Using Candidate Exploration and Ranking for Abbreviation Resolution in Clinical Documents

Jong-Beom Kim
Mobile Communications Company
LG Electronics Inc.
Seoul, South Korea
beom.kim@lge.com

Heung-Seon Oh, Sang-Soo Nam, Sung-Hyon Myaeng
Department of Computer Science
Korea Advanced Institute of Science and Technology
Daejeon, South Korea
ohs@kaist.ac.kr, sangsoo.nam@kaist.ac.kr, myaeng@kaist.ac.kr

***Abstract*— In biomedical texts, abbreviations are frequently used due to their inclusion of many technical expressions of some length. Accordingly, appropriate recognition of abbreviations and their full form pairs is an essential task in automatic text processing of biomedical documents. However, unlike the biomedical literature, clinical notes have many abbreviations without their full forms available in the text or without standard definitions in dictionaries due to the nature of the documents. This causes difficulties in adapting traditional approaches for abbreviation disambiguation such as classification among fixed candidates or pattern-based definition extraction. Because of this reason, we consider the task as search problem and propose an approach with two steps: a) exploring possible full form candidates from various resources and b) choosing most acceptable one among retrieved candidates by ranking. To discover full form candidates and their features, we exploited external academic resources such as MEDLINE and UMLS as well as the clinical note corpus itself. To rank the candidates properly based on human criteria, we adopted RankBoost, one of the learning-to-rank models developed from information retrieval and machine learning communities. Experimental results show the suggested two-step approach is promising for this kind of task.**

Keywords—Abbreviation Resolution; Learning to Rank; Medical Text Processing

I. INTRODUCTION

As the amount of medical records in electronic form grows and computing power improves, the importance of automatic processing of medical documents has been increasing. One of the most significant issues in automatic text processing is to recognize the original intent of the author for an ambiguous expression. Two major factors that determine the degree of ambiguity in this task are the number of meaning candidates and the firmness of candidates for the expression. The more possible meanings it has, the more error-prone the disambiguation process is. If a fixed set of meaning candidates or definitions are given, in addition, it would be easier to handle it than the case where the candidates are not fixed. Considering these aspects, it can be inferred that recognizing and disambiguating abbreviations in informal texts is challenging.

Our research team found the importance of an appropriate handling of medical abbreviations as we attempted to develop an automatic cancer staging system and an automatic literature retrieval system, both for clinical notes. Without a proper interpretation of abbreviations, accuracy of those systems would deteriorate. Wren et al. [23] presented the necessity of finding definitions of biomedical abbreviations by performing document searches using abbreviations and their full forms alternatively.

A few departments or whole medical institutions such as Mayo Clinic and Seoul National University Hospital have dictionaries defining abbreviations used in their clinical notes, which were constructed manually [14]. These dictionaries are not easily sharable with others since some definitions from those dictionaries are only appropriate and used inside the groups where they were built. Moreover, these kinds of dictionaries need to be updated periodically, since new abbreviations are introduced as time goes by.

Many studies have been done on handling abbreviations in the text mining and text classification communities, and we found that a significant portion of the research past research was conducted in the biomedical domain. They include the studies on tasks such as normalization of gene names, extraction of abbreviation definitions, and classification of abbreviation meanings [17, 24, 25]. Liu and her colleagues reported that about a third of abbreviations appearing in UMLS (Unified Medical Language System) have multiple meanings and that UMLS definitions can only cover about two third of abbreviations appeared in medical reports from the New York Presbyterian Hospital (NYPH) Clinical Data Repository [7].

Our survey on related prior studies shows that previous approaches to recognizing definitions of abbreviations from the medical literature are not applicable for our task, due to the difference of data characteristics and the lack of existing definitions [6, 13, 14]. Thus we propose a different approach to this problem, which makes use of the learning-to-rank method developed for information retrieval.

In the following sections, we discuss about what problems we found in our clinical note corpus and how we resolved them. In Sections 2 and 3, we describe our analysis of the data and accompanying issues and problems as well as the previous work related to our research issues. Section 4 explains the approaches we propose. A description of our experimental

* This research was funded by Basic Science Research Program through the National Research Foundation of Korea(NRF).

978-0-7695-5089-3/13 $26.00 © 2013 IEEE
DOI 10.1109/ICHI.2013.44

settings and results are given in Section 5, followed by the Sections for discussion about the results and conclusion.

II. Abbreviation Resolution

A. Abbreviations in Medical Documents

In the biomedical domain, the types of documents vary according to their purposes and usages: journal articles, case reports, discharge summaries, radiology records and so on. These documents of many kinds can be classified into two big categories – clinical documents and medical literature. *Clinical documents* include radiology reports, progress notes and discharge summaries, which are written quickly for clinical purposes. On the other hand, the documents in *medical literature* such as medical journal articles, systematic reviews and case reports are written with academic purposes in the biomedical field. One of the largest and well-known sources of medical literature is MEDLINE [29].

Since clinical notes and medical literature belong to the same domain, they share some common vocabulary such as names of body parts, diseases and treatments. However, they also show differences in styles and structures. The language used in medical literature is well-formed with controlled vocabulary. When a new term or expression appears, an explicit definition usually follows. Compared to the literature, clinical documents are generally written in informal language. For example, they usually contain undefined expressions, symbols and numbers although the degrees of informality may vary in accordance with the detailed document types. In addition, the lengths of most clinical documents are shorter than those of research article and case reports because observed facts and diagnoses are recorded without a verbose explanation.

