Int. J. Mach. Learn. & Cyber. (2018) 9:373–382
https://doi.org/10.1007/s13042-015-0426-6

ORIGINAL ARTICLE

A comparative study for biomedical named entity recognition

Xu Wang[1] · **Chen Yang**[2] · **Renchu Guan**[1]

Received: 31 March 2015 / Accepted: 7 September 2015 / Published online: 15 September 2015

Abstract With high-throughput technologies applied in biomedical research, the quantity of biomedical literatures grows exponentially. It becomes more and more important to quickly as well as accurately extract knowledge from manuscripts, especially in the era of big data. Named entity recognition (NER), aiming at identifying chunks of text that refers to specific entities, is essentially the initial step for information extraction. In this paper, we will review the three models of biomedical NER and two famous machine learning methods, Hidden Markov Model and Conditional Random Fields, which have been widely applied in bioinformatics. Based on these two methods, six excellent biomedical NER tools are compared in terms of programming language, feature sets, underlying mathematical methods, post-processing techniques and flowcharts. Experimental results of these tools against two widely used corpora, GENETAG and JNLPBA, are conducted. The comparison varies from different entity types to the overall performance. Furthermore, we put forward suggestions about the selection of Bio-NER tools for different applications.

Keywords Biomedical named entity recognition · Machine learning · HMM · CRF

✉ Renchu Guan
guanrenchu@jlu.edu.cn

[1] College of Computer Science and Technology, Jilin University, 2699 Qianjin Street, Changchun 130012, People's Republic of China

[2] College of Earth Sciences, Jilin University, 2699 Qianjin Street, Changchun 130012, People's Republic of China

1 Introduction

With the widespread application of high-throughput techniques and the burst of gene and protein analysis, the number of biomedical literatures is growing at an exponential speed. Moreover, benefiting from the Open Access, collections of manuscripts, ranging from very general and highly distributed ones to very specific and localized ones, are publicly available. For instance, the PubMed Central literature database contains over 3.3 million references to full-text journal papers, covering a wide range of biomedical fields. Owing to the large number of literatures, it is of great difficulty for biologists to keep up with the new development of this research area, even in a very specialized area such as gene regulation and protein structure prediction. Therefore, effective management of large amount of information and the accurate knowledge extraction from large volume literatures becomes much more vital. Considering manual annotation with biomedical experts is time-consuming and expensive, it is urgent to develop an automatic text mining method, which may help the biologists and doctors to well organize and structure these materials.

As one of the fundamental biomedical text mining tasks, Named Entity Recognition (NER), aiming at identifying chunks of text referring to specific entities of interest, plays a key role in disease-treatment relation extraction [1], gene function identification [2] and semantic relation extraction between concepts in a molecular biology ontology [3]. In general domain, such as newswire domain, the task of named entity recognition is to recognize the name of places, persons, organizations [4]. However, in biomedical domain, biologists and doctors pay much more attention to the entities like genes, proteins, DNA, RNA and so on. Recently, several attempts have been performed to

transform existing named entity recognition systems in general domain into biomedical area [5–8]. However, due to the non-standard nomenclature in biomedical research, few of them achieved satisfactory performance, thus biomedical named entity recognition (Bio-NER) continues to be a challenging task.

Comparisons of naming conventions between biomedical and newswire domain has already been discussed in [9–14], which is summarized as follows: (1) naming an entity descriptively raises great difficulty to identify the entity names' boundaries. For example, "specific immunoglobulin E" or "immediate-early gene" is named with multiple words. Zhou et al. found that nearly 18.6 % of biomedical entity names in the GENIA V3.0 corpus contained at least four words [9]. Figure 1 depicts the results. (2) There are conjunctions and disjunctions. Two or more entity names may share the same prefix noun by using conjunction or disjunction; for example, "mouse and human U6 DNA" indicates two entity names, which are "mouse U6 DNA" and "human U6 DNA", respectively. In GENIA V3.0, about 2.06 % of biomedical entity names fall into this form [9]. (3) There are no strict naming conventions in biomedical literatures. The capitalization and hyphen are to some extent casually used, e.g. Cholesterol, 5-Cholesten-3beta-ol and (3beta)-cholest-5-en-3-ol is the same chemical substance. The non-standardized names may result in low Recall and coverage of dictionaries [10, 11]. (4) Massive amount of abbreviations. Plenty of entities in biomedical domain have abbreviated names. The abbreviations could lead to ambiguity, which makes it difficult to classify them against the existing dictionary. For example, 'TCF' may refer to 'Tcell Factor' or 'Tissue Culture Fluid' in different articles. Chang et al. have shown that, in MEDLINE abstracts, 42.8 % of abstracts have at least one abbreviation and 23.7 % of abstracts have two or more [12]. Liu et al. showed that 81.2 % of the abbreviations are blurred and each abbreviation has 16.6 senses in MEDLINE abstracts on average [13]. (5) Cascaded construction. It is common to find that one biomedical entity name is embedded in another entity name. In [9], Zhou et al. pointed out that 16.57 % of biomedical entity names have such cascaded construction in GENIA V3.0.

