Named Entity Recognition with Word Embeddings and Wikipedia Categories for a Low-Resource Language

ARJUN DAS, University of Calcutta
DEBASIS GANGULY, Dublin City University
UTPAL GARAIN, Indian Statistical Institute

In this article, we propose a word embedding–based named entity recognition (NER) approach. NER is commonly approached as a sequence labeling task with the application of methods such as conditional random field (CRF). However, for low-resource languages without the presence of sufficiently large training data, methods such as CRF do not perform well. In our work, we make use of the proximity of the vector embeddings of words to approach the NER problem. The hypothesis is that word vectors belonging to the same name category, such as a person's name, occur in close vicinity in the abstract vector space of the embedded words. Assuming that this clustering hypothesis is true, we apply a standard classification approach on the vectors of words to learn a decision boundary between the NER classes. Our NER experiments are conducted on a morphologically rich and low-resource language, namely Bengali. Our approach significantly outperforms standard baseline CRF approaches that use cluster labels of word embeddings and gazetteers constructed from Wikipedia. Further, we propose an unsupervised approach (that uses an automatically created named entity (NE) gazetteer from Wikipedia in the absence of training data). For a low-resource language, the word vectors obtained from Wikipedia are not sufficient to train a classifier. As a result, we propose to make use of the distance measure between the vector embeddings of words to expand the set of Wikipedia training examples with additional NEs extracted from a monolingual corpus that yield significant improvement in the unsupervised NER performance. In fact, our expansion method performs better than the traditional CRF-based (supervised) approach (i.e., F-score of 65.4% vs. 64.2%). Finally, we compare our proposed approach to the official submission for the IJCNLP-2008 Bengali NER shared task and achieve an overall improvement of F-score 11.26% with respect to the best official system.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Named Entity Recognition*

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Word embedding, CRF-based NER, Wikipedia-based NER, unsupervised NER, language-independent NER, classifier

The second author is presently affiliated to the IBM Research Labs, Dublin.

A. Das was supported by the Indian Statistical Institute, Kolkata, India. This research is partly supported by the Science Foundation Ireland (SFI) as a part of the ADAPT Centre at DCU (Grant No. 1713/RC/2106). We are extremely grateful to the anonymous reviewers who have provided valuable feedback to greatly improve this paper.

Authors' addresses: A. Das, Department of Computer Science and Engineering, University of Calcutta, JD-2, Sector – III, Saltlake City, Kolkata-700106, India; email: arjundas.cs@gmail.com; D. Ganguly, ADAPT Centre, School of Computing, Dublin City University, Dublin, Ireland; email: debasis.ganguly1@ie.ibm.com; U. Garain, Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, 203, B.T. Road, Kolkata 700108, India; email: utpal@isical.ac.in.

DOI: http://dx.doi.org/10.1145/3015467

ACM Reference Format:
Arjun Das, Debasis Ganguly, and Utpal Garain. 2017. Named entity recognition with word embeddings and Wikipedia categories for a low-resource language. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 16, 3, Article 18 (January 2017), 19 pages.
DOI: http://dx.doi.org/10.1145/3015467

1. INTRODUCTION

Named entity recognition (NER) is an important natural language processing (NLP) problem. NER involves classifying a named entity (NE) into one of the predefined labels such as *person name*, *location name*, and *organization name*. A considerable amount of research has been done on NER, particularly for English [Bikel et al. 1999; Mikheev et al. 1999; Borthwick 1999; Zhou and Su 2002; McCallum and Li 2003]. The most simple approach to NER is to take the help of gazetteers—lists of names of people, organizations, locations, and other NEs. The NER then is just a simple list lookup to markup known strings in the text. The major limitation of this approach is that this list has to be considerably large in size for the NER to work satisfactorily well, because in this approach there is no way to recognize unknown names. One of the bottlenecks in designing NER systems with gazetteers is the limited availability of large gazetteers, particularly for different languages [Cucchiarelli et al. 1998].

Since context often plays an important role in determining an NE, a standard approach to NER is to treat it as a sequence labeling task. Broadly speaking, this involves training a probabilistic model such as the hidden Markov model (HMM) [Bikel et al. 1999] or a conditional random field (CRF) [McCallum and Li 2003] on a manually annotated list of documents. The training phase makes use of the contextual information of the NE classes and uses this information in the testing phase to recognize unknown NEs. A limitation of this approach is that the training data needs to be sufficiently large for the probabilistic models to yield effective results. However, the availability of a manually annotated dataset of a large quantity is a problem for low-resource languages. Moreover, the standard sequence labeling approach can also yield poor results for highly agglutinative languages due to the possibility of occurrence of several inflected NE forms [Singh 2008].

Since supervised approaches do not lead to satisfactory results for low-resource and morphologically rich languages, a possible solution, which works well in practice, is to use automatically constructed gazetteers from Wikipedia category pages (e.g., Wikipedia provides a list of people's names according to their nationalities).[1] Although this method works well for Wikipedia pages with sufficient resources [Kazama and Torisawa 2007], it may not lead to good results for languages with a low number of Wikipedia articles (e.g., see the study reported in Richman and Schone [2008] about how the size of a Wikipedia resource in a language can affect NER effectiveness). In fact, it was reported by Richman and Schone [2008] that the NER results for Ukranian, a low Wikipedia resourced language, is worse in comparison to other relatively high resourced languages, such as Spanish and French.

To alleviate the problems of low-resource availability for training a sequence labeling model, we propose to use vector embeddings of words that can be obtained in an unsupervised manner from a collection of documents [Mikolov et al. 2013b]. Our method intends to use the clustering hypothesis, according to which the relative distance between two NEs of the same category (e.g., *person name*) is lower than those between NEs in different categories. If this clustering hypothesis is true, an intuitive approach to NER is then to learn a decision boundary to distinguish between the NE classes. For instance, Figure 1(a) represents a two-dimensional representation of NE

[1] http://en.wikipedia.org/wiki/Lists_of_people_by_nationality.

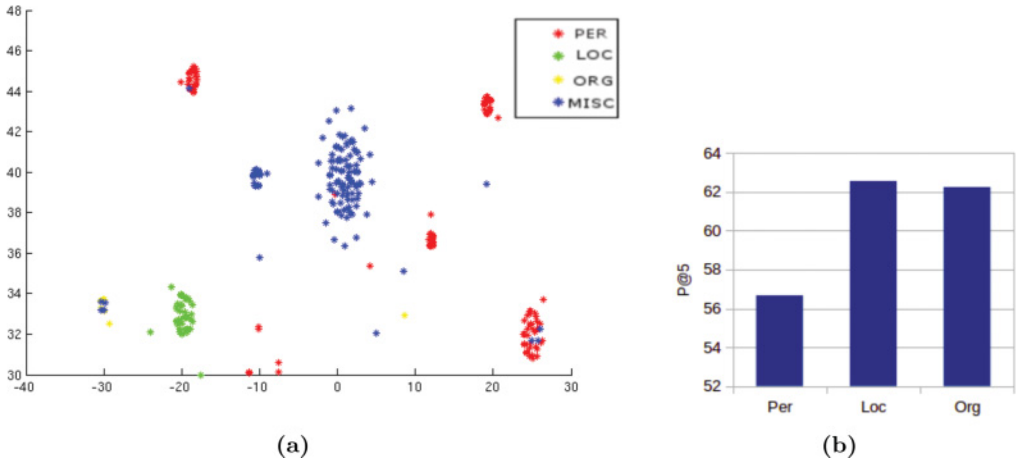

Fig. 1. (a) Two-dimensional representations of NE word vectors of corresponding NE words obtained with training word2vec on the IJCNLP 2008 Bengali NER shared task training data. (b) Average Precision@5 values for three NNP (i.e., proper noun) categories within a 5 neighborhood.

vectors from the IJCNLP-2008 Bengali NER shared task training dataset (more details are provided later in Table II) using *tSNE* [van der Maaten and Hinton 2008], where each color represents a particular NE class. Figure 1(a) reveals that the hypothesis is apparently true, which motivates us to believe that this method of doing NER could potentially work well. Such a word embedding–based classification approach makes use of the context in a different way than a sequence labeling method. We would like to emphasize that the context information in this case is implicitly used in the word vectors themselves since context information plays a significant part in deriving the vector representations.