While the documents in both types all contain many shortened expressions, abbreviations in clinical notes are more difficult to interpret because of the following reasons. First, definitions (full forms) of abbreviations are often absent in the corpus or difficult to find the connections even if they exist. Clinicians simply do not have the time to describe the long forms or feel the need to do so as the abbreviations are expected to be used by a small group of local people. Even if a full form of an abbreviation is used in some notes, documents containing the full form normally do not use the corresponding abbreviations. This makes it difficult to find a connection between an abbreviation and its full form. Second, some abbreviations in the clinical corpus do not have an explicit definition even outside the corpus. Some abbreviations in a certain department or institution are used among the members of the group without following standard abbreviation list because they tend to share the implied meanings. To find full forms of those abbreviations, the first thing to do is find possible full form candidates.

In this research, we are dealing with a clinical note corpus from Seoul National University Hospital. We extracted all the abbreviations used in the corpus manually. During the process, we found that abbreviations are ambiguous without the context, but the meaning of each abbreviation in the corpus is unique. That is because documents in the corpus share the same topic and the same department; the vocabulary is limited to the group. However, we recognized that the full forms of some abbreviations are not found in existing standard medical dictionaries. To verify this, we categorized all the abbreviations by ambiguity and existence of definitions, by mapping to the entries in the UMLS lexicon. The categorization result is described in Table 1.

They are categorized into four cases: A) abbreviations that have a single full form in UMLS and that is the answer, B) abbreviations that have a unique full form in UMLS but are not the intended ones, C) abbreviations that have multiple full form entries in UMLS, among which the correct answer exists, and D) abbreviations that multiple full form mappings but none of which is the answer.

B. Problem Definition

We have a set of clinical documents from a single department within a single domain. A list of abbreviations used in the set is available but their full forms are not unknown. The goal of this research in this situation is to identify the full form for each abbreviation as intended in the given corpus. The goal of the research is shown schematically in Figure 1.

TABLE I. Analysis of Abbreviations from Given Clinical Corpus

Case	Numbers (110 in total)	Example Abbreviation	UMLS full forms	Intended full form
A	10 (9.1%)	RLQ	**right lower quadrant** (Unique)	**right lower quadrant**
B	2 (1.8%)	LK	lymphokine (Unique)	**left kidney** (Not in UMLS)
C	68 (61.8%)	GB	**gallbladder**, gigabyte, glial bundle (3 meanings)	**gallbladder**
D	30 (27.3%)	CI	cephalic index, cerebral infarction, chemoimmunotherapy, … (15 meanings)	**clinical information** (Not in UMLS)

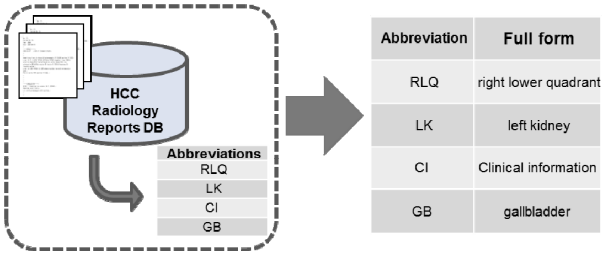

Figure 1. Problem Definition of the Research

III. Related Work

Since the abbreviation resolution problem is a significant issue in medical text processing, as mentioned in the previous sections, a large amount of research efforts has been devoted to the task in the biomedical informatics community. Among the various studies regarding abbreviations, those related to our task can be categorized into two classes: automatic definition extraction and abbreviation classification. Both have similarities and differences with our method in certain aspects, and the idea from one of those has been partially adopted for our approach.

A. Automatic Definition Extraction

One popular type of method dealing with abbreviations in the biomedical domain is extracting a definition from academic papers. Since journal articles and case reports are written with academic purposes, the meaning or definition of newly introduced terms are explicitly described, sometimes in the long form of an abbreviation, typically in the pattern of a parenthetical expression as in "left arterial pressure (LAP)" for example. Some studies exploited this property of abbreviation definitions to extract <abbreviation, full form> pairs. Schwartz and Hearst [17] utilized this pattern and proposed a simple java code implementing a rule-based method to result in high accuracy. They applied their algorithm to MEDLINE abstracts and achieved 95% precision and 82% recall. The paper is considered a standard method for abbreviation dictionary construction in MEDLINE. Okazaki and Ananiadou [11] also exploited the pattern but with a statistical model instead of a rule-based one. Their approach also shows promising results.

B. Abbreviation Classification

Another class of research on abbreviation in the biomedical field is abbreviation classification. Assuming that an abbreviation has multiple usages, the task is to identify the correct full form among multiple candidates that are selected from a lexicon such as UMLS or given by a human annotator. This task is very similar to general text classification such as document classification or POS tagging. An abbreviation is classified into one of the candidates using the features around the abbreviation. Studies in this task typically use machine learning approaches, either supervised or semi-supervised, which require a training example for each class. Most machine learning models already developed are applicable in this method.

Joshi and his colleagues [6] reported a study in which three supervised learning models were compared with different combinations of various features. Pakhomov and his colleagues [14] proposed a semi-supervised method for abbreviation classification and also built a sense inventory in order to expand the comparison set for full form matching. Although the semi-supervised method showed weaker performance, it provided a potential for this research direction.

C. Other Related Studies

MetaMap is one of the most useful tools in processing biomedical texts [1, 10]. Main function of MetaMap is to map a term or phrase in biomedical documents to UMLS concept. MetaMap includes capability of expanding abbreviation to full forms and classifying each term into biomedical vocabulary classes. However, those functions are not applicable on highly informal or non-English data like our clinical note corpus.

As one of the recent studies on this field, Okazaki and his colleagues suggested method to build high-quality sense inventory by supervised machine learning approach [3]. Their approach is noticeable among others and the result was promising, but it requires a lot of human effort, that both constructing inventory and disambiguating senses need human annotation.

