



Design And Development Of Hybrid Approach For Constructing Gene/Protein Names Dictionary

Dr.B.LShivakumar*

Associate Professor, Department of Computer Applications,
Sri Ramakrishna Engineering College,
Coimbatore, India
blshiva@yahoo.com

R.Porkodi

Assistant Professor, Department of Computer Science,
Bharathair University,
Coimbatore – 46, India
porkodi_r76@yahoo.co.in

Abstract: Gene/Protein names identification in biomedical texts is an important challenge in bioinformatics. Several approaches have been proposed to tackle this problem. Machine learning and statistical techniques proved to be useful. Other methods focus on linguistic techniques, or are based on the usage of dictionaries extracted from databases, ontologies, and other data sources. Some methods rely on the combination of dictionaries and linguistic/machine learning techniques. This paper focuses on the development of hybrid method that combines rule based and n-gram statistical technique to identify and extract gene and protein names and construct dictionary for it.

Keywords: Information Extraction, Gene name, Protein name, Regular Expression, Medline abstracts, Dictionary.

I. INTRODUCTION

The current MEDLINE database includes over 18 million computer-readable records in the biomedical domain and is expanding rapidly; it is a rich resource for biological knowledge including protein-protein interactions [1], gene regulation events [2], sub-cellular locations of proteins [3], and pathway discovery [4]. One way to automatically extract information stored in MEDLINE is to apply an information extraction system such as a natural language processing (NLP) parser [5]. Identifying gene/protein terms in MEDLINE abstracts is a necessary step towards an information extraction system. Genes and proteins are usually represented by symbols and names in literature. The names usually are the long forms of their symbols and describe the functions of the genes or proteins.

The identification of gene and protein names is a challenging task because both do not follow any standard nomenclature. This paper analyzed and studied the protein and gene names available in biomedical repositories for deeper understanding of their word formations in terms of short names and full names and also its helps to provide efficient techniques for identifying gene and protein names. Dictionary-based approaches normalize gene and protein names, reducing many synonyms and phrases representing the same concept to a single identifier for that gene or protein. In addition, dictionary-based approaches make use of the huge amount of information in curated genomics databases. Dictionary-based methods have used existing terminological resources and various string matching approaches to locate gene mentioned in text, and thus perform both tasks simultaneously (linking textual strings to matching database entries). Due to variability and ambiguity of gene names, simple pattern matching typically results in low precision and moderate recall. These approaches are

generally enhanced with additional rule- and token-class based techniques, while distinguishing between important and less important constituents.

Rule-based approach has an advantage that rules can be flexibly defined and extended as needed, whereas manually analyzing targeted domain texts and building rules are often time-consuming. Statistical approach is relatively easy to be adapted to different domains if appropriate training corpora are provided. On the other hand, statistical approach in general cannot reasonably deal with the cases that do not appear in the training corpora. Many approaches such as rule based, machine learning and statistical techniques have been developed in which rule based method proves better and efficient in terms of its simplicity. However, rule based method may fails to recognize new protein and gene symbols which are not given in training data set. In statistical techniques, many techniques have been used to identify gene and protein symbols from biomedical text and this may also have some limitations. Currently, researchers are working towards developing hybrid approach which combines both rule based approach and statistical methods. This paper focuses on the development of gene and protein names dictionary using hybrid approach. This work combines rule based approach and n-gram statistical method to identify gene and protein symbols from Medline abstract and construct gene and protein names dictionary.

The paper is organized as follows: section 2 discusses the related work, section 3 focuses on methodology which includes rule based approach and N-gram statistical method, section 4 discusses on results and finally the work is concluded in section 5.

II. RELATED WORK

Hui Yang, Goran Nenadic¹ et al. (2007) [6] have presented a generic and effective rule-based approach to link

gene mentions in the literature to referent genomic databases, where preprocessing of both gene synonyms in the databases and gene mentions in text are first applied. The mapping method employs a cascaded approach, which combines exact, exact-like and token-based approximate matching by using flexible representations of a gene synonym dictionary and gene mentions generated during the pre-processing phase. They also consider multi-gene name mentions and permutation of components in gene names. A systematic evaluation of the suggested methods has identified steps that are beneficial for improving either precision or recall in gene name identification.

Koning D, Sarkar I et al. (2006) [7] has proposed a rule-based tool, which consists of a number of rules based on regular expressions. Using an English-language dictionary, it finds all words that are not in the common-language dictionary, and applies rules based on character case and term order in order to determine whether a term is a species name or not. The idea is used in the work.

Martin Krallinger, Maria Padron et al. (2005) [8] have developed another sub-tag set containing protein variants which were generated through a rule based pipeline of protein name processing (e.g., *O00115: DNASE2, DNASE 2* and *DNASE-2*). Hanisch D, Fundel K et al. (2005) [9] proposed a gene dictionary that includes various spelling variants to support gene name matching, including an approximate matching procedure in which it treats each (candidate) string as a sequence of tokens, which are assigned to corresponding classes (e.g. measurement, digit, modifier, etc.). The classes are then used to weight mismatches in the approximate matching (e.g. the mismatch weight for the modifier class (which includes tokens such as receptor, precursor) is high).

Chang JT, Schutze H et al. (2004) [10] presented a supervised learning approach to acronym identification. In order to circumscribe the learning, they impose a strongly restrictive condition on candidate acronym-definition pairs, by searching only for “*definition (acronym)*” patterns. Interestingly, this pattern accounts for the majority of positive cases in their evaluation corpus. Chang *et al.*’s learning algorithm uses eight features describing the mapping between acronym letters and definition letters (e.g., percentage of letters aligned at the beginning of a word, number of definition words that are not aligned to the acronym, etc.). The learning algorithm they used is logistic regression.

Hong Yu,a, Vasileios Hatzivassiloglou (2003) [11] proposed genes and proteins are usually represented by symbols and names in literature. The names usually are the long forms of their symbols and describe the functions of the genes or proteins.

Schwartz, A. and Hearst, M. (2003) [12] proposed an approach that emphasis 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.

Tanabe L, Wilbur WJ (2002) [13] proposed an idea to retrained Brills tagger on the biomedical domain for gene/protein name-identification. Yu H, Hatzivassiloglou V(2002) [14] proposed the method for retrieved synonyms of proteins and genes from abstracts and full text, and identified more synonyms with higher precision in full text, with the introduction section defining the majority of synonyms.

