Extracting Conceptual Relationships from Specialized Documents

Bowen Hui[1] and Eric Yu[2]

[1] Department of Computer Science, University of Toronto bowen@cs.utoronto.ca
[2] Faculty of Information Studies, University of Toronto yu@fis.utoronto.ca

Abstract. Conceptual modeling has been fundamental to the management of structured data. However, its value is increasingly being recognized for knowledge management in general. In trying to develop suitable conceptual models for unstructured information, issues such as the level of representation and complexity of processing techniques arise. Here, we investigate the use of a conceptual model that is simple enough to allow efficient automatic extraction from documents. Our model focused on the problem-solution relationship that is central to the analysis of scientific papers. It also consists of supporting relationships such as benefits and drawbacks, assumptions, methods, extensions, and claims. Our study considered two kinds of documents – scientific research papers and patents. We evaluated the utility of the approach by building a prototype system and our user evaluation shows promising results.

1 Introduction

Conceptual modeling has been fundamental to the management of structured data. However, its value is increasingly being recognized for knowledge management in general. Much of human knowledge is expressed in the form of textual documents. There are many initiatives, such as the Semantic Web [1], to structure or mark up text into forms that will allow them to be interpreted and processed according to some semantic model. In trying to develop suitable conceptual models for unstructured information, one faces a number of issues and trade-offs. Ideally, one would like to have detailed conceptual distinctions and relationships that would enable specific inferences and conclusions to be drawn. However, extracting knowledge from text according to fine-grained conceptual models can be very labour-intensive and error-prone. It is also difficult to come up with models that can be widely agreed upon. Pragmatically, one would like to be able to adopt conceptual models that are sufficiently detailed to be useful for a given task, and that the cost of applying that conceptual model to the textual source is justified by the benefits. In this paper, we investigate the use of a conceptual model that is simple enough to allow efficient automatic extraction from documents. We evaluate the utility of the approach by building a prototype system to perform the extraction, then conducted empirical studies with users.

We focus on specialized classes of documents which have a dominant semantic relationship of interest. In scientific and technological research, a crucial relationship is that which relates a problem to a solution. Many documents identify a problem of current interest, then offer a solution to address the problem.

The ability to extract the problem-solution relationship quickly from this kind of document can aid in the cursory comprehension of the document, and to place this piece of means-ends knowledge within a network of such relationships. Supporting relationships include benefits and drawbacks, assumptions, methods, extensions, and claims.

Our study considered two kinds of documents which we believed to be amenable to this extraction approach – scientific research papers and patent documents. Information extraction techniques are used to automatically analyze documents according to the conceptual model. The extracted information is presented to users for evaluating the extent to which they find the tool useful and usable. The results are compared to those typically obtained in the information extraction (IE) and text summarization (TS).

First, we describe our proposed conceptual model in Section 2 and the environment under which we envision it will be used. Section 3 describes the components and algorithms of an implemented system that automatically extracts text into our conceptual model. In order to measure system performance, we use accuracy metrics, such as recall and precision, commonly used in information extraction and information retrieval (IR) literature. In addition, we use a well-known usability evaluation method from human-computer interaction (HCI) to measure the usability of our model. An empirical study is reported in Section 4. Finally, Sections 5 and 6 discuss the results, present the contributions, and identify future work.

2 Conceptual Model

Libraries have developed standard ways of organizing books and articles. Typical organization schemes include the use of the author's name, document title, or subject area, which are usually based on keywords [2, 3]. All of these are explicit *tags* of literal strings in the preamble of the document. These schemes are useful as search terms, but they play a very small role in addressing the task of a user who is reading the document. Since these literal tags do not reflect the semantic content written in the paper, they cannot be used to address user tasks. In particular, unless it is written explicitly in the title, these tags do not help in identifying the softgoals of a problem or the weaknesses of a proposed solution. Therefore, we need a way to organize document concepts and relate them to the goals of different user tasks. To achieve this, a conceptual model for documents needs to be developed. Here, we discuss the model for patents and scientific papers that propose original solutions.

Both scientific articles and patents offer unique solutions, using assumptions and stating claims, to various domain goals. By examining user tasks operated around these elements, such as analyzing and comparing different techniques, we create a set of core concepts based on these common goal-oriented elements. By structuring the concepts into relations based on how the concepts are used, the model shown in Figure 1 is formed.

