효과적인 HLA개체인식을 위한 부분매칭기법

채정민[†] · 정영희[†] · 이태민[†] · 채지은[††] · 오흥범[†††] · 정순영[††††]

요 약

생의학분야에서 문헌에 표기된 개체를 인식하기 위해 길이우선매칭기법을 빈번히 사용한다. 길이우선매칭기법은 사전을 이용한 개체인식기법으로 좋은 사전만 구축되어 있다면 빠르고 정확하게 개체를 찾아낼 수 있다는 장점을 가진다. 그러나 개체가 나열되고 중복된 단어가 생략될 경우에는 길이우선매칭기법을 이용할 경우 성능이 현저히 떨어지게 된다. 우리는 이러한 인식성능문제를 해결하기 위해 부분매칭기법을 제안한다. 제안된 부분매칭기법은 생략이 발생될 수 있다는 것을 가정하여 다수의 후보개체를 만들어 내고 그 후에 최적화 알고리즘을 통해 다수의 개체후보 중에서 가장 타당해 보이는 개체를 선택한다. 우리는 생의학분야의 개체 중에서 나열되는 경우가 빈번한 HLA 유전자, HLA 항원, HLA 대립유전자 개체들을 대상으로 길이우선매칭기법과 제안된 부분매칭기법의 개체인식성능을 분석하였다. 3종의 HLA 개체들을 인식하기 위해서 먼저 확장사전과 태그기반사전을 구축하였으며, 그 후 구축된 사전을 이용해 길이우선매칭과 부분매칭을 수행하였다. 실험결과에 따르면 길이우선매칭기법은 HLA 항원 개체에서 좋은 성능을 보였으며 부분매칭기법은 생략된 표현이 빈번한 HLA 유전자 개체, HLA 대립유전자 개체에서 좋은 성능을 보였다. 부분매칭기법은 HLA 대립유전자 개체를 대상으로 95.59%의 높은 F-score를 얻었다.

주제어 : 자연어처리, 개체인식; 인간백혈구항원

The partial matching method for effective recognizing HLA entities

Jeongmin Chae[†] · YoungHee Jung[†] · TaeMin Lee[†] · JiEun Chae[††]
HeungBum Oh[†††] · SoonYoung Jung[††††]

ABSTRACT

In the biomedical domain, the longest matching method is frequently used for recognizing named entity written in the literature. This method uses a dictionary as a resource for named entity recognition. If there exist appropriated dictionary about target domain, the longest matching method has the advantage of being able to recognize the entities of target domain quickly and exactly. However, the longest matching method is difficult to recognize the enumerated named entities, because these entities are frequently expressed as being omitted some words. In order to resolve this problem, we propose the partial matching method using a dictionary. The proposed method makes several candidate entities on the assumption that the ellipses may be included. After that, the method selects the most valid one among candidate entities through the optimization algorithm. We tested the longest and partial matching method about HLA entities: HLA gene, antigen, and allele entities, which are frequently enumerated among biomedical entities. As preparing for named entity recognition, we built two new resource, extended dictionary and tag-based dictionary about HLA entities. And later, we performed the longest and partial matching method using each dictionary. According to our experiment result, the longest matching method was effective in recognizing HLA antigen entities, in which the ellipses are rare, and the partial matching method was effective in recognizing HLA gene and allele entities, in which the ellipses are frequent. Especially, the partial matching method had a high F-score 95.59% about HLA alleles.

Keywords : Natural language processing; Named entity recognition; HLA

[†] 정 회 원: 고려대학교 컴퓨터교육과 박사과정
[††] 정 회 원: 네이버 검색서버팀 대리
[†††] 정 회 원: 울산의대 서울아산병원 진단검사의학과 교수
[††††] 종신회원: 고려대학교 컴퓨터교육과 교수(교신저자)

논문접수: 2011년 01월 21일, 심사완료: 2011년 03월 19일
* 본 연구는 2010년도 고려대학교 사범대학 특별연구비의 지원에 의하여 수행되었음

1. Introduction

Dictionary-based named entity recognition is a method to recognize entities in literature using dictionaries. This approach can recognize a term by searching the most similar one in the dictionary. However, the performance of this approach depends on the size and quality of the dictionary. For example, if the dictionary includes ambiguous words, a result has many false positive and if the dictionary does not have enough synonyms and variations, the result has many false negative.

In order to solve these problems, [1] proposed filtering rules for removing false positive of the disease name. [1],[2] removed many false negative using extended dictionary. Longest matching method extracts the longest matched entities without overlapping another entity using a dictionary. However, it cannot search partly omitted entities. When entities are enumerated in a sentence, repeated words among entities are frequently omitted. For example, when IL-4 and IL-13 are enumerated in "*IL-4 and -13*", 'IL' can be omitted. This type of omission is not easy to be recognized using longest matching method. [3] discussed that it was difficult to find author′s intended entities without removing the ambiguity of coordinating clause.

