Designing of an efficient algorithm for identifying Abbreviation definitions in biomedical Text

Shashank Singh[1], Gaurav Sharma[2] Abdul Nazeer K A[3]& *Shalini Singh[2]

[1]IBM India Private limited, Bangalore , [2]School of Biochemical Engineering IIT (BHU), Varanasi-221005 ,[3]NIT Calicut Kerala.
shalini.bioinfo@gmail.com

ABSTRACT

The size and growth rate of biomedical literature creates new challenges for researchers who need to keep up to date. The objective of the present study was to design a pattern matching method for mining acronyms and their definitions from biomedical text by considering the space reduction heuristic constraints have been proposed and implemented. The constraints mentioned are spacious-reduction heuristic constraints which will reduce the search space and will extract most of the true positive cases. The evaluation has been done on MEDLINE abstracts. The results show that the proposed algorithm is faster and more efficient than the previous approaches, in term of space and time complexities. The algorithm has a very good Recall (92%), Precision (97%) and F-factor (94%). One improvement that can be done is to consider all kinds of acronyms definition patterns. This algorithm only considers acronym−definition pairs of the form Acronym (Definition) Definition (Acronym) pairs. Improving the algorithm requires additional study and may reduce the precision even though it may increase the recall. The Algorithm is space efficient too. Input text of any large size can be mined using this algorithm because it requires less memory space to execute.

KEY WORDS

biomedical text ,MEDLINE, Recall, Precision, F-factor, Acronym

1. Introduction

Abbreviations are widely used in biomedical text. The size and growth rate of biomedical literature creates new challenges for researchers who need to keep up to date. One specific issue is the high rate at which new acronyms are introduced in biomedical texts. First, available database for acronyms and their definition for biomedical domain is incomplete. For example, running over 40,956 abstracts (one month of MEDLINE database) resulted in 9272 unique acronyms which were not identified in the existing databases. Existing databases, ontologies, and dictionaries must be continually updated with new acronyms and their definitions. Acronyms normally appear with common constructions such as NASA (National Aero- nautics and Space Administration). However there exist acronyms which differ from this structure and do not follow the above mentioned definition. Automatic Extraction of Acronyms and their definitions from biomedical domain is difficult as there is wide variance in conventions within biomedical communities on forming acronyms from their definition (long form). In an attempt to help

resolve the problem, new techniques have been introduced to automatically extract abbreviations and their definitions from MEDLINE abstracts [1,2,3].

2. Materilas and methods

2.1 Input data

For the evaluation of the proposed algorithm four abstracts–test1.txt ,test2.txt, data1.txt, data2.txt have been taken from MEDLINE[4].From these abstracts all true acronym–defi nition pairs were selected by manually annotating for testing.

2.2 Design and implementation

Algorithms are designed based on the heuristic constraints used for the identifiation of acronym-definition pairs. The order and strictness with which these heuristics are applied is a matter of tuning, which is context dependant. Space reduction heuristics for candidate acronyms. The acronym space (the set of choices for $A = t_i$) is reduced using syntactic constraints on the tokens, $T = t_1, ..., t_n$, expressed by the conjunction of the following statements:

- $A = t_i$, where $1 \leq i \leq n$.
- $Size(t_i) \geq 2$, where $Size(t_i)$ is the number of characters in the token ti (including numbers and internal punctuation). But the maximum number of alphanumeric characters is 10.
- Maximum number of tokens in the acronym is two.
- $NumLetter(t_i) \geq 1$, where $NumLetter(t_i)$ is the number of alphabetic letters in the token t_i.

2.3 Space reduction heuristics for candidate definitions

- Both acronym and definition must appear in the same sentence and the pattern will be of the form acronym(definition) or definition(acronym) [5].
- The first word of a definition must use the first letter of the acronym [6]. But first word of the definition can be a word preceded by character which is neither letter nor digit.
- The definition can skip any number of digits and punctuation characters inside the acronym.
- The maximum length for a definition is min(Acronymlength + 5, Acronymlength × 2) [3].
- A definition cannot contain a colon, semi-colon, question mark, nor exclamation mark. It can have comma.

