Finding Cue Expressions for Knowledge Extraction from Scientific Text: Early Results

Masashi Shimbo, Sayaka Tamamori, and Yuji Matsumoto

Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0192, Japan
{sayaka-t,shimbo,matsu}@is.naist.jp

Abstract. This paper investigates whether and how natural language processing and data mining techniques can be utilized for locating desired knowledge in a large text collection. This task amounts to finding cue words and phrases indicating the location of knowledge, where the challenge is to establish a methodology that can cope with the diversity of expressions. We examine the feasibility of mining cue expressions from the syntactic dependency structure obtained from parsed sentences. As a case study, the (phrasal) expressions concerning a variety of tests related to chronic hepatitis were sought in the Medline abstracts. We observed that dependency analysis helped to narrow down the candidates for verbal expressions, although it was ineffective for other types of expressions.

1 Introduction

With the recent growth in the number of text collections available in digital form, there has been increased interest in mining useful knowledge buried in a volume of text. In particular, knowledge extraction from medical literature is appealing from the standpoint of evidence-based medicine (EBM) [13], which practices "integrating individual clinical expertise with the best available external clinical evidence from systematic research" [12]. A source of external evidence is assumed to be clinically relevant research literature, and thus EBM is an immediate application of knowledge extraction from medical text.

The Medline database [16], available from the U. S. National Library of Medicine covers over 11 million bibliographic citations from more than 4 thousand research journals world-wide. It has been a standard corpus for medical knowledge extraction, as a large number of citations contain abstracts as well.

Previous work on knowledge extraction from Medline includes as follows. Blaschke et al. [2] extracted protein interaction relationships using the simple cue patterns of the form 'PROTEIN VERB PROTEIN,' where VERB includes 14 verbs indicating actions, such as *activate*, *bind*, and *suppress*. Rindflesch et al. [11] uses the specific predicate *bind* as a cue, and extracted binding relationships between macromolecules. Khoo et al. [7] attempted to identify the location of causal relationship description using the dependency subtree patterns.

Cue patterns, which work as an indicator of the location of desired knowledge, depend on the domain of text as well as the type of desired knowledge. Hence the first

step in knowledge extraction is to find effective cue patterns suitable for the domain and goal at hand.

However, no previous work has, to our knowledge, addressed how cue patterns can be efficiently identified. In all the literature cited above, cue patterns are given a priori, presumably devised by domain experts for the prescribed tasks[1]. It is true that the current technology does not admit finding effective cue patterns without human supervision, yet it should still be possible to narrow the number of candidate patterns from which human experts can sift with less efforts.

Moreover, most previous work views knowledge extraction from text as a cascaded process. In practice, it is rather a process involving a feedback; it iterates the subprocesses of identifying cue patterns, matching them to text, and evaluating the feasibility of matched passages. If cue patterns are too general, they should generate too many irrelevant passages to be inspected by humans; if they are too restrictive to the contrary, they should generate too few. In either case, cue patterns must be revised and the whole process must be reiterated.

Motivated by the above observations, we pursue a methodology to help human experts identify cue patterns effectively. As a case study, the problem of finding cue patterns in the domain of diagnosis tests for hepatitis is addressed.

2 Methodology

The major obstacle in collecting cue patterns is the diversity of semantically equivalent expressions. Consider retrieving Medline to see whether gradual increases in ADA level correlate with a certain change in the condition of a patient with chronic hepatitis. It is desirable to know typical expressions used for representing increases in ADA level, because it would reduce the volume of text that should be examined. However, there are a variety of verbs representing value increase in English, such as *increase*, *raise*, and *elevate*, to name a few. In addition, the increase may not be represented with a verb.

Hence, we would like to enumerate as many expressions potentially relevant to the user's objective as possible, yet without imposing on the domain experts a significant increase in the load to sift through the enumerated expressions. This goal leads to a trade-off. Increasing the number of patterns for a better cover rate leads to an enormous number of candidate passages.

Another challenge is how to present the enumerated passages to the domain experts. Since the number of passages are often huge, it is desirable to present only the relevant portion, rather than the whole sentence or abstract. The question remains on how such *relevant* portions can be determined.

To address these issues, we use syntactic dependency structure trees for representing cue patterns. A dependency tree bears information richer than the original sentence viewed simply as a string or a bag of words. Exploiting the structure within the tree allows us a fine-grained control over determining the relevant portions to be presented to the domain experts.

