A Ranking Model of Proximal and Structural Text Retrieval Based on Region Algebra

Katsuya Masuda
Department of Computer Science, Graduate School of Information Science and Technology,
University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo 113-0033, Japan
kmasuda@is.s.u-tokyo.ac.jp

Abstract

This paper investigates an application of the ranked region algebra to information retrieval from large scale but unannotated documents. We automatically annotated documents with document structure and semantic tags by using taggers, and retrieve information by specifying structure represented by tags and words using ranked region algebra. We report in detail what kind of data can be retrieved in the experiments by this approach.

1 Introduction

In the biomedical area, the number of papers is very large and increases, as it is difficult to search the information. Although keyword-based retrieval systems can be applied to a database of papers, users may not get the information they want since the relations between these keywords are not specified. If the document structures, such as “title”, “sentence”, “author”, and relation between terms are tagged in the texts, then the retrieval is improved by specifying such structures. Models of the retrieval specifying both structures and words are pursued by many researchers (Chinenyanga and Kushmerick, 2001; Wolff et al., 1999; Theobald and Weilkum, 2000; Deutsch et al., 1998; Salminen and Tompa, 1994; Clarke et al., 1995). However, these models are not robust unlike keyword-based retrieval, that is, they retrieve only the exact matches for queries.

In the previous research (Masuda et al., 2003), we proposed a new ranking model that enables proximal and structural search for structured text. This paper investigates an application of the ranked region algebra to information retrieval from large scale but unannotated documents. We reports in detail what kind of data can be retrieved in the experiments. Our approach is to annotate documents with document structures and semantic tags by taggers automatically, and to retrieve information by specifying both structures and words using ranked region algebra. In this paper, we apply our approach to the OHSUMED test collection (Hersh et al., 1994), which is a public test collection for information retrieval in the field of biomedical science but not tag-annotated. We annotate OHSUMED by various taggers and retrieve information from the tag-annotated corpus.

We have implemented the ranking model in our retrieval engine, and had preliminary experiments to evaluate our model. In the experiments, we used the GENIA corpus (Ohta et al., 2002) as a small but manually tag-annotated corpus, and OHSUMED as a large but automatically tag-annotated corpus. The experiments show that our model succeeded in retrieving the relevant answers that an exact-matching model fails to retrieve because of lack of robustness, and the relevant answers that a non-structured model fails because of lack of structural specification. We report how structural specification works and how it doesn’t work in the experiments with OHSUMED.

Section 2 explains the region algebra. In Section 3, we describe our ranking model for the structured query and texts. In Section 4, we show the experimental results of this system.

Expression	Description
$q_1 \triangleright q_2$	$G_{q_1 \triangleright q_2} = \Gamma(\{a \mid a \in G_{q_1} \wedge \exists b \in G_{q_2}.(b \sqsubset a)\})$
$q_1 \not\triangleright q_2$	$G_{q_1 \not\triangleright q_2} = \Gamma(\{a \mid a \in G_{q_1} \wedge \nexists b \in G_{q_2}.(b \sqsubset a)\})$
$q_1 \triangleleft q_2$	$G_{q_1 \triangleleft q_2} = \Gamma(\{a \mid a \in G_{q_1} \wedge \exists b \in G_{q_2}.(a \sqsubset b)\})$
$q_1 \not\triangleleft q_2$	$G_{q_1 \not\triangleleft q_2} = \Gamma(\{a \mid a \in G_{q_1} \wedge \nexists b \in G_{q_2}.(a \sqsubset b)\})$
$q_1 \bigtriangleup q_2$	$G_{q_1 \bigtriangleup q_2} = \Gamma(\{c \mid c \sqsubset (-\infty, \infty) \wedge \exists a \in G_{q_1}.\exists b \in G_{q_2}.(a \sqsubset c \wedge b \sqsubset c)\})$
$q_1 \bigtriangledown q_2$	$G_{q_1 \bigtriangledown q_2} = \Gamma(\{c \mid c \sqsubset (-\infty, \infty) \wedge \exists a \in G_{q_1}.\exists b \in G_{q_2}.(a \sqsubset c \vee b \sqsubset c)\})$
$q_1 \diamond q_2$	$G_{q_1 \diamond q_2} = \Gamma(\{c \mid c = (p_s, p'_e)$ where $\exists (p_s, p_e) \in G_{q_1}.\exists (p'_s, p'_e) \in G_{q_2}.(p_e < p'_s)\})$

