IJCTA, 8(4), 2015, pp. 1409-1419

A predictive alignment algorithm to identifythe definitions of abbreviation in biomedical texts

Ahmad Ghelichi*, Ahmad Faraahi, KomeilShahvari*****

Abstract: One of the grounds raised in the last few years is the search and extraction of data from biomedical literature. The size and growth of the biomedical literature has created new challenges. Text mining techniques will pave the way to answer this question. Just extracting the definitions for abbreviations and biology is very essential. One of the challenges is a high rate of new abbreviations which introduce, develop and occur in biomedical texts. In this article, we have suggested a combinatorial alignment algorithm to detect abbreviations from biomedical texts. The method is to identify short form and long form pairs where short form of any kind of character is mapped to the long form. In this algorithm, some abbreviations which were not found in the former method can be found. The evaluation found that the algorithm shows high precision compared to previous algorithms.

Key words: biomedical abbreviations, biomedical contexts, data mining, text mining.

1. INTRODUCTION

Text mining issue is to detect practical knowledge and unknown new data from non- structured and semi structured texts[1] which includes three major themes: 1) Information Retrieval(dependent documents collecting), 2) Information Extraction (extracting interesting data from these documents), and 3) data mining (detecting new dependence throughout the extracted data parts). Exploding growth of biomedical texts causes increasing enthusiasm in applying such techniques in biomedical and biological texts[2] and also MEDELINE's data base size multiplied rather than last decade; and this causes new challenges in the context of Information Retrieval[3].

A great number of new knowledge related to biomedical researches is saved as available contexts in the shape of journal articles or written formsin data base. Reliable progress of Language process systems which recovers related documents extracts related data and explores new data from free contexts can help biomedical researchers for better management of saved knowledge in this context[4].

A critical part in these systems is writing contextual chains for biomedical concepts. Because of complexity of the biomedical domain, biomedical sentences are often long[5]. They usually include the words which bode their corresponding semantic types such as: "Virus in epestein-barr virus", or "protein in latent membrane protein", or the words which describe characteristics of the referred entities such as: "Latent in latent membrane protein". In one time, maybe it is difficult to find describing and short sentences for biomedical concepts like genes and proteins. New abbreviations using the issue are being developed in biomedical text mining[6]. For more comfortable connections, short viewing of biomedical concepts like summery, abbreviations, and signs in context which occurs repeatedly or is hard to describe is being used. Since there are several names and abbreviations in many of biomedical existences, it is so good that an automatic mean facilitates text mining themes for collecting these synonym words and abbreviations. If all the words and abbreviations for one existence could be written as a single

* Department of Computer& Information Technology,Payame Noor University (PNU), P.O. Box, 19395-3697, Tehran (IRAN), *E-mail: Ahmad.ghelichi@gmail.com*

** Department of Computer& Information Technology,Payame Noor University (PNU), *E-mail: Afaraahi@pnu.ac.ir*

*** Department of Computer& Information Technology,Payame Noor University (PNU), *E-mail: K.shahvary@gmail.com*

sentence in the context, it will be a field work in Information Extraction issue, synonym words of a name decryption of gene and abbreviations of biomedical sentences[7]. Abbreviations and summary usually are used for illnesses and etcetera in biomedical contexts for names of gene. Since the changes of abbreviations-definitions are dependent to the context, they can cause ambiguity[8]. The ability to detect and extract abbreviations and writing them on an optimized definition for data extracting field could be useful[9].

2. RELATED WORKS

Published biomedical papers volume is growing up with increasingthe speed annually. By biomedical knowledge development with this fast pace, there are a lot of challenges for biology researchers who want to keep being updated. Thus, an automatic method for biomedical knowledge text mining is absolutely essential[10]. In biomedical text mining, the use of new abbreviations issue is being developed[6]. One of the related items to this field is the high rate of new abbreviations which are introduced in biomedical contexts. Data basis, anthologies, and existence of dictionaries should be updated to new abbreviations and their definitions continually. In an attempt done for this issue, new techniques are introduced which automatirecally extract the abbreviations with their definitions from the MEDELINE abstracts.

