Towards Generalizable Place Name Recognition Systems: Analysis and Enhancement of NER Systems on English News from India

Arda Akdemir
Koç University
Istanbul, Sarıyer
aakdemir@ku.edu.tr

Ali Hürriyetoğlu
Koç University
Istanbul, Sarıyer
ahurriyetoglu@ku.edu.tr

Erdem Yörük
Koç University
Istanbul, Sarıyer
eryoruk@ku.edu.tr

Burak Gürel
Koç University
Istanbul, Sarıyer
bgurel@ku.edu.tr

Çağrı Yoltar
Koç University
Istanbul, Sarıyer
cyoltar@ku.edu.tr

Deniz Yüret
Koç University
Istanbul, Sarıyer
dyuret@ku.edu.tr

ABSTRACT

Place name recognition is one of the key tasks in Information Extraction. In this paper, we tackle this task in English News from India. We first analyze the results obtained by using available tools and corpora and then train our own models to obtain better results. Most of the previous work done on entity recognition for English makes use of similar corpora for both training and testing. Yet we observe that the performance drops significantly when we test the models on different datasets. For this reason, we have trained various models using combinations of several corpora. Our results show that training models using combinations of several corpora improves the relative performance of these models but still more research on this area is necessary to obtain place name recognizers that generalize to any given dataset.

KEYWORDS

Place Name Recognition, Entity Extraction, Named Entity Recognition, Natural Language Processing, Machine Learning

ACM Reference Format:
Arda Akdemir, Ali Hürriyetoğlu, Erdem Yörük, Burak Gürel, Çağrı Yoltar, and Deniz Yüret. 2018. Towards Generalizable Place Name Recognition Systems: Analysis and Enhancement of NER Systems on English News from India. In *12th Workshop on Geographic Information Retrieval (GIR'18), November 6, 2018, Seattle, WA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3281354.3281363

1 INTRODUCTION

Place name recognition which can be considered as a subtask of more general Named Entity Recognition (NER) is one of the most important Information Extraction tasks. Like many other NLP tasks work done for English is more extensive compared to other languages for NER and some researchers consider the NER task in English as a solved problem. Also the emergence of recent machine learning methods to the field increased the performance of NLP systems significantly. Yet we observed that the state-of-the-art systems are sensitive to changes in the dataset and source location especially in the case of place name recognition.

When the language of the corpus is the same with a trained model but the source of the corpus is a different country (or genre) a common way to do NER is to use comprehensive gazetteers. Relying on gazetteer look-up systems provides convenience since they are relatively easy to develop and straightforward but their performance is limited so is their capability to generalize.

On the contrary, recent research on machine learning methods focuses on developing systems that can generalize better and are more portable. Similarly, we have tried various machine learning paradigms and developed our own machine learning based models to extract place names.

In this paper we analyze and enhance various state-of-the-art machine learning tools for place name recognition in English on Indian News and show that performance drops significantly when we test the tools in different datasets even though the language is the same. We tackled several challenges at once:

- The corpora we have used for training and our test set for Indian News is from different sources. This difference affects the performance significantly. Related to this issue, another problem is the language dynamics of English used in India. To overcome this we tried to incorporate specific external knowledge into the training corpora.
- The annotation guidelines of training and test sets are different which causes some entities to have different tags in training and test sets. This puts a certain upper limit to the success of any system and can not be handled without making changes in the datasets.

Our main contributions can be grouped into three categories: a) Creating various tools which will speed up research in this area, b) creating a gold standard corpus for place name recognition in Indian News written in English which will be publicly available

GIR'18, November 6, 2018, Seattle, WA, USA

ACM ISBN 978-1-4503-6034-0/18/11…$15.00
https://doi.org/10.1145/3281354.3281363

after publication[1], c) analysis of available tools on Indian News test set, and extending the available tools in various ways (learning gazetteers, merging different corpora, using specific word embeddings) to obtain relative improvements on all test sets. The paper is structured as follows:

We start with the related work done in NER and place name recognition. This will be followed by the description of the datasets and the tools we have used. Finally we will explain in detail the experiments we have done, the results we have obtained and the future work we are planning as a continuation of this study.

2 RELATED WORK

Jones et al. [13] give a detailed overview of Geographical Information Retrieval. They mention seven issues in GIR, first issue titled 'Detecting geographic references' is the most relevant one with the work explained in this paper. They explain the difficulty of detecting geographical entities and correctly disambiguating them from organization and people's names. This issue is tackled widely by Information Retrieval and Natural Language Processing researchers and forms the basis of GIR systems.

Purves et al. [20] state the progress made and future challenges in Geographic Information Retrieval. At the beginning of the paper they give a detailed list of exemplar GIR systems which can be considered as a brief summary of the work done until now on GIR.

Karimzadeh et al. [14] provide a tool named GeoTxt which tackles disambiguation and geolocation of place names as well as extraction of them. For extraction they make use of available Named Entity Recognition tools such as Stanford NER [12]. Then a Geocoder module is applied to the extracted place names for disambiguation and geolocation. Edinburgh Geoparser [2] makes use of a rule-based system to extract place names in a given text and then disambiguates them by making use of gazetteers. The named entity recognizer of the Parser is more interpretable compared to systems that rely on machine learning methods. Yet it has the disadvantages of systems that rely on rules. They make use of third party software for the low level analysis of a given text such as lemmatization and POS tagging. Then a rule based system performs chunking and NER by making use of the low level analysis results. We have used this parser as a baseline tool to compare our results against. The results are given in the Results section. Delozier et al. [8] provide a Gazetteer independent Toponym Resolution (TR) system that models the geographic distribution of words. Their system also make use of the Stanford NER tool for extracting the place names and then apply their TR module.