Figure 1 shows the ER model representing the structure of a document. A document identifies a problem and conveys a unique solution. Each problem may

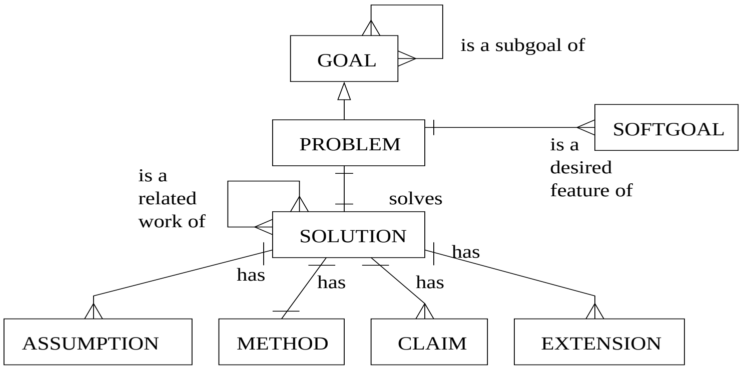

Fig. 1. The Conceptual Model Based On User Goals

be a subproblem of one or more larger goals. Note that a problem is just a small goal that is immediate to particular solutions. A problem also has softgoals, which particular attempted solutions may or may not have addressed. The solution is contrasted with other related works, and is composed of assumptions or beliefs, methods taken to achieve the solution, claims and limitations of the solution, and steps for future extensions. A document may have other kinds of information, such as background material or text structure. These kinds of information are not used in our model because they do not contribute to the goals of the users. More formally, our model has the following concepts: GOAL, PROBLEM, SOFTGOAL, SOLUTION, ASSUMPTION, METHOD, CLAIM, and EXTENSION. The relationships and cardinalities among the concepts are shown in Figure 1.

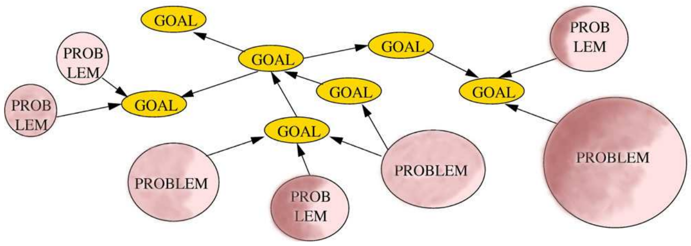

Fig. 2. Multiple Documents Connected By Our Ontology

The intention is to relate multiple documents together using an ontology of research problems consisting of PROBLEM and GOAL instance nodes, as illustrated in Figure 2. Each document can relate to multiple problems and goals and many documents can share the same problem. GOAL nodes may be linked to each other if one is a subgoal of another. The size of the PROBLEM node indicates the number of solutions associated with it and the colour density indicates the recency of publications.

Examining the landscape of ideas as in Figure 2 allows analysts to detect current patterns in research methodologies. Large PROBLEM nodes represent a well-tackled issue while small PROBLEM nodes represent widely opened questions. Colour is used as an indicator for chronology of the research. For example, a dark colour represents works in the recent decade, a neutral colour represents works from two decades before that, and a light colour represents all other previous works. Densely coloured nodes on the map represent hot topics while lightly coloured nodes represent less active research.

To address user tasks, we describe how an analyst might use a visionary system with our model and ontology. By implementing search and inference functions into the system, one could select a GOAL node and identify which companies or laboratories have done similar work. By comparing different METHOD nodes of the same PROBLEM, one could identify which methods have not yet been tried. Detecting a "collision of ideas" is also possible and can prevent intellectual property conflicts. Detecting "holes" in the ontology allows analysts to attempt novel ideas and set new trends in research and development. The ontology in Figure 2 serves as an overview of problems to tackle in our visionary system. Upon clicking on a particular node, the system would zoom into a document conceptual model that attempted to tackle the selected node. By clicking on a node in the document model, details would appear on demand. In this manner, the ontology provides an overview on one part of the screen while the concept structure of the paper is displayed beside it and detailed text beneath it. A mock-up of this application is illustrated in Figure 3. Details of the document model on the right side of the mock-up will be explained in Section 3. Coupled with this ontology, our model provides support for information researchers and addresses user goals effectively.