In particular, the first step in our proposed approach involves deriving the word embeddings using a set of documents. The next step is to use the word embeddings of known NEs (which may be obtained from a training set or an automatically constructed gazetteer using an external resource, e.g., Wikipedia categories) and their associated NE class labels to learn a decision boundary. Once the decision boundary is estimated, the model will be able to distinguish unknown NEs into one of the NE classes. It is important to note that although this approach requires a training set as well (similar to a sequence labeling method), the amount of training data needed to effectively train a decision boundary potentially is much less than training a sequence labeling method.

Further, we demonstrate a novel way of doing NER in complete absence of training data. For instance, the training set or points can automatically be constructed from Wikipedia (hence, in principle, our NER method is unsupervised). Generally speaking, Wikipedia for a low-resource language contains a smaller amount of information in comparison to high resourced languages, such as English, French, and Spanish. Therefore, the number of training examples for NEs constructed from Wikipedia for a low-resource language is much less in comparison to a resourceful language. This small number of training examples is often insufficient for training an NER model. To mitigate this drawback of resource scarcity, we make use of the clustering hypothesis to further expand the NEs obtained from Wikipedia.

In this article, we investigate whether the clustering hypothesis can be used to effectively train a decision boundary and outperform traditional sequence labeling approaches. We demonstrate the hypothesis that our proposed method can outperform the standard sequence labeling approaches for NER in the case of marginal training data (and even without the presence of training data altogether). For our experiments, we focus on a resource poor language, namely Bengali, which is a South Asian language. Considerable amount of research has been done on NER, especially for English [Bikel et al. 1999; Borthwick 1999; Zhou and Su 2002; McCallum and Li 2003]. However, the aforementioned research cannot be directly extended for Bengali, which unlike English poses some extra challenges due to some linguistic characteristics of Bengali, such as noncapitalized NEs, frequent use of dictionary words as NEs, morphological richness, and free ordering. Although we conduct experiments for NER in Bengali, we would like to emphasize that our proposed methodology is not comprised of any language-specific assumptions and can be applied for any other languages as well.

The rest of the article is organized as follows. In Section 2, related works are reviewed. Our method for word vector–based NER is presented in Section 3. Section 4 describes our experimental protocol, and the results, related discussion, and comparisons are presented in Section 5. Finally, Section 6 concludes the article and offers some directions for future work.

2. RELATED WORK

We group this section into two subsections where we first present the works related to word embedding–based NER. This is followed by a description of NER methods for low resource languages.

2.1. Word Embedding–Based NER

A recent work that utilizes word vector embeddings for NER is that of Demir and Ozgur [2014]. This approach first embeds every word as a vector using the word2vec[2] proposed by Mikolov et al. [2013b]. Next, it uses the word clusters as one of the features (along with other features, e.g., word context and POS tag context) to train a neural network (NN) classifier. The major limitations of this approach are as follows. First, it requires a training set to train a classifier, whereas in our work we show that using externally available resources such as the Wikipedia categories produces effective results. Second, the authors do not compare their results to a state-of-the-art NER technique, such as the CRF, whereas we use CRF as one of our baselines and show that our proposed method NER for low-resource languages significantly outperforms the CRF-based approach. Third, instead of using cluster labels as features (which suffers from the disadvantage of choosing the optimal number of clusters), we rely on a standard classification-based approach, using either training resources (if available) or automatically constructed gazetteers from Wikipedia. Similar to Demir and Ozgur [2014], the work of Turian et al. [2009] also uses the cluster labels as features and requires a training set and does not compare the results to a CRF.

Similar to our work, the approach reported by Guo et al. [2014] uses the skip-gram model with negative sampling [Mikolov et al. 2013b] for inducing vector embeddings for words. They perform K-means clustering to derive word clusters and use the cluster labels along with word context features to train a CRF. The main difference between their work and ours is that first we use the word vectors themselves as features instead of relying on a wide range of extracted features, and second, our proposed method does not require manually annotated data for training that is not readily available in sufficient quantities for resource-poor languages.

The work of Passos et al. [2014] uses word embeddings as features in a CRF classifier. They also use the Wikipedia English corpus to train word2vec for obtaining the vector representations from the words. However, similar to the approaches discussed

[2] The name *word2vec* comes from the name of the software released by Mikolov et al. [2013b] and is available at https://code.google.com/p/word2vec/.

previously (i.e., Demir and Ozgur [2014]; Turian et al. [2009]), this work also uses cluster label–based features instead of using the vectors themselves as features in a classifier. Moreover, the use of CRF depends on the availability of a training set and handcrafted language-dependent features, which are not prerequisites in our case.

2.2. NER for Low-Resource Languages

It is well known that the gazetteer (or entity dictionary) plays a very important role in the NER task in a low-resource setting (where training data is not sufficient to train a classifier). When preparing the gazetteer for a language, one common approach is to consult Wikipedia pages [Toral and Muñoz 2006]. The number of Wikipedia pages in any low-resource language is much lower than that of a high resourced one (e.g., English). For instance, as of August 2015, the number of English Wikipedia pages was 4,958K, whereas the number in Bengali was only 33K. A gazetteer constructed from Wikipedia would be less effective for Bengali NER mainly because of the absence of sufficient training examples.

Another limitation of a gazetteer is its static nature. The usual practice is to develop a gazetteer and use it for training a statistical-based classification method [Saha et al. 2008]. This approach would obviously fail to utilize the dynamic nature of Wikipedia with new pages (and hence new NEs) being added continuously. The only way to take into account the additional content is periodic updating of the gazetteer and retraining of the classifier.

Apart from preparing gazetteers, Wikipedia has been used for annotating multilingual NER datasets [Richman and Schone 2008]. The work of Richman and Schone [2008] involves manually collecting a small list of Wikipedia categories corresponding to the NEs and then using these category lists to identify the NEs in the title of an Wikipedia page. The work in Zhang and Iria [2009] also involves expanding a small list of specific NE seed entries using Wikipedia. The category labels of a Wikipedia page have also been used as features of a CRF-based NER system [Kazama and Torisawa 2007].

In our work, we make use of the Wikipedia categories to automatically extract the training set, the vector embeddings of which can be used subsequently to train the classification model to recognize other unknown NEs.

In relation to gazetteer preparation, McCallum and Li [2003] used the Web for augmenting additional information for a low-resource scenario. Their idea is based on the assumption that if a particular name (e.g., a golfer's name, say Arnold Palmer) taken from the training examples (labeled NER data) is searched on the Web, a large list of other golf players can be retrieved. These retrieved NEs from the Web can then be used for augmenting the lexicon of entities. Evans [2003] used the Web in a different way, where a capitalized phrase found in a document is searched on the Web to find potential hypernyms of the phrase. These hypernyms are then clustered to derive a typology of NEs for the document. The hypernyms of the capitalized phrases are eventually used to classify them with respect to this typology. The method is tested on a small corpus and is found to be quite effective for doing NER in the open domain.