Table 1: Operators of the Region algebra

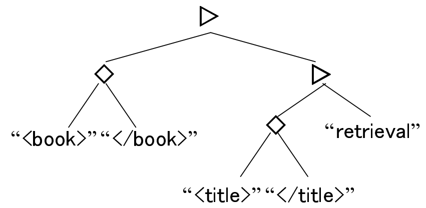

Figure 1: Tree of the query ‘[book] ▷ ([title] ▷ “retrieval”)’

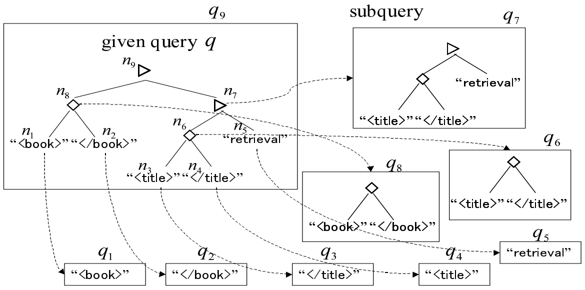

Figure 2: Subqueries of the query ‘[book] ▷ ([title] ▷ “retrieval”)’

2 Background: Region algebra

The region algebra (Salminen and Tompa, 1994; Clarke et al., 1995; Jaakkola and Kilpelainen, 1999) is a set of operators representing the relation between the *extents* (i.e. regions in texts), where an extent is represented by a pair of positions, beginning and ending position. Region algebra allows for the specification of the structure of text.

In this paper, we suppose the region algebra proposed in (Clarke et al., 1995). It has seven operators as shown in Table 1; four containment operators ($\triangleright$, $\not\triangleright$, $\triangleleft$, $\not\triangleleft$) representing the containment relation between the extents, two combination operators ($\bigtriangleup$, $\bigtriangledown$) corresponding to “and” and “or” operator of the boolean model, and ordering operator ($\diamond$) representing the order of words or structures in the texts. A containment relation between the extents is represented as follows: $e = (p_s, p_e)$ contains $e' = (p'_s, p'_e)$ iff $p_s \leq p'_s \leq p'_e \leq p_e$ (we express this relation as $e \sqsupset e'$). The result of retrieval is a set of non-nested extents, that is defined by the following function Γ over a set of extents S:

$$\Gamma(S) = \{e \mid e \in S \wedge \nexists e' \in S.(e' \neq e \wedge e' \sqsubset e)\}$$

Intuitively, $\Gamma(S)$ is an operation for finding the shortest matching. A set of non-nested extents matching query q is expressed as G_q.

For convenience of explanation, we represent a query as a tree structure as shown in Figure 1 (‘[x]’ is a abbreviation of ‘$\langle$x$\rangle$ $\diamond$ $\langle$/x$\rangle$’). This query represents ‘Retrieve the books whose title has the word “retrieval.” ’

The algorithm for finding an exact match of a query works efficiently. The time complexity of the algorithm is linear to the size of a query and the size of documents (Clarke et al., 1995).

3 A Ranking Model for Structured Queries and Texts

This section describes the definition of the relevance between a document and a structured query represented by the region algebra. The key idea is that a structured query is decomposed into subqueries, and the relevance of the whole query is represented as a vector of relevance measures of subqueries.

Our model assigns a relevance measure of the

query		matching extents in (1,15)	matching extents in (16,30)	constructed by
q_1	"⟨book⟩"	(1,1)	(16,16)	inverted list
q_2	"⟨/book⟩"	(15,15)	(30,30)	inverted list
q_3	"⟨title⟩"	(2,2), (7,7)	(17,17), (22,22)	inverted list
q_4	"⟨/title⟩"	(5,5), (11,11)	(20,20), (27,27)	inverted list
q_5	"retrieval"	(4,4), (13,13)	(28,28)	inverted list
q_6	'[title]'	(2,5), (7,11)	(17,20), (22,27)	G_{q_3}, G_{q_4}
q_7	'[title] ▷ "retrieval"'	(2,5)		G_{q_5}, G_{q_6}
q_8	'[book]'	(1,15)	(16,30)	G_{q_1}, G_{q_2}
q_9	'[book] ▷ ([title] ▷ "retrieval")'	(1,15)		G_{q_7}, G_{q_8}