Generalization of NLP systems is an emerging area of research among NLP scholars. Ettinger et al. [11] tackle this issue at their Workshop and Shared Task. Another Workshop on this issue is organized this year and is titled 'Workshop on New Forms of Generalization in Deep Learning and Natural Language Processing'.[2] As the name suggests the primary focus of the workshop is the generalizability of NLP and Deep Learning systems. Generalization of NER systems in particular is another emerging research area. Augenstein et al. [4] show that state-of-the-art NER systems struggle to generalize over genres with limited training data. Unseen NE's together with unseen features of those NE's limit the performance of such systems on diverse genres. They report that small changes in the percentage of unseen features significantly affect the F1 score. Thus memorization of both surface forms of NE's and their context is an issue for state-of-the-art systems that make use of deep learning methods.

Named Entity Recognition is a popular task especially for English and there are various studies in the recent decades. Nadeau et al. [17] gives a comprehensive survey of the work done on NER until 2007 which includes various conventional Machine Learning methods such as HMM, CRF etc. so we will not be restating them here again. Yet in the last decade Neural Networks and especially Deep Learning methods outperformed the conventional Machine Learning tools in NER task as well as in many other tasks. Chiu et al. [5] uses a hybrid bi-LSTM and CNN architecture. Their model which uses two lexicons has an F1 score of 91.62 on the Conll 2003 English dataset. For the first lexicon for each of the NE types they compiled a list of known NE's from DBpedia [3]. The second lexicon they used is the lexicon released by Collobert et al. [6]. Lample et al. [15] uses Bi-directional LSTM's and CRF using character and word embeddings. They report state-of-the-art results without using external lexicons for the 4 languages in the Conll 2002-2003 datasets (English, Dutch, Spanish, German). Their model have 90.94 F1 score without using gazetteers.

Previous work on Named Entity Recognition for Indian corpora in English includes the work done in the FIRE 2013[3]-2014[4] workshops. Prabhakar et al. [19] used a CRF based model with hand crafted features. Their work also focuses on learning sub categories of entries such as Government for ORG type entities but the scores are quite low on these early results. They make use of the Stanford NER tool during all the stages. They report 38.73 and 51.17 for precision and recall respectively. Abinaya et al. [1] also uses CRF for the same FIRE-2014 task. They use SVM's for other Indian Languages but report that CRF performs better in English compared to other Machine Learning methods. They make use of lexicons and gazetteers as binary features for CRF. Sub-token level information is also used as features (trigrams). They also use an extensive list of hand crafted features related to the special characters and digits. The main drawback of their approach is that their model rely on manual feature extraction. The F1 score for the outer layer which corresponds to the conventional NER is 59.37. Finally, Sanjay et al. [22] applied a CRF based model on Twitter posts in India. They also use both linguistic features such as POS tags and binary features to capture important patterns. They report the F1 scores for unigram and bigram based models as 32.50 and 33.15 respectively. Thus the work done until now on the Indian corpora in English is quite limited and they make use of similar methods.

3 DATA

This section explains the datasets we have used for training and testing our systems.

[1] this corpus will be described in detail in Data section
[2] https://newgeneralization.github.io/

[3] http://au-kbc.org/nlp/NER-FIRE2013/index.html
[4] https://www.isical.ac.in/ fire/2014/index.html

3.1 CoNLL-2003 English

One of the datasets we have used is the CoNLL-2003 English dataset[5] for CoNLL-2003 Shared Task. This corpus is taken from Reuters news stories between August 1996 and August 1997. More details about the corpus and the shared task is given by Tjong et al.[23]. For convenience we will be referring to this dataset as Conll corpus for the rest of this paper. During certain experiments we have merged all the Conll corpus to train models. We will be referring to this combined set as Call. Conll corpus is one of the most frequently used publicly available annotated datasets in this domain. The format of the corpus available online is in token-per-line format where each line contain a single token together with features like POS-tag and the label of the token at the end. Documents and sentences are separated with blank lines. There are many published results for this dataset and the state-of-the-art results are satisfactory(F1-91.62). The dataset contains 4 main entity types: PER, LOC, ORG, MISC. For the purpose of our project we used LOC and ORG type entities. Many of the MISC type entities also contain hints about the location information as Nationalities such as American, Turkish, English etc. In this paper we do not consider those entities. Table 1 gives the number of entities included in each subset of Conll corpus. For the rest of the paper we will be referring to Conll training, validation and test sets as Ctrain, Cvalid and Ctest respectively.

		Training	Validation	Test
Conll	LOC	8,297	2,094	1,925
	PER	11,129	3,149	2,773
	MISC	4,593	1,268	918
	ORG	10,025	2,092	2,496
ACE	LOC	4,176	942	684
	ORG	2,470	551	293
Indian News	LOC	-	-	593
	ORG	-	-	637
	PER	-	-	349

Table 1: Number of Entity Tokens in Conll2003 English Corpus

3.2 ACE 2005 English

Second dataset we have is the ACE 2005 corpus for automatic content extraction created by Linguistic Data Consortium[6]. The corpus is made up of news articles and news recordings from U.S. and U.K. and is annotated for entities, events, relations and their mentions. We only used the annotated entities in this corpus. The corpus also comes with different versions of annotation (annotated by a single person, discrepancy resolution done, etc.). In order to have a high quality corpus we used the version subject to dual annotation and discrepancy resolution. This version of the corpus is made up of 535 documents from weblogs, broadcast news, newsgroups and broadcast conversations and contains 216,545 words. The ACE corpus is in XML-format and annotated in a detailed way. In order to work conveniently on this corpus we created our own ACE specific tokenizer which converts the ACE corpus into the Conll format for the designated entity types are annotated. The tokenizer is publicly available and can be found in the GitHub repository [7] related to this paper and relthe ERC-funded project. Details about the tokenizer will be given in the Tools section.