3. WORD EMBEDDING–BASED NER

In this section, we first motivate the use of a word embedding technique for NER. We then describe our proposed methodology in detail.

3.1. Motivation

Recent advances in word embedding derivation techniques using NN-based language models has gained a lot of interest in the NLP community [Mikolov et al. 2013b]. Intuitively speaking, such word embedding techniques tend to embed two word vectors

$\vec{t}$ and $\vec{t'}$ corresponding to the words t and t', close in an abstract space of N dimensions (the number of dimensions corresponds to the number of output layers in the NN). If the terms t and t' have similar contexts and vice versa (i.e., the contexts in turn have similar words), then the vectors corresponding to the terms are placed in close distance [Goldberg and Levy 2014].

The objective function for learning the embedding of a word ensures maximizing the similarity of the current word with other words within its close proximity and minimizing the current word's similarity with several randomly sampled words from outside the context of the current word, a process called *negative sampling* [Mikolov et al. 2013b; Goldberg and Levy 2014]. It is reported that this process of negative sampling produces reliable word embeddings in a very efficient manner [Mikolov et al. 2013b] compared to other word embedding approaches such as those of Mnih and Hinton [2008] and Collobert and Weston [2008].

The motivation for using word vector embeddings is that words with identical NE classes should tend to co-occur in close proximity in the abstract space of N dimensions. In fact, previous research, such as that of Passos et al. [2014] and Ratinov and Roth [2009] made use of the clustering hypothesis in a different way by using the word vector cluster labels as features in a classifier, such as CRF. In this article, we argue that there may be no need to use the interword distances or the cluster labels explicitly as classifier features. Instead, it is much simpler (and much more efficient) to use the vectors themselves as features. In fact, it was reported by Guo et al. [2014] that directly using the continuous vectors as real valued features does not produce the best results. Guo et al. [2014] therefore propose a threshold-based binarization method to obtain a discrete feature vector representation, which is reported to work marginally better (from 86.21% to 86.75%). We argue that applying a state-of-the-art classifier, such as a random forest (RF) [Fernández-Delgado et al. 2014], may in fact be a more effective way to learn a decision boundary than binarizing the feature vectors in an ad hoc way.

3.2. Proposed Methodology

The first step in our proposed method is to obtain effective word and phrase embeddings. Embedding of a phrase is done by first identifying a sequence of words as a candidate phrase and then treating it as separate unit (or a pseudoword) in the word2vec word embedding algorithm [Mikolov et al. 2013b]. In Section 3.2.1, we describe the phrase identification process that we used for our experiments.

An intuitive assumption for NER-based word embedding is that the words in this abstract space obey the cluster hypothesis—that is, a person's name should be in close proximity to another person's name and far apart from a location name. In fact, our initial experiments with word vector embeddings (more details are provided in Section 4) revealed that this hypothesis is reasonably true. This can be seen in Figure 1(a), which shows a two-dimensional representation of NE vectors using tSNE [van der Maaten and Hinton 2008]. It can be observed that the relative distance between two NEs of the same category (e.g., *person name*) is lower than those between NEs in different categories. Moreover, the precision values in a 5-NN neighborhood around each word class are reasonably satisfactory, as shown in Figure 1(b).

To use this cluster hypothesis for classification, it is important to obtain a list of training samples (i.e., points with class labels where each label refers to a particular NE class, e.g., name, location). To approach the NER problem from a multi-class classification problem perspective, we presuppose the existence of a POS tagger[3] that annotates some of the words as NNPs. Obviously, the effectiveness of the NER

[3] For this experiment, we use the IJCNLP 2008 Bengali shared task dataset, which comes equipped with the POS information.

approach would then depend on the POS tagging accuracy, which in fact is reported to be quite satisfactory for Bengali (about 92% accuracy) [Ekbal et al. 2009].

The number of classes that one needs to consider in this multiclass classification problem is one more than the number of NE classes that needs to be recognized. This additional class represents the complementary class of other words tagged as NNPs but which do not belong to any of the NE classes (e.g., *person name*, *location name*). If a training set is available (in the supervised case), we can label each of the NNP tagged word vectors with its corresponding NE class from the training set annotation. These NNP tagged word vectors, along with their class labels, act as the training set for our classifier, where the objective is to estimate a decision boundary to classify unknown words into one of these classes (including the "other NNP" class) based on its proximity to any particular NE class.

Treating the task as a classification problem without considering the contextual information may indicate worse performance than standard NER approaches, such as CRF. However, in our proposed method, we make use of the contextual information in an implicit way.

Recall that the process of deriving the word embeddings itself takes into account the context of the words [Goldberg and Levy 2014], and hence the distance measured in the abstract space of NNP vectors should indicate contextual similarity between the NNPs. One more advantage of treating this task as a multiclass classification problem, rather than a sequence labeling task, is that the amount of training data needed to train a classifier is considerably lower than that to train a sequence labeling task. In fact, we empirically validate this hypothesis in Section 5, where we show that a CRF does not perform well with a small amount of training data.

However, treating the problem as a multiclass classification task brings us back to the presupposition of the availability of training data, which in fact is not available in sufficient quantities for a resource poor language. For example, the amount of the training dataset released as a part of the IJCNLP 2008 Bengali NER shared task is much lower than the benchmark English training (i.e., the train and development set combined together) dataset released as a part of the CoNLL 2003 NER shared task (the number of sentences is 18,453 vs. 6,030, respectively). A way to alleviate this training set preavailability problem is to construct the training set automatically from external resources such as Wikipedia categories, which we describe in Section 3.3.

3.2.1. Phrase Identification. Phrases are group of words that appear frequently together to form a single meaning. To identify phrases, we consider the approach of Mikolov et al. [2013b]. Specifically, we find words that appear frequently together and infrequently in other contexts. We use bigram and unigram counts from a corpus to identify multiword phrases from a corpus with the help of Equation (1).

$$Score(w_i, w_j) = \frac{count(w_i w_j) - \delta}{count(w_i) \times count(w_j)} \tag{1}$$

In Equation 1, δ works as a discounting coefficient that prevents infrequent words from becoming a phrase. Word pairs with scores above a particular threshold are identified as bigram phrases. We choose 200 as our initial threshold value for identifying bigram phrases. We group this bigram phrases into a single word and then continue the phrase identification process up to four times with decreasing threshold values. We decrease threshold values successively to restrict the number of longer phrases with infrequent occurrence.

Our method for phrase identification works best with a large corpus to discover meaningful phrases. Unlike other supervised methods for phrase identification, our method does not depend on large quantities of chunk annotated training data, which is difficult to acquire for a resource poor language such as Bengali. In fact, this unsupervised method for phrase identification is best suited for our NER approach because the primary focus of our NER approach is to classify NNPs, and hence the accuracy of the task is not affected by the presence of invalid phrases that are mostly non-NNPs. For recognition of multiword NNP, our approach yields precision, recall, and F1-measure values of 72.39%, 79.36%, and 75.71%, respectively.

3.3. Obtaining Training Points from Wikipedia

In this section, we explain our procedure of constructing the training examples (points) or gazetteers (for the *person*, *location*, *organization*, and *miscellaneous* NE classes) from Wikipedia in the case of marginal training data (and even without the presence of training data altogether). Wikipedia maintains a list of category for each of its title pages. Previously, researchers [Toral and Muñoz 2006; Zhang and Iria 2009; Richman and Schone 2008] made use of these category labels for gazetteer construction for the purpose of NER. For example, the Wikipedia categories “Living people,” “Birth,” and “Player” refer to the names of persons, whereas the category “Cities in India” refers to a location.