Park, Y., and Byrd, R.J., (2001) [15] proposed an approach that combines mechanisms such as text-markers and linguistic cues with pattern-based recognition. The same combination was used by Larkey. This removes some constraints on the acronyms that can be identified. The reason for these mechanisms is to cope with the growing popularity of acronyms that diverge from the tradition of using only the first letter of each word of the definition. They use cue expressions (e.g., “or”, “short”, “acronym”, “stand”) to reinforce the confidence in acronym-definition pairs. They also allow acronyms to include a digit at the beginning or the end.

Collier NH, Nobata C et al.(2000) [16] applied statistical methods (e.g., hidden Markov models, decision trees, and support vector machines) for detecting and classifying gene and gene product names including proteins. The features used in their methods are mostly the same as those used in rule-based approaches, that is, surface clues and parts of speech.

Krauthammer, M., Rzhetsky et al. (2000) [17] presented a Basic Local Alignment Search Tool (BLAST)-based system approach. This uses approximate string matching techniques and dictionaries to recognize spelling variations in gene or protein names. This encodes gene names and text in terms of the nucleotide alphabet and has used BLAST to look for ‘homologies’ between a query gene name and the text.

Larkey, L., Ogilvie (2000) [18] Compared various strategies and found their Canonical/Contextual method to be the most accurate. First they force candidate acronyms to be in upper-case, allowing only embedded lower case letters (internal or final), periods (possibly followed by spaces), hyphens (or diagonal slashes) and digits (at most one, non-final digit). They allow a maximum of nine alphanumeric characters in acronyms. They search for expansions in a window of 20 words, adjacent to the given acronym. Stop words can contribute to an inner letter, but only once for the entire acronym. Furthermore, an expansion is only valid if it fits a given pattern, such as being surrounded by parentheses or preceded by a cue phrase (e.g., “also known as”)

Yoshida M, Fukuda K et al. (2000) [19] developed, specifically for mapping protein symbols to full names PNAD-CSS (for “Protein full Name Abbreviation Dictionary - Construction Support System”). PNAD-CSS used morphological features to recognize proper nouns as

protein terms in biological abstracts. Knowing a phrase may contain a protein symbol and full name, PNAD-CSS recognized parentheses and determined whether the parenthetical phrase was an abbreviation of the outer phrase. To map a protein symbol to its name, PNAD-CSS broke up words of the preceding phrase, and determined whether the parenthetical abbreviation candidate maps to the initial letters of the broken-up phrase.

Nobata C, Collier N et al.(1999) [20] described the Machine-learning approaches in which hidden Markov Model and decision trees are used to classify gene/protein names. Fukuda K, Tamura A et al. (1998) [21] has proposed a number of rule-based, linguistic, statistical, machine-learning, and hybrid approaches have been developed to mark up gene/protein terms automatically in biological text. For example, Fukuda et al. applied morphological cues to identify protein terms (e.g., if a word contains uppercase letter(s) and special character(s), the word is a protein term).

III. METHODOLOGY

The methodology and framework of proposed approach for constructing Dictionary is shown in Figure 1 and the same is represented using pseudo code. The framework consists of three phases. In the first phase the Gene and Protein names are extracted from Medline abstract and added to dictionary using rule based approach..

In the second phase text mining technique is used to identify and extract Gene and Protein names automatically from Medline Abstracts and subsequently updates the created dictionary. The third phase verifies and validates the performance and efficiency of the created dictionary by using precision, recall and F-measure metrics.

The Gene/Protein dataset is downloaded from the NCBI Entrez and used to train the Medline abstracts. The NCBI dataset is the integrated, text-based search and retrieval system used at the major databases, including PubMed, Nucleotide and Protein Sequences, Protein Structures, Complete Genomes, Taxonomy, and others. The Gene and Protein dataset contains information such as Gene names, Protein names with their corresponding information like Official symbol, Official Full name, Primary Source, Locus Tag, Gene Type, RefSeq Status, Organism, and Lineage.

A. Preprocessing:

a. Filtering:

Each Medline abstract contains information along with two character prefixes. The retrieved Medline abstracts are selected individually and parsed to filter the important fields such as PMID, TI, MeSH terms and AB.

PMID	-PubMed Identifier which is used for indexing
TI	- Title of the Medline Abstracts
MeSH terms	-Medical Sub Heading terms used for indexing the abstracts in PubMed database.
AB	- Medline Document Abstract

These are the only fields taken into consideration for the proposed research work.

```

Procedure Create_Dictionary( )
// create Dictionary using Rule based approach
Let GP_Dataset[ ] be the details relevant to Genes and
Proteins from NCBI
Let Reg_Exp[ ] be the framed Regular Expressions //Refer
section 4.3.1.2
Let GP_Dictionary[ ] is initialized to Null
For each entity in GP_Dataset[ ] do
    //add entity into dictionary
    For each re in Reg_Exp[ ] do
        If entity matches with re then
            GP_Dictionary[ ] ←entity
        End if
    End for each
End for each
// create Dictionary using N-gram approach
Let MA_Dataset be the list of Medline abstracts of
particular interest
Let stop_words[ ] be the list of stop words
Let verb_words[ ] be the list of verbs
//compute stream of tokens from Medline abstracts
Let str_token[ ] ←Null
For each abs in MA_Dataset do
    Str_token[ ] ←Tokenization(abs)
    //remove stop words and verbs from set of tokens
    Str_token_arr [ ]← Remove(str_token, stop_word,
very_words)
    //add entity into dictionary
    For each i in str_token_arr do
        entity←identify entity by n-gram
approach
        if entity is not in GP_Dictionary then
            GP_Dictionary[ ]←entity
        End if
    End for each
End for each

```

b. Tokenization:

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 such as parsing or text mining. The Medline abstracts are converted into the token string by using the regular expression given in Eq. 9 in subsequent section.

c. Stop word Removal:

After filtering the four important fields, the stop words from Title and Abstract, i.e., common English words which do not provide meaningful information are considered for removal. The stop-words are high frequent words that carry no information (i.e. pronouns, prepositions, conjunctions etc.). Removal of stop-words improves information extraction results. The Medline abstracts contain protein names, gene symbols and other words. The other words may include stop words and verbs that may play no rule in the

proposed work, thus those words are removed by using the stop word list and verb list. Some of the stop words and verbs are shown in Table 1.