This could have a remarkable contribution automatirecally for recovering these contexts. In addition to other fields of text mining, if all the abbreviations for one existence could be written for a concept of the single phrase, they may have better application[7]. Usually, an abbreviation is a short form of a word or phrases which is recalled definition or long form. Our job is to detect < short form, long form > couples in which exists a writing of existing character in short form to long form characters[11]. The existing methods are faced problems in four fields. The methods are based on statistics, regulated, text alignment, and machine learning.

The methods based on statistics always tend to extract the abbreviations which appear alternately in biomedical contexts and need a big collection of contexts. Zhou et al,[12] made a data base recalled Adam which analyses statistical data about a collection of long form kinds (abbreviation) in MEDELINE. Ananiadou, and Chruszcz [13] made a dictionary of all MEDELINE abbreviations. Although based on statistic methods shows a high precision, they cannot find some special abbreviations and need more time and work.

The regulated method tries to use the best detection law; good laws (valid) could have good results (valid). Pustejovsky et al, [11] presented a regular phrase algorithm which was based on handmade regular phrases and discussed syntactic data for detecting range of nominal phrase. Ao et al, [4]built a system recalled Alice which was based on heuristic pattern matching rules. Park and Byrd[14] and Yu et al, [9] worked on match rules of their pattern separately. One weak point of regulated methods is that their application is specified by all the rules.

The methods based on machine learning usually include a learner and a predictor and is put in all kinds of biomedical contexts by learning. Chang et al. [15] presented a method for detecting the abbreviations by using learning machine supervised. The methods based on machine learning usually depend on learning data model and need to more time and work.

The methods based on text alignment always try to find an optimized alignment between definitions and abbreviations. Schwartz and Hearst [16] presented a simple algorithm for detecting definitions of abbreviations with only two indexes, an index for long form (Lindex), and an index for short form (Sindex), two index points to a point at the end of their related chain. For each character which it points at Sindex, Lindex reduces until a character match is found. Cohen & Hersh[7]and Taghva & Gilbreth[17] used the longest following subsequence in their method. Movshovitz-attias, and Cohen[8] presented an algorithm which considers between abbreviation character and probable definition for every match and has a good performance. Anyway, technical status of alignment algorithm is so that it cannot find irregular abbreviations and their precision is low.

In the method based on text alignment, irregular abbreviation is not being found, and detecting wrong abbreviations by previous algorithms causes our research to be necessary that we use of a text alignment algorithm. This text alignment algorithm is like binary sequence alignment that Schwartz and Hearst [16] used in their work. So we improve the algorithm in order to optimize it for detecting similar definitions for abbreviations. In alignment between

abbreviations and definitions, we use an array for matching abbreviation characters and definitions and put some limitations. Using array for reprocessing the saved data will improve the performance of the algorithm. In this method the algorithm has a high precision and detects some irregular abbreviations.

3. SUGGESTED ALGORITHM

Published biomedical papers volume is growing up with increasing the speed annually. Usually, abbreviations are used in biomedical text too much. An abbreviation is a short form of a word or phrase which is recalled definition or long form. Our job is to identify < short form, long form > couples in which exists a writing of existing character in short form to long form characters.

In suggested algorithm, we use a text alignment method like Schwartz and Hearst [16]alignment. We improve the algorithm in order to optimize it for detecting similar definitions for abbreviations. In suggested method we use a regulated method for detecting abbreviation-definition couples. In this algorithm, we use the array to compare abbreviation and definition characters. Because every letter is saved in an array cell, we have possibility of going back and using that letter for matching. The algorithm presumes that abbreviation or definition occurs in neighborhood of parentheses. Algorithm scans the definition the words from the end of abbreviation to the beginning. Yet it tries in every stage to find the match for every substring the abbreviation. In order to find a character of abbreviation through the definition words, the algorithm at first compares every abbreviation letters with all the definition letters if it found the matching, it searches for next abbreviation letter. In Schwartz and Hearst's algorithm [16], the action of comparing for finding next abbreviation letter, the search will continue from the last letter found in matching definition, but in order to improve algorithm precision, we search to letters at the end of the abbreviation from the end of definition letters; because in reviewing abbreviations, we concluded that some of them with last word of definition, match in two letters. In continue, the suggested algorithm is discussed and reviewed.