Table 2: Extents that match each subquery in the extent $(1, 15)$ and $(16, 30)$

⟨book⟩ 1	⟨title⟩ 2	ranked 3	retrieval 4	⟨/title⟩ 5	⟨chapter⟩ 6
⟨title⟩ 7	tf 8	and 9	idf 10	⟨/title⟩ 11	ranked 12
retrieval 13	⟨/chapter⟩ 14	⟨/book⟩ 15	⟨book⟩ 16	⟨title⟩ 17	structured 18
text 19	⟨/title⟩ 20	⟨chapter⟩ 21	⟨title⟩ 22	search 23	for 24
structured 25	text 26	⟨/title⟩ 27	retrieval 28	⟨/chapter⟩ 29	⟨/book⟩ 30

Figure 3: An example text

structured query as a vector of relevance measures of the subqueries. In other words, the relevance is defined by the number of portions matched with subqueries in a document. If an extent matches a subquery of query q, the extent will be somewhat relevant to q even when the extent does not exactly match q. Figure 2 shows an example of a query and its subqueries. In this example, even when an extent does not match the whole query exactly, if the extent matches "retrieval" or '[title]▷"retrieval"', the extent is considered to be relevant to the query. Subqueries are formally defined as follows.

Definition 1 (Subquery) *Let q be a given query and $n_1, ..., n_m$ be the nodes of q. Subqueries $q_1, ..., q_m$ of q are the subtrees of q. Each q_i has node n_i as a root node.*

When a relevance $\sigma(q_i, d)$ between a subquery q_i and a document d is given, the relevance of the whole query is defined as follows.

Definition 2 (Relevance of the whole query) *Let q be a given query, d be a document and $q_1, ..., q_m$ be subqueries of q. The relevance vector $\Sigma(q, d)$ of d is defined as follows:*

$$\Sigma(q, d) = \langle \sigma(q_1, d), \sigma(q_2, d), ..., \sigma(q_m, d) \rangle$$

A relevance of a subquery should be defined similarly to that of keyword-based queries in the traditional ranked retrieval. For example, TFIDF, which is used in our experiments in Section 4, is the most simple and straightforward one, while other relevance measures recently proposed (Robertson and Walker, 2000; Fuhr, 1992) can be applied. TF of a subquery is calculated using the number of extents matching the subquery, and IDF of a subquery is calculated using the number of documents including the extents matching the subquery. When a text is given as Figure 3 and document collection is {(1,15),(16,30)}, extents matching each subquery in each document are shown in Table 2. TF and IDF are calculated using the number of extents matching subquery in Table 2.

While we have defined a relevance of the structured query as a vector, we need to arrange the documents according to the relevance vectors. In this paper, we first map a vector into a scalar value, and then sort the documents according to this scalar measure.

Three methods are introduced for the mapping from the relevance vector to the scalar measure. The first one simply works out the sum of the elements of the relevance vector.

Definition 3 (Simple Sum)

$$\rho_{sum}(q, d) = \sum_{i=1}^{m} \sigma(q_i, d)$$

The second appends a coefficient representing the rareness of the structures. When the query is $A \rhd B$ or $A \lhd B$, if the number of extents matching the query is close to the number of extents matching A, matching the query does not seem to be very important because it means that the extents that match A mostly match $A \rhd B$ or $A \lhd B$. The case of the other operators is the same as with ▷ and ◁.

Num	Query
1	‘([cons] ▷ ([sem] ▷ “G#DNA_domain_or_region”)) △ (“in” ◇ ([cons] ▷ ([sem] ▷ (“G#tissue” ▽ “G#body_part”))))’
2	‘([event] ▷ ([obj] ▷ “gene”)) △ (“in” ◇ ([cons] ▷ ([sem] ▷ (“G#tissue” ▽ “G#body_part”))))’
3	‘([event]▷([obj]◇([sem]▷“G#DNA_domain_or_region”)))△(“in”◇([cons]▷([sem]▷(“G#tissue”▽“G#body_part”))))’

Table 3: Queries submitted in the experiments on the GENIA corpus

Definition 4 (Structure Coefficient) *When the operator op is $\triangle$, $\bigtriangledown$ or $\diamond$, the structure coefficient of the query $A\ op\ B$ is:*

$$sc_{AopB} = \frac{C(A) + C(B) - C(A\ op\ B)}{C(A) + C(B)}$$

and when the operator op is $\rhd$ or $\lhd$, the structure coefficient of the query $A\ op\ B$ is:

$$sc_{AopB} = \frac{C(A) - C(A\ op\ B)}{C(A)}$$

where A and B are the queries and $C(A)$ is the number of extents that match A in the document collection.

