Combined SVM-CRFs for Biological Named Entity Recognition with Maximal Bidirectional Squeezing

Fei Zhu[1,2], **Bairong Shen**[1,3,4]*

[1] Center for Systems Biology, Soochow University, Suzhou, Jiangsu, China, [2] School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu, China, [3] Institute of Biomedical Information Engineering, Soochow University, Suzhou, Jiangsu, China, [4] Department of Bioinformatics, Medical College, Soochow University, Suzhou, Jiangsu, China

Abstract

Biological named entity recognition, the identification of biological terms in text, is essential for biomedical information extraction. Machine learning-based approaches have been widely applied in this area. However, the recognition performance of current approaches could still be improved. Our novel approach is to combine support vector machines (SVMs) and conditional random fields (CRFs), which can complement and facilitate each other. During the hybrid process, we use SVM to separate biological terms from non-biological terms, before we use CRFs to determine the types of biological terms, which makes full use of the power of SVM as a binary-class classifier and the data-labeling capacity of CRFs. We then merge the results of SVM and CRFs. To remove any inconsistencies that might result from the merging, we develop a useful algorithm and apply two rules. To ensure biological terms with a maximum length are identified, we propose a maximal bidirectional squeezing approach that finds the longest term. We also add a positive gain to rare events to reinforce their probability and avoid bias. Our approach will also gradually extend the context so more contextual information can be included. We examined the performance of four approaches with GENIA corpus and JNLPBA04 data. The combination of SVM and CRFs improved performance. The macro-precision, macro-recall, and macro-F_1 of the SVM-CRFs hybrid approach surpassed conventional SVM and CRFs. After applying the new algorithms, the macro-F1 reached 91.67% with the GENIA corpus and 84.04% with the JNLPBA04 data.

Citation: Zhu F, Shen B (2012) Combined SVM-CRFs for Biological Named Entity Recognition with Maximal Bidirectional Squeezing. PLoS ONE 7(6): e39230. doi:10.1371/journal.pone.0039230

Editor: Jérémie Bourdon, Université de Nantes, France

Received March 19, 2012; **Accepted** May 21, 2012; **Published** June 26, 2012

Funding: This work was supported by the National Nature Science Foundation of China (31170795, 91029703), the Specialized Research Fund for the Doctoral Program of Higher Education of China (20113201110015), the International S&T Cooperation Program of Suzhou (SH201120), and the National 973 Programs of China (2010CB945600). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: bairong.shen@suda.edu.cn

Introduction

The development of biotechnology is contributing to the rapid growth of the biological literature. For example, PubMed (http://www.ncbi.nlm.nih.gov/pubmed/.), a free resource that is developed and maintained by National Center for Biotechnology Information (NCBI), contains more than 20 million citations of biomedical literature from MEDLINE, life science journals, and online books. The enormous volume of biological literature available provide a massive data resource for researchers, but it also a challenge for mining new information and discovering new knowledge, which has become a very important research subject.

Biological named entity recognition can be regarded as a sequence segmentation problem where each token in a sequence is assigned a biological name label (e.g. PROTEIN, DNA, RNA, CELL-LINE, CELL-TYPE,), which can be used to identify specified biological terms in text [1–2], or label OTHER which represents the term isn't a predefined type of biological one. Biological named entity recognition has a key role in biological text mining. It is fundamental for biological information extraction and mining techniques [2–6], such as biological relation extraction [7–8].

However, it is difficult to correctly identify biological terms in text because they use alphabets, digits, hyphens, and other characters [6,9–12]. Arbitrarily referring to biological terms makes it even harder to conduct automatic recognition using computers. In biological text, biological named entities are usually multi-word phrases and some have prefixes and/or suffixes, which makes it harder to determine the boundaries of terms. Biological terms are also affected by their context. In some cases, a biological term has different meaning among species. As a result, it is difficult for computers to recognize biological terms automatically.

Identifying biological terms from text is very important in bioinformatics. In this study, we propose a novel approach for biological named entity recognition.

Related Work

Biological term recognition is one of the hottest research areas. Many researchers are interesting in mining biomedical terms from text, which is a key step in extracting of knowledge with an overall aim of identifying specific terms, such as genes, proteins, diseases and drugs [1–2].

In general, several methods are used for biological named entity recognition [9,11], i.e., dictionary-based approaches [12], rule-based approaches, and machine learning-based approaches. However, dictionary-based approaches tend to miss undefined terms that are not mentioned in the dictionary [12]. The overall results of dictionary-based approaches rely heavily on a predefined dictionary. There is an enormous number of biological terms and new terms are constantly emerging, so it is impossible to produce a complete dictionary containing all biomedical terms. Therefore, the use of a dictionary can provide the highest precision, but we can also miss many terms. In rule-based biological term recognition systems, the rules used for identifying terms are critical, but there are generally no recognition rules that cover all cases [12]. Machine learning-based approaches train models using a training data set and the models can identify predefined types of terms.

Machine learning approaches are now a mainstream method of named entity recognition. Many algorithms are widely used, such as Bayesian approaches, Hidden Markov Model (HMM) [10], Support Vector Machines (SVM) [13–14], Conditional Random Fields (CRFs) [15–16], and Maximum Entropy (ME) [17–18]. For example, AbGene developed by Tanabe *et al.* [19] has an 85.7% precision rate, 66.7% recall rate, and 76.2% F1 rate when using the Bayesian method with manual post-processing. An HMM-based system designed and implemented by Zhou *et al.* [20] can recognize protein, DNA, RNA, cell-type, and cell-lines from text. Their system has a 72.55% F1 rate. Kazama *et al.* [21] used SVMs to identify protein, DNA, cell-type, cell-line, and lipid, with a 73.6% F1 rate. Tsai *et al.* [22] developed a CRF system to find protein mentions, achieving a 78.4% F1 rate. Lin *et al.* [23] used ME to recognize 23 categories of biological terms with a 72% F1 rate.