[4], [5], and [6] tried to solve a coordination problem in general discourse pattern. [6] deals with the coordination problem in the newspaper domain. They divided the types of conjunctions related with named entities into four different types : name internal conjunction, name external conjunction, right-copy separator, and left-copy separator. Their research had 84% overall performance. However, the performance about entities having ellipsis showed relatively low performance 58.5%.

[3], [7], and [8] tried to solve a coordination problem in the biomedical domain. [7] added the post-processing method to find coordinate phrase. This method had a little improvement of named entity recognition through recognizing some coordinate patterns. [8] proposed supervised machine learning-based approach to the resolution of elliptical coordination in noun phrases. In the GENIA corpus, they had 86% F-score.

We developed a partial matching method for named entity recognition, and this method applied to HLA domain. Human Leukocyte Antigen (HLA) is genes related to immune system function in humans. Even if people are contact with the same virus, not all of them catch the disease. Even though some people are recovering from their illness, their rate of progress may be either quite slow or fast. This difference has been regarded by the difference of HLA allele. So HLA has been interested in clinical researchers. On the one hand, HLA gene has a characteristic that is most polymorphic in human genes [9]. By this properties, researchers enumerate HLA gene, antigen, and allele in literature. For instance, HLA entities such as 'DR1', 'DR2', 'DR3', 'DR4', and 'DR13', were represented as "*DR1, 2, 3, 4, and 13*" in the literature. Like this, the repeated word, 'DR', was frequently omitted.

In previous research [10], we proposed a rule-based NER method using the standard naming convention of HLA gene, antigen, and allele. This system had a relatively higher precision, but lower recall. Because HLA gene and allele were omitted parts of entities and synonym of HLA antigens couldn't was recognized, this result is appeared. In the example below, it is difficult to recognize HLA antigens and alleles.

- *Forty-eight patients with various kinds of myositis were studied for HLA-A, B, C, and DR antigens*

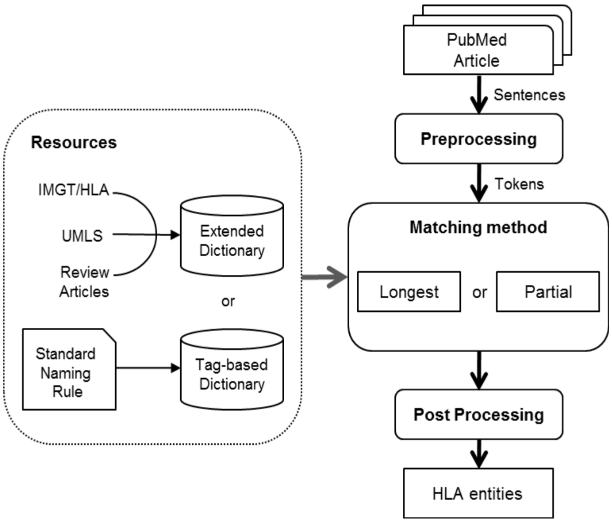
- *This was the only detected amino acid difference between A2.4c and A2.2Y*
- *The most prevalent DQB1 alleles in Kuwaiti schizophrenia patients were *0601 , *0201 and *0501*
- *The most frequent alleles were DRB1*1301 (23.5%), DQA1*0103 (29.4%), *0501/03/05 (29.4%), and DQB1*0301/09 (32.4%) in the Ket, and DRB1*0901 (25%), DQA1*0301 (39.6%), and DQB1*0301/09 (37.5%) in the Nganasan*

In this paper, we aim to develop partial matching method to improve a dictionary-based approach and apply to HLA domain. This method can identify entities that have omitted words. The longest matching method uses sequence and adjacency information of words. But, the partial matching method only uses sequence information of words. So this method isn't sensitive about omitted words.

The following sections give some details of the method. In particular, section 2 describes the task of our approach and explains our application. Section 3 shows the experimental results and discussion, and finally, section 4 presents our conclusion.

2. Method and its application

Our target entities are HLA gene, antigen, and allele. The proposed system consists of three steps: preprocessing, matching and post-processing steps. <Fig. 1> shows the system architecture. In the preprocessing step, each sentence is divided to tokens. In the matching step, our system extract HLA entities using two constructed dictionary and two matching method. One of the dictionaries is extended dictionary, which include synonym of entities. The other dictionary is a tag-based dictionary, which help to recognize entities that newly are discovered. In order to extract entities, we used longest matching method and partial matching method. In the post processing, we expanded the boundary of the entities through including some prefix (e.g. HLA, human) and some postfix (e.g. gene, antigen, and allele)



<Fig. 1> System architecture

2.1 Preprocessing

Tokenizing is the task of splitting a text into a word, called a token is used a basic processing unit. Tokenizing method varies by the character of domain since choosing a separator is dependent on domain. The special character (e.g. -, /, space, and so on) is a typical example of separators. However, this charaters are not sufficient for HLA domain. For example, if we extract 'DR1', 'DR2', 'DR3', 'DR4', 'DR7', 'DR8', 'DR11', 'DR12', and 'DR13' entities from *"DR1, 2, 3, 4, 7, 8, 11, 12 and 13 homozygous typing cells"* sentence, we must divide one token concurrently having characters and numbers into two or more tokens.