Since our NER experiments are conducted on a resource poor language, namely Bengali, we collect a list of seed NEs (for *person* and *location* categories) from Bengali Wiktionary.[4] For the *organization* category, we manually constructed a list (of 50) seed NEs from Wikipedia.

In our work, we have not considered a seed list for the *miscellaneous* class, which is a special NE class that can have instances of a multiple number of other NE classes within it. This make it difficult to manually construct a seed list comprising instances of the miscellaneous type. Consequently, we designed our algorithm in a way that it can take the seed list for the NE classes without considering the *miscellaneous* class.

We expand the seed list of each category with the help of Wikipedia category labels. We then search each seed NE in Wikipedia and extract the category information to build list of categories according to the NE’s corresponding NE class. For example, for the NE class *person*, its category list may contain categories such as “poet,” “singer,” “doctor,” “actor,” and “famous.” Similarly, for the category *location*, its list may contain categories such as “country,” “lane,” “river,” “hill,” and “famous.”

The next step in our algorithm is to remove the ambiguous category labels—that is, category labels that are present in more than one list, such as the category “famous” in our example. A list of such ambiguous categories is presented in Appendix. The purpose of removal of ambiguous category labels is to reduce confusion during the training phase of the classifier.

From this list of unique category labels pertaining to each NE class, we extract list of Wikipedia titles and their category labels using the complete Bengali Wikipedia dump[5] of 27,823 articles. Next, we search the category labels in the NE class-wise unique category list.[6] The unique list (say, the list corresponding to the *person* class) having the maximum matches is assigned as a NE class of that particular NE.

For example, consider the Wikipedia page of a famous Indian cricket player “Sachin Tendulkar” and its category labels, such as “births,” “cricketers,” and “people.” Our algorithm searches the category labels of “Sachin Tendulkar” in the NE class-wise

[4] https://en.wiktionary.org/wiki/Category:bn:Place_names; https://en.wiktionary.org/wiki/Category:Bengali_given_names.

[5] https://dumps.wikimedia.org/.

[6] The NE class-wise unique category list is available at https://drive.google.com/open?id=0BzVgTFHeBNsqb3pJeDh5TUZXSGM.

Table I. Example of Named Entity Instances Extracted from Wikipedia and the Five Nearest Neighbors of Each Instance in the Embedded Vector Space of the Words

NE Type	Wiki-Extracted NE	Topmost Five Similar NEs
Person	দিলীপ (Dilip)	স্বপন (Swapan), শ্যামল (Shyamal), প্রবীর (Prabir), সমীর (Samir), সুজিত (Sujit)
Location	ফ্রান্স (France)	জার্মানি (Germany), ইতালি (Italy), স্পেন (Spain), গ্রিস (Greece), সুইডেন (Sweden)
Miscellaneous	এপ্রিল (April)	মার্চ (March), মে (May), জুন (June), ফেব্রুয়ারি (February), জুলাই (July)

Note: English translations are shown within parentheses.

unique category list and finds that the list corresponding to the *person* class has the maximum number of matches. Consequently, our algorithm classifies “Sachin Tendulkar” as *person*. The number of NEs collected in this way for each category is shown in the rightmost column (i.e., the column named “Wiki 4 Tags”) of Table III. After expansion, our list contains 5,193 person names, 4,229 location names, and 795 organization names (see Table III).

Since we treat the NER problem as a classification problem, we also need an *other NNP* or *misc* class in addition to the person, location, and organization names. Any identified NE in the title of a Wikipedia page for which we do not find a category match is put into this class. The word embeddings of the NEs in each list along with their class labels (NE category) are used as training points in our proposed approach.

3.4. Word Vector Expansion of Wikipedia NEs

In Section 3.3, we described the procedure of retrieving training examples (points) from Wikipedia. In this section, we describe how the training points, obtained from Wikipedia, are further expanded. Our main motivation behind this expansion is to collect more diversified training points from a text corpus.

For expansion of training points, the first step is to obtain effective word embeddings, where our intuitive assumption is that the words in this abstract space obey the cluster hypothesis—that is, a person’s name should be in close proximity to another person’s name and farther apart from other NE classes. Therefore, in this abstract space for a given point (here, *point* refers to the vector embedding of a word extracted from the Wikipedia category pages belonging to a particular NE category), its nearest points can be easily identified using the *cosine* similarity metric. Following the cluster hypothesis, we assume that for a given Wikipedia training point of a particular NE type, its topmost nearest points can be used to expand the number of training points further. An example of the (top five) expanded points for given Wikipedia training points is provided in Table I.

Using the expanded training points, our word embedding–based NE classification algorithm can learn a more diversified decision boundary. In fact, our experiments with the expanded training points or vectors (more details are provided later in Table VI) reveal that this hypothesis is true. For obtaining word embeddings, we make use of the continuous bag-of-words (cbow) method with negative sampling [Mikolov et al. 2013b]. In our initial experiment with word2vec (reported in Table V), we found that the cbow method with negative sampling is best suited for our purpose.

4. EXPERIMENTAL SETUP

In this section, we describe the experiments to evaluate our proposed methodology. We particularly focus on a resource poor language, namely Bengali. We start with a description of the dataset used for our experiments and follow with a description of the baseline approaches and parameter settings.

Table II. Characteristics of the IJCNLP 2008 Bengali NER Shared Task Dataset

Dataset	NEs	Words	Sentences
Train	5,000	112,845	6,030
Test	1,723	38,708	1,835

Table III. NE Class Distributions for Training Our Classifier

IJCNLP 12 Tags		IJCNLP 4 Tags		Wiki 4 Tags	
Tag Name	Freq.	Tag Name	Freq.	Tag Name	Freq.
Person	1,253	Person	1,253	Person	5,193
Organization	213	Organization	213	Organization	795
Location	594	Location	594	Location	4,229
Title	68	Misc	4,186	Misc	1,453
Designation	800				
Abbreviation	77				
Brand	629				
Object	181				
Time	891				
Number	298				
Measure	386				
Terms	856				

4.1. Setup

We use the Bengali IJCNLP 2008 shared task[7] dataset[8] [Singh 2008] developed for Indian language NER research. Table II outlines the Bengali dataset characteristics.

The dataset comes with POS tag and chunking information. It is divided into training and test sets. Note that since we approach NER as a supervised classification task, we can use the supplied training data to obtain the NE class labels for the corresponding NE word vectors to train our classifier. Additionally, we perform experiments without using the training data, where we show that even without the presence of training data, one may use Wikipedia to automatically construct the training set.

In total the dataset comprises 12 NE tags (see Table III), including *designation*, *abbreviation*, and *brand*, in addition to the four widely used ones, namely *person*, *location*, *organization*, and *miscellaneous*.[9] In our experiments, we focus on the four common ones. This is because one of the objectives of our experiments is to show that our NER method can work even without the absence of training data, for which we use Wikipedia to automatically construct our training set. Since the Bengali Wikipedia does not have category pages for NE classes such as *designation* and *brand*, our Wikipedia-based training set construction can only be performed for four (i.e., *person*, *location*, *organization*, and *miscellaneous*) NE classes. However, for the experiments in which we use the IJCNLP training set, we report results on the complete set of 12 tags.

The first step in our proposed method is to obtain word embeddings. To obtain the word vector representations of NNP tags (the dataset comes with POS tags), we use the word2vec implementation with the cbow model and negative sampling on 200 dimensions [Mikolov et al. 2013b]. In fact, we tried the skip-gram model and varied the parameters value, and we found that the cbow model is best suited for our proposed work. The main reason cbow works well for our method is the availability of a large amount of training corpora [Mikolov et al. 2013a].

