INTERNATIONAL JOURNAL OF ADVANCED
STUDIES IN COMPUTER SCIENCE & ENGINEERING 12/31/2020
IJASCSE VOLUME 9 ISSUE 12, 2021

A Noble Approach for Recognition and Classification of Agricultural Named Entities using Word2Vec

Payal Biswas
School of Computer and Systems Sciences
Jawaharlal Nehru University
New Delhi
payal.biswas138@gmail.com

Aditi Sharan
School of Computer and Systems Sciences
Jawaharlal Nehru University
New Delhi
aditisharan@gmail.com

Abstract- **Named Entity Recognition and Classification in agriculture domain is a recent field of research in Natural language processing. This paper proposes a noble idea to recognize and classify agriculture entities using an Word2Vec. The aim of the experiment is to recognize and classify three class of entities that is crop, fertilizer and pest from the agricultural text. Employing just two features namely context feature and Part-of-Speech feature the proposed model achieves a significant accuracy of 85.36% for crop, 74.15% for fertilizer and 83.12% for pest class. This paper also proposes an excellent technique to handle multi word terms, which was a big issue in previous related works.**

Keyword: Named entity recognition, context feature, word2Vec, entity classification, part of speech, word vector, natural language processing.

I. INTRODUCTION

Named entity recognition and classification is the basic building block for any language-based applications. The term "Named Entity" was first coined for Sixth Message Understanding Conferences (MUC-6) held in 1995 (Grishman and Sundheim, 1996). The aim was to recognise the names of place, person and organizations in newspaper article. Various work in NER for general domain (Lample *et al.* 2016, Ritter *et al.* 2011, May *et al.* 2003) and biomedical domain (Perera et al. 2020, Kim *et al.* 2012) has been done. However, very few works have been done in agriculture domain (Guo *et al.* 2020, biswas *et al.* 2019, Malarkodi *et al.* 2016). If we want to design any agricultural base NLP application such as agricultural question answering system or search engine for agricultural domain, we would first require to recognize the agricultural entities from the agricultural text. However, agriculture entities do not follow any specific pattern as in case of general domain and biomedical domain. This makes the agricultural NER as most challenging and exciting research topic in the field of natural language processing.

This paper proposed a noble technique for named entity recognition and classification in agricultural domain using word2vec. It is the most recent and significant technique to derive the relationships between words and its context.

II. RELATED WORK

Research in the field of "Named Entity Recognition" has been started more than two and half decades ago. Primarily the research has been done in three domains: General domain, biomedical domain and agriculture.

General Domain NER: Lisa F.Rau (1991) has published the first research paper in

NER using heuristics and handcrafted rules. S. Coates-Stephens (1992) and C. Thielen (1995) aimed to recognize "proper names" from the general text. Then after the work has been extended for fined grained or sub categories (Fleischman and Hovy 2002, Bick 2004, Witten *et al.,* 1999 etc.). Research in NER has also been done in various languages (May *et al.* 2003, Huang 2005) etc.

Biomedical Domain: An ample amount of work has also been done in biomedical domain to extract the entities like "protein", "RNA", "DNA" etc. (Settles 2004), "drug" (Rindflesch *el al.* 1999). Machine learning technique has also been used in biomedical NER (Crichton *et al*. 2017). Perera *et al.* (2020) worked over biomedical entity relations.

Agriculture Domain: NER over agriculture domain have been started just few years before. Biswas *et al*. (2015) proposes a basic framework to design an Agricultural NER called AGNER. Fined grained agricultural entities has been extracted by Malarkodi *et al.* (2016) using CRF. Biswas *et al.* (2016) proposed a WordNet based agricultural NER. A recent work in Agricultural NER has been presented by Biswas *et al.* (2019) using context feature.

