Automated Indexing for Full Text Information Retrieval

Daniel C. Berrios, MD, M.P.H.
Veterans Affairs Palo Alto Health Care System, Palo Alto, CA
Stanford Medical Informatics, Stanford University School of Medicine, Stanford, CA.

ABSTRACT

We report our experience with a statistically based method of generating sentence-level indexing based on identified UMLS concepts and query and vector-space models. We evaluated the system using the consensus markup of two domain experts as the gold standard. UMLS concepts identified both from HTML headings and in paragraph text were valuable in proposing markup. Using both sources of concepts, the model proposed the correct set of concepts in the form of a query prototype 71% of the time. The correct query prototype was ranked first or second in 79% of cases.

INTRODUCTION

Current methods for indexing medical information are clearly limited. Manual indexing is inconsistent, time consuming, and limited by the representational abilities of the indexing language.[1] Word-statistical indexing is easy to automate, but retrieving documents then requires sophisticated linguistic support for query formulation and knowledge of which search terms are highly discriminant. Knowledge-based information retrieval systems improve search precision without novel indexing methods, which suggests that query and context models help to bridge the gap in knowledge between user queries and document indexes.[2] Furthermore, these models can be applied not only to the query-generation process, but also to the indexing task itself.

We have previously reported the development of a web-based, markup-authoring tool.[3] The tool allows knowledge engineers to mark instances of knowledge, thereby creating a detailed index into full text sources (Fig. 1). A specialized ontology building tool[4] defines the markup templates that correspond to queries that domain experts predict users will pose most often (Fig. 2). A third tool then matches search queries with authored markup instances to provide highly precise information retrieval.

We have enhanced the markup-authoring tool to propose indexing. A human reviews automatically generated, provisional markup for correctness and completeness. We have reported the performance of knowledge-based natural language processing techniques to identify domain concepts and relations in full text sources.[5] We now report our experience with a statistically based method of generating markup based on these concepts and relations and our evaluation of the ability of enhanced markup tool to index an entire medical textbook chapter.

METHODS

We have already detailed the methods we used to identify concepts in full text sources using syntactic, semantic and phrasal knowledge from the UMLS.[5] In addition to identifying concepts in full text, we designed a new method to capture contextual information from HTML-formatted documents. Purcell captured contextual information that was either explicit in journal article headings or implicit within full-text in her information retrieval system.[6] We used our full-text concept identification methods to extract UMLS concepts from HTML headings, and then stored this contextual information as XML-compliant HTML for each sentence in the source document. We then used these "context concepts" as well as concepts from sentences in a statistical indexing model.

We adapted Salton's query-document vector-based information retrieval model[7] for use in proposing markup. This model creates vectors for all documents in a collection and for any query that users compose. Each dimension in the vectors corresponds to unique terms in a document (or query). The magnitude of each dimension is a calculated statistic commonly based on term frequency (tf_w) and inverse document frequency (*idf*). A closeness measure[8] is then calculated for each document, *i*, in the collection compared with a given user query, *j*, over all terms, *t*:

$$\frac{\sum_{k=1}^{t}\left(term_{ik} \bullet qterm_{jk}\right)}{\sqrt{\sum_{1}^{t}\left(term_{ik}\right)^2 + \sum_{1}^{t}\left(qterm_{jk}\right)^2}}$$

where $term_{ik}$ are the document vector term weights and $qterm_{jk}$ are the query term vector weights.