[1] On the other hand, Thomas et al. [15] indicated explicitly that they collected common ways of describing protein interactions through the analysis of 200 abstracts by hand.

In this paper, we concentrate on processing at the sentence and sub-sentence levels, and do not deal with knowledge described over two or more sentences. This decision reflects the reported effectiveness of the text processing units in a text mining task [5].

3 Enumerating expressions relevant to hepatitis

3.1 Objective and applications

The long-term goal of our project is to help screening the association rules mined separately from time-series data on hepatitis-related tests.

Since data mining techniques typically output a number of association rules most of which do not make sense nor are novel, the cost of sifting these generated rules is often prohibitive. Even if a rule is supported by a vast amount of data, it may just represent a piece of common-sense knowledge, or it may already be known to the public by prior work thus having no novelty [8, 10]. Such knowledge, if obtained from published papers or their abstracts, should make it possible to filter out those 'uninteresting' rules.

Note that although there is a volume of literature (e.g., [1, 6, 9]) in the data mining community addressing the *interestingness* of mined rules, whether a rule is publicly known or not cannot be detected by these techniques, as they rely on statistical tests based on the same data used for mining rules. Whether a rule is covered by prior work can only be determined with reference to the work, which are generally published in the form of text documents.

3.2 Identifying expressions through syntactic dependency structure analysis

The issues discussed in Section 2 are closely related to the language used for representing cue patterns. As we mentioned earlier, we use a syntactic dependency tree but there are other possible choices, including

- Contiguous n words: frequent series of n words.
- Non-contiguous word sequences: frequent sequences of (non-contiguous) words.

However, the contiguous n-word representation is inflexible in that it cannot absorb the variations arising from insertion and omission of modifiers, while non-contiguous word sequences are prone to generate meaningless sequential patterns that consist only of individually frequent words, thus requiring extra post-processing to filter out these patterns.

On the other hand, the syntactic dependency tree, which is a form of syntactic parse trees, (i) allows modifiers to be easily removed by exploiting the structure of the tree, and (ii) indirect and direct dependence between words are represented as the locality in the dependency tree, and therefore meaningful portions of the sentences are easier to extract.

Based on these arguments, we use a dependency structure as our language for representing patterns.

4 Procedure for identifying cue expressions

We are interested in correlations among the outcomes of clinical tests related to hepatitis and the conditions of the disease. Hence we focus on the pattern of the form 'NP_1 V NP_2' or 'NP_2 V NP_1', where NP_1 contains the name of a clinical test, and V is a verbal expression (base verb phrase), and NP_2 is another noun phrase, presumably containing other diagnostic tests and the conditions of patients. Syntactic dependency analysis of a sentence admits extraction of these phrases, as it reveals the hierarchical structure among words within a sentence.

Our procedure for identifying cue patterns can be decomposed into four steps:

1. Keyword-based filtering of sentences.
2. Dependency structure analysis of the filtered sentences.
3. Expression extraction from syntactic dependency trees.
4. Filter and rank extracted expressions and hand over to the domain experts for further review.

The rest of this section will delineate each step.

4.1 Step 1: keyword filtering

Since syntactic dependency parsing is a computationally intensive task, it is not feasible to apply this process to the whole text collection. Hence we restricted the candidate sentences by using simple keyword matches.

We first filter abstracts containing the word *hepatitis* from the corpus. We then segment these abstracts into sentences, and further filter the sentences containing the names of the diagnostic tests of our interest. The keywords used for filtering are the names of the 660 clinical tests for diagnosing hepatitis, and are the same as the features used in [10] for mining association rules from time-dependent data. They consist of 503 different diagnosis tests[2], such as *glutamic pyruvic transaminaze*, and *glutanic oxalacetic transaminase*. The rest is their synonyms and abbreviations, e.g., *GPT* and *GOT*.

4.2 Step 2: dependency structure analysis

There are several ways to obtain syntactic dependency structure trees. In this paper, we take the same method as used in our previous work [14]. We first apply a phrase structure parser to the sentences filtered in Step 1 to obtain phrase structure trees. Charniak parser [3] was used as the phrase structure parser. This parser boasts approximately 90% accuracy at the phrase structure level, when applied to the Wall Street Journal corpus. The dependency structure trees are then obtained by extracting word dependencies from the phrase structure trees.

We illustrate the translation process using the phrase structure subtree in Fig. 1. This tree will eventually be translated into the dependency structure tree depicted in Fig. 3.