However, many methods that perform well in general text do not work as well as expected [20,24–28] because there are many obstacles in biological term recognition. First, a biomedical term may have several different written forms, e.g., epilepsy and falling sickness refer to the same disease, which is a disorder of the central nervous system that is characterized by loss of consciousness and convulsions [29]. Second, an entity can be represented using different types, e.g., cancer can be used to represent a disease as well as a genus of crabs in the family Cancridae. Third, abbreviations of terms, especially arbitrarily referred abbreviations, cause even more ambiguity problems. For example, PC may refer to prostate cancer, phosphatidyl choline, or even a personal computer. Fourth, many biomedical terms are phrases or compound words, or they may have a suffix or prefix. All of these factors make it more difficult for computers to identify biomedical terms automatically [9].

Researchers have applied many methods to improve the performance of machine learning approaches, such as combining different approaches and proposing a hybrid approach, conducting post-processing after machine learning, and adding biomedical domain knowledge to machine leaning-based term identification systems. In this paper, we combined all these methods to raise the precision and recall rate.

Results

We used SVM [6], Stanford CRFs [4] and two SVM-CRF hybrid approaches to identify biological terms from text. One SVM-CRF hybrid approach used SVM to separate biological terms from non-biological terms before using Stanford CRFs to identify the type of the biological term, while the other used SVM-CRFs to recognize biological terms before applying our proposed algorithms to improve the prediction results. The parameters for the SVM [6] and Stanford CRFs [4] used in the tests are listed in Table 1 and Table 2.

In the first round, we tested four approaches using data from the GENIA corpus [5]. The F1 score for the SVM-CRFs combined approach with amendment was better than the other three approaches in five classes and it was close to the best in the remaining classes. Its macro-F1 score was greater than those of the other three approaches. The detailed testing results are shown in Table 3. The macro-precision, macro-recall, and macro-F1 rates for the results are shown in Figure 1.

Table 1. Parameters for SVM in training and testing.

Parameter	Value	Parameter	Value
degree in kernel function	3	C cost of C-SVC	1
gamma in kernel function	1	tolerance of termination criterion	0.001
coef0 in kernel function	0	class weight	1

We use LIB SVM with the following settings in the experiment. The basis function is exp(-gamma*|u-v|^2).
doi:10.1371/journal.pone.0039230.t001

In the second round, we tested four approaches using data from JNLPBA04 [7]. The F1 scores for the two SVM-CRF approaches were better than those of the other approaches. The SVM-CRFs combined approach with amendment had the highest macro-F1 score. The detailed results are shown in Table 4. The macro-precision, macro-recall, and macro-F1 rate results are shown in Figure 2.

Discussion

The results showed that the SVM-CRFs hybrid approach could identify biological terms from text well and they performed better than conventional SVM and CRFs approaches. We found in some cases, that SVM had higher precision but it tended to miss terms and unstable when trained with a small-sized data set. If the positive data are much less than the negative one, its optimal hyper plane will be biased to negative. Moreover, when the number of feature dimensions is much higher than the size of training set, over-fitting is very likely to happen. For example, monocyte macrophage lineage associated surface antigen is a protein term. However, the result by SVM is not correct

monocyte	BG#protein
macrophage	IG#protein
lineage	IG#protein
associated	O
surface	IG#protein
antigen	IG#protein

Table 2. Parameters for CRFs in training and testing.

Parameter	Value	Parameter	Value
maxLeft	1	useDisjunctive	true
useClassFeature	True	useSequences	true
useWord	True	usePrevSequences	true
useNGrams	True	useTypeSeqs	true
noMidNGrams	True	useTypeSeqs2	true
maxNGramLeng	6	useTypeySequences	true
usePrev	True	wordShape	chris2useLC
useNext	True		

We use Stanford CRFs with the following settings in the experiment.
doi:10.1371/journal.pone.0039230.t002

Table 3. Testing results on GENIA data by four approaches.

	Result	SVM	CRFs	SVM-CRFs[1]	SVM-CRFs[2]
DNA	P	100	83.67	87.2	91.52
	R	23.39	74.57	84.83	87.43
	F1	37.91	78.86	86	89.43
RNA	P	100	90.87	89.93	95.02
	R	14.51	97.65	84.52	88.98
	F1	25.34	94.14	87.14	89.43
Cell line	P	100	82.31	91.13	93.24
	R	28.76	77.39	88.91	90.7
	F1	44.76	79.78	90.01	91.95
Cell type	P	35.46	79.61	91.95	93.24
	R	71	81.55	88.91	90.7
	F1	47.3	80.57	90.01	91.95
Protein	P	100	75.11	91.2	82.38
	R	17.35	59.19	86.99	92.92
	F1	29.57	66.2	89.04	87.33
O	P	90.77	82.59	91.33	95.11
	R	68.84	96.52	100	99.97
	F1	78.3	89.02	95.47	97.48

SVM-CRFs[1] refers to the SVM-CRFs without amending and SVM-CRFs[2] is SVM-CRFs with amending. P, R and F1 are precision, recall, and F1 respectively.

doi:10.1371/journal.pone.0039230.t003

where the word associated should be tagged as IG#protein. This error is caused because the number of positive samples of the word "associated" as IG#protein is much less than that of negative ones.