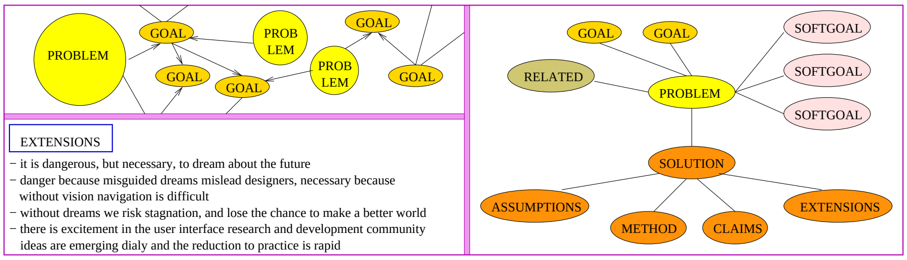

Fig. 3. Our Visionary System

3 System Implementation

The purpose of the system is to evaluate the effectiveness of our model. We implemented a system that creates the conceptual model for given documents. An automatic extraction algorithm is used to analyze the free text and information is displayed in a graphical layout. Major components of this system are preprocessing, segmentation, merging, and interface generation.

Scientific articles and patents are entered into the system and relevant sections are extracted in the preprocessing stage. Extracted sections are passed

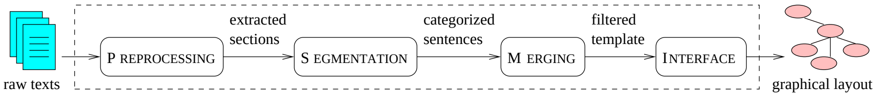

Fig. 4. Document Visualization System

into the segmentation module, where each sentence is assigned a category based on pre-defined hand-coded patterns. Categorized sentences are passed into the merging module, where similar sentences are merged and common phrases are removed. The output is a set of point-form phrases or sentences in a template format. Finally, the interface component takes the textual template and transform it into a graphical layout. Figure 4 shows the interaction of these components.

Slot Name	**Abbrev.**	**Description**
GOAL	G	high-level goals, domain goals
PROBLEM	P	immediate goals
RELATED	R	cited or discussed related works
SOLUTION	S	overall proposed approach
SOFTGOAL	F	desired benefits of the area
ASSUMPTIONS	A	assumptions, definitions, beliefs, trends
METHOD	M	particular algorithms, steps, devices
CLAIMS	C	results, claims, advantages, limitations
EXTENSIONS	Z	extensions, future directions
OTHER	X	other information: e.g. background, evidence

Table 1. Template Slots

All of these modules are based on a template structure which we have created and shown in Table 1. This canonical template is derived from the ER model in Figure 1 and is used as the target output so that information extracted from documents can fill up the slots. Because a text template represents the information of one document, some slots are different from the entities used in the ER model. For instance, in the ER model, a SOLUTION entity is related to other SOLUTION entities. In the text template, the proposed solution of the document is represented as SOLUTION and other works are put into RELATED. Also, rather than having multiple ASSUMPTION, CLAIM, and EXTENSION entities, we have created one slot for each of these that may store multiple extracted facts. Table 1 also includes a "dummy" slot, OTHER, which we have inserted to temporarily store information that is irrelevant for our system. This category and its information do not appear in the final interface to the users. A

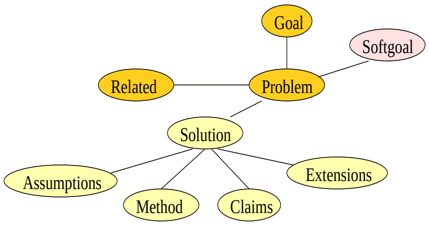

Fig. 5. Graphical Layout

graphical layout derived from the canonical template is shown in Figure 5. Each node corresponds to a slot in the template, with OTHER left out. This view of concepts is presented to the users.

3.1 Preprocessing

A document is converted into ASCII and excerpts are extracted and used as the basis of analysis. In particular, the abstract, introduction, and conclusion sections for scientific articles and the abstract and summary sections for patents are used. We believe that these sections are representative of the main ideas of the entire paper, and hence, sufficient in fulfilling our document analysis task. Documents that cannot be generated as text or that do not have the required sections are discarded.