Table 1 shows the number of entities included in each set of ACE corpus. We have divided the corpus into three parts with the following ratios: 70%, 20% and 10% for training, validation and test sets respectively. We used combinations of these in some of our experiments. In a similar fashion we will be referring to ACE training, validation and test sets as Atrain, Avalid and Atest respectively.

3.3 English News from India

The annotated corpus we have used is annotated by our annotation team of our project. The corpus is gold standard and each document is annotated by two annotators. Also the disagreements between annotators are resolved by a third person. It is created for the Event Extraction project of which the work described in this paper is only a small part. This first batch we use in this paper contains 116 annotated documents from Indian News. The data is tokenized using the UCTO tool[8] and annotated in the FOLIA format[9]. The number of entity tokens in this set is shown in Table 1. This data is mainly used to evaluate the performances of our models. An annotated example sentence in token-per-line format can be seen in Figure 1 for only illustrative purposes of the dataset format.

Several important characteristics of this corpus are as follows:

- Only the place names that are inside the event sentence is annotated. If no place name is mentioned inside an event sentence, then the mention of a place name that is closest to the event sentence is annotated. An event sentence is the sentence in a document (News article) that mentions the main event of the document.
- Almost none of the entities are capitalized in the original article. This is a problem related to the website we have taken the news from. Apparently this capitalization issue is resolved for the news articles published later. Yet we do not have labeled version of those articles at the time of submitting this article. Thus we only capitalized annotated entities manually for this set.
- The tags include Facility (FAC) type entities which does not exist in Conll corpus. Since we do not have a FAC type in Conll we have mapped these entities to Organization (ORG) type.
- We allowed overlapping annotation for a single token (One token may be shared by multiple entities). When converting to Conll tags, we assign tokens which are tagged multiple times with the tag of the longest parent phrase.
- Our dataset is taken from a specific time period and related to a specific topic (news related to contentious political events).

There are also some challenges unique to NER for Indian entity names:

- Some named entities are not capitalized which is the main feature of English named entities.

[5]https://www.clips.uantwerpen.be/conll2003/ner/
[6]https://catalog.ldc.upenn.edu/LDC2006T06
[7]https://github.com/emerging-welfare/Location-Recognition-Tools
[8]https://languagemachines.github.io/ucto/
[9]https://proycon.github.io/folia/

```
minor etype
tensions etype
prevailed O
in O
dharwad place
on O
friday O
evening O
, O
when O
some O
miscreants O
pelted etype
stones etype
at O
a loc
jewellery loc
shop loc
in O
gandhi place
chowk place
. O
```

Figure 1: Example annotated sentence from Indian News Corpus where 'etype' tag represents events

- Some words such as "Roja" both refer to common nouns and proper nouns.
- There is a spelling variation of entities. Certain named entities are spelled differently in Latin letters by different writers.

A detailed analysis of the nature and the characteristics of the Indian English is given in the work of Pingali et al. [18]. This work is valuable as it highlights the challenges researchers working on Indian English face.

Updated Indian News Set

Due to the above mentioned characteristics of the Indian News Test Set we have also used an updated version of this set. The updated version only contains the event sentences. Since other sentences contain unannotated entities including non-event sentences cause the performance of all models to drop. This drop seen among all models is not of primary concern as we focus on obtaining relative improvements. Results obtained using this updated version are also given in the Results section.

3.4 Difference between Annotation Guidelines

An important issue limiting the performances of our trained models is related to the annotation guidelines. As explained above we have used two different corpora during training and a third one which we used only for testing. All of these corpora are annotated using different guidelines. Thus the agreement between the annotation guidelines is an important factor. For the scope of this paper we focus on the agreement guidelines related to place names.

Conll and ACE corpora have similar definitions for ORG type entities. The LOC and GPE types in ACE correspond to LOC type in Conll. For this reason we mapped all the GPE type entities in the ACE dataset to LOC type when using both corpora.

Main issue is related to the Facility type we have in our Indian News annotation guideline. The definition in the annotation guideline states that any man made physical entity that an event takes place is a Facility. Some of the Facility type entities (FAC) in our annotation guideline correspond to ORG type in others (X University, Y Hospital etc.). On the other hand, some FAC's correspond to LOC type in other guidelines (District Names, Stadium Names etc.). Two possible methods to solve this issue is:

(1) Manually creating another annotated corpus by only changing the FAC type entities to LOC and ORG accordingly. This method requires a lot of manual labor and the benefits are limited because we are planning to use our own guideline during Event Extraction.
(2) Mapping FAC type entities to LOC or ORG type for place name recognition. This method requires no manual labor and is not a really bad simplification of the above mentioned laborious problem. Our aim is to detect place names and we would like to be confident that the LOC type entities in our test set refer to place names. By mapping FAC type to ORG we may lose some place names but we make sure that no noise is added to the entities with LOC tag.

The ACE dataset also contains a NE type called Facility. Since the Indian News annotation guideline uses as reference the ACE guideline the Facility type entities in both corpora overlap. Yet during the training we merged the Conll and ACE corpora so we did not take into account Facility type entities of the ACE dataset to make it compatible with Conll corpus which contains PER, LOC, ORG, MISC types. The MISC type have some overlaps with Facility type entities in ACE but it would not be wise to map all the Facility type entities of ACE to MISC.

Above mentioned phenomena must be taken into account when observing the performance results for our trained models. The noise in our datasets and the annotation agreements (disagreements) is significantly effecting the performance.

