A Hybrid Approach for Answering Definitional Questions

Sasha Blair-Goldensohn and **Kathleen R. McKeown** and **Andrew Hazen Schlaikjer**
Department of Computer Science
Columbia University
New York, NY 10027
{sashabg,kathy,hazen}@cs.columbia.edu

Abstract

We present DefScriber, a fully implemented system that combines knowledge-based and statistical methods in forming multi-sentence answers to open-ended definitional questions of the form, "What is X?" We show how a set of definitional predicates proposed as the knowledge-based side of our approach can be used to guide the selection of definitional sentences. Finally, we present results of an evaluation of definitions generated by DefScriber from Internet documents.

1 Introduction

Question answering (QA) systems have reached a remarkably high level of performance (NIS, 2002) due to the integration of techniques from computational linguistics and information retrieval. Much of the effort in QA until now has gone into building *short answer* QA systems, which answer questions for which the correct answer is a single word or short phrase. Many questions are not in this class; they are better answered with a longer description or explanation. Producing these kinds of answers is the focus of *long-answer* QA, an area still in early stages of development but already the subject of several recent pilot studies (ARD, 2002).

Our work is concerned specifically with *definitional* QA - answering questions of the form, "What is X?" with multi-sentence responses which we provisionally call *definitional descriptions*. Definitional descriptions can be thought of as longer and more descriptive than dictionary definitions, while shorter than definitions found in an encyclopedia. DefScriber is a fully implemented system that generates these descriptions using an innovative combination of top-down and bottom-up techniques.

Top-down techniques in DefScriber are based on key elements of definitions as identified in the literature and in our own empirical study of definitions. One such element is information on the term's category (Genus) and/or important properties (Species) (Sager and L'Homme, 1994). For instance, category, or Genus, information about the term "Hajj" is given in the sentence "The Hajj is a type of ritual." DefScriber specifically searches for sentences that convey these definitional information types, or *predicates*, in building a definitional description.

Since relevant information for a given definition may not be entirely modeled by predicates, we complement our top-down approach with data-driven techniques adapted from work in multi-document summarization. These techniques take advantage of redundancy on the web to identify good definitional sentences. Using centroid-based metrics and clustering, DefScriber finds similarities in documents that focus on a given term and includes them in the response. These techniques allow us to include core information in the definition even when we don't have a specific predicate to model its semantic type.

Lastly, we give evaluation results which demonstrate the promise of this combined approach for generating definitions of ad hoc terms from a large and heterogenous document collection, the Internet.

2 Related Work

Our work on generation of definitions builds on research in summarization and in generation. Previous work in multi-document summarization has developed solutions that identify similarities across documents as the basis for summary content (Carbonell and Goldstein, 1998; Radev et al., 2000; Hovy and Lin, 1997; Mani and Bloedorn, 1997). Whether similarities are included through sentence extraction or information fusion (Barzilay et al., 1999), all of these approaches are *data-driven* because similarities in the data determine content.

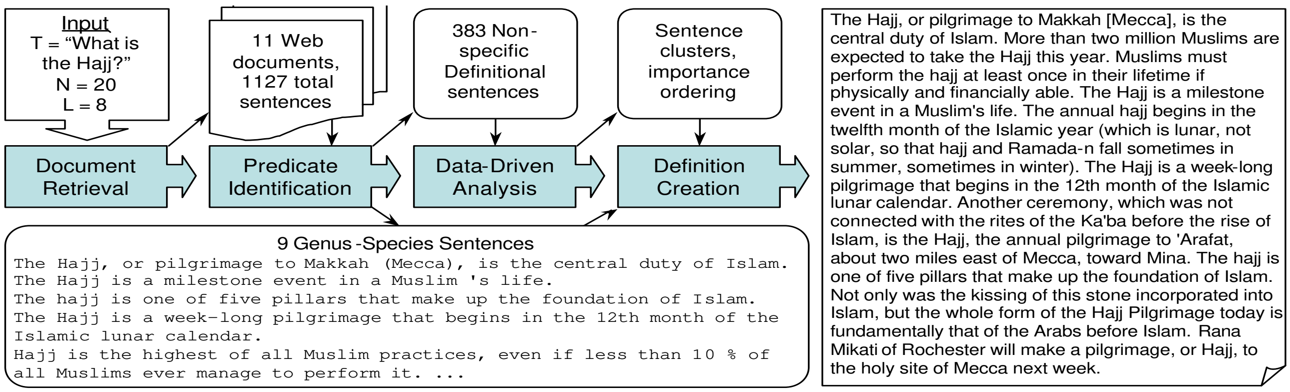

Figure 1: DefScriber creates a descriptive definition of the term "Hajj"