In short, the entity names in biomedical domain are much more complex than those in the general domain (such as newswire). However, it is a crucial step to explore more evidential features and effective methods to extract knowledge from biomedical literatures. In this paper, an introduction to three fundamental models of the Bio-NER, which are Dictionary-based, Rule-based and Machine Learning based models, is firstly given. We then introduce six effective Bio-NER tools, drawing a comparison between programming languages, features used, underlying models and post-processing techniques etc. Subsequently, we present the corpora used, the evaluation criteria and the results. Ground on analysis, we finally put forward the suggestions for biologist, doctors and computer scientists about the selection of Bio-NER tools.

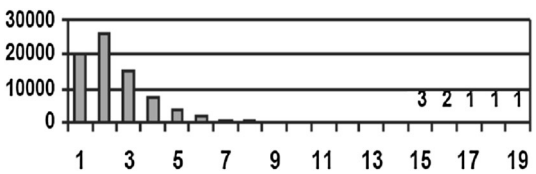

Fig. 1 Distribution of the number of words in biomedical entity names (GENIA V3.0) [9]

2 Biomedical named entity recognition models

Due to the complex naming conventions and its priority in biomedical domain, several Biomedical Named Entity Recognition (Bio-NER) systems have been developed to recognize the entities in biomedical texts. The models used in these Bio-NER systems fall into three categories, they are Rule-based methods, Dictionary-based methods and Machine Learning based methods.

2.1 Dictionary-based methods

Dictionaries are large collections of names, serving as entries for a specific entity class. Matching entries exactly against text is simple and precise, but it gives the way to low recall. To solve this problem, the user can either use incomplete matching techniques, or fuzzify the dictionary by generating typical spelling alternatives for each entry automatically. Compared with the rapid increasing amount of biomedical literatures, it becomes impossible to construct a dictionary that can cover all categories of different entities. Thus, it is impractical to imply dictionary-based methods to achieve high F-Score. However, dictionary-based methods can be integrated with other Bio-NER tools, which can improve the accuracy of the hybrid algorithm. For example, Tsuruoka and Tsujii et al. annotated proteins in GENIA V3.01 with a combination of dictionary and Naive Bayes, achieving an F-score of 66.6 % [15]. Yang et al. improves the recognition performance through the bio-entity name dictionary expansion, including Pre-keyword and Post-keyword expansion, POS expansion, merge of adjacent bio-entity names and the exploitation of the contextual cues [16].

2.2 Rule-based methods

Rule-based model employs plenty of rules to separate different classes. Handcrafted rules are used to describe the

composition of named entities and their context in early rule-based systems. For example, Fukuda et al. employed surface clues (capital letters, symbols, digits) to extract candidates for protein names [10]. Though these rule-based methods seemed promising initially, they failed to perform on larger datasets. For example when Proux et al. evaluated their performance on a larger corpus of 25,000 MEDLINE abstracts by sampling, the precision fell to 70 % [17]. Moreover, it is impossible for these systems to identify new named entities that never discovered before and cost a lot to discover new classes of entity.

2.3 Machine learning based methods

Machine Learning based Bio-NER model integrates various complex steps to incorporate different processing procedures, it performs better than the other two genre solutions [14]. With machine learning based algorithms, researchers do not have to compose the complex rules manually. In addition, these algorithms can also identify new named entities and classes excluded in standard dictionaries. In [11], Nobata et al. first experimented with three identification and two classification methods to recognize ten entity classes, including protein, DNA, RNA, cell type etc., achieving an F-measure between 58.98 and 66.24 % on 100 annotated MEDLINE abstracts using decision trees. Most of the machine learning models can be generally categorized as based on Supported Vector Machine (SVM), Hidden Markov Model (HMM) or Conditional Random Fields (CRFs). Considering the time-consuming of SVM, the SVM-based tools are not compared in this paper although they perform very well in classification and regression.

2.3.1 HMM based methods

Hidden Markov Model (HMM) is a generative type of sequence-based model [18]. Suppose x refers to the input token sequence and y is the output tag sequence. Generative models find the best tag sequence by computing the probability $p(x, y)$. In HMM model, the probability of $p(y|x)$ can be represented as a calculation utilizing its generative form $p(x, y)$ according to the Bayes rules:

$$p(y|x) = \frac{p(x,y)}{p(x)} \tag{1}$$

Assuming the current tag y_i depends on the previous tag y_{i-1}, and the current token x_i depends on the current tag y_i, then $p(x, y)$ turns to be:

$$p(x,y) = \prod_{i=1}^{n} p(y_{i-1}|y_i) * \prod_{i=1}^{n} p(x_i|y_i) \tag{2}$$

where n is the number of tokens in x. Because the objective function is to find the best $p(y|x)$, and $p(x)$ is a priori probability that remains the same for each possible tag class, it only needs to compare the probability of $p(x, y)$.

To solve the data sparseness problem caused by $p(x_i|y_i)$ in Eq. (2), sufficient training data are required for every possible value of x_i in order to calculate $p(x_i|y_i)$. However, in reality, the training data used to compute accurate probabilities is not enough when decoding new corpus. This problem is often solved using the Naïve Bayes. The decomposition of $p(x_i|y_i)$ is as follows:

$$p(x_i|y_i) = \prod_{j} p(f_{ij}|y_i) \tag{3}$$

where f_{ij} is the value of the jth feature of x_i.

