick trial run of the Summarizer on some of the IE System output for the Central America quake corpus before the evaluation. In running the Summarizer on a couple of subsets of the IE System output (not the same ones used in the evaluation), we found that the heuristics that had been developed with the TDT2 topic 89 corpus for determining when a damage report pertained to the event as a whole (vs. a report for a specific locale) were working quite poorly, perhaps because the quake's effects were spread across multiple countries in Central America. To help address this problem, we made use of a small handcrafted knowledge base of the locations mentioned in articles and their part-of relationships in heuristically determining whether a damage report was localized. While this adaptation improved the output, it did not help with cases where the location was missed, and did not begin to address other problems with merging.

Figure 1 shows a sample summary generated by RIPTIDES, for all the articles up to the second CNN article discussed above. Note that this summary only includes text generated from the IE templates in the output; we did not include extracted sentences in the summaries for this evaluation, since we were focusing on the success of discrepancy detection. This summary correctly identifies the extremes of the overall death toll reports up to that point (primarily in El Salvador), and also includes an accurate re-

Earthquake strikes Central America

A major earthquake struck Central America Saturday, January 13, 2001. The earthquake had a magnitude of 7.6 on the Richter scale.

Damage

Estimates of the death toll varied. Associated Press (ap-20010113-1717) provided the highest estimate of at least 12 dead, whereas CNN (cnn-20010113-1636) gave the lowest estimate of two dead. Associated Press (ap-20010113-1717) and CNN (cnn-20010113-1636) reported that hundreds of rescuers were injured. Associated Press (ap-20010113-1717) reported that, a 41-year-old woman was injured. Power was without power.

Windows were destroyed. A centuries-old church was damaged. A pair of homes were destroyed. Hundreds of houses were destroyed. The scene ripping at the earth was damaged.

In Jalpataua, a man and a 2-year-old girl was killed, while three other people were injured. A pair of homes were destroyed.

In San Salvador, scores of homes were damaged.

Figure 1: Sample RIPTIDES Summary

	Death Toll	
	Completeness/Accuracy	Contains Missing Info
Articles	1.75	—
RIPTIDES	1.65	65%

	Num. Injured	
	Completeness/Accuracy	Contains Missing Info
Articles	2.05	—
RIPTIDES	2.35	60%

Table 1: Results of Detecting Discrepancies Evaluation

port of two killed in Jalpataua (Guatemala), and consequently provides a more complete and accurate picture of the overall death toll than the CNN article; though we may note that it is not entirely complete, since it fails to include the intermediate Reuters and BBC estimates. In contrast, this summary fares poorly on the number injured, since the rescuers mentioned were incorrectly identified as the ones injured. It also includes some clearly erroneous statements such as *Power was without power.*

2.4 Evaluation Method and Results

To determine the inputs for the evaluation, we selected 10 of the first 22 articles that failed to completely and accurately report the overall death toll. For each article, we then ran the RIPTIDES system on the articles up to and including that article, producing summaries of 200 words or less. Next we had two judges[3] rate each selected article and its corresponding summary on the completeness and accuracy of its reporting of both the overall death toll and the overall number injured. The ratings were given on a four point scale, where 1 = 'not at all,' 2 = 'somewhat,' 3 = 'mostly,' and 4 = 'entirely.' (Note that given the way the articles were selected, none received a 4 for its death toll reporting.) For each article/summary pair, and for both the death toll and the number injured, each judge also determined whether the summary contained some useful information that was missing from the article.

Table 1 shows the results averaged across the two judges. Although

[3]The first two authors were the judges.

RIPTIDES did a slightly better job on average than the full text of the selected news articles on completely and accurately reporting the available information about the number injured, it did slightly worse on reporting the death toll. Given that these articles were selected because they were at least somewhat inaccurate or incomplete in their reporting of the death toll, it is clear that the RIPTIDES system in its current state of development cannot be said to reliably detect discrepancies in numerical estimates. While we were disappointed with this negative result, we do consider it interesting to find that detecting such discrepancies is too difficult a problem to be solved using simple heuristics and techniques for merging extracted information within and across documents. At the same time, we were pleased to find that RIPTIDES did include some useful missing information about the death toll and number injured about 60% of the time; in an interactive interface, this capability could prove quite useful. We will elaborate on both of these points in the next subsection. Finally, one should bear in mind that although RIPTIDES is not yet capable of reliably identifying discrepancies, its summaries do surface important facts (such as the number injured) that are often buried in or missing from many articles, while also helping to address the problem of massive redundancy one encounters when looking at a series of articles on the same event.

