GLEANS: A Generator of Logical Extracts and Abstracts for Nice Summaries

Hal Daumé III, Abdessamad Echihabi, Daniel Marcu, Dragos Stefan Munteanu, and Radu Soricut

Information Sciences Institute and Department of Computer Science
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA, 90292
{hdaume,echihabi,marcu,dragos,radu}@isi.edu

Abstract

We describe GLEANS, a summarization system that uses four novel techniques for summarizating document collections. (i) GLEANS first maps all documents in a collection into a canonical, database-like representation that makes explicit the main entities and relations in a document collection. (ii) GLEANS also classifies each document collection into one of four categories: collections about a single person, single event, multiple event, and natural disaster. (iii) For each type of document collection, GLEANS generates a short headline using a set of predefined templates. For multi-event documents, GLEANS also generates from scratch, using predefined templates, the first two sentences in the abstract. (iv) The rest of the summary is then generated by extracting from the database sentences that conform to a set of predefined schemas and by presenting them in an order that reflects coherence constraints specific to each collection category.

1 Motivation

As all the other entries in the DUC-2001 evaluation, our DUC-2001 multidocument summarization system produced extractive summaries by identifying important sentences in document collections (Marcu, 2001). Although our summarizer was among the best participating systems in DUC-2001, we felt that the sentence-based extraction techniques that it employed could not be easily extended to address a variety of summarization needs and challenges. For example, we could not see how we could easily modify our sentence-based summarizer to generate coherent abstracts. Also, we could not see how to generate extracts/abstracts tailored to different audiences, collection types, and genres. In summary, we came to realize that the extraction-based summarizer we built for DUC-2001 could not provide us with a scafold that would support an incremental research program in intelligent text summarization where gradual improvements of various components would lead to gradual improvements in summary quality.

To address these limitations, for DUC-2002, we decided to take a radically different approach. We started from scratch, and over a period of three weeks, we designed and implemented a complete multidocument summarization system that we believe has some of the ingredients that support incremental progress in intelligent text summarization. In this paper, we describe our system and assess its strengths and weaknesses.

2 Description of the system

The architecture of GLEANS, our DUC-2002 multidocument summarization system, is shown in Figure 1. In the rest of this section, we describe in detail its main components and focus on the modules that are novel.

Database constructor. The database constructor consists of the following modules:

- Contex[1], a decision-based syntactic parser (Hermjakob, 1997), is first used to parse all sentences in all documents.
- An extractor is then used to map each parse tree into a canonical representation that makes explicit the surface string that corresponds to the sentence, the logical subject, head verb, and main complement of the sentence.
- A set of specialized modules enhance the sentence database by making explicit the

[1] Contex uses the BBN named-entity tagger (Bikel et al., 1999) as preprocessor.