3.5 Test sets

During training we combined the available corpora to see if the performance improves. In order to assess the performance of each model fairly we used the same three test sets throughout this paper: Conll test set (Ctest), ACE test set (Atest) and Indian News Test set (Itest). Conll and ACE test sets are important to show that the results of trained models drop when we test them on a test set coming from a different dataset. For example, after training a model using the Ctrain we show its performance on both Ctest and Atest. Yet we are ultimately interested in how each model perform on our Indian News test set. Thus training section explains the results we obtain on Atest and Ctest to show relative performances of each system but a separate results section is devoted to the results we obtained for Itest.

4 TOOLS

We have developed several tools such as a converter for the ACE dataset and a word embedding trainer using unannotated Indian news articles. The ACE 2005 dataset is in XML format and the corpus is annotated in a detailed way. Thus it takes some time and effort to extract the relevant entities for a project. For this reason we have created a converter tool which extract the proper nouns with specified labels. This tool enables both fast extraction of the named entities from the ACE documents and also puts the corpus into token-per-line format. This format with empty spaces between each sentence (each sequence) is the most common data format for sequence classification tasks such as POS tagging and NER.

We believe that researchers can benefit from this tool during their research in NER. In the default mode the tool extracts the LOC, GPE, ORG and PER type entities but this can be changed by giving the corresponding parameters.

The word embedding trainer tool is a straightforward program that calculates and saves the word vectors for a given corpus using Gensim word2vec library [21]. The Python library itself is fairly easy to use but still our tool is useful as it is an end-to-end tool providing the final output (word vectors).

We have extended a lookup table tool according to our needs and developed a new tool. We have extended the tool named **MER** (Minimal Named-Entity Recognizer) [7] which given a text and a lexicon outputs the index of the entities. We have extended this tool so that it takes a token-per-line corpus and a gazetteer and outputs a corpus with entities in the gazetteer tagged. The entities are marked as LOC and others as O so that the output of the tool can directly be used to calculate performance scores. This extension to the MER tool is another contribution which will speed up the future research that will make use of similar tools. It is named 'lookup' and is also available on our GitHub repository.

The available tools we have used for training our models are **Wapiti**[10] and **NeuroNER**.[11] Both are publicly available and can be downloaded online. Wapiti is a sequence classifier using algorithms such as Maximum Entropy, Maximum Entropy Markov and Conditional Random Fields models.

NeuroNER is a tool designed by Dernoncourt et al. [9] specifically for Named Entity Recognition task. It comes together with a model already trained on Conll corpus with F1 score of 90.66% which is comparable to the state-of-the-art result of 90.94 without using gazetteers. NeuroNER uses Bi-LSTM and CRF which are proven to be quite successful methods in NER task in various genres and languages. Following subsections explain each tool in detail separately.

4.1 Wapiti Toolkit

This section explains the Wapiti toolkit by Lavergne et al.[16]. The section is divided into segments devoted to a specific aspect about the tool.

4.1.1 Data format. Wapiti can use only a specific data format. The data for training and testing must be in the same token-per-line format with same number of features. The last token of each line must be the label of the token. The sequence that corresponds to sentences must be separated by a blank line. If not the program considers the document as a single sequence and learns transition probabilities from previous sentences meanwhile decreasing the training speed.

4.1.2 Modes. Wapiti has 2 main modes: training and labeling. During training the user can choose among different Machine Learning methods (Maxent, Memm and Crf) and different learning algorithms (SGD, quasi-newton optimization etc). This makes the wapiti toolkit flexible. More details about the configurations of the tool will be given in the configuration section. The training mode requires a pattern file which contains the information about how to generate feature functions in CRF which will be used in transition likelihood calculations. Given a pattern file and a training corpus the tool creates a model which contains the transition probabilities. This model is used for prediction.

Labeling mode is the prediction mode of the wapiti program and requires the model file and test file as inputs and outputs the prediction file containing the label prediction at the end of each line for each token. The tool has the option to calculate and output the recall, precision and F1-score at the end of the prediction. The scores will not be correct if the tagging format is different from the format of the corpus(BIO, BIOES etc.).

4.1.3 Configurations. Wapiti toolkit allows full flexibility in configuration. The user can change every hyper parameter of the model by a simple command. We took advantage of this flexibility and trained and tested the model multiple times by changing the hyper-parameter to find the optimal configurations. Hyper parameters include the backpropagation algorithm types, L1 and L2 complexity penalties etc. Details about the experiments will be given in the results section. For now it suffices to say that changes in the hyper-parameters does not change the results significantly, whereas in ANN-based models a small change in the learning rate determines whether the model works at all or not.

4.1.4 Patterns. The patterns given as input determine the feature functions that will be generated. Feature functions are binary-valued functions such as: 1 if previous word is 'in' else 0. They can combined to produce more complicated patterns. The patterns can be the features given in the corpus as shown in the example above or Regex patterns. Wapiti allows simple Regex patterns which enables adding features like capitalization without making changes in the corpus. As generating features like POS tags are costly and must be done for each corpus separately we have only used patterns such as the words themselves and character level features about them. The window size we have used is +-2.

Patterns used in our experiments:

- Context words: The stem form of words around the current word.
- Capitalization features: first-letter capital, all capital and mixed capitalization.
- Digit features: Contains digit, all digit .
- Punctuation: Contains punctuation, contains punctuation inside, all punctuation.
- Suffix: Current word containing all possible 1,2,3 and 4 character-long suffixes.
- Prefix: Current word containing all possible 1,2,3 and 4 character-long prefixes.

Suffix and Prefix patterns are used only for the current word since increasing the window size for all possible suffixes increases the amount of feature functions exponentially. Since our primary aim in this work is to try various different methods and learn the baseline results, the feature engineering and feature tuning is considered as future work. That is why we tried to keep our patterns and features as general as possible.

[10] https://wapiti.limsi.fr/ . We have used the v1.5.0 release of the toolkit from 18.12.2013 which is the latest version by the time of writing this paper.
[11] https://github.com/Franck-Dernoncourt/NeuroNER . We have used the version available on GitHub after the final commit which was done in August 1, 2017.