2.5 Lessons Learned

One main lesson learned from this case study is that the inherent difficulty of accurately merging extracted information within and across documents is the primary obstacle to identifying discrepancies in numerical estimates in summaries generated from IE templates. To identify when two articles contain such conflicting information, one must determine when two different estimates are actually comparable, as well as whether the reports are sufficiently current to be considered in conflict.

In the natural disasters domain, we have identified the following factors that should be taken into consideration when determining whether two estimates are indeed comparable: (i) whether the estimates are for the current event, vs. a related event such as a previous quake in the same place; (ii) whether the estimates pertain to the consequences of the event as a whole, vs. the consequences in a particular locale; (iii) whether the estimates are for the main event in question, vs. a sub-event such as an ensuing landslide; (iv) the confidence of the estimates, e.g. confirmed vs. projected; (v) the source of the estimates, when attributed to specific persons. In this case study, the failure to accurately identify related events and localized reports

caused the most problems with inaccurate estimates showing up in the generated summaries. In ongoing work, as we improve our clustering approach to event coreference and extend it to the multidocument setting, we are hopeful that improvements in merging will ultimately enable the reliable detection of discrepancies in numerical estimates.

In order to determine which reported estimates are current, we have observed that simply taking the latest estimate from each source provides a good starting point, though we did observe multiple cases where a later article from the same news agency contained less current information than an earlier one. There is also the problem of multiple estimates appearing in the same article, which often occurs when a rough estimate (e.g. *hundreds*) precedes a more precise one (e.g. *at least 200*), or when the article itself reports different estimates with different confidence levels or source attributions.

The heuristics we used for this study to determine the current estimates were (i) consider a later report from the same source to supercede an earlier one when it is at least as specific or higher; and (ii) in a single article, consider an earlier report to supercede a later one of the same specificity, irrespective of confidence level or attributed source. While it is clear that these heuristics are not perfect in light of the subtleties mentioned above, the biggest problem with these heuristics is that they interacted poorly with our current simplistic approach to merging. For example, there were several cases where the IE system did find the latest AP estimate of the death toll in one article, but since it also found an estimate of the death toll of the 1986 El Salvador quake without identifying it as such, the latter (incorrect) estimate was taken to supercede the former (correct) one.

This observation leads to the second main lesson learned, namely that it would make more sense for the Summarizer to conveniently present all available estimates to the reader, so that the reader may easily make his or her own judgements, rather than just attempting to distill the estimates down to a sentence or two. One way to do this might be to present all comparable estimates in a chart or table, so that the reader can see the consensus as well as outliers. The Summarizer should also include hyperlinks to the original article, so that the reader can easily check outliers for accuracy.

3 Improving Intelligibility in the DUC Multidocument Task

3.1 Motivation

In multidocument summarization, extractive systems often cluster similar sentences across documents, and then use clusters to both (i) help find sentences that are important in the document set as a whole, and (ii) help reduce redundancy in the summary by penalizing the inclusion of more than one sentence per cluster. A question that arises in this context is whether some cluster representatives might fit in with the context of the rest of the summary better than others. If so, certain sets of cluster representatives would yield more intelligible summaries than others, in which case one might expect that a context-sensitive selection process for cluster representatives could improve intelligibility without greatly affecting informativeness.

To explore this hypothesis, we have experimented with using a "coherence boost" to favor the inclusion of adjacent extracted sentences, especially when connected by an initial pronoun or a strong rhetorical marker (e.g. *however*), and using a randomized local search procedure to choose the highest ranked set of sentences. Since our informal observations suggested that it may be possible to improve intelligibility in this way, we decided to try this approach with our DUC multidocument summarization system.