3.1. Suggested algorithm routine

To find < short form, long form > couples, algorithm can be divided to two subtasks. One is to identify abbreviations and the other is to identifythe definitions. Our main job is second subtask; identifying the definitions. Identifying the

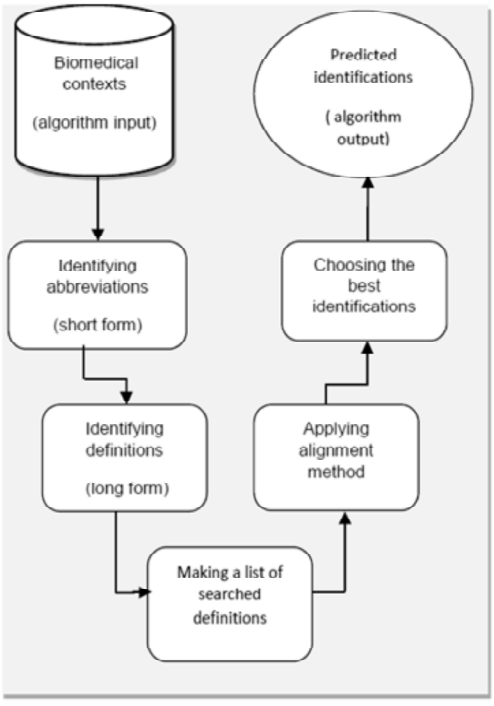

Figure1: abbreviation extraction algorithm diagram and their corresponding definitions

```
booleanisValidAbbreviation(String str)
{ Identify the correct abbreviation;
Check the validity of the abbreviation; }
booleanhasLetter(String str)
{ Check the validity of characters abbreviation; }
}
Vector extractAbbrPairs(String inFile)
{ Identify the correct definition; }
voidextractAbbrPair(String Abbreviation, String Defination)
{ abbreviation and definition pairs extraction (Making a list of searched definitions)}

String AlignmentAlgorithm(String Abbreviation, String Defination)
    {Applying alignment method and Extract the correct definition; }
```

Figure2: pseudo-code implementing algorithm

definitions also could be divided to two parts; searching the definitions, and identifying the definitions. Figure 1 shows the complete routine of algorithm diagram and figure 2 shows pseudo-code implementation algorithm.

3.2. Identifying abbreviations

In order to identify abbreviations, we basirecally use the presented detection method by Park and Byrd[14]. Their method is to build up on some rules that include abbreviation characteristics (or short form) and syntactic signs in context. Abbreviation characteristics include the following items:

Its first character should be from alphabet or number, at least includes one letter, its lengths should be between 2 to 10 character, and at last include 2 words.

Syntactic signs include following patterns:

1. Long form (short form) or Long form [short form], 2. short form (long form) or short form [long form], 3. short form = long form, 4. long form = short form, 5. short form or long form, 6. long form or short form, 7. short form. . . stands/short/acronym. . . long form, 8. long form, short form for short. In action, most of the abbreviations appear with parentheses, like “protein, Kinasec (PKC)”. In this work we use a similar approach which most of the researchers use for detecting abbreviations and just review pattern 1 and 2. For pattern 2, the short form of one or two words is before left parentheses and the long form is just the phrase inside the parentheses. For pattern 1, the abbreviation is inside the parenthesis but the long form is not found easily and we need to search before the left parenthesis in order to find it.

By reviewing abbreviation exist in biomedical context, we concluded abbreviation do not include the letters (=, <, > !), so the algorithm does not identify the abbreviation which have these as abbreviations and also when extracting abbreviation from the context, we identify the abbreviation which include: “;”, until we reach this letter, because the letters after “;” are explanations about abbreviation. In Schwartz and Hearst’s algorithm[16], this rule is not conducted.

3.3. Searching definitions

We have the identified abbreviation; the next step is to identify definitions of candidates.Definitions of candidates appear at the same sentence which we found abbreviation and that definition could be found in a search space. The size of search space at last is from the open parenthesis of abbreviation to the left side until reaching the beginning of the sentence.

Park and Byrd[14] analyzed about 4500 abbreviations and their definitions, and then they decided that, for relatively short abbreviations (from two to four characters), the maximum length of a definition should not be greater than twice the abbreviation length (the number of the characters in an abbreviation); for long abbreviations (five or more characters), the definition should not be longer than the abbreviation length + 5. Thus, we refer to their work for the length of a definition DEF andlength of aabbreviation ABBR. So we have the relation 1:

$$\text{Max}|\text{DEF}|=\text{Min}(|\text{ABBR}|+5\ ,|\text{ABBR}|*2) \quad (1)$$

In this relation, Max|DEF| is Maximum length definition and | ABBR | is the number of characters in the stands.

