DEVELOPMENT OF ALGORITHM FOR EXTRACTING ACRONYM DEFINITIONS FROM GURMUKHI TEXT

Manpreet Kaur*, Jagroop Kaur**
*(Computer Science Department, UCOE, Punjabi University, Patiala,India)
** (Asst. Prof. Department of Computer Science, UCOE, Punjabi University, Patiala,India)
* manpreetaulakh65@gmail.com
** jagroop_80@rediffmail.com

ABSTRACT

In the recent years, an exponential increase of using new linguistic feature Acronyms in the available electronic information documents to produce shorter text causes a big necessity to quickly understand large volumes of data. In the recent years, Automatic Recognition of Acronym-Definition Pairs has gained a One challenge derives from the common and uncontrolled use of acronyms in the medical and political field. In this paper we have developed an Algorithm for Extracting Acronym Definitions from Gurmukhi Text by using First-Letter matching technique. For improving the recall we have provided Common Acronym-Definition Pairs database. We tested our system over two types of data- Medical and Political data. We have applied three standard measures 1) Recall 2) precision 3) F-Score. The proposed system achieved 86% recall, 86.04% precision, yielding F-score of 86.01% for Medical document and 86.53% recall, 95.55% precision, yielding F-score of 90.81% for Political document by using Database of common Acronym-Definition pairs.

Keywords— *Acronyms, Acronym identification, Abbreviation, Tokenization, normalization, Recall, Precision, Medline*

I. INTRODUCTION

People will find it helpful if we develop a system that can automatically recognize acronyms and their expansions from Text. This is because there are many online large documents in Punjabi language related to various domains like political, medical which contain many acronyms[13]. In many cases, however, acronyms occur frequently enough to make it difficult for outsiders to comprehend text. The documents under consideration are typically large and contain many domain-specific and document-specific acronyms, which are defined and used only within the same document or within a small set of documents produced by the same company or government department. These acronyms cannot be found in general dictionaries and are often introduced in different and sometimes uncommon formats. An automatic method to define abbreviations would help researchers by providing a self updating abbreviation dictionary and also facilitate computer analysis of text. The biomedical literature now available online in Punjabi language [13] presents special challenges for both human readers and automatic algorithms. One such challenge derives from the common and uncontrolled use of abbreviations in the literature[10] which makes difficult to understand document. One specific issue is the high rate at which new abbreviations are introduced in biomedical texts[4]. Existing databases, ontologies, and dictionaries must be continually updated with new abbreviations and their definitions. In an attempt to help resolve the problem, new techniques have been introduced to automatically extract abbreviations and their definitions from MEDLINE abstracts[13] in Punjabi language. Further, since Punjabi political documents and newspapers contain a large number of acronyms, a useful tool for the reader would be a routine that can provide acronym definitions immediately. So Acronym-definitions pair routine will help readers to read online Punjabi Newspaper[2]. An Acronym is usually introduced along with its definition when it is first mentioned in a document. Since the acronyms are found in the text with their definitions, the probability that they are correct is quite high; they can be used to build a database of acronyms automatically and locate instances of these acronyms in the current document or other documents. We conclude that the correct mapping of acronyms to their expansions is very important for understanding the documents and for extracting information from them. We noticed that medical documents (MEDLINE) and Political documents or newspapers in Punjabi languages contain a lot of abbreviated terms, which carry important knowledge about the domains. So ability to organize and extract acronyms can be very useful for Information Extraction tasks and for the complete understanding of text.

A. Acronyms Definition And its Properties

Acronyms are contractions of words or phrases which are used in place of their full versions , where their meaning is clear from the context in which they appear. Acronyms are a type of abbreviation made up of the initial letters or syllables of other words. Acronym is word formed from the initial letter or letters of each of major parts of a compound term or a word formed from first letters of a series of words .Acronyms have following special properties[11]:

- Generally, acronyms do not appear in standard dictionaries. To make their meaning clear, authors may give their expansions at their first use.
- Acronyms may be nested.
- Acronyms are not necessary unique.
- Acronyms are generally three to ten characters in length.