Top-down approaches are more often found in generation. Schemas (McKeown, 1985), rhetorical structure theory (Marcu, 1997; Moore and Paris, 1992) and plan-based approaches (Reiter and Dale, 2000) are examples of top-down approaches, where the schema or plan specifies the kind of information to include in a generated text. In early work, schemas were used to generate definitions (McKeown, 1985), but the information for the definitional text was found in a knowledge base. In more recent work, information extraction is used to create a top-down approach to summarization (Radev and McKeown, 1998) by searching for specific types of information which can be extracted from the input texts (e.g., perpetrator in a news article on terrorism). Here, the summary briefs the user on domain-specific information assumed *a priori* to be of interest.

Other long-answer QA approaches (ARD, 2002) are still in early stages and, for the most part, are very different from ours. For definition questions, many sites are using information extraction to find phrases providing specific types of content and presenting the answer in a formatted list.

3 DefScriber: Architecture Overview

Figure 1 shows the main stages of DefScriber's operation, and gives an example trace of input and output of each stage. This trace is of an actual answer generated for the question "What is the Hajj?" done as part of our evaluation (Section 6; this run used the "GS" configuration).

Input DefScriber's input is a triple (T, N, L) composed of a definitional question T, maximum number of documents to retrieve N, and output length L.

Document The information retrieval (IR) module uses a fixed set of patterns to identify the term in T and generate a set of queries in order of decreasing expected precision with respect to that term. Queries are sent in order to a web search engine until all have been sent or N URLs are retrieved.

Predicate Identification Documents returned by IR are analyzed for instances of the three definitional predicates implemented in DefScriber: Non-specific Definitional (NSD), Genus and Species. First, NSD sentences are identified, then the subset of these containing Genus and Species predicates in the same sentence are identified.

Data-Driven Analysis Techniques from summarization are used to cluster and order the entire set of NSD sentences based on properties of the data set as a whole. These data-driven techniques are designed to identify common themes in the data.

Definition Generation The output definition is generated by combining predicate information and data-driven analysis, ordering predicate sentences first.

4 Definitional Predicates: An Abstracting Top-Down Approach

Answering a "What is X?" definitional question and creating a summary of query results for the search term "X" are strongly related problems. Yet as readers, we have more specific expectations for a definition than for a general-use summary. What are these special properties of a definition, and how can we use them in creating descriptive definitions?

4.1 The Predicate Set

Our working set of predicates is shown in Table 1[1]. Currently, the system automatically identifies three of these predicates: Genus, Species and Non-specific Definitional (NSD). NSD is crucial because it is a cue to the presence of other predicates; it also removes noise and gives a set of useful information which can be presented using bottom-up methods even when it is not further classified. We choose Genus and Species as the first more specific predicates to implement because they are at the core of what definitions are: all related work identifies these two concepts as key parts of defining a term.

Previous work on definitions – from fields such as terminological theory (Sager and L'Homme, 1994) and philosophy (Swartz, 1997), and computational linguistics (Sarner and Cardberry, 1988; McKeown, 1985) – helped form the foundation for our predicate taxonomy. For instance, (Sarner and Cardberry, 1988) propose three "strategic predicates," including Identification and Properties predicates which are analogous to our Genus and Species, respectively. Although neither (Sager and L'Homme, 1994) nor (Swartz, 1997) posit an explicit predicate taxonomy, each theorizes that the type of information modeled by many of our predicates (including Genus, Species, Synonym and Target Partition) is crucial to descriptive-type definitions.

4.2 Identifying Definitional Predicates in Documents

To use these predicates in our system, we must identify sentences which contain them. We use two approaches: feature-based classification and pattern-recognition. Both approaches require a set of training data from which rules and/or patterns could be extracted; we therefore began by building a corpus of definitional texts annotated with predicates.