IV. Methodology

A. Overall Approach

In the course of analyzing the abbreviation resolution problem in clinical notes, we identified two major issues to be handled. First, the candidates for the correct full form corresponding to an abbreviation are not necessarily known in advance. Unlike the previous studies where either an explicit definition exists near the abbreviation or a few candidates (classes) of the full form exist, our task does not assume a firm set of candidates that includes the answer. Second, assuming that a set of possible full form candidates is given and the answer is included in the list, selecting the appropriate candidate from the set is a ranking problem in some sense, rather than a traditional classification task.

To make it easier to understand the task, we would like to compare the problem with the document retrieval task. As documents are retrieved by a search engine based on their relevance with a given query, possible full form candidates corresponding to a given abbreviation are retrieved at the candidate exploration step. After that, in the way retrieved documents are ranked by relevance scores and/or other measures (e.g. PageRank) for document retrieval, full form candidates are ranked by their likelihood of being a correct long form of the given abbreviation.

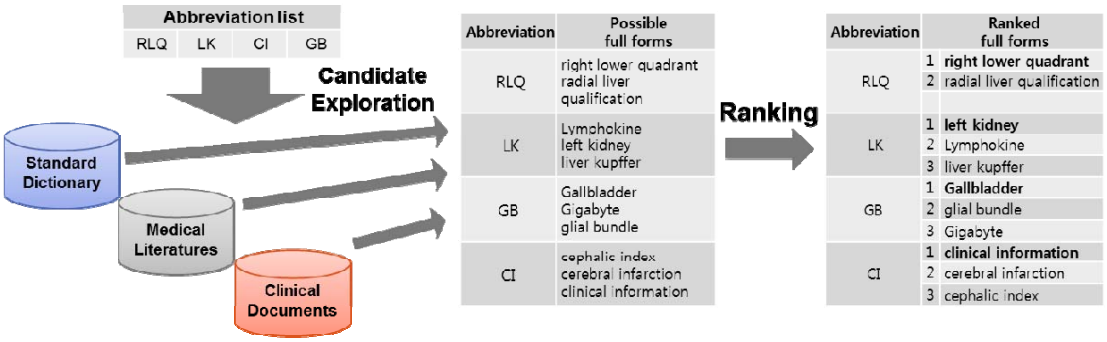

Figure 2. Overall Flow of the Proposed Approach

In the following subsections, we discuss about the two main issues in more detail and explain our approach to solving the problem and the tools and resources adopted at each of the steps. Some methods discussed here have already been used in traditional medical text processing and/or information retrieval. The discussion in this section is at theoretical level, and additional detailed and practical information will be reported in the next Experimental Results section.

B. *Candidate Exploration*

In other studies on the abbreviation resolution problem, the set of full form candidate is fixed, from which an answer is chosen. In our task, however, there is no fixed set of candidates because clinical notes we deal with do not usually contain the definitions of the abbreviations. This is because such documents are written briefly by clinicians under a time constraint and hence unlikely to contain definitions of the abbreviations. To handle this problem, we utilized some external resources for the purpose of generating candidates.

We exploited three types of candidate sources – a standard dictionary, a clinical note corpus, and a medical literature corpus. Although a specific example is given for each type in this section, other medical text or vocabulary sources can be also utilized according to their types, either instead of given sources or in addition to them.

In the following, we explain about properties of the source types and how we utilized each of them for candidate exploration.

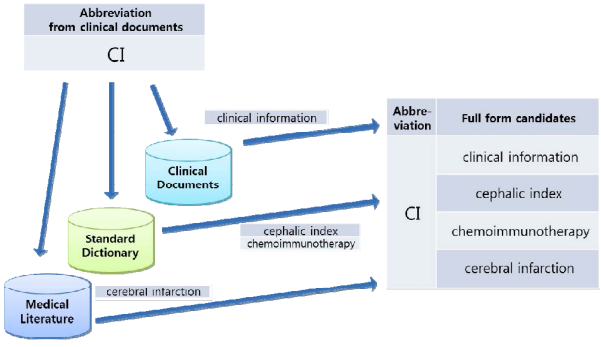

Figure 3. Outline of the Candidate Exploration Process for "CT" as an Abbreviation

1) *Standard dictionary*

Although some non-standard expressions and undefined abbreviations are often found in clinical notes, still many standard medical abbreviations are used (such as HCC representing Hepatocellular Carcinoma) and defined in medical dictionaries. Most abbreviations are mapped to more than one full form in dictionaries, which are then added to candidate full form lists corresponding to the abbreviations. While an abbreviation-only dictionary is most helpful, a general-purpose medical dictionary can be utilized if abbreviations can be distinguished from other non-abbreviated words. In this study, we employed UMLS, a general-purpose standard medical dictionary [28].

A standard dictionary is the easiest type to be used in candidate exploration among other sources since <abbreviation, full form> pairs are defined in a well-structured form. Candidates are just extracted by looking up for an entry that contains abbreviation to be resolved. These definitions can also be used as features, which will be explained more in the experiment section.

2) *Clinical note corpus*

Many clinical abbreviations are found in clinical notes. By 'clinical abbreviation' we mean an abbreviation of which its full form is not defined in a standard dictionary and used only in a specific institution or department with a tacit definition agreed among its members. For ease of writing and time efficiency, frequently used long expressions (phrases that consist of more than two separate words) are written as abbreviations in clinical notes, but sometimes they are written in original long form, according to the author's writing style. For example, some clinicians write 'LC' for 'liver cirrhosis' but others use the whole phrase. This phenomenon allows us to adopt the clinical note corpus as a source for candidate exploration as well as for abbreviations themselves.