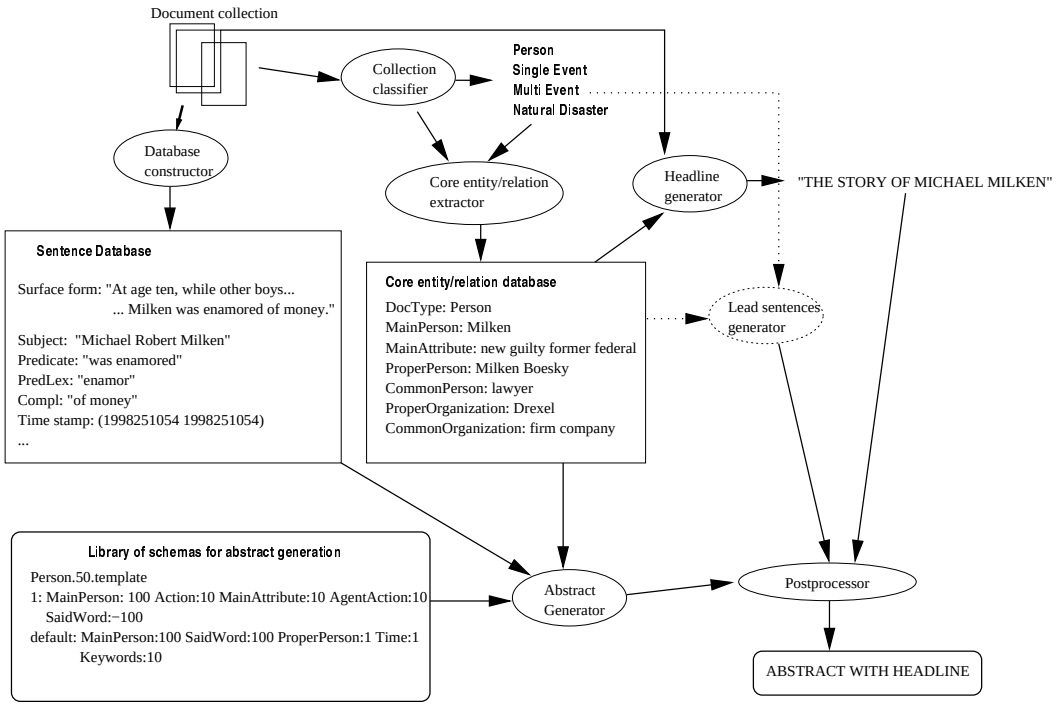

Figure 1: The architecture of GLEANS.

entities to which third person pronouns should be resolved and the time stamps of the main event of each sentence.

Collection classifier. We developed a four-way classifier in order to produce summaries that are customized to the type of document collection given as input. Each document collection is classified as being about a *person, single event, multiple event,* or *natural disaster*. The classifier uses hand-written rules that were developed by examining the collections in the training corpus. The rules use counts collected over the most frequent persons, places, and events that are described in an input document collection, as tagged by Contex. To reduce the noise specific to the counts, the rules are calibrated in the TF-IDF style by collecting counts on all the documents in all collections. On the DUC-2001 test corpus, this component accurately classified 28 of the 29 document collections.

Core entity/relation extractor. Given a document collection type from the classifier, the core entity/relation extractor generates a small entity/relation database that contains the most salient words in the collection. A typical entity/relation database that was extracted automatically for collection d02a, for example, is shown in Figure 1. For other collection types, the fields in the entity/relation database are different.

Headline generator. The label assigned by the classifier to a given collection and the core entity/relation database are used to generate a customized headline. For document collections of type "person", we generate a headline using the template "The story of *MainPerson*", where the *MainPerson* variable is replaced with information extracted from the core entity/relation database. For a "multi event" collection, we generate a headline using the template "*MainEvent*s in *Location1*, *Location2*, and *Location3*". For collection d02a, for example, we produce the headline "The story of Michael Milken". For collection d08b, which is assigned the type "multi event", we produce the headline "Eclipses in Hawaii, Mexico, and Bay Area".

On the DUC-2001 test corpus, this component produced 27 headlines that we subjectively judged to be correct (out of 29).

Lead sentences generator for multi-event collections. For multi event collections, we

```
3:  MainEvent:100 MainPlaces:100:-100:-100 SaidWord:-100 Time:1 Date:100 Number:100
4:  MainEvent:100 MainPlaces:-100:100:-100 SaidWord:-100 Time:1 Date:100 Number:100
5:  MainEvent:100 MainPlaces:-100:-100:100 SaidWord:-100 Time:1 Date:100 Number:100
6:  MainEvent:100 MainPlaces:-100:-100:-100 SaidWord:-100 AgentAction:100 Number:-100
      Date:-100 ProperPlace:-100 Other:100 StopDangling:1
default:  MainEvent:-50 MainPlaces:-100:-100:-100 SaidWord:-100 AgentAction:100
      Number:-100 Date:-100 ProperPlace:-100 Other:100 StopDangling:1
```

Figure 2: Template for generating a 200 word multi event summary.

also use templates to generate the first two sentences. The first sentence is generated using the template "A series of *MainEvent*s happened in *Location1*, *Location2*, and *Location3*, and other places between *Date1* and *Date2*. *MainEvent*s from *Date3*, *Date4*, ..., and *Daten* are also mentioned." As in headline generation, the variables in the template are instantiated from the core entity/relation database. On collection d08b, this template yields: "A series of eclipses happened in Hawaii, Mexico, Bay Area, and other places between Mar. 5, 1223 and July 11, 1991. Eclipses from 1868, 1948, 1970, 1979 , 1984 and 1991 are also mentioned." For space considerations, the second of the lead sentences is not produced when generating 50 word summaries.