B. *Difference between Acronyms and Abbreviations*

Abbreviations are contractions of words or phrases which are used in place of their full versions, where their meaning is clear from the context in which they appear. Acronyms are a type of abbreviation made up of the initial letters or syllables of other words[16]. Key differences between acronyms and other abbreviations include the lack of symbols such as apostrophe (') and fullstops (.) in acronyms, more standard construction and the use of capital letters.

C. *Types of Acronyms*

1) *Common or Global Acronyms*: Common abbreviations are those that have become widely accepted as synonyms. These represent common fundamental and important terms and are often used, although not explicitly defined within the text[7]. So there is need to explicitly define them into database.
2) *Dynamic or Local Acronym:* Dynamic abbreviations, are defined by the author and used within a particular article[7]. An Local or Dynamic acronyms are usually introduced along with its definition when these are first mentioned in a document.

C. Acronym Identification

Acronym identification is the task of processing text to extract pairs consisting of a word (the acronym) and an expansion (the definition), where the word is the short form of (or stands for) the expansion. . Current Applications of Acronym Identification:

- Useful tool for reader.
- Used to build new tools based on gathered acronym data
- Used to self updating existing abbreviation dictionary in Medline.
- Used for hypertext browsing system.
- Used to enhance text or information retrieval.
- Help existing tools work more smoothly.
- Annotation and decoration of text presented to user in digital libraries.
- Improves the quality of spell checker.
- Used in Post Processing System(PPS).
- Used in Multi Word Expression (MWE) Identification.

II. RELATED WORK

Taghva and Gilbreth (1999) [12] present the Acronyms Finding Program (AF), based on pattern matching. Their program seeks for acronym candidates which appear as upper case words. They calculate a heuristic score for each competing definition by classifying words into: (1) stop words (”the”, ”of”, ”and”), (2) hyphenated words (3) normal words (words that don't fall into any of the above categories) and (4) the acronyms themselves (since an acronym can sometimes be a part of the definition). The AFP utilizes the Longest Common Subsequence (LCS) algorithm (Hunt and Szymanski, 1977) to find all possible alignments of the acronym to the text, followed by simple scoring rules which are based on matches. The performance reported from their experiment are: recall of 86% at precision of 98%. Yeates (1999) [16] proposes the automatic extraction of acronyms-definitions pairs in a program called TLA (Three Letter Acronyms). Although the name suggests that acronyms must have three letters, the system can find *n*-letter acronyms as well. The algorithm divides text into chunks using commas, periods, and parentheses as delimiters. It then checks whether adjacent chunks have acronym letters matching one or more of the initial three letters of the definition words. Further heuristics are then applied to each candidate, ensuring that the acronym is uppercase, is shorter than the definition, contains the initial letters of most of the definition words, and has a certain ratio of words to stopwords.

Another strategy, also developed for the medical field, is from Schwartz and Hearst (2003) [11]. Their approach is similar to Pustejovsky *et al.*'s (2001) strategy and the emphasis is again on complicated acronym-definition patterns for cases in which only a few letters match (e.g., "Gen-5 Related N-acetyltransferase" [GNAT]). They first identify candidate acronym-definition pairs by looking for patterns, particularly "*acronym (definition)*" and "*definition (acronym)*". They require the number of words in the definition to be at most min(A + 5, A ´ 2) , where A is the number of letters in the acronym.2 They then count the number of overlapping letters in the acronym and its definition and compare the count to a given threshold. The first letter of the acronym must match with the first letter of a definition word. They also handle various cases where an acronym is entirely contained in a single definition word.

