Towards Generating Patient Specific Summaries of Medical Articles

Noemie Elhadad and Kathleen R. McKeown
Computer Science Department
Columbia University
New York, 10027, NY
{noemie, kathy}@cs.columbia.edu

Abstract

The end users of medical digital libraries need quick access to information that is specific to the patients under their care. We present a summarization system that finds and extracts results from multiple medical journal articles returned by a search, filters results that match the patient and merges and orders the remaining facts for the summary. Our approach features an integration of text categorization, information extraction, information fusion and text reformulation for the summarization task.

1 Introduction

As end users of medical digital libraries, clinicians have quite specific needs. They are concerned with patients under their care and are under extreme time pressure, with little time to browse. Thus, they need information at the point of patient care, that succinctly provides important facts relevant to their patient's problem. A summary containing just those facts from retrieved articles that pertain to the patient and problem under consideration allows the clinician to more quickly gain needed information. If the summary provides links to the articles from which the facts were drawn, the clinician can read the full article when needed. In this paper, we present a system we are developing to provide patient specific summaries over multiple journal articles returned in response to a query. Our system is part of PERSIVAL, a digital library project which provides tailored access for patients and physicians to a distributed patient care digital library (McKeown et al., 2001).

In earlier work, we carried out a feasibility study that showed that physicians were interested in the results from a journal article that pertained to the patient under question (McKeown et al., 1998). In order to generate a summary containing patient specific results, our system first extracts results from a journal article, then matches the journal article results against the findings represented in the patient record. It merges and orders the results extracted in all the input articles to produce summary content and finally, uses language generation to produce summary wording.

Our approach integrates a range of techniques that have been used separately in the past for summary generation. It uses a combination of text categorization and information extraction to locate article results in the Results section of the article. Since the extracted information consists of semantically typed, full phrases found in the article, summary generation uses a combination of traditional language generation with reformulation of text. Finally, we propose an internal representation of extracted phrases that facilitates merging of repetitive, contradictory, and related facts drawn from the different articles.

In the following sections, we first present a requirements analysis of summary generation in this domain; each requirement triggers the use of specific summarization techniques. After describing system input, we present an overview of the system and describe each module. We focus on the task of merging and ordering extracted information to produce a coherent summary.

2 Quality and Evaluation Criteria

We captured the main quality criteria on the summarization algorithm through interviews with physicians and analysis of our corpus of medical journal articles. To our knowledge there is no collection of existing summaries of multiple journal articles relating information specific to a patient. Furthermore, naturally occurring summaries of multiple documents in any domain are hard to find. The absence of a gold-standard makes the design and development of an algorithm for summarization more difficult. As a first step to design, we describe operational ways to measure the identified quality criteria. These criteria can also ultimately be used for evaluation as well.

Completeness and Accuracy When planning the treatment of a patient, physicians are typically interested in the results of different clinical studies that are relevant to the patient. Based on this observation, we consider that a good summary should contain results from clinical studies (as opposed to patient group descriptions, methods or discussion of the study). We define a result as a tuple $(Parameter(s), Finding, Relation)$ (Figure 1). We identified six types of relation between parameters and findings, namely *association, prediction, risk, absence of association, absence of prediction* and *absence of risk*.

Age > 65 years and prior angina were the only *univariate predictors* of **in-hospital mortality**.

Figure 1: Result example. The Parameters are underlined, the Finding is in bold and the Relation is in italic.

The summarization algorithm should be able to extract results that are relevant to the input patient. The summary should be complete, *i.e.* all relevant results from the input articles should be included in the summary. It should also be accurate, *i.e.* only relevant results should be included in the summary. Completeness and accuracy can be evaluated by computing the precision and recall of the results extraction on a random set of articles.

Identification of repetitions and contradictions If a result is reported in several studies, it is likely to be an important fact for the physician to know. Similarly, if a result is reported in one study but a contradictory result is also available, it suggests that the relation between parameters and findings is controversial and, therefore, worth knowing about. Our summarization algorithm should in consequence be able to identify repetitions and contradictions. To do so, we have to design a representation of results that allows us to define relations between different results such as subsumption and contradiction. Evaluation can then be performed by comparing the number of actual repetitions and contradictions in a set of random articles and the number of automatically identified repetitions and contradictions.