The results showed that although the performance of CRFs was medium, they maintained a balance between precision and recall rate, indicating that this was a stable approach. All the results suggested that combining SVM and CRFs can provide better performance because this hybrid technique was complementary. The basic idea of our approach was to make full use of the power of SVMs as a binary-class classifier, which facilitates data labeling with CRFs. However, SVM and CRFs are the two very different algorithms, so simply combining them could cause inconsistencies. The proposed amendment algorithms were designed to correct any inconsistencies and promote their performance.

Materials and Methods

Materials

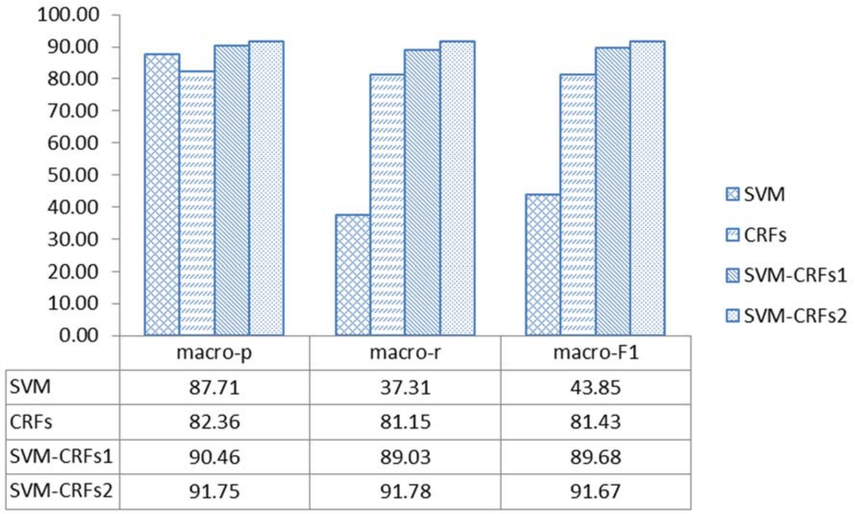
There are many benchmark corpuses for biological named entity recognition, such as the GENIA [5] data set, JNLPBA04 shared task data set [7], GENETAG data set [8], and MEDSTRACT data set [8]. The GENIA corpus was developed for applying natural language processing technology to biological text mining. It contains 2,000 MEDLINE abstracts with more than 400,000 words and almost 100,000 annotations of biological terms [5]. JNLPBA04 [7] has several shared tasks for natural language processing in biomedicine and its application. Bio-entity recognition is one of the tasks of JNLPBA04. The JNLPBA04 data set is often used as a benchmark data set for evaluation methods.

In the first round of testing, we divided data from the GENIA corpus into two parts, i.e., one part for training and the other for testing. We randomly picked 2000 DNA terms, 683 RNA terms, 2000 protein terms, 2000 cell line terms, 2000 cell type terms, and 2000 other types of terms for training. We then selected 400 DNA terms, 166 RNA terms, 400 protein terms, 400 cell line terms, 400 cell type terms, and 400 other types of terms for testing.

In the second round of testing, we randomly selected 2000 DNA terms, 950 RNA terms, 2000 protein terms, 2000 cell line terms, 2000 cell type terms, and 2000 other types of terms from JNLPBA04. We then picked 400 DNA terms, 118 RNA terms, 400 protein terms, 400 cell line terms, 400 cell type terms, and 400 other types of terms for testing.

SVM Terms Identifier

SVM performs well in solving small sample size, nonlinear, and high-dimensional pattern recognition problems and other machine learning problems [30]. Assume that we are given data



	macro-p	macro-r	macro-F1
SVM	87.71	37.31	43.85
CRFs	82.36	81.15	81.43
SVM-CRFs1	90.46	89.03	89.68
SVM-CRFs2	91.75	91.78	91.67

Figure 1. The macro-precision, macro-recall, and macro-F1 rate results using GENIA data with the four approaches. SVM-CRFs[1] refers to SVM-CRFs without amendment while SVM-CRFs[2] is SVM-CRFs with amendment.

doi:10.1371/journal.pone.0039230.g001

Table 4. Testing results on JNLPBA04 data by four approaches.

	Result	SVM	CRFs	SVM-CRFs[1]	SVM-CRFs[2]
DNA	P	100	46.25	74.84	76.80
	R	27.75	92.90	87.20	87.25
	F1	43.44	61.76	81.18	81.69
RNA	P	100	55.84	76.66	78.32
	R	10.94	79.67	86.27	86.49
	F1	19.72	65.66	81.18	82.20
Cell line	P	100	53.69	76.74	79.52
	R	29.82	88.69	95.67	95.01
	F1	45.94	66.56	85.16	86.58
Cell type	P	42.10	52.53	79.12	81.30
	R	78.63	81.88	89.11	89.43
	F1	54.84	64.00	83.81	85.17
Protein	P	100	34.88	71.21	65.06
	R	24.94	69.02	89.72	91.34
	F1	39.90	46.34	79.40	75.99
O	P	93.01	94.15	100	100
	R	72.43	46.36	86.38	86.20
	F1	81.44	62.17	92.69	92.59

SVM- CRFs[1] refers to the SVM-CRFs without amending and SVM-CRFs2 is SVM-CRFs with amending. P, R and F1 are precision, recall, and F_1 respectively.
doi:10.1371/journal.pone.0039230.t004

$D=\{(x_i,y_i)|x_i\in□,y_i\in(-1,1)\}(i=1,\ldots,n)$ where y_i is either 1 or -1, indicating the class of xi. In our previous experiment [31], we used SVM to identify biological terms from text. We used word, word shape, part-of-speech, and morphology as features for identification, as shown in Table 5. The results [31] were good.