Each non-leaf node in a phrase structure tree is labeled with a syntactic category, and each leaf node is labeled with a surface word. In Fig. 1, syntactic categories are typeset

[2] Provided by courtesy of Chiba University Hospital and Shizuoka University.

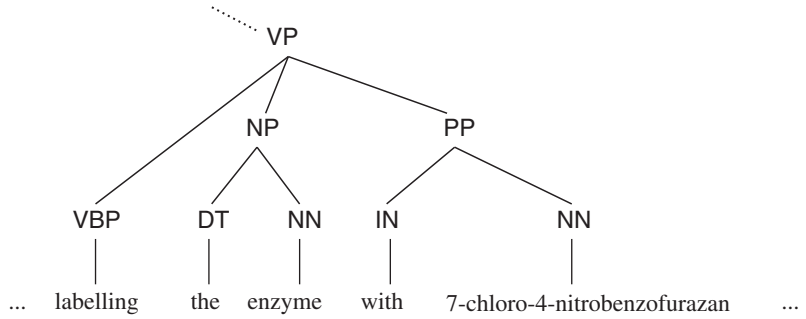

Fig. 1. Phrase structure subtree. Leaf nodes correspond to surface words, and each non-leaf node is labeled with a syntax category.

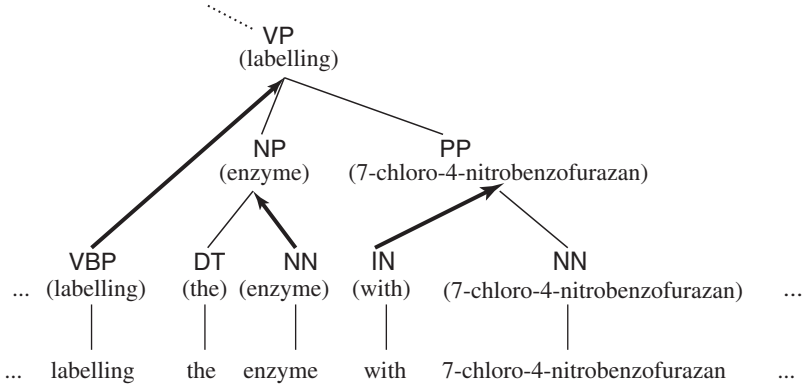

Fig. 2. Phrase structure subtree labeled with headwords. Bold arrows depicts the inheritance of head words by the head rules, and inherited head words are shown in parentheses.

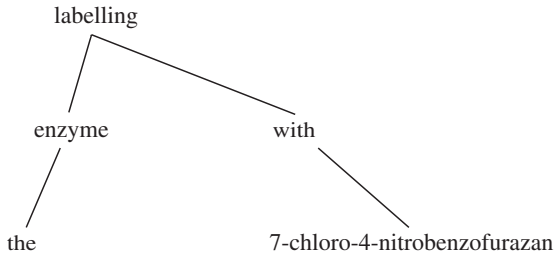

Fig. 3. Dependency tree translated from the phrase structure tree in Fig. 1.

in sans serif; e.g., NP (noun phrase), VP (verb phrase) and PP (prepositional phrase). A parent-child relationship in phrase structure trees corresponds to the application of a context-free grammar rule. To be precise, given a parent node and its n children, let u be the syntactic category of a parent node, and $v_1, \ldots, v_n$ be the syntactic categories (or surface words) of the n children. This relationship represents the application of a context-free (CFG) grammar rule $u \rightarrow v_1, \ldots, v_n$.

To translate a phrase structure tree (PST) into a dependency structure tree, we first label each *non*-leaf node in the PST with a surface word. This makes every nodes in the tree to be associated with a surface word, henceforth called the *head (word)* of the node. The head word of a non-leaf node is inherited from a child of the node. If node u has two or more children, the so-called *head rule*[3] as associated with each CFG rule determines from which one of the n children v_i, $1 \leq i \leq n$ the head word should be inherited. The head rule uniquely determines the index i of the children v_i (called the *head constituent*) from which the head word should be inherited to the parent node u.

Fig. 2 shows the result of headword labeling scheme applied to the PST in Fig. 1. The inherited head words are shown in parentheses below the syntactic category, and a bold arrow represents the inheritance of a head word. For example, the arrow from VBP to VP denotes the head constituent is VBP, but not NP or PP for the CFG rule $\mathsf{VP} \rightarrow \mathsf{VBP}, \mathsf{NP}, \mathsf{PP}$.