Hyperparameter	Default Value
using character lstm	True
char embedding dimension	25
char lstm dimension	50
token emb pretrained file	glove.txt
token embedding dimension	100
token lstm dimension	100
using crf	True
random initial transitions	True
dropout	0.5
optimizer	sgd
learning rate	0.005
gradient clipping value	5
patience	10
maximum number of epochs	100
maximum training time	10
number of cpu threads	8

Table 2: Hyperparameters of the NeuroNER tool

4.2 NeuroNER

NeuroNER [9] is a Named Entity Recognition tool that uses Bi-LSTM and CRF with word and character embeddings. The main advantage of this tool over others is that it makes use of no features besides the raw text input. Thus the tool can be tested on any raw text. No data formatting or feature embedding is necessary for the pretrained systems to be applied on a given dataset. The tool has state-of-the-art F1 score of 90.5[9] for the frequently used CTest.

NeuroNER is publicly available and comes together with pre-trained NER models. The model that is most relevant to our task is trained on the Conll corpus. In the original version the system uses the GloVe for token embeddings. As the training takes a long time, the tool has no option for learning embeddings from scratch, yet allows using other pretrained embeddings. The details regarding the architecture of the tool can be read in the work of Dernoncourt et al. [10].

NeuroNER has two main modes: Training and Testing. Testing is also done at the end of each epoch during training. There are various hyperparameters of the system as in the case with almost all ANN based tools. The list of the hyper parameters and their default values are given in Table 2. This table can also be considered as a nice summary of the inner details of the NeuroNER tool. We made different experiments by changing values of some of the relatively less sensitive hyperparameters but mainly followed the default values. ANN based tools are highly sensitive to changes in the hyperparameters and generally fine tuned default values give the best results.

5 TRAINING

We have trained our models on the Conll and ACE corpora and tested them on the Itest. Even though the genre of the training and testing corpora is similar (News genre), the overlapping entities are scarce because the test set is from Indian News. Also it is important to note here that we only consider LOC type entities when we refer to performances of all models since these are the most relevant entities to the place name recognition task.

We have conducted various experiments and during these experiments used different combinations of the datasets we have for training and testing. To avoid confusion we have only combined the parts of the datasets described in the data section (training, test and validation sets). For example for some experiments we combined the training set of ACE with all the sets(training,validation and test) of Conll for training our NeuroNER models and did the testing on the Atest. We will be explaining the datasets used in each experiment explicitly. As explained in the test sets section, this section only gives the results on Atest and Ctest. Results section is devoted to the results on the Itest. All of the results explained in this section is given in Table 3.

We started with reimplementing the previous models on the same datasets. First we achieve the previously obtained results for our newly trained models on the Conll and ACE corpora. Then we changed the configurations and combined the datasets to have various different new models.

5.1 Wapiti Models

First, we trained several models on Ctrain using Wapiti. First model is trained using the recommended configuration in the website of Wapiti. In the recommended settings Wapiti uses CRF based model with L-BFGS as the optimization algorithm. Then we trained a second model without using L-1 regularization which makes the trained model more complicated and takes relatively more time to train. First two models used the Conll corpus without taking into account the sentence boundaries . Finally we used a sentence-splitted version of the Conll corpus. We tested these models on Cvalid, Ctest and Atest. Results are given in Table 3. These results also show that taking into account the information from previous and following sentences by discarding the sentence boundaries decreases the performance. So we did not discard the sentence boundaries in the subsequent experiments.

Next, we trained models using Atrain and tested them on Atest. This dataset is relatively small and we consider these models as baseline for ACE. We trained two different models with 5 and 1 for L1-regularization penalty. Results are shown in the second part of Table 3.

Then we merged the two datasets together and trained models on this larger corpus. During merging we used all the Conll corpus (including the validation and test sets) and training set of the ACE corpus. For testing we used the Atest and Ctest. We again trained many models with slightly different configurations. Testing these models on Atest shows that all models have similar performance and no configuration have significant superiority over others. Important results of these models are also shown in Table 3. Results show that merging the corpora increased the performance on the same test set.

5.2 NeuroNER Models

We started with reimplementing the reported result which is comparable to the state-of-the-art for Conll corpus using NeuroNER. We have trained a NER model using the exact same configurations with

Training Set	CTrain								
Test Set	Ctest			Cvalid			Atest		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Wapiti	0.802	0.813	0.808	0.846	0.872	0.859	-		
+noL1reg	0.872	0.853	0.862	0.914	0.916	0.915	-		
+sentbound	0.852	0.862	0.857	0.914	0.920	0.917	0.477	0.751	0.583
NeuroNER (Neuro1)	0.925	0.928	0.927	-			0.761	0.602	0.672
Training Set	Atrain						ATrain+Call (Merged corpus)		
Test Set	Ctest			Atest			Atest		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Wapiti +sentbound	-			0.665	0.838	0.741	0.841	0.808	0.824
+L1reg1 (Wapiti 0,1)	0.467	0.674	0.552	0.714	0.910	0.800	0.835	0.818	0.826
NeuroNER (Neuro 2,3)	0.618	0.851	0.716	0.832	0.914	0.872	0.857	0.922	0.888

Table 3: Results obtained for experiments done in Section 5. Itest is not used at this point for testing here. +sentbound takes into account the sentence boundaries and +noL1reg is the Wapiti model that does not apply L1 regularization, L1reg1 means L1 regularization penalty is 1. Default penalty is 5. NeuroNER models trained using Ctrain, Atrain and merged corpus are given names Neuro 1, 2, 3 respectively. Likewise Wapiti models trained using Atrain and merged corpus are called Wapiti 0, 1 respectively. '-' sign indicate that the experiment with the specific tool/training set/test set combination is not done.

the pretrained model. We achieved the same performance scores for the Ctest (Table 4). This model is our baseline NeuroNER model. For the following trained models we always used the Ctest and Atest to measure the performance. First we tested the state-of-the-art model trained on Ctrain (Neuro1) on Atest. The F1 score of this model for LOC type entities drops from 0.927 to 0.672 when we change the test set from Ctest to Atest. This shows that the results drop significantly when we use a test set created from a different corpus even though the genre of both corpora are similar.