3.2 System Description

For our DUC multidocument summarization experiment, we began with a simple scoring model that strikes a balance between the latest document baseline and the first sentence from each document baseline, namely a weighted sum of the document recency and the within-document sentence position. To avoid overly favoring the later articles, we used a small weight for recency; and to keep the summaries focused on sentences early in the documents, we used the inverse of the within-document sentence position, so that the position score dropped off quickly as one went deeper into a document. Since we deemed clustering to be very important for the multidocument summarization task, we added to this simple model a moderate cluster size bonus and a sizeable cluster repetition penalty, using the clusters produced by Columbia's SimFinder tool [6].[4]

[4]We excluded clusters that appeared to be low quality based on a simple check of the similarity values.

To explore the context-sensitive selection of sentences, we then added to the scoring model a modest bonus for including adjacent sentences, a larger bonus when including a sentence that begins with an initial pronoun as well as the previous one, and a sizeable bonus when including a sentence that begins with a strong rhetorical marker (e.g. *however*) as well as its predecessor (where the latter two bonuses include the adjacency bonus). For this experiment, we used a set of about 70 connective phrases. Finally, we added a penalty for short sentences (under 12 tokens) when appearing without an adjacent one.

Given this scoring model, we used a simple stochastic search method, namely to employ local search from multiple random starting points (cf. [11]). For the first iteration, we began with the highest scoring sentences without cluster repetition, up to the word limit. For subsequent iterations, we randomly selected sentences, weighted according to their scores, up to the word limit. During each iteration, we greedily swapped sentences in and out of the summary until no more improvements could be made. More precisely, we repeatedly chose one sentence to add to the summary, and zero or more (typically one) sentences to remove from the summary, such that the word limit was still met, and this combination of sentences represented the best swap available according to the scoring model, until no such combination could be found.

An example sentence pair from the training corpus where the algorithm fared reasonably well appears below:

> Senate Democrats promised today to scrutinize Supreme Court nominee Clarence Thomas' views on abortion and other divisive issues. But Republicans said he should not divulge his feelings about controversies that might come before the court.

In this pair, the second sentence helps to provide a more balanced and complete the picture than the first sentence by itself. An example where the algorithm fared worse is as follows:

> Fire long ago destroyed the house where Clarence Thomas spent his boyhood. But nearby, down a woodsy, one-lane, white-sand road outside Savannah, Ga., sits a reminder of what might have been — the tired cottage where his sister still lives, by a broad, shining marsh called Moon River.

Here the second sentence is pulled in despite its marginal relevance, since the discourse marker *but* strongly connects it to the previous sentence, itself

All Summaries

	Cohesion	Organization	Mean Coverage
M	**2.18**	**2.40**	**0.55**
1	2.63	2.80	**0.37**
2	**1.72**	**1.65**	0.63 (n/s)
Sys	1.90	2.01	0.63 (n/s)
Hum	2.74	3.18	1.24

Table 2: DUC Multidocument Results

of marginal relevance. Conceivably, with a better underlying scoring model, the use of a "coherence boost" to improve intelligibility would yield fewer cases such as this one, where informativeness suffers.

3.3 Results

To judge the effectiveness of our approach, we looked at how the two DUC metrics related to intelligibility, cohesion and organization, varied in relation to mean coverage, the most agreed upon metric for content. The averages of these metrics across all multidocument summaries are shown in Table 2 for our system (M), baselines 1 and 2, all systems together (Sys), and all human authors (Hum).

We were pleased to find that our system substantially outperformed baseline 2 on cohesion and organization (2.18 and 2.40 vs. 1.72 and 1.65, resp.), while staying close to this baseline on content (0.55 vs. 0.63). As expected, we scored lower than baseline 1 on cohesion and organization, though considerably better on content (0.55 vs. 0.37). We were also pleasantly surprised to find that we scored well when compared to the average of all systems on cohesion and organization (2.18 and 2.40 vs. 1.90 and 2.01, resp.), though we did worse than average on content (0.55 vs. 0.63). Not surprisingly, we scored well below the human authors on all three metrics. Using a two-tailed t-test assuming unequal variances, we found that the differences between our system and the others were all significant at least at the 0.01 level, except for the difference between our system and baseline 2 on coverage ($p = 0.18$) and the difference between our system and all systems together on coverage ($p = 0.09$).