After all nodes are labeled with head words in the phrase structure tree, its dependency structure tree is extracted by recursively coalescing head constituent nodes with their parents until no more coalescing can be performed.

In Fig. 2, this process corresponds to coalescing every parent-child pair connected with a bold arrow. The node after coalescing inherits the same head word as nodes being coalesced, which should have had the same head given that the child is the head constituent.

The parent-child relationship in the dependency structure tree thus obtained (Fig. 3) represents the head word of a child (directly) depends on the head of the parent; and we say a node u depends *indirectly* on another node v, if v is an ancestor of u but is not its parent. For instance, in Fig. 3 the determinant *the* depends directly on *enzyme*, and indirectly on *labeling*.

4.3 Step 3: extracting expressions relevant to diagnostic tests

Given the dependency structure trees, we extract the noun phrase containing the names of the clinical tests, the verbal expression, and other phrases depending on the same verbal expression from each tree. We illustrate this step with the dependency tree in Fig. 4. This figure depicts the dependency tree of the sentence 'A stepwise increase in serum ADA level was observed with increasing severity of liver cirrhosis.'

Given a sentence $S = w_1 w_2 \cdots w_n$ consisting of n words, let c_i ($i = 1, \ldots, n$) be the syntactic category of w_i, i.e., the syntactic category assigned to the parent of the leaf node corresponding to w_i in the PST for S. Let $T(S)$ denote the dependency tree of S. We identify the index i for the i-th word in S as the node corresponding to the word in the tree. Let the predefined set of the diagnosis test names be D.

[3] We used the head rules due to Collins [4].

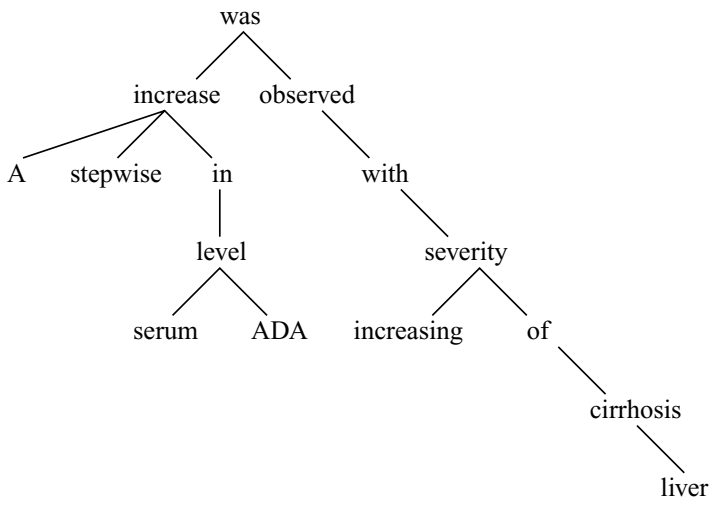

Fig. 4. Syntactic dependency tree of 'A stepwise increase in serum ADA level was observed with increasing severity of liver cirrhosis.'

1. First, locate the names of the tests occurring in a given sentence and register their locations in the set P; i.e., $P \leftarrow \{j \mid w_i w_{i+1} \ldots w_j \in D\}$. When the test name consists of multiple words (i.e., $i \neq j$ in the above formula), the location j of the last word is registered. This heuristic reflects the fact that in most cases, the last constituent word is the head of the noun phrase.
   In Fig. 4, we find $w_6 = \mathit{ADA} \in D$ in this step. Hence, $P = \{6\}$.
2. For each node $p \in P$ denoting a test (henceforth called the *pivot*), do:
   (a) Extract noun phrases containing the names of the diagnostic tests.
      i. Let the set $\mathrm{NP}_1 \leftarrow \emptyset$.
      ii. Starting from the pivot $p \in P$, ascend the tree $T(S)$ towards the root until a node is reached such that it is either (1) a verb, (2) an auxiliary verb, or (3) a preposition whose parent node is a verb or an auxiliary verb. Let v be such a node. Put the node encountered along the way to the set NP_1.
      In Fig. 4, we see that the traversal in this step ends when $w_v = \mathit{was}$ is encountered, at which point NP_1 contains the indices for *ADA*, *level*, *in*, and *increase*.
      iii. Rearrange nodes in NP_1 in the order of their indices, concatenate the words corresponding to the nodes in that order.
      In the example, this yields the noun phrase *increase in ADA level.* Note that the modifiers, i.e., *a* and *stepwise*, are excluded from the extracted pattern. This helps to absorb slight difference in modifiers, and also facilitate the reviewing process by the domain experts.
   (b) Extract verbal expressions.
      i. Let node v denotes a verb, auxiliary verb, or preposition that stopped the traversal in Step 2(a)ii. Let the new set $\mathrm{VP} \leftarrow \{v\}$. In this example, the index for *was* enters the list.