Table 1 Sample set of Stop words and Verbs

Stop word list	Verb list
'a'	'accept'
'aand'	'add'
'able'	'admire'
'abnormally'	'admit'
'about'	'advise'
'above'	'afford'
'abs'	'agree'
'absent'	'alert'
'absolutely'	'allow'
'accompanied'	'amuse'
'accompanies'	'analyse'
'accompany'	'announce'
'accompanying'	'annoy'

Stemming means the process of suffix removal to generate word stems. Suffix removal algorithms do not rely on a lookup table that consists of inflected forms and root form relations. Instead, a typically smaller list of "rules" is stored which provides a path for the algorithm, given an input word form, to find its root form. Some examples of the rules include:

- if the word ends in 'ed', remove the 'ed'
- if the word ends in 'ing', remove the 'ing'
- if the word ends in 'ly', remove the 'ly'

The Porter stemmer [17] which is a well-known algorithm is used in this work to remove suffixes in the Medline abstracts. It is a highly effective, simple algorithm that removes word suffixes in order to reduce related words (e.g. connected, connection) to the same stem (e.g. connect).

d. Stemming:

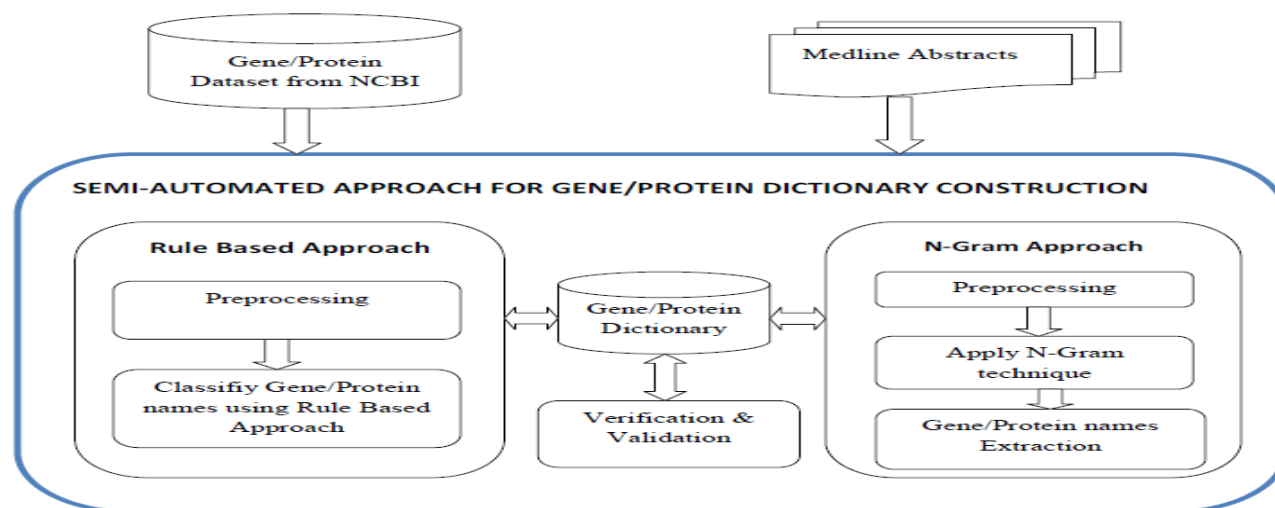


Figure 1 A Framework for constructing Dictionary for Gene and Protein Names

B. Identifying Gene And Protein Names Using Rule Based Approach:

The extraction of Gene and Protein names involves set of rules which are identified by learning the nomenclature of genes and proteins that are maintained in different biomedical repositories and also from the biomedical literatures. The rules are framed using Regular Expression (REG EXP) and the following regular expressions were framed to generalize the gene and protein symbol naming conventions to construct the dictionary from the full description of protein names extracted from the dataset. The snapshot of rules for creating gene and protein names dictionary is shown in Table 2.

Table 2 Some of the rules for extracting Gene and Protein names

S.No.	RULES	REG EXP
1	The abbreviation letter matches the first letter of each word in the full name.	'[A-Z]\w*\d* [a-z]\w*\d*\$'
2	The abbreviation letter matches the last capital letter of a word in the full name.	'[A-Z]*'
3	The abbreviation matches the first letter of each word with Roman alphabets.	'[A-Z]\w*\d*'
4	The abbreviation matches the first word of upper case followed by the normal word of protein name.	'[A-Z]\w*[a-z]\w*\d*\$'
5	The abbreviation matches the first word of upper case followed by the Arabic numerals followed by the normal word of protein name.	'[A-Z]*\d*'
6	The abbreviation matches the special abbreviation of Gene and Protein symbol combine with the normal word of the protein name.	'[A-Z]\w*'

7	The abbreviation letter matches the first capital letter of a word in the full name.	'w*[A-Z]\d'
8	Using some special abbreviation of the full name.	'[A-Z]\w*\d*\[a-z]\d*\$'

C. Identifying Gene And Protein Names Using N-Gram Approach:

The regular expression given in Eq. 9 is used to split the Medline abstract into number of tokens. The token string can have all the combination of stop words, verbs, and protein names and gene symbols and so on. The stop word and verbs were removed from the list downloaded from the NCBI website. In this preprocessing step we removed all the unnecessary words such as stop words and verbs from the MEDLINE abstracts to generate token strings.

A word n -gram model is used to detect word position which indicates whether a word is the beginning, in-between, or ending word in the multi-word term. In proposed approach the biological terms are identified by a set of character types, such as uppercase letters, lowercase letters, digits, symbols and so on.

After removing the stop words and verbs from the Medline abstract, the remaining words are matched with the created dictionary, to find that position of the string and the corresponding position of the word is fetched from the created dictionary.

According to the words the n -gram approaches uses 2 gram approach, or 3 gram approach, or 4 gram approach. For example, the word 'glycoprotein' uses the one gram approach to fetch the word 'transmembrane' and create the protein word 'Transmembrane glycoprotein'. Similarly the other n -grams are used to extract the protein names from the Medline abstracts. Using this approach the protein names are automatically updated to the manually created dictionary.

In Medline abstract the protein names are mentioned in terms of capital letter words, proteins, receptors, chains, and combination of upper case word followed by the number. Using this method the Gene and Protein names are extracted from the Medline abstract. For example 'growth' is a word extracted using 2-gram approach for the protein 'Keratinocyte growth factor'. Similarly all the words that exist in proteins are extracted using 2-gram to 5-gram from Medline abstracts. To identify genes 1-gram is enough because gene names always a single word that is represented in capital letter which includes numerals, Greek letter, etc. After applying the N-gram approach, we ended with the protein names and gene symbols.