After tokenizing process, if token is related to appropriated tag shown <Table 1>, the tag is attached at the token. This tag will be used by the longest and partial matching method using tag-based dictionary.

<Table 1> Tagged token about HLA antigen and allele

Tag	Token string
Antigen locus	A, B, Bw, Cw, Dw, DR, DRw, DQ, DQw(DQ),DP(DPw), DPw
Antigen specificity	{specificity} #1st~4th digits
Allele locus	A, B, C(Cw), Cw, DMA, DMB DMB1(DMB), DOA, DOB, DOB1(DOB) DPA1, DPA(DPA1), DPB1, DPB(DPB1) DQA1, DQA(DQA1), DQB1 DQB(DQB1), DRA, DRA1(DRA) DRB(DRB1), DRB1, DRB2, DRB3 DRB4, DRB5, DRB6, DRB7, DRB8 DRB9, E, F, G, H, J, K, L, MICA, MICB P, TAP1, TAP2, V
Allele specificity	{specificity} #2nd~8th digits
Allele suffix	L, N, S, C, A, Q

2.2 HLA Dictionary

We built two new dictionaries, extended dictionary and tag-based dictionary for recognizing HLA entities.

2.2.1 Extended dictionary

We built new extended dictionary by using various sources. We collected HLA terminology by using IMGT/HLA, UMLS Metathesaurus, and two review articles. Unified Medical Language System (UMLS) Metathesaurus is very large databases. About 100 biomedical dictionaries has unified and organized by concepts. There are 54 HLA gene concepts in 2007 UMLS. IMGT/HLA provides information about 53 HLA gene, 78 previous equivalents about HLA gene, 166 antigen and 3174 allele. In order to expand the dictionary, we also collected 1,233 official name and 982 synonym about HLA allele from [11] and 1,999 official name and 1,569 synonym about HLA allele from [12].

The collected terminology has dictionary extension and general keyword filtering process. There are many variations in HLA entities. HLA gene could start with 'HLA' instead of 'HLA-'. HLA gene, antigen, and allele could have 'w', 'W' or nothing after locus. HLA allele has either suffix such as L, N, S, C, A and Q or no suffix. HLA alleles are the same whether their suffix is omitted or not. We removed record containing 'DNA', 'C2' or 'C4' in our extended dictionary since they are likely to be DNA or Protein names rather than HLA terminology.

2.2.2 Tag-based dictionary

Extended dictionary makes it easy to find synonyms of official names. However, it is not enough to find new entities and variations of the existing entities. To solve these problems, we marked a naming rule using a standard naming convention of HLA gene, antigen, and allele. Tag-based dictionary was constructed by this rule. The rules are represent the tag(<Table 1>) and are as follows. The Symbol '{ }' is tag.

1) HLA gene is represented by either 'HLA-{Antigen locus}' pattern or 'HLA-{Allele locus}' pattern.
2) HLA antigen is represented by '{Antigen locus}{Antigen specificity}' pattern.
3) HLA allele is represented by '{Allele locus}* {Allele specificity}{Allele suffix}' pattern.
4) 'HLA-' keyword in HLA gene could be omitted when its locus part has more than 2 letters.
5) 'HLA-' keyword can appear in HLA antigen and allele.
6) Space can appear before locus part instead of '-'.
7) 'w' can appear follow {Antigen locus} and {Allele locus}.
8) Space can appear before 'w' in 7.

9) ‘A’ and ‘B’ that appear at third letter in {Allele locus} can be substitute for ‘alpha’ and ‘beta’ respectively.

10) {Allele suffix} can be omitted.

11) {Antigen specificity} and {Allele specificity} should satisfy the following conditions:
   a) Specificity has at most 8 numbers.
   b) Specificity does not have range(e.g. 3-5)
   c) Specificity does not have symbols like ‘+’, ‘*’,‘=’,‘<’,‘>’ before numbers and symbol like ‘$’ and ‘%’ after numbers.
   d) In ‘{specificity}/{specificity}’ pattern, the number count of second specificity smaller than the ones of first specificity hints at omission of some part of first {specificity}.
   e) ‘({Specificity}/{specificity})’ pattern is not specificity.

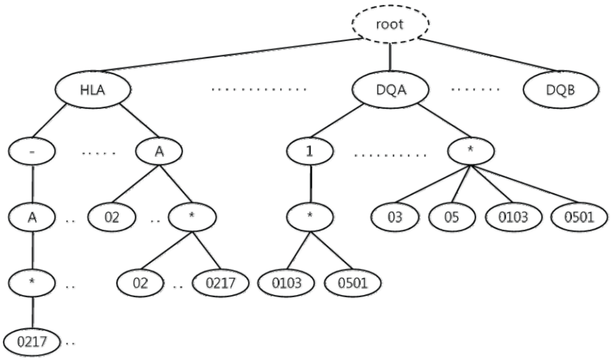
Tag-based dictionary has an advantage that can extract entities, which are not in the extended dictionary but also has a disadvantage that can extract unrelated entities. If the system used tag-based dictionary, it needs a validity analysis in order to remove unrelated entities. In detail, allele specificity noted with a variety of levels of detail. The first two digits mean serology specification. The third through fourth digits mean the difference of amino acids. Digits five through six denote nucleotide and the seventh and eighth digits denote the difference of noncoding regions.