	Precision	**Recall**	**F1 Score**
LOC	0.925	0.928	0.927
MISC	0.811	0.802	0.807
ORG	0.870	0.893	0.881
PER	0.964	0.948	0.956

Table 4: Results for the model trained on Ctrain using NeuroNER and tested on Ctest for each entity type (Neuro1)

Then we trained a new model using Atrain only (Neuro2). The results show that doing the training and testing on the same dataset (ACE corpus) gives better results. Neuro2 has an F1 score of 0.872 whereas Neuro1 trained on Ctrain has an F1 score of 0.672. Finally we trained new models on the merged corpus (Call and Atrain) and tested on the Atest. Best results are achieved using this model trained on the merged corpus (Neuro3). This model has an F1 score of 0.888. This results is better than the models trained using Wapiti as well.

We finish this section by restating the change in performance when we train a tool on one corpus and test it on the other for Conll and ACE. The significant drop in F1 scores shows us that the results obtained by only using one of these corpora for both training and testing is often misleading. This shows the importance of creating models which can perform similarly on different datasets. Table 5 and Table 5 gives these results for models trained using Wapiti and NeuroNER respectively. In each table first row gives the F1 scores for LOC type entities when we train a model using CTrain. For both tools the score drops around 0.3 when we change the test set.

	Train/Test	CTest	ATest
Wapiti	CTrain	**0.857**	0.583
	ATrain	0.552	**0.800**
NeuroNER	CTrain	**0.927**	0.672
	ATrain	0.716	**0.872**

Table 5: F1 Scores for cross testing using Wapiti and NeuroNER

Until this point, we tested our models on the Atest and Ctest to have an idea about the relative performances of the various tools with different configurations and training sets. Next section will show the results of testing these models on English News from India set. We compare the results obtained using each trained model and using several baseline tools.

6 RESULTS

In this section we give the results we obtained for our models on Itest. It is described in test sets section and in Table 1. We start with baseline models as simple as lookup tables and continue with the Edinburgh Parser and Stanford NER. Then we give the results for

our various trained models. All of the results in this section are given in Table 7. We only consider the 'LOC' type entities in this section since it is the most relevant entity type for this study.

6.1 Baseline Models

As a baseline method we initially used lookup table programs. The gazetteers were obtained from the GeoNames website[12]. Both of the gazetteers are compilations of Indian place names (cities, towns etc.). Even though the news articles are written in English, there are only a few LOC type entities annotated apart from Indian place names. Thus using a gazetteer which does not include place names from other countries does not necessarily affect the performance of the baseline systems and also speed up the look up process.

We used gazetteers of sizes 11,097 and 130,830 place names which we call Baseline 1, 2 respectively. The lookup model with larger gazetteer has a recall of 0.528 and the precision scores are quite low. Analysis of the errors reveal that the vast number of place names (names of villages, towns etc.) is not captured even in the comprehensive gazetteer from GeoNames which is a characteristic of the data from India.

As additional baseline models we tested two freely available tools. Edinburgh Parser is a GIR system which also performs disambiguation and geolocation of place names in addition to detection. We used the intermediate output of the system which gives the detected place names. Detection is done using a rule based system.

Stanford NER is a widely used NER tool for GIR related. Many freely available tools perform other GIR related tasks such as disambiguation, geolocation and entity linking to the output of Stanford NER.

The results of this section is given in Table 6 and results obtained for Itest is restated in Table 7 for clarity.

Toolname	Test Set	Precision	Recall	F1 Score
Edinburgh Parser	Ctest	0.72	0.78	0.75
	Itest	0.55	0.49	0.52
Stanford NER	Ctest	0.81	0.90	0.85
	Itest	0.72	0.46	0.56

Table 6: Results for Edinburgh Parser and Stanford NER on Ctest and ITest. These are considered as baseline systems we compare our trained models against. We restate the results on Itest again in Table 7 for clarity.

6.2 Trained Models

We started our explorations on the Indian news test set by using the best performing Wapiti model trained on the merged dataset (Wapiti1). Lack of same place names in the training dataset is probably the primary cause of the low recall score. Then we used a different method to make use of the gazetteers. We treated them as separate documents and used them as a part of training corpus. First we appended 2 copies of the smaller gazetteer into the training corpus and we obtained a slight increase in the F1 score. Then we appended 5 copies of the larger gazetteer into the training corpus and trained another Wapiti model. This also resulted in a slight increase in the performance of the Wapiti trained models. The values 2 and 5 are arbitrary but since the gazetteer contains 130k entries and the results did not improve significantly when we increase the value from 2 to 5, we did not further investigate appending different number of copies of the gazetteer.

Then we tested the models that we trained on Atrain using NeuroNER. The model trained only using Atrain (Neuro2) is observed to perform poorly when tested on this test set from a different dataset. Neuro1 which was trained using Conll performs significantly better than the previous NeuroNER model (Neuro2). Third NeuroNER model tested on this Itest is trained by merging Call and Atrain (Neuro3). This final model has the best Precision and F1 score on this test set. This relative improvement obtained by merging corpora is also verified by the results on other test set given in the previous section.