Then a definition of the candidate list is constructed from the search space, and the possible definition is just one item of it. The list-creating algorithm is described below.

Steps to create a candidate definitions list (CDL):

1: Initiate an empty candidate definition list CDL;

2: Num = the number of words from the beginning of the sentence which contains the abbreviation to the left parenthesis;

3: if (Num < Max.|DEF|) *{*

SearchSpaceString = the string from the beginning of the sentence to the left parenthesis;

} else *{*

SearchSpaceString = the string that contains Max.|DEF| words before the left parenthesis;

}

4: WordNum = min (Num, Max.|DEF|);

5: for (N = 0; N < WordNum; N++) *{*

CandidateDef = SearchSpaceString with the leftmost N words deleted; insert CandidateDef into CDL; *}*

3.4. Identifying definition

Now we have a candidate definition list. Each time we retrieve an item from the list, and align it with the abbreviation employing our alignment algorithm. Afterward we select the optimal definition.

3.5. Pre-processing of data

Usually a definition is abbreviated with a new addition of a special character (e.g., < Myo3/5p, Myo3p and Myo5p >), and the lowercase letter from a definition may be changed into its corresponding capital letter. Before we identify the definition corresponding to an abbreviation, some data preprocessing steps must be taken. We delete the character that is neither alphabetic nor numeric in the abbreviation and change all capital letters in both the abbreviation and definition into their corresponding lowercase letters.

3.6. Alignment method

The definition identification is a process of comparison between the abbreviation and definition. In the process, the smallest unit of comparison is a pair of characters, one from the abbreviation, and the other from the definition. All possible comparisons sons are made from the smallest unit while allowing gap insertions in the abbreviation. Among the comparisons the definition with the best match is chosen as the correct definition.

We put the obtained strings in the array. A[i] is the ith character of the abbreviation string and D[j] is the jth character of the definition string. We begin from the end of abbreviation string and comparing it with definition's

last character. Every array cell A[i] with array cell D[j]. If two compared characters are equal, we move left in definition string. In algorithm, in order to improve the precision, by reviewing done to existing abbreviation in biomedical contexts, some abbreviations have two matching letters in the last word of definition, for this reason, in order to find the last letter and the letter before the last in the definition; we begin the comparison from the end of definition. If the is any matching between two compared character, we move forward the definition string one character and if we did not reach at the end of abbreviation, we move the abbreviation one character to the beginning of the character. We continue these comparisons until we reach the end of array A. at the end of comparison, if there is no matching between abbreviation and definition, and the amount of index array D, the definition is smaller than zero, we stop the process of searching optimized definitions. Index array D at the end of action shows a letter in definition which abbreviation matching has finished there.According to the studies on biomedical abbreviations, the first letter of the abbreviation is usually matched with the first letter of a word of definition, thus by obtaining the last letter's index in definition; we search an empty space in order to obtain optimized definition through the candidate definitions. Some of the definitions, to separate the word from each other used the letter "/" instead of empty space, so in this algorithm, we use "/" beside the empty space, in order in order to find better optimized definition.

We check algorithm stages with some examples. The first example is related to situation in which the suggested algorithm, finds correct definition for abbreviation. In this example, "VEPs" as abbreviation and,"Electroretinograms and flash and pattern visual evoked potentials" as definition of extracted candidate are obtained. Table 1 shows comparison stages for finding optimized definition for "VEPs" abbreviation every array element A is equal to an abbreviation letter and every array letter D is equal to an extracted definition letter. Dark cells show a matching between definition and abbreviation.