4.2.1 Document Markup

To produce the training data, coders marked 81 total documents for instances of the predicates in Table 1. The data included 55 documents marked by one coder, and 13 marked by two coders. To gather documents, 14 terms were first selected for broad coverage from several diverse categories: Geopolitical, Science, Health, and Miscellaneous. Then, we retrieved approximately 5 web documents for each term using a process similar to our system's IR component.

4.2.2 Rule Extraction: Statistical Techniques

Using the machine-learning tool Ripper (Cohen, 1995), we built a decision-tree model to predict the presence or absence of a given predicate on the sentence level. We select features for this model in part using observations from document markup. For instance, we include several features measuring a sentence's "term concentration", i.e. the term's frequency within the sentence and nearby sentences, based on the observation that appearance of the term is a good predictor of nearby relevant material. We also include features for relative and absolute position of a sentence in a document, based on the observation that useful information tends to concentrate toward the top of documents. Other features, such as presence of punctuation, are added to detect full-sentence text (as opposed to headings or other fragments), since most predicates other than NSD seem to occur mainly in full sentences. Some "blind" features such as bag-of-words are also used.

The Non-specific Definitional (NSD) predicate, which indicates a sentence's relevance to any aspect of defining the term, fares well using rules that consider term concentration and position in document. Using cross-validation, accuracy of 81 percent was obtained. This accuracy is sufficient for DefScriber since this predicate is not used to place sentences directly into the definition, but rather to pare down noisy and voluminous input by pulling out sentences which merit further examination.

In addition, the Historical predicate showed promise, achieving approximately 65 percent accuracy via this method, using similar features to NSD combined with a feature measuring occurance of four-digit numbers, i.e. years. In future work we will add a feature to model date-like strings more inclusively to improve accuracy in identifying the History predicate, so that it can be used by DefScriber.

[1] Note that in Table 1 and elsewhere we use the word "term" to mean not only the exact lexical term being defined, but any word/phrase that refers to the same referent. If a piece of text is a predicate instance for a given term, the same text replacing the term by a synonym is also an instance.

Predicate	Description	Instance Example
Genus	Conveys a category or set to which the term conceptually belongs.	The Hajj is a type of ritual.
Species	Describes properties of the term *other than OR in addition to* the category to which it belongs.	The annual hajj begins in the twelfth month of the Islamic year.
Synonym	Conveys a word or phrase which can be used as a synonym or abbreviation for the term. The text must not only use synonym but express its synonymy, using apposition, explicit "AKA", etc.	The Hajj, or Pilgrimage to Mecca, is the central duty of Islam.
Target Partition	Divides the term into two or more categories, conceptual and/or physical.	Qiran, Tamattu', and Ifrad are three different types of Hajj.
Cause	States that the term is the cause of something. The statement should be explicit. (e.g., "X appeared and Y disappeared." does not qualify as X causing anything.)	Doing a Hajj can cause all past sins of a muslim to be forgiven.
Effect	States explicitly that the term is caused by something.	The Hajj tradition was started to commemorate the sacrifice of the wife of Abraham.
Historical	Gives historical information about or strongly related to the term.	Mohammed, the founder of Islam, started the tradition of the Hajj in 632 C.E.
Etymology	Information on the term's genesis, from another language or adaptation via some other process, e.g. named after a person.	In Arabic, the word, Hajj means a resolve of magnificent duty.
Non-specific Definitional (NSD)	Text which contains information which would be relevant in a multi-page definition of the term. Any instance of another predicate is also an instance of NSD, but the opposite is not necessarily true.	Costs: Pilgrims may pay substantial tariffs to the occupiers of Makkah and the rulers of the lands they passed through...

Table 1: Definitional Predicates: Descriptions and Example Instances for the term "Hajj"

4.2.3 Rule Extraction: Syntactic and Lexical Patterns

Using lexicosyntactic patterns extracted from the document markup phase, we create a set of high-precision extraction patterns the two predicates most core to definitions: Genus and Species.

We model patterns to match sentences containing both Genus and Species information in the same sentence. These Genus-Species (G-S) sentences are often key to a strong definition because they provide both a context for the term as well as its key traits. In addition, the species information given in a G-S sentence is more likely than a standalone Species sentence to address core traits of the term, as it is being provided at the same time that the category of the term is given.