3.2 Segmentation

The segmentation phase is responsible for categorizing sentences into template slots. An initial set of hand-coded patterns that use discourse cues were created from a small training sample. All patterns in the system are hand-coded because extensive results show that hand-coded patterns outperform machine-learned patterns [4]. The segmentation first tries to identify patterns that match sentences (for most of the categories) and then match patterns at the word level (for GOAL and SOFTGOAL only). If a sentence does not have an assigned category, the system tries to copy the category of the previous sentence. If by the end there are sentences still left unassigned then they get the category OTHER. These steps are illustrated in Algorithm 1.

```
for each document excerpt
{
    1. classify each sentence into a category
    2. if a sentence does not have a pattern after step 1, then
           if its previous sentence has an assigned category, then
              inherit that category
    3. identify patterns for phrases matching goals and softgoals
    4. classify all unassigned sentences as OTHER
}
```

Algorithm 1: The Segmentation Algorithm

Patterns are defined at the word level, matching sequences of words in the text. About 20 documents (8 patents,12 scientific articles – henceforth denoted as: 8 pat/12 sci) were used to develop these patterns. The current prototype has a total of 316 rules[3]. Examples of patterns and the number of patterns defined for each template slot are given in Table 2.

[3] Our patterns are implemented using regular expressions in Perl. When we count the number of patterns in this component, we are giving the number of *lines* of regular expressions we used. On average, patterns matching GOAL and SOFTGOAL were as short as 2 words per regular expression. The other patterns vary greatly; one regular expression may consist of just one phrase, while another may have a phrase that has a disjunction of 10 nouns and 10 verbs.

Unlike most IE systems, discourse heuristics are not used. However, these heuristics can be embedded into the module between step 2 and step 3.

Slots	**Sample Patterns**	**Num**
PROBLEM	"Our objective in ... is the study of"	38
RELATED	"Another successful approach"	26
SOLUTION	"We propose"	8
ASSUMPTIONS	"A long standing objection"	24
METHOD	"to achieve ... is used"	50
CLAIMS	"significant improvement"	54
EXTENSIONS	"yet to be investigated"	10
OTHER	"In this section, we ..."	46
GOAL	"technology in education"	17
SOFTGOAL	"immediate feedback"	43

Table 2. Sample Patterns

Based on these patterns, sentences from the document excerpts that are deemed relevant are categorized into one of the slots. These categorized sentences are the output of the segmentation phase.

3.3 Merging

The algorithm for the merging phase is shown in Algorithm 2.

```
for each categorized sentence,
{
    1. order each sentence into the appropriate slot in a template
    for each slot in the template
    {
       2. for each sentence in this slot
          - remove common phrases and transition phrases
       3. for every pair of sentences in this slot
          - stem the sentences
          - if the stemmed pair is similar, then
               keep the more concise one
    }
}
```

Algorithm 2: The Merging Algorithm

The merging algorithm puts the categorized sentences into the appropriate slot (step 1) and makes the output more condensed (steps 2-3). In step 2, sentences are shortened by removing common cue phrases. In addition, topicalized phrases fewer than five words (such as "In addition" and "To our knowledge") are removed. Step 3 compares pairs of sentences within each slot and merges them if they are similar. Similarity occurs when a sentence is a substring of the other or when the two sentences have more than 80% identical words after stemming the sentences using the Porter algorithm [5]. To merge a pair of sentences, the system chooses the shorter one because it considers it to be more concise.

3.4 Interface

The interface produces the graphical model of the template and shows textual details on the side when a node is clicked on. If a node does not have any associated text – either because the algorithm did not detect relevant sentences or that the original excerpts did not have any information – then the interface displays “Text not available from excerpts.” Figure 6 shows a snapshot of the system prototype, with the title of the paper placed at the top of the screen.

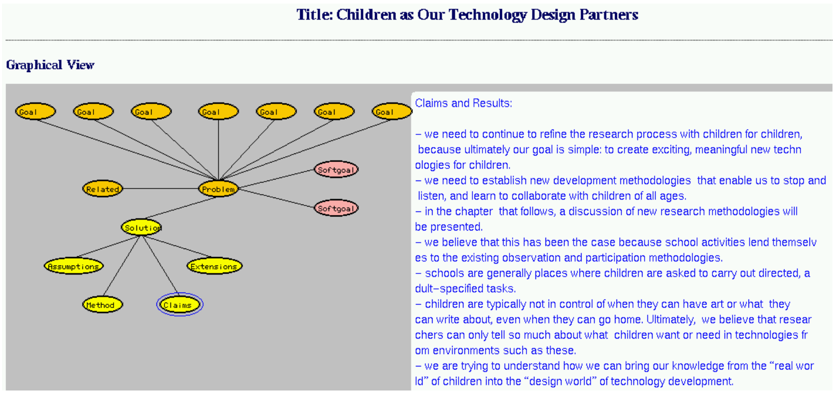

Fig. 6. Sample System Use

This figure shows that the CLAIMS node is highlighted to indicate that it has been clicked on by the user. The details of the node appears in point-form on the right-hand side of the screen. The interface maintains the structural view of the paper while displaying details on demand.