	Precision	Recall	F1 Score
Baseline1	0.428	0.149	0.221
Baseline2	0.212	0.528	0.301
Edinburgh Parser	0.550	0.490	0.520
Stanford NER	0.720	0.460	0.560
Wapiti1	0.607	0.216	0.318
+gazet2	0.567	0.258	0.355
+gazet5	0.567	0.298	0.391
Neuro1	0.711	**0.749**	0.730
Neuro2	0.407	0.516	0.455
Neuro3	**0.779**	0.729	**0.753**

Table 7: Results for all the models on the Itest. In addition to our own simple lookup based models, we also give results for two other tools as baseline. Both of the tools out perform our Wapiti based models. Wapiti1+gazet2,+gazet5 are referred as Wapiti 2, 3 in the Results section respectively. Wapiti1+gazet2 means the Wapiti model is trained with same configurations with Wapiti1 but on the two times gazetteer appended version of Wapiti1's training set.

6.3 Results on the Updated Test Set

We adjusted the entity tags according to the Conll and ACE specifications (we used Place type entities from our annotated dataset as our definition for Place tag is the most similar tag to LOC type in ACE and Conll) and used only the Event sentences since only the entities inside the events are annotated in this corpus. We obtained some results which show that making these changes increases the precision and F1 scores. We tested the best performing models of each tool (Wapiti3 and Neuro3). The results are included in Table 8.

Error Analysis

Although we obtained a relative improvement on Itest we find it important to analyze the errors of our trained models on Itest manually. We believe doing an error analysis provides valuable insights for our future work. Since we will be using a different

[12]http://www.geonames.org/

	Precision	Recall	F1 Score
Wapiti3	0.692	0.458	0.552
Neuro3	0.879	0.700	0.790

Table 8: Results for the updated test set

dataset in the future to measure the performance of our models we believe that looking at the mistakes being made in the testing is not a misconduct. We can safely treat the test set as our development set.

During the error analysis we looked at the errors the Neuro2 model makes since the performance is relatively low especially the precision score meaning that the model makes a lot of wrong predictions of the entities (false positive). So we focused on the false positives first. For each false positive prediction with the LOC label we checked whether the token really is an entity or not. An example line from Itest with a false positive looks as follows:

vadodara test_text_00000 3986 3994 O B-LOC

where the first token 'vadodara' is the word itself, second to last is the Gold label and the last token is the prediction in BIO scheme. We wrote another program to automatically check whether the token is included in the gazetteers we have described earlier for each false positive prediction. If the token is not included in the gazetteers then we did a manual check to determine whether the token is a LOC type entity or not. In total we have 366 false positives of LOC type. We have manually analyzed 120 of them and indeed we saw that 104 of the false positives are either included in the gazetteers or a quick search reveals that they are location names (web-based search). Thus even though the overall precision is around 40% this manual analysis reveals that the precision on this 120 entities is 104/120 which is equal to 0.867 which is significantly higher.

We can use statistics to have a rough estimate about the true performance of our model without going over all of the errors manually. If we assume that the above mentioned trend will continue for the rest of the false positives we will have around 0.85x366 = 310 additional true positives in total. The estimated performance scores after the analysis is given in the Table 9. As the table shows the precision jumps around 40% after the analysis.

	Precision	Recall	F1 Score
Neuro2-before	0.407	0.516	0.455
Neuro2-after	0.820	0.643	0.720

Table 9: Results before and after the error analysis

We believe it is important to point out here that this should not be considered as the flaw of the test set. The test set is annotated for Event Extraction. So only a single occurrence of a LOC type entity which is closest to the Event sentence is annotated. Yet we measured the performance using the conventional metrics so we wanted to give more realistic estimates on the performance of our trained models.

```
near test_text_00000 3422 3426 O O
Vijay test_text_00000 3427 3432 B-ORG B-LOC
Theatre test_text_00000 3433 3440 I-ORG I-LOC
Ground test_text_00000 3441 3447 I-ORG I-LOC
against test_text_00000 3448 3455 O O
```

Figure 2: An example error caused by the difference between annotation guidelines

Error analysis successfully showed that the models can have high precision values for the place name recognition task without suffering from low recall scores. Also error analysis revealed that unseen place names are difficult to detect which is the primary reason of the resulting low recall score.

Finally through error analysis we saw that due to the disagreements between annotation guidelines the precision scores are limited and even misleading. Certain entities have different tags in different corpora which is a limiting factor for any data dependent model. An example to the errors caused by annotation guideline disagreements is given in Figure 2. The entity 'Vijay Theatre Ground' is annotated as Facility in the Indian News annotation guidelines which we mapped to Organization type since Conll tags does not include Facility type. The model predicts 'Vijay Theatre Ground' as a Location type entity which can be considered sensible considering the fact that it refers to a specific place. We can not give an estimate of performance improvement for correcting errors caused by annotation guideline difference because we have not done an extensive analysis as we did for errors caused by unannotated entities.

7 CONCLUSION

In this paper we first did an analysis of two available tools and showed that their performance drops significantly when we test them on our Itest. For example the NeuroNER model which was trained using only Ctrain has a remarkable F1 score of 0.927 for LOC type entities when tested on Ctest. Yet the F1 score drops to 0.672 and 0.730 when the same model is tested on Atest and Itest respectively.

Then we have trained our own models by merging different corpora in various ways with different configurations. Our best performing model outperformed other models trained using only a single corpus and has an F1 score of 0.753. So we have obtained a relative improvement by merging two corpora.

The results show the importance of generalization. This phenomenon becomes even more important when the change is not only in the dataset but also in the way the language is being used. English in Indian News is structured differently causing our models trained on Conll and ACE corpora to fail to detect important patterns and wrongly detect misleading patterns in Indian English. We have seen in our numerous experiments that models trained using NeuroNER and Wapiti perform similarly on Conll and ACE sets. Yet the performance difference is significant when we test them on the Itest. This leads us to conclude that Neural models are better at generalizing on the other hand CRF-like sequential taggers with hand crafted features are more dataset dependent and are more prone to the memorization problem.