Table 1
comparison for extracting optimized definition for VEPs abbreviation.

phase	A[i] D[j]	phase	A[i] D[j]	phase	A[i] D[j]
1	A[3]→s = D[70]→s	8	A[2]→p!= D[64]→e	17	A[0]→v!= D[54]→e
2	Compare the beginning of the definition: A[2]→p!= D[70]→s	9	A[2]→p!= D[63]→t	18	A[0]→v!= D[53]→
		10	A[2]→p!= D[62]→o	19	A[0]→v!= D[52]→l
		11	A[2]→p = D[61]→p	20	A[0]→v!= D[51]→a
3	A[2]→p!= D[69]→l	12	A[1]→e!= D[60]→	21	A[0]→v!= D[50]→u
4	A[2]→p!= D[68]→a	13	A[1]→e!= D[59]→d	22	A[0]→v!= D[49]→s
5	A[2]→p!= D[67]→i	14	A[1]→e = D[58]→e	23	A[0]→v!= D[48]→i
6	A[2]→p!= D[66]→t	15	A[0]→v!= D[57]→k	24	A[0]→v = D[47]→v Definition found!
7	A[2]→p!= D[65]→n	16	A[0]→v!= D[56]→o		

In table 1, the letter 47 of definition, is the point that is the beginning of definition of optimized, so the phrase "Visual evoked potentials "is extracted as a correct definition for "VEPs" abbreviation.

In second example, we study a situation in which the suggested algorithm, correctly detects the phrase that is wrongly identified as abbreviation. In this example, "4Ac" is as abbreviation, and "Than for a mutation that enhances the interaction" is as the definition of extracted candidate. Table 2 shows comparison stages for finding optimized definition for "4Ac" abbreviation.

In table 2, the definition and the abbreviation do not have any matches, and algorithm detects it correctly.

The suggested algorithm has a high precision, because the use of some rules in order to extract accurate definitions for abbreviations. We will show the correctness of this action by evaluating suggested algorithm, because the algorithm detects some irregular abbreviations.

Table 2
comparison for extracting optimized definition for abbreviation 4Ac

phase	A[i] D[j]	phase	A[i] D[j]	phase	A[i] D[j]
1	A[2]→c!= D[48]→n	19	A[0]→4!= D[35]→h	38	A[0]→4!= D[16]→i
2	A[2]→c!= D[47]→o	20	A[0]→4!= D[34]→t	39	A[0]→4!= D[15]→t
3	A[2]→c!= D[46]→i	21	A[0]→4!= D[33]→	40	A[0]→4!= D[14]→a
4	A[2]→c!= D[45]→t	22	A[0]→4!= D[32]→s	41	A[0]→4!= D[13]→t
5	A[2]→c = D[44]→c	23	A[0]→4!= D[31]→e	42	A[0]→4!= D[12]→u
6	Compare the beginning of the definition: A[1]→a!= D[48]→n	24	A[0]→4!= D[30]→c	43	A[0]→4!= D[11]→m
		25	A[0]→4!= D[29]→n	44	A[0]→4!= D[10]→
7	A[1]→a!= D[47]→o	26	A[0]→4!= D[28]→a	45	A[0]→4!= D[9]→a
8	A[1]→a!= D[46]→i	27	A[0]→4!= D[27]→h	46	A[0]→4!= D[8]→
9	A[1]→a!= D[45]→t	28	A[0]→4!= D[26]→n	47	A[0]→4!= D[7]→r
10	A[1]→a!= D[44]→c	29	A[0]→4!= D[25]→e	48	A[0]→4!= D[6]→o
11	A[1]→a = D[43]→a	30	A[0]→4!= D[24]→	49	A[0]→4!= D[5]→f
12	A[0]→4!= D[42]→r	31	A[0]→4!= D[23]→t	50	A[0]→4!= D[4]→
13	A[0]→4!= D[41]→e	32	A[0]→4!= D[22]→a	51	A[0]→4!= D[3]→n
14	A[0]→4!= D[40]→t	33	A[0]→4!= D[21]→h	52	A[0]→4!= D[2]→a
15	A[0]→4!= D[39]→n	34	A[0]→4!= D[20]→t	53	A[0]→4!= D[1]→h
16	A[0]→4!= D[38]→i	35	A[0]→4!= D[19]→	54	A[0]→4!= D[0]→t There is no adjustment!
17	A[0]→4!= D[37]→	36	A[0]→4!= D[18]→n		
18	A[0]→4!= D[36]→e	37	A[0]→4!= D[17]→o		

4. RESEARCH FINDINGS

In this part, we present the regular alignment algorithm evaluation results and the results of the analysis and comparison with other algorithms. Implementing this algorithm is done by JAWA programming language in Eclipse IDE for Java Developers environment. The experiments are done on a PC with Mononuclear CPU, 2 GB RAM, and windows7.