1067—5027/00/$5.00

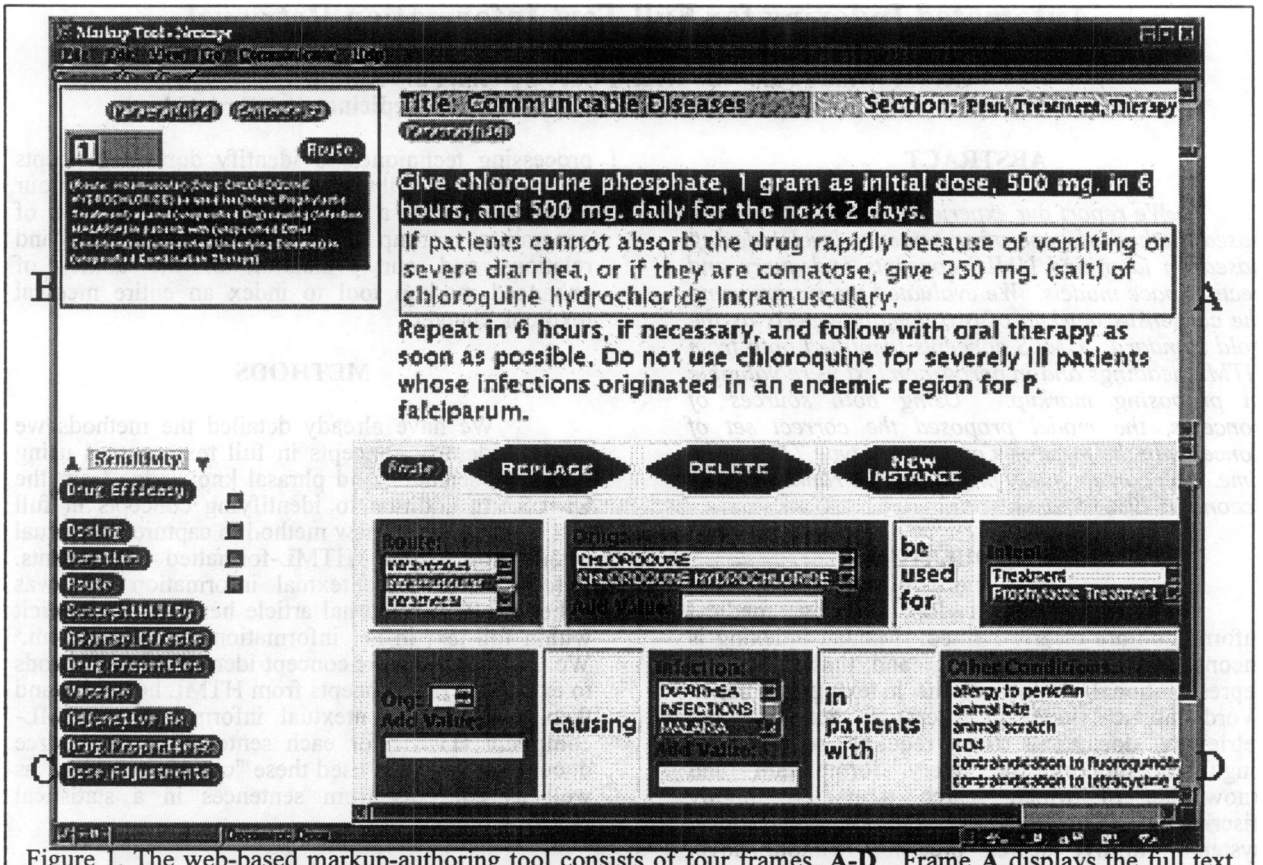

Figure 1. The web-based markup-authoring tool consists of four frames, **A-D**. Frame **A** displays the full text whose paragraphs and sentences can be graphically navigated. The current sentence or paragraph has a red border. In addition, text with markup already associated has a background color matching the query template buttons in Frame C. Frame **B** shows the markup instances for a given sentence or paragraph. Frame **C** lists the query prototypes for a given domain as buttons that, when clicked, load each query prototype into frame **D** for markup authoring. As an example, a "Route" query prototype is shown in frame D. Frame C also can display query prototypes suggestions, with relative strength indicated by the number of indicator squares next to each button. The user creates, replaces, or deletes instances of markup using Frame **D**. The query prototypes consist of UMLS concepts or semantic types (e.g., "pharmacologic substance," the second field in the prototype, labeled "Drug") separated by domain-expert-modeled plain text, and is read from left to right and top to bottom (only a portion of this prototype is shown). Instances of each concept or semantic type that have been extracted from the current paragraph are listed as possible markup, and those instances that occur in the current sentence or that have been indexed already are highlighted. Below each semantic type, the indexer may add values as needed.

For the markup proposal task, we calculated closeness measures for each query prototype for any text section (sentence or paragraph) that a user would like to mark up. In essence, we considered each sentence in the full text to be an entire document. In addition, since the query prototypes were composed of defined concepts and values (e.g., the set of Metathesaurus concepts with a the semantic type "disease or syndrome"), we considered only these concepts as "terms" in all vectors.

Adapting the vector-space model to the automated indexing task necessitated changes in the calculation of the vector components. For any given text section (or query prototype), term frequency is likely to be relatively uniform, as sentence lengths are relatively constant and most terms occur at most once in a sentence. Therefore the tf_w component of the vector weights has very little discriminating power. Since we knew the probability distribution of possible query terms (from the defined set of query prototypes, e.g., as shown in Figure 1D) *a priori,* we defined a statistic analogous to *idf* based on this distribution called *iqtf*, the *inverse query term frequency*, calculated in a manner similar to *idf* (i.e., $\log(N/N_c)+1$, where N is the total number of query prototypes and N_c is the number which contain the concept c). This statistic increases with the scarcity of a concept over a set of query prototypes. We used

Table 1. Expert markup consistency. Instances of markup by Indexer A (columns) vs B (rows).