The extracted tokens are checked with the dictionary for its availability, if it is found leave it, otherwise add it into dictionary.

D. Verification and Validation:

Verification and Validation is another phase of the Gene and Protein name dictionary. In this phase the protein names are correctly identified by evaluating using the validation metrics. We evaluated the dictionary for the Gene and Protein name by using "precision", "recall", and "F-score" or "F-Measure" metrics. Precision is a measure of 'exactness'.

Recall is a measure of 'completeness'. Precision is defined as the number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search, and recall is defined as the number of relevant documents retrieved by a search divided by the total number of existing relevant documents (which should have been retrieved). F-measure is the harmonic mean of recall and precision.

IV. RESULTS AND DISCUSSION

This part discusses the implementation and results of the proposed dictionary.

The rule 1 is used to extract the normal form of a protein name abbreviation which will be in the form of starting with Capital letter and ending with Arabic numerals. More than 3500 Gene and Protein names are generated using this regular expression, in which few Gene and Protein names are shown in Table 3.

The rule 2 is used to extract the protein name abbreviation ending with Capital letter word. More than 500 Gene and Protein names are extracted in which few results are shown in Table 4.

The rule 3 is used to extract the normal form of a protein name abbreviation ending with Roman alphabets. More than 750 Gene and Protein names are generated which few results are shown in Table 5.

The rule 4 is used to extract the protein name starting with upper case word and ending with Arabic numeral. More than 1250 Gene and Protein names are generated which few results are shown in Table 6.

The rule 5 is used to extract the protein name starting with capital letter word followed by the Arabic numerals end with Arabic numeral. More than 1300 Gene and Protein names are generated which few results are shown in Table 7.

Table 3 Gene and Protein names ending with Arabic numerals

Protein Names	Protein/Gene Symbol
Fibroblast growth factor 1'	FGF1'
Heparin-binding growth factor 2'	HBGF2'
Keratinocyte growth factor 4'	KGF4'
Glia-activating factor 10'	GAF10'
Follicle-stimulating hormone receptor 3'	FSHR3'
Guanylate-binding protein 22'	GBP22'
Growth/differentiation factor 5'	GDF5'
Growth hormone receptor 11'	GHR11'
Growth hormone-releasing hormone receptor 12'	GHRHR12'
Glutamate receptor, ionotropic kainate 5'	KRIK5'
Glia-derived nexin 7'	GDN7'
Gastrin-releasing peptide receptor 8'	GRPR8'
Glutathione S-transferase kappa 23'	GSTK23'
Glutathione S-transferase Mu 1'	GSTM1'
Glycerol kinase 5'	GK5'
Host cell factor 2'	HCF2'

Table 4 Protein names ending with capital letter word

Protein Names	Protein/Gene Symbol
'Proto-oncogene tyrosine-protein kinase FGR'	'FGR'
'Pre-mRNA 3'-end-processing factor FIP1'	'FIP1'
'Peptidyl-prolyl cis-trans isomerase KFBP1A'	'KFBP1A'
'Peptidyl-prolyl cis-trans isomerase KFBP1B'	'KFBP1B'
'Pre-mRNA-splicing regulator WTAP'	'WTAP'
'Zinc finger protein ZFPM1'	'ZFPM1'
'Zinc finger protein ZFPM2'	'ZFPM2'
'ARF GTPase-activating protein GIT1'	'GIT1'
'ARF GTPase-activating protein GIT2'	'GIT2'
'Poly(A) RNA polymerase GLD2'	'GLD2'
'Nucleoporin GLE1'	'GLE1'
'Tetratricopeptide repeat protein GNN'	'GNN'
'Synaptic glycoprotein SC2'	'SC2'

Table 5 Protein names ending with Roman alphabets

Protein Name	Protein/Gene Symbol	
'Hepatocyte nuclear factor 3-alpha'	'HNF3A'	HNF3-alpha'
'Hepatocyte nuclear factor 3-beta'	'HNF3B'	HNF3-beta'
'Hepatocyte nuclear factor 3-gamma'	'HNF3G'	HNF3-gamma'
'Hepatocyte nuclear factor 1-alpha'	'HNF1A'	HNF1-gamma'
'Hepatocyte nuclear factor 1-beta'	'HNF1B'	HNF1-beta'
'Hepatocyte nuclear factor 4-alpha'	'HNF4A'	HNF4-alpha'
'Hepatocyte nuclear 4-gamma'	'HN4G'	HNF4-gamma'
'Heat shock protein 11 beta'	HSP11B'	HSP11-beta'
'Heat shock protein 1 beta'	HSP1B'	HSP1-beta'
'Heat shock protein 2beta'	HSP2B'	HSP2-beta'
'Heat shock protein 3 beta'	HSP3B'	HSP3-beta'
'Heat shock protein 6 beta'	HSP6B'	HSP6-beta'
'Heat shock protein 7 beta'	HSP7B'	HSP7-beta'
'Heat shock protein 8 beta'	HSP8B'	HSP8-beta'
'Heat shock protein 9 beta'	HSP9B'	HSP9-beta'

Table 6 Gene and Protein names starting with upper case and ending with Arabic numerals

Protein Names	Protein/Gene Symbol
'GA-binding protein subunit beta-1'	'GABPB1'
'GA-binding protein alpha chain'	'GABPA'
'GMP reductase 1'	'GMPR1'
'GMP reductase 2'	'GMPR2'
'RAS guanyl-releasing protein 1'	'RASGRP1'
'RAS guanyl-releasing protein 2'	'RASGRP2'
'GTP-binding protein 1'	GTPBP1'
'GTP-binding protein 2'	GTPBP2'
'GTP-binding protein 3'	GTPBP3'
'GTP-binding protein 4'	GTPBP4'
'GTP-binding protein 5'	GTPBP5'
'GTP-binding protein 6'	GTPBP6'
'GTP-binding protein 7'	GTPBP7'

The rule 6 is used to the extract special abbreviation of the Gene and Protein name. More than 300 Gene and Protein names are generated which few results are shown in Table 8.

The rule 7 is used to extract the protein name starting with upper case word followed by the Arabic numeral. More than 200 Gene and Protein names are generated which few results are shown in Table 9.