2.3 Matching method

The longest matching method is widely used in the biomedical domain. At perspective of the practical use, the longest matching method uses a trie, ordered tree data structure, for speed. We proposed partial matching method, which can recognize an entity having ellipses. This method also uses a trie for speed.

2.3.1 Longest matching method

Longest matching method extract entity set that does not overlap between entities. When entities are overlapped, we give priority to the longer entity. For example, we want to get the exact entity ‘HLA-A*0217’ in *“The peptide-binding motif of HLA-A*0217”*. When we use the longest matching method in the sentence, two entities ‘HLA-A*0217’ and ‘A*0217’ are identified. Then ‘HLA-A*0217’ is extracted, because it is the longest one.



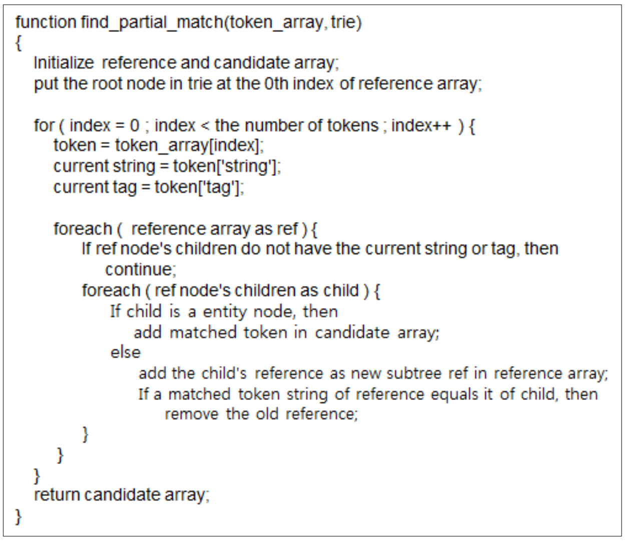
<Fig. 2> Trie of extended dictionary

We use priorities that are reflected a character in HLA domain, in addition to this method. There are two priorities for longest matching method. In the first, if matched entities are overlapped each others with the same length, and then we select the most-right entity because the noun phrase has a head on the right side. If HLA gene and allele are overlapped in candidate entities, HLA allele is extracted because HLA alleles always include HLA gene information. We constructed the trie from two dictionaries. <Fig. 2> shows the part of the trie derived from the extended dictionary. In the trie, each node is a token in the dictionary.

2.3.2 Partial matching method

The partial matching method is able to recognizing entities having ellipses word. For briefing our method, we defines some terms firstly. We define the entity in which all tokens are adjacent as a _simple entity_, the entity in which all tokens aren't adjacent as an _extended entity_. We define tokens which aren't shared with other entities as an _entity base_, and tokens which are shared with other entities as a _hidden token._ We define the number of tokens which aren't matched with the entity between the leftmost token and the rightmost as a _miss count_ of extended entity. For example, in the sentence "_…with DNA from DR1, 2, 3, 4, and 13 homozygous typing…_", we can recognize 'DR1', 'DR2', 'DR3', 'DR4' and 'DR13' using partial matching method. 'DR1' is a simple entity. 'DR2', 'DR3', 'DR4' and 'DR13' are extended entities. In the 'DR2' entity, an entity base is the '2', and hidden tokens of the entity is the 'DR', and a miss count of the entity is 2.

The partial matching method divided two part : finding candidate entities and selecting the most valid candidate entity. <Fig. 3> is the pseudo code of finding candidate entities.



```
function find_partial_match(token_array, trie)
{
  Initialize  reference and candidate array;
  put the root node in trie at the 0th index of reference array;

  for ( index = 0 ; index < the number of tokens ; index++ ) {
     token = token_array[index];
     current string = token['string'];
     current tag = token['tag'];

     foreach (  reference array as ref ) {
         If ref node's children do not have the current string or tag, then
             continue;
         foreach ( ref node's children as child ) {
             If child is a entity node, then
                 add matched token in candidate array;
             else
                 add the child's reference as new subtree ref in reference array;
                 If a matched token string of reference equals it of child, then
                     remove the old reference;
         }
     }
  }
  return candidate array;
}
```