The scalar measure $\rho_{sc}(q_i, d)$ is then defined as

$$\rho_{sc}(q, d) = \sum_{i=1}^{m} sc_{q_i} \cdot \sigma(q_i, d)$$

The third is a combination of the measure of the query itself and the measure of the subqueries. Although we calculate the score of extents by subqueries instead of using only the whole query, the score of subqueries can not be compared with the score of other subqueries. We assume normalized weight of each subquery and interpolate the weight of parent node and children nodes.

Definition 5 (Interpolated Coefficient) *The interpolated coefficient of the query q_i is recursively defined as follows:*

$$\rho_{ic}(q_i, d) = \lambda \cdot \sigma(q_i, d) + (1 - \lambda) \frac{\sum_{c_i} \rho_{ic}(q_{c_i}, d)}{l}$$

where c_i is the child of node n_i, l is the number of children of node n_i, and $0 \leq \lambda \leq 1$.

This formula means that the weight of each node is defined by a weighted average of the weight of the query and its subqueries. When $\lambda = 1$, the weight of a query is normalized weight of the query. When $\lambda = 0$, the weight of a query is calculated from the weight of the subqueries, i.e. the weight is calculated by only the weight of the words used in the query.

4 Experiments

In this section, we show the results of our preliminary experiments of text retrieval using our model. We used the GENIA corpus (Ohta et al., 2002) and the OHSUMED test collection (Hersh et al., 1994).

We compared three retrieval models, i) our model, ii) exact matching of the region algebra (*exact*), and iii) not structured model (*flat*). The queries submitted to our system are shown in Table 3 and 4. In the *flat* model, the query was submitted as a query composed of the words in the queries connected by the “and” operator (△). For example, in the case of Query 1, the query submitted to the system in the *flat* model is ‘ “G#DNA_domain_or_region” △ “in” △ “G#tissue” △ “G#body_part” .’ The system output the ten results that had the highest relevance for each model.

In the following experiments, we used a computer that had Pentium III 1.27GHz CPU, 4GB memory. The system was implemented in C++ with Berkeley DB library.

4.1 GENIA corpus

The GENIA corpus is an XML document composed of paper abstracts in the field of biomedical science. The corpus consisted of 1,990 articles, 873,087 words (including tags), and 16,391 sentences. In the GENIA corpus, the document structure was annotated by tags such as “⟨article⟩” and “⟨sentence⟩”, technical terms were annotated by “⟨cons⟩”, and events were annotated by “⟨event⟩”.

The queries in Table 3 are made by an expert in the field of biomedicine. The document was “sentence” in this experiments. Query 1 retrieves sentences including a gene in a tissue. Queries 2 and 3 retrieve sentences representing an event having a gene as an object and occurring in a tissue. In Query 2, a gene was represented by the word “gene,” and in Query 3, a gene was represented by the annotation “G#DNA_domain_or_region.”