III. PROPOSED WORK

In case of general domain NER and biomedical domain NER, various word level features and document level features like case feature, digit feature, prefix, suffix feature, punctuation feature play a vital role in extracting the named entities from the text data (Nadeau and Sekine, 2007). This is possible because the entities in these domains follow some specific structure. For an instance, name of a person place or organization always starts with capital letter and entities in biomedical domain have specific suffix or prefix for example sulph***ate***, phosph***ate***, nitr***ate*** etc. However, the nature of agricultural entities is completely different. Like proper nouns neither they always start with capital letter nor they follow any specific structure. This actually makes the Agriculture Domain NER as most challenging task.

Context Feature

It can be observed that context of a word plays a big role in finding the class or sense of a word. For an example, the context words: planet, satellite, universe etc. of a word ***star*** indicates that the word ***star*** is used as an astronomical object. While the context words movie, actor, actress reveals that the word ***star*** belongs to film industry. It has been seen that in case of agriculture domain also context could act as a good feature. Biswas *et al.* (2019) carried out agricultural named entity recognition using context feature and achieved a good result.

Word2Vec: Now a day Word2Vec is one of the most recent revolutions in the field of natural language processing (Mikolov *et al.,* 2013). Word2Vec derives relationships between a word and its context words. It is a method to represent a word mathematically in the form of vector. Using the concept of vector and cosine similarity between the vectors, Word2Vec finds the similarities and differences between the words in the corpus.

Word2Vec Vs One-hot encoding

One-hot encoding is also a simple and straight forward method to map a word to a vector. However, the reason behind using Word2Vec for entity extraction and classification in this work rather than one-hot encoding is that Word2Vec efficiently measure the semantic relationship between

words mathematically while one-hot encoding is unable to capture the semantic similarity and relationships between the words. Let us consider two statements to understand this:

Today's weather is good
Today's weather is nice

The vocabulary V would be
V = {Today's, weather, is, good, nice}
and in one-hot encoding the words are represented as:

Today's = {1,0,0,0,0}
Weather = {0,1,0,0,0}
is = {0,0,1,0,0}
good = {0,0,0,1,0}
nice = {0,1,0,0,1}

It shows that, the representation of words ***good*** and ***nice*** is different in one-hot encoding. That is although these two words are semantically same but they will occupy different dimension in the space.

Advancement of Word2Vec over one-hot encoding is that Word2Vec on the basis of context word creates a word embedding or word vectorization which represents each word numerically. After that it calculates the cosine of these vectors to find out the similarity or differences between each word in the corpus (Figure 1).

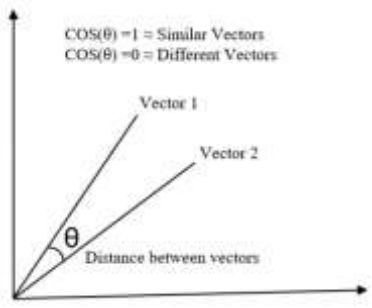

Figure 1. Cosine similarity/Difference of vectors

Thus, as context play a big role in predicting the class or sense of any word or entity and Word2Vec represents words using the context, this paper proposes a noble idea to extract and classify agricultural entities using the most recent Word2Vec technique.

Working Methodology

Named entity recognition in agriculture domain is a new and essential field of research in NLP. However still this field is unenlightened. A very small work has been done in agriculture domain NER. As this is recent field of research, no bench mark dataset is available. Moreover, certain works which has been done in last few years have prepared their dataset by their own and those datasets are even not public. Therefore, in order to perform the experiment, we also have to prepare our own dataset of agriculture domain.

Dataset Preparation: As data available in web is not considered as reliable, we have collected the data from an authenticate source of agriculture called AGRIS. It is a repository of abstracts of agricultural research papers and worldwide technical information on food and agriculture. AGRIS is maintained by the Food and Agriculture Organization (FAO) of the United Nations and is serving the users since 1974. In order to prepare the dataset, we made a list of agricultural entities including crop name, fertilizer name and pest names. Taking each entity from this list we have crawled abstract to prepare the dataset.