As definitions rarely appear in the clinical note corpus, we need a method for exploring candidates. A basic approach is to look for all possible word sequences which are likely to be full form of an abbreviation. For this step, various abbreviation generation schemes can be used in reverse order. For example, Tsuruoka et al. [20] used a machine learning approach to generate acronyms from medical phrases. Although this kind of complicated methods can be used for candidate exploration, we simply extracted n contiguous words of which the head characters form an abbreviation of n characters. For example,

for an abbreviation 'LLL', word sequences like 'left liver lobe', 'left lower leg' would be extracted from the corpus as full form candidates.

3) Medical literature corpus:

One of the most famous medical literature sources is MEDLINE containing a large number of research articles in biomedicine. Given than it is used for academic purposes, the articles are well-structured with standard language and formal vocabulary. Therefore, the medical literature corpus needs to be differently from the clinical note corpus in generating <abbreviation, full form> pair candidates. It turns out that some abbreviations in medical articles can be mapped to their original long forms easily, as many abbreviations come with their definitions with parenthetical expressions. We empolyed two different method for candidate generation:

a) Pattern-based extraction: Here we utilize the property of medical articles that they use rigorous language. Because the authors of the articles usually want to make expressions as clearly as possible, most newly appearing vocabulary accompanies with an additional explanation. Schwartz and Hearst [17] exploited this property to build an abbreviation dictionary successfully; their simple algorithm captures <abbreviation, full form> pairs from MEDLINE abstracts efficiently. In this research, we adopted Schwartz and Hearst's algorithm to capture full form candidates. Even though this method does not generate as many candidates as the other methods, it generates some candidates that the others do not; 1) it includes newly defined abbreviations in the article, which have not been entered in a standard dictionary and 2) it generates complex abbreviations that simple abbreviation generation schemes used in the previous methods cannot handle.

b) Sequence search: We employed the same process of word sequence search used for the clinical note copruse. The result of the search was different from that of the clinical note search, however, because the range and style of vocabulary in medical research articles are different from those of the clinical notes. For example, an abbreviation 'LLL' is used in our clinical note corpus for its original form 'left lower lobe' (more than 80%), but only few articles in the same domain (hepatocellular carcinoma) use it as the abbreviation for the expression. Most articles use the whole phrase instead. Given that the clinical note corpus in this research is from a specific domain, the <abbreviation, full form> pairs generated from the entire scope of medical journals would be an overkill. In this research, we simply limited the scope of the medical literature corpus using PubMed search.

C. Candidate Ranking

1) Characteristics of the task

Once full form candidates for an abbreviation are generated from the previous step, the remaining question is how we can pick the correct one from among the full form candidates. The criteria we used for selection include: how many times a candidate phrase appeared in the given corpus, whether it has been defined explicitly or not, and how similar its context is to that of abbreviation. Since these factors are not always independent from each other, they need to be considered at the same time when they are used as features in the experiment to be discussed later.

Since the task is to select the best full form among a set of candidates, it can be easily confused with a classification task. The following explains how this task is different from classification or categorization and why we need to rank the candidates instead. Fig. 4 shows a simple example of candidate ranking with four abbreviations and their corresponding candidates.

a) Different candidates among abbreviations: We can assume that an abbreviation in this corpus is likely to have a unique meaning. Even if an abbreviation is used in multiple documents, all the occurrences can be interpreted as having the same meaning because the whole corpus share the same domain and the same interest. Nonetheless, each abbreviation has its own full form candidatates. In classification, all instances (an abbreviations in our case) are determined to belong to one of the classes (candidates). However, different abbreviations have different class sets in our case. Since abbreviations do not share the same classes, in other words, it is clear that our task is different from the traditional classification work.

b) Varying candidates for an abbreviation: Another issue is that a set of candidates corresponding to an abbreviation is not fixed. If we have a complete and fixed candidate set, we might be able to gather examples of each candidate's usage by augmenting data source and utilize them as a training data. However, in the previous step, we collected many possible candidates to resolve the problem of absence of firm definition of an abbreviation. Since candidates are not fixed, we cannot generate complete training data which have examples for all the candidates. Since choosing an answer cannot be done with an existing classification model, we need to handle this issue as a ranking task, similar to ranking retrieved documents in a search engine. To rank the collected full form candidates, we adopt some features extracted from a given abbreviation and candidates and a ranking model developed in the information retrieval community.

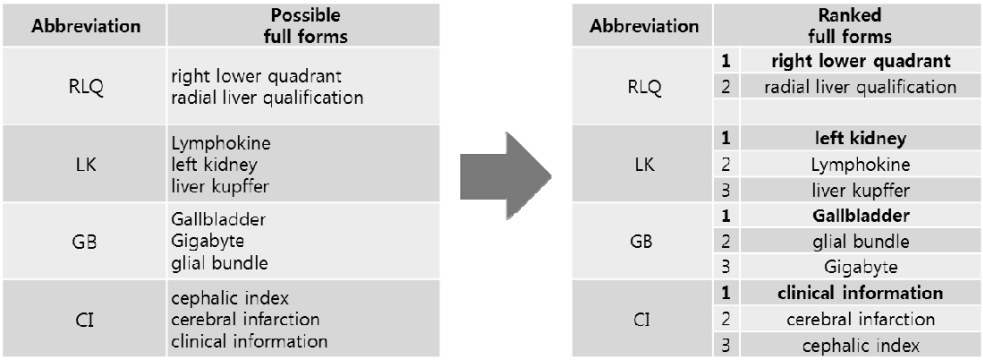

Figure 4. Brief Example of Candidate Ranking

2) Features

To decide which full form is most likely to be an answer among other candidates, one should observe the characteristics of the abbreviation, the answer, and the other candidates. Compared to queries and documents in web search, our abbreviations and full forms have limited information due to the short length; abbreviations have only a few letters (in the given corpus, no more than 6 letters) and the corresponding full forms only a few words. Considering this, we tried to exploit external resources as well as given data and came up with features as follows. Here we denote A_n to be an abbreviation and C_n^i, C_n^j to be A_n's i-th and j-th full form candidates, respectively.

a) Context similarity: In checking a match between A_n and C_n^i, a major cluse is how similar their contexts are. In clinical notes where A_n and C_n^i are used each, if the contexts near A_n and C_n^i share many words, one can assume that C_n^i can be replaced with A_n. Following the tradition of other text processing, we compute similarity of context vectors composed of 10 words around A_n and C_n^i respectively, using the cosine measure that is a commonly-used method for computing similarity in the vector space model [27].