Abstract generator. We analyzed manually all the abstracts in the training corpus and we produced a library of schemas for abstract generation. The schemas encode the canonical/desired structure and information content for a summary. Our schemas generalize the types of schemas introduced in the NLG field by McKeown (1985). Each line in a schema corresponds to a single sentence in the summary. The specification associated with each sentence consists of a combination of fields drawn from the core entity/relation database, each with corresponding weights. For each variable listed in a schema entry, if that variable can be matched against a sentence, that sentence gets weighted according to the corresponding feature weight.

For example, when generating 200 word multi-event summaries, the lead sentences are generated from scratch, as described above. The other sentences are extracted from the sentence database so as to obey the schema in Figure 2. After the first two sentences provide an overview of the multi event and provide the three main locations where the event took place, sentences 3 to 5 provide detailed information about the single events at each of the three locations (*MainEvent* has a high value, 100). Sentences are extracted from the database and presented in chronological order (Time:1) and are extracted so as to contain Date and Number information. Opinion sentences are not favoured (SaidWord:-100). Sentence 6 is extracted so as to focus on the most salient AgentAction pair in the core entity/relation database. The other sentences that can be produced without exceeding the 200 word threshold are selected so as to present Other entities and relations that are imporant in the collection but are not MainPlaces, Numbers, Dates, ProperPlaces, etc.).

In general, sentences are selected so as to avoid the generation of summaries that contain redundant information (Carbonell and Goldstein, 1998) and dangling discourse references (Marcu, 2001).

Postprocessor. A postprocessing module removes from the generated abstract the dangling discourse markers, decides when to use pronouns and when to use full named entities, infers specific dates from relative temporal expressions and document time stamps and represents them in a canonical fashion.

When given as input document collection d081, for example, GLEANS produces the 200-word summary in Figure 3. By inspecting the summary, one can easily assess the strengths and weakenesses of our system.

3 Evaluation of GLEANS on the DUC-2002 corpus

3.1 Internal evaluation

The collection classifier and the headline generator are the two components of our system that can be easily evaluated. The collection clas-

STRIKES IN UKRAINE, DONETSK, AND VORKUTA
A series of strikes happened in Ukraine, Donetsk, Vorkuta, and other places be tween Aug. 4, 1987 and June 29, 1994. Strikes from 1981, 1985, 1990 and 1991 a re also mentioned.
Coal miners in Siberia ended their strike on July 21, 1989 after exacting promises of better food, housing and working conditions, but the wave of unrest they launched continued in other key coal regions. Strikes by tens of thousands of miners in this Siberian coal region spread to four more of the Soviet Union's coal fields, Tass reported on July 20, 1989. Slyunkov held marathon meetings with strikers on July 16, 1989 night and on July 17, 1989 and addressed 30,000 people in the main square of Prokopyevsk, 2,100 miles east of Moscow. Port workers in Szczecin and southwestern coal miners on Aug. 17, 1988 declared sympathy strikes with miners demanding the legalization of Solidarity, opposition activists said. One United Mine Workers representative said on June 16, 1989 that the Indiana walkout was prompted by on June 14, 1989 night's telecast of CBS' "48 Hours," which focused on the Pittston strike, in which 1,600 miners have been off the job since April. Workers in the same trade union confederation as the miners have ignored appeals by their leaders to strike in solidarity.

Figure 3: The 200-word summary output of GLEANS for document collection d081.

sifier correctly labeled 43 of the 59 document collections in the DUC-2002 test corpus (72.8% accuracy) into one these four classes: *person, single event, multiple event,* and *natural disaster*. Most of the errors (10) were single event collections that were misclassified as multiple events. And six errors were person-type collections that were misclassified as event or multiple event. We were pleasantly surprised to see that a rule-based classifier can achieve 72.8% accurracy. This suggests that the collection classification problem is well defined. Machine learning-based approaches to this problem are likely to increase the performance of this component considerably.

