Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia - Social and Behavioral Sciences 27 (2011) 281 – 290

Procedia
Social and Behavioral Sciences

Pacific Association for Computational Linguistics (PACLING 2011)

Semantic Search based on the Online Integration of NLP Techniques

Katsuya Masuda[a*] , Takuya Matsuzaki[b], Jun'ichi Tsujii[c]

[a]Center for Knowledge Structuring, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
[b]Department of Computer Science, Graduate School of Information Science and Technology, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
[c]Microsoft Research Asia, Building 2, No. 5 Dan Ling Street, Haidian District, Beijing, 100080, P. R. China

Abstract

This paper introduces a framework for semantic information retrieval based on the integration of various natural language processing (NLP) techniques, each of which annotates a base text with different kinds of information extracted from the text. Instead of running the NLP modules on the fly for individual search requests, the NLP modules are applied to the text in advance and the results are indexed in a way that enables flexible and efficient integration of them. The query language is based on a variant of the region algebra, in which we can specify a sub-structure in the annotated text that may involve different kinds of annotations. Given a query, the retrieval engine searches for the sub-structure by aggregating the different kinds of annotations through a search algorithm for the extended region algebra. We demonstrate the effectiveness and flexibility of the proposed framework through experiments with TREC Genomics Track data.

Keywords: Information Retrieval, Semantic Search, Tag Annotations

1. Introduction

[*] Corresponding author. Tel.: +81-3-5841-0894; Fax: +81-3-5841-0749
E-mail address: masuda@cks.u-tokyo.ac.jp

1877-7058 © 2011 Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
Selection and/or peer-review under responsibility of PACLING Organizing Committee.
doi:10.1016/j.sbspro.2011.10.609

NLP techniques such as named-entity recognition (NER) and syntactic/semantic analysis have been found useful in various text mining applications. However, a traditional way of utilizing those techniques has been to apply them to a subset of text drawn from a base textset (e.g., from the Web) after collecting them by using a simple query such as keywords. This is presumably due to the lack of an effective and efficient method of utilizing NLP results during the search for the textset of interest. Meanwhile, there is an emerging trend of enriching texts with various kinds of information in the form of tag annotations, and using them for information services [1, 2, 3], or for querying texts with annotations [4, 5]. The trend has brought on a great prospect in which sophisticated NLP techniques are applied in advance to make abstract levels of linguistic representation explicit, which are used to deduce more user-oriented information on the fly.

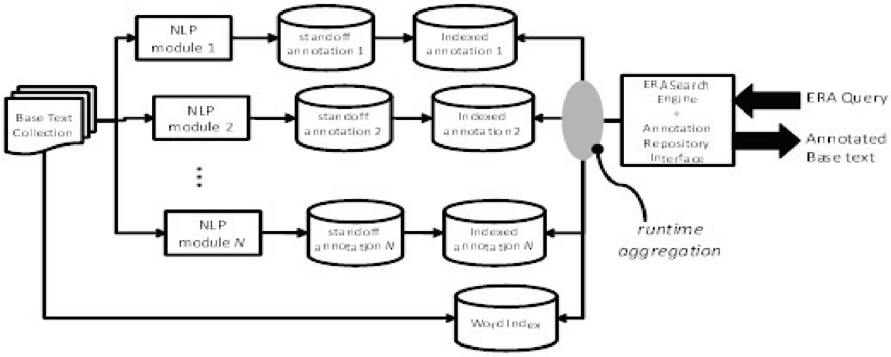


Fig. 1. Overview of our search architecture

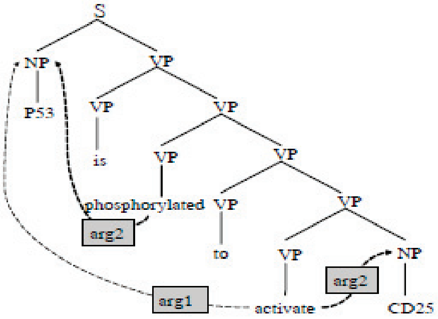


Fig. 2. Syntactic/semantic structure

In this paper, we propose a new search architecture for text and stand-off NLP annotations. A set of annotation data produced by an NLP module can be freely added or updated in our architecture without reconstructing the whole database. Figure 1 shows the overview of the architecture. Our search algorithm is based on a variant of the region algebra [6], which we further extended to handle annotations.