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad (1)$$

Since this can be computed without optimization, a learning model would not be necessary if context similarity could rank all the candidates by itself. Considering that the given data type is clinical notes containing informal expressions, symbols and numbers, however, using context similarity alone is not enough for this task. Thus this feature is utilized in conjunction with the following features [30].

b) Repetition: If C_n^i appears more frequently than C_n^j in the given clinical note corpus or relevant medical literature corpus, we may consider that C_n^i is a more important expression in the domain and likely to form an abbreviation. To apply this criterion, we simply count the number of repetitions in both clinical and medical corpus separately.

c) Formality: Since part of full form candidates are retrieved by word sequence matching from the corpora, a candidate might contain words that are not so meaningful or suitable for medical expressions. Suppose that both C_n^i and C_n^j are composed of 4 words each. When C_n^i contains an article, a conjunction and two nouns and C_n^j consists of adjectives and nouns only, it is reasonable to conclude that C_n^j is more apt to be a medical expression and hence can be reduced to an abbreviation. To adopt this property, stop words in candidates are counted and added as one of features.

d) Definition: If the conditions of C_n^i and C_n^j such as the degree of repetition in the corpus and context are identical, we prefer to choose the one that has been formally defined in a dictionary or in a research article. However, it should be noted that a formal definition alone is not a confirmation to be the answer; an abbreviation may have multiple definitions according to its context. As a binary feature, it is tested whether a candidate has a definition in a standard dictionary or in a MEDLINE abstract (using Schwartz' algorithm).

3) Learning to Rank

We have shown the characteristics of the candidate selection task and features available for the task through the preceding subsections. Multiple criteria can be used to decide which a candidate is preferable to others, but still there is a question on how we determine the importance of each factor and dependency between them. If we have a model that can rank given candidates considering those features and can locate the most probable one at the top, we may use the model in this candidate selection task. The model is desired to be able to learn how a human decides the answer considering the given features.

In information retrieval and machine learning communities, a method called learning-to-rank has been introduced [19]. When ordering documents retrieved from a web search engine, instead of computing relevance score or authority of the document with fixed criteria, they wanted to make a supervised model that learns how human annotators rank the retrieved documents by considering the weights and dependency of given features. To help understanding adoption of a learning-to-rank model to our research, we recommend one to compare abbreviations and corresponding candidates with queries and retrieved documents, respectively.

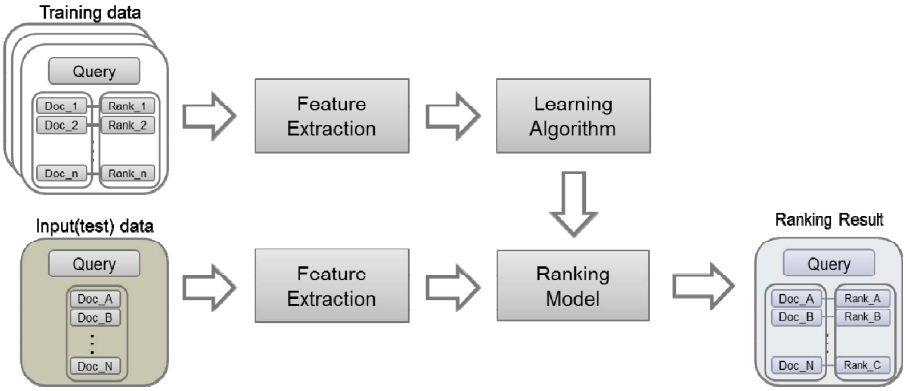

Figure 5. Overall Flow of Learning to Rank for Document Retrieval

Since introduced by Manning et al. [9], many studies have been done on learning-to-rank for information retrieval, and numerous models have been suggested [8]. The methods used in the research are categorized generally into three types – pointwise, pairwise and listwise approaches [2]. In pointwise approach, the task is considered as a regression problem assuming that an ideal function computes the score of each query-document pair exists. To utilize this approach, exact relevance scores are needed for all retrieved documents for training. In pairwise approach, the ranking problem is transformed to a binary classification task. The ranking model is to decide better (more relevant to given query) one between two documents. In order to train a model of the pairwise approach, either score-annotated documents set or a ranked list of documents can be used. In the list-wise approach, a ranked list itself is the object calculated. Rather than abstracting a given task by computing each document's score or judging between two documents, a model is demanded to optimize an ordered list's relevance rank directly. For this approach, a complete ranked list is needed as training data.

Getting back to our research, a major issue is which approach is most appropriate for our purpose and how we will generate training data. What we have for training data is abbreviation and original full form (answer) pairs and other 'not an answer but possible' full form candidates. Regarding full form candidates, it is hard to tell how much score each candidate should be given, except that an answer gets a perfect score. That means, the pointwise approach is not appropriate in this case. Besides, constructing a complete ordered list as training data is impossible because candidates may vary according to resources and exploration method, and they are just too many, for example 27 abbreviations among 110 in our data has more than 100 candidates, and 3 of them has more than 1000. Therefore listwise approach is also not suitable for our case.