Implementation: Experiment starts with Pre-processing the agricultural data which has been crawled from AGRIS. Pre-processing includes: Noise Removal, replacing contractions, removing

punctuations, and transforming whole data into a single case (either upper or lower case) character. As many named entities in agriculture domain contains multi word terms like beet root, leaf roller, elephant beetle, finger millet, boric acid etc. These words need to be concatenated to make a single term. Otherwise, the co-term of the multiword term will be treated as context word of each other. In the data set these multi terms are concatenated using the concept of conditional probability.

A copy of this concatenated multiword dataset is passed to the *Stanford Parser* to obtain the Part-of-Speech (POS) tag of each word present in the dataset. From this POS tagged dataset a dictionary of unique words has been created containing unique words with their POS tags. This dictionary is kept to be used for mapping later on in the experiment. Another copy of the concatenated multiword dataset is exercised for tokenization and then normalization. This normalized data is then passed to the Word2Vec model to obtain a vector of similar words. After getting the vector space of similar words, three separate list of seed words from each class *i.e.*, crop fertilizer and pest has been taken. For each entity present in the list of seed words, we will extract the vector of similar words individually for each class.

Consider the elements present in the vector of similar words corresponding to each seed word as Candidate Similar Word (CSW). Out of these CSWs all are not *NOUN*. As it is known that named entities are proper nouns, we will check for the POS of these CSWs. Here, the dictionary of unique words and its POS which we have extracted at the time of beginning the experiment has been utilized. For a CSW, if all the values of POS in the Word-POS dictionary are found to be *NOUN* or variant of *NOUN*, then it is assigned same class as of its seed word and appended in the list of seed word of that particular class. All other CSWs whose any value of POS is not *NOUN* then it is discarded. This loop of finding vector of seed words, checking for POS and appending in the seed list continues until the graph of getting new entities gets constant.

Work Flow Diagram: The implementation of the experiment can be visualize using the flow diagram (Figure 2).

IV. EXPERIMENT AND RESULT

Dataset Preparation: The dataset has been prepared by crawling the abstract from AGRIS. To extract the abstracts, we have prepared three separate list of agriculture entity namely crop, fertilizer and pest using experts. The crop list contains 247 crops

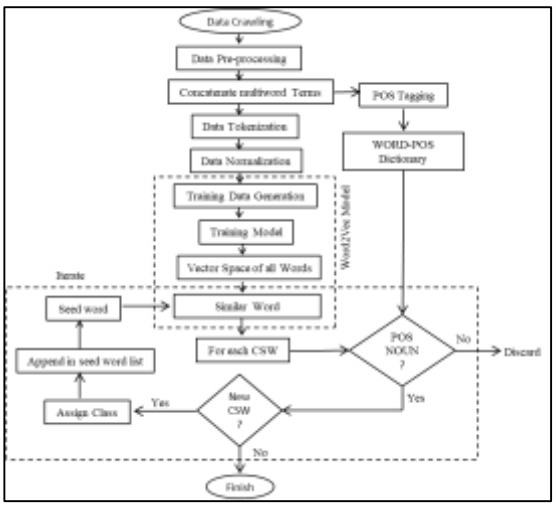

Figure 2. Work flow diagram

names, fertilizer list contains 50 fertilizers and pest list contains 41 pest names. For each entity we have crawled abstracts from around top ten links if available. Statistics of dataset is shown in the table [1].

Table 1. Statistics of Dataset

Entity Class	Abstracts	Lines	Words
Crop	2684	23507	540877
Fertilizer	522	6666	154997
Pest	359	7309	163597

After pre-processing the data, we need to handle the multiword terms present in the dataset. To do so we have used the concept of conditional probability.

