BMC Bioinformatics

RESEARCH Open Access

Machine learning with naturally labeled data for identifying abbreviation definitions

Lana Yeganova*, Donald C Comeau, W John Wilbur

From Machine Learning for Biomedical Literature Analysis and Text Retrieval in the International Conference for Machine Learning and Applications 2010
Washington, DC, USA. 12-14 December 2010

Abstract

Background: The rapid growth of biomedical literature requires accurate text analysis and text processing tools. Detecting abbreviations and identifying their definitions is an important component of such tools. Most existing approaches for the abbreviation definition identification task employ rule-based methods. While achieving high precision, rule-based methods are limited to the rules defined and fail to capture many uncommon definition patterns. Supervised learning techniques, which offer more flexibility in detecting abbreviation definitions, have also been applied to the problem. However, they require manually labeled training data.

Methods: In this work, we develop a machine learning algorithm for abbreviation definition identification in text which makes use of what we term naturally labeled data. Positive training examples are naturally occurring potential abbreviation-definition pairs in text. Negative training examples are generated by randomly mixing potential abbreviations with unrelated potential definitions. The machine learner is trained to distinguish between these two sets of examples. Then, the learned feature weights are used to identify the abbreviation full form. This approach does not require manually labeled training data.

Results: We evaluate the performance of our algorithm on the Ab3P, BIOADI and Medstract corpora. Our system demonstrated results that compare favourably to the existing Ab3P and BIOADI systems. We achieve an F-measure of 91.36% on Ab3P corpus, and an F-measure of 87.13% on BIOADI corpus which are superior to the results reported by Ab3P and BIOADI systems. Moreover, we outperform these systems in terms of recall, which is one of our goals.

Background

Introduction

This research project is a part of the continuous effort at the National Library of Medicine® (NLM) at the National Institutes of Health to improve information access and retrieval from MEDLINE®, a collection of approximately 20 million biomedical journal citations as of August 2010. PubMed® is a search engine, developed and maintained by the National Center for Biotechnology Information at NLM, which provides access to MEDLINE and processes an average of 2 million queries a day.

* Correspondence: yeganova@mail.nih.gov
National Center for Biotechnology Information, NLM, NIH, Bethesda, MD, USA
Full list of author information is available at the end of the article

The size of MEDLINE has doubled within the last decade presenting new challenges to the information retrieval task. One of the challenges is the abundance of abbreviations in text. Chang et al. [1] report that in Medline abstracts, about 64,000 new abbreviations were added in 2004, with the total estimated number of abbreviation occurrences being about 9 million in 2007 as reported by Sohn et al. [2]. Liu et al. [3] estimate that about 81% of abbreviations in MEDLINE are ambiguous with an average of 16.6 senses per abbreviation. After ignoring senses that occur less than 5 times, they find that 65% of abbreviations are still ambiguous with an average of 4.9 senses per abbreviation. They also observe that every sense has on average 7.7 different but equivalent expansions. With such wide presence of acronyms and abbreviations in the fast growing body of biomedical literature, automatic tools for recognizing them have become essential.

Abbreviation identification is the task of processing text to extract explicit occurrences of abbreviation-definition pairs. The task requires both the identification of sentences that contain potential abbreviation-definition pairs, and identification of definition boundaries. Our process starts with extracting sentences that contain parenthetical expressions to identify potential abbreviation-definition pairs. For example, given a sentence

The hydrolysis of lipids in human high-density lipoprotein (HDL) by esterases first separated on a polyvinylidene fluoride membrane...

HDL is identified as a potential abbreviation and *The hydrolysis of lipids in human high-density lipoprotein* as a potential definition. Then the algorithm attempts to identify the definition boundary for a given abbreviation, i.e. recognize that *HDL* stands for *high-density lipoprotein.* We will interchangeably use the term short form (SF) for an abbreviation or acronym, and long form (LF) for its definition. We will also refer to a potential short form as a PSF, to a potential long form as a PLF, and to a potential abbreviation-definition pair as a PSF-PLF pair.