4 An Empirical Study

An evaluation of system performance is measured via standard accuracy tests and an assessment of our document conceptual model is measured via standard usability tests. In order to assess accuracy and usability, the two major components – segmentation and merging – were evaluated separately. This section briefly describes the evaluation techniques used. For further details, please refer to cited references. The procedure and results of the two evaluations are reported.

4.1 Accuracy Evaluation

Participants. To achieve reliability, we performed an intercoder reliability study on 8 documents (4 pat/4 sci). The three coders were $\alpha = 43\%$ reliable above chance when considering all the categories and $\alpha = 53\%$ reliable above chance when considering only sentence level categories (i.e., without GOAL and SOFTGOAL). Then, a third-party coder continued and evaluated the system on 93 more documents (47 pat/46 sci).

Stimuli. A random set of patents were retrieved from the web site at the United States Patent and Trademark Office[4] by using search keys such as "chinese" and "fonts", or "multiplication" and "learning". A random set of scientific articles were retrieved from many sources over the Internet. Topics of articles used in training included psychology, linguistics, human-computer interaction, computer graphics, statistics, and economics. The specific topics and the number of documents used for testing are listed as follows:

- **Patents:** colour toy (11), education mathematics (12), design blouse (12), modern chair (12)
- **Scientific Articles:** women and language (10), children-related HCI (12), interface-related HCI (12), computational linguistics (12)

In sum, 51 patents and 50 scientific articles were used in this study.
Procedure. The coding system used were a set of letters to identify which sentence exhibit relevant words or phrases for particular slots in a template. Hence, we used the slot names as classification labels (cf. Section 3).
Analysis. Standard recall and precision measures from IR and IE were used:

$$recall = \frac{items\ gathered}{items\ in\ standard} \qquad precision = \frac{items\ correctly\ gathered}{items\ in\ standard}$$

where an *item* is typically a syntactic unit or a semantic concept. Combining precision and recall produces a single metric, *F-score*:

$$F - score = \frac{(\beta^2+1)P * R}{(\beta^2)P + R} \qquad when\ \beta = 1, F - score = \frac{2PR}{P+R}$$

This kind of evaluation measures fine-grained accuracy against a "gold standard". In our evaluation, the gold standard used was the manual validation of a third-party human evaluator.
Results. Segmentation results are shown from two perspectives – performance by domain and performance by category. The former is presented in Tables 3 and 4 and the latter in Table 5. (Note: N_c is the number of correct items, N_g is the number of items gathered by the system, and N_s is the number of standard items (i.e., items marked by the evaluator).)

Domain	**N_c**	**N_g**	**N_s**	**Precision**	**Recall**
Pat 1: colour toy	236	319	324	73.981	72.839
Pat 2: education mathematics	349	429	430	81.351	81.162
Pat 3: design blouse	254	386	400	65.803	63.500
Pat 4: modern chair	219	326	311	67.177	70.418
Average	264	365	366	72.078	71.979

Table 3. Segmentation Results for Patents by Domain

From Tables 3 and 4, we see that accuracy results were generally low for scientific articles, about 48%, and significantly higher for patents, about 72%.

[4] The web site address is `http://www.uspto.gov`.

We believe that these results are credited to the consistent writing style that patents exhibit. In Table 5, Related, Assumptions, and Extensions have less than 10% recall. This means that much of the information belonging to these categories were not extracted. Related and Assumptions also have low precision scores of 35% and 33%. This means that much of the information that was classified in these slots were incorrect. The remaining categories received decent or good accuracies, as they are comparable to extraction systems that compete in Message Understanding Conferences (MUC) whose average F-score on the overall IE task is approximately 60 [4].

Domain	$\mathbf{N}_c$	$\mathbf{N}_g$	$\mathbf{N}_s$	**Precision**	**Recall**
Sci 1: women and language	202	534	527	37.827	38.330
Sci 2: children-related HCI	428	769	791	55.656	55.656
Sci 3: interface-related HCI	260	507	502	51.282	51.792
Sci 4: computational linguistics	314	636	640	49.371	49.062
Average	301	612	615	48.534	48.710