Chang et al. present an algorithm that uses linear regression on a pre-selected set of features, achieving 80% precision at a recall level of 83%, and 95% precision at 75% recall on the same evaluation collection (this increases to 82% recall and 99% precision on a corrected version).[c] Their algorithm uses dynamic programming to find potential alignments between short and long form, and uses the results of this to compute feature vectors for correctly identified definitions. They then use binary logistic regression to train a classifier on 1000 candidate pairs

Dana Dannells [6] applies a rule-based method to solve the acronym recognition task and compares and evaluates the results of different machine learning algorithms on the same task. The method proposed is based on the approach that acronym-definition pairs follow a set of patterns and other regularities that can be usefully applied for the acronym identification task. Supervised machine learning was applied to monitor the performance of the rule-based method, using Memory Based Learning (MBL). The rule-based algorithm was evaluated on a hand tagged acronym corpus and performance was measured using standard measures recall, precision and f-score.

Nadeau and Turney (2005) [8] present a machine learning approach that uses weak constraints to reduce the search space of the acronym candidates and the definition candidates, they reached recall of 89% at precision of 88%.

III. SYSTEM ARCH ITECTURE

The goal of this research is to implement a system that can be able to find Acronyms and their full form or Expansion from text in Punjabi Language to make dictionary of Acronym-Definition pairs.

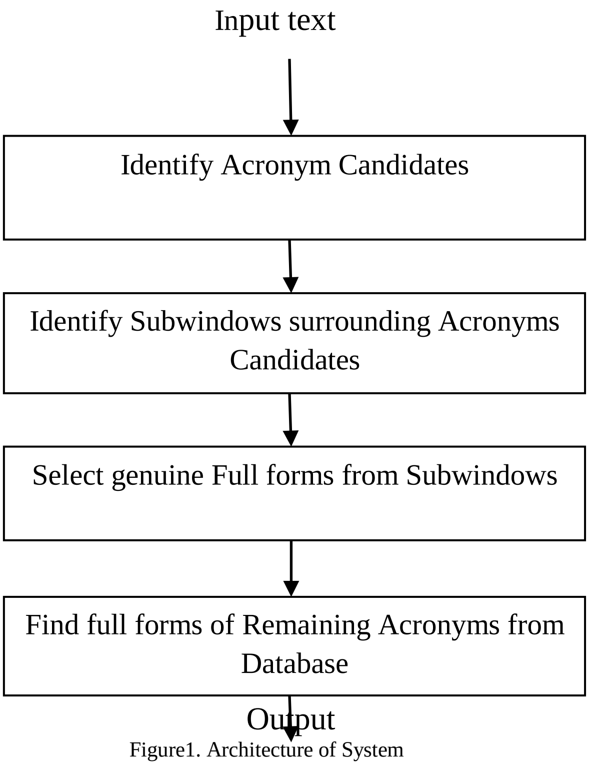
Input text
Identify Acronym Candidates
Identify Subwindows surrounding Acronyms Candidates
Select genuine Full forms from Subwindows
Find full forms of Remaining Acronyms from Database
Output

Figure1. Architecture of System

This is a closed domain system i.e Medical and Political domain. The system will first find Acronyms from text and then their full forms from the surrounding text. There are some common Acronyms whose full form not given in surrounding text so system will find Expansions of common Acronyms from Database. This is closed domain system i.e Medical and Political domain. We choose medical field because more Acronyms are used in this field and political domain because it will help to read political news in punjabi newspapers online.

IV. OUTLINE OF THE ACRONYM DEFINITION FINDING ALGORITHM

A. Algorithm

- Gurmukhi text is entered as input.
- Input text is divided into tokens of single words by markers ‘ ‘, ‘.’, ‘/n’, ‘/r’, ‘(‘, ‘)’.
- Input is normalized by removing empty entries.
- Find Preposition words by using collected punctuation words database and if word is punctuation then attach with the word come before that Preposition word.
- Find All Acronym candidates using rules.
- Find Acronyms Full forms from surrounding text using rules to find full forms.
- If not all Full Forms found then found from common Acronym full form database.
- Acronyms-Definition Pairs are found and the process is terminated.
- Add newly found Acronyms and full forms from surrounding text to Database.