In this study, we use a supervised learning method for the automated abbreviation identification task. We take advantage of the fact that SF-LF pairs naturally occur in text. We extract PSF-PLF pairs from text and treat them as positive examples for learning. Negative examples are artificially constructed by randomly mixing instances of PSFs and PLFs that do not correspond to each other. We refer to these negative examples as the random background. The learner is trained to distinguish between naturally occurring PSF-PLF pairs and the random background. The resulting feature weights are then used to identify the exact abbreviation definition from the PLF.

Literature review

Several approaches have been proposed for automatic acronym extraction, including machine learning-based [1,4,5], rule-based [2,6-8], text alignment [9], statistical [10,11], and various combinations of these.

In 2003, Schwartz and Hearst [9] proposed a simple and fast algorithm that starts at the end of both PSF and PLF and searches backwards to find the shortest LF that matches a SF. A character in a SF can match any character in a LF, except the first character of a SF which must match the initial character of the first word in a LF. They achieved 96% precision and 82% recall on the Medstract corpus. Their algorithm is simple, efficient and does not require any training.

Yu et al. [8] developed a pattern-matching algorithm to map SFs to LFs in biomedical text. Their algorithm extracts all potential LFs that begin with the first letter of the short form and iteratively applies a set of pattern-matching rules on the potential LFs from the shortest to longest, until a LF is found. The pattern matching rules are applied sequentially in pre-defined order. They achieved 95% precision and 70% recall on a small set of biomedical articles.

In 2008, Sohn et al. [2] proposed another pattern-matching approach. Similar to Yu et al. [8] they sequentially applied different strategies attempting to map SF to LF, until a LF is identified. However, they defined a richer set of patterns, which they called strategies. They imposed more relaxed length restrictions and tried to find the best candidate by searching for the most reliable successful strategy out of seventeen strategies that they applied in order. One of the major advantages of their algorithm is that they computed a reliability estimate for each of their strategies. Thus, their algorithm rated the reliability of identified SF-LF pairs without any human judgment. They achieved 97% precision and 85% recall on the Medstract corpus. They also annotated 1,250 randomly selected MEDLINE records as a gold standard, called the Ab3P corpus. On that set they achieved 96.5% precision and 83.2% recall, which compares favorably to the Schwartz and Hearst algorithm.

A common flaw of rule-based and letter alignment algorithms is their inability to identify non-typical pairs, such as *three dimensional (3-D)*, or out-of-order matches, such as *melting temperature (T(m))*. In general, they fail to capture uncommon definition patterns and are limited to the pre-defined rules. Machine learning methods have the potential of recognizing such non-trivial or irregular pairs, if enough training data is provided.

One of the first studies that applied supervised learning to abbreviation definition identification was by Nadeau and Turney [5]. The authors start with the assumption that any token in a sentence is a potential acronym and the LF can be defined as any combination of one or more consecutive tokens from the left context or from the right context. Then, they use weak constraints to reduce the space and thus produce potential SF-LF pairs. They define seventeen features, describing the mapping between acronym and definition characters and employ supervised learning algorithms to identify the abbreviation definition. The algorithm achieves 92.5% precision and 84.4% recall on the Medstract corpus.

Chang et al. [1] used dynamic programming to align SFs with their LFs and to enumerate possible SF-LF candidates. They computed feature vectors from the results of the alignments and used a logistic regression-based learning algorithm to compute feature scores and score alignments. They defined nine features describing the mapping between acronym letters and definition letters. They achieved 80% precision and 83% recall on the Medstract corpus.

Kuo et al. [4] presented yet another machine learning-based system, BIOADI. They defined four sets of features that describe various properties of abbreviation-definition pairs. These features include string morphological features of SF and LF, LF tokens, numeric features that count character usage between a SF and a LF, and contextual features. They annotated 1200 MEDLINE abstracts which were derived from the BioCreative II gene normalization corpus, which they refer to as the BIOADI corpus. Trained on the BIOADI corpus, their system achieved 95.86% precision and 84.64% recall on the Ab3P corpus. When trained on the Ab3P corpus they achieved 93.52% precision and 79.95% recall on the BIOADI corpus.