ii. From v, ascend the tree $T(S)$ towards the root while the current node is a verb, an auxiliary verb, or a preposition, and while the node is non-root. Add the encountered nodes along the way in VP. Since *was* is the root of the tree, this processing ends immediately in Fig. 4.

iii. If the reached node is the root of the dependency tree, descend the tree beginning from the child nodes of the root which are either a verb, an auxiliary verb, or a preposition, putting the words encountered in VP. In the figure, two words *observed* and *with* enter VP.

iv. Sort words corresponding to the nodes in VP in the order of their appearance in the original sentence.

This processing yields the verbal expression *was observed with* in the example of Fig. 4.

(c) Extract phrases depending on the verbal expression.

Traverse down from the child nodes that directly depends on the verbal expression. However, we only take into account the children k where the word w_k occurs at the opposite side of the pivot word with respect to the verb w_v.

To be precise, let p and v be the indices for the pivot, and the verb located in Step 2(a)ii. if $p < v$, i.e., pivot word w_p occur before the verb w_v in S, then traverse only from the children $k \in R$ where $R = \{k \mid \mathrm{Child}(v) \text{ and } v < k\}$. To the contrary, if $i > j$, then traverse only from the elements in $R = \{k \mid \mathrm{Child}(v) \text{ and } k < v\}$.

Let $\mathrm{NP}_2 \leftarrow \emptyset$. For each $k \in R$, traverse all descending paths emanating from k, on condition that traversal should be cut off immediately when a conjunction, or an interrogative is encountered. Put all the obtained paths to NP_2.

In effect, two phrasal expressions are obtained from the tree in Fig. 4, i.e., *increasing severity*, and *severity of liver cirrhosis*.

4.4 Step 4: sorting obtained expressions for review

Finally, the collected expressions should be ranked and reordered according to some criterion to be subsequently reviewed by the domain experts. In this paper, the extracted expressions are simply sorted by the frequency of occurrences. In the future work, we will pursue the use of more sophisticated ranking methods based on statistical measures, and also to extract frequent sub-patterns in the extracted expressions.

5 Experimental results and discussions

We applied the method of Section 4 to the abstracts contained in the Medline 2003 database. In Step 1 of the extraction procedure, 57,987 abstracts contained the word *hepatitis*. From these abstracts, 130,306 sentences were identified as containing the names of the hepatitis-related diagnostic tests. We applied the procedures of Steps 2 and 3 to these sentences and extracted the noun phrases and the verb phrases.

Table 1 shows the 20 most frequent expressions containing the noun phrase containing diagnostic tests, filtered by humans from a total of 91,427 different noun phrases

Table 1. Expressions representing change in the test results

Rank	Frequency	Phrase
52	113	iron overload
59	100	positive for HCV RNA
73	84	detection of HCV RNA
84	72	presence of HBV DNA
121	54	iron concentration
155	43	HCV-RNA negative
157	43	HCV seropositivity
186	37	clearance of HCV RNA
243	29	loss of HCV RNA
261	27	iron deposition
267	27	copper concentrations
288	26	copper accumulation
309	25	dose of interferon
395	21	iron depletion
527	16	copper excretion
575	15	CT findings
586	14	disappearance of HCV-RNA
715	11	low density
717	11	iron reduction
718	11	interferon plus

Table 2. Verbal expressions representing causal relationships

Rank	Frequency	Expression
7	1664	was detected in
13	709	was found in
17	615	revealed
18	611	correlated
24	518	developed
32	400	was associated with
38	372	demonstrated
47	316	was observed in
51	294	occurred
52	294	induced
65	235	show
70	223	suggest
76	215	report
93	193	resulted in
119	153	indicate
129	139	causes
137	129	represents
143	126	seems to be
144	126	performed
149	122	was related to

Table 3. Expressions representing diseases, symptoms, conditions, etc.