B. *Description of Algorithm*

In our Algorithm first we have done correct mapping of acronyms to their expansions in surrounding text using first-letter matching technique to find Acronym Definitions. Then find remaining Acronym Definitions from Database.

1) Tokenization:

Normalization is the process Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. The list of tokens becomes input for further processing . Here in our Algorithm we divide input into tokens of single words using separators like ‘ ‘, ‘.’, ‘/n’, ‘/r’, ‘(‘, ‘)’ and add tokens to list.

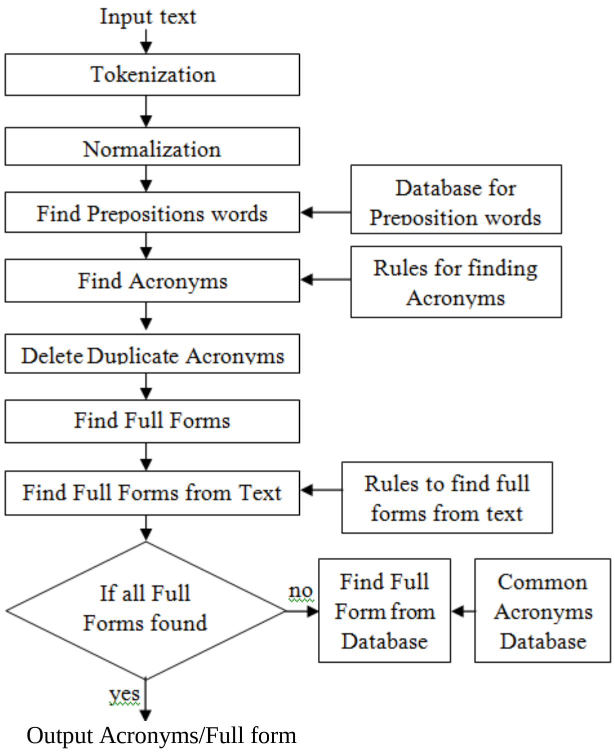
Input text
Tokenization
Normalization
Find Prepositions words
Database for Preposition words
Find Acronyms
Rules for finding Acronyms
Delete Duplicate Acronyms
Find Full Forms
Find Full Forms from Text
Rules to find full forms from text
If all Full Forms found
no
Find Full Form from Database
Common Acronyms Database
yes
Output Acronyms/Full form

Figure2. Flowchart of System

2) Normalization:

Normalization is the process of reorganizing data so that it meets basic requirements. In our Algorithm we normalized text stored in nodes of list by removing the blank spaces ,” ‘ ”, “ ! ”, “ | ” , “ ? “ etc. so to increase the performance of system.

3) To find Preposition words:

Prepositions link words and phrases in a sentence to other words and phrases. In our Algorithm we use the First word matching technique. So we match first letters of Acronyms and Expansions to find genuine expansion for Acronyms. But Prepositions first letter is not involve in the Acronym letters. So to finding the exact match of Acronyms and Expansions and to involve prepositions in output Definition, we first find if word of link list node is preposition or not. If word of link list node is preposition then attach Prepositions with the previous link list node which is previous word in input text because we tokenized input into words and put into link list. Prepositions in the Full form are of mostly single word so we take the condition that preposition should be of one word.

4) To find Acronyms:

There are two steps to find Acronyms, firstly find nodes of link list whose words are letters of Acronym by applying rules and mark them , then combine contiguous letters of Acronym.