Table 7 Proteins starting with capital letter followed by numeral ending with Arabic numeral

Protein Names	Protein/Gene Symbol
Hsp70-binding protein 1'	'HSPBP1'
Hsp70-binding protein 2'	HSPBP2'
Hsp70-binding protein 3'	HSPBP3'
Hsp70-binding protein 4'	HSPBP4'
Hsp70-binding protein 5'	HSPBP5'
Hsp70-binding protein 6'	HSPBP6'
Hsp70-binding protein 7'	HSPBP7'
Hsp70-binding protein 8'	HSPBP8'
Hsp70-binding protein 9'	HSPBP9'
Hsp70-binding protein 10'	HSPBP10'
Hsp70-binding protein 11'	HSPBP11'
Hsp70-binding protein 12'	HSPBP12'

Table 8 Proteins extracted using REG EXP for special abbreviation

Protein Names	Protein/Gene Symbol
'Forehead box protein J1'	'FOXJ1'
'Forehead box protein K1'	'FOXK1'
'Forehead box protein L1'	'FOXL1'
'Forehead box protein M1'	'FOXMI1'
'Forehead box protein Q1'	'FOXQ1'
'Forehead box protein S1'	'FOXSI1'
'Forehead box protein D4-like 4'	'FOXDL4'
'G antigen 13'	GAGE13'
'G antigen 1'	'GAGE1'
'G antigen 3'	'GAGE3'
'G antigen 4'	'GAGE4'
'G antigen 5'	'GAGE5'
'G antigen 6'	'GAGE6'

Table 9 Gene and Proteins names starting with upper case followed by Arabic numeral

Protein Name	Protein/Gene Symbol
INT-1 proto-oncogene protein'	INT-1'
INT-2 proto-oncogene protein'	INT-2'
INT-3 proto-oncogene protein'	INT-3'
INT-4 proto-oncogene protein'	INT-4'
INT-5 proto-oncogene protein'	INT-5'
INT-6 proto-oncogene protein'	INT-6'
INT-7 proto-oncogene protein'	INT-7'
INT-8 proto-oncogene protein'	INT-8'
INT-9 proto-oncogene protein'	INT-9'
INT-10 proto-oncogene protein'	INT-10'
INT-11 proto-oncogene protein'	INT-11'
INT-12 proto-oncogene protein'	INT-12'
INT-13 proto-oncogene protein'	INT-13'
INT-14 proto-oncogene protein'	INT-14'
INT-99 proto-oncogene protein'	INT-99'
6PGL endoplasmic bitfunctional protein	6PGL'

The rule 8 is used to extract the protein name abbreviation which will be in the form of some special abbreviation. More than 220 Gene and Protein names are generated which few results are shown in Table 10.

The number of Gene and Protein extracted against with each rule is summarized in Table 11 and same is represented using bar chart for better understanding in Figure 2.

Table 10 Protein names with special abbreviation

Protein Name	Protein/Gene Symbol
'Granyme A'	'GZMA'
'Granyme B'	'GZMB'
'Granyme H'	'GZMH'
'Granyme K'	'GZMK'
'Girdin'	'GRDN'
'Gremlin-1'	'GREM1'
'Fizzy'	'FZY'
'Granulins'	'GRN'
'Huntingtin'	'HTT'
'Hepcidin'	'HEPC'
'Histatin-1'	'HTN1'
'Histatin-3'	'HTN3'
'Hemicentin-1'	'HMCN1'
'Heparanase'	'HEP'
'Filamin'	'FLN'
'Flotillin'	'FLOT'
'Indoleamine'	'IN'
'ligand'	'LG'
'Interferon'	'IFN'
'Phosphatase'	'PC'
'Galactose'	'GAL'
'Neuroleukin'	'NLK'

Table 11 Total No. of Protein /Gene extracted using Rules

Regular Expression Rules	Gene and Protein Count
Regular Expression Rule 1	3500
Regular Expression Rule 2	500
Regular Expression Rule 3	750
Regular Expression Rule 4	1250
Regular Expression Rule 5	1300
Regular Expression Rule 6	300
Regular Expression Rule 7	200
Regular Expression Rule 8	220

Rules Vs No. of Gene/Protein Names

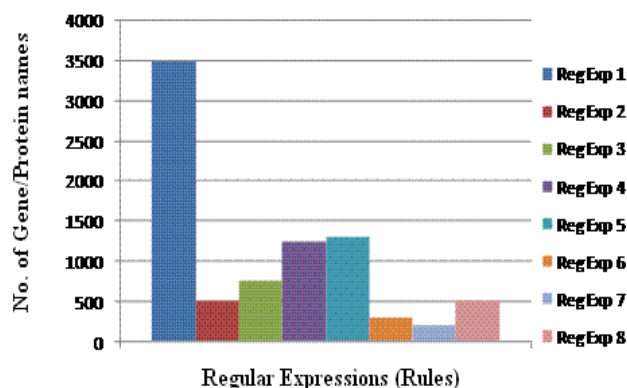


Figure 2 Gene and Protein extratced using REG EXP

The extracted Gene and Protein names from Medline abstract using rule based approach are evaluated for its correctness using precision, recall and F-Measure as shown in Table 12 and the same is represented in graph as shown in Figure 3.

Table 12 Precision, Recall and F-Measure

Medline Abstracts	Precision (%)	Recall (%)	F-Measure (%)
10	80	83	82
30	79	83	81
50	76	84	80

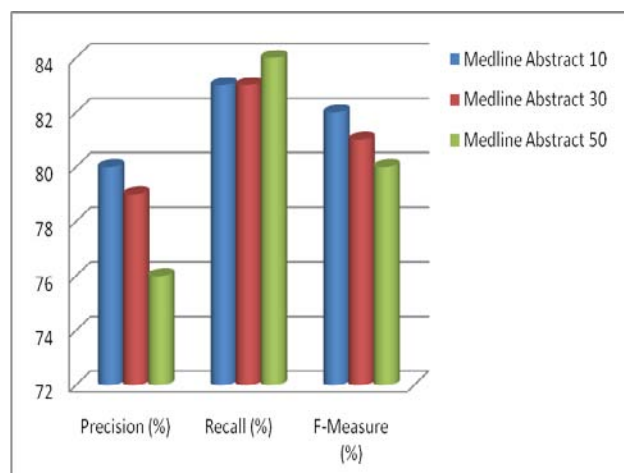


Figure 3 Precision, Recall and F-Measure metrics Vs Medline abstract

From the above results the precision value of 10 Medline abstract is 80%, Recall value is 83% and F-Measure value is 82%. Similarly for 30 Medline abstract the precision value is 79%, Recall value is 83% and F-Measure value is 81%. Similarly for 50 Medline abstract the Precision value is 76%, Recall value is 84% and F-Measure value is 80%. The evaluation of the metric the F-measure was found to be 81% as average.