Even with the above solution, HMMs suffer from another two limitations. The first one derives from the Naïve Bayes assumption against standard NER rules, which would benefit from a richer representation of observations in terms of many overlapping features, such as capitalization, affixes, part-of-speech (POS) tags, and surface word features. However, these features depend on each other, which violate the Naïve Bayes assumption. The second problem with HMM is that it sets its parameters to maximize the likelihood of the observation sequence, but the task is to predict the state sequence. Namely HMM inappropriately uses a generative joint model to solve a conditional problem [18].

2.3.2 CRFs based methods

Named entity recognition can be considered as a sequence segmentation problem which means each word is a token in a sequence to be assigned a label. Conditional Random Fields (CRFs) are undirected statistical graphical models, a special case of which is a linear chain that corresponds to a conditionally trained finite state machine. It is widely applied in many areas, including computer vision [19], shallow parsing [20] and biomedical named entity recognition [21]. Several famous tools such as NERSuite, Gimli, etc. are all based on CRF. Its mathematical model can be depicted as follows.

x denotes random variables over data sequences to be labeled, and y denotes the random labels over corresponding label sequences. In an undirected graph $G = (V, E)$, a node $v \in V$ corresponding to each of the random variables representing an element y_v of y. (y, x) is a conditional random fields when each random variable y_v obeys the Markov property, which means $p(y_v|x, y_w, w \neq v) = p(y_v|x, y_w, w \sim v)$. During modeling sequences, the most common graph structure is that the nodes

corresponding to elements of y from simple first-order chain, as illustrated in Fig. 2.

A conditional model $p(y|x)$, which is the probability of a particular label sequence y given observational sequence x can be defined as a normalized product of potential functions. A transition feature function of potential function is

$$\exp\left(\sum_j \lambda_j t_j(y_{i-1}, y_i, x, i) + \sum_k \mu_k s_k(y_i, x, i)\right) \tag{4}$$

where $t_j(y_{i-1}, y_i, x, i)$ is a transition feature function of both the observation sequence, the labels at position i and $i - 1$ in the label sequence. $s_k(y_i, x, i)$ is a state feature function of the label at position i and the observation sequence. λ_j and μ_k are parameters to be predicted from training data.

A set of real-valued features $g(x, i)$ of the observation can be defined as feature functions in order to describe some characteristics of the empirical distribution of the training data. Below is an example:

$$g(x, i) = \begin{cases} 1 \\ 0 \end{cases} \tag{5}$$

When the current state (in the case of a state function) or previous and current states (in the case of a transition function) take on particular values, the feature function will take on the value of 1. The state function $s(y_{i-1}, y_i, x, i)$ and transition function $t(y_{i\_1}, y_i, x, i)$ can be denoted with $f_i(y_{i-1}, y_i, x, i)$, thus the $F_j(y, x)$ can be defined as:

$$F_j(y, x) = \sum_{i=1}^{n} f_i(y_{i-1}, y_i, x, i) \tag{6}$$

With the function $F_j(y, x)$, the probabilities of a label sequence y on the observation sequence x can be expressed as:

$$p(y|x, \lambda) = \frac{1}{Z(x)} \exp\left(\sum_j \lambda_j F_j(y, x)\right) \tag{7}$$

where Z(x) is a normalization factor.

The conditional nature of CRF is its main advantage, which resulting in the relaxation of the independence assumptions required by HMMs in order to ensure tractable inference. Moreover, CRF is a discriminatively trained model for labeling and segmenting sequence. It also combines arbitrary, overlapping and agglomerative observation features from both the past and the future. CRF methods can benefit from efficient training and decoding based on dynamic programming. The parameter estimation guarantees that the global optimum can be found.

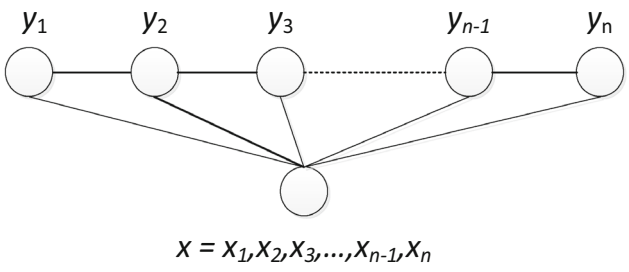

Fig. 2 Graphical structure of a chain-structured CRF

3 Comparisons on the Bio-NER tools

To deeply understand the current situation and future development of the Bio-NER tools, six excellent Bio-NER tools are introduced with the purpose of comparing their performance, which are ABNER [22], LingPipe [23], BANNER [24], NERSuite [25] and Gimli [26], GENIA Tagger [27]. Table 1 presents an overview of their characteristics, supported corpora, feature sets, mathematical models and post-processing techniques. Functions of the six tools overlap with each other, we are going to find out which tool performs best overall and which tool is suitable for specific entity type like DNA or RNA.

3.1 ABNER

A Biomedical Named Entity Recognizer (ABNER) is open source software which is capable of analyzing molecular biology text that can be used to recognize DNA, RNA, protein, Cell Line and Cell type. It employs conditional random fields with a variety of orthographic and contextual features. It has a presentative graphical interface and contains two modules for tagging entities (e.g. protein and cell line) trained on standard corpora.

This tool is written in Java and employs graphical window objects in the Swing library. The CRFs methods are implemented using a quasi-Newton method named as L-BFGS, which could help to find the optimal feature weights. ABNER employs a deterministic finite-state scanner using the Jlex tool for tokenization. It provides a Java application interface which allows users to incorporate ABNER into their own systems and train models on new corpora [22].