Coherence and Cohesion Identifying repetitions and contradictions is not enough to produce a summary of high quality. A summary with a listing of repetitions and/or contradictions would not be coherent. The summary should therefore be able to signal to the user the presence of repetition or contradiction.

The aspects of coherence of a text that we evaluate are measured as the accurate aggregation and ordering of related results. For instance suppose a summary includes three sentences: two about the predictors of "*new acute myocardial infarct*" and one sentence about the risks associated with being treated with "*amiodarone*". A coherent presentation will not intermingle the three sentences but rather group the sentences about "*myocardial infarct*" together.

We define cohesion for our task as the following: two sentences are part of the same paragraph if and only if they are related. By related, we mean that sentences should present either the same finding or the same parameter(s).

3 Input to the System

Input to the summarization system is composed of a preprocessed patient record, a set of medical journal articles and a user query.

The Patient Record The patient record contains the clinical history of the patient as well as reports on all the tests and procedures performed on the patient. Some reports (*e.g.* laboratory tests) are in tabular form while others (*e.g.* discharge or operation reports) contain non-structured text. Patient records can be very large because, over time, information is added and never deleted. Only a subset of the information present in the patient record is relevant for our task. Our input is a processed patient record (see Figure 2). In the PERSIVAL system, the processing (Mendoca et al., 2001) produces a set of identified terms and their value if applicable (*e.g.*, "*high (value) blood pressure (term)*"). This set of attribute-value pairs provides a representation of important patient parameters as a patient profile.

```
<concept id='C09253' lex='gender' val='female'/>
<concept id='C02555' lex='age' val='44'/>
<concept id='C18802' lex='congestive heart failure'/>
<concept id='C89482' lex='ejection fraction' val='30%'/>
```

Figure 2: Extract from a Preprocessed Patient Record.

The Journal Medical Articles We are interested in journal medical articles that contain information relevant to a specific patient. An article is said to globally match the patient if it contains such information. Knowing *a priori* that an article globally matches the patient allows the summarization system to only analyze the local context of results without having to perform full semantic analysis of the article.

In a preprocessing stage, only articles that globally match the input patient record are kept as input to the summarization system. In the PERSIVAL system, preprocessing is done by (Teufel et al., 2001). First, medical terms are identified in the article and are semantically tagged using the unique concept identifiers (called CUI) of the UMLS (NLM, 1995), a front-end to several large-scale medical knowledge bases. When a significant number of characteristics from the patient record are present in the patient study group described in an article, the article is considered globally matching.

The User Query The user, in our scenario a physician or a medical student, can ask a question relative to some specific characteristics of the patient record (Figure 3). The system's goal is not to provide the user with an answer to the user query, but rather to present a synthesis of any relevant result for a specific patient. Nevertheless, the query provides a good indication of what the user is primarily interested in when reading the summary.

What are the predictors for unstable angina?

Figure 3: A User Query example.

4 System Architecture and Implementation

Our system follows a pipeline architecture, shown in Figure 4. As described, the input to the system comprises a set of articles, a patient record and a user query.[1] Each input article is first classified according to its main clinical task (*i.e.* diagnosis, prognosis, or treatment). It is then passed to the Result Extraction component which builds a set of templates from the articles. The templates that are not relevant to the input patient record filtered out by the Patient Matching component. In the next phase, all the relevant templates are merged into a graph. Identification of repetitions and contradictions is performed, and heuristics are applied to the graph to determine in which order to present the information to the user. The Sentence Planner and Realization components transform the graph into an fluent English text.