4.1. Performance evaluation criteria

We use the precision and recall which is usually used in evaluating. Precisionrange, is correct identifying definition precision, which is the number of correct couples from < short form, long form > in algorithm's output on the number of all the couples in algorithm's output. andrecall is the range of the numbers of correct couples at algorithm output on the number of all correct couples of collection of given data. Labelling TP as positive corrects, it means correct identified couples, EP, wrong positives, it means, the wrong identified couples, and wrong negative FN which is thetotal couples of file that in result, are not identified by the algorithm. The relation 2 is precision and relation 3 is recall and relation 4, gives us the computing formula F-measure:

$$\text{Precision}=\frac{TP}{TP+FP} \tag{2}$$

$$\text{Recall}=\frac{TP}{TP+FN} \tag{3}$$

$$F-measure=\frac{2*precision*recall}{precision+recall} \tag{4}$$

4.2. Evaluating the suggested algorithm

To evaluate algorithm we used standard golden biomedical context (medastract) which is available for public and includes 400 abstracts of biomedical articles. With hand labelling of this contexts, we produced a 461 list of correct abbreviation-definition couples. The results should be exactly on the nonelabelled contexts and the extracted couples should exactly match with signed couples. In order to obtain accurate and scientific results, we use the collection of couples which was used by Moushitz and Cohen in HMM alignment algorithm. They obtained all the meaningless and meaningful abbreviation-definition couples from golden standard in their work which totally extracted 483 couples toward 461 existing couples and labelled them. Also in order to be able to compare the algorithm with its older examples, we use previous existing golden standard contexts which Schwartz and Hearst [16] used in their work and included 168 abbreviation-definition couples.

As the result of performing algorithm on golden standard context with its implementing program by Java language, the algorithm identified 450 abbreviation-definition couple. By reviewing and comparing obtained results and labelled contexts, 444 correct couples were detected of 450 already detected couples, and the algorithm could not identify 17 couples with hand labelling and 39 couples with Moushoitz and Cohen labelling. Table 3, shows the couples which are not identified by algorithm.

Continuing the review and comparison, we found that the algorithm identified six couples wrongly as abbreviation-definition. Table 4 shows these couples.

As the result of performing algorithm on previous golden standard contexts, the algorithm has an excellent result during which found 167 couples that 166 couples of them were correct and only one couple was wrongly detected by the algorithm.

Thus, with this result, golden standard method, with 461 couple by hand labelling we have: TP = 444 (number of correct couples identified by the algorithm), FP = 6 (number of couples that algorithm identified wrongly), and

Table 3
Couples identified by the algorithm in golden standard contexts

Line	*Abbreviation*	*Defination*
1	CI	Confidence Interval
2	22K Hgh	22 Kda Growth Hormone
3	EDI-2	Eating Disorders Inventory
4	TAS20	Toronto Alexithymia Scale
5	2-D	Forced Exoiratory Volume
6	3-D	N-Telopeptides of type 1 collagen
7	RARS/RXRS	Retinoids And Their Multiple Receptors
8	IFN-alpha	Interferon-alfa
9	SDS	Interferon-Alpha
10	GHQ-28	General Health Questionnaire
11	AECP	Antiepiligrin(LaMinin5) CicatricialPemphigoid
12	BHLF	Bam Hi-H, L-Fragment
13	NBS	European Association Of Pathologists
14	C	Chemotherapy
15	U	Unpurged
16	P	Purged
17	VSV-G	Vesicular Stomatitis Virus

Table 4
Couples identified wrongly by the algorithm as abbreviation- definition

Line	*Abbreviation*	*Defination*
1	SV-IV	SV Epithelium
2	anti-Tac	Antibody to the alpha subunit of the IL-2 receptor
3	PR	P = 0.04
4	RT1.Aa	Recipients 1 day prior to heterotopic ACI
5	NO	Nmol l (-1)
6	NO	Nmol l (-1)

FN = 17 (number of correct couples which algorithm couldn't identify them). As a result, according to obtained data, we have:

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{444}{444+6}*100 = \frac{444}{450}*100 \cong 99\%$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{444}{444+17}*100 = \frac{444}{461}*100 = 96.5\%$$

$$F-measure = \frac{2*precision*recall}{precision+recall} = \frac{2*0.99*0.965}{0.99+0.965} = 0.98$$