	Query
4	‘ “postmenopausal” △ ([neoplastic] ▷ (“breast” ◇ “cancer”)) △ ([therapeutic] ▷ (“replacement” ◇ “therapy”)) ’ 55 year old female, postmenopausal does estrogen replacement therapy cause breast cancer
5	‘ ([disease]▷(“copd”▽(“chronic”◇“obstructive”◇“pulmonary”◇“disease”)))△“theophylline”△([disease]▷“asthma”) ’ 50 year old with copd theophylline uses–chronic and acute asthma
6	‘ ([neoplastic] ▷ (“lung” ◇ “cancer”)) △ ([therapeutic] ▷ (“radiation” ◇ “therapy”)) ’ lung cancer lung cancer, radiation therapy
7	‘([disease]▷“pancytopenia”)△([neoplastic]▷(“acute”◇“megakaryocytic”◇“leukemia”))△(“treatment▽“prognosis”)’ 70 year old male who presented with pancytopenia acute megakaryocytic leukemia, treatment and prognosis
8	‘([disease]▷“hypercalcemia”)△([neoplastic]▷“carcinoma”)△(([therapeutic]▷“gallium”)▽(“gallium”◇“therapy”))’ 57 year old male with hypercalcemia secondary to carcinoma effectiveness of gallium therapy for hypercalcemia
9	‘(“lupus”◇“nephritis”)△(“thrombotic”◇([disease]▷(“thrombocytopenic”◇“purpura”))△(“management”▽“diagnosis”)’ 18 year old with lupus nephritis and thrombotic thrombocytopenic purpura lupus nephritis, diagnosis and management
10	‘ ([mesh] ▷ “treatment”) △ ([disease] ▷ “endocarditis”) △ ([sentence] ▷ (“oral” ◇ “antibiotics”) ’ 28 year old male with endocarditis treatment of endocarditis with oral antibiotics
11	‘ ([mesh] ▷ “female”) △ ([disease] ▷ (“anorexia” △ bulimia)) △ ([disease] ▷ “complication”) ’ 25 year old female with anorexia/bulimia complications and management of anorexia and bulimia
12	‘ ([disease] ▷ “diabete”) △ ([disease] ▷ (“peripheral” ◇ “neuropathy”)) △ ([therapeutic] ▷ “pentoxifylline”) ’ 50 year old diabetic with peripheral neuropathy use of Trental for neuropathy, does it work?
13	‘ (“cerebral” ◇ “edema”) △ ([disease] ▷ “infection”) △ (“diagnosis” ▽ ([therapeutic] ▷ “treatment”)) ’ 22 year old with fever, leukocytosis, increased intracranial pressure, and central herniation cerebral edema secondary to infection, diagnosis and treatment
14	‘ ([mesh] ▷ “female”) △ ([disease] ▷ (“urinary” ◇ “tract” ◇ “infection”)) △ ([therapeutic] ▷ “treatment”) ’ 23 year old woman dysuria Urinary Tract Infection, criteria for treatment and admission
15	‘ ([disease] ▷ (“chronic” ◇ “fatigue” ◇ “syndrome”)) △ ([therapeutic] ▷ “treatment”) ’ chronic fatigue syndrome chronic fatigue syndrome, management and treatment

Table 4: Queries submitted in the experiments on the OHSUMED test collection and original queries of OHSUMED. The first line is a query submitted to the system, the second and third lines are the original query of the OHSUMED test collection, the second is information of patient and the third is request information.

For the *exact* model, ten results were selected randomly from the exactly matched results if the number of results was more than ten. The results are blind tested, i.e., after we had the results for each model, we shuffled these results randomly for each query, and the shuffled results were judged by an expert in the field of biomedicine whether they were relevant or not.

Table 5 shows the number of the results that were judged relevant in the top ten results. The results show that our model was superior to the *exact* and *flat* models for all queries. Compared to the *exact* model, our model output more relevant documents, since our model allows the partial matching of the query, which shows the robustness of our model. In addition, our model gives a better result than the *flat* model, which means that the structural specification of the query was effective for finding the relevant documents.

Comparing our models, the number of relevant results using ρ_{sc} was the same as that of ρ_{sum}. The results using ρ_{ic} varied between the results of the *flat* model and the results of the *exact* model depending on the value of λ.

4.2 OHSUMED test collection

The OHSUMED test collection is a document set composed of paper abstracts in the field of biomed-

Query	*our model*					*exact*	*flat*
	ρ_{sum}	ρ_{sc}	ρ_{ic}	ρ_{ic}	ρ_{ic}		
			(λ = 0.25)	(λ = 0.5)	(λ = 0.75)		
1	10/10	10/10	8/10	9/10	9/10	9/10	9/10
2	6/10	6/10	6/10	6/10	6/10	5/ 5	3/10
3	10/10	10/10	10/10	10/10	10/10	9/ 9	8/10

Table 5: (The number of relevant results) / (the number of all results) in top 10 results on the GENIA corpus