3.2 GENIA tagger

With optimization for biomedical text, such as MEDILINE abstracts, GENIA tagger functions well in biomedical domain. It's a good option to extract information from biomedical documents because it is trained on three corpus, the Wall Street Journal corpus, the GENIA corpus and the PennBioIE corpus [28], respectively. The developers apply the bidirectional algorithms and achieved equivalent performance compared with other machine learning models.

Table 1 Overview of the six Bio-NER tools

	ABNER	GENIA tagger	LingPipe	BANNER	NERSuite	Gimli
Release year	2005	2005	2007	2008	2010	2011
Programming language	Java	C ++	Java	Java	C ++	Java
Features						
Linguistic						
Normalization				√	√	√
Chunking		√	√		√	√
POS		√			√	√
Orthographic						
Symbols	√			√	√	√
Counting				√	√	√
Capitalization	√			√	√	√
Word class	√			√	√	
Morphological						
Char n-grams			√	√	√	√
Suffix and prefix	√	√		√		√
Word shape	√				√	√
Lexicons						
Target names	√				√	√
Trigger names	√					√
Model	CRF	MEMM	HMM	CRF	CRF	CRF
Post-processing						
Parentheses				√	√	√
Abbreviation	√			√		√

The algorithm finds the highest probability sequence and the corresponding decomposition structure in polynomial time among all the possible enumerated decomposition structures. The tagging result of GENIA Tagger contains five entities, which are protein, DNA, RNA, Cell Line and Cell type.

3.3 LingPipe

LingPipe is a tool kit for processing text using computational linguistics. It is originally used to find the names of people, organizations or locations in news. Its confidence-based chunkers are first-order hidden Markov methods with emission probabilities estimated by Character Language Models. Using a generalized form of best-first search over the lattice that produced by the forward–backward algorithm, these chunkers are able to iterate an arbitrary number of chunks in confidence-ranked order.

LingPipe's architecture is efficient, scalable, reusable and robust. It also equips a Java API with source code and unit tests and could deal with multi-lingual, multi-domain and multi-genre corpus and mine new data for new tasks [23]. LingPipe contains a model trained on GENETAG [29] corpora, which makes it capable of recognize named entity like proteins, genes, etc. in biomedical text.

3.4 BANNER

BANNER is implemented in Java and based on CRFs model. It is designed to maximize domain independence by neither employing brittle semantic features nor rule-based models [24]. BANNER's processing can be divided into three steps, which is depicted in Fig. 3. Raw sentences are tokenized, converted to features, and labeled. The Dragon toolkit [30] and Mallet [31] are used for part of the implementation. The stream of tokens is converted to features, each of which is a name pair. All of the information about the token is encapsulated by the set of features. The stream of features is labeled so that each token is given the corresponding label. BANNER can extract gene and protein entities from molecular biology text efficiently. BANNER flowchart is shown Fig. 3 [24]:

3.5 NERSuite

NERSuite is designed as a pipe-lined system to facilitate research experiments using the various combinations of different NLP applications [25]. It is written in C++ and contains a tokenizer, a modified version of the GENIA tagger and a named entity recognizer and each of them is an independent module. For a given text in sentence-bag model (each sentence as a vector) document file, NERSuite

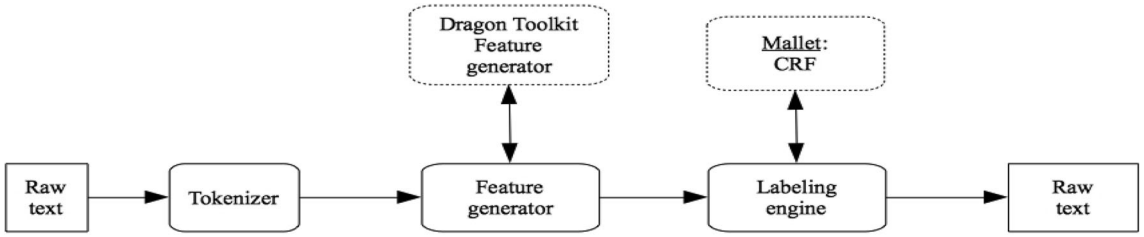

Fig. 3 BANNER's flowchart [24]

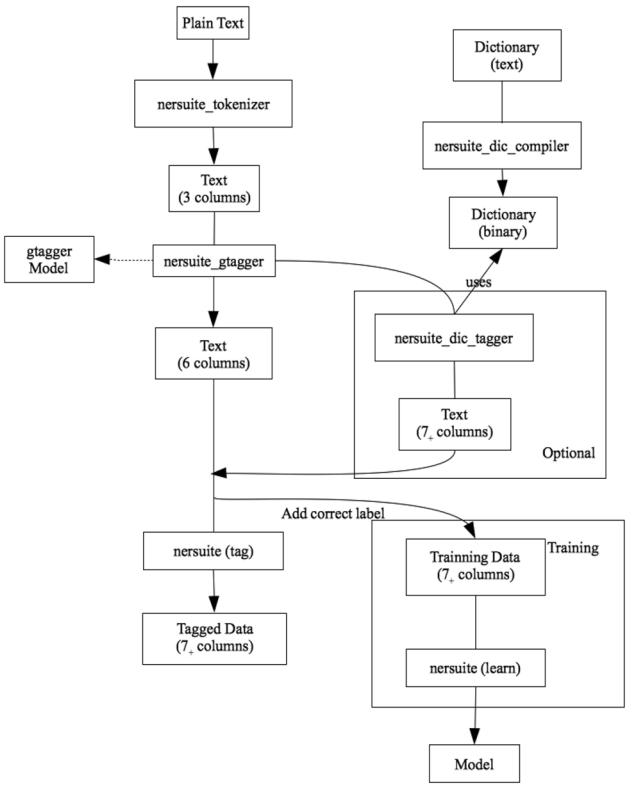

Fig. 4 NERSuite's flowchart [25]