The N-gram approach is applied to the Medline abstract to extract the Gene and Protein names. From the Medline abstracts the extracted tokens were converted to gene or protein names using n-gram approach. Out of 5000 token after pre-processing, 3400 protein names were identified. The identified Gene and Protein names are checked with the dictionary for its availability. If it is not available, consider that as a new Gene and Protein name and add it into Dictionary. The results of Gene and Protein names extracted for construction of dictionary using N-gram approach from Medline abstract as shown in Figure 4. The summarization of results is shown in Table 13 and the same is represented using bar chart as shown in Figure 5.

Table 13 Summary of Results

Medline Abstract Tokens	Count
Medline Abstract	50
Token words count	5000
After preprocessed the token count	3400
Full name identified from the abstract	800
Added to the dictionary	500
Updated to the dictionary	300

The extracted gene and protein names are validated using the precision, recall and F-measure metrics. The calculated TP, TN, FP, FN values are shown in Table 14. The result of precision, recall and F-measure is tabulated in Table 15 and the same is represented in bar chart in Figure 6.

Table 14 Cross Matrix

Medline Abstract		Positive	Negative
10	True	522 Words	50 Words
	False	62 Words	112 Words
30	True	1625 Words	250 Words
	False	325 Words	312 Words
50	True	3425 Words	468 Words
	False	620 Words	540 Words

Table 4.15 Precision Recall, F-Measure using N-gram for Medline Abstracts

Medline Abstracts	Precision (%)	Recall (%)	F-Measure (%)
10	89	82	85
30	83	83	83
50	84	86	85

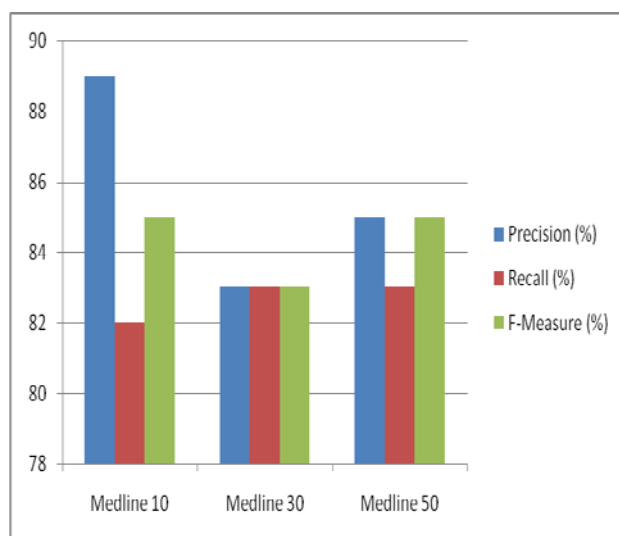
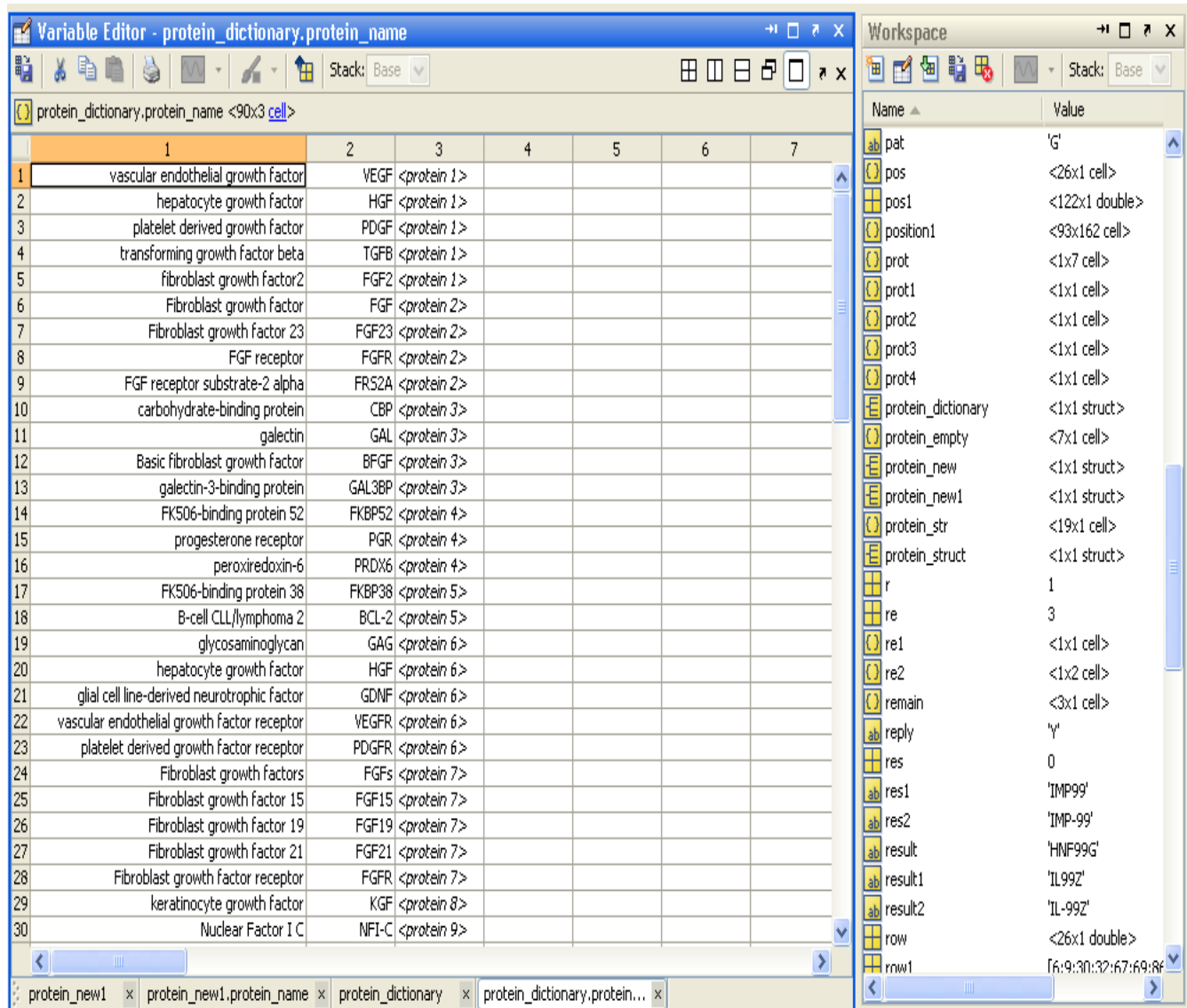


Figure 6 Precision, Recall and F-Measure metrics Vs Medline abstract

From the above results the precision value of 10 Medline abstract is 89%, Recall value is 82% and F-Measure value is 85%. Similarly for 30 Medline abstract the precision value is 83%, Recall value is 83% and F-Measure value is 83%. Similarly for 50 Medline abstract the Precision value is 84%, Recall value is 86% and F-Measure value is 85%. The evaluation of the metric the F-measure was found to be 85% as average.