SVM uses a line or surface to separate the data [30]. Thus, SVM is suitable for binary classification problems but not multiple-class problems where there are more than two candidate objective classes [32]. In most cases, name entity recognition is a multiple-class task. As a result, the initial binary SVM is not fit for most name entity recognition tasks. We can use two main types of approaches to solve multiple-class problems. One is to update an SVM kernel function that can merge the multiple classification surface problems into an optimization so as to solve multiple class classification in one pass. The alternative is to apply multiple binary classifiers until they finish the job [32].

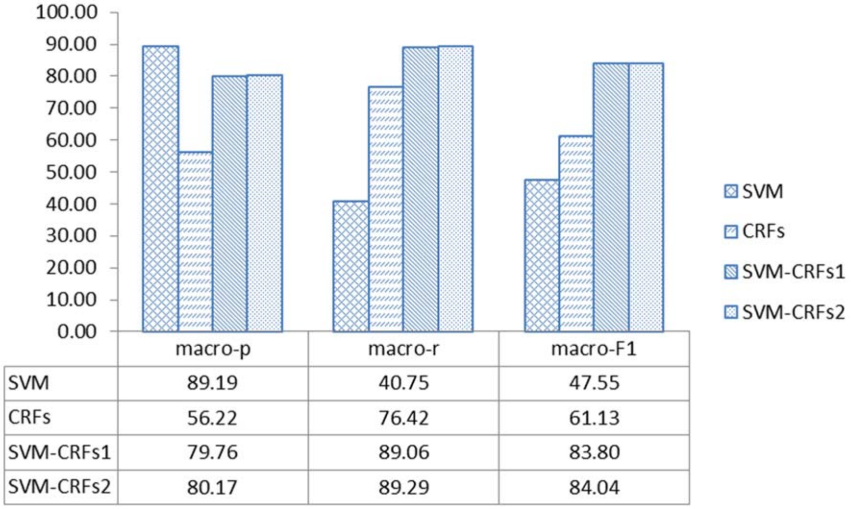
CRFs Terms Identifier

CRFs are often used for the labeling or parsing of sequential data, such as natural language text or biological sequences [33]. CRFs work well in named entity recognition tasks. Many features can be used in CRFs. For example, term appearance (e.g., capitalization, affixes, etc.) and orthographic features (e.g., alphanumeric characters, dashes, Roman numeral characters, etc.) are used frequently.

However, CRFs have many drawbacks. First, CRFs use a limited size of context rather than the whole text because of computational limitation, thereby limiting the contextual information. Second, splitting the context of the whole text into small pieces of context will generally separate inherent relationships among them, and simply combining these pieces of context again cannot reproduce the original context due to the loss of relationships during splitting. For example, a CRF biological term identifier uses a two-word context. The whole text could be split into many pieces of two-word contexts. As a result, the same term in the different places of the text could be tagged with different results due to the variation in the context. However, SVM deals with the whole text so it does not have such restrictions. Third, CRFs are affected by the data distribution. If we want to achieve better results, the data should have an exponential distribution. However, biological terms in texts generally do not meet this data distribution prerequisite.

SVM-CRFs Combined Biological Name Entity Recognition

One of the new research areas in machine learning is combining useful algorithms together to provide better performance or for



	macro-p	macro-r	macro-F1
SVM	89.19	40.75	47.55
CRFs	56.22	76.42	61.13
SVM-CRFs1	79.76	89.06	83.80
SVM-CRFs2	80.17	89.29	84.04

Figure 2. The macro-precision, macro-recall, and macro-F1 rate results using JNLPBA04 data with the four approaches. SVM-CRFs[1] refers to SVM-CRFs without amendment while SVM-CRFs[2] is SVM-CRFs with amendment.
doi:10.1371/journal.pone.0039230.g002

Table 5. Features that are generally used for SVM named entity recognition.

	Features		Features
1	All figures	12	With '%'
2	With figures and letters	13	With ','
3	With capitalized letters	14	With '.'
4	All capitalized letters	15	With ':'
5	First letter is a capitalized letter	16	With '−'
6	First letter is a capitalized letter and followed by '.'	17	Combination of letters and '$'
7	With capitalized letter in the middle of the word	18	Combination of capital letters and '.'
8	All lower-case letters	19	Combination of letters and '.'
9	With two '/'	20	Combination of letters and '−'
10	With one '/'	21	Combination of figures, letters and '/'
11	With '$'		

doi:10.1371/journal.pone.0039230.t005

achieving smooth and stable performance. SVM and CRFs are two conventional algorithms that can deal with named entity recognition tasks well. As stated earlier, the feature context used by SVM is global and it does not have the same constraints as CRFs. SVM is initially the best fit for binary-class tasks and it does not perform well on multiple-class tasks. CRFs generally require more computational time and space than SVMs. Thus, although CRFs have many drawbacks, they are very good at sequential data tagging tasks, which is a typical problem in name entity recognition. Thus, we combined SVM and CRFs because they can complement and facilitate each other.

In our approach, biological named entity recognition was regarded as a two-step task. The first step was to determine whether a candidate term was a biological one. If it was a biological term, we determine its class of entity. The first step was a binary classification task where the result was either yes or no, before we could fully use SVM to complete the task. We then used CRFs to infer the type of biological term. Finally, we merged the results returned by SVM and CRFs, before performing an amendment process.