<Fig. 3> Candidate search algorithm pseudo code

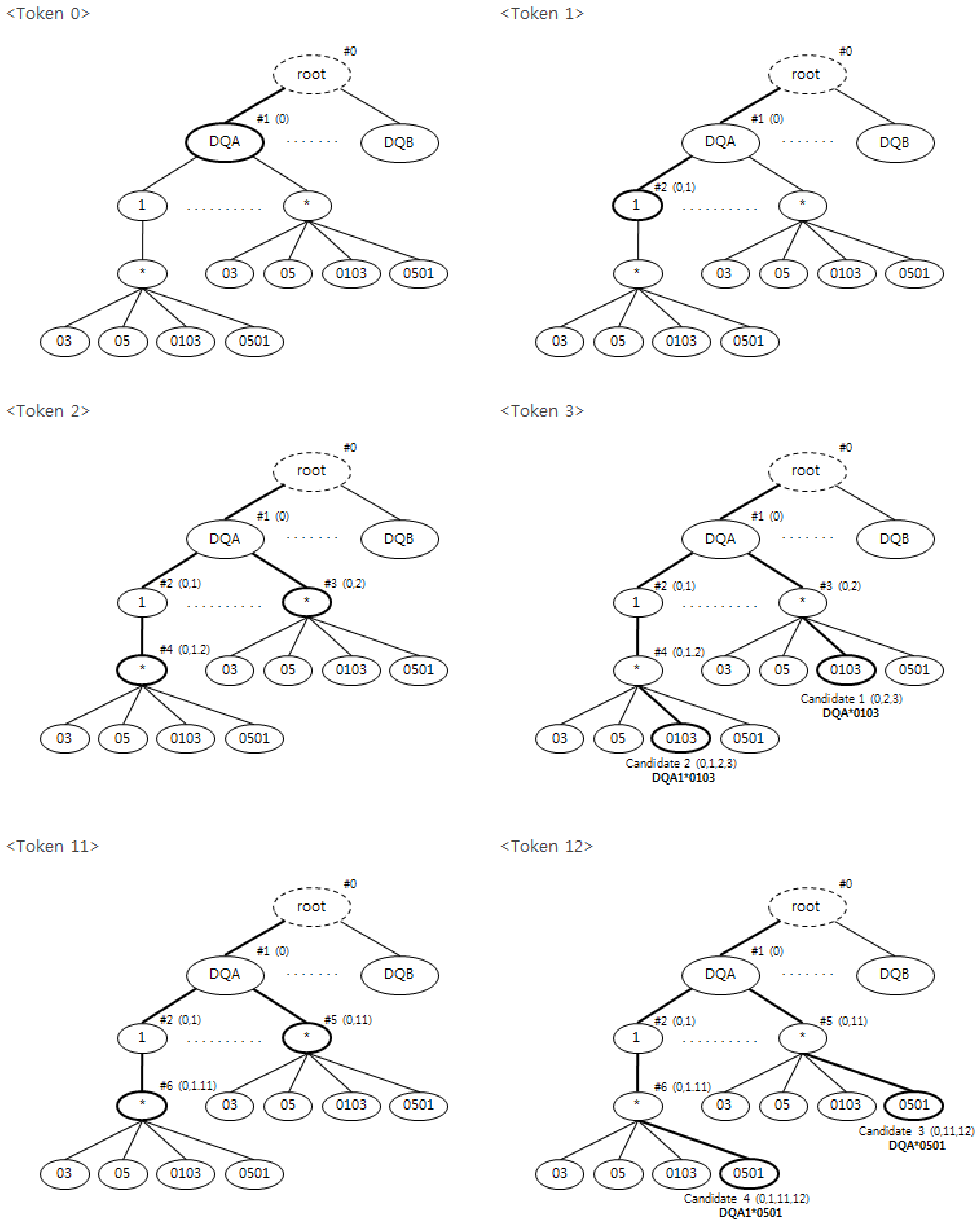
<Table 2> and <Fig. 4> describe the process of finding candidate entities about "… _DQA1*0103 (29.4%), *0501/03/05 (29.4%), and DQB1*0301/09 …_ " sentence. Tokens that were not registered in dictionary were omitted for saving spaces in <Table 2>.

<Table 2> Partial matching step

Token		Found ref	New Subtree ref	Matched Token idx	Matched token string
idx	string				
0	DQA	#0	#1	0	DQA
1	1	#1	#2	0,1	DQA 1
2	*	#1	#3	0,2	DQA *
		#2	#4	0,1,2	DQA 1 *
3	0103	#3		0,2,3	**DQA * 0103 (1)**
		#4		0,1,2,3	**DQA 1 * 0103 (2)**
11	*	#1	#3→#5	0,11	DQA *
		#2	#4→#6	0,1,11	DQA 1 *
12	0501	#5		0,11,12	**DQA * 0501 (3)**
		#6		0,1,11,12	**DQA 1 * 0501 (4)**
14	03	#5		0,11,14	**DQA * 03 (5)**
		#6		0,1,11,14	**DQA 1 * 03 (6)**
16	05	#5		0,11,16	**DQA * 05 (7)**
		#6		0,1,11,16	**DQA 1 * 05 (8)**
25	DQB	#0	#7	25	DQB
26	1	#1	#2→#8	0,26	DQA 1
		#5		0,11,26	**DQA * 1 (9)**
		#6		0,1,11,26	**DQA 1 * 1 (10)**
		#7	#9	25,26	DQB 1
27	*	#1	#5→#10	0,27	DQA *
		#7	#11	25,27	DQB *
		#8	#6→#12	0,26,27	DQA 1 *
		#9	#13	25,26,27	DQB 1 *
28	0301	#10		0,27,28	**DQA * 0301 (11)**
		#11		25,27,28	**DQB * 0301 (12)**
		#12		0,26,27,28	**DQA 1 * 0301 (13)**
		#13		25,26,27,28	**DQB 1 * 0301 (14)**
30	09	#10		0,27,30	**DQA * 09 (15)**
		#11		25,27,30	**DQB * 09 (16)**
		#12		0,26,27,30	**DQA 1 * 09 (17)**
		#13		25,26,27,30	**DQB 1 * 09 (18)**