2. Background

2.1. Region Algebra

The Region Algebra [7] is a framework for searching annotated text with partially overlapping annotations. The framework is more suitable for stand-off annotations than XML databases because it can be applied to annotations that cross each other easily.

Region algebra consists of binary operators on region sets as shown in Table 1, and basic expressions for region sets of words and annotations. For example, '(> [sentence] "search")' represents a set of text spans annotated as sentences containing the word "search." In the current paper, we use [X] to denote the set of regions annotated as X and a quoted string ”Y” to denote the set of word regions of Y. The search algorithm is defined by using the basic functions which searches a nearest region matching the query from the input position. These functions are recursively defined by using the functions of the subquery. We extended the region algebra to efficiently search for a sub-structure in a tag-annotated text in the previous work [6]. The extension includes a new algorithm to handle nesting of the same type of annotations and value-equivalence conditions among attribute values, by which we can handle nesting structure of parse trees and query for a semantic dependency structure among words and phrases that are difficult to represent only with the operators on region sets.

```
0  7  s    id="1"   | <s id="1">
0  2  np   id="2"   | <np id="2">He</np>
3  7  vp   id="3"   | <vp id="3">runs</vp></s>
```

Fig. 3. Stand-Off annotation (left) and its inline representation (right)

Table 1. Operators of the region algebra

(> A B)	A containing B
(< A B)	A contained in B
(& A B)	A and B
(| A B)	A or B
(− A B)	Starting with A and ending with B

We added a small but important improvement to the search algorithm that allows asynchronous indexing of different types of annotations. This mechanism is essential for the ever-extending nature of the annotated textbase, to which we continue to add new annotations produced by novel NLP techniques. To make it possible, we chose a simple stand-off annotation as the common data format for the input and the internal representation of the annotations. In the stand-off annotation format, a tag is represented as a tuple <b, e, t, as> where b and e are respectively the begin and end position of the tagged text span, t is the name of the tag, and as is a set of attribute-value pairs. Figure 3 shows an example of stand-off annotation and corresponding XML-like inline representation. Thanks to the stand-off representation, we don't have to make the inline representation that includes all annotations for the indexing, and can build an index file for each NLP module individually. Another advantage of the stand-off representation is that it can naturally represent crossing annotations, while an additional mechanism is necessary to do it in an XML-like inline format.

2.2. NLP modules for relational concept search

We give an overview of the NLP modules that used in the experiments. Table 2.1 shows a list of NLP modules and corresponding annotation tags. We describe the detail of each NLP module in the followings.

HPSGparser. A wide-coverage HPSG parser, Enju [8], was used for syntactic and semantic analysis of the sentences. The parser is so-called ‘deep parser’, which provides both the syntactic and semantic structures represented in a constituent tree and the semantic analysis represented as predicate-argument relations. Fig-ure 2 illustrates an example of Enju’s output with a syntactic tree structure and semantic predicate-argument relations among words overlayed on it. By using the results of deep parsing, we can directly search for a semantic relation represented differently in various surface structures.

Named Entity Recognizers. While the parser is used to map the surface word sequence to a relational struc-ture among words in the syntactic/semantic level, a few named entity recognizers are used to map entity names appeared in the text to a concept identifier defined in a taxonomy. Specifically, we used a statisti-cal gene/protein name recognizer trained on the GENIA corpus [9] and a dictionary-matching based term recognizer for various term types including, disease, symptom, enzyme and drag names, which are very often mentioned with different textual expressions (i.e., synonyms), such as ”P53”, ”p53”, ”p-53” and ”p53 protein”, or ”cancer” and ”carcinoma.” Those synonymous expressions are tagged with a unique concept identifier. Thus the search can be done in the level of concept, not in the surface textual expressions.