However, we can certainly make candidate pairs that have decisions on which one is better, by coupling answer and candidates one by one. In this case, even though we cannot make use of all the possible combinations, still we can exploit part of combinations as training data for the pairwise approach. Experiments show that this approach is applicable.

D. *Training Data Generation*

As described in the previous subsection, the only information we have for training data is abbreviation-full form pair. Because of the limitation, when a ranked list from our data is used in training data during the learning process, it lacks of order information except that the answer is on the top. In other words, except for the answer, all the other candidates are considered to have same likelihood. However, even though they aren't answers, some of candidates have higher possibility to be full forms than the others, for example, for an abbreviation AP (arterial portography), 'abdominal pain' is more probable candidate than 'and partial.' It would be too costly to make some human annotator mark this kind of likelihood over whole candidate list and it would also be easy to be incorrect. For this reason, we wanted to make pseudo-rank over given candidates, exploiting suggested features. Since it is automatically computed (why it is called 'pseudo'), it isn't accurate enough to be used as training data for listwise approach, but we expect this pseudo-rank can reflect our intuition proposing the features. Pseudo-score for pseudo rank is proposed as described in the following equation with n the number of features.

$$Score^{i}_{Pseudo} = \frac{1}{n+1}\left(\sum_{j=1}^{n}\frac{Feature^{i}_{j}}{\underset{k}{Max}\,(Feature^{k}_{j})}\right) \quad (2)$$

Pseudo score of ith candidate can be computed by the formula. For each feature, we find the maximum value of the feature among all the candidates and take it as denominator. With the denominator, we can normalize the feature of each candidate by 0 to 1 scale. After that, we take average over all features. Here, we use $n+1$ instead of n as denominator in order to prevent non-answer candidate from getting full score as answer candidate.

V. EVALUATION

To evaluate validity and potential of the proposed approaches, we conducted experiments with given clinical

corpus and external medical academic resources available on the Web. Major goal of the trials is to check two factors: a) how well candidate exploration mines essential candidates and b) how well candidate ranking puts the answer on the top among full form candidates.

A. Data and Resources

1) Clinical note corpus

In this research, clinical note corpus plays two different roles; one is a source data which gives abbreviations and their contexts, and another is a source corpus for candidate exploration. For this task, we gathered total 2182 CT radiology reports of patients diagnosed as Hepatocellular Carcinoma (HCC) from Seoul National University Hospital. Because written by Korean clinicians, notes are composed of mixed language – Korean and English. Most professional terms are in English, but a few in Korean. Also, sentences are short and ill-formed as they are not written for academic purpose. We consulted a medical affairs recorder to collect abbreviations used in the corpus and their full forms. From this process, we got 110 unique English abbreviations.

2) Medical literature corpus

To supplement informality of clinical corpus, we adopted a part of MEDLINE as medical literature corpus. MEDLINE contains journal citations and medical articles' abstracts from many biomedical literature sources. Abstracts of all articles that belong to MEDLINE are accessible online, while some of them are even available with full text. Using PubMed, one can conduct search over entire MEDLINE abstracts database. As our clinical notes are regarding HCC patients only, to limit the scope of abstracts, we retrieved documents by putting 'hepatocellular' as a query in PubMed. Total 72225 documents are obtained, and their titles and abstracts construct our medical literature corpus. Besides, to extract definitions as full form candidates and as binary feature, we applied Schwartz's algorithm over this corpus.

3) Standard dictionary

For standard dictionary, we chose Unified Medical Language System (UMLS). There are three knowledge sources of UMLS – metathesaurus, semantic network, SPECIALIST lexicon and lexical tools. We acquired 2012 SPECIALIST lexicon and then extracted only entries with 'acronym_of' or 'abbreviation_of' relations. These relations are used as both explored candidates and features for ranking.

B. Tools and Evaluation Measures

1) Tools

As an important issue in both information retrieval and machine learning societies, a lot of learning to rank models are developed, and implementations of some of them are accessible online. Among many models, we chose RankBoost algorithm which follows pairwise approach [3]. RankBoost implementation included in RankLib, by Van Dang, was used in the experiments [26].

TABLE II. TYPES, EXTRACTION METHODS AND VALUES OF FEATURES FOR CANDIDATE RANKING

Feature	Method	Value
Context similarity	Cosine similarity	Float ($0.0 \leqslant x$)
Count in clinical corpus	Simple count by searching	Integer ($0 \leqslant x$)
Count in MEDLINE corpus	Simple count by searching	Integer ($0 \leqslant x$)
UMLS definition	Dictionary lookup	Binary($x = 0$ or 1)
MEDLINE definition	Pattern-based extraction	Binary($x = 0$ or 1)
Portion of stop words	Computation	Float ($0.0 \leqslant x \leqslant 1.0$)

2) Evaluation Measures

To show effectiveness of each module, we report results of candidate exploration and candidate ranking separately. For candidate exploration, we checked how many candidate sets contain the answers of corresponding abbreviations, and for candidate ranking, 'precision at 1 (prec@1)' is suggested; the percentage of abbreviations of which answers are put on the top after candidate ranking.

3) Features

We constructed a feature vector having 6 axes for each abbreviation-candidate pair. As the reason they are used were already explained in the methodology section, here we only show how those features are extracted and what form they have in order to help understanding, in table 2.

C. Results

1) Candidate Exploration

Results of candidate exploration are presented in Fig. 6. We tried candidate exploration with two kinds of data sources: a) independent sources b) combination of multiple resources. Left side of Fig. 6 contains the results of exploration using resources independently. It shows UMLS itself has the highest coverage among all four types of sources and clinical notes the lowest.