Rather than modeling the patterns at the word level, i.e. as templates with slots to fill, we model them as partially specified syntax trees. One such pattern can match a large class of semantically similar sentences without having to model every type of possible lexical variation. Information extraction (IE) has used similar techniques (Grishman, 1997) with partial *sub*trees for matching domain-specific concepts and named entities because automatic derivation of full parse trees is not always reliable. However, data-driven techniques (Section 5) offer additional protection from false or extraneous matches by lowering the importance ranking of information not corroborated elsewhere in the data.

For instance, in our "Hajj" example, the system matches the G-S sentence: "The Hajj was Muhammad's compromise with Arabian Paganism." This sentence is in principle a correct match, but the Genus and Species given here are extraneous and metaphorical. The fact that this information is less central to the definition is reflected by a low statistical "centrality" ordering for this sentence, and thus it is excluded from the definition.

Figure 2 illustrates the transformation from example sentence to pattern, and then shows a matching sentence. Our patterns are flexible – note that the example and matched sentences have somewhat different trees. Also, the example was extracted from a G-S sentence on "Hindu Kush," whereas it is used to detect a G-S sentence for the term "Hajj". Another point of flexibility is the verb phrase tree; in the pattern it contains *FormativeVb*, which stands for a list of verbs which our matching algorithm considers expressive of "belonging" to a category, i.e. indicating Genus (e.g., "be," "exemplify").

We extracted 18 distinct patterns which match G-S sentences. These 18 patterns provide sufficient recall to reliably find at least one instance in modestly sized document sets; over our evaluation test set (Sec-

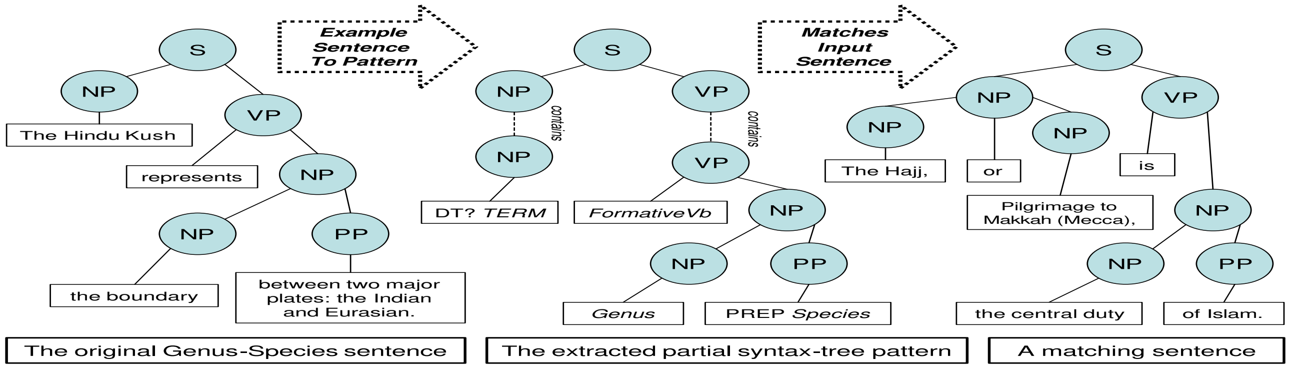

Figure 2: Pattern extraction and matching for a Genus-Species sentence from an example sentence.

tion 6), at least one G-S sentence was identified for 16 of 19 terms, with a mean of 3.5 G-S sentences per term (culled from a mean of 15 documents retrieved). Precision was 96 percent, recall unknown. Since we include only the top-ranking Genus-Species sentence in our output definition (see Section 5), this level of recall is satisfactory, particularly as precision is high.

5 Data-Driven Techniques: Applying Summarization

While our set of predicates, including Genus and Species, are domain-neutral, they are not meant to model all possible important information for a given term definition. Furthermore, some of the kind of information that we would like to include via predicates may be hard to define computationally *a priori*. For example, it is difficult to specify all ways in which Species information may be realized in a text. The sentence "Not only was the kissing of this stone incorporated into Islam, but the whole form of the Hajj Pilgrimage today is fundamentally that of the Arabs before Islam." includes Species information, yet it would be hard to build a general-purpose recognizer for Species that could identify this sentence.