Table 2. NLP module and tags

NLP module	Tags
HPSG parser	sentence, cons, tok
Named entity	entity_name
Event expression	event expression
Biomolecular event	Event, Trigger
Gene-disease association	GDA

event expression tagger. We also annotated the text with the results of event expression tagger, which maps a semantic relation given by the parser to a pre-defined event class such as ‘positive regulation relation.’ Roughly speaking, this is a normalization of verbal expressions to a event types. For example, those expressions like ‘A activates B’, ‘A induces B’ etc. are all mapped to a event of ‘positive regulation (Agent = A, Theme = B).’ The tagger applies event-expression patterns extracted from GENIA event corpus [10] against the output of the parser and identify a event type and the arguments of the event. Since the patterns do not include any restrictions on the class of the arguments (e.g., gene, protein, disease), the results cover a wide range of relations. e.g., “a gene causes a disease,” “a protein induces a chemical reaction in a cell” and “a chemical substance causes death” are all recognized as an instance of ‘positive regulation’ event.

Specialized Relation Recognizers. In addition to the event expression tagger explained above, we used two systems that are respectively specialized to the recognition of gene-disease association [11] and biomolecu-lar events mentioned in the text [12]. Both systems uses machine-learning techniques to identify the specific types of relations based on the features extracted from the parsing results. While the gene-disease association is defined as a single binary relation between a gene name and a disease name, the biomolecular events can have more complex structure, such as “binding of protein A to protein B inhibit another type of binding be-tween protein C and protein D,” which is represented as a nested data structure, ‘negative regulation(Cause = binding(protein-A, protein-B), Theme = binding(protein-C, protein-D)).’

In addition to the above mentioned modules, we used low-level NLP modules like a sentence splitter and POS tagger as well as a more high-level processing module like a sentence rhetorical role tagger [13]. As shortly explained, the flexibility of the extended region algebra allows a query that simultaneously specifies structures in all layers of annotations, e.g., semantic structure, syntactic structure, and rhetorical structure, by combining subqueries for each layer with &-operator and tying them together by using variables.

3. Method

3.1. Ranking algorithm

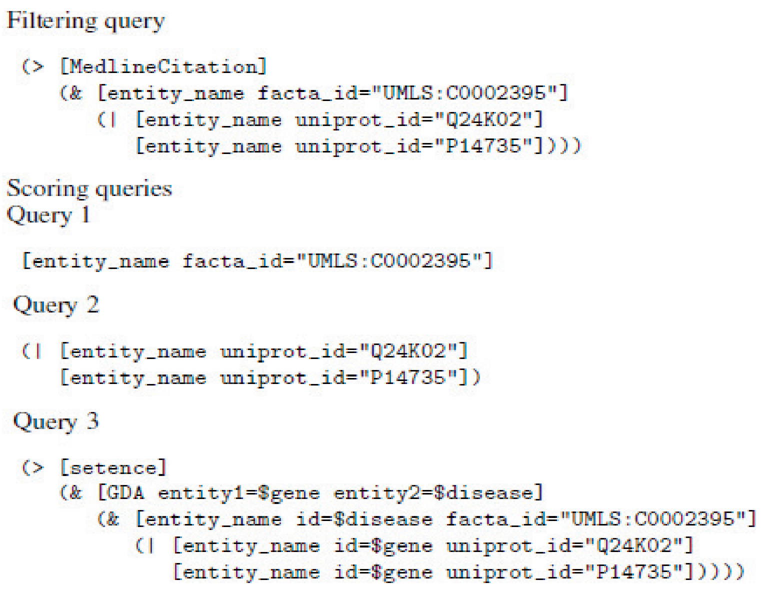
We used two types of queries to construct a ranking list of documents, one is a Filtering Query, which is a boolean query containing essential conditions to filter out the documents not concerning to the topic and the other is one or more Scoring Queries for calculating the scores of a document. The algorithm to construct a ranking list is as follows: First, we searched documents with the Filtering Query. Then, a score is calculated for each document based on whether or not the document matches a Scoring Query and a ranking list is constructed based on the scores. Given a list of Scoring Queries $q_1, ..., q_n$, the score of a document D is calculated with Okapi BM25 [14] simply extended for regions instead of terms, that is,

