An Improved Method for Extracting Acronym-Definition Pairs From Biomedical Literature

Saneesh Mohammed N
Department of Computer Science and Engineering,
Heera College of Engineering and Technology ,
Trivandrum, India - 695568
Email: saneeshmohammed88@gmail.com

K A Abdul Nazeer
Department of Computer Science and Engineering,
National Institute of Technology Calicut,
Calicut, India - 673601
Email: nazeer@nitc.ac.in

***Abstract*—This paper deals with the problem of extracting acronym-definition pairs from biomedical text. We propose an improved Text mining system based on pattern matching method and space reduction heuristics which increases both recall and precision. Three metrics were used for evaluating the system – *recall* (measure of how much relevant data the system has extracted from text), *precision* (measure of how much data returned by the system is actually correct) and *f-factor* (combined value of recall and precision). Experimental results achieved 98.68% recall and 98.68% precision.**

I. Introduction

Text Mining and Information retrieval methods help researchers to extract useful pieces of data from large volumes of textual data. In recent times, there is a rapid increase in the number of articles being published in the area of Bioinformatics. Such articles generally include a good number of acronyms. Specialized databases are necessary to store acronyms and their definitions. Acronym-definition pairs are to be identified and extracted from the literature for storing in such databases. Due to the high rate at which new acronyms are introduced in the field of Bioinformatics, existing databases of acronyms and their definitions are far from complete. This points to the need for an efficient method for mining acronym definition pairs from Bioinformatics literature domain.

Acronyms are generally created from the first letters of all the words in the definition, such as ISRO (Indian Space Research Organisation) and NIT (National Institute of Technology). But in Bioinformatics there exist no standardized method for creating the acronyms from their definitions. This poses a big challenge to the process of mining acronyms and their definitions from the articles in the biomedical literature. Letters in the acronym do not always match the initial words in the definition. See Table I for example acronyms from biomedical literature

Several methods were proposed for mining acronym-definition pairs which include pattern matching method, statistical and machine learning method. Some of the works did not consider Biomedical domain which contains many variations in the acronyms and definitions. Pattern matching methods generally do not consider some of the space reduction heuristic constraints.

This paper is an improvement over Rafeeque et.al[2] approach which uses pattern matching method[4] along with space reduction heuristics[3] to extract acronym-definition pairs from textual data. This improved approach will increase the recall and precision. The method described in[2] does not consider acronyms which contain digits and acronym-definition pairs which appear in multiple lines. The metrics used for evaluating the performance are Recall (measure of how much relevant information the system has extracted from text), Precision (measure of how much information returned by the system is actually correct) and F-factor (combined value of recall and precision)[2].

II. Related Work

Schwartz et.al[4] used a pattern matching technique for extracting acronyms (short forms) together with their definitions (long forms) from biomedical text documents. In this algorithm, acronyms are considered valid candidates for extraction only if they consist of atmost two words, their length is between two to ten characters, atleast one of these characters is a letter and first character is alphanumeric. The algorithm fails to identify acronym-definition pairs if there is no exact character match in the definition string, e.g. Asf1 (anti-silencing function) and if characters do not appear in the same order as in the definition string. Algorithm fails to extract definitions inside parenthesis with one or two tokens(words) and definitions containig comma character.

Nadeau et.al[3] proposed a supervised learning approach to the acronym identification task. The usage of weak heuristics resulted in the identification of a large number of acronym-definition pairs. However there are some acronym-definition pairs that the algorithm fails to identify, such as cyclophilin seven suppressor (CNS1) and 3-N-maleimidyl-propionyl biocytin (MPB).

Rafeeque et.al[2] present a combination of pattern matching and space reduction heuristics to extract acronym-definition pairs of the form acronym(definition) or definition(acronym).

978-1-4799-0575-1/13/$31.00 ©2013 IEEE

TABLE I
EXAMPLES FROM MEDLINE

Acronym	Definition
BG	Beta-1, 4-glucosidase
myc-glu-glu	mycosporine-glutaminol-glucoside
MPB	3-N-maleimidyl-propionyl-biocytin
T3S	type III secretion
Asf1	anti-silencing function

However there are some acronym-definition pairs that the algorithm fails to identify, such as cyclophilin seven suppressor (CNS1) and 5GL (Fifth Generation Language).

Yeates [5] algorithm uses general heuristics to match characters of the candidate acronym with letters in the definition string. It does not recognize abbreviated expressions containing more than one upper-case letter from the acronym (e.g., DataBase Management System(DBMS)).

III. SPACE REDUCTION HEURISTICS

Algorithms are developed based on the constraints used for the identification of acronym-definition pairs. Space reduction constraints [3] are used for extracting acronym-definition pairs. These constraints help to reduce the search space considerably. At the same time, these constraints are weak enough to select almost all positive cases. A pattern matching method is designed considering these constraints.

