Comparative study of Named-Entity Recognition methods in the Agronomy domain

Huy Do[1], Khoat Than[3], and Pierre Larmande[1,2]

[1] University of Science and Technology of Hanoi (USTH), ICT Lab
[2] Institute of Research for Development (IRD), LMI RICE
pierre.larmande@ird.fr
[3] Hanoi University of Science and Technology (HUST)
Hanoi, Vietnam
khoattq@soict.hust.edu.vn

Abstract. Text mining is becoming an important part of research topic in biology with the original purpose to extract biological entities such as genes, proteins and traits to extend the knowledge from scientific papers. However, few thorough studies on text mining and applications are developed for plant molecular biology data, especially rice, thus resulting a lack of datasets available to train models. Since there is no/rare benchmark for rice, we have to face various difficulties in exploiting advanced machine learning methods for accurate analysis of rice bibliography. In this article, we developed a new training datasets (Oryzabase) as the benchmark. Then, we evaluated the performance of several current approaches to find a methodology with the best results and assigned it as the state of the art method for our own technique in the future. We applied Name Entities Recognition (NER) tagger, which is built from a Long Short Term Memory model, and combined with Conditional Random Fields (LSTM-CRF) to extract information of plant genes and proteins. We analyzed the performance of LSTM-CRF when applying to the Oryzabase dataset and improved the results up to 86% in F_1. We found that on average, the result from LSTM-CRF is more exploitable with the new benchmark.

Keywords: Text mining, LSTM-CRF, NER, Bioinformatics, Plant Genomics

1 Introduction

1.1 Problem Statement

The last few decades have witnessed the massive explosion of information in life science. However, an important proportion of information relevant to this field is not available from databases but is instead present in unstructured scientific documents, such as journal articles, reviews and reports. Agronomy is an overarching field, that comprises of diverse domains such as Genetics, Plant Molecular Biology, Ecology and Soil Science. Despite the advancement in information technologies, scientific communication in agronomy is still largely based on text because it is the common way to report scientific advancements. To effectively develop applications to improve crop production through sustainable methods, it is important to overlay research findings from these fields as they are highly inter-connected. However, the collection of content is growing continuously and the information are currently available as unstructured text. Using these resources more efficiently and taking advantage of associated cross-disciplinary research opportunities poses a major challenge to both domain scientists and information technologists.

An important task is to identify biological entities and their classification, which is also called the recognition of name entities (NER). Several text mining methods and tools have been developed to solve this problem, divided into four main approaches [1]. The most basic and traditional methodology is rule-based approach. The technique identifies entities by a group of written rules, which are manually done by the domain scientists with linguistic knowledge. As a result, it is time-consuming and easily error-prone. The second approach is dictionary-based methods. The model matches the candidate entities with a dictionary that contains all the known entities to detect whether the candidate belongs to a defined category or not. However, if there exists new entities, for instances, from new discovery and not in the available dictionary, the system is not able to recognize them, which reduces the efficiency of the model. The third approach is based on machine learning, which uses a statistical classifier to extract the features (prefixes, suffixes, number of capital letters, etc.) that are able to recognize the entities. Several familiar algorithms have been proposed such as Naive Bayes, Conditional Random Fields (CRF) [2], and so on. However, this method need an important corpus annotated manually to train and test efficiently. Last but not least, hybrid approach is proposed as the combination of two or more of the previous techniques to take advantage of the strength and reduce the weak points among them.

Identifying Biological Entities is not trivial. Despite the fact that there exists many available approaches to handle this problem in general and in the Biomedical domain, few thorough studies have been implemented for plants, especially rice. Moreover, we found rare benchmarks available for plant species and none for rice. Thus, we faced various difficulties to exploit advanced machine learning methods for accurate analysis of rice.