2.4 Differerent phases of design

The proposed algorithm has been divided in two steps.Its modified version of Nazeer & Rafeeque [7] approach. It consist of only two steps . There is no need of further refinement as the second step itself only considers the true pairs. This makes this algorithm much faster and space effecient. The phases are explained though the Flow chart in Figure 1.

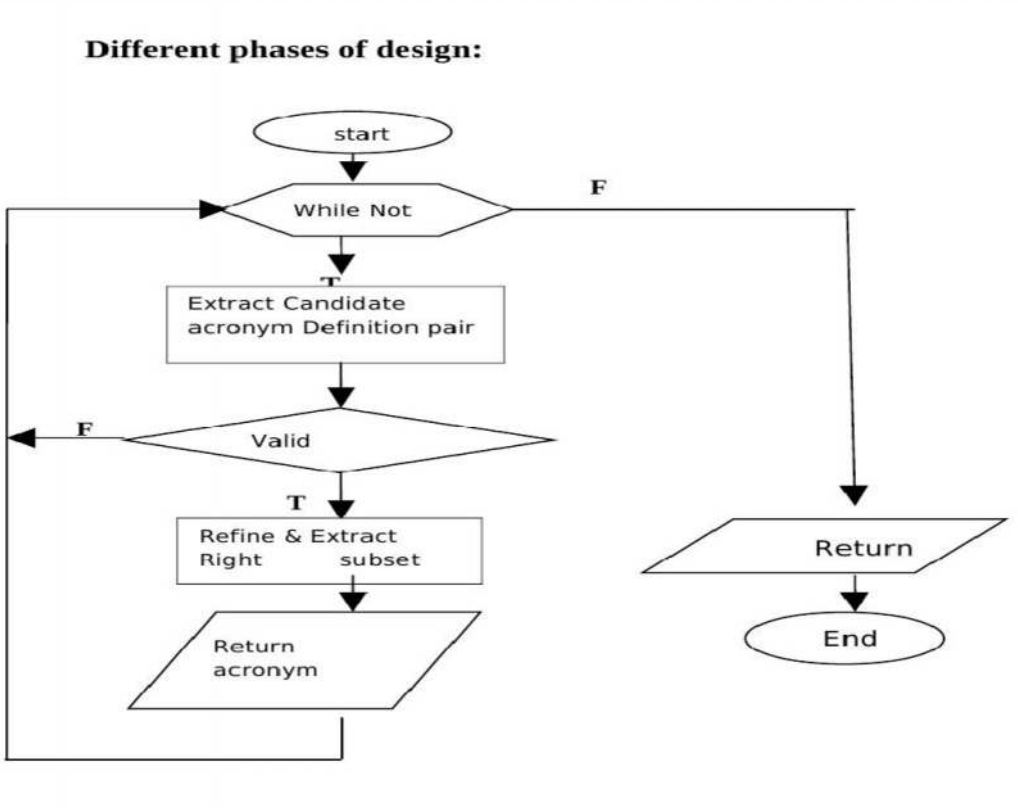

Figure 1: Flow chart of the Different phase of design

2.5 Extract Candidate Acronym Definition Pairs

This stage is for extracting candidate acronym-definition pairs. It is based on the strongest constraint in the pattern that both acronym and definition must appear in the same sentence and acronym-definition pattern will be of the form acronym (definition) or definition (acronym). This is a variation of Schwartz's [6] approach. Main steps in this stage are as follows:

• Identify the pattern acronym (definition) or defiition (acronym) from the current sen- tence.
• Resolving the ambiguity to confirm whether the acronym is inside or outside parenthesis. Definition inside parenthesis with one or two words (token) also to be considered.
• Resolve the problem in which the acronym itself contains parenthesis. Pseudocode has been submitted as (supplymentry material).