[7] http://ltrc.iiit.ac.in/ner-ssea-08/index.cgi?topic=8.

[8] We have used the Singh [2008] maximal NE category.

[9] Used for evaluation in GermEval 2014 (https://sites.google.com/site/germeval2014ner/) and CoNLL 2003 (http://www.cnts.ua.ac.be/conll2003/ner/), among many others.

Table IV. CRF Features

Feature Name	Feature Description
Context words	Previous and next words of a particular word
Prefix & suffix	Character *n*-gram prefix and suffix (*n* ranging from 3 to 5) for all NNPs (proper nouns)
POS & chunk	POS and chunk tags of a word
First & last words	First word and last word of a sentence
Digit	Binary valued feature indicating the presence or absence of a digit in a token
Token id	Provides useful information about the position of the NE tag
Action verb	Information about the nearest action verb for a word

For training the word embedding model and phrase identification, we used the following three Bengali document collections:

(1) A total of 123,048 documents from the FIRE ad hoc IR collection,[10] comprised of news articles published from Bangladesh and India.
(2) The complete Bengali Wikipedia dump[11] of 27,823 articles.
(3) The complete Bengali IJCNLP 2008 shared task dataset consisting of 7,865 sentences.

For conducting NER in a morphologically rich language such as Bengali, one needs to take into account the various inflected forms of an NE while deriving the word embeddings. Concretely speaking, for each NE, there may be multiple vector representations, each corresponding to a particular inflected form, which is clearly not desirable. It is therefore reasonable to apply a stemmer and normalize the inflected forms into a single representation. Consequently, for obtaining the word embeddings, we make use of a freely available rule-based Bengali stemmer [Ganguly et al. 2013] to normalize the words.

4.2. Baselines

To compare our proposed method to state-of-the-art NER results, we select three baselines. The first of these baselines, named BL1, involves applying a CRF with features that has been reported to perform well for the NER task the context words, *n*-gram prefix and suffix, POS and chunk tags, first and last words of a sentence, nearest action verb, etc.) [McCallum and Li 2003]. The list of features that we use for our CRF baseline (BL1) is outlined in Table IV.

Additionally, to compare our method to recent word embedding–based NER approaches [Guo et al. 2014; Passos et al. 2014], we used the NE vector cluster labels as a supplementary feature in the CRF. This constitutes our second baseline (BL2). The word embeddings were estimated from the document collection described in Section 4. The number of clusters was set to 500 as proposed in Guo et al. [2014].

To compare our proposed method to an approach that only uses an automatically constructed gazetteer [Richman and Schone 2008], we employ a gazetteer-based NER, with the gazetteer in this case constructed from the Bengali Wikipedia. This constitutes our third baseline (BL3).

4.3. Tools and Parameter Settings

The CRF implementation that we used for our experiments is an open-source package, namely CRF++.[12] For training the model, we maximized the log-likelihood on the training set. To avoid overfitting, we penalize the likelihood with a spherical Gaussian

[10] http://www.isical.ac.in/∼clia/data.html.
[11] https://dumps.wikimedia.org/.
[12] https://github.com/taku910/crfpp.

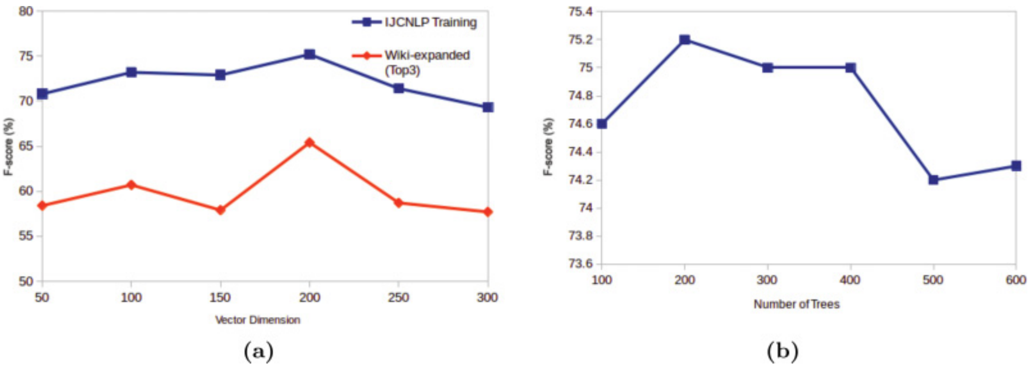

Fig. 2. NER performance sensitivity with respect to vector space dimension for word embeddings (a) and the number of trees in the RF classifier (b) on the IJCNLP dataset.

weight prior [Sha and Pereira 2003]. “L-BFGS” [Sha and Pereira 2003] was used for CRF training.

Since we treat the NER problem as a multiclass classification problem on an abstract vector space of word embeddings, to obtain a reasonable classifier settings, we employed a state-of-the-art classifier, namely the RF [Breiman 2001]. The rationale for using RFs for our classification task is motivated by the reported work in Fernández-Delgado et al. [2014], where 179 classifiers were evaluated on 121 datasets and the results of RF were reported to be the best.

We set the parameters (i.e., the number of trees in RF to 200) with unlimited depth (more details about parameter tuning are available in Section 5.3). The classifier was trained on word embeddings of the four different NE classes (*person*, *organization*, *location*, and *other NNP* or *MISC*), as outlined in Table III.

For obtaining word embeddings, we varied the output vector dimension from 50 to 300 (with a step size of 50) and set the dimension of the vector space of word embeddings to a value of 200 for the best results (see Figure 2(a) of Section 5.3 for more details).

5. RESULTS

In this section, we present the performance of our proposed method relative to the baselines. We first report our results obtained with the help of the official training set of the IJCNLP 2008 Bengali NER data. Next, to show that our method works well even in the absence of any training data, we report experiments using the extracted NEs from Wikipedia.

5.1. Supervised Training

For the experiments described in this section, we used the official IJCNLP Bengali NER training set for estimating the decision boundary between the NE classes. Further, for these experiments, we focus only on the four widely used NE classes—that is, *person*, *location*, *organization*, and *miscellaneous*.

Table V shows that the Wiki-based gazetteer (BL3) yields the worst result, which indicates that only using the category information may not yield satisfactory performance for a low-resource language. The number of Wikipedia pages in Bengali (about 27K) is much less than that of a high resourced one, such as English (more than 4M). This indicates that the number of NEs that can be collected from Bengali Wikipedia is much less compared to the English one. Table V shows that the CRF (BL1) performs much better than the pure or direct mapping of Wikipedia gazetteer in Table VI. However, the results are much lower in comparison to the results generally obtained for English,

Table V. Results (Supervised) on Four NE Categories

Method	Precision (%)	Recall (%)	F-score (%)
BL1: CRF	68.6	60.3	64.2
BL2: CRF with word vector cluster as feature	73.7	68.2	70.8
BL3: CRF with Wiki gazetteer feature	75.8	69.2	72.4
S1.1: Classification of NNP word vectors (skip-gram)	73.9	73.2	72.9
S1.2: Classification of NNP word vectors (cbow)	**76.4**	**75.0**	**75.2**
S2: Classification of NNP word vectors with their context word vectors (cbow)	74.4	74.9	74.6

Note: The word2vec architecture used for this experiment is written within parentheses. Other parameters, such as vector dimension, are set to 200 with a window size of 8.