A. Comparison of N-gram approach with GENIA Tagger:

The proposed work is compared with the existing biological tagger GENIA. The performance of our approach is almost equal to GENIA tagger for some Medline abstracts. The proposed approach extracted 126 tokens and identified 40 protein names and the GENIA tagger extracted 219 tokens and identified 46 protein names.



Variable Editor - protein_dictionary.protein_name

protein_dictionary.protein_name <90x3 cell>

	1	2	3	4	5	6	7
1	vascular endothelial growth factor	VEGF	<protein 1>				
2	hepatocyte growth factor	HGF	<protein 1>				
3	platelet derived growth factor	PDGF	<protein 1>				
4	transforming growth factor beta	TGFB	<protein 1>				
5	fibroblast growth factor2	FGF2	<protein 1>				
6	Fibroblast growth factor	FGF	<protein 2>				
7	Fibroblast growth factor 23	FGF23	<protein 2>				
8	FGF receptor	FGFR	<protein 2>				
9	FGF receptor substrate-2 alpha	FRS2A	<protein 2>				
10	carbohydrate-binding protein	CBP	<protein 3>				
11	galectin	GAL	<protein 3>				
12	Basic fibroblast growth factor	BFGF	<protein 3>				
13	galectin-3-binding protein	GAL3BP	<protein 3>				
14	FK506-binding protein 52	FKBP52	<protein 4>				
15	progesterone receptor	PGR	<protein 4>				
16	peroxiredoxin-6	PRDX6	<protein 4>				
17	FK506-binding protein 38	FKBP38	<protein 5>				
18	B-cell CLL/lymphoma 2	BCL-2	<protein 5>				
19	glycosaminoglycan	GAG	<protein 6>				
20	hepatocyte growth factor	HGF	<protein 6>				
21	glial cell line-derived neurotrophic factor	GDNF	<protein 6>				
22	vascular endothelial growth factor receptor	VEGFR	<protein 6>				
23	platelet derived growth factor receptor	PDGFR	<protein 6>				
24	Fibroblast growth factors	FGFs	<protein 7>				
25	Fibroblast growth factor 15	FGF15	<protein 7>				
26	Fibroblast growth factor 19	FGF19	<protein 7>				
27	Fibroblast growth factor 21	FGF21	<protein 7>				
28	Fibroblast growth factor receptor	FGFR	<protein 7>				
29	keratinocyte growth factor	KGF	<protein 8>				
30	Nuclear Factor I C	NFI-C	<protein 9>				

Workspace

Name	Value
pat	'G'
pos	<26x1 cell>
pos1	<122x1 double>
position1	<93x162 cell>
prot	<1x7 cell>
prot1	<1x1 cell>
prot2	<1x1 cell>
prot3	<1x1 cell>
prot4	<1x1 cell>
protein_dictionary	<1x1 struct>
protein_empty	<7x1 cell>
protein_new	<1x1 struct>
protein_new1	<1x1 struct>
protein_str	<19x1 cell>
protein_struct	<1x1 struct>
r	1
re	3
re1	<1x1 cell>
re2	<1x2 cell>
remain	<3x1 cell>
reply	'Y'
res	0
res1	'IMP99'
res2	'IMP-99'
result	'HNF99G'
result1	'IL99Z'
result2	'IL-99Z'
row	<26x1 double>
row1	[6:9:30:32:67:69:86]