2.6 Extract True subset of words

Extracting the candidate acronym and definition will give a list of candidate definition words corresponding to acronym. So the next task is to choose the true subset of words. For this we used an algorithm similar to Schwartz [6]. The main idea is to find the shortest definition that matches the acronym by scanning from the end of both acronym and definition to the left. Every character in the acronym should have a match in the defnition and matched character in the definition must be in the same order as characters in the acronym. One important restriction is

that match of the character at the beginning of the acronym must match a character in the initial position of the leftmost (first) word in the definition. This initial position can be the first letter of a word that is connected to other words by hyphen and other nonnumeric characters.

The algorithm count the number of words in the definition and if number of words > min(Acronymlength+5, Acronymlength × 2) it will discard those candidates here itself. Other constraints are also checked , like length of acronym must be less then 10, so that after this steps only true pairs are considered and there is no need of further refinement. This is an improvement from the previous approach [7]. The Pseudocode for the algorithm has been submitted as supplymentory material.

2.7 Implementation

The algorithm has been implemented in Java to show the performance improvement. The user can input a text file which contains the biomedical data and the output is produced in the specified file. The output file contains the extracted acronym definition pair, the execution time in millisecond and the Java virtual machine free memory in bytes. The execution time and free memory space will be used to analyze the performance for different text input.

3. RESULTS & DISCUSSION

3.1 Caculation of Recall, Precision & F-factor

The proposed algorithm uses space reduction heuristic constraints similar to Nazeer & Rafeeque[7] which will reduce the number of candidate definitions. As in this approach no constraint has been relaxed and it has got the same recall and precision. For calculating the recall and precision Test1.text and test2.text have been used. The results of the experiment have been shown in Figure 2.1. Recall and precision can be calculated from the following formula.

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

$$Recall\ (R)=\frac{No\ of\ TPR}{Total\ no\ TPF}$$

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

$$Precision = \frac{Total\ no\ of\ TPR}{Total\ no\ of\ EP}$$

Where TPR is the true pairs retuned and TPF is the true pairs in file and EF is the extracted pairs. **F**-factor is a combined value of recall and precision.when precision and recall given equal weight, F-factor will be as follows

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

The proposed algorithm have a very good value of Recall(92%),Precision(97%) and F-factor(94%). Which depict the superiority of this algorithm over previously defined algorithms.

3.2 Calculation of Execution Time

The proposed algorithm is faster then the algorithm given by Nazeer & Rafeeque [7]. Better time complexity has been archived by restricting the search of candidate acronyms and merging two phases in one. Test1.txt, Test2.txt, data1.txt and data2.txt are the data files used as input files to show the execution time results. RafNaz1 and RafNaz2 are the two different implementation of previous approach[7] and Newalgo is the proposed modified algorithm . The results are summarized in figure 2.2.

3.3 Calculation of Space Requirement

In the implementation of the proposed algorithm, the free space in java virtual machine also has been calculated. The results show that it is much more space efficient than Nazeer & Rafeeque's algorithm[7]. test1.txt and test2.txt have been used in a graphical analysis to show that the proposed algorithm take the less space than the other two versions of previous approach[7].The results are summarized in figure 2.3.

The proposed algorithm used the space reduction heuristic constraints by Nadeau D and P Turney [8]. But they used supervised learning method. The most serious problem with supervised learning systems is that a large training set is essential for good performance. Building a training set by human labeling is time consuming, labour intensive and expensive [9].

3.4 Calculation of Time Complexity

The ExtractTrueSubset in the proposed algorithm is actually a modified version of Extract RightSubset function of Nazeer & Rafeeque's approach[7]. The ExtractTrueSubet search for acronym definition pair. It return only true pairs so any kind of refinement is not needed for the returned pairs. ExtractTruesubset is actually a advanced version of ExtractRightSubset with the additional functionality of refinement. The proposed algorithm has a better time complexity then the previous approach [7].