Table VI. Unsupervised Results on Four NE Categories

Method	Precision (%)	Recall (%)	F-score (%)
Wiki gazetteer (direct mapping)	65.2	21.5	32.3
Classification of word vectors using Wiki training points	61.5	53.7	54.2
Classification of word vectors using expanded (top 5 NN) Wiki training points	**67.6**	56.2	61.4
Classification of word vectors using expanded (top 3 NN) Wiki training points	66.2	**66.0**	**65.4**

Note: The word2vec parameters used for this experiment are cbow with (-ve)sampling and vector dimension set to 200 with window size of 8.

which is around 90% (e.g., see the results in Passos et al. [2014]). One reason for this is the low volume of training data available for low-resource languages. Another reason, which may contribute to the low results, is that in contrast to the capitalization of names in English, there are no obvious cues for recognizing NEs in Bengali. Results improve when the Wikipedia gazetteer is used as a feature in CRF (BL3).

The next observation, which in fact conforms to Passos et al. [2014], Guo et al. [2014], and Demir and Ozgur [2014], is that the cluster information obtained from the word embeddings (BL2) plays an important role in improving the performance of the CRF (an increase in F-score from 64.2% to 70.8%).

Context information plays an important role in improving NE effectiveness by disambiguating among NE types. To consider context information, we classify the NE word vectors along with their context word (context of size 5 around each word) as vectors representation analogous to Collobert et al. [2011] (we call this S2).

The best results are obtained with our proposed method (S1.2) with cbow architecture, which classifies the NE word vectors. This method produces significantly better results in comparison to the baseline approach (i.e., BL2, which also uses word embeddings but in a different way) and a similar approach (S1.1) with different architecture (i.e., skip-gram). Algorithmically, cbow and skip-gram models are similar, except the training objective of the cbow model is to combine the representations of surrounding words to predict the word in the middle, whereas for skip-gram the training objective is to learn word vector representations that are good at predicting its context in the same sentence [Mikolov et al. 2013a]. In the context of NER, which is essentially a sequence labeling task, word ordering is very important. This is precisely the reason we found that the word vectors obtained with the cbow method perform much better than with skip-gram.

In fact, our approach S1.2 (classification using word vector) compared to S2 (classification using word vectors with their context word vectors) results in only small apparent improvements that are not statistically significant (run S1.2:S2; p-value > 0.1 by a two-tailed *t*-test). This is because the contextual information, which is of utmost importance in a sequence labeling task, is encoded implicitly in the word vector embeddings themselves, resulting in only small apparent improvements. In summary, the results indicate the following:

(1) A better way of using the clustering hypothesis of the NE embeddings is to directly use the vectors themselves for obtaining decision boundaries between different NE classes rather than using the vectors as CRF features or as a conjunction of context vectors as additional information.
(2) Sparse training data has less adverse effect on our proposed method than CRF-based approaches.
(3) The classification approach is able to produce better results more efficiently because it takes much less time to estimate the decision boundary with RF than training a CRF.

5.2. Unsupervised Training

For a resource poor language, manually annotated training data may be absent altogether. To see whether our proposed method can work well under such a scenario, we used the list of automatically extracted NEs from Wikipedia for training the classifiers, instead of using the official training set. Unfortunately, without the training set, we cannot use the CRF results as our baselines.

Table VI shows that the Wikipedia-based gazetteer yields the worst result (in comparison to all methods in Tables V and VI), which indicates that only using the category information may not yield satisfactory performance for a low-resource language.

The number of Wikipedia pages in Bengali (about 27K) is much less than that of a high resourced one, such as English (more than 4M). This indicates that the number of NEs that can be collected from Bengali Wikipedia is much less compared to the English one. This, in turn, causes the Wikipedia-based gazetteer method to perform poorly.

Although results obtained with the classification of word vectors using Wikipedia NEs (see Table VI) are worse than that obtained with the IJCNLP training set (see Table V), the results (recall and F-score) are still considerably better than the Wikipedia gazetteer baseline. This shows that our method works effectively even in the absence of any training data. Additionally, this indicates that an automatically constructed list of NEs is the only prerequisite for obtaining satisfactory NER effectiveness for a low resourced language.

A rational explanation of the worse performance of our method (i.e., classification word vectors using Wiki training points or Wiki NEs in Table VI) with respect to Table V is that the test set NEs are in closer proximity to the IJCNLP training set NEs than to the NEs constructed from Wikipedia in the vector space of word embeddings. This explains the lower effectiveness of our method when trained with the Wikipedia NEs than that when trained with the IJCNLP training set.

An interesting observation is that our classification-based method results in a consistent and significant increase in recall for all NE classes compared to the baseline. On the other hand, precision drops consistently across all NE classes. However, the decrease in precision is much lower than the increase of recall. This is most likely caused by the diversity of the Wikipedia-extracted NEs, which may lead to false positives, thus decreasing precision.

Table VI shows that classification of word vectors using expanded Wikipedia training points with three nearest neighbors performs much better (an increase of 11.2% in the F-score) than the classification of word vectors using only the Wikipedia training points. Due to the expanded training set, our word embedding–based NE classification algorithm can learn a more diversified decision boundary, and the classifier's performance generalizes well to the IJCNLP test set. In fact, our proposed best unsupervised approach (i.e., a word embedding–based NER approach using expanded Wikipedia

Table VII. Comparisons with IJCNLP Bengali NER Shared Task Official Submissions

System	System Description	F-score (%)
Saha et al. [2008]	MaxEnt, Rules, Context	**59.54**
Ekbal et al. [2008]	CRF, Language features	55.36
Gali et al. [2008]	CRF, Language heuristics	35.65
Kumar and Kiran [2008]	HMM, Language features	31.48
Our baseline BL1	CRF, Language-independent features	56.14
Our baseline BL3	CRF, Wiki gazetteer	57.23
Our baseline BL2	CRF, Word vector cluster	59.10
Our proposed method	Multiclass classification on NE word embeddings	**70.80**

Note: The word2vec parameters used for this experiment are *cbow* with *(-ve)sampling*. The vector dimension was set to 200 with a window size of 8.

training points in Table VI) performs slightly better than the supervised approach (BL1) in Table V.

5.3. Parameter Sensitivity

In this section, we report NER performance under different parameter selections, such as the vector dimension in word2vec, and the number of trees and depth in RF. The number of layers in the output layer of the recurrent NN architecture used in word2vec corresponds to the dimension of the word vectors. We treat the dimension of the vectors as another parameter. For our experiments, we varied the vector dimension from 50 to 300 with step sizes of 50. Figure 2(a) plots NER performance with respect to different vector dimensions.

For RF, the two parameters that we vary are the number of trees to be generated and the maximum depth of the trees. From our experiment, we found that the NER accuracy is relatively insensitive to the maximum depth of the trees. From Figure 2(b), we observe that the optimal F-score is achieved when the number of trees is set to 200.

5.4. Comparisons with Official Submissions

We now compare our proposed system to the official submissions for the IJCNLP 2008 Bengali NER shared task [Singh 2008]. For this comparison, instead of using four commonly used NE classes, we use the complete set of NE classes used in the shared task. A prerequisite for using the complete set of 12 NE classes is to use the official training set for our NER method instead of using the Wikipedia-extracted NE classes, because Wikipedia contains category pages only for the three NE classes.

Table VII presents a comparative study between our proposed system and other systems participating in the IJCNLP 2008 shared task. For our method, we used RF as the classifier with the number of trees set to 200, which is the best settings reported in Tables V and VI. The result with 12 NE classes is worse than the results for 4 classes, because with an increase in the number of classes, the number of parameters in a classifier (support vectors for SVM or the tree depth in RF) increases as well, which in turn typically tends to overfitting and decreased performance on the test set. We note a similar decrease in NER effectiveness for our CRF baseline methods as well.