1.2 Objective

In the large scale, we are currently building an RDF knowledge base, Agronomic Linked Data (AgroLD – `www.agrold.org`). The knowledge base is designed to integrate data from various publicly plant centric databases such as Gramene [3] , Oryzabase , TAIR[4] to name a few. The aim of AgroLD project is to provide an integrated portal for bioinformatics and domain experts to exploit the homogenized data model towards filling the knowledge gaps. In this landscape, we aim to extract relevant information from the literature in order to enrich the content of integrated datasets.
Due to the scope of the article, we exploit information from Oryzabase database as our benchmark and focus on researching and evaluating the performance of the current approaches and assigning the method with the best accuracy and efficiency as the baseline for our own technique in the future. After researching and analyzing, Long Short Term Memory - Conditional Random Fields (LSTM - CRF) has been chosen for further analysis due to their competency and efficiency compared to others.

2 Materials and methods

2.1 NER-tagger with LSTM-CRF

Long Short Term Memory - Conditional Random Field (or LSTM-CRF) is a generic method based on deep learning and statistical word embeddings. To deeply understand the whole architecture, it is necessary to understand the concept of LSTM and CRF first.

2.1.1 LSTM

Long Short Term Memory (or LSTM) [5,6] is a type of recurrent neural network (RNN) focusing on learning order dependence in sequence prediction problems. When RNNs are applied for learning long dependencies in practice, they tend to fail the tasks with the bias towards the most recent inputs of the sequence. Long Short-term Memory Networks (LSTMs) is the answer for this issue. It proves to capture long-range dependencies by incorporating a memory-cell and using several gates that control the proportion of the input to give to the memory cell, and the proportion from the previous state to forget [6]. Taking a sequence of vectors $(x_1, x_2, ..., x_t)$ as an input, LSTM returns the output in the form of another sequence $(h_1, h_2, ..., h_t)$ of equal length to represent the information of the input. The implementation of LSTM is based on the equation below:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{1}$$

$$c_t = (1 - i_t) \times (c_{t-1} + i_t) \times (tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{2}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{3}$$

$$h_t = o_t \times \tanh(c_t) \tag{4}$$

where σ is the element-wise sigmoid function, and $\times$ is the element-wise product.

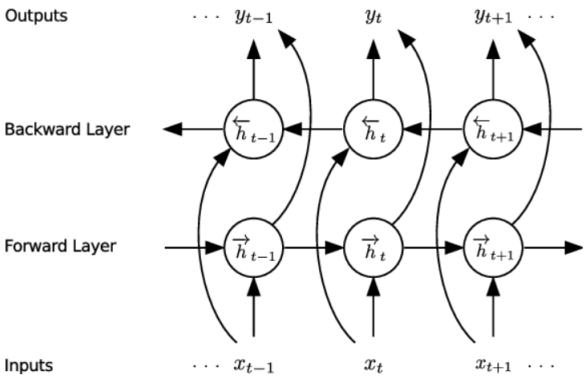

Fig. 1. LSTM model [7]

The workflow follows the combination of LSTM pair (forward and backward LSTM) called a bidirectional LSTM [8]. After receiving the input $(x_1, x_2, ..., x_t)$ with each element represented as a d-dimensional vector, the first LSTM (or forward LSTM) computes the representation $\overrightarrow{h}_t$ of the left context of the sentence at every word t and the second one (or backward LSTM) reads the same input sequence in reverse to obtain the representation $\overleftarrow{h}_t$ of the right context using different parameters. Then, the model concatenate its left and right context representations, $h_t = [\overrightarrow{h_t}\overleftarrow{h_t}]$ to achieve the representation of a word, which is useful for various tagging applications.

2.1.2 CRF

Conditional random fields (CRFs) is a tagging model for labeling and segmenting structured data, such as sequences, trees and lattices. The general idea of this model [9] is to make independent tagging decisions for each output y_t using the features h_t. However, for NER tagger, using traditional CRF with independent classification decisions is insufficient and impossible with numerous constraint because of the strong dependencies cross the output labels.
In this article, we focus on the jointly CRF model proposed by Lafferty et al.[2]. For an input $X = (x_1, x_2, ..., x_t)$, the matrix of scores output P is obtained from the bidirectional LSTM network with n x k in size, where k is the number of tags and $P_{i,j}$ regarding the score of j^{th} tag of i^{th} word in a sentence [10]. Meanwhile, for the sequence of predictions $Y = (y_1, y_2, ..., y_t)$, the score is:

$$s(X, y) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=0}^{n} P_{i, y_i} \tag{5}$$

where A represents a [k + 2] matrix of transition scores from the tag y_i to tag y_j.