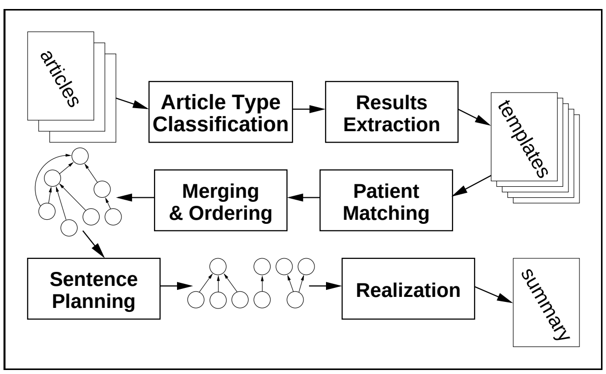

Figure 4: System Architecture.

4.1 Article Type Classification

Information conveyed in technical medical articles depends on the clinical task described. Following the medical literature, we classify articles into three classes: prognosis, treatment and diagnosis.
The main motivation behind this classification is that physicians have a natural tendency to distinguish results of the different types; the summary should therefore reflect this fact by separately presenting summaries about prognosis, treatment and diagnosis articles. We used text classification to classify articles, using a training corpus of 200 articles from the American Heart Journal. We use Rainbow's classifier and obtain 94% accuracy. For comparison, a baseline classification choosing the most common category yields 59% accuracy.

[1]The patient record and the user query are not represented in the figure for readability.

4.2 Results Extraction

Medical journal articles have a rigid format; an article typically has the following sections: Abstract, Introduction, Methods, Results and Discussion. It may also have other optional sections such as Statistics. Descriptions of study results are typically presented in the Abstract and Results sections. One could argue that the Result Extraction component should look only at the results in the Abstract section. This is not a good strategy because the criteria of relevance for our task differs from the ones of the authors of the studies described. A result is important if it matches with the input patient. Therefore, we are trying first to extract all possible results and focus on the Results section of the input articles.

However, not the entire Results section is of interest for extraction; after analyzing our corpus, we identified three kinds of information conveyed in this section (see Figure 5)– description of the patients at the beginning of the study (5(a)), outcome of the study (5(b)) and statistically significant results (5(c)). Because we present information gathered from several articles, it would be confusing for physicians to include descriptions of patients in different patient studies and their outcome. Only general results can be included (5(c)). This complies with our definition of a result, *i.e.* a tuple $(Parameter(s), Finding, Relation)$.

(a) The women in the population were significantly older than men. (b) Three patients died after a month. (c) Age > 65 years and prior angina were the only univariate predictors of in-hospital mortality.

Figure 5: Different kinds of information in the Results section of a medical journal article.

Our Result Extraction module takes as input sentences from the Results section. It operates in two phases; first non-result sentences are filtered out. Templates are then built from the result sentences using information extraction techniques.

Selection For some sentences in the Results section, it is obvious whether or not these are results; even medically naive humans can distinguish these. Examples of such sentences are "*Results of the multivariate analysis are shown in Table III*" or the sentence shown in Figure 5(c). Some other sentences are harder to classify into result/non result without medical knowledge and/or without taking the context surrounding the sentence into consideration. The sentence in Figure 5(a) is a comparison between two groups of patients and might erroneously be considered as a result sentence. Automatically differentiating between patient descriptions and result sentences is a hard task, because the authors of the articles use the same syntactic structure and similar medical terms when writing them. We have to use additional cues to differentiate between them, such as the sentence position in the section.

We implemented this as a classification task; given a sentence in the Results section, is it likely to be a result, according to our own definition or not? We manually annotated sentences of the Results section of 200 articles from the American Heart Journal. We identified two kinds of features: position of the sentence in the section (patient descriptions tend to occur first in the section) and lexical cues. The classification achieves 60% precision and 83% recall. This classification acts mainly as a filter, and therefore recall is more appropriate measure than precision for this task.

This filtering on the Results section of each article has two effects. First, it reduces the number of input sentences sent to the Extraction phase. In average, only a third of the Results section comprises result sentences. Second, it ensures that sentences passed to the Extraction phase are only results, and therefore extracted information will be meaningful. If a sentence of the kind population description (5(a)) is passed to the Extraction phase, information would be extracted because of its similarity to typical result sentences, but this information would be useless in a summary.