In Nazeer and Rafeeque's approach the while loop in ExtractRightSubset t times where t can be defined as number of characters in candidate definition. Let the length of the candidate definition be tc *k , where tc is the no of tokens in the candidate definition and k is the average length of a word. The time complexity of ExtractRightSubset is O(tc.k). But tc can be any large number. There is no upper bound for tc. In the further refinement step we are rejecting those candidates for which tc > min(Acronymlength+5, Acronymlength×2). It means that ExtractRightSubset is returning or can return some unwanted candidates, which need to be refined further spendingO(tc.k) time. On the other hand in ExtractTrueSubset(), the while loop is executing <= n times where n=K×min(Acronymlength+5, Acronymlength ×2). Thus the time complexity reduced to O(n.k) .Since the unwanted candidates are rejected here itself , no further refinement is required which eliminate further phase of refinement.

3.5 Space Requirement

The proposed algorithm is much more space efficient than Nazeer and Rafeeque's algorithm [7]. This algorithm uses only two phases to find the acronym definition pairs. So there is no need of

saving the extracted pair after phase two for further refinement which is the case in previous approach. To show the results again test1.txt, test2.txt, data1.txt, data2.txt have been used. Nazeer & Rafeeque's algorithm has been implemented in two ways, one which is buffering the pairs after phase two on RAM and other which is saving on the disk. First one (RafNaz1) is taking slightly more time than proposed algorithm but consumed a lot of space. The other one (RafNaz1) take almost the same space but take more time for the execution. It shows that the proposed algorithm give better performance in both ways, space and time, than the previous approach. The results are summarized in figure 2.1, 2.2 and 2.3.