The probability for the sequence y then is defined by a softmax over all possible tag sequences as:

$$p(y|X) = \frac{e^{s(X,y)}}{\sum_{\tilde{y}\epsilon Y_x} e^{s(X,y)}} \tag{6}$$

To encourage the network to produce a valid sequence of output labels, the log-probability of the correct tag sequence is maximized in training process as:

$$\log\left(p(y|X)\right) = s(X,y) - \log\left(\sum_{\tilde{y}\in Y_x} e^{s(X,\tilde{y})}\right) \tag{7}$$

where Y_X indicates all possible tag sequences for a sentence X.
After that, we predict the output sequence in decoding step following the equation:

$$y* = \arg\max_{\tilde{y}\in Y_x} s(X,\tilde{y}) \tag{8}$$

2.1.3 LSTM-CRF

The architecture of LSTM-CRF is illustrated in Fig.1 [5]. The whole system includes three main layers: the embedding layer as input, bi-directional LSTM layer, and CRF layer as output. Given the raw sentence made of the sequence of words w_1; w_2; ...; w_n as the input, the embedding layer produces an embedding vector x_1; x_2; ... ; x_n for each word. Every embedding vector regarding distinct word is a concatenation of two components: word- and a character-level embedding. We retrieve the word-level embedding from a lookup table of word embeddings, meanwhile, we apply a bi-directional LSTM to the character sequence of each word and then concatenate both directions to achieve the character-level embedding. That means the resulting sequence of embeddings x_1; x_2; ... ; x_n is fed into bi-directional LSTM-layer to produce a refined representation of the input and we take as the input to a final CRF layer. The final output from this layer is obtained by applying the classical Viterbi algorithm. We trained the whole network by back-propagation.

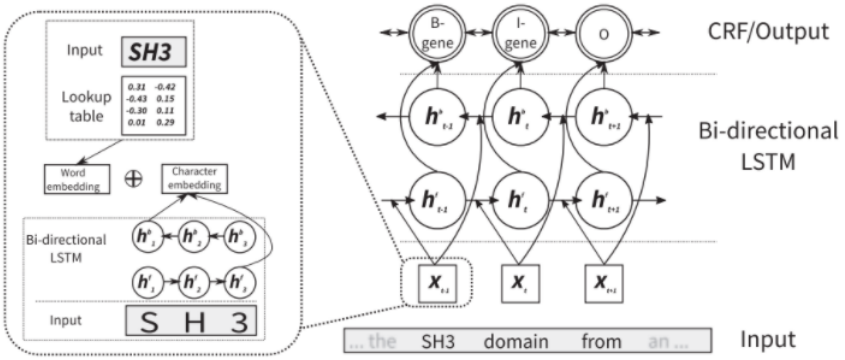

Fig. 2. LSTM-CRF model[5]

2.2 NER-hybrid approach

This approach bases on a dictionary lookup entity recognizer combining with machine learning classifiers. First, we used the OGER entity recognizer to annotate the objects in selected domain ontologies. Next, the Distiller framework is used to extract this information as a feature for a machine learning algorithms to select relevant entities. For this approach, we implemented two different machine learning algorithms: Conditional Random Fields and Neural Networks.