A. For Candidate Acronyms

The acronym space (the set of choices for ACR = tok_i) is reduced using syntatic constraints on the tokens, T = tok_1,...,tok_n expressed by the conjuction of the following statements:

- ACR = tok_i , where $1 \leq i \leq n$.
- $Size(tok_i) \geq 2$, where Size(tok_i) is the number of characters in the token tok_i (including numbers and internal punctuation). But the maximum number of alphanumeric characters is 10.
- Maximum number of tokens in the acronym is 2
- $NumLetter(tok_i) \geq 1$ where NumLetter(tok_i) is the number of alphabetic letters in the token tok_i.
- The token should not be a conjuction, determiner, particle or preposition.

B. For Candidate Definitions

- The pattern must be of the form acronym (definition) or definition(acronym) [4]
- The first word of a definition must use the first letter of the acronym unless it is a digit. But first word of the definition can be a word preceded by a character which is neither a letter nor a digit.
- The definition can skip any number of punctuation characters inside the acronym.
- The maximum length for a definition is min(Acronymlength+3, Acronymlength$*$2). (definition length is the number of words in the definition and acronym length is the number of characters present in the acronym).
- A definition cannot contain a colon, semicolon, question mark or exclamation mark. It can contain comma, hyphen or forward slash .

IV. PROPOSED DESIGN

The proposed text mining system is divided into four stages- Extract Candidates, Extracting the Correct Subset of words, Checking the validity of acronyms and Checking the validity of definitions.

A. Extract Candidates

This stage is for extracting acronym-definition pairs. It is based on the constraint that the pattern of acronym-definition pair should be of the form acronym (definition) or definition (acronym). This method is an improvement over the corresponding phase in[2] which extract only acronym-definition pairs which appear in the same line. Our method extracts acronym-definition pairs even if they appear in different lines which gives a way to extract all acronym-definition pairs from textual data resulting in increased recall and precision. Main steps in this stage are as follows :

- Identify and extract the pattern acronym (definition) or definition (acronym) from textual data .
- Resolve the ambiguity to confirm whether acronym is inside or outside the paranthesis.

B. Extracting the Correct Subset of Words - The Improved Method

This method is an improvement over the corresponding phase described in[2]. The aim of this phase is to find the shortest definition that matches the acronym which is achieved by scanning from the end of both acronym and definition to the left. Every character in the acronym should have a match in the definition and the matched character in the definition must be in the same order as the characters in the acronym. Character at the beginning of the acronym should match the initial character of the first word in the definition unless it is a digit. Thus first letter of acronym can be the first letter of a word that is connected to other words by hyphen and the other nonnumeric characters. If the acronym character is a digit, the token in the definition is matched with a set containing all the possible definitions for that digit. The algorithm for the above function is given in Algorithm 1.

C. Validity of Acronyms

This stage is for checking the validity of acronyms based on some constraints. According to the constraints, the number of alphanumeric characters in the acronym is limited to 10

and it should contain atleast one alphabetic character. Then check whether the acronym is conjunction, determiner, particle or preposition. A valid acronym does not belong to any of these classes. The algorithm for the above function is given in Algorithm 2.

D. Validity of Definitions

This stage is for checking the validity of definitions based on some constraints. These constraints include the limitation on the maximum length of the definition and the characters which are to be excluded from the definition. For this purpose, the candidate definitions need to be tokenized to get the number of tokens in the definition. If the number of tokens in a candidate definition is greater than (acronymlength $*$ 2) or (acronymlength + 3) then that definition may be discarded. The algorithm for the above function is given in Algorithm 3.

V. Implementation and Results

The text mining system is designed in such a way that multiple text documents can be read as input. Our system uses Java programming language as Front end and SQL database as Back end.

A. Evaluation measures

The gold standard allows evaluation of different algorithms according to three metrics - Recall, Precision and F-factor. These are considered as the standard evaluation metrics for comparing alternative techniques in the field of NLP[3]

Recall is a measure of how much relevant information the system has extracted from the text.

Recall (R) = $\frac{NumberofTADPR}{TotalNumberofTADPF}$

Precision, also known as accuracy, is a measure of how much information returned by the system is actually correct.

Precision (P) = $\frac{NumberofTADPR}{TotalNumberofEADP}$

where TADPR is True Acronym-Definition Pairs Returned, TADPF is True Acronym-Definition Pairs in File and EADP is Extracted Acronym-Definition Pairs

F-factor is a combined value of recall and precision. When recall and precision are given equal weight, F-factor will be as follows:

F-factor(F) = $\frac{2RP}{R+P}$

Algorithm 1: Extracting Correct Subset of Words - The Improved method

```
begin
    Set ACIndex at the end of ACR ;
    Set DEFIndex at the end of DEF ;
    while (ACRIndex > 0) do
        currchar = char(ACIndex);
        if (not letterordigitorhyphen(currchar)) then
            return null;
        end
        if (Digit(currchar)) then
            while (DEFIndex > 0 and
            char(DEFIndex)<>space) do
                DEFIndex = DEFIndex − 1;
            end
            ENDIndex=DEFIndex;
            while (DEFIndex > 0 and
            char(DEFIndex)<>space) do
                DEFIndex = DEFIndex − 1;
            end
            Str=ExtractString(DEFIndex, ENDIndex);
            DEFIndex = DEFIndex -1;
            if (not Str appear in the definition set of
            currchar) then
                return null ;
            end
        else if (Letter(currchar)) then
            while ((DEFIndex>0) and
            currchar<>char(DEFIndex)) do
                DEFIndex = DEFIndex − 1;
            end
        if (DEFIndex<1) then
            return null;
        end
        ACIndex = ACIndex − 1;
        DEFIndex = DEFIndex − 1;
    end
    DEFIndex = Index of space from DEFIndex + 1;
    DEFNew = Left over characters from DEFIndex;
    return DEFNew;
end
```

Algorithm 2: Checking the validity of Acronym

```
IsValidAcronym(ACR) begin
    Numchar = Number of alphanumeric characters in
    the ACR;
    Numletter = Number of letters in the ACR;
    if (Numchar>10) then
        return null;
    end
    if ((Numletter >= 1) and NotConjunction(ACR) and
    Notdeterminer(ACR) and NotPreposition(ACR) and
    NotParticle(ACR)) then
        return true;
    end
end
```

Algorithm 3: Checking the validity of Definition

```
IsValidDef(DEFNew) begin
    Numtokens = Number of tokens in DEFNEW;
    ACRSize = length(ACR);
    if (Numtokens > ACRSize×2) or (Numtokens>
    ACRSize + 5) then
        return null;
    end
    if [';', ': ', '!', '?']in DEFNew then
        return null;
    end
    else
        return DEFNew;
    end
end
```

TABLE II
EXPERIMENTAL RESULTS

Algorithm	Metrics	Values
Rafeeque et.al[2] approach	Number of True Pairs	152
	Total Extracted	144
	True Positives	140
	Recall	92.11%
	Precision	97.22%
	F-factor	94.6%
Our Improved approach	Number of True Pairs	152
	Total Extracted	152
	True Positives	150
	Recall	98.68%
	Precision	98.68%
	F-factor	98.68%

B. Experimental Results

For evaluating the performance of the system, abstracts of **Yeast** and **Eukaryote** are selected. From these abstracts all true acronym-definition pairs are selected by manually annotating for testing. Then the text mining system is executed by using these abstracts.

C. Discussion

Rafeeque et.al[2] combined pattern matching along with space reduction heuristics for extracting acronym-definition pairs. But their approach did not extract acronyms which contain digits. Their approach extract acronym-definition pairs only if they are present in the same line. Our approach extracts acronym-definition pairs which contain digits as well as those appearing in multiple lines. Recall of their system is 92% and precision is 97.2%. Our approach has resulted in 98.68% recall and 98.68% precision which is not yet achieved by any of the existing approaches. See Table II for experimental results.

VI. CONCLUSION

An improved method for extracting acronym-definition pair from bioinformatics literature is proposed in this paper. The method is based on pattern matching approach along with space reduction heuristic constraints to reduce the search space of acronyms. The evaluation is done by taking the MEDLINE abstracts and achieved better recall, precision and f-factor compared to existing approaches. It is observed that many acronym-definition pairs which were not identified by the existing approaches are getting identified and extracted by our approach. The usage of space reduction heuristic constraints reduces the search space and extracts most of the true positive cases. Since a pattern matching method is employed, the mining process becomes simple as it does not require large sets of training data to run.

A major limitation of the system is that it imposes a restriction on the format of candidate acronym-definition pairs. They should appear either as acronym (definition) or definition (acronym). Overriding this limitation may be attempted as future work.

REFERENCES

[1] Medline Abstracts. http://www.ncbi.nlm.nih.gov.

[2] Rafeeque P. C. and Abdul Nazeer K. A. Text mining for finding acronym-definition pairs from biomedical text using pattern matching method with space reduction heuristics,. In *Proceedings of the 15th International Conference on Advanced Computing and Communications, IIT Guwahati, India, IEEE Computer Society*, pages 295–300, 2007.

[3] D Nadeau and P Turney. A supervised learning approach to acronym identification. In *Proceedings of 18th Conference of the Canadian Society for Computational Studies of Intelligence Victoria, BC, Canada*, pages 319–329, 2005.

[4] A S Schwartz and M A Hearst. A simple algorithm for identifying abbreviation definitions in biomedical texts. In *Proceedings of the Pacific Symposium on BioComputing (PSB) University of California, Berkeley*, 2003.

[5] S. Yeates. Automatic extraction of acronyms from text,. In *Proceedings of the Newzealand Computer Science Research students conference, University of Waikato, Hamilton, newZealand*, 1999.