Right side of Fig. 6 presents the results of exploration using combination of the resources. In the figure, CC, ML(D), ML(S) mean clinical notes corpus, explicit definitions in medical literature corpus and words sequence in medical literature corpus, respectively. As we expected, the best result was found when the resources are exploited altogether. The coverage is greater than 85%. That indicates corpora such as clinical notes and medical literature can help improving exploration, when used with the standard dictionary. Another noticeable issue is that clinical corpus showed better result than Medline definitions when combined with UMLS, even thought it was worse when used independently. From this result, we can say that clinical corpus covers what UMLS can't more efficiently than Medline abbreviation definition dictionary does.

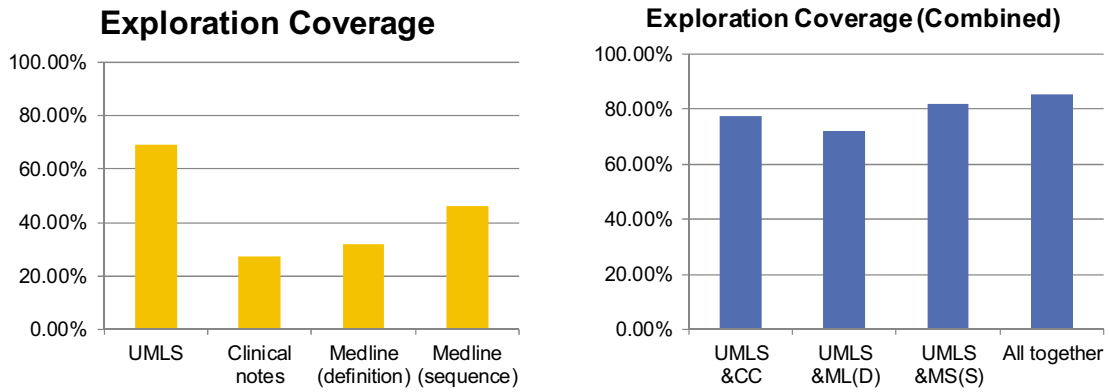

Figure 6. Graphs Comparing Results of Candidate Exploration

2) Candidate Exploration: Table 3 and 4 show the results of candidate ranking using learning to rank algorithm. Baseline has been constructed to evaluate how well our ranking method discover correct full form among the candidates. To make the baseline, we computed cosine-similarity score as introduced in section IV, without using any learning algorithm. L2R-SF mean learning to rank method with similarity(S) feature and formality(F) feature, L2R-SFD mean definitions(D) from UMLS and MEDLINE are added to SF features, and L2R-All mean all prepared features are used by adding repetition features from clinical notes corpus and MEDLINE.

Table 3 presents comparison of baseline and learning to rank results with various combinations of features. The best result was attained when all the features are used. When the features were insufficient that only similarity and formality features are used, learning to rank showed even worse result than the baseline. This supports the effectiveness of our feature set.

Table 4 was given to show effect of pseudo ranking method on training data. When the training data was ranked with pseudo scoring, a little improvement in performance was shown. Considering the pseudo scoring method was simple linear combination, we can expect that further improvement may be attained if more complicated and effective scoring method is adopted.

TABLE III. RESULTS OF CANDIDATE RANKING USING VARIOUS COMBINATIONS OF FEATURES

Method	Baseline*	L2R-SF*	L2R-SFD*	L2R-All*
prec@1	0.4149	0.3929	0.5357	**0.6857**

TABLE IV. RESULTS OF CANDIDATE RANKING USING DIFFERENT RANKING METHODS ON TRAINING DATA

Method	Baseline*	L2R-Binary*	L2R-Pseudo*
prec@1	0.4149	0.6673	**0.6857**

VI. DISCUSSION

A. Error Analysis and Limitation

First of all, the candidate exploration step has a weakness. Even though a large number of candidates are generated by the proposed exploration method, some unusual ones are left unfound. Major errors came from non-acronym abbreviations such as 'DDx' which is a reduced form of 'differential diagnosis.' Other issues came from the acronyms of which full form candidates are not detected from any of the resources; it is impossible to find the full form of an acronym by only looking at the documents. In this case, a consultation to the members of the institute is necessary.

About candidate ranking, there is a room for improvements in terms of accuracy because the current result was attained from imperfect training data for the learning-to-rank model. The rankings in the training data was not manually annotated. That is, we did not manually rank full form candidates list in the training data because of the cost.

B. Future Work

The inability to handle non-acronym abbreviations might be handled by applying more complicated candidate expansion methods. Park and Byrd, Wren and Garner, Okazaki and Ananiadou and many others have worked on the task that generates abbreviation from long phrases [12, 15, 22].

For the lack of order information in applying the learning-to-rank method, two different approaches can be considered: improving the pseudo ranking bu adopting more resources and other machine learning methods and some modification to the learning-to-rank model for this kind of partially ordered training data.

Okazaki et al.'s work on building sense inventory could also improve the candidate exploration and ranking steps if annotations for the candidates are also obtainable. It would help the exploration module concentrate on more important candidates and hence improving precision of the ranking module.

VII. CONCLUSION

In this paper, we investigated the issue of resolving abbreviations in clinical documents, showed impracticability of adopting traditional methods, and introduced a new perspective to solve this problem. Given clinical documents in a specific domain, the previous approaches of classifying an abbreviation into a fixed set of full form candidates or utilizing parenthetical expression patterns for automatic definition extractions are not applicable. The major reason is the lack of predefined full forms and parenthetical patterns in clinical documents. Instead, we treat abbreviation resolution as a search problem. We proposed a method of retrieving possible full form candidates from various resources and ranking them using a learning-to-rank method.

Both in exploration of possible full forms and in ranking them, we tried exploiting multiple resources both independently and in combination. The experimental result showed effectiveness of the proposed approach and identified a useful combination of features. Although there is more space to improve performance on both exploration and ranking, we claim that the proposed two-step approach is promising.