Query	*our model*					*exact*	*flat*
	ρ_{sum}	ρ_{sc}	ρ_{ic}	ρ_{ic}	ρ_{ic}		
			(λ = 0.25)	(λ = 0.5)	(λ = 0.75)		
4	7/10	7/10	4/10	4/10	4/10	5/12	4/10
5	4/10	3/10	2/10	3/10	3/10	2/9	2/10
6	8/10	8/10	7/10	7/10	7/10	12/34	6/10
7	1/10	0/10	0/10	0/10	0/10	0/0	0/10
8	5/10	5/10	4/10	2/10	2/10	2/2	5/10
9	0/10	0/10	4/10	5/10	4/10	0/1	0/10
10	1/10	1/10	1/10	1/10	0/10	0/0	1/10
11	4/10	4/10	2/10	3/10	5/10	0/0	4/10
12	3/10	3/10	2/10	2/10	2/10	0/0	3/10
13	2/10	1/10	0/10	1/10	0/10	0/1	3/10
14	1/10	1/10	1/10	1/10	1/10	0/5	3/10
15	3/10	3/10	5/10	2/10	3/10	0/1	8/10

Table 6: (The number of relevant results) / (the number of all results) in top 10 judged results on the OHSUMED test collection (“all results” are relevance-judged results in the *exact* model)

ical science. The collection has a query set and a list of relevant documents for each query. From 50 to 300 documents are judged whether or not relevant to each query. The query consisted of patient information and information request. We used title, abstract, and human-assigned MeSH term fields of documents in the experiments. Since the original OHSUMED is not annotated with tags, we annotated it with tags representing document structures such as “⟨article⟩” and “⟨sentence⟩”, and annotated technical terms with tags such as “⟨disease⟩” and “⟨therapeutic⟩” by longest matching of terms of Unified Medical Language System (UMLS). In the OHSUMED, relations between technical terms such as events were not annotated unlike the GENIA corpus. The collection consisted of 348,566 articles, 78,207,514 words (including tags), and 1,731,953 sentences.

12 of 106 queries of OHSUMED are converted into structured queries of Region Algebra by an expert in the field of biomedicine. These queries are shown in Table 4, and submitted to the system. The document was “article” in this experiments. For the *exact* model, all exact matches of the whole query were judged. Since there are documents that are not judged whether or not relevant to the query in the OHSUMED, we picked up only the documents that are judged.

Table 6 shows the number of relevant results in top ten results. The results show that our model succeeded in finding the relevant results that the *exact* model could not find, and was superior to the *flat* model for Query 4, 5, and 6. However, our model was inferior to the *flat* model for Query 14 and 15.

Comparing our models, the number of relevant results using ρ_{sc} and ρ_{ic} was lower than that using ρ_{sum}.

Query	*our model*	*exact*
1	1.94 s	0.75 s
2	1.69 s	0.34 s
3	2.02 s	0.49 s

Table 7: The retrieval time (sec.) on GENIA corpus

Query	*our model*	*exact*
4	25.13 s	2.17 s
5	24.77 s	3.13 s
6	23.84 s	2.18 s
7	24.00 s	2.70 s
8	27.62 s	3.50 s
9	20.62 s	2.22 s
10	30.72 s	7.60 s
11	25.88 s	4.59 s
12	25.44 s	4.28 s
13	21.94 s	3.30 s
14	28.44 s	4.38 s
15	20.36 s	3.15 s

Table 8: The retrieval time (sec.) on OHSUMED test collection

4.3 Discussion

In the experiments on OHSUMED, the number of relevant documents of our model were less than that of the *flat* model in some queries. We think this is because i) specifying structures was not effective, ii) weighting subqueries didn't work, iii) MeSH terms embedded in the documents are effective for the *flat* model and not effective for our model, iv) or there are many documents that our system found relevant but were not judged since the OHSUMED test collection was made using keyword-based retrieval.

As for i), structural specification in the queries is not well-written because the exact model failed to achieve high precision and its coverage is very low. We used only tags specifying technical terms as structures in the experiments on OHSUMED. This structure was not so effective because these tags are annotated by longest match of terms. We need to use the tags representing relations between technical terms to improve the results. Moreover, structured query used in the experiments may not specify the request information exactly. Therefore we think converting queries written by natural language into the appropriate structured queries is important, and lead to the question answering using variously tag-annotated texts.

As for ii), we think the weighting didn't work because we simply use frequency of subqueries for weighting. To improve the weighting, we have to assign high weight to the structure concerned with user's intention, that are written in the request information. This is shown in the results of Query 9. In Query 9, relevant documents were not retrieved except the model using ρ_{ic}, because although the request information was information concerned "lupus nephritis", the weight concerned with "lupus nephritis" was smaller than that concerned with "thrombotic" and "thrombocytopenic purpura" in the models except ρ_{ic}. Because the structures concerning with user's intention did not match the most weighted structures in the model, the relevant documents were not retrieved.