Table 4. Segmentation Results for Scientific Articles by Domain

Category	$\mathbf{N}_c$	$\mathbf{N}_g$	$\mathbf{N}_s$	**Precision**	**Recall**
Goal	232	242	252	95.867	92.063
Problem	133	216	261	61.574	50.957
Related	6	17	113	35.294	05.309
Solution	95	110	119	86.363	79.831
Softgoal	202	329	300	61.398	67.333
Assumptions	9	27	109	33.333	08.256
Method	782	1516	1031	51.583	75.848
Claims	387	656	826	58.993	46.852
Extensions	3	4	42	75.000	07.142
Other	413	789	872	52.344	47.362
Total	2262	3906	3925	57.911	57.631

Table 5. Segmentation Results By Category

4.2 Usability Evaluation

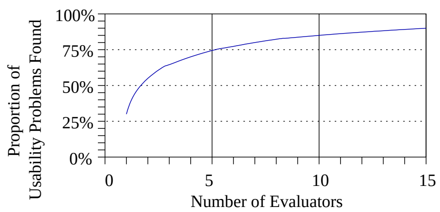

Fig. 7. Usability problems found by heuristic evaluation as a function of evaluators (taken from [6]).

Heuristic evaluation is an inspection method for assessing system usability. The evaluator's job is to inspect the interface by carrying out purposeful tasks that allows him/her to navigate through different parts of the interface. Evaluators are asked to comment on the interface with respect to a set of design principles. The result of a heuristic evaluation session is a compilation of problems with the system. Since different evaluators tend to identify different problems, Nielsen [6] empirically shows that 50% to 75% of problems can be identified with just 3 to 5 evaluators (see Figure 7). Hence, this method is very cost-effective.

Because our system extracts relevant information and puts it into a more concise representation, our system is in essence performing a basic summarization task. Therefore, we use text summarization metrics to evaluate the output of our system. First, we amalgamated facts on "summary worthiness" [7–10] and derived an initial set of heuristics, or *principles*, for the evaluation, as follows:

1. **Conciseness:** Components should not contain information that is irrelevant or redundant. Every extra unit of information competes with the relevant units of information and diminishes their relative visibility. All information should appear in a natural and logical order.
2. **Retention:** Information retained in the system output should be representative of the key concepts and main points made in the original document. Are the major objectives of the paper captured in the summary? What about the major steps in the proposed solution and the results?
3. **Coherence:** All information should be coherent within each component as well as the overall summary. Sentences need not be perfectly grammatical, but each point should make sense in its context.
4. **Consistency:** Each component should be expressed clearly in words, phrases, and concepts consistent with those in the original document. Users should not have to wonder whether different words, situations, or actions mean the same thing.
5. **Informativeness:** Information should be presented in a useful and easily accessible way. Some interface issues may be influential here as well. Irrelevant information should be omitted and words should not clutter the display of the information.
6. **Comprehensibility:** Each point of information should be easy to understand. Users should not have to look up related information in another part of the system in order to understand a particular component.
7. **Fit For Audience:** The information and the style of presentation fits for the intended audience. Audience may vary in the experience of the domain knowledge. Access to different kinds of information should be easy and clear. The ability to show, modify, and hide information should be made obvious to the users.
8. **Fit For Purpose:** The information and the style of presentation fits for the intended task and purpose. These two points should be taken into consideration.

To let evaluators assess the usefulness of the model, we used a question-answering task modeled after a conference reviewer's task. This way, evaluators acted as reviewers using only the system output. To keep the workload of the evaluator to a minimum, we limited the task to three questions:

1. What is the problem addressed by this work? Does it describe why the problem is significant?
2. Does the work present the approach taken to solve the problem targeted? Is the design or implementation of a system described in terms of key ideas of the approach?

3. What are the contributions of this work? Are the benefits and limitations clear? Are the results positive or negative?

The rest of this section describes the details of our user study.

Participants. Four University of Toronto graduate students with extensive experience in writing summaries and abstracts for scientific papers participated in this study. Each of them evaluated 4 system generated output shown on the system's graphical interface. They are henceforth referred to as evaluators.

Stimuli. Evaluators were provided with the system prototype on a web browser. On different pages, the browser displayed the instructions, two document models, the question-answering task description, and the summarization principles. When commenting on the principles, evaluators were allowed to read to the original documents. Of the 93 documents used in the accuracy experiment (cf. Section 4.1), four were chosen for this study based on these criteria:

- patent that scored the lowest on recall and precision
- patent that scored the highest on recall and precision
- scientific article that scored the lowest on recall and precision
- scientific article that scored the highest on recall and precision

Because some document excerpts were extremely short, an additional criteria in our selection is that the document must have more than 25 sentences. Finally, the documents used in this study had the accuracies shown in Table 6.