Rank	Frequency	Expression
19	428	chronic hepatitis
21	397	HCV infection
48	245	liver disease
75	181	risk factors
133	134	hepatocellular carcinoma
145	127	chronic infection
246	89	liver cirrhosis
252	87	active hepatitis
258	86	acute hepatitis
333	71	anti-HCV positive
384	62	liver damage
427	57	cause of liver disease
433	56	viral hepatitis
545	47	chronic carriers
597	43	non-A hepatitis
601	43	liver injury
811	33	severe disease
818	33	detection of HCV RNA
903	30	chronic hepatitis cirrhosis
1096	25	inhibitory effect

obtained in Step 1 of Section 4.3. Likewise, Table 2 shows the list of the verbal expressions representing some form of relationship, interactions and actions, filtered from the 37,780 verbal expressions obtained through Step 2 of Section 4.3. Finally, Table 3 shows the list of diseases, symptoms and conditions filtered from 251,409 noun phrases that depend on the verbal expressions (Step 3).

To summarize, we had to inspect the top 150 extracted verbal expressions to collect 20 expressions of interest (Table 2). On the other hand, for noun phrases containing the test names (Table 1) we needed to inspect more than 700 patterns to collect 20 meaningful patterns, and for diseases, symptoms, and conditions we had to examine over 1000 expressions (Table 3). The latter two cases impose an enormous load to the human inspector.

The possible remedies to further reduce the number of expressions are as follows.

- First filter sentences using verbal expressions as cue, and then extract the rest from the survived sentences.
- Use existing dictionaries and thesauri to restrict the variations in expressions.

As we claimed previously, the subtree representation allows fine-grained control over how the found patterns can be presented to the domain experts (but not fully discussed or demonstrated in this paper). Note however that this claim applies only to inspecting the validity of cue patterns coarsely, but not to the eventual knowledge inspection that should also be conducted by the domain experts. Specifically, although it is possible to present the matched pattern of the form 'NP VBP NP' obtained with the method of Section 4.3 omitting the modifiers not dependent on test items, it is not

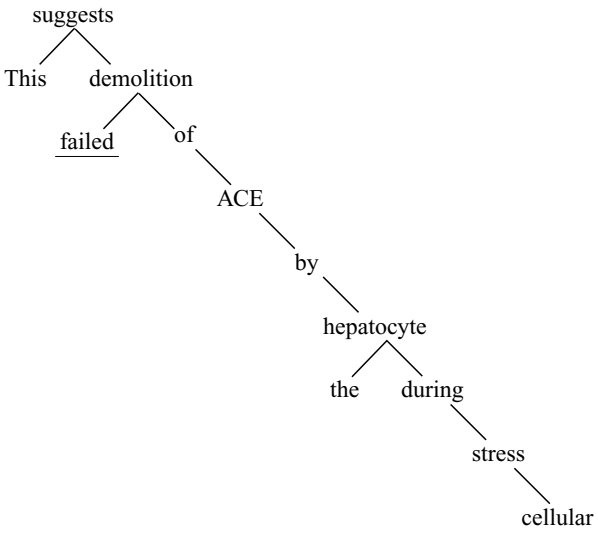

Fig. 5. The dependency tree of the sentence "This suggests failed demolition of ACE by the hepatocyte during cellular stress." The adjective *failed* is not an ancestor of the pivot word *ACE*.

always feasible to show only this portion because important modifier words can be missed out. Consider the dependency tree depicted in Fig. 5. If we apply the method of Section 4.3 to this tree, it is possible to extract the noun phrases, *demolition of ACE by the hepatocyte* and *demolition of ACE by hepatocyte during cellular stress.* However, since the word *failed* does not have a direct or indirect dependency relation with the pivot word *ACE*, it never enters the list of collected noun phrases. Since the omission of *failed* leads to the opposite meaning, the above portion is not an acceptable form of knowledge representation. It is, on the other hand, completely acceptable to omit *failed* when sifting cue patterns is concerned, as addressed in this paper.

6 Conclusions

The first step in knowledge extraction from large text data is locating relevant passages. This paper discussed how cue patterns for locating passages can be discovered efficiently. We used syntactic dependency parsing to obtain frequent patterns in the three categories:

1. Noun phrases containing diagnostic tests.
2. Verb expressions representing a relationship, interaction, or action.
3. Symptoms and conditions of hepatitis and other diseases.