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Consider an example of *beet root.* We will compute two conditional probabilities:

$$P_1(beet|root) = \frac{P(beet \cap root)}{P(root)}$$

$$P_2(root|beet) = \frac{P(beet \cap root)}{P(beet)}$$

P_1 shows probability of occurrence of beet given root while P_2 shows probability of occurrence of root given beet. If any one of these probabilities have value 1 then we consider these two terms as a single word. Once after combining all the multiword terms into single word, tokenization and normalization is to be done to remove the stop words and perform stemming to get a uniform dataset. This data is then passed to the Word2Vec model. At first the training data is generated using the function:

generate_training_data (corpus, setting)

among the two parameters *corpus* is the dataset which we have passed to the Word2Vec and *settings* includes window size, embedding size, epochs and learning rate. In this experiment we kept settings as: window_size, w = 5, word_embedding, n = 100, epochs = 50, learning rate- 0.01. Using the specified settings, the function *generate_training_data* will produce the one-hot representation for each word in the corpus which is then get trained for the Word2Vec model. To train the model we randomly generate two weight matrixes. During training the model these weights gets adjusted using forward and back propagation. Once model gets trained after 50 epochs, we can get a vector corresponding to each word in the corpus. For example, the snap shot of the vector for the word *mango* is shown in the figure [3].

```
print(model['mango'])
```

```
[ 0.7717826  -0.09680894 -0.53725994  0.58440185  0.18916759  0.20541416
  0.5079394  -0.24158813  0.3666595   0.41585237  0.19434093  0.12217779
  0.51637995  0.7003596  -0.24215153  0.15510581 -0.507016   -0.01357721
  0.11747257  1.2838497  -0.26566377  0.41802356 -0.16978003 -0.38482893
  0.18593766 -0.30251533 -0.6603457   0.93956745  0.74871445 -0.9630603
  0.9405049  -0.10778298  0.519697    -0.72999126 -0.8117282  -0.3401303
 -0.8092354  -0.01007729 -0.34434685 -0.16592875  0.5247965   0.31702772
 -0.2562897   0.2642651   0.2181076   0.8148505   1.0059565  -0.37365398
  0.36413002  1.0094469   0.33085358 -0.1806776   0.5088099  -0.5573319
 -0.40753165 -0.3846438   1.1176488  -0.18205532  0.05044159  0.5238988
  0.19067347 -0.1283647   0.3733703   0.29654813  0.728616    0.0138044
 -0.54930875  0.08400822  0.32573822 -0.30173167  0.47902292 -0.28692296
  0.2114214  -0.33058032  0.15065163 -0.5464927   0.67150116  0.40104735
 -1.0366374  -0.5288773  -0.3921134  -0.3481229  -0.31426528  0.23069297
 -0.7476015   0.0456914  -0.7725047  -0.5352467   0.32538036 -0.39657947
  0.43474895  0.0098058   0.6760211   0.555018   -1.4293709   0.1734576
  0.06192533  0.24749646 -0.5980077   0.5599987 ]
```

Figure 3. Word embedding of term "*mango*"

As we get the vector for each word in the corpus, we can find the similar words using the function:

model.most_similar (entity)

It will return the words which are similar to the word passed as a parameter to the function. Screen shot of the list of similar words for the word *cabbage* is shown in figure [4].

```
model.most_similar('cabbage')
```

```
[('recording', 0.9869822263717651),
 ('cucumber', 0.9807466268539429),
 ('tallest', 0.9798338413238525),
 ('radish', 0.9790763854980469),
 ('cobs', 0.9788035154342651),
 ('emerged', 0.9779913425445557),
 ('pod', 0.9776888489723206),
 ('survived', 0.9775211811065674),
 ('dbm', 0.9772588014602661),
 ('seedlings', 0.9763957858085632)]
```

Figure 4. List of similar words for entity cabbage

Once the vector space of similar words is obtained, we will create a list of seed words. Table [2] shows the list of seed words.

Table 2. List of seed words

Entity Class	Seed Words
Crop	rice, mango, finger_ millet, cabbage, rose
Fertilizer	Urea, Sulphur, Boric_ Acid, Borax, bone_meal
Pest	looper, mealybug, termite, weevel, longhorn_beetle

For each seed word we will find the similar vector separately for each class. Corresponding to each seed word the elements which we will get, are expected to be candidate similar words (CSW). As we can see in figure [4] that all the CSWs are not named entities and it is known that named entities are *NOUNs*, hence we will match for the POS of these CSWs with the word_POS dictionary. For those CSWs whose POS is NOUN is appended in the seed word list of same class as that of seed word and discard the others. For example, in figure [4] cucumber, radish and cob will get appended in the seed word list of crop class and other words will get discarded. This process continues until we stop getting new words (Figure 5).