Label	Document Type	Precision	Recall
P1	Patent	32.0	25.806
P2	Patent	100.0	100.0
S1	Scientific article	29.730	30.556
S2	Scientific article	82.222	78.723

Table 6. Accuracy of Documents Used in Usability Evaluation

Procedure.[5] An overview of the system and an explanation of the evaluation procedure and the summarization principles were provided for the evaluators. Each evaluator evaluated 4 document outputs independently. Evaluators were asked to type in their comments in an email and summarize their results on a 5-point scale. Principles that could not be assessed in the session were assigned as non-applicable, which corresponds to a score of 0.

Analysis. To generate a summary of the collected data, the scores from each evaluator were tallied for each principle and document. These values gave a sense of how usable the system was with respect to each principle. Since there were four evaluators and four documents, each principle may have a maximum score of 80 points (i.e., if all judgments score a 5) and a minimum of 0 points (i.e., if all judgments were non-applicable).

Results. Each evaluator spent 1.5 to 2 hours. Combining the scores of all the evaluators together, the quantitative evaluation for each document, for each document type, and for all the documents, are presented in Table 7.

[5] The evaluation was presented and performed online. The entire procedure is available at `http://www.cs.utoronto.ca/~bowen/masters/realeval/heuristic-eval.html`.

Principle	P1	P2	Sum	Avg	S1	S2	Sum	Avg	Total	Average
Conciseness	11	11	22	2.7	12	16	28	3.5	50	3.1
Retention	11	09	20	2.5	17	14	31	3.9	51	3.2
Coherence	10	09	19	2.4	14	17	31	3.9	50	3.1
Consistency	16	11	27	3.4	17	18	35	4.4	52	3.3
Informativeness	11	09	20	2.5	15	14	29	3.6	49	3.1
Comprehensibility	04	08	12	1.5	17	15	32	4.0	44	2.8
Fit for Audience	13	12	25	3.1	16	16	32	4.0	57	3.6
Fit for Purpose	15	13	28	3.5	17	17	34	4.3	62	3.9

Table 7. Summary of Usability Results by Principles

Contrary to the accuracy evaluation in Section 4.1, the reverse results were found in our user study between the two types of documents – scientific articles were much better received than patents. The average usability for patents was 2.7 out of 5.0 and for scientific articles was 3.9 out of 5.0. Despite a large accuracy range between the two patents (28.6% and 100%), all the evaluators found them to be hard to understand. This problem may be due to the unfamiliar writing style of patents in general, as one of the evaluators found the title of the first patent hard to parse and asked "Is this supposed to mean something to me?" Moreover, the segmentation inaccuracy dampened user satisfaction. Comments from evaluators were consistent with these results: "Sometimes the info[rmation] was dead on and other times I couldn't decipher it", "The summary really lack in comprehensibility", and "If the correct information [were] pulled out consistently, this system would be very useful". We hypothesize that the usability scores would be higher if given better segmentation accuracy. Still, the current state of matters shows that users found our model helpful and useful, as an evaluator commented: "I really liked the graphical representation of the parts of the document and the fact that a user can click and see the summarized data on the right".

5 Discussion

The need for modeling unstructured data increases as more information becomes abundant in these formats. Increasing efforts have been placed on developing methods for information dissemination, extraction, summarization, and visualization [11, 12, 4, 13]. Related works in these areas have focused on developing or discovering elaborate ontologies for detailed information with applications in databases [14–17]. Issues to consider in developing suitable models and methods include how fine-grained the model should be and what level of processing is needed, both computationally and cognitively. Thus, the difficulty with elaborate models is filling in the details correctly; it is very difficult to develop automated extraction techniques with high accuracy and generalizability, and it is extremely labour-intensive and error-prone to manually annotate an elaborate model. Furthermore, if more "intelligent" language processing techniques were used, then the processing task becomes too intensive and are often intractable [18].

In contrast, we motivated a simple conceptual model (see Figure 1) for two kinds of specialized documents – patents and scientific articles that propose original solutions. The concepts in our model are based on user-centered goals, such as problem-solution, benefits and drawbacks, assumptions, methods, extensions, and claims. These concepts capture the high-level argumentative structure depicted in the documents. We have demonstrated our techniques on two document classes while achieving reasonable accuracy. To generalize our approach over other document types, one could group the patterns in the segmentation component into modules that correspond to particular document types. We believe this technique can also be used in the current system to improve the accuracy of patents and scientific articles. Based on our user feedback, we hypothesize that an increase in accuracy will also raise user acceptance.