However, we can identify these kind of sentences if we exploit redundancy that naturally occurs in large document collections, particularly the web. We do this by adapting statistical techniques from general purpose multi-document summarization to identify similar sentences across documents. To restrict retrieval of similar information to useful material, we apply these techniques only to material which is identified by the predicate analysis stage as Non-specific Definitional. This data-driven approach is especially important in early stages of system development, where implementation of specific predicates is limited to Genus-Species sentences, order to present a balanced overview of other kinds of information about the term being defined.

Our adaptation of summarization techniques involves a suite of methods. Centroid-based metrics allow us to find information that is central to the definition. Clustering allows us to avoid redundancy, separating out different aspects of central information. To suppress information that tends to appear in every sentence, we augment these metrics with a local IDF measure based on the input document set. Finally, we apply ordering techniques based on cohesion to order the response sentences.

A *definition centroid* is computed by creating a stemmed-word vector from all NSD sentences identified. Then the individual NSD sentences are sorted in order of decreasing "centrality," as approximated by IDF-weighted cosine distance from the definition centroid. This method creates a definition of length L by taking the first L non-identical sentences out of this sorted order. We call this the *TopN* and use it in our evaluation as a baseline. Note that this method approximates centroid-based summarization, and is a competitive baseline technique.

Clustering has been used in summarization (Hovy and Lin, 1997; Radev et al., 2000) to group similar sentences and thus reduce redundancy. We augment TopN with a non-heirarchical, sequential clustering method, using a similarity function of IDF-weighted cosine distance between candidate sentence and existing cluster centroid(s). The resulting clusters can be used to create a definition by taking the first sentence from each of the top L clusters. We call this the *SimpleCluster* method.

These methods sometimes result in overweighting of specialized terms. For instance, NSD sentences from documents retrieved for the “Hajj” example have a high occurence of the word “hajj” and strongly related terms like “Mecca” and “koran.” Since these words are quite rare in general corpora, their IDF values are very high and any two sentences which use them will have high similarity measures and likely cluster together, whereas in this context there are different aspects of Mecca that we want to cluster separately. To account for this, we augment the cosine distance calculation, selectively weighting with local IDF values calculated dynamically from the pool of NSD sentences. Specifically, the weight of terms whose ratio of Local to general IDF is above a threshold is reduced to the mean of these two values. We call this adjustment *LIDF* weighting.

Neither TopN nor SimpleCluster consider issues of sentence ordering to create a cohesive definition. DefScriber uses an ordering technique that takes into account the content of previous sentences in addition to statistical importance information like that derived in TopN. We pick the first sentence as in TopN or SimpleCluster, then pick sentences 2 through L as the first sentence from the cluster which maximizes an equal-weighted combination of overall importance to the definition (approximated by cosine distance between candidate and definitional centroid) and cohesion with the previous sentence (approximated by cosine distance between candidate sentence/cluster and previous sentence/cluster). We call this method *Principled Ordering*. Note that it can be applied to any of the previous methods.

DefScriber's default configuration integrates all the above techniques – SimpleCluster, LIDF weighting, and Principled Ordering – combines them with Genus-Species sentence identification. It places the top-ranking G-S sentence first in the definition, and uses the combination of data-driven techniques to add the remaining sentences in the definition. We call this integrated method *GS*.

6 Evaluation

Our evaluation used human judgments to measure the performance of DefScriber's definitions over a set of varied terms. By surveying users on definitions generated by different configurations of DefScriber, we are able to show how certain features of our system effect specific definitional features.

Category	Question
Structure	How would you rate the structure, or organization of the definition?
Relevance	Approximately how many sentences are relevant to describing or defining the term?
Coverage	How would you describe the breadth of coverage of the term by the information in the passage?
Redundancy	Approximately how many sentences are redundant with some other sentence(s) in the passage?
Term Understanding	How would you rate your overall understanding of the term after reading the passage?

Table 2: Questions used for the evaluation.

Source	Terms
Pilot Evaluation	asceticism, Aum Shinrikyo, battery, fibromyalgia, *gluons*, goth, Hajj, Mobilization for Global Justice, nanoparticles, religious right, *Shining Path*, Yahoo!
Hand-picked	autism, Booker Prize, Caspian Sea, East Timor, *hemophilia*, *MIRV*, orchid, pancreas, passive sonar, skin cancer, tachyons, *tsunami*

Table 3: Evaluation terms used for defscriber. Terms in *italics* were in the training set, while the rest were in the test set.