Extraction The main pieces of information we want to extract, given a result sentence, are the following: the finding(s), the parameters, the relation, the degree of dependence of the parameters, the article and the sentence it has been extracted from and some other minor informations. The extraction builds a template for each sentence (Figure 6). We are currently using a set of hand crafted patterns to extract templates. We constructed a set of 20 patterns, analyzing 30 articles from the American Heart Journal and the journal of American College of Cardiology. In a first evaluation on 27 test articles, we achieve a precision of 92%, and a recall of 50%. The recall is not satisfying, and we fear that when extending our corpus to a new set of articles, it will drop. To overcome this problem, we are investigating ways to collect extraction patterns using supervised learning. The main issue we have to address is that both the finding and the parameters expect the same semantic type of information.

```
<TEMPLATE ID='12'>
  <REL>association</REL>
  <ANA_TYPE>independent</ANA_TYPE>
  <FILE_NO>ahj_137_02_0424</FILE_NO>
  <SENT_NO>S-98</SENT_NO>
 <FINDING>
  <ITEM>
    <W C='JJ'>new</W>
    <TERM CUI='C55626'>acute myocardial infarction</TERM>
  </ITEM>
  <ITEM>
    <TERM CUI='C11065'>death</TERM>
  </ITEM>
 </FINDING>
 <POS_PARAM>
  <ITEM>
    <TERM CUI='C20887'>ST-segment depression</TERM>
    <PARX>(OR 2.00, 95% CI 1.20 to 3.40; P = .008)</PARX>
  </ITEM>
  <ITEM>
    <W C='JJ'>prior</W>
    <TERM CUI='C02962'>angina</TERM>
    <PARX>(OR 2.70, 95% CI 1.34 to 5.57; P = .001)</PARX>
  </ITEM>
  <ITEM>
    <TERM CUI='C02555'>age</TERM>  <W C='SYM'>></W>
    <W C='CD'>65</W>  <W C='UNIT'>years</W>
    <PARX>(OR 1.64, 95% CI 1.00 to 2.70; P = .01)</PARX>
  </ITEM>
 </POS_PARAM>
</TEMPLATE>
```

Figure 6: Template before Patient Matching.

```
<TEMPLATE ID='12'>
  <REL>association</REL>
  <ANA_TYPE>independent</ANA_TYPE>
  <FILE_NO>ahj_137_02_0424</FILE_NO>
  <SENT_NO>S-98</SENT_NO>
 <FINDING>
  <ITEM>
    <W C='JJ'>new</W>
    <TERM CUI='C55626'>acute myocardial infarction</TERM>
  </ITEM>
  <ITEM>
    <TERM CUI='C11065'>death</TERM>
  </ITEM>
 </FINDING>
 <POS_PARAM>
  <ITEM>
    <TERM CUI='C20887'>ST-segment depression</TERM>
    <PARX>(OR 2.00, 95% CI 1.20 to 3.40; P = .008)</PARX>
  </ITEM>
  <ITEM>
    <W C='JJ'>prior</W>
    <TERM CUI='C02962'>angina</TERM>
    <PARX>(OR 2.70, 95% CI 1.34 to 5.57; P = .001)</PARX>
  </ITEM>
 </POS_PARAM>
</TEMPLATE>
```

Figure 7: Template after Patient Matching.

4.3 Matching with the Patient

Next in the pipeline, the information extracted from the input articles is matched against the patient record. The Patient Matching component determines whether a given template is relevant to the patient record and should accordingly be included in the summary or not.

Considering the following result sentence from an article: "*In a multivariate analysis, chest pain and male gender were identified as the only independent predictors for unstable angina.*". We match it against a female patient with chest pains but no unstable angina diagnosed. From a medical point of view, some results are pertaining to the patient, namely, chest pain predicts independently unstable angina.

The result sentence contains the medical terms "*chest pain*", "*gender*" with value "*male*" and "*unstable angina*". One naive strategy for matching is to count the number of terms in the sentence that are in common with the patient characteristics. In this example only one term is matching out of three. This strategy yields a wrong output decision; any finding that is independently predicted by the presence of "*Chest pain*" is relevant to the patient from a medical point of view. The result concerning "*male gender*" on the opposite is not relevant to her. A better but still naive strategy would be to count the number of parameters terms only. In our example, half of the parameters are matching. This is not enough to decide whether the sentence is relevant or not.