- Find letters of Acronyms from input using rules

Rules are made to find the Acronyms from the input text. We check words of link list nodes one by one by applying rules to find the nodes whose words are letters of Acronym. The nodes whose words match with the any of rule means that words are used as letters of Acronyms. That nodes whose words match with the words in rule are marked. Rules used are defined as follows:-

```
if (str=" ਏ ")

return true;
else if (str=" ਬੀ ")

return true;
else if (str=" ਸੀ ")

return true;
else if (str=" ਡੀ ")

return true;
else if (str=" ਈ ")

return true;
else if (str=" ਐਫ ")

return true;
else if (str=" ਜੀ ")

return true;
else if (str=" ਐਚ ")

return true;
else if (str=" ਆਈ ")

return true;
else if (str=" ਜੇ ")

return true;
else if (str=" ਕੇ ")

return true;
else if (str=" ਐਲ ")

return true;
else if (str=" ਐਮ ")

return true;
else if (str=" ਐਨ ")

return true;
else if (str=" ਓ ")

return true;
else if (str=" ਪੀ ")

return true;
else if (str=" ਕਿਊ ")

return true;
else if (str=" ਆਰ ")

return true;
else if (str=" ਐਸ ")

return true;
else if (str=" ਟੀ ")

return true;
else if (str=" ਯੂ ")

return true;
else if (str=" ਵੀ ")

return true;
else if (str=" ਡਬਲਊ ")

return true;
else if (str=" ਐਕਸ ")

return true;
else if (str=" ਵਾਈ ")

return true;
else if (str=" ਜ਼ੇਡ ")

return true;
else
return false;
```

- Combine contiguous letters of Acronyms

In this step, if words of contiguous link list nodes are letters of Acronyms then to find the complete Acronym combine them into one node with one condition. The condition is that, if more than one nodes whose words are letters of Acronym are contiguous, only then nodes are combined and put into one node and that node is complete Acronym candidate.

5) *Delete Duplicate Acronyms:*

Same Acronyms may be used many times in the text. So nodes that contain the words of input text will contain duplicate words many times. But because of this duplicate Acronyms output will show same Acronym as many times used in input text and this will decrease the performance. So we remove the link list nodes that contain duplicate Acronyms. In this step we compare the link list nodes that contain Acronyms to check if they contain duplicate Acronym or not. If contain duplicate Acronyms then that should be removed.

6) *Find Full Forms:*

We find full forms of Acronyms in two ways, from surrounding text and from database. There are some Global Acronyms whose full form is given in the text with acronym when first time used in text. There exist some Acronyms which having many expansions so there will be problem of ambiguity. Because full forms that will be given in text with Acronym will be always correct, so we first find full form of Acronym first from surrounding text to remove ambiguity and then from database made of common Acronyms.

- Find Full Forms from Surrounding text

There are some Global Acronyms whose full form is given in the text with acronym when first time used in text. To find the Genuine full forms of Acronyms from text we used First Letter Matching technique[12]. In Punjabi language we use First Letter Matching technique , by matching the first letters of node's word with sounds of letters of Acronym by using Lettersound Database. All Sounds of all Letters that involve in Acronyms is given as below:-

Letters of Acronym	Sound
ਏ	ਅ,ਆ, ਏ, ਐ
ਬੀ	ਬ
ਸੀ	ਸ,ਕ, ਚ
ਡੀ	ਦ,ਡ
ਈ	ਈ,ਏ, ਇ, ਅ
ਐਫ	ਫ਼,ਫ
ਜੀ	ਜ,ਗ
ਐਚ	ਹ
ਆਈ	ਇ,ਆਈ
ਜੇ	ਜ, ਝ
ਕੇ	ਕ
ਐਲ	ਲ
ਐਮ	ਮ
ਐਨ	ਨ
ਓ	ਵ,ਓ, ਔ, ਆ, ਅ
ਪੀ	ਪ, ਫ
ਕਿਊ	ਕ
ਆਰ	ਰ
ਐਸ	ਸ,ਜ਼, ਸ਼
ਟੀ	ਤ,ਟ, ਦ
ਯੂ	ਉ,ਯੂ, ਅੰ, ਅ
ਵੀ	ਵ
ਡਬਲਯੂ	ਵ
ਐਕਸ	ਕਸ,ਕਸ਼
ਵਾਈ	ਈ,ਯ
ਜ਼ੇਡ	ਜ਼