In sum, our tests show that the users are getting value from the fairly simple modeling and processing of our approach.

6 Conclusions

We have argued for the need of modeling unstructured data and methods for processing it. In this paper, we have explored the use of a conceptual model that is *simple* enough to allow efficient automatic extraction from documents and *representative* enough so that users got value from using it. Our focus was on the problem-solution relationship – a particular kind of relationship that is crucial to research in science and technology. The model also provides supporting relationships, such as benefits and drawbacks, assumptions, methods, extensions, and claims. All of these concepts are designed based on user-centered goals in order to facilitate the processing of scientific documents by humans. The overall framework of this model is intended to be used in the context of an ontology of research problems, as illustrated in Figure 2, so that users may see how different problems relate to each other. Our ultimate goal in information management is to provide researchers with a map of on-going activities and tool support to do task specific searches.

Because we are modeling unstructured data, the validation of our model relies on user acceptance. We conducted two evaluations – an accuracy evaluation that measures extraction performance on our segmentation task and a usability evaluation that measures the acceptance of our document conceptual model. Our users found our model useful, although the displayed text was not always perfect. In the future, we hope to iterate on more elaborate conceptual models and improve the accuracy of the extraction techniques for more in-depth user studies. We expect domain novices (and potentially experts as well) will find that simpler models are more usable than elaborate ones. We hope to improve the system and make it available for real use.

Acknowledgments

Bowen Hui is grateful for many fruitful discussions with Eric Yu. She would also like to thank Fanis Tsandilas for his support throughout this project, the evaluators' efforts, and the valuable feedback from the anonymous reviewers. This project was in part funded by NSERC.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American (2001)
2. Weibel, S., Godby, J., Miller, E., R. Daniel, J.: OCLC/NCSA metadata workshop report. (1995)
3. MARC: US MARC Standards. (2001)
4. MUC-7: Proceedings of the 7th Message Understanding Conference (MUC-7). (1998)
5. Porter, M.: An algorithm for suffix stripping. Program **14(3)** (1980) 130–137
6. Nielsen, J.: Usability Engineering. Academic Press, Inc. (1993)
7. Sparck-Jones, K., Galliers, J.: Evaluating natural language processing systems: an analysis and review. New York: Springer (1996)
8. Hovy, E., Marcu, D.: Automated Text Summarization: Tutorial Notes. COLING-ACL'98, University of Montreal, Montreal, Quebec, Canada. (1998)
9. Teufel, S., Moens, M.: Discourse-level argumentation in scientific articles: human and automatic annotation. In: ACL Workshop on Towards Standards and Tools for Discourse Tagging. (1999)
10. Jing, H., Barzilay, R., McKeown, K., Elhadad, M.: Summarization Evaluation Methods: Experiments and Analysis. In: AAAI Intelligent Text Summarization Workshop. (1998) 60–68
11. Overmyer, S., Lavoie, B., Rambow, O.: Conceptual Modeling through Linguistic Analysis Using LIDA. In: Proceedings of the 23rd International Conference on Software Engineering (ICSE 2001). Volume Toronto, Canada. (2001)
12. Maybury, M.: Tools for the Knowledge Analyst: An Information Superiority Visionary Demonstration. IEEE COMPSAC 98. 22nd Annual International Computer Software and Applications Conference, Vienna, Austria. (1998)
13. TREC-10: Proceedings of the 10th Text REtrieval Conference (TREC-10). (2001)
14. Embley, D., Campbell, D., Jiang, Y., Liddle, S., Lonsdale, D., Ng, Y., Smith, R.: Conceptual-Model-Based Data Extraction from Multiple-Record Web Pages. Data & Knowledge Engineering **31** (1999) 227–251
15. Loh, S., Wives, L., de Oliveira, J.: Concept-Based Knowledge Discovery in Texts Extracted from the Web. SIGKDD Explorations **2:1** (2000)
16. Aussenac-Gilles, N., Biébow, B., Szulman, S.: Corpus analysis for conceptual modelling. In: CEUR Workshop Proceedings. Volume 51. (2000)
17. Shum, S.B., Motta, E., Dominique, J.: ScholOnto: An Ontology-Based Digital Library Server for Research Documents and Discourse. International Journal on Digital Libraries **3(3)** (2000) 237–248
18. Rounds, W.: The relevance of computational complexity theory to natural language processing. MIT Press (1991)