From these observations we draw two conclusions. First, matching should not be performed on the 'finding' part of a result sentence; the fact that the patient has a finding recorded in her record does not affect the relevance of the result to her case. Second, among parameters, different matching policies should be applied to determine if a sentence is matching according to the degree of dependence of the parameters.

From a processing point of view, to match a result with a patient record, we need to identify the finding, the different parameters and whether they are independent. The templates extracted in the previous stage of the system provide such a description of the result sentence. This explains why patient matching takes place at this stage and further supports our decision to extract result sentences into templates.

Our strategy for matching is the following. For each parameter in the template, check whether it matches the patient record. If the result reports independence on the parameters, we perform a logical *or* of all matching parameters. The matching parameters are kept as input for the next component in the system, while the others are discarded. In contrast, if the parameters are dependent, *i.e.* their combination relates to the finding, we perform a logical *and* of the matched parameters. If one parameter doesn't match, the whole template is discarded; if they all match, the whole template is passed as it is to the next component.

For instance, given the template in Figure 6 and a 44 years old female patient with unstable angina and ST segment depression, because the parameters are independent, we keep only the matching parameters in the template (the age parameter is dismissed). The matched template appears in Figure 7.

At this point of the pipeline, all the relevant pieces of information are extracted from the articles. We also know that they are matching to the patient. The requirements of completeness and accuracy are met.

4.4 Merging and Ordering

The Merging and Ordering component takes as input a set of templates containing only relevant information for the patient record. We first describe how the information collected over all input articles is merged into a single internal representation. We then present different principles we used to order the information in a coherent manner.

Merging As described in the previous section, each template is mainly composed of findings, parameters and a relation type. Findings and parameters are medical terms (uniquely identified by their concept id.). A medical term can be both a parameter in one template, and a finding in another[2]. Merging consists of combining all the results extracted in the different input articles into one single internal representation, namely a semantic graph of results.

In the graph, nodes are concepts (parameters and findings of the templates), while vertices are relations from the templates (*e.g.* prediction or risk). A vertex of type r connects two nodes p and f if there exists a template containing the result (p, f, r), where p is a parameter, f is a finding and r is the relation between them. Vertices have different types, as many as there are different relation types (in our case six). Nodes are indexed on their unique semantic identifier (CUIs). We refer to the triplet (p, f, r) in the graph as a relation.

The graph is built in an incremental fashion. For each template, each finding and each parameter is converted into a node; if another node with the same CUI already exists in the graph, the two nodes are merged. Figure 8 shows a graph built from different templates.

When building the graph, if a result contains several parameters, it is broken down into several relations (one for each individual parameter). This is done both for independent and dependent results. From a medical point of view, our representation is accurate as long as no relation from a dependent result is presented as independent in the summary, or vice versa. To ensure that we are accurately presenting the results, we store into vertices the template id of the result, so that later in the pipeline when ordering and generating the relations, we present relations from dependent results together.

[2] There are exceptions, for instance "*death*" can only be a finding while "*age*" can only be a parameter.

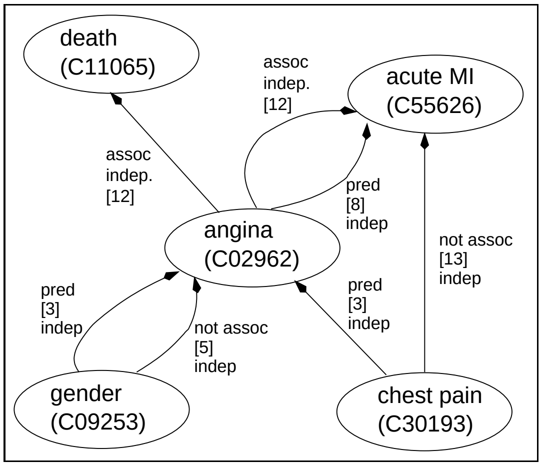

Figure 8: A Graph built from different templates. Template ids are in brackets, indep represents a relation from an independent result, pred stands for predict relation and assoc stands for association.