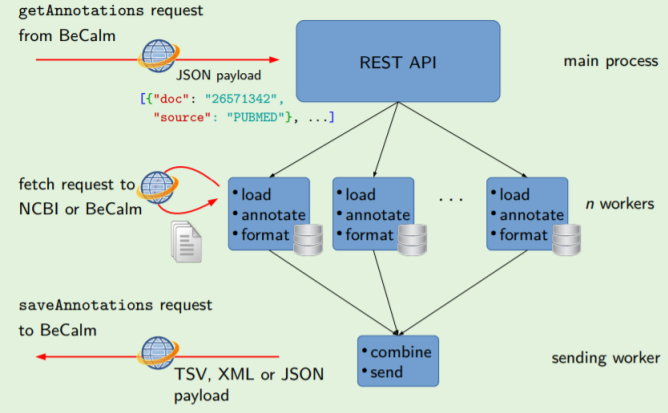

Fig. 3. the structure of OGER

2.2.1 OGER

The OntoGene group has developed an approach for biomedical entity recognition based on dictionary lookup and flexible matching.[1] Their method has been applied in several competitive evaluation of biological text mining technologies, and often archive high results.
OGER perform a flexible interface for dictionary based annotation. OGER provides the annotated terms along with the corresponding identifiers either in a simple tab-separated text file. For the given input which is Oryzabase references included over 10000 abstracts and title of scientific articles, we implement OGER as a permanent web service which is accessible through an API. Before implementation, input data was extracted into several files, with the same format to make the evaluation process be more easily. OGER architecture was configured to have a strategy towards to a greater false positives which leads to a lower precision value, but less false negatives which can get a greater recall.
Because our project focus more on the information of plant's genes and proteins and their relationship, we choosed the gene dictionary of Gene Ontology Consortium[11] and the protein dictionary of [12] came from back-end software of Bio TermHub, which is a meta-resource for biological resources. In OGER, the entities of the term dictionary were then preprocessed in the same way as the documents with respect to tokenization, stemming, and case sensitivity, as described below. Next, the input documents will be compared to the dictionary

with an exact-match strategy.

2.2.2 Distiller

The Distiller framework is an open-source project which is developed to build a flexible, extensible system for natural language processing fields.[1] The main process of Distiller based on the performance of Automatic Key-phrase Extraction (AKE) to extract information from text. AKE seems to be different from NER, as while the former is interested in finding the small set of the most relevant information in a document, and then focus on finding all the information of selected types. Besides, AKE can be performed as both an unsupervised and supervised algorithms, and Distiller actually has its roots in an unsupervised approach.
About the architecture, Distiller is organized in series of single-knowledge oriented modules where each module is designed to perform a single tasks efficiently [13] such that part-of-speech (POS) tagging, statistic analysis, and so on. It performs with the ability to implementing different pipeline for different tasks. All modules is required to share the entity extracted results on a"blackboard", so that a module can share the knowledge with another ones. Implementing KE tasks with Distiller is reduced to specifying pipeline with annotators.[13] A task is normally dividing into steps: text pre-processing , candidate key-phrase generating, and candidate raking. The pipeline process of Distiller is described in figure 4.

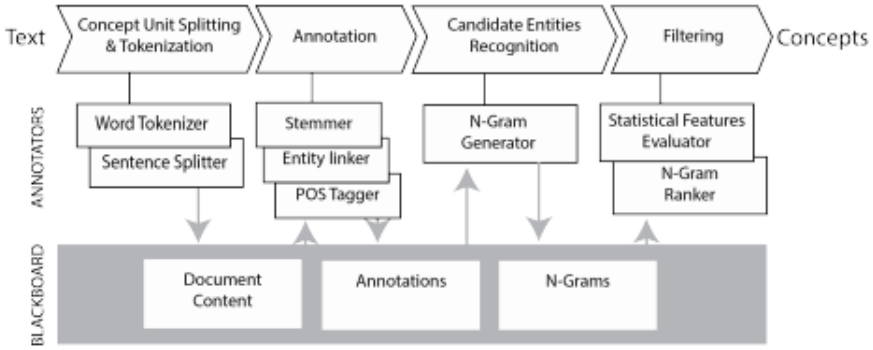

Fig. 4. Knowledge extraction pipeline

In this case, we performed supervised AKE methods using supervised machine learning algorithms. Normally, the first step is generating the key-phrases, which use POS tagger to select patterns. Then, we assign features to keyphrase, using statistical, linguistic, or semantic knowledge. Finally, a machine learning algorithm is trained and then evaluated over a set of documents associated with ground truth. However, in our case, we replace the generating key-phrase step by OGER's output.