4 Concluding Remarks

In this paper, we have reported on two preliminary evaluations of RIPTIDES, a prototype system that combines information extraction (IE), extraction-based summarization, and natural language generation to support user-directed multidocument summarization.

In the first evaluation, the study of the Central American quake collection showed significant variation in the facts reported during an evolving disaster across various news sources, suggesting that there is ample opportunity for a multidocument summarizer to surface key facts that may be missing from the beginning of the most recent article. The study also found a considerable number of cases where the latest article failed to provide a complete and accurate picture of the available information, suggesting that there is a real need for automated support of detecting discrepancies across news sources. In evaluating the system's ability to identify such discrepancies in numeric estimates, we found that it was unable to do so reliably with the simple heuristics currently employed, though it was able to include some useful missing information about the death toll and number injured about 60% of the time. The evaluation pointed out a number of areas of improvement for the IE and summarization systems, and we are hopeful that our ongoing efforts will greatly enhance the system's usefulness.

In the second evaluation, we reported on an experiment with using a simple stochastic search procedure to improve the intelligibility of the summaries produced by our extraction-based summarizer component, without unduly sacrificing content relevance. Using scores from the DUC multidocument task, we found that we were able to substantially outperform the lead sentences baseline (baseline 2) on cohesion and organization, while staying close to this baseline on mean coverage. Our system also scored well when compared to the average of all systems on cohesion and organization, though we did slightly worse than average on mean coverage. In ongoing work, we are improving our underlying scoring model, in the hopes of being able to improve intelligibility while maintaining a reasonably high level of informativeness.

Acknowledgements

We thank Regina Barzilay for her help in using the SimFinder tool on the DUC corpus, and Tanya Korelsky for comments and discussion. This work has been partially supported by DARPA TIDES contract no. N66001-00-C-

8009.

References

[1] D. Bikel, S. Miller, R. Schwartz, and R. Weischedel. Nymble: A High-Performance Learning Name-Finder. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 194–201, San Francisco, CA, 1997. Morgan Kaufmann.

[2] C. Cardie. Empirical Methods in Information Extraction. *AI Magazine*, 18(4):65–79, 1997.

[3] Eugene Charniak. A maximum-entropy-inspired parser. Technical Report CS99-12, Brown University, 1999.

[4] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.

[5] R. Grishman. TIPSTER Architecture Design Document Version 2.2. Technical report, DARPA, 1996. Available at `http://www.tipster.org/`.

[6] Vasileios Hatzivassiloglou, Judith L. Klavans, Melissa L. Holcombe, Regina Barzilay, Min-Yen Kan, and Kathleen R. McKeown. Simfinder: A flexible clustering tool for summarization. In *Proceedings of the NAACL 2001 Workshop on Automatic Summarization*, Pittsburgh, PA, 2001.

[7] M. Marcus, M. Marcinkiewicz, and B. Santorini. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[8] The Apache XML Project. Xalan Java, 2001.

[9] Dragomir R. Radev and Kathleen R. McKeown. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500, 1988.

[10] E. Riloff. Automatically Generating Extraction Patterns from Untagged Text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1044–1049, Portland, OR, 1996. AAAI Press / MIT Press.

[11] Bart Selman and Henry Kautz. Noise Strategies for Improving Local Search. In *Proceedings of AAAI-94*, 1994.

[12] Michael White and Ted Caldwell. EXEMPLARS: A practical, extensible framework for dynamic text generation. In *Proceedings of the 9th International Workshop on Natural Language Generation*, pages 266–275, Niagara-on-the-Lake, Ontario, 1998.

[13] Michael White, Tanya Korelsky, Claire Cardie, Vincent Ng, David Pierce, and Kiri Wagstaff. Multidocument Summarization via Information Extraction. In *Proceedings of the First International Conference on Human Language Technology Research*, San Diego, CA, 2001.