This internal representation allows the identification of repetitions and contradictions. We consider that a repetition occurs if two nodes are connected by more than one vertex and the vertices have similar types. Similarly, a contradiction occurs in the same situation but when the vertices have different types. In interviews with physicians, we established which of the different types of relation are similar or contradictory. For instance, in Figure 8, our method detects one repetition (two templates identified "*angina*" and "*acute myocardial infarct*", or "*acute MI*" as associated[3]) and one contradiction (the relation between "*gender*" and presence of "*angina*" is contradictory).

Identifying repetitions and contradictions across documents is known to be a hard task (McKeown et al., 1999). Our method tackles this issue by exploiting a semantic representation, which is similar to the approach described in (Radev and McKeown, 1998) in the news domain. However, in our domain, contradictions and repetitions are more complex to identify. For instance, in Figure 8, physicians would consider the two relations *(chest pain, angina, predict)* in conjunction with *(angina, myocardial infarct, predict)* a contradiction with the relation *(chest pain, myocardial infarct, not associated)*. But, in the general case, we cannot assume transitivity for the relations in the graph, for this might produce inaccurate medical inferences. We are investigating ways to identify more subtle repetitions and contradictions. This involves defining in a formal way the concepts of repetition and contradiction as well as extending our representation to take these new definitions into account.

[3]The relation of prediction subsumes association for physicians.

Our internal representation contains the sum of information from the different input articles. It also contains additional information about the collection of input articles, namely repetitions and contradictions. We exploit this structure to produce a more coherent summary. Additional knowledge about the input articles (as opposed to knowledge from the articles) can help us decide in which order to present the information. From this point of view, our representation meets the requirements captured in the analysis, that is, presenting a synthesis of the information.

Ordering Once all the information is merged into a single internal representation, it needs to be presented to the user in a coherent and cohesive manner. The Ordering sub-component assigns priorities to the different results, so that when the graph is traversed to output the information to the next component, high priority results are presented first. Each relation in the graph is assigned a weight according to the following principles:

- query based — a relation that answers the user query is weighted higher,
- salience based — repetitions and contradictions are weighted higher,
- domain based — studies with physicians show that some relation types are more interesting than others. For instance a risk relation is weighted higher than an association relation,
- source based — dependent relations multiplied out from a same template are presented together, because they represent a result as a whole.

4.5 Generation

Since we are selecting pieces of information from different input articles, we cannot use simple extraction methods for the preparation of the summary. We therefore rely on a text generator to combine the information encoded in our graph into a fluent summary. Following a standard generation architecture, we distinguish a sentence planner from the realization module.

The sentence planner builds a rhetorical structure from the graph, highlighting repetitions and contradictions. For example, Figure 9 shows the target summary for the graph shown in Figure 8.

The fact that a contradiction was found between templates 3 and 5 is rendered in the first sentence. Several linguistic devices are used to express the contradiction (introductory clause, connective, punctuation). Similarly, the repetition between template 8 and 12 is expressed in two ways: the two findings are merged in a single sentence, and the double reference to [8, 12] highlights the repetition.

The relation between angina and gender is controversial: in [3], gender was found to predict angina, while [5] finds proof of no association.

Chest pain predicts angina [3] but it is not associated with acute MI [13].

Angina predicts acute MI [8, 12]. Angina is also associated with a decreased event-free survival [12].

Figure 9: Target summary for the graph shown above. In brackets are the templates number the information is generated from. Note the rhetorical rendering of contradictions and repetitions.

We are currently working on a sentence planner which converts our semantic graph into a rhetorical graph that can encode different types of relations beyond repetitions and contradictions: chaining and comparison are also candidates, and can be identified using our semantic representation. The output rhetorical structure is an RST-like tree structure (Mann and Thompson, 1988). The identification of the rhetorical structure meets the requirement of building a coherent summary, which, in addition to presenting the basic information extracted from articles, highlights the relation between the findings.