firstly split each sentence into tokens, and computes the detailed positions of each token. The modified GENIA tagger performs POS-tagging, lemmatization and chunking. Finally, with a pre-trained model [25] or user-trained model, NERSutie can deal with biomedical text containing DNA, RNA, protein, Cell Line or Cell type. Figure 4 shows the flowchart of NERSuite.

3.6 Gimli

Gimli provides a trained and optimized model for recognition of biomedical entities like DNA, RNA, protein, Cell line and Cell type from scientific text. It is implemented with Java and can be used as a command line tool. It offers rich functionalities, including training new models, customization of the feature set and parameters' adjustment through a configuration file. Gimli takes advantage of various publicly available tools and resources. The implementation of CRF is provided by MALLET. GDep is employed for tokenization and linguistic processing, i.e. lemmatization, POS tagging, chunking and dependency parsing [32]. In terms of lexical resources, it adopts BioThesaurus and BioLexicon as the resource for biomedical domain terms [33]. Recently, it becomes a state-of-the-art solution for biomedical NER, contributing to faster and better research results. Figure 5 shows the flowchart of Gimli, presenting the workflow of the required steps, tools and external resources [26].

4 Evaluation of the six tools

4.1 Datasets

To compare the BioNER tools in detail, we select GENETAG [29] and JNLPBA [34] to evaluate these six BioNER tools. These two benchmark datasets are most widely used in biomedical named entity recognition domain [26, 29, 34, 35].

4.1.1 GENETAG

GENETAG is composed of 20,000 sentences extracted from MEDLINE abstracts, not being focused on any specific domain [26]. It contains the annotations of proteins, DNAs and RNAs (grouped in only one semantic type). Experts of biochemistry, genetics and molecular biology provided the annotations. This corpus was used in the BioCreative II gene mention challenge [35], providing 15,000 sentences for training and 5000 sentences for testing.

4.1.2 JNLPBA

The JNLPBA corpus contains 2404 abstracts extracted from MEDLINE with the MeSH terms like "human", "blood cell" and "transcription factor". The manual annotation was based on five classes of the GENIA

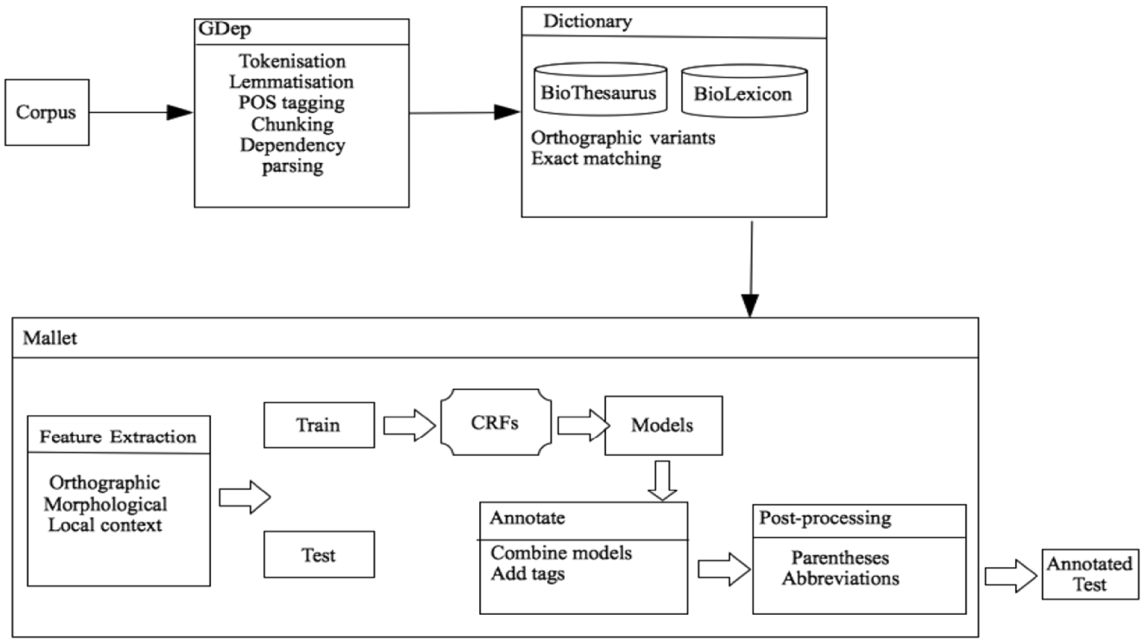

Fig. 5 Gimli's flowchart [26]

ontology, namely protein, DNA, RNA, Cell Line, and Cell type. It was used in BioNLP/NLPBA 2004 [34], providing 2000 abstracts for training and the remaining 404 abstracts for testing.