$$S(D, q_1, ..., q_n) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{rf_{q_i} \cdot (k_1 + 1)}{rf_{q_i} + k_1 \cdot (1 - b + b \cdot \frac{|D|}{DL_{ave}})}$$

where r fqi is the number of regions matching to the query qi in the document D, k1 and b are parameters,(we used k1 = 2.0 and b = 0.75 in the experiments), |D| is the length of the document D, DLave is the average

Table 3. Topic types

No.	Information Need
1	Description of standard methods or protocols
2	Description of the role of a gene involved in a disease
3	Description of the role of a gene in a biological process
4.1.	Description of interactions between two genes in the function of an organ or in a disease
5	Description of mutations of a gene and its biological impact



Filtering query

```
(> [MedlineCitation]
   (& [entity_name facta_id="UMLS:C0002395"]
      (| [entity_name uniprot_id="Q24K02"]
         [entity_name uniprot_id="P14735"])))
```

Scoring queries
Query 1

```
[entity_name facta_id="UMLS:C0002395"]
```

Query 2

```
(| [entity_name uniprot_id="Q24K02"]
   [entity_name uniprot_id="P14735"])
```

Query 3

```
(> [setence]
   (& [GDA entity1=$gene entity2=$disease]
      (& [entity_name id=$disease facta_id="UMLS:C0002395"]
         (| [entity_name id=$gene uniprot_id="Q24K02"]
            [entity_name id=$gene uniprot_id="P14735"]))))
```

Fig. 4. Queries for Topic 112

length of documents in the collection and IDF(q_i) is the inverse document frequency for the query q_i, which is defined as

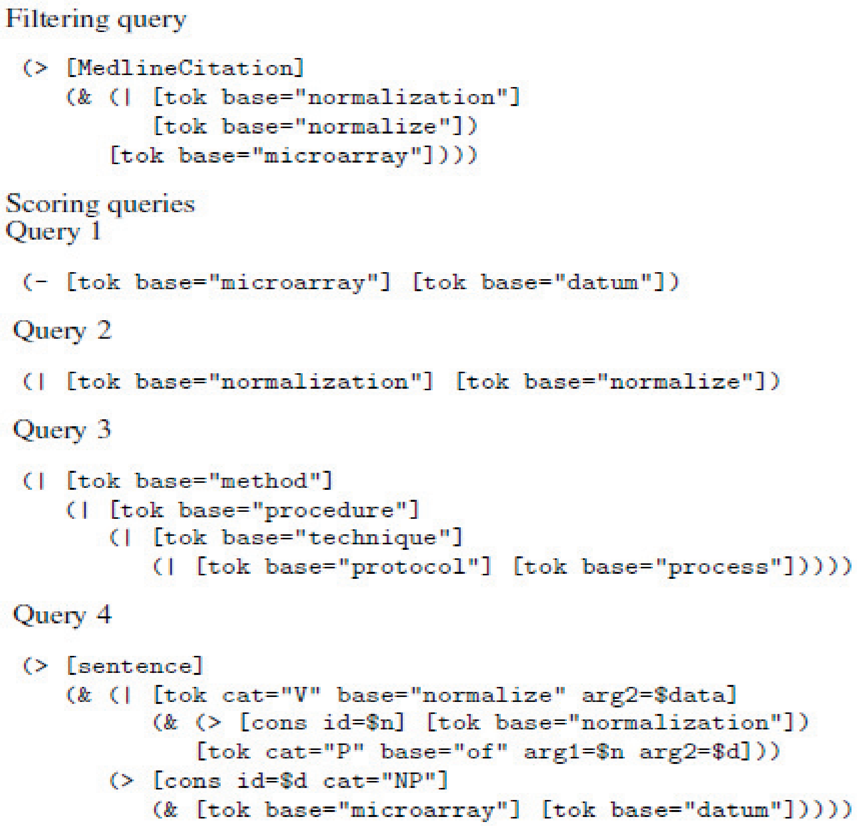
$$IDF(q_i) = log\frac{N - n_{q_i} + 0.5}{n_{q_i} + 0.5}$$