The common characteristic of the above machine learning approaches is the supervised nature of learning, which requires manually labeled training data. Such data is generally time and labor-intensive. The elegance of our method lies in the fact that while our learner is in principle supervised, it does not require manually labeled training data. Positive examples are simply extracted from the text, and negative examples are artificially generated. Details are presented in the next section.

The paper is organized as follows. In the next section we define features as mappings between characters in the PSF and the PLF, explain the machine learning framework and evaluation measures. In section 3 we describe our experiments and provide the results. In sections 4 and 5 we discuss our approach and draw conclusions.

Methods

In this study, we address two related questions:

1. Can a machine learning method be trained to distinguish between natural PSF-PLF pairs and random pairs?

2. Can we use the feature weights obtained during the above training to identify the correct abbreviation definitions?

We start this section by identifying potential short form (PSF) and potential long form (PLF) pairs. We then define features as mappings between characters in the PSF and the PLF. Naturally occurring PSF-PLF pairs and artificially created unrelated pairs, the random background, are then converted into a feature vector representation and used as input to the model, which is trained to distinguish between these two sets. Then, any PSF-PLF pair is scored as the sum of the weights of the features that describe the mapping between PSF and PLF.

Feature construction

PSF-PLF pairs

We search text for parenthetical expressions to identify PSF-PLF pairs. Following Sohn et al. [2], we require that both SF and LF appear in the same sentence and follow the LF (SF), LF [SF], SF (LF), or SF [LF] patterns. We use the sentence-segmenting function in MedPost to segment title and abstract text into sentences [12] and assume that white space and punctuation marks delineate word boundaries.

In a LF followed by a SF pattern, PSF consists of one or two tokens within parentheses and is limited to at most ten characters in total length. If the text inside parenthesis contains ';' or ',', we treat the text before these punctuation marks as the PSF (e.g., *alpha beta (AB, see reference) - AB* is extracted as the PSF). A PLF consists of all tokens preceding a PSF in the same sentence. In a SF followed by a LF pattern, PSF is the token immediately preceding the left parenthesis containing at least one uppercase letter and PLF is the content within parenthesis.

A PSF must begin with an alphanumeric character and contain at least one alphabetic character. We include single alphabetic characters as PSFs because such abbreviations occur frequently in MEDLINE. Sequence or list indicators (e.g., (a) (b) (c), (i) (ii) (iii), etc.) and common strings ('see', 'e.g.', 'and', 'p<', ...) are recognized and filtered out.

General feature types

Our features are inspired by the basic rules defined by Sohn et al. [2]. Every rule describes a mapping between a character in a PSF and a character in a PLF. They combine these rules to create 17 different strategies that attempt to identify a complete match between abbreviation and definition. We, in contrast, do not combine these basic rules, that we call features, into hand-crafted strategies. We provide the learner with all these features and feature pairs and let the training process weight them. Feature weights are then used to identify abbreviation definitions.

Here we define nine general types of features that describe mappings between characters in PSFs to characters in PLFs. Table 1 explains and illustrates these feature types.

Simple features

Given a PSF-PLF pair, we assign position numbers to tokens in PLF and characters in PSF from right to left. For example, for PSF of length N the rightmost character will be assigned position 1, and the leftmost