Table 1. Sounds of Letters

An appropriate window around the Acronym is searched. We taken both Prewindow and Postwindow . The maximum size of subwindow is calculated as: (|A|* 2) Where |A| is number of letters in Acronyms .After describing the window size, to find genuine candidates first letters of words of those nodes are checked against the sound of letters of Acronyms which involve in subwindows. The condition for genuine full form found is:-The Sound of all letters of Acronyms should match with the first letters of node's word and these all nodes should be contiguous.

- Find Full Forms from Database

we create two Databases:

-Database of Common Medical Acronyms-Definition pairs

-Database of common political Acronym-Definition pairs

If full forms are not found from surrounding text then we find Acronym full form from Medical Database or Political Database according to domain chosen for input text. We have taken 594 common Medical Acronyms and 315 common Political Acronyms. We taken common Medical Acronyms from websites [5] and from medical books in Punjabi language. We have taken common Political Acronyms from newspaper[2].

V. EVALUATION

The Algorithm is tested by using different medical and political documents. We evaluated our system by using common Acronym database and without using common Acronym database. Automatic Acronym-Definition Pairs finding Algorithm has been tested over two types of data- Medical Documents[13], Political data. Performance is measured using three Standard measures Recall, Precision, F-score .

- **Recall:** Recall is defined as number of correct Acronym definitions found by our system from total number of definitions in the document[11]. Recall measures how thoroughly the system finds all Acronyms.

$$\text{Recall} = \frac{\#\ \text{Correct Acronym definitions found by system}}{\text{Total \# Acronym}}$$

definitions in document

- **Precision**: Precision is defined as the correct number of Acronym definitions found by system from total number of Acronyms found by our system[11] . Precision indicates the no. of errors produced.

$$\text{Precision} = \frac{\text{\# correct Acronym definitions found by system}}{\text{Total \# Acronym definitions found by system}}$$

- **F-Score:** F-score is a composite measure which benefits algorithms with higher sensitivity and challenges algorithms with higher specificity. F-Score is the harmonic mean of precision and recall.

$$F = 2\ \frac{\text{Precision. Recall}}{\text{Precision + Recall}}$$

For evaluation we have compared the number of Acronym-Definitions manually calculated from document with number of Acronym-Definitions found by our system..

We have conducted experiments over two documents: Abstract of MEDLINE (D1) and Political document (D2). The data used in the experiments and experimental result are shown in Table 2 performance is evaluated using recall and precision.

Document		D1	D2
Size(# of words)		17342	15289
No. of Acronyms in document		50	52
No. of Acronyms definitions in document		43	45
Definitions Found by system from surrounding text	Correct	34	36
	Incorrect	6	4
	Total	40	41

Table 2. Test Data and Experimental Results without using common Acronym Database

For Medical document system found 40 Acronyms and their definitions but among them 6 Acronyms are wrong. The result shown were 79.06% recall and 85% precision, yielding F-score of 81.92%. For D2, it found 43 Acronyms and their definitions but among them 5 Acronyms are wrong. The result shown were 80% recall and 87.80% precision, yielding F-score of 83.71%. So when we provide database of common Acronyms in medical and database of common Acronyms in political.

We Evaluated system by giving same medical document(D1) and political document(D2) as input to check recall by using common Acronym databases . Now we calculate Recall as:

$$\text{Recall} = \frac{\text{\# Acronym definitions found by system}}{\text{Total \# Acronyms in document}}$$

Document		D1	D2
Size(# of words)		17342	15289
No. of Acronyms in document		50	52
No. of Acronyms definitions in document		43	45
Definitions Found by system from both surrounding text and Database	Correct	37	43
	Incorrect	6	2
	Total	43	45

Table 3. Test Data and Experimental Results by using common Acronym Database

So By using Database of common Acronym-Definition pairs Recall is improved. The result shown were 86% recall and 86.04% precision, yielding F-score of 86.01% for D1. The result shown were 86.53% recall and 95.55% precision, yielding F-score of 90.81% for D2.