Query Prototype ID	1	2	4	5	6	7	8	10	11	12	(blank)	Total
1	5	1			1						13	20
2		46	4		2						53	105
3		1									1	2
4		5	44						4	1	21	75
5				2							5	7
6	1	2			14						7	24
7		1				2					9	12
8							1				3	4
10		2						3			3	8
11									3		4	7
(blank)		21	4	1	13	2						41
Total	6	79	52	3	30	4	1	3	7	1	119	305
Kappa, all instances	0.25											
Kappa, Adjusted*	0.82											

*Excluding instances where one or the other marker did not suggest markup.

the logarithmic and smoothed versions of both *idf* and *iqtf* and calculated vector weights as *idf*tf*iqtf*.

For each text section, we created term vectors and iterated the calculation of closeness over the set of query prototypes. Closeness was calculated with and without vector length normalization.[8] For each sentence, we ranked the closeness measures between the text vector and each of the twelve query prototype vectors. Ties for highest or second-highest rank were resolved by random selection.

To evaluate the performance of the vector-space markup proposal model, we asked two domain experts, A and B, to mark up a chapter from a textbook of Infectious Disease.[9] The chapter consisted of 70 paragraphs and 242 sentences. One expert used the automated markup tool (which displayed automatically identified concepts, but did **not** propose which query prototype to index) and one created markup using the same prototypes manually (on paper). The two indexers created a total of 305 instances of markup (Table 1). There was poor overall agreement on query prototype between the indexers (kappa, the proportion of inter-indexer agreement beyond chance, 0.25). However the vast majority of discrepancies occurred because one indexer felt no markup was warranted and the other disagreed, rather than both indexers creating different markup. For example, for a paragraph with five sentences, indexer A felt sentences 2 and 3 should be marked up with query prototype 2, whereas indexer B felt sentences 3 and 4 should have that same markup. This phenomenon is a consequence of the fine granularity of our indexing and retrieval system. Also, if an indexer marked an entire paragraph of text at once, we distributed this instance over all the sentences in the paragraph, generating a large number of these types of discrepancies. If we exclude from the analysis instances that were discrepant because one indexer did not create markup, kappa increased to 0.82, indicating a large amount of agreement when both indexers felt markup was essential.

Table 2. Distribution of consensus query prototypes.

Query Prototype ID	n	(%)
1	5	(0.04)
2	46	(0.38)
4	44	(0.37)
5	2	(0.02)
6	14	(0.12)
7	2	(0.02)
8	1	(0.01)
10	3	(0.03)
11	3	(0.03)
Total	120	(100)

We selected as a gold standard the 120 instances of consensus markup from two indexers. The distribution of query prototypes in this consensus was highly skewed (Table 2), reflecting lengthy discussions in the chapter of drug therapy (query prototype 2) and organism susceptibility (query prototype 4). The remainder of this analysis concerns only the selection of query prototype by the automated indexing model compared with the gold standard.

```
anchor: p30s3
drug administration routes: intravenous
organism: staphylococcus aureus
pharmacologic substance: methicillin%nafcillin
therapeutic or preventive procedure: treatment
qid:7


<therapeutic_or_preventive_procedure value="treatment">
      <treats>
            <disease_or_syndrome>
                  <caused_by>
                        <organism value="staphylococcus aureus">
      <uses>
            <pharmacologic_substance value="methicillin%nafcillin">
                  <has_route>
                        <drug_administration_route value="intravenous">
```

Figure 2. The instance of markup depicted in Figure 1, **D**, formatted for the MYCIN II system (top). "anchor" indicates the location indexed (paragraph and sentence numbers) and "qid" is the numeric identifier for the indexed query prototype. The same instance is shown below formatted in XML using a UMLS-semantic-network-based document type definition (bottom, closing XML tags omitted for brevity).

RESULTS

The mean closeness for each of the twelve query prototypes ranged from 0.016 for query prototype four (a pharmacokinetics query) to 1.79 for query prototype eight (a drug-susceptibility query). The average, top ranked closeness measures ranged from 1.91 to 3.39 for the 12 query prototypes. The mean top-ranked closeness measure for sentences that had no consensus markup was not significantly different than for sentences with consensus markup (3.02 vs. 3.03).

Table 3 shows the performance of the vector-space model controlling for source of identified concepts. Using contextual information increased the number of correctly proposed markup instances seven-fold. While the "context concepts" only model was more accurate than the model that used only concepts from sentences, its kappa remained low due to the skewed distribution of query prototypes. Using concepts in each sentence and in HTML headings was the most accurate model and had the highest kappa. Vector length had no impact on the accuracy of the proposed markup in the combined concept model.