Table 1 Description of general feature types with examples

FC	A character of a SF matches the 1st character of a word in a LF	***C**erebrospinal fluid (**C**SF)*
FC-ST	A character of a SF matches the 1st character of a stop-word in a LF	*people **w**ith AIDS coalition (P**W**A)*
FCG	A character of a SF matches the character following a non-alphanumeric non-space character in a LF	*GH-**r**eleasing peptide (GH**R**P)*
SBW	A character of a SF matches a character within a token in a LF such that token splits at that character into two substrings, one or both of which are defined words.	*Cerebro**s**pinal fluid (C**S**F)*
LS	The last character of a SF is 's' and last token in a LF ends in 's' or 'i'	*plasma concentration**s** (PC**s**)*
ALC	A letter of a SF matches a capital non-1st character letter in a LF	*gamma-vinyl**G**ABA (GV**G**)*
ALS	All characters of a SF appear anywhere in a single token in a LF in the correct order	***br**omo**d**eoxy**u**ridine (**BrdU**)*
LT	Look-up table match between a character in a SF and a token in a LF	***Current** (**I**)*
CL	A substring of a SF matches two or more consecutive characters of a token in a LF	*methyl-**beta**-cyclodextrin (M**beta**CD)*

character will be assigned position N. Similarly, for the PLF consisting of M tokens, the rightmost token will be assigned position 1, and the leftmost token position M. The example below illustrates the point.

$$\underset{8}{\underline{The}}\ \underset{7}{\underline{hydrolysis}}\ \underset{6}{\underline{of}}\ \underset{5}{\underline{lipids}}\ \underset{4}{\underline{in}}\ \underset{3}{\underline{human}}\ \underset{2}{\underline{high\text{-}density}}\ \underset{1}{\underline{lipoprotein}}\ \underset{3}{\underline{(H}}\ \underset{2}{\underline{D}}\ \underset{1}{\underline{L)}}$$

Simple features are represented using the general feature type (FT) and positions of matching PLF and PSF characters. A general pattern for simple features is *iFTj*, where *j* is the position of character in PSF, *i* the position of matching token in PLF, and FT stands for one of the nine feature types. In the above example,

The hydrolysis of lipids in human high-density lipoprotein (HDL)

characters H and L in the SF have several allowed matches in LF, while character D has only one allowed match. Simple features generated are:

H: {2FC3, 3FC3, 7FC3}; ***D***: {2FCG2}; ***L***: {5FC1, 1FC1}.

Compound features

To enrich the set of features, we pair single features into compound features. We define two types of compound features. The first type of compound feature links two simple feature types corresponding to two consecutive characters in a PSF. The second type of compound feature links two simple feature types corresponding to characters in a PSF with one character in between.

A general pattern for compound features of type 1, linking two consecutive characters in a PSF, is $FT_j(d)$ FT_{j-1}, where FT_j and FT_{j-1} are general feature types corresponding to characters j and j-1 in a PSF, and d is the distance between matching tokens in a PLF. A general pattern for compound features of type 2 is $FT_j(d)FT_{j-2}\$$. Going back to the *HDL* example, we create compound features linking characters *H&D*, *D&L*, and *H&L*. Note that in compound features we have dropped position-related information.

Compound features type 1:

$$\left.\begin{array}{ll} H: & \text{3FC3} \\ D: & \text{2FCG2} \end{array}\right\} \rightarrow \text{FC(+1)FCG} \qquad \left.\begin{array}{ll} D: & \text{2FCG2} \\ L: & \text{1FC1} \end{array}\right\} \rightarrow \text{FCG(+1)FC}$$

Compound feature type 2:

$$\left.\begin{array}{ll} H: & \text{3FC3} \\ L: & \text{1FC1} \end{array}\right\} \rightarrow \text{FC(+2)FC\$}$$

The number in parenthesis between two feature types in compound features is the distance between matching tokens in a PLF. The distance sign is positive if the matching PLF tokens are in the order from left to right, and negative otherwise. If two PSF characters match different characters in the same token in PLF, the distance is 0.

We distinguish compound features of type 1 and type 2 by appending a special character '$' to the latter. We also prepend a special character '*' to the features that refer to the leftmost character in a PSF, as it has more importance then other characters in a PSF. If the same compound feature appears more than once, we prepend the count to its left end.

Each compound feature captures the transition information between a pair of PLF tokens as we transition through the PSF characters. Unlike simple features, compound features are position independent. It is an important characteristic, because it makes compound features independent of the PSF length.