4.2 Evaluation measures

The evaluation was performed by comparing the six tools' output, in terms of the precision (p), recall (r) and their harmonic mean, the F-measure. They are based on the number of true positives (TP), false positives (FP) and false negative (FN) returned by the system:

TP denotes the number of correctly found name entity chunks; FP denotes the number of found name entity chunks which do not exist in the corpus; FN denotes the number of found name entity chunks that are not found by the Bio-NER tools.

4.3 Performance analysis

GENETAG, having more heterogeneous annotations than JNLPBA, is not focused on any specific biomedical domain. In order to find which system is more appropriate for widely application in biology and medicine, we first conduct our experiment on GENETAG. The results are presented in Table 2 and the best results achieved by all the six tools are in boldface. As one of the tools, Dingare et al. combined Maximum Entropy Markov Model and limited Memory Quasi-Newton maximizier together to perform Named entity task. Moreover, they take advantage of Google web-querying technique, the TnT POS tagger, a gazetteer and other external resources to improve the overall performance of their system [36].

To sum up, Gimli performs better than all the other five tools on GENETAG corpus, achieving the accuracy of 90.22 % and F-measure of 87.17 %, which are 1.56 and 0.74 % improvement over the second best tool, BANNER, respectively. Compared with NERSuite, Gimli is of 1.72 % higher on F-measure. Although LingPipe ranks number five in all the six systems generally, it achieves a highest recall of 88.49 %. From Table 2 we can find that LingPipe is constructed on HMM method, and other four systems all constructed on CRFs method except Dingare's tool. If we can modify the HMM method to compensate the LingPipe's results on precision, it will be an effective method.

Compared with GENETAG, JNLPBA focused more on specific biomedical domain, thus a model trained on the JNLPBA corpus may provide annotations optimized for research on human blood cell transcription factors. JNLPBA splits various types into different semantic groups. Because LingPipe and BANNER do not support JNLPBA corpus, the comparison for results of ABNER, NERSuite, Gimli and GENIA Tagger are shown in Table 3 and the best results achieved by all the six tools are in boldface. Instead of supervised machine learning, Zhang et al. tackle the problem with a stepwise unsupervised solution. Their approach is independent from hand-built rules or examples of annotated entities, which makes it possible to adapt their system to different semantic categories and text genres easily [37].

Except Zhang's tool, it can be seen that the other five tools' overall performance is similar to each other. For all

Table 2 Results obtained on GENETAG corpus

	ABNER (%)	LingPipe (%)	BANNER (%)	NERSuite (%)	Gimli (%)	Dingare's tool [36]
Precision	86.93	72.95	88.66	88.81	**90.22**	82.8
Recall	51.49	**88.49**	82.34	82.43	84.82	83.5
F-measure	64.88	79.97	86.43	85.45	**87.17**	83.2

Table 3 Results obtained from JNLPBA corpus

	ABNER (%)	GENIA tagger (%)	NERSuite (%)	Dingare's tool [36] (%)	Zhang's tool [37] (%)	Gimli (%)
Protein	72.60	72.79	72.74	72.7	67.2	**74.68**
DNA	65.10	66.20	68.58	67.9	55.6	**69.83**
RNA	61.60	64.29	67.23	**68.8**	55.6	67.24
Cell type	72.00	**74.31**	72.11	52.4	50.9	70.49
Cell line	56.00	**57.81**	56.11	69.1	19.9	58.64
Overall	70.50	71.37	71.07	70.1	49.84	**72.23**

the five categories of biomedical name entities, Gimli achieves the highest F-measure, which is 0.86 and 116 % higher than the other ones in top 3. For "Cell Type", GENIA Tagger is 2.2 and 2.31 % higher than NERSuite and ABNER, separately. All the rest six performs poorly with no one higher than 60 %. Moreover, for "DNA" and "RNA" name entities, Gimli and Dingare's tool performs best, repectively. However, all the systems' performances range from 55 to 70 %.

The difference between NERSuite and Gimli lies on "Protein" and "Cell Line" comparisons. Gimli achieves 1.94 and 2.53 % higher performance than NERSuite correspondingly. Although Zhang's tool doesnot perform well on JNLPBA corpus, it exceeds other tools considering generalization, which makes it a good option to deal with diverse group of text.

Because of the complex solutions that include the application of linguistic, lexicon features and the combination of various CRF methods, Gimli outperforms the other five tools and achieves the highest overall performance, again. All the six tools do not perform well in recognizing "Cell Line" names. None of them gets a result over 60 %, this bottom field apparently decreases the overall performance.

4.4 Speed analysis

Besides the comparison of F-measure, we regarded the processing speed as a critical evaluation criterion considering the burst amount of recently published literature. We recorded the tagging time when use these tools to tag 5000 sentences in a machine with two processing cores @ 3.20 GHz and 8 GB of RAM running Linux. The details for the speed of each tool can be found in Table 4. In our experiment, ABNER is the fastest system; however it performs worst on GENETAG corpus. BANNER is not as fast as ABNER, but it ranks first considering speed and F-measure, thus it is a suitable option for biomedical named entity recognition. Gimli can obtain highest F-measure on two corpuses, but it is the slowest. We can clearly find that the promotion of F-measure is obtained by a more sophisticated software framework and longer processing time.