By use of more formal labelling (483 couples for golden standard) we have: TP = 444, FP = 6 and FN = 15. As a result, according to obtained data, we have:

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{444}{444+6}*100 = \frac{444}{450}*100 \cong 99\%$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{444}{444+39}*100 = \frac{444}{483}*100 = 92\%$$

$$F-measure = \frac{2*precision*recall}{precision+recall} = \frac{2*0.99*0.92}{0.99+0.92} = 0.96$$

To evaluate algorithm in golden standard contexts which Schwartz and Hearst [16] used in their work, TP = 166, and FP = 1, and FN = 2. So we have:

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{166}{166+1}*100 = \frac{166}{167}*100 \cong 99.5\%$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{166}{166+2}*100 = \frac{166}{168}*100 = 99\%$$

$$F-measure = \frac{2*precision*recall}{precision+recall} = \frac{2*0.99.5*0.99}{0.99.5+0.99} = 0.99$$

Precision in new golden standard in both case is 99 %, in previous golden standard is 99/5 % and rerecall in hand labelling 96/5 % and in Moushitz and Cohen labelling 92 % and previous golden standard 99%. As a result, F-measure in hand labelling is equal to 0/98 %, in formal labelling 0/96 % and in old golden standard 0/99 %. Table 5 shows the summary of the results.

Table 5
Algorithm evaluation results' summary

Type of test data	*Total couples*	*Couples identified by the algorithm*	*Couples not identified (FN)*	*TP*	*FP*	*Precision (percent)*	*Rerecall (percent)*	*F-measure*
The new gold standard by labeling manually	461	450	17	444	6	99	96.5	0.98
The new gold standard labeling with Movshovitz-Attias	483	450	39	444	6	99	92	0.96
Gold Standard with 168 couples	168	167	2	166	1	99.5	99	0.99

4.3. Comparing the suggested algorithm with other algorithms

In order to compare algorithm's performance, we first compare the result of algorithm on golden standard with 168 couple, with three algorithms of Schwartz and Hearst [16], Chang et al, [15] and Pustejovsky et al. [11] table 6 shows the result of comparing with three algorithms on golden standard with 168 couples. In order to compare algorithm result on golden standard with 483 couple algorithm, we use two abbreviation identifying algorithm on biomedical contexts: "Automatic Precision Estimates" algorithm [18] and "Alignment HHM" algorithm [8]. Table 7 shows precision factor, rerecall, and F- measure comparison with other algorithms on golden standard biomedical contexts.

Table 6
Comparison with other algorithms on golden standard with 168 abbreviations- biomedical contexts definition.

Algorithm	*Rerecall (present)*	*Precision (present)*	*F-measure*
Chang	80	83	0.82
Pustijovsky	98	72	0.83
Schwartz & Hearst	96	82	0.88
The proposed algorithm	99.5	99	0.99

Table7
Comparison with other algorithms on golden standard with 483 abbreviation- definition couples of biomedical contexts.

Algorithm	*Rerecall (present)*	*Precision (present)*	*F-measure*
Automatic PrecisionEstimates	85	97	0.91
Alignment HMM	93	98	0.96
The proposed algorithm	92	99	0.96

The results of comparing the algorithm with other algorithms shows that the suggested algorithm in golden standard, including 168 abbreviations- definition couple is so much better than other algorithms in terms of precision and rerecall. Also the algorithm in golden standard including 483 abbreviations- definition couples in terms of precision has done very good, it means that the abbreviation- definition couple that the algorithm found is mostly correct and this is why the algorithm follows certain rule and has less wrong identify about abbreviation-definition couples. The algorithm also has done well in terms of rerecallingand the 39 couple which did not identify, mostly did not follow abbreviation- definition rule and rather than previous algorithm has had better performance. The algorithm also gave us good results in terms of F-measure, the result which only Alignment HMM algorithm reached.

5. CONCLUSION AND SUGGESTIONS

In this paper, we presented an alignment algorithm to identify definitions corresponding to abbreviations. In provision for alignment algorithm, we needed to detect abbreviations and definitions of searching space for their definitions. We made a list of candidate's definitions, every item of this list was studied with abbreviations by using alignment algorithm and then the optimized definition was chosen. Because of using some special rules, algorithm shows high precision. Also, the algorithm can find some irregular abbreviations and obtain a good result in biomedical contexts. Next we presented the results of evaluating regulated alignment algorithm for identifying definitions corresponding to the abbreviations. The algorithm was implemented by Java programming language. To evaluate the algorithm,Medstract golden standard contexts (old and new) were used. The results of evaluation showed suggested algorithm than previous methods in terms of precision and rerecall and as a result F-measure has higher performance. This algorithm's main score is its high precision which is the cause of using a regulated method in becoming algorithm.