The realization component turns the rhetorical structure into English sentences. It combines a semantic representation (findings, parameters, relations) with full expressions extracted from the input articles. We are, therefore, developing a *hybrid generator*, integrating shallow merging of existing noun-phrases and deep realization (aspects of lexical choice and syntactic realization). We are investigating the use of the SURGE realization grammar (Elhadad and Robin, 1996) for this task.

5 Related Work

While most of the work in summarization aims at developing domain-independent techniques (Mani and Maybury, 1999), there are a number of approaches focusing on a specific domain. When tailoring a summarization system to a particular domain, domain specific characteristics improve the quality of summaries. In our work, we rely heavily on user studies to capture the users needs and better meet their requirements.

Radev and McKeown (1998) developed a domain dependent system for summarization of multiple news articles on the same event in the terrorist domain, highlighting how the perspective of the event changes over time. The input of their system consists of the result of an information extraction system and the output is a summary in fluent English. We build on their architecture, pairing information extraction and natural language generation techniques. However, the generation techniques used in (Radev and McKeown, 1998) are more limited, and cannot deal with the level of uncertainty present in our domain for two reasons. First, Radev and McKeown (1998) recognize repetitions and contradictions in a straightforward way. For instance, if two templates, related to the same terrorist act, contain different terrorists affiliations, they are considered to be a contradiction. In our case, a simple string comparison is not sufficient to determine a contradiction and more sophisticated inferences are required. Second, the input to our system is not only a set of articles but also a patient record and a user query. The combination of the three determines not only what is worth including in the summary, but also how to present information to the user. We therefore need more flexible and scalable methods for generation.

In recent work, Barzilay et al. (1999) presented methods for identifying similarities across documents in a domain-independent framework, using lexical and syntactic features. We believe that by exploiting an internal semantic representation of the information we want to summarize, we can achieve higher accuracy at discovering repetitions. We also provide a mechanism to identify contradictions which is a much harder task in a domain-independent setting.

6 Scalability

The summarization system we presented is designed and trained to work on technical medical articles. In our work to date, we focus on the cardiology domain. From a technological point of view, moving to a new medical domain, such as diabetes, can be done easily. The Results Extraction component would require the most changes, because new extraction patterns would have to be constructed. Other components rely on medical knowledge and resources that can be reused for any medical domain. Our system is undoubtedly domain dependent, but the domain it operates on is vast. According to the Ulrich's Periodicals Directory (Ulr, 2001), there are 312 journals in cardiovascular disease alone, which is one among the many subfields of cardiology. In the field of cardiology in general, there are at least 700 journals and many more yet in medicine overall.

It is an open question how feasible the application of the system to another domain is. In our framework, both the set of input articles and the patient record define which information should be included in the summary. When moving to a new domain, one would have to determine what could substitute for the patient record in the new domain. In biology, for instance, even though the genre of articles is similar to the medical domain (experiments and results of studies are reported), there is no obvious equivalent of a patient record that would determine salient information.

As explained in previous sections, when extracting relevant information, we are not interested in results reported in the Abstract section of the input articles. The main reason is that our salience criteria comes from the main characteristics of the patient record. In a similar way, if we were to run a multi-document domain independent summarization system on our set of input articles, the output summary would not satisfy physicians, in part because it would extract information indiscriminately from different sections of the articles and in part because it wouldn't provide them with information pertaining to a specific patient record.

7 Planned Evaluation

We have, so far, evaluated some components of our system individually. The Article Type Classification component achieves 94% when trained on a small corpus of articles. The Results Extraction component has also been evaluated showing 92% precision and 50% recall, which is expected given the manual construction of extraction patterns. The Patient Matching component has to be evaluated by physicians and its evaluation is more difficult to set up. In a first set of experiments, we asked physicians to identify the result sentences or phrases that match with a specific patient record in a set of articles. The results of the evaluation could not be exploited because our guidelines were not specific enough and physicians showed very different behavior. We are in the process of designing a better evaluation for this component, taking into account the problems of our first attempt.