Clearly, not all pairings of single features are compatible. For example, no two characters in a PSF can match the same character in a PLF. In *American Heart Association (AHA)*, the simple features *1FC3* and *1FC1* are not compatible, because both *A*'s in the SF match the same character *A* at the beginning of the word *Association*. Table 2 illustrates the complete set of features for a PSF-PLF pair.

Machine Learning

After segmenting MEDLINE abstracts into sentences, we identified about 14 million naturally occurring PSF-PLF pairs. We used that set to generate the data for our experiments. We randomly selected 1 million pairs from these 14 million PSF-PLF pairs as our positive training set. We artificially constructed a set of 1 million negative examples by randomly mixing the instances of PSFs and PLFs that do not correspond to each other. We refer to them as the random background. For example:

PSF-PLF pairs:

The operations studied were proximal gastric vagotomy (PGV)

Insulin release in response to alpha-ketooctanoic (KO)

A procedure suitable for preparation of germinal vesicles (GV)

Ecologic studies of Venezuelan encephalitis(VE)

Random Background:

The operations studied were proximal gastric vagotomy (GV)

Insulin release in response to alpha-ketooctanoic (VE)

A procedure suitable for preparation of germinal vesicles (PGV)

Ecologic studies of Venezuelan encephalitis(KO)

Of course, not every PSF-PLF contains an abbreviation and definition pair, for example:

Action of this compound may be its ability to affect (reduce)...

Likewise, it is possible for a random pair to contain mappings from PSF characters to PLF tokens, as in

The operations studied were proximal gastric vagotomy (GV)...,

where *GV* happens to match *gastric vagotomy*. However, we train on large amounts of data, and expect that natural relationships occurring in the PSF-PLF pairs between potential abbreviations and potential definitions will dominate incidental matches in the random background.

In this study, we used the wide margin classifier with modified Huber loss function [13] to learn the difference between the positive and negative sets. During training, the machine learner computes the weights of the features that appear in the training data. Then, we use the resulting feature weights to score possible SF-LF candidates derived from a PSF-PLF pair in an effort to predict the correct definition.

Given the set of all simple features for a PSF-PLF pair, we combine them in all compatible ways, using one feature per PSF character, to create candidate definitions (CD). In the above example

The hydrolysis of lipids in human high-density lipoprotein (HDL)

the character *H* has 3 potential mappings in the LF, character *D* has 1 potential mapping, and character *L* has two. Therefore, we can generate up to *3x1x2=6* different feature combinations. Some of these candidate definitions are shown in the Table 3.

Note that we allow feature combinations that result in definitions where LF tokens are not matched in the order from left to right, as in CD3. Such definitions frequently occur in the literature and by allowing out-of-order matches we add flexibility to our algorithm. However, as we discussed earlier, not all feature combinations generate compatible candidate definitions.

To test a new potential abbreviation-definition pair, all candidate definitions are generated and scored as the sum of simple and compound features that appeared in a combination. Finally, the candidate definition with the highest score is selected as the answer.

To evaluate the ability of the classifier to distinguish between PSF-PLF pairs and random pairs we applied the standard information retrieval measures *Precision*, *Recall* and *F-measure*. Precision is defined as the number of correct pairs retrieved divided by the total number of pairs predicted. Recall is defined as the number of correct pairs identified divided by the total number of

Table 2 Simple and compound features generated for the PSF-PLF pair

	Hydrolysis of lipids in human high-density lipoprotein (HDL)
H	2FC3, 3FC3, 7FC3
D	2FCG2
L	5FC1, 1FC1
H&D	*FC(+0)FCG, *FC(+1)FCG, *FC(+5)FCG
D&L	FCG(-3)FC, FCG(+1)FC
H&L	*FC(-3)FC$, *FC(-2)FC$, *FC(+2)FC$, *FC (+1)FC$, *FC (+6)FC$, *2FC (+2)FC$

Table 3 Simple Features and Corresponding Candidate Definitions

	H	D	L	Candidate Definition
CD1	2FC3	2FCG2	1FC1	***h****igh-****d****ensity* ***l****ipoprotein*
CD2	3FC3	2FCG2	4FC1	***h****uman high-****d****ensity* ***l****ipoprotein*
CD3	2FC3	2FCG2	5FC1	***l****ipids in human* ***h****igh-****d****ensity lipoprotein*

correct pairs. And, F-measure is the harmonic mean of precision and recall.