We subjectively judged 25 of the 59 headlines generated by our system (42.3%) to be perfect both in terms of grammaticality and coverage. Some of the observed errors are attributable to the BBN tagger and the parser, which mislabeled various person names as places (e.g., Hugo and Bismark). Other errors are attributable to the system's lack of commonsense knowledge. The headline generator does not know, for example, that Guangdong is a province in China, that Texas is a United States state, and that San Francisco is part of the Bay Area, so it generates headlines that are semantically infelicitous, such as: "Flood affects Changsha, Guandong province and China"; "Droughts in United States, Texas, and Washington"; and "Earthquake affects San Francisco, Bay Area, and Charleston". In order to repair other headlines, one would require access to more esoteric knowledge. The headline "Floods in Ohio, Ohio River, and Wegee Creek" is incorrect because it is not reasonable to juxtapose a state and a river name; and a river and a creek name. Clearly, constructing and accessing semantic knowledge of this type can significantly improve the performance of our summarizer.

We believe that the following kind of error, which our headline generator makes, can be easily fixed. In our implementation, we relied entirely on unigram statistics in order to determine the concepts that are important in a text. This led to the generation of headlines like "Hamburgers in South Korea, Moscow, and Soviet Union" and "Dogs in Manhattan". If we used bigrams as well, we could have probably generated much better headlines instead, such as "McDonald's openings in South Korea, Moscow, and Soviet Union" and "Dog competitions in Manhattan".

3.2 NIST Evaluation

3.2.1 Coverage

The evaluation done by NIST resulted in the following mean coverage results: 0.14 for 10-word summaries (headlines), 0.10 for 50-word summaries, 0.12 for 100-word summaries, and 0.16 for 200-word summaries.

For our 10-word summaries, the best mean coverage, 0.40, was obtained for the eight collections on natural disasters that we labeled correctly. These scores compare favorably with the overall performance of the headline generators

that participated in DUC-2002.

In contrast, for the 11 collections that focused on a person, we only achieved a mean coverage of 0.09. The coverage results for the person-type collections came as a surprise to us, especially since in our internal evaluation we judged that all these headlines were correct. When we looked closer at the NIST evaluations, we found high discrepancies between various assessments. For example, our headline "The story of Prime Minister Margaret Thatcher" was judged as covering 40% of the reference headline "Britain's Iron Lady, Prime Minister Margaret Thatcher, resigns on Thursday", whereas our headline "The story of Emperor Hirohito" was judged as covering 0% of the reference headline "Emperor Hirohito, controversial World War II leader of Japan dies." We were extremely surprised to notice that 7 out of 11 headlines in this category received a 0% coverage score although our headlines correctly identified the person the document collection was about and sometimes other information-bearing modifiers.

For the 50, 100, and 200-word summaries we observed a similar pattern: the best mean coverage was obtained for the natural-disaster-type collections, whereas person-type collections were consistently assigned by NIST lower scores than our prediction. As in the headline case, our 50-word summary in Figure 4 received a 0% coverage score when judged against the reference summary, which is shown in the same figure, although the reference and automatically generated summary contain overlapping information.

3.2.2 Grammaticality, cohesion, and coherence

Because SEE, the evaluation interface used by NIST, does not enable one to distinguish between headlines and abstracts, we decided to render the headlines in our summaries using only upper-case letters, as shown in Figure 4. Also, in choosing the templates for our headlines, we decided to mimic to the best of our abilities the stylistic constraints specific to the headline genre. For example, we deliberately chose to not include verbs in the headlines. Both these decisions hurt significantly our scores on grammaticality, cohesion, and coherence. Almost all our summaries were judged as having capitalization errors (question 1 in the NIST evaluation) and incomplete sentences (question 4). However, different judges had different opinions. One judge consistently answered 0 to question 1 (deciding that our summaries had no capitalization errors), while others answered 1 or 2, and others consistently answered 3.