Most participants [Ekbal et al. 2008; Gali et al. 2008; Kumar and Kiran 2008] used language-dependent features or heuristics and CRF or HMM as classifiers. The results achieved by our baselines that do not use the word embedding feature are close to the score reported in Ekbal et al. [2008]. Our baseline in this article that uses the word vector cluster labels performs better than the system reported in Ekbal et al. [2008]. However, this baseline is not able to outperform the approach reported in Saha et al. [2008] (the best official submission for this task).

The approach in Saha et al. [2008] involves a hybrid system that applies a maximum entropy model as a classifier in conjunction with language-specific rules, gazetteers, and context patterns. As classifier features, the authors used orthographic

Table VIII. Comparison of NER Performance between the 12- and 4-Class NER Task

	IJCNLP 12 Tags			IJCNLP 4 Tags		
NE Class	Precision (%)	Recall (%)	F-score (%)	Precision (%)	Recall (%)	F-score (%)
Person	72.5	**94.1**	**81.9**	**84.4**	69.8	76.4
Location	68.7	54.1	60.5	**77.7**	**76.0**	**76.9**
Organization	**33.3**	04.5	**08.0**	05.7	**07.1**	06.3
Average	71.9	**83.9**	**77.5**	**78.8**	68.7	73.4

features, information about suffixes and prefixes, morphological features, part of speech information, and information about the surrounding words. They used rules for numbers, measures, and time classes. For *designation*, *person*, and *location*, they used gazetteers. Even without using any language-specific features and NE-specific rules, our proposed method achieves considerably higher NER accuracy than Saha et al. [2008] (i.e., F-score of 70.80% vs. 59.54%).

5.5. Comparisons with Some Existing Systems for Bengali NER

In this section, we compare our system performance to some of the recent existing systems for Bengali NER. Recent works [Ekbal and Saha 2011, 2012; Sikdar et al. 2012] applied a classifier ensemble technique for doing Bengali NER. For example, Ekbal and Saha [2012] used a multiobjective optimization (MOO)-based classifier ensemble technique for Bengali NER and achieved an F-score value of 70.86% (compared to our F-score value of 70.80% on the same dataset) in the IJCNLP 2008 NERSSEAL shared task dataset. The authors also reported an F-score value of 92.46% using a different (closed) dataset for Bengali NER.

In another work, Ekbal and Saha [2011] applied a genetic algorithm (GA) to construct a weighted vote-based classifier ensemble technique for Bengali NER and yielded 92.15% of the F-score value using a closed dataset. However, a GA-based approach tends to consume time, parameter tuning is difficult, and there is no guarantee of finding a global maxima. In a similar work, Sikdar et al. [2012] used a differential evolution (DE)-based two-stage evolutionary approach for Bengali NER and achieved an F-score value of 88.89% using a closed dataset. The DE-based approach has similar difficulties as the GA-based approach. Moreover, the use of a multiple base classifier makes the overall ensemble process a bit slower. Direct comparison of the preceding two systems (i.e., Ekbal and Saha [2011]; Sikdar et al. [2012]) and our system is not possible due to the unavailability of source code (or even finer system details) and the dataset (closed) used in their experiments.

5.6. Further Analysis

5.6.1. Comparisons between the 12- and 4-Class NER Tasks. In this section, we conduct another experiment to compare NER performance with respect to 12 NE classes versus 4 NE classes. Both datasets have 3 NE classes (i.e., *person*, *location*, and *organization*) in common, and the other 9 NE classes in the 12-class problem are treated as a single *miscellaneous* class for the 4-class problem. Results (refer to Table VIII) show that accuracy of the 12-class problem is better than that of the 4-class problem. This is due to the data imbalance problem present in the case of the 4-class problem. From Table III, we see that the most dominant class for the 4-class problem is *miscellaneous*, with 4,186 samples that are distributed over 9 classes for the 12-class problem.

5.6.2. Comparison between Unigram and N-gram NER Tasks. To compare the results between unigram and n-gram NEs, we conducted two separate experiments on the IJCNLP (12 NE class) dataset. Results (refer to Table IX) show that overall, the classification of unigram NEs performs better with respect to the n-gram NEs. Unlike the unigram NER task, the performance of the n-gram NER task additionally depends on

Table IX. Comparison of NER Performance between Unigram and *N*-gram NER Tasks

	Unigram NEs			*N*-gram NEs		
	Precision (%)	Recall (%)	F-score (%)	Precision (%)	Recall (%)	F-score (%)
Average	**80.8**	**79.9**	**78.7**	64.7	64.8	61.8

a phrase identification step. This additional step may result in additional errors for the *n*-gram NER task.

6. CONCLUSIONS

In this article, we have proposed an NER approach that particularly suits a resource poor language—that is, a language for which training data with manually annotated NEs is not available in large quantities. The traditional sequence labeling approaches that work well in the presence of sufficient training data fail to work satisfactorily well in resource-poor languages.

In this work, we have shown that word embeddings that utilize the clustering hypothesis of NE word classes (i.e., the hypothesis that two person's names will be in closer proximity to each other than a *person name* and a *location name*) can be utilized to improve the NER effectiveness of resource-poor languages. Previous research used this cluster hypothesis by including the cluster labels as features into a CRF. In contrast, we propose to directly use the NE word vectors and the corresponding NE classes as labels to train a multiclass classifier. Experiments validate that the classification approach works significantly better than when using word cluster labels as a supplementary CRF feature.

The reason our method works better in comparison to word vector cluster features integrated into CRF is that generating an effective decision boundary requires fewer training resources than a sequence labeling task. Moreover, the contextual information, which is of utmost importance in a sequence labeling task, is encoded implicitly in the word vector embeddings themselves, which makes our method robust as well.

In our experiments, we demonstrate the limitations of traditional approaches, such as CRF, under resource-poor settings. Automatically constructed gazetteers from Bengali Wikipedia pages when used in conjunction with the CRF are able to improve the results. We also show that even with the complete absence of training data, an automatically constructed gazetteer from external resources, such as Wikipedia, can be used for training the classifier and achieve satisfactory NER effectiveness. We also demonstrate that our proposed approach outperforms the best official result in the IJCNLP 2008 Bengali NER task without using any language-specific rules, gazetteers, and complex feature sets.

As part of future work, we plan to investigate the effectiveness of our method on a wide variety of resource-poor languages from South Asia and elsewhere. Another possible future work would be to investigate the performance of our approach in the coordination of a statistical chunker for phrase identification.

APPENDIX

LIST OF AMBIGUOUS CATEGORIES

ব্যক্তিত্ব, সাহিত্য, ধর্মগ্রন্থ, দ্বন্দ্বমূলক, দর্শনীয় স্থান, নবজাগরণ, চলচ্চিত্র, তীর্থস্থান, অবৈধ তারিখ, প্রতিষ্ঠিত, দেবদেবী, দেবী, ধর্মগ্রন্থ, পরিবার, পর্যটনকেন্দ্র, অবৈধ তারিখ, ইতিহাস, আন্দোলন, অনাথ নিবন্ধসমূহ, সুরক্ষা টেমপ্লেটের সাথে উইকিপিডিয়ার পাতা, চিত্র সংযোগসহ পাতাসমূহ, ভাষার বহিঃসংযোগের সাথে নিবন্ধসমূহ, ত্রুটিসহ পাতা, বহিঃসংযোগ সহ সমস্ত নিবন্, যার সম্প্রসারণ প্রয়োজন, ছাড়া ও সংগ্রহের তারিখসহ উদ্ধৃতি ব্যবহার করা পাতা, মাইক্রোবিন্যাসের সাথে নিবন্ধসমূহ, প্যারামিটার ব্যবহার করা উদ্ধৃতিসহ পাতা, পাতাসমূহ অবচিত পরামিতিসহ উদ্ধৃত টেমপ্লেট, প্যারামিটারসহ নিবন্ধ, প্যারামিটারসহ নিবন্ধসমূহ.