VI. CONCLUSION AND FUTURE WORK

Our algorithm did quite well on Medical and Political document collections and found almost all Acronyms and their definitions. The algorithm is extremely simple, it is highly effective, and is less specific – and therefore less potentially brittle – than other approaches. Another advantage of the simplicity of the algorithm is its fast running time performance. Recall is improved by our algorithm by using Database of common Acronym-Definition pairs. We identified main classes of Acronym definitions missed by our algorithm are Acronyms formed from more than one letter of each of major parts of compound term, Acronyms containing special characters like & etc., Definitions containing composite prepositions. Further Increasing Subwindow size used surrounding the Acronym, Extending system to open domain, Removing Ambiguity , managing when composite preposition used in Definition, handle special characters in Acronym and Using other techniques improvement can be done in future.

ACKNOWLWDGEMENT

First and foremost, I would like to thank God almighty for life itself. All that I have is due to His grace and I give all glory to him. With deep sense of gratitude I express my sincere thanks to my esteemed and worthy supervisor Er. Jagroop Kaur, Assistant professor, Department of Computer Engineering, Punjabi University, Patiala for her valuable guidance in carrying out this work.

REFERENCES

[1] Acronym finder :<http://www.acronymfinder.com/>

[2] Ajit Punjabi newspaper < www.ajitjalandhar.com>.

[3] Anna Yarygina. “High-recall extraction of acronym-definition pairs with relevance feedback”.Saint-Petersburg University.

[4] Ariel S. Schwartz and Marti A. Hearst. 2003. “A simple algorithm for identifying abbreviation definitions in biomedical texts”. Proc. of the Pacific Symposium on Biocomputing. University of California, Berkeley.

[5] Common medical abbreviations< www.abbreviation.com/ category/ MEDICAL>.

[6] Dana Dannells,”Automatic Acronym Recognition”.

[7] Dana Movshovitz-Attias. “Alignment-HMM-BASED Extraction of Abbreviations from Biomedical Text”. Carnegie Mellon University.

[8] David Nadeau and Peter Turney. 2005.” A Supervised Learning Approach to Acronym Identification”. Information Technology National Research Council,Ottawa, Ontario, Canada.

[9] Jablonski S (ed). “Dictionary of Medical Acronyms and Abbreviations”. Philadelphia, Hanley & Belfus, 1998 .

[10] J.T. Chang, H Schütze, and R.B. Altman, “Creating an Online Dictionary of Abbreviations from MEDLINE” JAMIA, to appear.

[11] J. Xu and Y. Huang. “Using svm to extract acronyms from text”. Soft Comput., 11:369{373, November 2006.

[12] Kazen Taghva and Jeff Gilbreth. 1999. Technical Report. “Recognizing Acronyms and their Definitions”.University of Nevada, Las Vegas.

[13] MEDLINE Abstracts in Punjabi < www.nlm.nih.gov/medlineplus/languages/punjabi.html> .

[14] N. Okazaki and S. Ananiadou.” A term recognition approach to acronym recognition”. In Proceedings of the COLING/ACL on Main conference poster sessions, COLING-ACL '06, pages 643{650, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[15] Park Y, Byrd RJ (2001) “Hybrid text mining for finding abbreviations and their definitions”. In: Proceedings of the 2001 conference on empirical methods in natural language processing, Pittsburgh, pp 126–133.

[16] Stuart Yeates. 1999.” Automatic extraction of acronyms from text”. Proc. of the Third New Zealand Computer Science Research Students’ Conference. University of Waikato, New Zealand.

[17] Yeates S, Bainbridge D, Witten IH (2000) Using compression to identify acronyms in text. In: Proceedings of data compression conference, IEEE Press, New York, pp 582.