where N is the total number of documents in the collection and n_{qi} is the number of documents which contain at least one region matching the query q_i. [1]

3.2. Manual query conversion

In order to present the effectiveness and flexibility of the proposed framework, we evaluated the framework on the test collection of Ad Hoc Task in TREC Genomics Track 2005 [15], which consists of a 10-year subset of MEDLINE (4,591,008 articles), 50 topics and relevance judgements. The topics are categorized into 5 types of information needs as shown in Table 3.1. In order to search the target documents with integration of NLP techniques on the proposed framework, we processed the documents with various NLP modules.

[1] Note that n_{qi} in the whole document collection cannot be counted in advance because the variation of q_i is innumerable.



Filtering query

```
(> [MedlineCitation]
   (& (| [tok base="normalization"]
         [tok base="normalize"])
      [tok base="microarray"]))
```

Scoring queries
Query 1

```
(- [tok base="microarray"] [tok base="datum"])
```

Query 2

```
(| [tok base="normalization"] [tok base="normalize"])
```

Query 3

```
(| [tok base="method"]
   (| [tok base="procedure"]
      (| [tok base="technique"]
         (| [tok base="protocol"] [tok base="process"]))))
```

Query 4

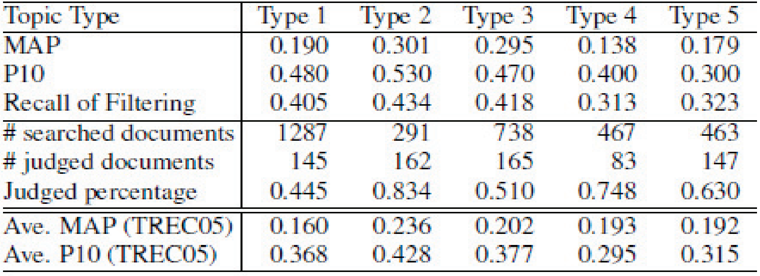
```
(> [sentence]
   (& (| [tok cat="V" base="normalize" arg2=$data]
         (& (> [cons id=$n] [tok base="normalization"])
            [tok cat="P" base="of" arg1=$n arg2=$d]))
      (> [cons id=$d cat="NP"]
         (& [tok base="microarray"] [tok base="datum"]))))
```

Fig. 5. Queries for Topic 107

The topics were manually converted to queries written in extended region algebra. Figure 4 shows an example of queries for Topic 112, “Articles describing the role of ‘IDE gene’ involved in ‘Alzheimer’s Dis-ease’. ” Since the articles relevant to the topic should mention to ‘IDE gene’ and ‘Alzheimer’s Disease,’ we filtered out the documents not including them by the Filtering Query. Thus the Filtering Query for this topic type expresses the condition “Documents contains both the target gene and the target disease.” Note that in the Filtering Query for topic 112, these terms are converted to the annotations [entity_name] with the IDs, ‘IDE gene’ and ‘Alzheimer’s Disease’ are converted to the query on annotations [entity_name] with the UniProt IDs for ‘IDE gene’ and UMLS IDs for ‘Alzheimer’s Disease,’ ‘Q24K02’ or ‘P14735’ and ‘C0002395’ respectively, which is the annotations for the synonymous expressions of the terms. We used three queries as Scoring Queries. First and second queries are the expansion of terms. The last query ex-presses the gene-disease associations between ‘IDE gene’ and ‘Alzheimer’s Disease,’ which are recognized in advance and annotated to documents by [GDA] annotations. The queries for other topics of topic type 2 were constructed by the same way. In the score calculation, the number of regions matching to each Scor-ing Query is counted in each document searched with Filtering Query and the score of the document are calculated with the above formula.