REFERENCES

Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. 1999. An algorithm that learns what's in a name. *Machine Learning* 34, 1–3, 211–231. DOI:http://dx.doi.org/10.1023/A:1007558221122

Andrew Eliot Borthwick. 1999. *A Maximum Entropy Approach to Named Entity Recognition*. Ph.D. Dissertation. New York University, New York, NY.

Leo Breiman. 2001. Random forests. *Machine Learning* 45, 1, 5–32. DOI:http://dx.doi.org/10.1023/A:1010933404324

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning (ICML'08)*. ACM, New York, NY, 160–167. DOI:http://dx.doi.org/10.1145/1390156.1390177

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 2493–2537. http://dl.acm.org/citation.cfm?id=1953048.2078186.

Alessandro Cucchiarelli, Danilo Luzi, and Paola Velardi. 1998. Automatic semantic tagging of unknown proper names. In *Proceedings of the 17th International Conference on Computational Linguistics—Volume 1 (COLING'98)*. 286–292. DOI:http://dx.doi.org/10.3115/980451.980892

Hakan Demir and Arzucan Ozgur. 2014. Improving named entity recognition for morphologically rich languages using word embeddings. In *Proceedings of the 13th International Conference on Machine Learning and Applications (ICMLA'14)*. 117–122.

Asif Ekbal, Rejwanul Haque, Amitava Das, Venkateswarlu Poka, and Sivaji Bandyopadhyay. 2008. Language independent named entity recognition in Indian languages. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*. http://aclweb.org/anthology/I08-5006.

Asif Ekbal, Mohammed Hasanuzzaman, and Sivaji Bandyopadhyay. 2009. Voted approach for part of speech tagging in Bengali. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information, and Computation (PACLIC 23)*. 120–129. http://www.aclweb.org/anthology/Y09-1014.

Asif Ekbal and Sriparna Saha. 2011. Weighted vote-based classifier ensemble for named entity recognition: A genetic algorithm-based approach. *ACM Transactions on Asian Language Information Processing* 10, 2, Article No. 9. DOI:http://dx.doi.org/10.1145/1967293.1967296

Asif Ekbal and Sriparna Saha. 2012. Multiobjective optimization for classifier ensemble and feature selection: An application to named entity recognition. *International Journal on Document Analysis and Recognition* 15, 2, 143–166. DOI:http://dx.doi.org/10.1007/s10032-011-0155-7

Richard J. Evans. 2003. A framework for named entity recognition in the open domain. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP'03)*.

Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. 2014. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research* 15, 1, 3133–3181. http://dl.acm.org/citation.cfm?id=2627435.2697065.

Karthik Gali, Harshit Surana, Ashwini Vaidya, Praneeth Shishtla, and Dipti Misra Sharma. 2008. Aggregating machine learning and rule based heuristics for named entity recognition. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*. http://aclweb.org/anthology/I08-5005.

Debasis Ganguly, Johannes Leveling, and Gareth J. F. Jones. 2013. DCU@Morpheme extraction task of FIRE-2012: Rule-based stemmers for Bengali and Hindi. In *Proceedings of the 5th Forum on Information Retrieval Evaluation (FIRE'13)*. 12.

Yoav Goldberg and Omer Levy. 2014. Word2vec explained: Deriving Mikolov et al.'s negative-sampling word-embedding method. arXiv:1402.3722. http://arxiv.org/abs/1402.3722.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Revisiting embedding features for simple semi-supervised learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 110–120. http://www.aclweb.org/anthology/D14-1012.

Jun'ichi Kazama and Kentaro Torisawa. 2007. Exploiting Wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'07)*. http://aclweb.org/anthology/D07-1073.

P. Praveen Kumar and V. Ravi Kiran. 2008. Hybrid named entity recognition system for South and South East Asian languages. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*. http://aclweb.org/anthology/I08-5012.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and Web-enhanced lexicons. In *Proceedings of the 7th Conference on*

Natural Language Learning at HLT-NAACL 2003, Volume 4 (CoNLL'03). 188–191. DOI:http://dx.doi.org/10.3115/1119176.1119206

Andrei Mikheev, Marc Moens, and Claire Grover. 1999. Named entity recognition without gazetteers. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL'99)*. 1–8. DOI:http://dx.doi.org/10.3115/977035.977037

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. arXiv:1301.3781. http://arxiv.org/abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Neural Information Processing Systems Conference (NIPS'13)*. 3111–3119.

Andriy Mnih and Geoffrey E. Hinton. 2008. A scalable hierarchical distributed language model. In *Proceedings of the Neural Information Processing Systems Conference (NIPS'08)*. 1081–1088. http://papers.nips.cc/paper/3583-a-scalable-hierarchical-distributed-language-model.

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. In *Proceedings of the 18th Conference on Computational Natural Language Learning*. 78–86. http://www.aclweb.org/anthology/W/W14/W14-1609.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL'09)*. 147–155. http://dl.acm.org/citation.cfm?id=1596374.1596399.

E. Alexander Richman and Patrick Schone. 2008. Mining Wiki resources for multilingual named entity recognition. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the Human Language Technology Conference (ACL-08: HLT)*. 1–9. http://aclweb.org/anthology/P08-1001.

Sujan K. Saha, Sanjay Chatterji, Sandipan Dandapat, Sudeshna Sarkar, and Pabitra Mitra. 2008. A hybrid named entity recognition system for South and South East Asian languages. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*. http://aclweb.org/anthology/I08-5004.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology—Volume 1 (NAACL'03)*. 134–141. DOI:http://dx.doi.org/10.3115/1073445.1073473

Utpal Kumar Sikdar, Asif Ekbal, and Sriparna Saha. 2012. Differential evolution based feature selection and classifier ensemble for named entity recognition. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING'12)*. 2475–2490. http://dblp.uni-trier.de/db/conf/coling/coling2012.html#SikdarES12.

Anil K. Singh. 2008. Named entity recognition for South and South East Asian languages: Taking stock. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*. http://aclweb.org/anthology/I08-5003.

Antonio Toral and Rafael Muñoz. 2006. *A Proposal to Automatically Build and Maintain Gazetteers for Named Entity Recognition by Using Wikipedia*. Technical Report. Available at http://www.aclweb.org/anthology/W06-2809.pdf.

Joseph Turian, Yoshua Bengi, Lev Ratinov, and Dan Roth. 2009. A preliminary evaluation of word representations for named-entity recognition. In *Proceedings of the NIPS Workshop on Grammar Induction, Representation of Language, and Language Learning*. http://citeseerx.ist.psu.edu/citeseerx/viewdoc/summary?doi=10.1.1.174.1362.

L. J. P. van der Maaten and G. E. Hinton. 2008. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research* 9, 2579–2605.

Ziqi Zhang and José Iria. 2009. A novel approach to automatic gazetteer generation using Wikipedia. In *Proceedings of the 2009 Workshop on the People's Web Meets NLP: Collaboratively Constructed Semantic Resources (People's Web'09)*. 1–9. http://dl.acm.org/citation.cfm?id=1699765.1699766.

GuoDong Zhou and Jian Su. 2002. Named entity recognition using an HMM-based chunk tagger. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*. 473–480. DOI:http://dx.doi.org/10.3115/1073083.1073163

Received September 2015; revised October 2016; accepted November 2016