<Fig 4> shows the status of the trie about token 0, 1, 2, 3, 11, and 12. The description of six tokens is as follows.

- Token 0 : The method started with token 'DQA', and found 'DQA' node from root(#0) reference. The 'DQA' node was named as reference #1.
- Token 1 : Since root node (#0) does not have '1' as child node, the method skipped the root node. Because reference #1 had '1' as child node, the '1' node was named as reference #2.



<Fig. 4> Partial matching step in trie

- Token 2 : The method found the two references having '*' as child. So, two references #3 and #4 are added.

- Token 3 : There were two nodes marked as entity node. So the method extracted all node in the path from root to the final node as candidate entities (DQA*0103, DQA1*0103)

- Token 11 : The method could find two '*' nodes at children of DQA and 1, but the children already had the reference number, so the method assigned them with new reference #5 and #6.
- Token 12 : Like Token 3, the method extracted two candidate entities (DQA * 0501, DQA 1 * 0501)

The partial method found the 18 candidate entities represented with bold font in <Table 2>. However, these candidate entities included not only true positive entity, but also false positive entity. The partial matching method use sequence information for extracting entities from literature rather than adjacency information. Because the partial matching method ignores adjacent information, there exist a lots of candidate entities. For example, the method extract 4 candidate entity such as, 'DQB3*0105', 'DQB3*0302', 'DQB1*0302' and 'DQA1*0302' according to sequence information from sentence *"A few haplotypes carrying DQB3*0105 had DQA1*0302"*.

Candidate entities that are extracted with the partial matching method needed to have a filtering process to remove false positive entities. There are two steps for the filtering. The first step is to remove inappropriate entity base. The second step is to select best entities among candidate entities that share the same entity base.

In detail, for the first step, we removed some candidate entities by choosing only one candidate among candidates whose base is overlapped partially. Since a base of entities should not be shared with other entities, only one among the overlapped entities is correct. We use following priority (first listed is the highest priority, last listed is the lowest) to select non-overlapped entity bases.

1) Candidate entity having longer entity base
2) Candidate entity having smaller miss count
3) Candidate entity having total matched token (hidden + base token)
4) Candidate entity having rightmost entity base

2, 3, 4, 5, 6, 7, 8, 14, 15, 16, 17 and 18 candidates in the table are passed.

The second step is to select the correct hidden token in candidate entities that share the same entity base to each other. In the previous example, the 15, 16, 17 and 18 candidate entities share the same entity base with their own different hidden tokens. The priority (first listed is the highest priority, last listed is the lowest) in order to search unique candidate per each entity base is as follows.

1) Candidate entity having small miss count
2) Candidate entity having large extended length
3) Candidate entity having leftmost entity base

2, 14, 4, 18, 6, 8 candidate entities are passed from 2, (3,4), (5,6), (7,8), 14, (15,16,17,18).

Finally, if there is no simple candidate entity in the literature, extracted all extended entity is removed because the probability of being true positive is very low.

2.4 Postprocessing

From the above NER result, we extend an entity boundary. In order to build entity full name, we perform this process. Entity boundary is extended by using special token, such as 'HLA-', 'HLA', '-', 'gene', 'antigen', and 'allele'. Generally, 'HLA-', 'HLA', and '-' tokens appear on the left side of entities and 'gene', 'antigen', and 'allele' tokens appear on the right side of entities. Thus we generate entity full name by adding special token on the left and right side of entity.

3. Results and discussion

We randomly collected 100 papers for training and 200 papers for testing from 50,848 papers that were searched by the keyword *"HLA antigens[MeSH Terms] AND (1[PDAT]: 2006[PDAT])"*. One domain expert extracted HLA gene, antigen, and allele entities in the collected papers. If an entity is cascaded, the expert selects the longest entity. Thus we found 448 HLA genes, 589 HLA antigens, and 140 HLA allele. Our proposed partial matching method shows high recall with acceptable precision.