The disagreements between the annotation guidelines of our test set and the ACE and Conll corpora is another major limiting factor. The error analysis section explains this issue in more detail.

The findings of this study will direct our future research. We encourage researchers in this area to tackle the issues and challenges described in this paper as we believe that they are critical in creating robust Information Extraction systems.

8 FUTURE WORK

This paper is a part of a larger project aimed at Event Extraction. We have obtained some initial results for Place Name Recognition, mainly to identify the feasibility of the current methodologies available. Thus we will be working on expanding the work described in this project in numerous ways.

As a result of our experiments models developed using NeuroNER outperforms Wapiti models. Thus we are planning to focus our experiments on developing models using NeuroNER or other neural network based tools and develop our own neural network based tools.

We have done some initial explorations with training our own word vectors but at the time of the submission of this paper could not obtain significant improvements over using pretrained word embeddings. Yet we still believe that dataset specific word embeddings contain valuable information. We will keep on training our own word embeddings on the Indian News dataset.

Our annotation team continuously work on expanding our annotated corpora. In future we will be applying the models we have developed on these expanded corpora. In parallel our annotation team will annotate all place names in an event sentence so that the corpus will not contain unannotated place names. Also the corpora will include data from other countries and other languages. So our work will not be limited to English.

As mentioned earlier the annotation guidelines are important limiting factors. Our future work includes using different entity types from ACE dataset which can be more relevant for our task. Especially we can take into account the Facility type entities of ACE as these entities overlap with our own Facility definitions.

The quality and the size of the dataset are the most important aspects of any model that relies on Machine Learning methods especially in the domain of computational linguistics. Thus in future we are planning to expand the size of our training data by using other annotated corpora. The datasets used in FIRE 2013-2014 NER workshops are suitable for our purposes[13]. We believe that incorporating a specific corpus from India will significantly increase the performance of our models. The WNUT dataset may also be used to train and test our models. Another dataset we are planning to include is GROBID NER dataset[14].

9 ACKNOWLEDGEMENTS

We thank our team members for their feedbacks and suggestions regarding this paper.

[13] http://www.au-kbc.org/nlp/NER-FIRE2014/

[14] https://grobid-ner.readthedocs.io/en/latest/training-ner-model/

REFERENCES

[1] N Abinaya, Neethu John, Barathi HB Ganesh, Anand M Kumar, and KP Soman. Amrita_cen@ fire-2014: Named entity recognition for indian languages using rich features. In *Proceedings of the Forum for Information Retrieval Evaluation*, pages 103–111. ACM, 2014.

[2] Beatrice Alex, Kate Byrne, Claire Grover, and Richard Tobin. Adapting the edinburgh geoparser for historical georeferencing. *International Journal of Humanities and Arts Computing*, 9(1):15–35, 2015.

[3] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[4] Isabelle Augenstein, Leon Derczynski, and Kalina Bontcheva. Generalisation in named entity recognition: A quantitative analysis. *Computer Speech & Language*, 44:61–83, 2017.

[5] Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. *arXiv preprint arXiv:1511.08308*, 2015.

[6] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[7] Francisco Couto, Luis Campos, and Andre Lamurias. Mer: a minimal named-entity recognition tagger and annotation server. 04 2017.

[8] Grant DeLozier, Jason Baldridge, and Loretta London. Gazetteer-independent toponym resolution using geographic word profiles. In *AAAI*, pages 2382–2388, 2015.

[9] Franck Dernoncourt, Ji Young Lee, and Peter Szolovits. NeuroNER: an easy-to-use program for named-entity recognition based on neural networks. *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, 2017.

[10] Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association (JAMIA)*, 2016.

[11] Allyson Ettinger, Sudha Rao, Hal Daumé III, and Emily M Bender. Towards linguistically generalizable nlp systems: A workshop and shared task. *arXiv preprint arXiv:1711.01505*, 2017.

[12] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[13] Christopher B. Jones and Ross S. Purves. Geographical information retrieval. *International Journal of Geographical Information Science*, 22(3):219–228, 2008.

[14] Morteza Karimzadeh, Wenyi Huang, Siddhartha Banerjee, Jan Oliver Wallgrün, Frank Hardisty, Scott Pezanowski, Prasenjit Mitra, and Alan M MacEachren. Geotxt: a web api to leverage place references in text. In *Proceedings of the 7th workshop on geographic information retrieval*, pages 72–73. ACM, 2013.

[15] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

[16] Thomas Lavergne, Olivier Cappé, and François Yvon. Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513. Association for Computational Linguistics, July 2010.

[17] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[18] Sailaja Pingali. *Indian English*. Edinburgh University Press, 2009.

[19] Dinesh Kumar Prabhakar, Shantanu Dubey, Bharti Goel, and Sukomal Pal. Ism@ fire-2014: Named entity recognition for indian languages. In *Proceedings of the Forum for Information Retrieval Evaluation*, pages 98–102. ACM, 2014.

[20] Ross S Purves, Paul Clough, Christopher B Jones, Mark H Hall, Vanessa Murdock, et al. Geographic information retrieval: Progress and challenges in spatial search of text. *Foundations and Trends® in Information Retrieval*, 12(2-3):164–318, 2018.

[21] Radim Rehurek and Petr Sojka. Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Citeseer, 2010.

[22] SP Sanjay, M Anand Kumar, and KP Soman. Amrita_cen-nlp@ fire 2015: Crf based named entity extractor for twitter microposts. In *FIRE Workshops*, pages 96–99, 2015.

[23] Erik F Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics, 2003.