Inconsistency Removal

In this paper, we used a BIO pattern for the resulting tags: tag that started with the character B began a term; tags starting with the character I represented the intermediate words of a term; while tags starting with the character O indicated that the word was not a biological term. For example, the tag BG#protein shows that the word is the starting word of a protein, while the tag IG#protein is an intermediate word for a protein. Thus, the following words with tags

NOTCH1	BG#protein
ankyrin	IG#protein
repeat	IG#protein
region	IG#protein

can be composed as a complete protein term: *NOTCH1 ankyrin repeat region*.

Given the statement above, we propose a phased approach (Algorithm 1) for determining whether a term is a biological term, as shown in Algorithm 1.

```
**Algorithm** 1 : Biological terms determining
**Input** : $T$ : words to be tagged
**Output** : $T'$ : words with tags
**for all** $t \in T$ **do**
    determine if $t$ is a predefined type of biological word by using SVM
    **if** $t$ is a predefined type of biological word **then**
        $tag \Leftarrow Bio$
    **else**
        $tag_t \Leftarrow O$
    **end if**
**end for**
$T_1 \Leftarrow \{t | t \in T \wedge tag_t \Leftarrow O\}$
$T_2 \Leftarrow \{t | t \in T \wedge tag_t \Leftarrow Bio\}$
**for all** $t \in T_2$ do
    determine which type of biological word by using CRFs
$tag_t \Leftarrow$ the type of biological word
**end for**
$T' \Leftarrow T_1 \cup T_2$
**return** $T'$
```

Algorithm 1 determined whether a term was a biological one. The input was the word set of all terms. The output was words with the tag *Bio* showing that the word was part of a biological term or the tag *O* showing that the word was not a biological term. Words tagged with *Bio* are further processed by CRFs to determine their biological classes.

However, SVM and CRFs are two different algorithms. Simply merging the results returned by SVM and CRFs could cause inconsistency. For example, the term *CsA treated cell* is a cell line mention. Its correct tag should be

CsA	BG#cell_line
treated	IG#cell_line
cells	IG#cell_line.

The SVM identifier predicted the word *CsA* and word *cells* as biological words, but the word *treated* was predicted as a non-biological term. The final results of the SVM and CRFs are

CsA	BG#cell_line
treated	O
cells	IG#cell_line

Therefore, we needed to amend any inconsistencies to improve the results. Before the amendment, we determined which terms were inconsistent. We use the following two rules to identify inconsistent terms:

- Rule 1: If the precursor and the successor of a word are both middle words of a biological term, the word should be also a middle word of the term.
- Rule 2: A term begins with a word tagged with a start tag.

Rule 1 and Rule 2 removed any inconsistencies caused by shifts in context. We used Algorithm 2 to carry out the term consistency analysis, as shown as follows.

```
Algorithm 2 : Biological terms inconsistence analyzing
Input : $T$ merged results inferred by SVM and CRFs
Output : a pending terms list containing inconsistent terms
pending⇐false
for $i=1$ to $\|T\|$ do
    if tag$_{t_1}$ starts with $B$ and $pending = true$ then
        $end\_pos \Leftarrow i-1$
        Add $word_{start\_pos...end\_pos}$ to pending terms list
    pending⇐false
    end if
if $tag_{t_1}$ starts with $I$ and $pending = true$ then
vend_pos⇐$i-1$
start_pos⇐$i-1$
      Add $word_{start\_pos...end\_pos}$ to pending terms list
  else if $pending = false$ then
pending⇐true
start_pos⇐$i$
   end if
   if $tag_{t_1}$ starts with $O$ and $pending = true$ then
      $end\_pos \Leftarrow i-1$
      Add $word_{start\_pos...end\_pos}$ to pending terms list
      pending⇐false
   end if
end for
return pending terms list
```

Algorithm 2 determined word inconsistency of a term by merging the results of SVM and CRFs, and returning a pending inconsistent terms list.

Term Length Maximizing

Using Rule 1 and Rule 2, we can identify and eliminate inconsistencies. In the example, the prediction results for the term *CsA treated cell*

CsA	BG#cell_line
treated	O
cells	O

will be treated as correct, although the results are not exactly the best fit. Thus, we propose a new rule to address this type of inconsistency.

- Rule 3: The length of a biological term is expected to be as long as possible.

According to Rule 3, biological terms should be as long as possible. Using our approach, we extend a term from left to right to validate whether the extended terms are biological terms. Thus, given $word_0 word_1,\ldots,word_{n-1} wordn$, if $word_1,\ldots,word_{n-1}$ is tagged as a biological term, we have to check:

if $word_0 word_1,\ldots,word_{n-1}$ is a biological term;

if $word_1,\ldots,word_{n-1} wordn$ is a biological term;

if $word_0 word_1,\ldots,word_{n-1} wordn$ is a biological term.

If any of the extended terms are in a biological term list, it is definitely a biological term. However, it is impossible to produce a complete biological term dictionary. Therefore, we need to make some deductions to predict the tags of the extended word.

We used a maximal forward and backward probability squeezing approach to extend the term. The maximal forward probability approach determines each forward output probability of state t on the basis of state t−1, while the maximal backward probability determines each backward output probability of state t on the basis of the state t+1 [34]. Our approach identifies the output with the maximal product result for the forward probability and the backward probability.