2.2.3 Models and Features

In our project, we proposed two machine learning algorithms: Neural Network (NN) and Conditional Random Fields which are considered to perform the exploitable results.[1] The performance of Distiller is different between two methods. In case of CRF, it uses the annotated output of OGER as a feature, and considers any token in the text as an entity to predict. In contrast, Distiller only focus on filtering OGER's output, and process to classify for each entity phrase. To sum up, the work flow for the hybrid approach will be implement as follows:

1. Based on traditional dictionary lookup, tagging object from text data using OGER from given input.
2. Integrating output of OGER with Distiller framework to assign new features, preparing them to be processed by a machine learning system.
3. Build a machine learning system including a simple neural network and a CRF, selects the relevant entities in the document using the information generated in the previous steps.

For CRF, we considered each word as a token and tried to add some features. To do some experiments, we have trained several models with different kinds of features. Finally, we have selected a number of features that effect to the perfomance of CRF models. The list of CRF features is showed in the table 2.

	Conditional Random Fields
Input	Single tokens
Features	
Candidate POS-tagger	Yes
Candidate tag (IOB format)	Yes
Candidate contains numbers	Label yes/no
Candidate ends with numbers	Label yes/no
Candidate pre-selected by OGER	Label yes/no
Neighbor tokens	Yes
Neighbor token POS-taggers	Yes
Total features for each tokens	23

Table 1. The details of CRF features

For NN, we consider input as n-grams that selected by OGER. The features are generated from Distiller and some manual generated features. The list of CRF features is showed in the table 3.

	Neural Network
Input	n-grams selected by OGER
Features	
Candidate tag (IOB format)	Yes
Distiller features	Yes
Candidate contains numbers	Count
Candidate ends with numbers	Label yes/no
Total features	13