In recent years, taking account of unconstrained growth in thesis and biomedical database, researchers are seeking for efficient methods to process and extract enormous information. Granted that there is no need for high F-measure and ABNER is used for analysis, the tagging speed is still relatively slow when it comes to Terabytes of data, let alone other tools. In order to catch up with the growing speed of literature, Tang et al. proposed a CRFs based parallel biomedical named entity recognition algorithm employing MapReduce framework, which reduces the model training time for large-scale training samples [38]. Li et al. also developed a parallel CRFs algorithm called MRCRF (MapReduce CRF) containing two parallel sub-algorithms, MRLB (MapReduce L-BFGS) and MRVtb (MapReduce Viterbi), respectively. They polish up the performance significantly without the overmuch decrease of correctness [39].

5 Conclusion

In our paper, we present a review on the research of biomedical named entity recognition, especially on the three fundamental models and six open source Bio-NER tools. It is clear that to identify and classify the named entities in biomedical literatures is an extremely sophisticated work. With the help of machine learning algorithms,

Table 4 Tagging speed for each tool

	ABNER	GENIA tagger	NERSuite	LingPipe	BANNER	Gimli
Tagging speed (sentences/s)	581	135	90	93	186	58

now we can achieve a result much better than dictionary-based or rule-based methods. Gimli becomes one of the state-of-the-art solutions for biomedical named entity recognition. But facing different applications, the user still need to consider different tools. The suggestions are as follows:

(a) For overall performance on common biomedical corpus, BANNER and Gimli can achieve satisfied results;
(b) To exactly find out the "Cell Type" entities, GENIA Tagger or NERSuite will perform well;
(c) Discovery on "DNA" and "RNA" names, we can choose NERSuite and Gimli;
(d) For "Protein" name entity recognition, Gimli is the unique candidate.
(e) Dingares's tool is suitable for "Cell Line" recognition.
(f) BANNER can perform NER task accurately with a relatively fast speed.
(g) In order to cope with huge amount of data, we need more paralleled algorithms.

There is no doubt that Gimli could achieve the champion because it is newly proposed (see Table 1) and integrated more modules than the other tools.

We also find some issues where future research is likely to be concentrated. Most of current tools are developed focus on two corpora, GENETAG and JNLPBA. These tools may be tuned or modified to achieve better results on the two datasets. However, they may perform badly on real application. It will be much better if we can do research on the generalization ability or make the benchmark corpus update continually. In addition, from the experiments, we can find that machine learning methods integrated with biomedical dictionary (such as NERSuite and Gimli) may transcend the ones without. However, existing dictionary in real biomedical data may not be well coincide with the machine learning methods. A successful Bio-NER tool demands newly compiled biomedical dictionaries covering different research areas, though it is time-consuming and costly.

Acknowledgments This paper is supported by the National Key Basic Research Program of China (No. 2015CB453000), National Natural Science Foundation of China (Nos. 61572228, 41101376, 61272207 and 61300147), and the Science Technology Development Project of Jilin Province of China (20130101070JC, 20130522106JH and 20140520070JH).

References

1. Rosario B, Hearst MA (2004) Classifying semantic relations in bioscience texts. In: Proceedings 42nd annual meeting association computional linguistics. doi:10.3115/1218955.1219010
2. Chiang J-H, Yu H-C (2003) MeKE: discovering the functions of gene products from biomedical literature via sentence alignment. Bioinformatics 19:1417–1422. doi:10.1093/bioinformatics/btg160
3. Ciaramita M, Gangemi A, Ratsch E et al (2005) Unsupervised learning of semantic relations between concepts of a molecular biology ontology. In: IJCAI. pp 659–664
4. Zhou G, Su J (2002) Named entity recognition using an hmm-based chunk tagger. In: Proceedings 40th annual meeting association computational linguistics. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 473–480
5. Collier N, Nobata C, Tsujii J (2000) Extracting the names of genes and gene products with a hidden markov model. In: Proceedings 18th conference computional linguistics, vol 1. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 201–207
6. Gaizauskas R, Demetriou G, Humphreys K (2000) Term recognition and classification in biological science journal articles. In: Proceedings computional terminology for medical and biological applications workshop 2nd international conference NLP. pp 37–44
7. Kazama J, Makino T, Ohta Y, Tsujii J (2002) Tuning support vector machines for biomedical named entity recognition. In: Proceedings ACL-02 workshop natural language processing in the biomedicine domain, vol 3. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 1–8
8. Takeuchi K, Collier N (2002) Use of support vector machines in extended named entity recognition. In: Proceedings 6th Conference Natural Language Learn, vol 20. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 1–7
9. Zhou G, Zhang J, Su J et al (2004) Recognizing names in biomedical texts: a machine learning approach. Bioinformatics 20:1178–1190. doi:10.1093/bioinformatics/bth060
10. Fukuda K, Tamura A, Tsunoda T, Takagi T (1998) Toward information extraction: identifying protein names from biological papers. Pacific Symposium Biocomputing Pacific Symposium Biocomputional. pp 707–718
11. Nobata C, Collier N, Tsujii J (1999) Automatic term identification and classification in biology texts. In: Proceedings 5th NLPRS. pp 369–374
12. Chang JT, Schütze H, Altman RB (2002) Creating an online dictionary of abbreviations from MEDLINE. J Am Med Inform Assoc JAMIA 9:612–620
13. Liu H, Aronson AR, Friedman C (2002) A study of abbreviations in MEDLINE abstracts. In: Proceedings AMIA annual symposium AMIA symposium. pp 464–468
14. Sondhi P A survey on named entity extraction in the biomedical domain. Available online at http://sifaka.cs.uiuc.edu/~sondhi1/survey1.pdf
15. Tsuruoka Y, Tsujii J (2003) Boosting precision and recall of dictionary-based protein name recognition. In: Proceedings ACL 2003 workshop natural language processing biomedicine, vol 13. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 41–48