We assume an output sequence $O_1,\ldots,O_T(T>t)$and a hidden state sequence $X_1,\ldots,X_T(T>t)$. Let a_{ij} be the transfer probability from state t−1 to state t, while α_{t-1} is the probability of observing all of the given data up to state t−1. At state t−1, given an output sequence $O_1,\ldots,O_T$ and a hidden state X_{t-1}, we can find the forward output using the following equations [34].

$$\alpha_{t-1(k)} \equiv p(O_1,\ldots,O_n,X_{t-1}) \qquad (7)$$

$$\alpha_{t(k)} = \sum_{i=1}^{n} \left(\alpha_{t-1(i)} * a_{ik}\right) \qquad (8)$$

Let, b_{kj} be the output probability from state t to state t +1 and α_{t+1} be the probability of all future data from state t +1 to state t. At state t+1, given output sequence $O_1,\ldots,O_T$ and hidden state X_{t+1}, we can conduct inference and find the backward output using the following equations [34].

$$\alpha_{t+1(k)} \equiv p(O_{t+1},\ldots,O_T,X_{t+1}) \tag{9}$$

$$\alpha_{t(k)} = \sum_{i=1}^{n} \left(\alpha_{t+1(i)} * b_{ki}\right) \tag{10}$$

The final result maximizes the product of the result returned by forward inference and backward inference, as shown in the following equation. An illustration of maximal forward and backward probability squeezing is shown in Figure 3.

$$p(O|X) = \arg\max_{\substack{i=1,\ldots,n\\ j=1,\ldots,n}} \left(\sum_{i=1}^{n} \left(\alpha_{t-1(j)} * a_{jk}\right) * \sum_{j=1}^{n} \left(\alpha_{t+1(i)} * b_{ki}\right) \right) \tag{11}$$

The maximal bidirectional probability squeezing method that uses the forward probability and backward probability to predict the outputs of intermediate states tends to lead to bias when dealing with states that are rare. Thus, we add positive gain to rare event cases to reinforce their probability and avoid bias, as shown in Algorithm 3.

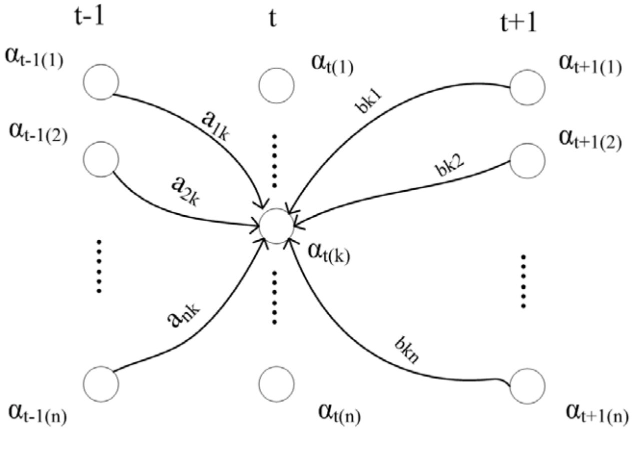


Figure 3. Forward and backward probability squeezing takes the product of the probability obtained by forward inference and the probability obtained by backword inference. Here $a_{ij}(1\leq i\leq n, 1\leq j\leq n)$ and $b_{kj}(1\leq k\leq n, 1\leq j\leq n)$ are the transfer probability, while $\alpha_{t(i)}(1\leq i\leq n)$ is the probability of taking O_i.
doi:10.1371/journal.pone.0039230.g003

Algorithm 3 : Getting terms with maximal positive gain

Input : biological term $w_{left}\ldots w_{right}$

Output : maximal matching term as well as tags

```
$i \Leftarrow left$
$j \Leftarrow right$
$left\_ext \Leftarrow 0$
$right\_ext \Leftarrow 0$
$final\_left \Leftarrow left$
$final\_right \Leftarrow right$
while $w_{left}\ldots w_{right}$ is a positive instance do
    $left \Leftarrow left-1$
end while
$left\_ext \Leftarrow i-left$
while $w_{left}\ldots w_{right}$ is a positive instance do
    $right \Leftarrow right+1$
end while
$right\_ext \Leftarrow right-j$
$max\_total\_ext \Leftarrow left\_ext+right\_ext$
$final\_left \Leftarrow left$
$final\_right \Leftarrow right$
$left \Leftarrow i$
$right \Leftarrow j$
while $w_{left}\ldots w_{right}$ is a positive instance do
    $right \Leftarrow right+1$
end while
$right\_ext \Leftarrow right-j$
while $w_{left}\ldots w_{right}$ is a positive instance do
    $left \Leftarrow left-1$
end while
$left\_ext \Leftarrow i-left$
if $(left\_ext+right\_ext) > max\_total\_ext$ then
    $final\_left \Leftarrow left$
    $final\_right \Leftarrow right$
end if
return $w_{final-left}\ldots w_{final\_right}$ and $O_{final-left}\ldots O_{final\_right}$
```

Algorithm 3 adds positive gain to rare cases to reinforce their probability and avoid bias.