Table 2. The details of NN features

2.3 Dataset

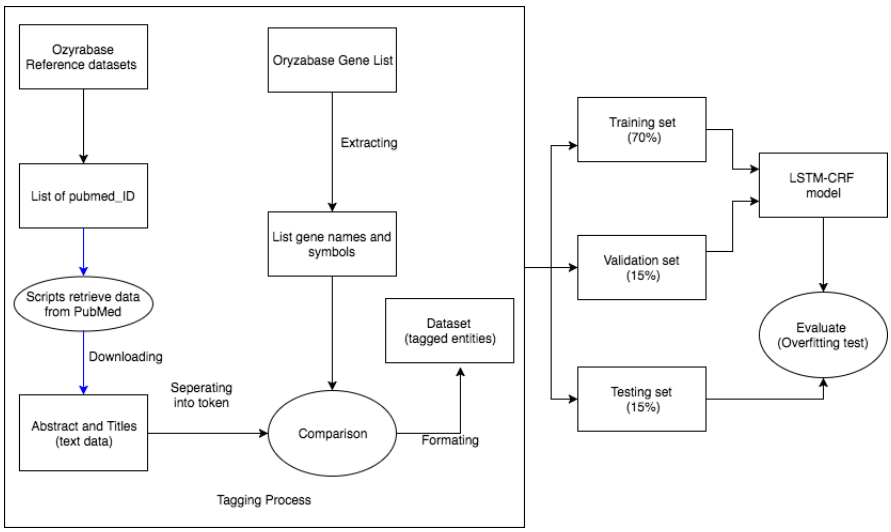

Fig. 5. Schema of the training process

In this project, we use datasets from Oryzabase[4], an integrated rice science database. Concentrating on the entities of rice genes, we downloaded the Gene List and Reference datasets, then took them as the training data and ground-truth, respectively for diverse approaches to extract gene entities. The Oryzabase Gene List contains 21739 different genes with detailed Trait Gene Id, GSNL Gene Symbol, Gene symbol synonym(s), CGSNL Gene Name, Gene name synonym(s), Protein Name, Allele Chromosome No. Explanation, Trait Class, RAP ID, Gramene ID, Arm Locate(cM), Gene Ontology, Trait Ontology, Plant Ontology. Meanwhile, Oryzabase Reference includes about 44837 references from distinct scientific articles which are ranged in the form of ReferenceID, PubMedId, Author, Title, Journal Volume, Pages, Year, CGSNL Gene Symbol, Gene Name Synonym.

[4] `https://shigen.nig.ac.jp/rice/oryzabase/`

Dataset	Text genre	Text type	Entity Type
Oryzabase	Scientific article	Abstract and title	Genes

Number of articles	Number of sentences	Number of words
10400	75096	2697726

Table 3. The details of main dataset

3 Implementation

3.1 Text pre-processing

The original format of all data were formed into separated tabs, which include the pubmed ID, the title, released year, journal and abstract part. Due to the size of dataset, we separated them into several files to avoid the problems of Internet connection during the downloading process as well as make the error trace becoming more easily.
The first step is converting the abstracts and titles into a uniform format. To tokenizing data, each word considering as a token, is given in the following lines, one token per line, and included 3 tabs: the first one being the word itself, the part-of-speech tags (POS tag) and the last one being the entity type. Each token is on a separate line, ends with dots and has an empty line after each sentence. In addition, an empty line indicates the end of a document, and the next document starts after this empty line.
From the beginning, the formating data was processed manually by our own scripts, and all of them were written in Python.
Firstly, we split the abstracts and titles into sentences, and from sentences into tokens. Then, we compared the word with the Oryzabase list gene dictionary and applied the Inside-Outside-Beginning (IOB) format to determine the token which is entiry or not. To make the tag become more meaningful for applying CRF, we use POS tag on tokens by using a specific library of Python for natural language processing. During the impletation of two methods, we also use this pre-processed data to build and train the model with some addition features based on our purpose, therefore this step is very important. To avoid some bugs, we decide to convert data into lower case and filter some

3.2 NER-tagger

Before going to training process, we need to define some properties. First, the LSTM(-CRF) will work at words levels which include an entity tag to determine itself and the input data going through model is sentence level. The Named Entity Recognition tagger method (NER tagger), which is a LSTM (with CRF) model, was built and the original source code are available at `https://github.com/glample/tagger`). However, we had re-configured this model a bit to fit with our dataset instead of using immediately.
On the training process, at the beginning, we have done several experiments

to find the best parameters for the model. In our models, we used a learning rate equal to 0.001 instead of the default value 0.005. We saw that the default one was too high and it made the model not converge. Besides, we have trained LSTM model with and without the CRF to evaluate the effect of this part to the general result of dataset.

3.3 Hybrid Methods

First step, we use OGER supported by Gene Ontology and Protein Ontology dictionary to tag all objects in Oryzabase dataset. The output will be formatted into table which contains information of genes. Then, we have evaluated the performance of OGER on the whole dataset. The purpose of this evaluation is to see how OGER effectively work on this dataset and based on that to improve the result.
For CRF and NN, we also started with POS-tag and candidate tag prepared in the pre-processsing part and combined with new features that are generated and listed in the previous part.

3.4 Evaluation parameters

We saw that Oryzabase dataset is quite big, thus we decided to separate dataset into 3 subsets with different ratio: 70% for training set, 15% for testing set, and the last 15% of dataset for validation which can help the model avoid over-fitting issue. (Fig.3).
To evaluate a deep learning model, we also used the precision, recall and $F_1 - score$ on the test set. By evaluating the set of true positive, false positive and false negative, we computed the value of $F_1 - score$.
Besides, we used some small tests to ensure that the model would not be over-fitting as well as check the efficiency of models.