Summary of the abbreviation definition identification process:

1. Extract naturally occurring PSF-PLF pairs from text and artificially create random pairs;
2. Generate simple and compound features for every potential and random pair;
3. Train a machine learning algorithm to distinguish the two types of pairs;
4. Test a new PSF-PLF pair:

   a. Create and score every candidate definition, by adding up the weights of all simple and compound features that are present in a candidate definition;
   b. Choose the highest scoring candidate definition as the answer;
   c. If no candidate definition is detected or the score of all candidate definitions are lower than a preset threshold, then we predict that PSF-PLF instance is not a SF-LF pair.

Results

The goal of this study is to show that by training PLF-PSF pairs against the random background we are able to

1. Distinguish between the naturally occurring and random PSF-PLF pairs;
2. Given a PLF-PSF pair, use feature weights obtained from training to identify definition boundaries.

To answer the first question we used the training set described above to perform a 3-fold cross validation and observed that we can distinguish positive and negative data with 99.3% average precision and 96.7% break-even score.

To answer the second question, we applied feature weights to identify correct definitions in PSF-PLF pairs. We evaluated the performance of our algorithm, which we refer to as NatLAb (Natural Learning for Abbreviations), on three corpora: Ab3P, BIOADI, and Medstract. The Ab3P gold standard includes 1250 PubMed abstracts and 1221 true SF-LF pairs. The BIOADI gold standard includes 1200 PubMed abstract and 1668 true SF-LF pairs. The Medstract corpus includes 168 PFS-PLF pairs, as annotated by Sohn et al. [2] for their study.

Tables 4, 5, and 6 compare the performance of our NatLAb system to the performance of the Ab3P and BIOADI systems on these corpora where published results are available. The F-measures obtained by our system are comparable to both state-of-the art methods. Moreover we demonstrate an improvement in recall values. We have designed our algorithm to be more flexible than existing systems by allowing out-of-order matches between characters in PSF and PLF and allowing unused characters in PSF. On the Ab3P corpus our Original version achieved a recall value of 87.63% as compared to 83.20% reported by the Ab3P system, and 84.64% reported by the BIOADI system. We also achieve a recall of 82.25% on the BIOADI corpus as compared to 79.95% reported by the BIOADI system. While these modifications clearly benefitted the recall, here are some examples where they harmed the precision.

- *'Vasopressin receptor (V2R)'* instead of *'V2 Vasopressin receptor (V2R)'*
- *'human nicotinic acetylcholine receptor (nAChR)'* instead of *'nicotinic acetylcholine receptor (nAChR)'*
- *'containing histone deacetylase (HDAC)'* instead of *'histone deacetylase (HDAC)'*
- *'c oxidase (COX)'* instead of *'cytochrome c oxidasse(COX)'*

This basic scheme works well, but we have found that we can obtain a slight improvement in performance by revising our training set. We use the feature weights to identify abbreviation definition boundaries in PSF-PLF pairs of the positive training set and use the resulting SF-LF pairs for training. Now the revised positive training set consists of SF-LF pairs corresponding to the PSF-PLF pairs of the original training set. Examples below illustrate the point.