3.1 NER Results in HLA domain

We used regular expression using a standard naming convention[10] and had four different NER process that combined by two matching methods and two dictionaries. The results, which recognized HLA gene, antigen, and allele, are shown in <Table 3>. Each rows shows the precision, recall, and F-score in cell. The best F-score was 97.59% when HLA allele is extracted using the partial matching method and tag-based dictionary.

In HLA gene results, the F-score was 91.18% of the experiment 4. This is the recall 40% higher than the experiment s. The recall of an experiment s is the lowest because it does not have many variation notations about HLA gene. The extended dictionary which is constructed through the UMLS and IMGT/HLA includes gene synonym. However, the extended dictionary and tagged based dictionary had an equal recall. We could know that the extended dictionary did not affect recognition of HLA gene. On the other hand, when we compared longest matching method with partial matching method, the precision a little degraded while the recall efficiency improved. That is, partial matching method improved the performance of the recall.

<Table 3> NER Results in HLA domain

(s):expression using standard naming convention, (1):longest matching method + extended dictionary, (2):longest matching method + tag-based dictionary, (3):partial matching method + extended dictionary, (4):partial matching method + tag-based dictionary

Target	(s)	Longest matching method		Partial matching method	
		(1)	(2)	(3)	(4)
HLA gene	94.59% 40.70% 56.91%	91.32% 85.66% 88.40%	92.08% 85.66% 88.76%	85.52% 96.12% 90.51%	86.71% 96.12% **91.18%**
HLA antigen	79.48% 76.25% 77.83%	88.22% 91.25% **89.71%**	90.10% 82.50% 86.13%	83.33% 92.19% 87.54%	87.25% 83.44% 85.30%
HLA allele	85.71% 70.59% 77.42%	100.00% 84.71% 91.72%	100.00% 87.06% 93.08%	100.00% 87.06% 93.08%	100.00% 95.29% **97.59%**

In HLA antigen results, experiment 1 had the highest F-score of 89.71%. We improved the recall using the extended dictionary. In order to expand the dictionary, we adopted many synonyms using two review articles. It was meaningful to extract HLA antigens, because HLA antigens have many synonyms. However, the recall of the longest matching method and partial matching method(experiment 1 vs 3, experiment 2 vs 4) marked almost no difference, because HLA antigens appeared few omissions of a repeated word.

In HLA allele results, experiment 4 had the highest F-score of 97.59%. The recall of the tag-based dictionary is higher than the recall of the extended dictionary. And partial matching method efficiently also helps to identify HLA alleles, which were omitted a repeated word. Specially, experiment 1, 2, 3, and 4 had the precision of 100%. We think that this result is due to use the proposed filter for removing false positive in the partial matching method. And '*' token in HLA allele also reduces ambiguity of NER.

3.2 Discussion

Partial matching method is effective on the domain that notation form is standardized. '{Allele locus}*{Allele specificity}{Allele suffix}' pattern is one of such examples. This method is easily applicable for alleles of apolipoprotein E, group-specific component, ABO blood group, duffy blood group, SLC14A1 gene because their notation forms are similar with HLA allele's forms. In addition, it also can improve the performance for recognition of gene and protein. Expressions containing omission of words such as *"HNF-3alpha, -3beta, -3gamma, -4gamma, and -6"* or *"IL-2, -4, -7, -9, and -15"* can be recognized with partial matching method.

However, partial matching method also has a limitation due to its assumption that words on the left of sentences are likely to be omitted more than words on the right. Therefore, when words on the right of sentences are omitted, this kind of omissions may be hard to be recognized with this method. For example, in the *"T and B lymphocytes"*, if 'lymphocytes' token is omitted, then it is hard to guess 'T lymphocyte' with such omission. This kind of problem can be solvable with the modification of candidate entities filtering process for such kind of omission.

4. Conclusion

We proposed the partial matching method that is applicable to the domain that parts of entities can be frequently omitted. HLA domain is one of the such domains. With the traditional method, the recall of NER is lower than precision. To improve performance of named entity recognition, we devised the partial matching method and developed the extended and tag-based dictionary. Experiments show that the longest matching method was effective in recognizing HLA antigen entities, in which the ellipses are rare, and the partial matching method was effective in recognizing HLA gene and allele entities, in which the ellipses are frequent. In result, we can achieve high F-scores of 91.18%, 89.71% and 97.59% for HLA gene, HLA antigen and HLA allele respectively.

References

[1] H. W. Chun, Y. Tsuruoka, J. D. Kim, R. S hiba, N. Nagata, T. Hishiki, and J. Tsujii, "Extraction of gene-disease relations from medline using domain dictionaries and mac hine earning," vol. 11. Citeseer, 2006, p. 4 15, pac Symp Biocomput.

[2] T. C. Rindflesch, L. Tanabe, J. N. Weinstei n, and L. Hunter, "Edgar: extraction of dru gs, genes and relations from the biomedica l literature," vol. 2000, 2000, pp. 515 - 524, pac Symp Biocomput.