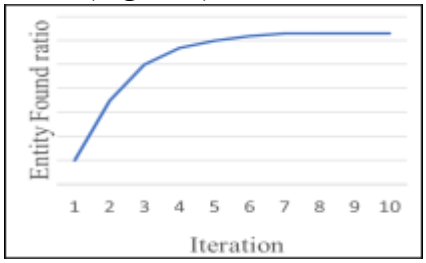

Figure 5. Entity found ration Vs Iteration

Accuracy has been calculated separately for each class in presence of an expert using the confusion matrix (Table 3).

Table 3. Confusion Matrix

		Named Entity	
		Yes	No
Classified Correctly	Yes	TT	TF
	No	FT	FF

TT: Named entity classified correctly
TF: Non named entity classified correctly
FT: Named entity not classified correctly
FF: Non named entity not classified correctly

For crop and pest class average accuracy is found to be 85.36% and 83.12% respectively. While for fertilizer class the accuracy is little bit low *i.e.*, 74.15% as compared to crop and pest. The reason behind this is that most of the fertilizer are multiword terms and those are overlapping like ammonium sulphate, ammonium nitrate, potassium sulphate etc. Due to their overlapping nature, they could not be concatenated and hence treated as a separate word which reduces the accuracy. The value of precision, recall and F-Score separately for crop, fertilizer and pest is shown in figure [6]. The experiment has been repeated for ten different set of seed word list. Figure [7] shows the box plot for different values of precision recall and F-score for crop, fertilizer and pest. Figure [7] indicates the distribution of experimental values of precision, recall and F-Score for each iteration with different set of seed words. The results which we have obtained is upgraded as compared to previous attempts in Agricultural NER (biswas *et al.* 2016, biswas *et al.* 2019).

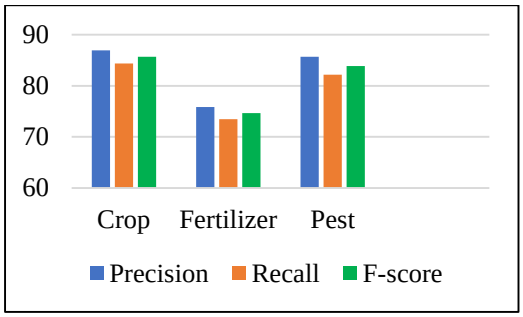

Figure 6. Precision Recall and F-Score

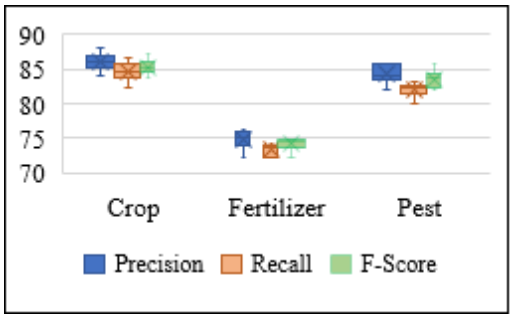

Figure 7. Entity found ration Vs Iteration

V. CONCLUSION

Agricultural NER is the most challenging and exciting research topic in the field of natural language processing. In this work we have taken three classes of agricultural entities namely crop, fertilizer and pest. The paper presented an admirable method to design agricultural NER using an outstanding Word2Vec technique. We have created agriculture dataset using AGRIS. The problem of multi word terms has also been addressed here. The proposed approach acquires an accuracy of 85.36%, 74.15%, and 83.12% for crop, fertilizer and pest respectively. The result is appreciable in the present state of art.