<u>*PSF-PLF pairs of the original Training Set:*</u>

The operations studied were proximal gastric vagotomy (PGV)

Insulin release in response to alpha-ketooctanoic (KO)

Table 4 Comparison of NatLAb with Ab3P system on the Medstract Corpus

Medstract	Precision	Recall	F-measure
NatLAb	93%	95%	94%
Ab3P	97%	85%	91%

Table 5 Comparison of NatLAb and Modified NatLAb with BIOADI and Ab3P on the Ab3P Corpus

Ab3P Corpus	Precision	Recall	F-measure
Modified NatLAb	93.56%	89.27%	91.36%
NatLAb	91.61%	87.63%	89.58%
Ab3P	96.50%	83.20%	89.36%
BIOADI	95.86%	84.64%	89.90%

Table 6 Comparison of NatLAb and Modified NatLAb with BIOADI on the BIOADI Corpus

BIOADI Corpus	Precision	Recall	F-measure
Modified NatLAb	91.93%	82.81%	87.13%
NatLAb	90.74%	82.25%	86.29%
BIOADI	93.52%	79.95%	86.20%

A procedure suitable for preparation of germinal vesicles (GV)
Ecologic studies of Venezuelan encephalitis(VE)
SF-LF pairs of modified Training Set:
proximal gastric vagotomy (PGV)
alpha-ketooctanoic(KO)
germinal vesicles (GV)
Venezuelan encephalitis(VE)

We retrained our model using this modified positive training set and original random background to obtained new feature weights. We then applied these new feature weights to the same three corpora: Ab3P, BIOADI, and Medstract. There was no change in Medstract corpus in terms of either recall or precision. However, we observed an improvement in F-measure for both Ab3P and BIOADI corpora. F-measure increased from 89.58% to 91.36% on Ab3P corpus, and from 86.29% to 87.13% on BIOADI corpus as we transitioned from original training set of PSF-PLF pairs to modified training set of corresponding SF-LF pairs. Tables 5 and 6 compare the performance of our modified system to the performance of the Ab3P and BIOADI systems on these corpora. We think the improvement is due to less noise in the positive training set. For example, now our system is more reluctant to reach back for *'containing histone deacetylase (HDAC)'* as it correctly identifies *'histone deacetylase (HDAC)'*.

Discussion

Identification of abbreviations and their definitions has multiple potential applications at different stages of information retrieval. Abbreviations and their definitions identified at the indexing stage address ambiguity (e.g., *FMF* abbreviates 'Familial Mediterranean Fever' and 'Follicular Mycosis Fungoides', etc.) and variation (e.g. 5-HT, 5HT, serotonin all stand for 5-hydroxytriptamine) issues in text. We can also use abbreviation-definition knowledge to resolve undefined abbreviations in abstracts by identifying their intended sense from the surrounding text. Identifying the intended sense of an abbreviation is useful at the querying stage as well. Islamaj et al. [14] estimate that 16% of all queries to PubMed contain abbreviations and, for better search results, it is essential to disambiguate polysemous abbreviations in queries. We may be able to either predict an abbreviation definition from content words present in the query, or simply suggest definitions to choose from. Finally, automatic tools for identifying abbreviation-definition pairs are regularly used to populate and update databases, ontologies, and dictionaries.

The common characteristic of most abbreviation definition identification systems is that they consider abbreviation definition pairs where either PSF or PLF is enclosed in parentheses, except for the work by Nadeau at al. [5], who relaxed pattern definition constraints. Their system produced more PSF-PLF pairs compared to others, thus offering a potential for improving the recall. However, they mention that in order to achieve performance of the hand-built systems, they had to include the feature 'whether the acronym or definition is between parentheses'. Moreover, that feature turned out to be the third most important feature. That suggests that considering patterns SF (LF) and LF (SF) is rational as it reliably covers most occurrences of abbreviation-definition pairs in text. However, their approach does offer more PSF-PLF pairs, and in future work we would like to test the NatLAb system on such a wider set of potential pairs.

Many PSFs contain special characters, such as "/()[]%-", which we have ignored when identifying definitions. However, special characters provide additional information about the PSF-PLF pair. For example, slash '/' may map to a LF token 'ratio', and '%' frequently maps to the token 'percentage'. In the future, we would like to make an intelligent use of special characters in a PSF.

Another category of examples that are hard to reliably match with their SFs are chemical names. While we do reasonably well on chemical names, a special purpose tool would have several opportunities for improvement. We would like to address that in future work as well.