For other topic types, we used queries described in the following: For topic type 1, we used the parsing results and keywords expansions with NER result when the topic include a gene name or a disease name since the topics have a wide variety. Figure 5 shows an example of queries for Topic 107, "Articles describing about 'normalization procedures that are used for microarray data'. " Scoring Queries 1, 2 and 3 expresses the keywords, and Query 4 specifies the expression such as 'normalize microarray data' or 'normalization of microarray data' using the parsing results. For topic type 3, "Articles describing the role of gene in a specific biological process," because currently there is no annotations that directly expressing 'biological process' since no annotations are stored in the current framework describing 'specific biological process' directly, we used various types of annotations such as the parsing results, NER, GDA and event recognizer to specify a biological process. For topic type 4, " Articles describing interactions between two genes in the function of an organ or in a disease," we constructed a query expressing the interactions between the target genes with results of event recognizer with additional keywords of 'function of organ' and 'disease' expanded with NER results. For topic type 5, " Articles describing mutations of a given gene and its biological impact," we used the results of event recognizer expressing 'mutation', and added expanded keywords in 'biological impact.'

Table 4. Search results for each type of topics



Topic Type	Type 1	Type 2	Type 3	Type 4	Type 5
MAP	0.190	0.301	0.295	0.138	0.179
P10	0.480	0.530	0.470	0.400	0.300
Recall of Filtering	0.405	0.434	0.418	0.313	0.323
# searched documents	1287	291	738	467	463
# judged documents	145	162	165	83	147
Judged percentage	0.445	0.834	0.510	0.748	0.630
Ave. MAP (TREC05)	0.160	0.236	0.202	0.193	0.192
Ave. P10 (TREC05)	0.368	0.428	0.377	0.295	0.315

4. Experiments

Table 4 shows the mean average precision (MAP), precision in top 10 results and recall averaged over each topic type. MAP and precision are calculated ignoring the documents not judged in the test collection. Recall is calculated in documents searched by Filtering Query. When the number of articles searched by Filtering Query is less than 10, the total precision is used as P10. The results show that MAP of the proposed framework is higher than average MAP of the runs in TREC 2005 Genomics Track in topic type 1, 2 and 3, but lower in topic type 4 and 5. The precision in top 10 results is significantly higher than that of runs in TREC 2005. One of reasons for low MAP is the Filtering Query. As shown in Table 4, the recall of the Filtering Queries is lower than 0.5 for all topic types, that is, more than half of relevant documents are filtered out with the Filtering Queries. For example, in Topic 111 "Articles describing the role of 'PRNP' involved in 'Mad Cow Disease', " the recall for search with Filtering Query "Documents containing both 'PRNP' and 'Mad Cow Disease' " is only 0.163 despite a query expansion with the result of named entity recognition. Some of the relevant results contain other 'prion diseases' such as "Creutzfeldt-Jakob disease," which is similar disease with "Mad Cow Disease," or does not contain the corresponding disease name. Our current system have annotations which enables the system to regard different expressions of a disease or a gene as the same object, but does not contain annotations for 'knowledge' such that different diseases can be considered as the same concept. In order to satisfy both of the search speed and the accuracy, improvement of Filtering Query with addition of annotations are required to search more relevant documents in the filtering step.

Table 4 shows MAP for topics of topic type 2 with different types of queries used in scoring. The queries Q_{gene}, $Q_{disease}$ and Q_{GDA} correspond to Scoring Query 1, 2 and 3 in Figure 4 respectively. This results show that calculating score considering not only the expanded keywords but the structured relation of keywords, GDA in this case, is effective to improve the accuracy of retrieval.