We found it ironic that our deliberate choice of rendering headlines in capital letters using a language specific to the headline genre "helped" us score worst or next-to-worst in many of the within-sentence judgments (incorrect word order, wrong subject-verb agreement, unrelated fragments joined in one sentence). We suspect that our headline generator is the main culprit because most of our summary sentences were picked from the original text and modified only for pronoun and temporal resolution (which had no adverse effect on the grammaticality of each individual sentence).

On the across-sentence judgments, we do best for the questions related to pronoun resolution. We have a fairly good score for question 7, pronouns with incorrect antecedents. On question 9, nouns that should have been replaced by a pronoun, we have the best cumulative score amongst all the systems. This suggests that our anaphora resolution module was sufficient for the summarization task.

We do badly on question 8 (nouns with no referent) and question 10 (dangling conjunctions). The latter is again surprising, since for all sentences beginning with a conjunction we also included the previous sentence in the summary, which generally solves the problem. Our internal evaluation found very few cases of dangling conjunctions in our submission.

The last two questions address issues concerning redundant information (question 11) and coherence (question 12) in the generated summary. Our cumulated information redundancy score is right on the average. The headlines hurt us here again, because the information from them is often repeated in the subsequent text. On coherence we rank about fifth out of seven systems. This suggests that the templates we induced manually were not sufficient for enforcing inter-sentential coherence.

Reference: Japan's Emperor Hirohito, renamed Showa after his Jan. 7 death, reigned for 62 years. He will be remembered for surrendering in WW II. Though mourned by many, Asians often viewed him as a symbol of Japanese aggression. A BBC documentary claims Truman helped Hirohito avoid a war crimes trial.

GLEANS summary: THE STORY OF EMPEROR HIROHITO
Tokyo (AP Emperor Hirohito will be remembered for one overwhelming decision during his 62-year reign: Japan must bow to defeat for the first time in its history and end the Pacific conflict in World War II.

Figure 4: Example of a 50-word summary produced by GLEANS, which was judged as covering 0% of the reference summary. The underlined segments pertain to content that is present in the automatically generated summary.

4 Single-document summarization

We did not do any work on single document summarization. We simply run our multidocument summarizer on each individual document in the test corpus.

5 Discussion

Overall, we are extremely pleased with the lessons we learned from our participation in the DUC evaluation. The performance of GLEANS was not high; but this is to be expected from a system designed to test new summarization and generation algorithms.

We believe that mapping the texts in a collection to one or more canonical structures (databases) enables one to determine more easily what entities are important in a collection and what relations hold between them. We initially thought that we could work with a database of clauses (not sentences), but generating well-formed sentences from arbitrary clauses and clause fragments proved to be too difficult to achieve. In order to generate summaries that are likely to be grammatical and coherent, we ended up saving sentences in our database, although we believe that sentences are too coarse for building high-quality summarization systems. We believe that a canonical database format of the kind shown in Figure 1 is more appropriate than a predefined IE-like template because it imposes much fewer restrictions with respect to the kinds of information one is constrained/expected to process.

We believe that significant progress can be made on the task of document collection classification and headline generation. Our current algorithms only prove that the tasks are well-defined and that reasonable results can be obtained with minimal effort.

We expect that libraries of schemata similar to that implemented by GLEANS are going to play an increasingly important role in the future. The generation of coherent texts is a task that needs to be given proper attention not only in summarization, but also in machine translation, and answer aggregation.

References

Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. 1999. An algorithm that learns what's in a name. *Machine Learning*, 34(1/3), February.

Jaime G. Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Research and Development in Information Retrieval*, pages 335–336.

Ulf Hermjakob. 1997. *Learning Parse and Translation Decisions From Examples With Rich Context.* Ph.D. thesis, University of Texas at Austin. Also published as Dept. of Computer Sciences TR 97-12.

Daniel Marcu. 2001. Discourse-based summarization in DUC-2001. In *Proceedings of the Document Understanding Conference*, pages 109–116, New Orleans, LA, Sept 13-14.

Kathleen R. McKeown. 1985. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text.* Cambridge University Press.