Figure 4 Gene and Proteins names identified using N-gram approach

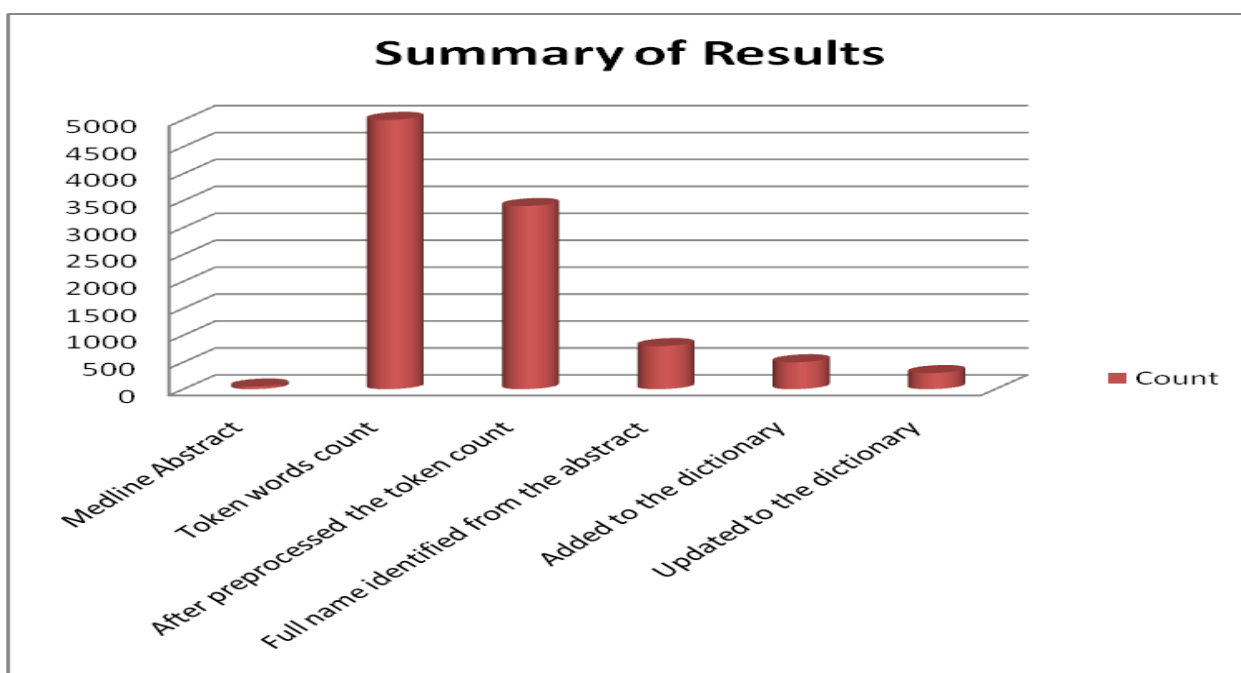


Figure 5 Summary of Results by Proposed approach

V. CONCLUSION

The proposed work presents the hybrid approach which combines rule based approach and N-gram statistical method for constructing gene and protein names dictionary from Medline abstracts. that consists of four main tasks. In the first step, pre-processing is carried out to remove the inconsistencies from the dataset. In the second step, the Gene and Protein names are extracted from Medline abstracts using regular expressions and added to dictionary, in the second step, the Gene and Protein names are extracted from Medline abstracts using N-gram statistical method and added to dictionary and in fourth step the extracted gene and protein names are validated and verified using precision, recall and F-measure. The experimental result shows that the rule based approach provides 81% accuracy in identifying Gene and Protein names, which is evaluated and verified using the Precision, Recall and F-Measure and the N-gram statistical method shows 85%. The limitations include the ambiguity in usage of gene/protein terms. For example, we do not differentiate a gene term from a protein one. We do not differentiate a general gene/protein term (e.g., growth factors) from a specific one (e.g., protein kinase A). The proposed works also do not identify to which organism, tissue, cell type, and sub location a gene/protein term refers, this may be considered in our next coming approach. In future, we have an idea to propose an approach for disambiguating gene/protein terms and also hope to develop statistical NLP approaches for further disambiguation.

VI. ACKNOWLEDGMENT

This work was performed as part of the Minor Research Project, which is supported and funded by University Grants Commission, New Delhi, India

VII. REFERENCES

- [1]. Blaschke C et al. Automatic extraction of biological information from scientific text: protein-protein interactions. *Proc Int Conf Intell Syst Mol Biol* 1999:60-7.
- [2]. Stapley BJ, Benoit G. Biobibliometrics: information retrieval and visualization from co- occurrences of gene names in Medline abstracts. *Pac Symp Biocomput* 2000:529-40.
- [3]. Stapley BJ, Kelley LA, Sternberg MJE. Predicting the sub-cellular location of proteins from text using support vector machines. In: *PSB, Hawaii*, 2002.
- [4]. Ng SK, Wong M. Toward routine automatic pathway discovery from on-line scientific text abstracts. *Genome Inform Ser Workshop Genome Inform* 1999;10:104-12.
- [5]. Carol Friedman, P.K., Michael Krauthammer, Hong Yu, Andrey Rzhetsky. GENIES: A Natural-Language Processing System for the Extraction of Molecular Pathways from Complete Journal Articles. In: *ISMB*, 2001.
- [6]. Hui Yang, Goran Nenadic¹, John A. Keane¹ (2007), "A cascaded approach to normalising gene mentions in biomedical literature", *Biomedical Informatics Publishing Group*, 2007.
- [7]. Koning D, Sarkar I, Moritz T: TaxonGrab (2006) "Extracting taxonomic names from text ", *Biodiversity Informatics* 2006, 2:79-82.
- [8]. Martin Krallinger, Maria Padron and Alfonso Valencia (2005), "A sentence sliding window approach to extract protein annotations from biomedical articles", *BMC Bioinformatics* 2005, 6 (Suppl 1):S19 doi: 10.1186/1471-2105-6-S1-S19.

- [9]. Hanisch D, Fundel K, Mevissen HT, Zimmer R, Fluck J (2005), ProMiner: rule-based protein and gene entity recognition, BMC Bioinformatics, 6: S14 (2005).
- [10]. Chang JT, Schutze H, Altman RB (2004), "GAPSCORE: finding gene and protein names one word at a time". Bioinformatics 2004, 20(2):216-225.
- [11]. Hong Yu,a, Vasileios Hatzivassiloglou,a Andrey Rzhetsky,b and W. John Wilburc (2003), "Automatically identifying gene/protein terms in MEDLINE abstracts". Biomedical Informatics 2003.
- [12]. Schwartz, A. and Hearst, M. (2003), "A simple algorithm for identifying abbreviation definitions in biomedical texts", In Proceedings of the Pacific Symposium on Biocomputing (PSB).
- [13]. Tanabe L, Wilbur WJ (2002), "Tagging gene and protein names in biomedical text". Bioinformatics 2002, 18(8):1124-1132.
- [14]. Yu H, Hatzivassiloglou V, Friedman C, Rzhetsky A, Wilbur WJ (2002), "Automatic extraction of gene and protein synonyms from MEDLINE and journal articles". Proc AMIA Symp 2002:919-923.
- [15]. Park, Y., and Byrd, R.J., (2001), "Hybrid Text Mining for Finding Abbreviations and Their Definitions", Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing, Pittsburgh, PA.
- [16]. Collier NH, Nobata C, Tshjii J(2000), "Extracting the names of genes and gene products with a hidden markov model". In: Proceedings of the 18th International Conference on Computational Linguistics, 2000, p. 201–7.
- [17]. Krauthammer, M., Rzhetsky, A., Morozov, P., and Friedman C. (2000), "Using blast for identifying gene and protein names in journal articles", Gene. 245-152.
- [18]. Larkey, L., Ogilvie, P., Price, A. and Tamilio, B. (2000), "Acrophile: An Automated Acronym Extractor and Server", In Proceedings of the ACM Digital Libraries conference, pp. 205-214.
- [19]. Yoshida M, Fukuda K, Takagi T (2000), "PNAD-CSS: a workbench for constructing a protein name abbreviation dictionary". Bioinformatics 2000; 16(2):169–75.
- [20]. Nobata C, Collier N, Tsujii J (1999), "Automatic term identification and classification in biology texts". In Proceedings of the Natural Language Pacific Rim Symposium (NLPRS_99), 1999.
- [21]. Fukuda K, Tamura A, Tsunoda T, Takagi T. "Toward information extraction: identifying protein names from biological papers". Pac Symp Biocomput 1998:707–718.
- [22]. M.F.Porter, "An Algorithm for Suffix Stripping", Program 14(3), July 1980, pp.130-137.