CONCLUSIONS

The type of fine-grained indexing that the MYCIN II system requires can provide highly precise full text retrieval. However, constructing the index, even with computer assistance, is prohibitively time-consuming. We designed an artificial intelligence method combining natural language processing techniques and statistical inference to enable our indexing tool to suggest appropriate markup for any sentence of full text. The method leveraged domain knowledge (in the UMLS) as well as knowledge contained in the document structure (from HTML headings) and new statistical techniques in a text vector-space indexing model.

The markup proposed by our automated indexing method was nearly as consistent with a consensus markup as two human indexers were with each other. The method was able to select the correct query prototype by closeness to the text vector in over two-thirds of instances compared with the gold standard. Furthermore, the system ranked the correct query prototype first or second about 80% of the time.

Both contextual concepts and those identified locally in sentences were essential to predicting markup accurately. Without contextual

Table 3. Query prototype prediction using query-document-vector closeness.

Concepts from					
Contexts	Sentences	Model	Rank 1 n (%)	Rank 1 or Rank 2 n (%)	Kappa*
Yes	No	With Normalization	30 (25)	53 (44)	0.03
No	Yes	With Normalization	12 (10)	15 (13)	0.05
Yes	Yes	Without Normalization	85 (71)	94 (79)	0.67
Yes	Yes	With Normalization	85 (71)	94 (79)	0.67

*Top-ranked query prototype from each model vs. gold standard agreement.

information, the algorithm performed poorly, as do some information retrieval systems that also ignore context. Vector-space models can have myriad variations in the definition of vector weights or the calculation of closeness. We found no benefit to vector length normalization in our system, and continue to explore alternate concept weighting schemes.

The average highest-ranked closeness measure for text sections that had consensus markup was not different from those that did not. This suggests that while the vector-space model might be adequate to select the best query prototype to index, other types of automated methods should be used to determine if markup is appropriate at all. One such technique we are investigating is information extraction through the use of a domain-specific grammar. Information extraction can yield highly precise results, although generating grammars to achieve high levels of high recall is frequently laborious. Nevertheless, if a sentence of full text fit the grammar, the tool would indicate to the indexer that markup is strongly suggested.

Our experiments have several limitations. We only examined the ability of the tool to select query prototypes for indexing, not individual concepts and values that comprise these prototypes. The methods the tool uses to identify these concepts are accurate in about two-thirds of cases.[5] In our next experiments we will evaluate how accurately the tool proposes both query prototype(s) and concepts for each instance of markup, which could differ from the results we have presented. In addition, we only evaluated the tool compared with two human indexers. Further comparisons with a more diverse selection of indexers would strengthen our findings.

Our results suggest an automated markup proposal system can perform well on full text medical information. We anticipate this automation will greatly speed the indexing process to the point where rapid, accurate indexing of large full-text documents for electronic publishing becomes feasible.

Acknowledgements

Supported by the Veterans Affairs Office of Academic Affairs and Health Services Research, Development Service Research funds and the Office of the Chief Information Officer. We wish to thank Lexical Technologies, Victor L. Yu, MD of the Univ. of Pittsburgh, Dept. of Infectious Disease, Lawrence Fagan, MD PhD, Stanford Medical Informatics, Andrew Kehler, PhD, SRI, Menlo Park, CA, Russell Cucina MD, and Ms. Mary Kate Wahl, Stanford Univ.

References

1. Funk ME, Reid CA. Indexing consistency in MEDLINE. Bulletin of the Medical Library Association 1983;71(2):176-83.
2. Pratt W. Dynamic organization of search results using the UMLS. Proc AMIA Symp 1997:480-4.
3. Berrios DC, Kehler A, Kim DK, Fagan LM, Yu VL. Automated Text Markup for Information Retrieval from an Electronic Textbook of Infectious Disease. Proc Amia Symp 1998:975.
4. Dugan JM, Berrios DC, Liu X, Kim DK, Kaizer H, Fagan LM. Automation and integration of components for generalized semantic markup of electronic medical texts. Proc AMIA Symp 1999:736-40.
5. Berrios DC, Kehler A, Fagan LM. Knowledge Requirements for Automated Inference of Medical Textbook Markup. Proc Amia Symp 1999:676-80.
6. Purcell GP, Shortliffe EH. Contextual models of clinical publications for enhancing retrieval from full-text databases. In: Gardner RM, editor. 19th Annual Symposium on Computer Applications in Medical Care; 1995; New Orleans, LA: Hanley & Belfus, Inc.; 1995. p. 851-57.
7. Salton G, McGill MJ. Introduction to modern information retrieval. New York: McGraw-Hill, Inc.; 1983.
8. Hersh WR. Information Retrieval: A Health Care Perspective. New York: Springer-Verlag; 1995, pp 143-5.
9. Victor Yu, Thomas Merigan, Barriere S. Antimicrobial Therapy and Vaccines. Philadelphia: Lippincott Williams & Wilkins; 1998.