4 Results and discussion

4.1 Result

We assessed the performance of NER tagger method by evaluating the LSTM-CRF, on the Oryzabase dataset covering severals different types of rice genes. LSTM-CRF uses as features only low-dimensional representations of the words in the vicinity of the to-be-classified tokens, created by mixing word-level embeddings created in an unsupervised fashion with character-level embeddings trained on the respective corpus.
Besides, we also mentioned to the result of the first part of hybrid method which is the dictionary lookup process. Based on this output, we applied some techniques to improve the result
Results were compared between a original LSTM model; a LSTM with CRF layers which have same function to recognize entity in the text data using tag information; OGER - a dictionary lookup method which are first part of second method, OGER combining with CRF model ; and OGER with NN.
We have evaluated the performance of all models on Oryzabase dataset. Results

in terms of precision, recall, $F_1 - score$ for each model are shown in table 4. LSTM-CRF achieves the best performance between several models. On average, F_1-score is 86.72% for the generic LSTM-CRF method, 80.44% for the generic LSTM method.

	Precision(%)		Recall(%)		$F_1 - score$(%)	
	(i)	(ii)	(i)	(ii)	(i)	(ii)
LSTM	80.16	78.06	79.16	82.97	79.66	80.44
LSTM-CRF	87.24	87.32	84.73	86.13	85.97	86.72

Table 4. the best result of performance values in terms of precision, recall and $F_1 - score$ for pure LSTM and LSTM-CRF method with different training parameters: (i)learning rate=0.001, dropout =0.3 ,(ii) learning rate=0.001, dropout =0.5

On the other side, the result of hybrid method is very exploitable. The performance of OGER with Conditional Random Field got the best result on testing set among 3 approachs which is 86.72% on average. The second rank is OGER combining with Neural Network, which has reached 67% accuracy on average. Although the improvement are not high as we expected but it still has some improvement rather than the only OGER which is arounf 58.5%. The result is shown in the table 5.

	Precision(%)	Recall(%)	$F_1 - score$(%)
OGER	53.03	65.23	58.50
OG+NN	63.93	71.10	67.32
OG+CRF	88.39	82.24	85.08

Table 5. the result of performance of hybrid method in terms of precision, recall and $F_1 - score$

Then, to re-check the accuracy and efficiency of the model as well as to be sure that model not over-fit, we have done some tests manually with a small set of data, we saw that the result of trained model are very exploitable.

4.2 Discussion

During the process to complete the method, we have dealt with several issues which came from the data format process as well as the configuration of model. About the dataset, the training data is created from abstract and title of over 10.000 scientific articles in Oryzabase reference. We cannot use all raw texts of an article because it will increase the size of dataset too largely. With the aim to find the appearance of gene names or symbols in text data, we expect that the these kinds of information could be appeared in the first part of articles. However, the huge number of data (2.7 millions words) with different kinds of writing styles takes a lot of time for the pre-processing.

After that, we saw that there exists the imbalance of each entity types in the dataset. We need the text information of gene names/symbols but in fact, the ratio between genes and normal words are really huge. After some first training times, we got the models which reach over 65.5% in accuracy but it did not satisfy our expectation. The imbalance of data can lead to worse result during the training process, and the traditional oversampling methods seem not available for text data, which are really meaningful in the connection of words and sentences.
To handle this problem, we based on the configuration of NER models that have input at sentence level and process input at word level. Therefore, we can collect the all sentences which contain the entity tags without effect the meaning of text data. This solution seems to be effective to improve the performance of the model when it reaches over 86% in accuracy of $F_1 - score$.
In the result section, we have shown the result of the performance for a specific parameter which is the value of dropout functions. For both LSTM and LSTM-CRF, at the beginning, we used the default value equal to 0.005 but these models could not converge. After that, the value of learning rate equal to 0.001 has been chosen through several experiments and evaluation. For the value of dropout, we tried with different values of dropout because we want to see the performance of model as well as the effect of parameters to the training process. The default value of dropout equal to 0.5 was firstly chosen for the training process. However, according to Lample et al.(2016) [10], the value of dropout equal to 0.3 was optimal for most dataset evaluated. Therefore, we want to try for both values and the result is not really effective.
In recent years, pattern- and dictionary-based methods have been replaced by new approachs based on machine learning in general and deep learning more specifically. Nowadays, several methods relying the combination of deep neural network with other techniques are used to develop applications which are able to detect entities automatically rather than manually as the traditional ways. It makes a lot of benefits for the research works as well as for human life. For examples, it can help people to extend the databases which can be used for several purposes of human, or can help scientists finds some new entities. However, scientists normally focus more on the biomedical or somethings that related to human genes. The research in Agronomy now is very popular and plays an important role for human nutrition.