16. Yang Z, Lin H, Li Y (2008) Exploiting the performance of dictionary-based bio-entity name recognition in biomedical literature. Comput Biol Chem 32:287–291 **(2008.03.008)**
17. Proux D, Rechenmann F, Julliard V et al (1998) Detecting gene symbols and names in biological texts: a first step toward pertinent information extraction. Genome Inform Workshop Genome Inform 9:72–80
18. Tsai RT, Sung C-L, Dai H-J et al (2006) NERBio: using selected word conjunctions, term normalization, and global patterns to improve biomedical named entity recognition. BMC Bioinform 7:S11. doi:10.1186/1471-2105-7-S5-S11
19. He X, Zemel RS, Carreira-Perpindn MA (2004) Multiscale conditional random fields for image labeling. In: Proceedings 2004 IEEE computional society conference computional vis. pattern recognition 2004 CVPR 2004, vol 2. pp II–695–II–702
20. Sha F, Pereira F (2003) Shallow parsing with conditional random fields. In: Proceedings 2003 conference North America chapter association computional linguistics human language technology, vol 1. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 134–141
21. Settles B (2004) Biomedical named entity recognition using conditional random fields and rich feature sets. In: Proceedings international joint workshop natural language processing biomedicine its application. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 104–107
22. Settles B (2005) ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text. Bioinformatics 21:3191–3192. doi:10.1093/bioinformatics/bti475
23. Baldwin B, Carpenter B (2003) LingPipe. World Wide Web Httpalias-Comlingpipe
24. Leaman R, Gonzalez G, others (2008) BANNER: an executable survey of advances in biomedical named entity recognition. In: Pacific Symposium Biocomputing. pp 652–663
25. Cho HC (2010) NERsuite: a named entity recognition toolkit. Tsujii Laboratory, Department of Information Science, University of Tokyo, Tokyo, Japan. http://nersuite.nlplab.org. http://nersuite.nlplab.org/. Accessed 14 Nov 2014
26. Campos D, Matos S, Oliveira JL (2013) Gimli: open source and high-performance biomedical name recognition. BMC Bioinform 14:54. doi:10.1186/1471-2105-14-54
27. Tsuruoka Y (2006) GENIA tagger: Part-of-speech tagging, shallow parsing, and named entity recognition for biomedical text
28. Tsuruoka Y, Tsujii J (2005) Bidirectional inference with the easiest-first strategy for tagging sequence data. In: Proceedings conference human language technology empirical methods natural language processing. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 467–474
29. Tanabe L et al (2005) GENETAG: a tagged corpus for gene/protein named entity recognition. BMC bioinform 6(Suppl 1):S3
30. Zhou X, Zhang X, Hu X (2007) Dragon toolkit: incorporating auto-learned semantic knowledge into large-scale text retrieval and mining. In: Tools artificial intelligence 2007 ICTAI 2007 19th IEEE international Conference on IEEE. pp 197–201
31. McCallum AK (2002) Mallet: a machine learning for language toolkit. Available online at https://people.cs.umass.edu/~mccallum/mallet/
32. Sagae K, Tsujii J (2007) Dependency parsing and domain adaptation with LR models and parser ensembles. In: EMNLP-CoNLL. pp 1044–1050
33. Liu H, Hu Z-Z, Zhang J, Wu C (2006) BioThesaurus: a web-based thesaurus of protein and gene names. Bioinformatics 22:103–105. doi:10.1093/bioinformatics/bti749
34. Kim J-D, Ohta T, Tsuruoka Y et al (2004) Introduction to the bio-entity recognition task at JNLPBA. In: Proceeding international joint workshop natural language processing biomedicine its applications. Association for Computational Linguistics, pp 70–75
35. Smith L, Tanabe LK, Ando RJ et al (2008) Overview of BioCreative II gene mention recognition. Genome Biol 9:S2
36. Dingare S, Nissim M, Finkel J et al (2005) A system for identifying named entities in biomedical text: how results from two evaluations reflect on both the system and the evaluations. Comp Funct Genom 6:77–85. doi:10.1002/cfg.457
37. Zhang S, Elhadad N (2013) Unsupervised biomedical named entity recognition: experiments with clinical and biological texts. J Biomed Inform. doi:10.1016/j.jbi.2013.08.004
38. Tang Z, Jiang L, Yang L et al (2015) CRFs based parallel biomedical named entity recognition algorithm employing MapReduce framework. Clust Comput 18:493–505. doi:10.1007/s10586-015-0426-z
39. Li K, Ai W, Tang Z et al (2015) Hadoop recognition of biomedical named entity using conditional random fields. In: IEEE transaction parallel distribution system. pp 1–1. doi:10.1109/TPDS.2014.2368568