The proposed method yielded a better result (a smaller number of candidates) for the second class (verbal expressions), compared with the rest. The first class (non phrases containing diagnostic tests) and the third class (symptoms and conditions of diseases) required a vast amount of human reviews to filter results, and was not satisfactory.

Our future research includes developing efficient methods to further sift through these candidate patterns. We also plan to apply the techniques of collocation identification and tree mining to the extracted expressions, in order to obtain more compact representation of the expressions. Another issue to be addressed is to discriminate words which test names do not directly or indirectly depend on but are still important, such as the adjective *failed* in the example previously mentioned.

References

[1] C. C. Aggarwal and P. S. Yu. A new framework for itemset generation. In *Proceedings of the 17th ACM Symposium on Principles of Database Systems (PODS'98)*, pages 18–24, Seattle, WA, USA, 1998.

[2] C. Blaschke, M. A. Andrade, C. Ouzounis, and A. Valencia. Automatic extraction of biological information from scientific text: protein-protein interactions. In *Proceesings, Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 60–67, Heidelberg, Germany, 1999. AAAI Press.

[3] E. Charniak. A maximum-entropy-inspired parser. In *Proceedings of the Second Meeting of North American Chapter of Association for Computational Linguistics (NAACL-2000)*, pages 132–139, 2000.

[4] M. Collins. *Head-Driven Statistical Models for Natural Language Processing*. PhD dissertation, University of Pennsylvania, 1999.

[5] J. Ding, D. Berleant, D. Nettleton, and E. Wurtele. Mining MEDLINE: abstracts, sentences, or phrases? In *Pacific Symposium on Biocomputing 7*, pages 326–337, Kaua'i, Hawaii, 2002.

[6] S. Jaroszewicz and D. A. Simovici. A general measure of rule interestingness. In *Principles of Data Mining and Knowledge Discovery: 5th European Conference, PKDD 2001, Proceedings*, Lecture Notes in Artificial Intelligence. Springer, 2001.

[7] C. S. G. Khoo, S. Chan, and Y. Niu. Extracting causal knowledge from a medical database using graphical patterns. In *Proceedings of the 38th Annual Meetings of the Association for Computational Linguistics*, pages 336–343, Hong Kong, 2000.

[8] Y. Kitamura, A. Iida, K. Park, and S. Tatsumi. Micro-view and macro-view approaches to discovered rule filtering. In *Proceedings of the Second International Workshop on Active Mining (AM 2003)*, pages 14–21, Maebashi, Gunma, Japan, 2003.

[9] S. Morishita and J. Sese. Traversing itemset lattices with statistical metric pruning. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS-2000)*, 2000.

[10] M. Ohsaki, Y. Sato, S. Kitaguchi, H. Yokoi, and T. Yamaguchi. Investigation of rule interestingness in medical data mining. In *Proceedings of the Second International Workshop on Active Mining (AM 2003)*, pages 85–97, Maebashi, Gunma, Japan, 2003.

[11] T. C. Rindflesch, J. V. Rajan, and L. Hunter. Extracting molecular binding relationships from biomedical text. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP-2000)*, pages 188–195, 2000.

[12] D. L. Sackett, W. M. C. Rosenberg, J. A. M. Gray, R. B. Haynes, and W. S. Richardson. *Evidence-based medicine: what it is and what it isn't*. Oxford Center for Evidence-Based Medicine, http://www.cebm.net/ebm_is_isnt.asp, 2004. Article based on an editorial from the British Medical Journal on 13th January 1996 (BMJ 1996; 312: 71–2).

[13] D. L. Sackett, S. E. Straus, W. S. Richardson, W. Rosenberg, and R. B. Haynes, editors. *Evidence-Based Medicine: How to Practice and Teach EBM*. Churchill Livingstone, second edition, 2000.

[14] M. Shimbo, H. Yamada, and Y. Matsumoto. Using syntactic dependency information for classification of technical terms. In *Proceedings of the 7th Pacific Rim Knowledge Acquisition Workshop (PKAW 2002)*, pages 131–143, Tokyo, Japan, 2002.

[15] J. Thomas, D. Milward, C. Ouzounis, S. Pulman, and M. Carroll. Automatic extraction of protein interactions from scientific abstracts. In *Proceedings of the 5th Pacific Symposium on Biocomputing*, pages 538–549, 2000.

[16] U. S. National Library of Medicine. MEDLINE. http://www.nlm.nih.gov/databases/databases_medline.html, 2003.