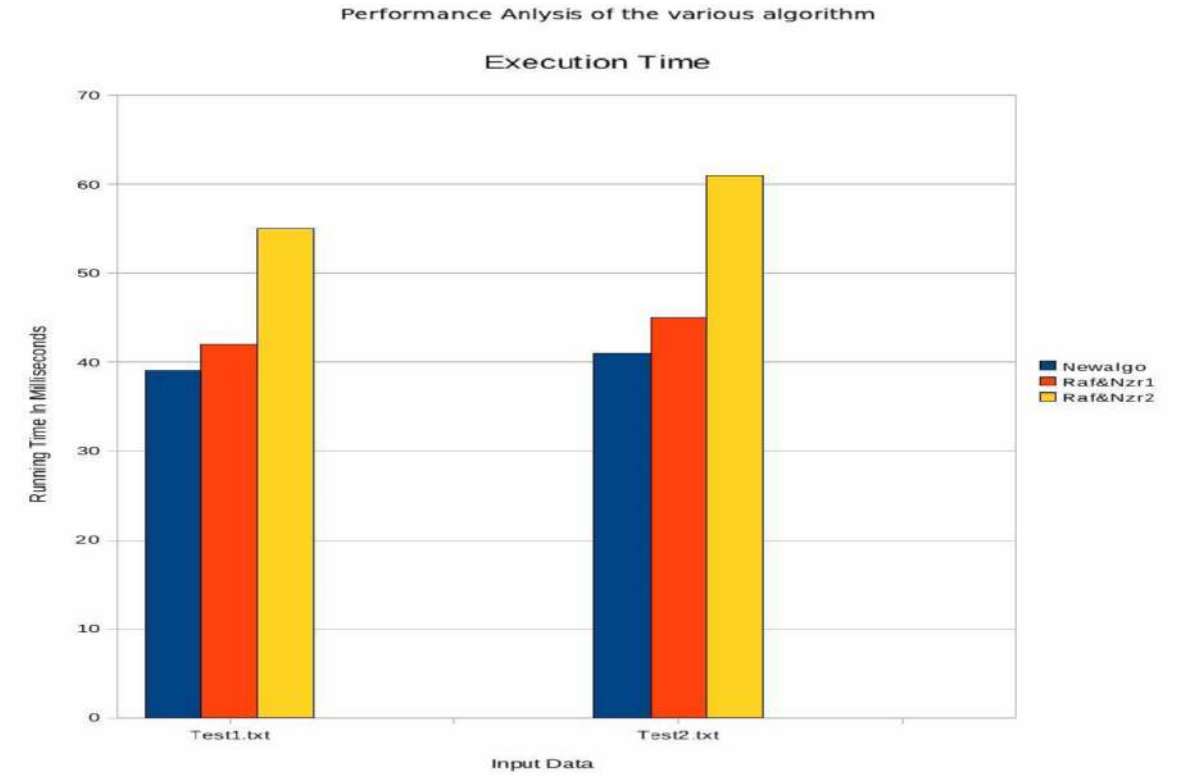

Figure 2.1 : Performance evaluation of Newalgo with previously defined algorithm

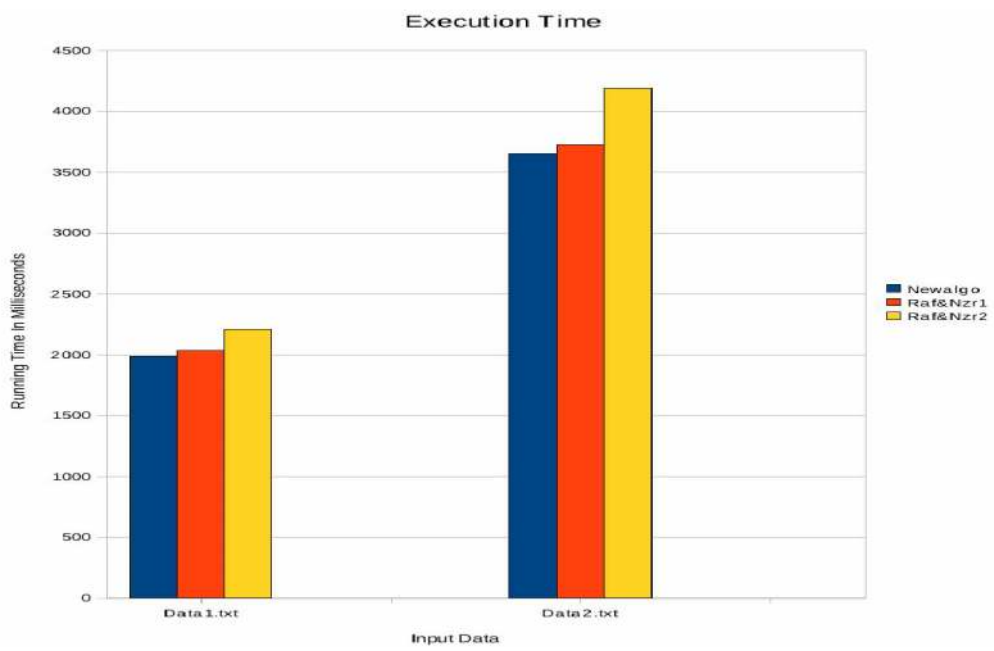

Figure 2.2 : Comparative study of execution time of Newalgo with previously defined algorithm