We also maintain the context window as large as possible, so the output has the maximal positive gain, as shown in Algorithm 4.

```
Algorithm 4 : Maximal probability squeezing with positive gain
Input : tag $O_{left} \ldots O_{right}$ of biological term $w_{left} \ldots w_{right}$
Output : resulting tag of $w_k$
$sum(\alpha) \Leftarrow 0$
$sum(\beta) \Leftarrow 0$
for $i=1$ to types of resulting output do
    for $j=1$ to types of resulting output do
        $sum(\alpha) \Leftarrow sum(\alpha) + \alpha_{t(i)} * a_{ij}$
        $um(\beta) \Leftarrow sum(\beta) + \beta_{t+1(i)} * b_{ij}$
    end for
end for
$p\_max \Leftarrow 0$
$p\_cur \Leftarrow 0$
$k \Leftarrow -1$
$positive \Leftarrow false$
for $i=1$ to types of resulting output do
    $p\_cuur \Leftarrow sum(\alpha) + sum(\beta)$
    if $w_{left} \ldots w_{right}$ is part of a term with maximal
        positive gain then
        $postive \Leftarrow true$
        $p\_cur \Leftarrow p\_cur + \Delta$
    end if
    if $p\_max < p\_cur$ then
        $p\_max \Leftarrow p\_cur$
        $k \Leftarrow i$
        $positive \Leftarrow false$
    end if
end for
if $postive = true$ then
    return output of $O_k$ from $O_1 \ldots O_n$
else
    return output of type $k$
end if
```

Algorithm 4 is maximal bidirectional probability squeezing, which uses the forward probability and backward probability to predict the output. Algorithm 4 also maintains a maximal context window so the output has the maximal positive gain.

When we use Rule 3 to maximize the term length, we gradually extend the context window size. We initially set the context window size for the tag $O_tO_{t+1},\ldots,O_T(T>t)$as 3. The sequence piece of the context window will then be $O_{t-1}O_tO_{t+1}$, while the pending sequence is extended to $O_{t-1}O_tO_{t+1},\ldots,O_T$. We take the piece $O_{t-1}O_tO_{t+1}$and use Algorithm 4 to infer the resulting tag O_t. We then judge whether it is correct using Algorithm 2. If correct, the output of the sequence will be revised, but otherwise the context window will be extended left one step and right one step, making it $O_{t-2}O_{t-1}O_tO_{t+1}O_{t+2}$. The pending sequence will also be extended to $O_{t-1}O_tO_{t+1},\ldots,O_TO_{T+1}$. We then determine the state of O_t using Algorithm 4 with the context window $O_{t-2}O_{t-1}O_tO_{t+1}O_{t+2}$. This is conducted iteratively until the predictive tag result is correct according to Algorithm 2 or we still cannot find the correct result after the whole output sequence has been treated. The amendment of the output sequence in various contexts is performed using Algorithm 5

```
Algorithm 5 : Amending with variable context
Input : resulting tag $O_tO_{t+1} \ldots O_T$ to be amended
Output : revised resulting tag $O_tO_{t+1} \ldots O_T$
$context \Leftarrow null$
$i \Leftarrow 0$
while $i < t$ do
    $i \Leftarrow i+1$
    $context \Leftarrow O_{t-1} \ldots O_t \ldots O_{t+1}$
    get output of $O_t$by Algorithm 4
    if $O_tO_{t+1} \ldots O_T$ is predicted to be correct by Algorithm
        2 then
        revise $O_tO_{t+1} \ldots O_T$
    end if
end while
```

Algorithm 5 ensures that the results in context will be adaptively extended gradually.

Performance Evaluation

We evaluate the results in terms of precision, recall rate, and F_1 rate. Precision, recall rate, and F_1 are given by the following equations [3].

$$precision = \frac{number\ of\ true\ positive}{number\ of\ true\ positive + number\ of\ false\ positive} \quad (1)$$

$$recall = \frac{number\ of\ true\ positive}{number\ of\ true\ positive + number\ of\ false\ negative} \quad (2)$$

$$F1 = \frac{2 * precision * recall}{precision + recall} \quad (3)$$

For example, when we identify a protein term, the definition of true positive, false positive, true negative, and false negative are regarded as:

True positive: protein term correctly identified as protein.

False positive: non-protein term incorrectly identified as protein.

True negative: non-protein term correctly identified as non-protein.

False negative: protein term incorrectly identified as non-protein.

We also used macro-precision, macro-recall and macro-F$_1$, to evaluate the overall performance of the identifiers. Their definitions are as follows [3]:

$$Macro-\mathrm{Pr}\,ecision=\frac{\sum_{i=1}^{n} precision_i}{n} \tag{4}$$

$$Macro-\mathrm{Re}call=\frac{\sum_{i=1}^{n} recall_i}{n} \tag{5}$$

$$Macro-F1=\frac{\sum_{i=1}^{n} F1_i}{n} \tag{6}$$

Conclusions

The vast biological literatures provide a highly reliable information source for biological research. Mining information and finding new knowledge is a very important new subject, where the identification of biological terms is fundamental. We propose a novel machine learning approach to achieve biological named entity recognition. This approach used an SVM to determine whether the term is a biological term, before CRFs were used to infer the type of a biological term. We then judged whether the merged result was consistent in the new global context and applied an amendment approach that used maximal bidirectional squeezing with positive gain in an adaptive context algorithm for correcting inconsistent terms. The results showed that our approach could achieve biological named entity recognition and it performed better than CRFs and SVM alone.

Acknowledgments

LIBSVM and the Stanford Named Entity Recognizer used in this work are highly appreciated.

Author Contributions

Conceived and designed the experiments: BS. Performed the experiments: FZ. Analyzed the data: FZ. Contributed reagents/materials/analysis tools: FZ BS. Wrote the paper: FZ BS.