6.1 What To Measure

The recent pilot study on definitional QA as part of the Aquaint project ((ARD, 2002)) has sparked discussion between participants as to what constitutes an intrinsically “good” definition, with a significant level of convergence in opinion. Precision and recall are widely seen as main criteria. The more subjective element of “importance,” has also been raised, i.e. the idea that some pieces of information are more valuable in the definition than others. Minimizing redundancy is also a goal. Since many participant systems produce definitions as lists of properties ascribed to the term, issues such as structure and coherence are not of common concern to the group; our system produces a full-text definition so we wish to evaluate these features as well. We use a set of five questions (Table 2) to collect ratings for relevance (precision), redundancy, structure, breadth of coverage, and term understanding[2].

6.2 Evaluation Setup

We chose a set of 24 terms for which to run DefScriber (Table 3). We picked half of these ourselves, aiming

[2]To understand why coverage is not simply the opposite of redundancy, imagine a definition of Hajj that is a completely non-redundant history of Mecca.

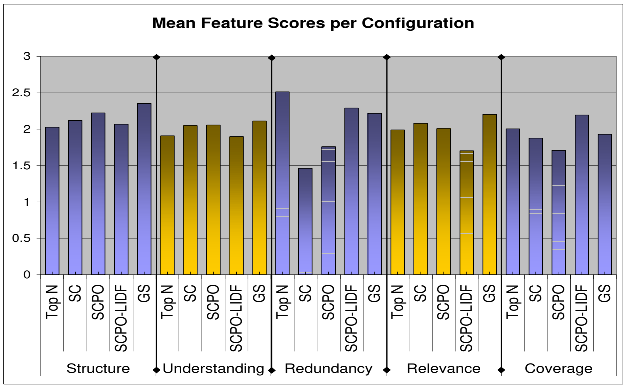

Figure 3: Evaluation Results

for varied domain coverage; the other half were randomly chosen from the definitional questions in the Aquaint pilot evaluation (ARD, 2002). From these 24 terms, five were used for training purposes, while the other 19 were used for testing (see Table 3). For each of the test terms, five separate descriptive definitions were generated using the following system configurations: TopN, SimpleCluster (*SC*), SC with Principled Ordering (*SC-PO*), PO using LIDF weighting for similarity measurements (*SCPO-LIDF*) and GS (see Section 5 for details of these configurations.) Other parameters of DefScriber were fixed: N, the maximum number of documents to retrieve, was 20. L, the desired summary length, was 8.

6.3 Results and Analysis

38 judges participated in the evaluation. Judges were asked to rate a sample of definitions for 10 different terms, so that each of the above configurations was used for 2 definitions in the sample. Some judges did not rate all definitions, so we average 16.6 (instead of 20) rated samples for each of the 95 definitions (19 test terms, 5 DefScriber configurations).

To normalize these slight differences in sample size, we use mean feature scores to consolidate rating data for each of the 95 definitions into a single independent data point. Figure 3 shows the resulting mean feature scores for each system configuration.

Since the rating scales used by the judges are ordered metrics (e.g.{Extremely Poor ... So-so ... Extremely Good}), we analyze the results with Ridit analysis (Fleiss, 1981). Ridit analyzes differences taking into account the natural ordering information in these type of ratings, i.e. the fact that "Extremely Good" and "Good" are both *above* "So-so", rather than simply being different categories. Using Ridit, we found that:

Structure The GS configuration outperformed all other methods with significance $P \leq .10$.

Term Understanding: GS achieved the best performance, but the margin is not statistically significant.

Redundancy All methods were significantly less redundant than TopN with $P \leq .10$.

Relevance GS has the best performance, i.e. the highest proportion of sentences rated relevant in the definition. However, the margin is not statistically significant. SCPO-LIDF does worst.

Coverage SCPO-LIDF does best, i.e. is rated with the broadest coverage, but the difference is not statistically significant.

GS is clearly the best overall configuration. This indicates that a leading Genus-Species sentence gives readers strong orientation with respect to the rest of the definition. GS's top scores in term understanding and relevance, although not statistically superior, also suggest that the leading Genus-Species sentence helps contextualize the other information in the definition, since the pure data-driven techniques in the other configurations may emphasize detail.

TopN is significantly more redundant than other methods; this indicates that these other methods, which use clustering, are successful in grouping together related concepts. SC has least redundancy, and we observe a tradeoff whereby the other methods, which implement ordering heuristics, appear to add back in some redundancy.