5. Conclusion and Future Work

We have described a search framework for semantic information retrieval based on the integration of NLP techniques. The framework integrates the annotations from various NLP techniques such as parsing, NER, event recognition and GDA recognition and searches across these annotations based on a framework of region algebra. We evaluated the framework with a simple scoring method on the test collection of TREC Genomics Track and showed the effectiveness of specifying annotations expressing keyword expansions and structural relations. Although the experiments focused on MEDLINE articles in the test collection of TREC Genomics Track, our search framework can be applied to documents in any domain by using NLP applications for that domain.

However, several functionalities are lacking in the current framework. One of the lacks of functionality is a scoring function considering structural relations. Although we used Okapi BM25 by simply expanding to region of annotations instead of keywords in this experiments, the scoring method is not necessarily appropriate for our framework since duplication of annotations exists between the scoring queries. We plan to improve scoring methods by using a probabilistic model and automatic query construction methods using annotated informations.

Acknowledgement

This work was supported by KAKENHI (Grant-in-Aid for Scientific Research (C), No. 21500131).

References

[1] D. Ferrucci, A. Lally, Building an example application with the unstructured information management architecture, IBM Systems Journal 43 (3) (2004) 455–575.

[2] H. Mima, S. Ananiadou, G. Nenadic, J. Tsujii, A Methodology for Terminology-based Knowledge Acquisition and Integration, in: Proceedings of Coling 2002, 2002, pp. 667–673.

[3] TEI Consortium, Text Encoding Initiative (2004). URL http://www.tei-c.org/P4X/

[4] W. Alink, V. Jijkoun, D. Ahn, M. de Rijke, P. Boncz, A. de Vries, Representing and Querying Multi-dimensional Markup for Question Answering, in: Proceedings of the 5th Workshop on NLP and XML, 2006, pp. 3–9.

[5] P. Siniakov, Querying XML documents with multi-dimensional markup, in: Proceedings of the 5th Workshop on NLP and XML, 2006, pp. 43–50.

[6] K. Masuda, J. Tsujii, Tag-annotated text search using extended region algebra, IEICE Transactions 92-D (12) (2009) 2369–2377.

[7] C. L. A. Clarke, G. V. Cormack, F. J. Burkowski, An algebra for structured text search and a framework for its implementation, The computer Journal 38 (1) (1995) 43–56.

[8] Y. Miyao, J. Tsujii, Probabilistic disambiguation models for wide-coverage HPSG parsing, in: Proceedings of ACL 2005, 2005, pp. 83–90.

[9] J.-D. Kim, T. Ohta, Y. Teteisi, J. Tsujii, GENIA corpus - a semantically annotated corpus for bio-textmining, Bioinformatics 19 (suppl. 1) (2003) i180–i182.

[10] J.-D. Kim, T. Ohta, J. Tsujii, Corpus annotation for mining biomedical events from literature, BMC Bioinformatics 9 (1) (2008) 10, iSSN 1471-2105.

[11] H.-W. Chun, Y. Tsuruoka, J.-D. Kim, R. Shiba, N. Nagata, T. Hishiki, J. Tsujii, Automatic recognition of topic-classified relations between prostate cancer and genes using medline abstracts, BMC-Bioinformatics 7 (Suppl 3) (2006) S4.

[12] M. Miwa, R. Sætre, J.-D. Kim, J. Tsujii, Event extraction with complex event classification using rich features, Journal of Bioinformatics and Computational Biology (JBCB) 8 (1) (2010) 131–146.

[13] K. Hirohata, N. Okazaki, S. Ananiadou, M. Ishizuka, Identifying sections in scientific abstracts using conditional random fields, in: Proceedings of IJCNLP 2008, 2008, pp. 381–388.

[14] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, M. Gatford, Okapi at trec-3, in: Overview of the Third Text REtrieval Conference (TREC-3), 1996, pp. 109–126.

[15] W. Hersh, A. Cohen, J. Yang, R. T. Bhupatiraju, P. Roberts, M. Hearst, Trec 2005 genomics track overview, in: In TREC 2005 notebook, 2005, pp. 14–25.