Conclusions

In this work, we used the idea of naturally labeled data to develop a machine learning approach for identifying abbreviation definitions in text. We automatically extracted naturally co-occurring PSF-PLF pairs from text and treated them as positive examples. We constructed negative examples by randomly mixing unrelated instances of PSFs and PLFs. We applied a classifier to learn the difference between these two classes of examples and used the resulting weights to identify the exact abbreviation definitions in test PSF-PLF pairs. Our system demonstrated results that are comparable to the existing Ab3P and BIOADI systems. We then applied these weights to identify exact abbreviation definitions of pairs in the positive training set. Hence, we obtain SF-LF pairs corresponding to PSF-PLF pairs of original positive training set. We then retrained our classifier using these SF-LF pairs as our positive training set and the original random background as our negative training set. With this new training we outperform the existing systems in terms of both F-measure and recall.

Acknowledgements
This research was supported by the Intramural Research Program of NIH, National Library of Medicine.
This article has been published as part of *BMC Bioinformatics* Volume 12 Supplement 3, 2011: Machine Learning for Biomedical Literature Analysis

and Text Retrieval. The full contents of the supplement are available online at http://www.biomedcentral.com/1471-2105/12?issue=S3.

Authors' contributions

LY, DC, and JW equally contributed into design, development, implementation and evaluation of methods. LY drafted the manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Published: 9 June 2011

References

1. **Creating an online dictionary of abbreviations from MEDLINE..** Chang J, Schutze H, Altman R 2002, 612-620.
2. Sohn S, Comeau D, Kim W, J W: **Abbreviation definition Identification based on automatic precision estimates.** *BMC Bioinformatics* 2008, **9**:402.
3. Liu H, Aronson A, Friedman C: **A Study of Abbreviations in MEDLINE Abstracts.** *Proc AMIA Symp.* 2002, 464-468.
4. Kuo C, Ling M, Lin K, Hsu C: **BIOADI: a machine learning approach to identifying abbreviations and definitions in biological literature.** *BMC Bioinformatics* 2009, **10.**
5. Nadeau D, Turney P: **A supervised learning approach to Acronym Identification.** *Stud Health Technol Inform. 2001;84(Pt 1):371-5.* 2005, 319-329.
6. Park Y, Byrd R: **Hybrid Text Mining for Finding Abbreviations and Their Definitions.** *EMNLP* 2001, 126-133.
7. Pustejovsky J, Castaño J, Cochran B, Kotecki M, Morrell M: **Automatic extraction of acronym-meaning pairs from MEDLINE databases.** *Stud Health Technol Inform.* 2001, **84**:371-375.
8. Yu H, Hripcsak G, Friedman C: **Mapping Abbreviations to Full Forms in Biomedical Articles.** *JAMIA* 2002, **9**:262-272.
9. Schwartz A, Hearst M: **A simple algorithm for identifying abbreviation definitions in biomedical text.** *Pac Symp Biocomput.* 2003, 451-462.
10. Okazaki N, Ananiadou S: **A term Recognition Approach to Acronym Recognition.** *Proceedings of COLING/ACL* 2006, 643-650.
11. Zhou W, Torvik VI, Smalheiser NR: **ADAM: another database of abbreviations in MEDLINE.** *Bioinformatics* 2006, **22**:2813-2818.
12. Smith L, Rindflesch T, Wilbur WJ: **MedPost: A part of speech tagger for biomedical text.** *Bioinformatics* 2004, **20**:2320-2321.
13. Zhang T: **Solving large scale linear prediction problems using stochastic gradient descent algorithms.** *Twenty-first International Conference on Machine learning* Omnipress; 2004, 918-922.
14. Islamaj R, Murray C, Neveol A, Lu Z: **Understanding PubMed user search behavior through log analysis.** *Database (Oxford).* 2009.

doi:10.1186/1471-2105-12-S3-S6
Cite this article as: Yeganova *et al.*: **Machine learning with naturally labeled data for identifying abbreviation definitions.** *BMC Bioinformatics* 2011 **12**(Suppl 3):S6.