As for iii), MeSH terms are human-assigned keywords for each documents, and no relation exists across a boundary of each MeSH terms. in the *flat* model, these MeSH term will improve the results. However, in *our* model, the structure sometimes matches that are not expected. For example, In the case of Query 14, the subquery ' "chronic" $\diamond$ "fatigue" $\diamond$ "syndrome" ' matched in the field of MeSH term across a boundary of terms when the MeSH term field was text such as "Affective Disorders/*CO; Chronic Disease; Fatigue/*PX; Human; Syndrome " because the operator $\diamond$ has no limitation of distance.

As for iv), the OHSUMED test collection was constructed by attaching the relevance judgement to the documents retrieved by keyword-based retrieval.

To show the effectiveness of structured retrieval more clearly, we need test collection with (structured) query and lists of relevant documents, and the tag-annotated documents, for example, tags representing the relation between the technical terms such as "event", or taggers that can annotate such tags.

Table 7 and 8 show that the retrieval time increases corresponding to the size of the document collection. The system is efficient enough for information retrieval for a rather small document set like GENIA corpus. To apply to the huge databases such as Web-based applications, we might require a constant time algorithm, which should be the subject of future research.

5 Conclusions and Future work

We proposed an approach to retrieve information from documents which are not annotated with any tags. We annotated documents with document structures and semantic tags by taggers, and retrieved information by using ranked region algebra. We showed what kind of data can be retrieved from documents in the experiments.

In the discussion, we showed several points about the ranked retrieval for structured texts. Our future work is to improve a model, corpus etc. to improve the ranked retrieval for structured texts.

Acknowledgments

I am grateful to my supervisor, Jun'ichi Tsujii, for his support and many valuable advices. I also thank to Takashi Ninomiya, Yusuke Miyao for their valuable advices, Yoshimasa Tsuruoka for providing me with a tagger, Tomoko Ohta for making queries, and anonymous reviewers for their helpful comments. This work is a part of the Kototoi project[1] supported by CREST of JST (Japan Science and Technology Corporation).

References

Taurai Chinenyanga and Nicholas Kushmerick. 2001. Expressive and efficient ranked querying of XML data. In *Proceedings of WebDB-2001 (SIGMOD Workshop on the Web and Databases)*.

Charles L. A. Clarke, Gordon V. Cormack, and Forbes J. Burkowski. 1995. An algebra for structured text search and a framework for its implementation. *The computer Journal*, 38(1):43–56.

Alin Deutsch, Mary Fernandez, Daniela Florescu, Alon Levy, and Dan Suciu. 1998. XML-QL: A query language for XML. In *Proceedings of WWW The Query Language Workshop*.

Norbert Fuhr. 1992. Probabilistic models in information retrieval. *The Computer Journal*, 35(3):243–255.

William Hersh, Chris Buckley, T. J. Leone, and David Hickam. 1994. OHSUMED: an interactive retrieval evaluation and new large test collection for research. In *Proceedings of the 17th International ACM SIGIR Conference*, pages 192–201.

Jani Jaakkola and Pekka Kilpelainen. 1999. Nested text-region algebra. Technical Report C-1999-2, University of Helsinki.

Katsuya Masuda, Takashi Ninomiya, Yusuke Miyao, Tomoko Ohta, and Jun'ichi Tsujii. 2003. A robust retrieval engine for proximal and structural search. In *Proceedings of the HLT-NAACL 2003 short papers*.

Tomoko Ohta, Yuka Tateisi, Hideki Mima, and Jun'ichi Tsujii. 2002. GENIA corpus: an annotated research abstract corpus in molecular biology domain. In *Proceedings of the HLT 2002*.

Stephen E. Robertson and Steve Walker. 2000. Okapi/Keenbow at TREC-8. In *Proceedings of TREC-8*, pages 151–161.

Airi Salminen and Frank Tompa. 1994. Pat expressions: an algebra for text search. *Acta Linguistica Hungarica*, 41(1-4):277–306.

Anja Theobald and Gerhard Weilkum. 2000. Adding relevance to XML. In *Proceedings of WebDB'00*.

Jens Wolff, Holger Flörke, and Armin Cremers. 1999. XPRES: a Ranking Approach to Retrieval on Structured Documents. Technical Report IAI-TR-99-12, University of Bonn.

[1] http://www.kototoi.org/