[3] Finkel, J., S. Dingare, et al. (2005). "Explor ing the boundaries: Gene and protein identi fication in biomedical text." BMC bioinform atics 6(Suppl 1): S5.

[4] M. Goldberg, "An unsupervised model for statistically determining coordinate phrase attachment." Association for Computational Linguistics Morristown, NJ, USA, 1999, pp. 610 - 614, proceedings of the 37thannualmee tingoftheAssociationforComputationalLinguis ticsonComputationalLinguistics.

[5] F. Chantree, A. Kilgarriff, A. De Roeck, an d A. Willis, "Disambiguating coordinations using word distribution information," 2005, proceedings of Recent Advances in Natural Language Processing (RANLP).

[6] Dale, R. and P. Mazur (2007). "Handling con junctions in named entities." LECTURE NO TES IN COMPUTER SCIENCE 4394: 131.

[7] Tanabe, L. and W. Wilbur (2002). "Tagging gene and protein names in biomedical text." Bioinformatics 18(8): 1124.

[8] E. Buyko, K. Tomanek, and U. Hahn, "Resolution of coordination ellipses in biological named entities using conditional random fields," 2007, pp. 163 - 171, pACLING 2007-Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics.

[9] EMBL-EBI. (2009) Imgt/hla database. [Online]. Available: http://www.ebi.ac.uk/imgt/hla/

[10] J. M. Chae, K. N. Park, Y. H. Jung, S. Y. Jung, J. E. Chae, and H. B. Oh, "Hadextract: Extracting hla-disease interaction information from biomedical literature," vol. 3, 2008, future Generation Communication and Networking, 2008. GCN'08. Second International Conference on.

[11] G. M. T. Schreuder, C. K. Hurley, S. G. E. Marsh, M. Lau, M. A. Fernandez-Vina, H. J. Noreen, M. Setterholm, and M. Maiers, "Hla dictionary 2004: Summary of hla-a,-b,-c,-drb1/3/4/5,-dqb1 alleles and their association with serologically defined hla-a,-b,-c,-dr, and-dq antigens," Human immunology, vol. 66, no. 2, pp. 170 - 210, 2005.

[12] S. G. E. Marsh, E. D. Albert, W. F. Bodmer, R. E. Bontrop, B. Dupont, H. A. Erlich, D. E. Geraghty, J. A. Hansen, C. K. Hurley, and B. Mach, "Nomenclature for factors of the hla system, 2004," Human immunology, vol. 66, no. 5, pp. 571 - 636, 2005.

채정민

2003 고려대학교
컴퓨터교육과(이학학사)
2005 고려대학교
컴퓨터교육과(이학석사)
2005~현재 고려대학교 컴퓨터교육과 박사과정
관심분야: 텍스트마이닝, 바이오인포매틱스
E-Mail: onacloud@korea.ac.kr

정영희

2001 한성대학교
정보공학과(공학사)
2005 고려대학교 교육대학원
컴퓨터교육과(교육학석사)
2005~현재 고려대학교 컴퓨터교육과 박사과정
관심분야: 컴퓨터교육, 텍스트마이닝, 정보검색,
질의응답시스템
E-Mail: coolof@korea.ac.kr

이태민

2008 고려대학교
컴퓨터교육과(이학학사)
2008~현재 고려대학교
컴퓨터교육과 박사과정
관심분야: 텍스트마이닝, 감성분석, 정보공학
E-Mail: persuade@gmail.com

채지은

1998 동국대학교
정보통신학과(이학학사)
2007 고려대학교
컴퓨터교육과(이학석사)
2009 University of Pennsylvania Computer and Information Science(이학석사)
2009~현재 네이버 검색서버팀 대리
관심분야: 기계번역, Information extraction
E-Mail: jieun.chae@nhn.com

오흥범

1996 서울대학교병원(전공의)
1996~1998 대한적십자사
혈액수혈연구원(연구과장)
2001~2002 Harvard School of
Public Health 방문교수
1998~2000 울산의대 서울아산병원
진단검사의학과 전임강사
2000~2004 울산의대 서울아산병원
진단검사의학과 조교수
2004~2009 울산의대 서울아산병원
진단검사의학과 부교수
2009~현재 울산의대 서울아산병원
진단검사의학과 교수
관심분야: 진단검사, 통계학, 바이오인포메틱스 등
E-Mail: hboh@amc.seoul.kr

정순영

1990 고려대학교 이과대학
전산과학과(이학사)
1992 고려대학교 이과대학
전산과학과(이학석사)
1997 고려대학교 이과대학
전산과학과(이학박사)
2000~2003 고려대학교 컴퓨터교육과 조교수
2003~2008 고려대학교 컴퓨터교육과 부교수
2008~현재 고려대학교 컴퓨터교육과 교수
2006~2007 플로리다 주립대학교 방문교수
2008~현재 한국컴퓨터교육학회
논문지편집위원회 부위원장
관심분야: 데이터베이스, 텍스트마이닝,
웹기반교육시스템, 컴퓨터교육 등
E-Mail: jsy@korea.ac.kr