5 Future work

Our purpose is finding a method to extract the information of genes, proteins in terms of name or symbols from text data to integrate the database of plant genes for several research purposes. Many different experiments, tests and optimization have been implemented during this project. The result of the model is positive for further research and at this moment, it can be considered as a good baseline method for our project.
In the future, we would like to deploy a couple of optimized ideas so as to improve

the result of the current work and start to implement the new methodology to not only improve the accuracy but also optimize the time computing. The following ideas could be tested.
For optimizing the accuracy and time-computing, during the process of building our own deep neural network model to extract biological entities from raw texts that include thousands of scientific paper, we want to apply high performance computing (HPC) on GPU to optimize the computing time of training process for a deep sequential model to handle a large dataset. There would be a challenge when applying parallel computing for the sequential model. However, it is a useful and exploitable direction for future development.

References

1. Basaldella, M., Furrer, L., Tasso, C., Rinaldi, F.: Entity recognition in the biomedical domain using a hybrid approach. Journal of biomedical semantics **8**(1) (2017) 51
2. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001)
3. Monaco, M.K., Stein, J., Naithani, S., Wei, S., Dharmawardhana, P., Kumari, S., Amarasinghe, V., Youens-Clark, K., Thomason, J., Preece, J., Pasternak, S., Olson, A., Jiao, Y., Lu, Z., Bolser, D., Kerhornou, A., Staines, D., Walts, B., Wu, G., D'Eustachio, P., Haw, R., Croft, D., Kersey, P.J., Stein, L., Jaiswal, P., Ware, D.: Gramene 2013: Comparative plant genomics resources. Nucleic Acids Research **42**(D1) (2014)
4. Swarbreck, D., Wilks, C., Lamesch, P., Berardini, T.Z., Garcia-Hernandez, M., Foerster, H., Li, D., Meyer, T., Muller, R., Ploetz, L., et al.: The Arabidopsis Information Resource (TAIR): gene structure and function annotation. Nucleic Acids Research **36**(Database issue) (jan 2008) D1009–1014
5. Habibi, M., Weber, L., Neves, M., Wiegandt, D.L., Leser, U.: Deep learning with word embeddings improves biomedical named entity recognition. Bioinformatics **33**(14) (2017) i37–i48
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8) (1997) 1735–1780
7. Graves, A., Jaitly, N., Mohamed, A.r.: Hybrid speech recognition with deep bidirectional lstm. In: Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on, IEEE (2013) 273–278
8. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional lstm networks. In: Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on. Volume 4., IEEE (2005) 2047–2052
9. Ling, W., Luís, T., Marujo, L., Astudillo, R.F., Amir, S., Dyer, C., Black, A.W., Trancoso, I.: Finding function in form: Compositional character models for open vocabulary word representation. arXiv preprint arXiv:1508.02096 (2015)
10. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 (2016)
11. Gene: Ontology consortium. `http://www.geneontology.org/`
12. Protein: Ontology. `https://pir.georgetown.edu/pro/`
13. Basaldella, M., De Nart, D., Tasso, C.: Introducing distiller: A unifying framework for knowledge extraction. IT@ LIA@ AI* IA **1509** (2015)