References

[1] Rajman, M. and romaric, B. (1997),Text Mining, knowledge extraction from unstructured textual data. *Proc. of EUROSTAT Conference, Francfort (Deutchland), may, 1997*.

[2] Hirschman, L. & Blaschke, C. (2006), Evaluation of text Mining in biology. *In Text Mining for Biology and Biomedicine conference*, pp. 213–245.

[3] Yeganova, L., Comeau, DC. & Wilbur, WJ. (2011), Machine learning with naturally labeled data for identifying abbreviation definitions.*BMC Bioinformatics*, Vol. 8, pp. 351-358.

[4] Ao, H. & Takagi, T. (2005), An Algorithm to Extract Abbreviations fromMEDLINE. *J. AM. Med. Inform. Assoc.*,Vol 12, pp. 576- 586.

[5] Dai, H.J., Chang Y.C. & Tsai R.T.H. (2010), New challenges for biological text-Mining in the next decade. *JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY*, Vol. 5, pp. 169-180.

[6] Fred, HL. & Cheng, TO. (2005), Acronymesis: The Exploding misuse of acronyms. *Tex Heart Inst J*, Vol. 3, pp. 251-257.

[7] Cohen, A. &Hersh, W. (2005), A Survey of Current Work in Biomedical Text Mining.*Briefing in Bioinformatics*, Vol. 6, pp. 57-71.

[8] Movshovitz-attias, D. & Cohen, W. (2012), Alignment-HMM-based Extraction of Abbreviations from Biomedical Text. *In Proceedings of the 2012 Workshop on Biomedical Natural Language Processing (BioNLP 2012)* (June 2012), pp. 47-55.

[9] Yu, Hang., kim, Won., Hatzivassiloglou, Vasileios. & Wilbur, W. John. (2007), Using MEDLINE as a knowledge source for disambiguating abbreviations and acronyms in full-text biomedical journal articles.*Journal of Biomedical Informatics*, Vol. 4, pp. 150–159.

[10] Jensen, LJ., Saric, J. & Bork P. (2006), Literature Mining for the biologist: from information retrieval to biological discovery. *Nat. Rev. Gen*,Vol. 7, pp. 119-129.

[11] Pustejovsky, J., Castano, J. & Cochran, B. (2001), Automatic extraction of acronym-meaning pairs from medline databases, *Medinfo*, Vol. 10, pp. 371-375.

[12] Zhou, W., Torvik, VI. &Smalheiser, NR. (2006), ADAM: another database of abbreviations in MEDLINE. *Bioinformatics*, Vol. 12, pp. 2813-2818.

[13] Ananiadou, S. &Chruszcz, J. (2007), The National ventre for text Mining: Aims and objectives. *In Proc. UKKDD2007, Kent, UK, April 25, 2007*, 6-12.

[14] Park, Y. & Byrd, RJ. (2001), Hybrid Text Mining for Finding Abbreviations and Their Definitions.*In Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing: 03-04 June 2001; Pittsburgh*, pp. 126-133.

[15] Chang, JT., Schutze, H. & Altman, RB. (2002), Creating an Online Dictionary of Abbreviations fromMEDLINE. *J. Am. Med. Inform Assoc*, Vol. 9, pp. 612-620.

[16] Schwartz, AS. & Hearst MA. (2003), A Simple Algorithm for IdentifyingAbbreviationDefinitions in Biomedical Text.*In Proceedings of the 8th Pacific Sym-posium on Biocomputing: 03-07 January 2003; Lihue, Hawaii*, pp. 451-462.

[17] Taghva, K. & Gilbreth, J. (1995), Recognizing Acronyms and Their Definitions.*Technical Report, Information Science Research Institute, University of Nevada, January 1995.*

[18] Sohen, Set. (2008). Abbreviation definition identification based on automatic precisionestimates.*BMC Bioinformatics*, Vol. 9:402.