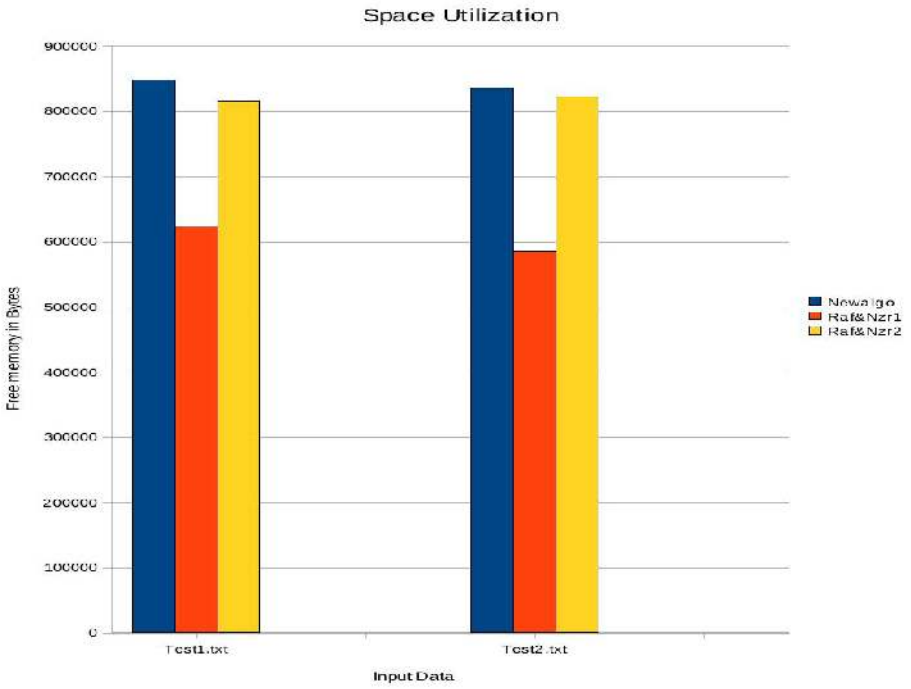

Figure 2.3 : comparative study of space utilization of new algo with other previously defined algorithm

4. Conclusions

In this present work a pattern matching method for mining acronyms and their definitions from biomedical text by considering the space reduction heuristic constraints [9] has been proposed and implemented. The constraints mentioned are space-reduction heuristic constraints which will reduce the search space and will extract most of the true positive cases. The evaluation has been done on MEDLINE abstracts. The results show that the proposed algorithm is faster and more effecient than the previous approaches, in term of space and time complexities. The algorithm have a very good Recall(92%),Precision(97%) and F-factor(94%). One improvement that can be done is to consider all kind of acronym–definition patterns. This algorithm only considers acronym–definition pairs of the form Acronym (Definition) Defiinition (Acronym) pairs. Improving the algorithm requires additional study and may reduce the precision even though it may increase the recall.

References

[1] Andrade , M.A.,et al, (1997), "Automatic annotation for biological sequences by extraction of keywords from MEDLINE abstracts. Development of a prototype system." ISMB . 5: pp 25-32.

[2] Pustejovsky , J. , et al, (2001), "Automation Extraction of Acronym-Meaning Pairs from Medline Databases" Medinfo . 10: pp 371-375.

[3] Rindflesch, T.C., et al, (2000) "Extracting Molecular Binding Relationships from Biomedical Text" Proceeding of the ANLP-NAACL 2000, pp 188-195

[4] Medline abstracts. "http://www.ncbi.nlm.nih.gov."

[5] Pustejovsky, J., Castao, J., Cochran, B., Kotecki, M., Morrell., and Rumshisky, A., "Extraction and Disambiguation of Acronym-meaning pairs in medline"., unpublished manuscript (2001)

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

[7] Abdul Nazeer, K. A., Rafeeque, P. C., (2007).:"Text Mining for Finding Acronym-Definition Pairsfrom Biomedical Text using Pattern Matching Method with Space Reduction Heuristics",15 The International Conference on Advanced Computing and Communications,. ADCOM 2007

[8] Nadeau, D., and Turney, P., (2005) "A supervised learning approach to acronym identification",In Proceedings of 18th Conference of the Canadian Society for Computational Studies of Intelligence Victoria, BC, Canada, pp 319-329

[9] Yeates, S., (1999). "Automatic extraction of acronyms from text". In Proceedings of the Newzealand Computer Science Research students conference, University of Waikato, Hamilton, newZealand, 117-124

Authors

Shashank Singh

Complited his MCA form NIT Calicut kerala from year 2008.After that he had joined the IBM private limetd Bangalore as developer. Resently he is working in same company as senior software developer.he is awarded by so many company awards, like IBM Global service excellence,IBM hall of the fames during these periods.

Shalini Singh

Completed her masters in bioinformatics from BHU in year 2008. Varanasi, after that she joined IIT (BHU) as research student in year 2009. Her area of research is designing the novel methods for genomic analysis as well as the computational drug designing.