We plan to evaluate the Merging and Ordering component along with the Generation component by asking physicians to compare and grade three types of summaries: summaries composed of sentences extracted from the articles with no ordering or merging of the information and without any regeneration, summaries with ordered sentences extracted from the articles, and summaries produced by our system. This will allow us to evaluate our system according to the requirements identified in section 2. This will also provide us with an overall evaluation of the system, determining if the content of summaries is useful and relevant for a physician when treating a specific patient.

8 Current and Future Work

To date, we have implemented a prototype system with modules for each stage, focusing on producing a complete end-to-end summarization system. Our current implementation works on a small number of examples, demonstrating feasibility of our approach.

We are currently working on increasing robustness by extending the range of input data that it can handle. For example, currently our approach can handle prognosis articles only. In order to extend to handle diagnosis and treatment articles, we need to extend the kinds of patterns used in the information extraction module, but other modules should remain the same.

In order to increase the number of contradictions that our system can identify, we are currently annotating a corpus of articles with contradictions. From this data, we will train the system to recognize new relations and develop inferences for implicit contradictions. We can use a subset of this corpus as test data to evaluate the system.

Finally, to make portability to new article types easier, we are investigating the use of machine learning to automatically build the patterns for the Results Extraction component.

9 Conclusion

We have presented a prototype summarization system that generates a summary of multiple medical journal articles, tailored to the problems of a patient under a clinician's care. Our approach is data driven, producing a summary that uses relations between the extracted results, such as contradiction, repetition and relatedness, to organize and order summary content. By integrating a range of techniques that have previously been used separately for summarization, our approach allows the selection and organization of just those facts that clinicians have expressed interest in, patient specific results.

A key feature of our approach is the use of a semantic graph to represent and link facts that are contained in separate templates. The graph allows us to identify concepts that are repeated across articles. It also facilitates the identification of contradictory relations between the same concepts. Using the merged representation of facts extracted from different articles, ordering of summary content can be achieved by different traversals of the graph which ensure that salient information such as repetitions and contradictions appear first, while related results will be placed together in groups in the generated summary.

References

R. Barzilay, K. McKeown, and M. Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proc. of the 37th Annual Meeting of the Assoc. of Computational Linguistics.*

M. Elhadad and J. Robin. 1996. An overview of surge: a re-usable comprehensive syntactic realization component. In *INLG'96, Brighton, UK. (Demonstration Session).*

I. Mani and M. Maybury, editors. 1999. *Advances in Automatic Text Summarization.* MIT Press.

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

K. McKeown, D. Jordan, and V. Hatzivassiloglou. 1998. Generating patient-specific summaries of online literature. In *AAAI Spring Symposium Series, Intelligent Text Summarization*.

K. McKeown, J. Klavans, V. Hatzivassiloglou, R. Barzilay, and E. Eskin. 1999. Towards multidocument summarization by reformulation: Progress and prospects. In *Proc. of the 17th National Conference on Artificial Intelligence*.

K. McKeown, S. Chang, J. Cimino, S. Feiner, C. Friedman, L. Gravano, V. Hatzivassiloglou, S. Johnson, D. Jordan, J. Klavans, and S. Teufel. 2001. Persival, a system for personalized search and summarization over multimedia healthcare information. In *To Appear in Proc. of The Joint Conference on Digital Libraries 2001.*

E. Mendoca, J. Cimino, S. Johnson, and Y. Seol. 2001. Accessing heterogeneous sources of evidence to answer clinical questions. Technical report, Columbia University.

National Library of Medicine, Bethesda, Maryland, 1995. *Unified Medical Language System (UMLS) Knowledge Sources*. `http://www.nlm.nih.gov/research/umls/`.

D. Radev and K. McKeown. 1998. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500.

S. Teufel, V. Hatzivassiloglou, K. McKeown, K. Dunn, D. Jordan, A. Kushniruk, C. Friedman, and S. Sigelman. 2001. Personalized medical article selection using patient record information. Technical report, Columbia University.

2001. Ulrich's periodicals directory. Available at `http://www.ulrichsweb.com`.