The contribution of this research is three-fold. First, the task of finding the correct meaning of an abbreviation in a clinical note is unique. Second, for the abbreviation resolution problem without specified full form candidates, we proposed a new perspective of treating it as a search problem rather than a classification one as done in the past. Third, we introduced a new way of utilizing learning-to-rank model outside the information retrieval domain.

ACKNOWLEDGMENT

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2009-0090853), and by the MSIP(Ministry of Science, ICT & Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2013-H0301-13-3005) supervised by the NIPA (National IT Industry Promotion Agency).

REFERENCES

[1] Aronson, A. and Lang, F. (2010), “An Overview of MetaMap: Historical Perspective and Recent Advances.” Journal of American Medical Informatics Association, 17(3), pp. 229-236

[2] Cao, et al. (2007), "Learning to Rank: from Pairwise Approach to Listwise Approach", Proceedings of the 24th International Conference on Machine Learning, pp. 129-136

[3] Freund, Y., et al. (2003). "An Efficient Boosting Algorithm for Combining Preferences", Journal of Machine Learning Research, 4, pp. 933-969

[4] Gaudan, S., Kirsch, H. and Rebholz-Schuhmann, D. (2005), “Resolving Abbreviations to Their Senses in Medline.” Bioinformatics, 21, pp. 3658-3664

[5] Jimeno-Yepes, A. and Aronson, A. (2010), “Knowledge-based Biomedical Word Sense Disambiguation: Comparison of Approaches.” Bioinformatics, pp. 565-574

[6] Joshi, M., et al. (2006). "A Comparative Study of Supervised Learning as Applied to Acronym Expansion in Clinical Reports." Proceedings of Annual AMIA Symposium, pp. 399-403

[7] Liu, H., et al. (2002). "A Study of Abbreviations in MEDLINE Abstracts." Proceedings of Annual AMIA Symposium, pp. 464-468

[8] Liu, T. (2009) “Learning to Rank for Information Retrieval.” Foundations and Trends in Information Retrieval, 3, pp. 225-331

[9] Manning C., Raghavan P. and Schutze H. (2008), Introduction to Information Retrieval, Cambridge University Press. Sections 7.4 and 15.5

[10] McInnes, B. (2009) “Supervised and Knowledge-based Methods for Disambiguating Terms in Biomedical Text using the UMLS and MetaMap.” Ph.D. Thesis, University of Minnesota, Minneapolis, MN, 234 pages.

[11] Okazaki, N. and Ananiadou, S., (2006). "A Term Recognition Approach to Acronym Recognition." Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pp. 643-650

[12] Okazaki, N., Ananiadou, S., and Tsujii, J. (2008), "A Discriminative Alignment Model for Abbreviation Recognition." Proceedings of the 22nd International Conference on Computational Linguistics, pp. 657-664

[13] Okazaki, N., Ananiadou, S., and Tsujii, J. (2010), “Building a High-quality Sense Inventory for Improved Abbreviation Disambiguation.” Bioinformatics, 26, pp. 1246-1253

[14] Pakhomov, S., et al. (2005). "Abbreviation and Acronym Disambiguation in Clinical Discourse." Proceedings of Annual AMIA Symposium, pp. 589-593

[15] Park, Y. and Byrd, R. (2001), "Hybrid Text Mining for Finding Abbreviations and Their Definitions." Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing, pp. 126-133

[16] Savova, G., et al. (2008), “Word Sense Disambiguation across Two Domains: Biomedical Literature and Clinical Notes.” Journal of Biomedical Informatics, 41, pp. 1088-1100

[17] Schwartz, A., and Hearst, M., (2003). "A Simple Algorithm for Identifying Abbreviation Definitions in Biomedical Text." Biocomputing, pp. 451-462

[18] Stevenson, M., et al. (2009), “Disambiguation of Biomedical Abbreviations.” Proceedings of the Workshop on BioNLP, Association for Computational Linguistics, pp. 71-79

[19] Trotman, A. (2005). “Learning to Rank.” Information Retrieval, 8(3), pp.359-381

[20] Tsuruoka, Y., et al. (2005). “A Machine Learning Approach to Acronym Generation.” Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases, pp. 25-31

[21] Vanopstal, K., Desmet, B. and Hoste, V. (2010), “Towards a Learning Approach for Abbreviation Detection and Resolution.” Proceedings of the 7th International Conference on Language Resources and Evaluation, pp. 1043-1049

[22] Wren, J. and Garner, H. (2002), “Heuristics for Identification of Acronym-definition Patterns within Text: Towards an Automated Construction of Comprehensive Acronym-definition Dictionaries.” Methods of Information in Medicine, pp. 426-434

[23] Wren, J.D., et al. (2005). “Biomedical term mapping databases.” Nucleic Acids Research (Database Issue), 33, pp.289-293

[24] Yu, H., et al. (2002). “Mapping Abbreviations to Full Forms in Biomedical Articles.” Journal of American Medical Informatics, 9, pp. 262-272

[25] Zhou, G., et al. (2005) “Recognition of Protein/Gene Names from Text Using an Ensemble of Classifiers.” Bioinformatics, 6(Supplement 1):S7

[26] Dang, V., Rank Lib. http://people.cs.umass.edu/~vdang/ranklib.html 2012.

[27] Lee, D.L., Chuang, H., and Seamons, K. (1997), "Document ranking and the vector-space model." Software, IEEE 14.2 pp. 67-75.

[28] UMLS Knowledge Sources, 2012 Edition.

[29] MEDLINE. http://www.nlm.nih.gov. 2012.

[30] “Cosine similarity” in Wikipedia, The Free Encyclopedia. 2013.