References

1. Jensen LJ, Saric J, Bork P (2006) Literature mining for the biologist: from information retrieval to biological discovery. Nat Rev Genet 7: 119–129.
2. Dai HJ, Chang YC, Tsai TH, Hsu WL (2010) New Challenges for Biological Text-Mining in the Next Decade. Journal of Computer Science and Technology 25: 169–179.
3. Hjfirland B (2010) The foundation of the concept of relevance. Journal of the American Society for Information Science and Technology 61: 217–237.
4. Rose J, Finkel TG, Manning C (2005) Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling.Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005). 363–370.
5. Kim JD, Ohta T, Tateisi Y, Tsujii J (2003) GENIA corpus–semantically annotated corpus for bio-textmining. Bioinformatics 19 Suppl 1: i180–182.
6. Chung C, Lin CJ (2011) LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2: 27:21–27:27.
7. Kim JD, Ohta T, Tsuruoka Y, Tateisi Y, Collier N (2004) Introduction to the Bio-Entity Recognition Task at JNLPBA. Proceedings of the International Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-04). 70–75.
8. Tanabe L, Xie N, Thom LH, Matten W, Wilbur WJ (2005) GENETAG: a tagged corpus for gene/protein named entity recognition. BMC Bioinformatics 6 Suppl 1: S3.
9. Cohen AM, Hersh WR (2005) A survey of current work in biomedical text mining. Brief Bioinform 6: 57–71.
10. Ephraim Y, Merhav N (2002) Hidden Markov processes. IEEE Trans Inform Theory 48: 1518–1569.
11. Li L, Zhou R, Huang D (2009) Two-phase biomedical named entity recognition using CRFs. Comput Biol Chem 33: 334–338.
12. Rebholz-Schuhmann D, Yepes AJ, Li C, Kafkas S, Lewin I, et al. (2011) Assessment of NER solutions against the first and second CALBC Silver Standard Corpus. J Biomed Semantics 2 Suppl 5: S11:1–12.
13. Habib MS, Kalita J (2010) Scalable biomedical Named Entity Recognition: investigation of a database-supported SVM approach. Int J Bioinform Res Appl 6: 191–208.
14. Tsochantaridis I, Hofmann T, Thorsten J, Altun Y (2004) Support vector machine learning for interdependent and structured output spaces Proceedings of the Twenty-First International Conference on Machine Learning (ICML 2004):104–112.
15. He Y, Kayaalp M (2008) Biological entity recognition with conditional random fields. AMIA Annu Symp Proc: 293–297.
16. Wallach HM (2004) Conditional random fields: An introduction. Technical report MS-CIS-04-21, University of Pennsylvania.
17. Jaynes ET (1957) Information Theory and Statistical Mechanics. Physical Review Series II 106: 620–630.
18. Jaynes ET (1968) Prior Probabilities. IEEE Transactions on Systems Science and Cybernetics 4: 227–241.
19. Tanabe L, Wilbur WJ (2002) Tagging gene and protein names in biomedical text. Bioinformatics 18: 1124–1132.
20. Zhou G, Zhang J, Su J, Shen D, Tan C (2004) Recognizing names in biomedical texts: a machine learning approach. Bioinformatics 20: 1178–1190.
21. Kazama Ji, Makino T, Ohta Y, Tsujii J (2002) Tuning Support Vector Machines for Biomedical Named Entity Recognition. Morristown, NJ, USA. Association for Computational Linguistics.
22. Tsai TH, Chou WC, Wu SH, Sung TY, Hsiang J, et al. (2006) Integrating linguistic knowledge into a conditional random fieldframework to identify biomedical named entities. Expert Systems with Applications 30: 117–128.
23. Lin YF, Tsai TH, Chou WC, Wu KP, Sung TY, et al. (2004) A maximum entropy approach to biomedical named entity recognition. 4th Workshop on Data Mining in Bioinformatics:56–61.
24. Ananiadou S, Sullivan D, Black W, Levow GA, Gillespie JJ, et al. (2011) Named entity recognition for bacterial Type IV secretion systems. PLoS One 6: e14780.
25. Kuo CJ, Ling MH, Lin KT, Hsu CN (2009) BIOADI: a machine learning approach to identifying abbreviations and definitions in biological literature. BMC Bioinformatics 10 Suppl 15: S7: 1–10.
26. Li Y, Lin H, Yang Z (2009) Incorporating rich background knowledge for gene named entity classification and recognition. BMC Bioinformatics 10: 223: 1–15.
27. Tsai RT, Lai PT, Dai HJ, Huang CH, Bow YY, et al. (2009) HypertenGene: extracting key hypertension genes from biomedical literature with position and automatically-generated template features. BMC Bioinformatics 10 Suppl 15: S9: 1–11.
28. Winnenburg R, Plake C, Schroeder M (2009) Improved mutation tagging with gene identifiers applied to membrane protein stability prediction. BMC Bioinformatics 10 Suppl 8: S3: 1–11.
29. Dagar A, Chandra PS, Chaudhary K, Avnish C, Bal CS, et al. (2011) Epilepsy Surgery in a Pediatric Population: A Retrospective Study of 129 Children from a Tertiary Care Hospital in a Developing Country along with Assessment of Quality of Life. Pediatr Neurosurg. 47:186–193.
30. Cortes C, Vapnik V (1995) Support-vector networks. Machine Learning 20: 273–297.
31. Ju Z, Zhou M, Zhu F (2011) Identifying Biological Terms from Text by Support Vector Machine. 6th IEEE Conference onIndustrial Electronics and Applications Beijing. 455–458.
32. Hsu CW, Lin CJ (2002) A Comparison of Methods for Multiclass Support Vector Machines. IEEE Transactions on Neural Networks.
33. Lafferty J, McCallum A, Pereira F (2001) Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001): 282–289.
34. Bishop CM (2007) Pattern Recognition And Machine Learning: Springer:393–418.