SCPO-LIDF has a better mean coverage rating than other methods, although the difference is not significant. This suggests that LIDF weighting improves coverage over different topics in the definition. However, it is puzzling that TopN achieves the second highest score here; perhaps TopN has the best chances of covering central ideas that judges reward with more coverage "points", whereas other methods may include less core information in trying to cover more ground.

Lastly, we note that in a time-limited task the significantly better structure of the GS definitions would likely have a magnification effect on other features like term understanding, given that a well-structured passage is easier to comprehend in limited time than an ill-structured one.

7 Future Work

A key area of future work is to increase the number of predicate types we can automatically identify. We would also like to improve precision and recall over currently identified types. Improving feature-based identification, like that used for NSD sentences, will involve some combination of marking up more data and modeling more features. For pattern-based identification, we plan to investigate the ability of bootstrapping techniques (Agichtein and Gravano, 2000) to mine more patterns.

Another area of research involves detecting polysemous terms early in the DefScriber pipeline, perhaps after document retrieval by using clustering techniques to see if documents "about" the term in question form two or more unrelated clusters. This issue might be resolved by creating a definition for each sense, or soliciting user clarification.

8 Conclusion

Generation of responses to open-ended questions from textual material available on the web will open up a new avenue for online research. In this paper, we presented and evaluated DefScriber, a system for generating definitions, which scored better than all other techniques tested on text structure, term understanding and relevance. Our approach includes a goal-driven method for screening definitional material from the large volume of non-definitional material returned from search, and for identifying sentences containing types of information typically found in definitions. It complements this approach with a method for exploiting redundancy on the web, using summarization techniques to identify similarities. This method will catch good definitional information that may be impossible to classify *a priori*, but which can be found if it occurs often in the data. While our ongoing work will incorporate additional predicates into DefScriber, this prototype demonstrates the power of our approach.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proc. of 5th ACM Int'l Conf. on Digital Libraries*.

ARDA and NIST. 2002. *Aquaint R&D Program 12 Month Workshop*, Arlington, VA.

Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proc. of 37th ACL*, pages 550–557.

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.

William W. Cohen. 1995. Fast effective rule induction. In *Proc. of 12th Int'l Conf. on Machine Learning*, pages 115–123.

Joseph L. Fleiss. 1981. *Statistical Methods for Rates and Proportions*. John Wiley & Sons, New York.

Ralph Grishman, 1997. *Information Extraction: Techniques and Challenges*. Springer-Verlag.

Eduard Hovy and C.Y. Lin. 1997. Automated text summarization in summarist. In *Proc. ACL '97 WS on Intelligent Scalable Text Summarization*, pages 18–24.

Inderjeet Mani and E. Bloedorn. 1997. Multi-document summarization by graph search and matching. In *Proc. 15th Nat'l Conf on AI*, pages 622–628.

Daniel Marcu. 1997. Rhetorical parsing of natural language texts. In *In Proc. ACL–EACL 97*, pages 96–103.

Kathleen R. McKeown. 1985. *Text Generation*. Cambridge Univ. Press.

Johanna D. Moore and Cecile L. Paris. 1992. Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Comp Ling*, 19(4):651–695.

NIST. 2002. *Text Retrieval Conference (TREC 02)*, Gaithersburg, MD.

Dragomir R. Radev and Kathleen R. McKeown. 1998. Generating natural language summaries from multiple on-line sources. *Comp Ling*, 24(3):469–500.

Dragomir R. Radev, H. Jing, and M. Budzikowska. 2000. Centroid-based summarization of multiple documents. In *ANLP–NAACL WS on Summarization*.

E. Reiter and R. Dale, 2000. *Building Natural Language Generation Systems*, chapter 4. Cambridge Univ. Press.

Juan C. Sager and M.C. L'Homme. 1994. A model for definition of concepts. *Terminology*, pages 351–374.

Margaret Sarner and Sandra Cardberry. 1988. A new strategy for providing definitions in task oriented dialogues. In *Proc. Int'l Conf. on Computational Linguistics (COLING-88)*, volume 1.

Norman Swartz. 1997. Definitions, dictionaries and meanings. Posted online http://www.sfu.ca/philosophy/definitn.htm.