VI. FUTURE WORK

Named Entity Recognition (NER) in agriculture domain is a recent field of interest in NLP. As discussed earlier various work has been done in general and biomedical domain, but a few works have been done in agriculture domain, thus a huge scope is present in Agriculture NER (AGNER). It could be appended for various entity class or sub-classes of agriculture domain. Discrete feature set could also be explored which may be implemented over agriculture domain. Several machine learning techniques could be applied for designing AGNER.

Reference

Bick, E. (2004, May). A Named Entity Recognizer for Danish. In Language Resources and Evaluation.

Biswas, P., Sharan, A., & Kumar, A. (2015, March). AGNER: Entity tagger in agriculture domain. In *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 1134-1138). IEEE.

Biswas, P., Sharan, A., & Kumar, A. (2019). Context Pattern Based Agricultural Named Entity Recognition. *Research in Computing Science*, *148*, 383-399.

Biswas, P., Sharan, A., & Verma, S. (2016). Named entity recognition for agriculture domain using word net. *Int J Comput Math Sci*, *5*(10), 29-36.

Coates-Stephens, S. (1992). The analysis and acquisition of proper names for the understanding of free text. *Computers and the Humanities*, *26*(5-6), 441-456.

Crichton, G., Pyysalo, S., Chiu, B., & Korhonen, A. (2017). A neural network multi-task learning approach to biomedical named entity recognition. *BMC bioinformatics*, *18*(1), 368.

Fleischman, M., & Hovy, E. (2002). Fine grained classification of named entities. In *COLING 2002: The 19th International Conference on Computational Linguistics*

Guo, X., Zhou, H., Su, J., Hao, X., Tang, Z., Diao, L., & Li, L. (2020). Chinese agricultural diseases and pests named entity recognition with multi-scale local context features and self-attention mechanism. *Computers and Electronics in Agriculture, 179, 105830.*

Grishman, R., & Sundheim, B. M. (1996). Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Huang, F. (2005). Multilingual named entity extraction and translation from text and

speech (Doctoral dissertation, Carnegie Mellon University, Language Technologies Institute, School of Computer Science).

Kim, J., Ko, Y., & Seo, J. (2019). A bootstrapping approach with CRF and deep learning models for improving the biomedical named entity recognition in multi-domains. *IEEE Access, 7*, 70308-70318.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360.*

Malarkodi, C. S., Lex, E., & Devi, S. L. (2016). Named Entity Recognition for the Agricultural Domain. *Res. Comput. Sci.*, *117*, 121-132.

May, J., Brunstein, A., Natarajan, P., & Weischedel, R. (2003). Surprise! What's in a Cebuano or Hindi Name?. *ACM Transactions on Asian Language Information Processing (TALIP)*, *2*(3), 169-180.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, *30*(1), 3-26.

Perera, N., Dehmer, M., & Emmert-Streib, F. (2020). Named Entity Recognition and Relation Detection for Biomedical Information Extraction. *Frontiers in Cell and Developmental Biology*, *8*, 673.

Rau, L. F. (1991, January). Extracting company names from text. In *Proceedings The Seventh IEEE Conference on Artificial Intelligence Application* (pp. 29-30). IEEE Computer Society.

Rindflesch, T. C., Tanabe, L., Weinstein, J. N., & Hunter, L. (1999). EDGAR: extraction of drugs, genes and relations from the biomedical literature. In *Biocomputing 2000* (pp. 517-528).

Ritter, A., Clark, S., & Etzioni, O. (2011, July). Named entity recognition in tweets: an experimental study. *In Proceedings of the 2011 conference on empirical methods in natural language processing* (pp. 1524-1534).

Settles, B. (2004). Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)* (pp. 107-110).

Thielen, C. (1995). An approach to proper name tagging for german. *arXiv preprint cmp-lg/9506024*.

Witten, I. H., Bray, Z., Mahoui, M., & Teahan, W. J. (1999, June). Using language models for generic entity extraction. In *Proceedings of the ICML Workshop on Text Mining* (p. 14).