Analysis of Various Machine Learning Tools for Place Name Recognition in EmergingWelfare Task*

Arda Akdemir [1] , Ali Hurriyetoglu [2]

***Abstract*— Named Entity Recognition is one of the key tasks in Information Extraction. In this paper, we tackle the task of Location Recognition in the domain of Indian News in English. Our main contribution is that we have done an extensive analysis of several different methods with various configurations on this domain (Indian News in English).**

I. INTRODUCTION

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.

In this paper we analyze various state-of-the-art tools for NER in English in Indian News domain and show that performance drops significantly when we test the tools in different domains even though the language is the same. We especially focus on recognizing place names.The paper is as follows: We start with the related work done in NER in general and in English. 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 and the results we have obtained.

bahsetmeyi dusunduklerim:

- Description of our project
- Importance and challenges of NER and place name recognition for IR and for our project
- Domain degisince basarinin dusmesi
- Indian News ozelindeki problemler ve zorluklar (ilerde)

NOTE: Aadaki blm fire2013 sayfasndan koydum kendimize ile ilgili ksmlar yazabiliriz buraya veya dataset tantma ksmna

A. Challenges in Indian Language NER

Indian languages belong to several language families, the major ones being the Indo-European languages, Indo-Aryan and the Dravidian languages. The challenges in NER arise due to several factors. Some of the main factors are listed below Morphologically rich - identification of root is difficult, require use of morphological analysers No Capitalization feature - In English, capitalization is one of the main features, whereas that is not there in Indian languages Ambiguity - ambiguity between common and proper nouns. Eg: common words such as ”Roja” meaning Rose flower is a name of a person Spell variations - In the web data is that we find different people spell the same entity differently - for example : In Tamil person name -Roja is spelt as ”rosa”, ”roja”.

*Text to be inserted about the project

[1]A. Akdemir is with Department of Computer Engineering, Bogazici University,Istanbul, Turkey arda.akdemir at boun.edu.tr

[2] Text to be inserted about the author

II. RELATED WORK

In this section we go over the previous work done related to this paper. First we give the previous work on NER for English and NER in general. This will be followed by the previous work on Place Name Recognition in English. Finally we mention the work done for finding Indian Place Names in English corpus (var midir emin degilim :))))).

NOTE: Burayi doldurmam icin makale secmemiz lazim. English NER icin makaleler var bende ama diger related work icin arastirmamiz lazim.

Named Entity Recognition is a popular task especially for English and there are various studies in the recent decades.Nadeau et al. [7] gives a very 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.[2] uses a hybrid bi-LSTM and CNN architecture. Their model which uses two lexicons has an F1 score of 91.62. Lample et al.[5] 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.

Place Name Recognition can be considered as a subtask of more general NER with its unique difficulties. Previous work on Named Entity Recognition for Indian corpora in English includes the work done in the FIRE 2013-2014 workshops [1]. Prabhakar et al. [8] 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

[1]http://au-kbc.org/nlp/NER-FIRE2013/index.html

and recall respectively. Abinaya et al.[1] also uses CRF for the same FIRE-2014 task. They use SVM's for other Indian Languages and 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 too much on manual feature extraction. The F1 score for the outer layer which corresponds to the conventional NER is 59.37. Finally, Sanjay et al. [9] 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.

The work done until now on the Indian corpora in English is quite limited and they make use of similar methods(CRF).

III. DATA

NOTE: Indian News datasetimizi de anlatmam lazim burada.

A. Conll-2003 English

One of the datasets we have used is the Conll-2003 English dataset [2] for Conll 2003 Shared Task. For convenience we will be referring to this dataset as Conll corpus for the rest of this paper. 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. 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.

	Train	Valid	Test
LOC	8297	2094	1925
PER	11129	3149	2773
MISC	4593	1268	918
ORG	10025	2092	2496

TABLE I
CONLL2003 ENGLISH CORPUS

[2]https://www.clips.uantwerpen.be/conll2003/ner/.

B. ACE 2005 English

Second dataset we have is the ACE 2005 corpus for automatic content extraction created by Linguistic Data Consortium [3]. The corpus 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. The corpus contains 216545 words in total. The ACE corpus is in XML-format and annotated in a very 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. It is publicly available and can be found in the github repository [4] about this paper. Details about the tokenizer will be given in the following sections.

	Train	Valid	Test
LOC	4176	942	684
ORG	2470	551	293

TABLE II
ACE CORPUS

C. Indian News Corpora

This part explains the corpora we used from Indian News domain. Due to the size limitations we used the first annotated corpora only for testing our models.

The first annotated corpus we have used is annotated by our annotation team of the EmergingWelfare project. This first batch contains 116 annotated documents from Times of India which is news source. The data is tokenized using the UCTO tool and annotated in the FOLIA format[5].The number of entities in this set is shown in Table III. This data is mainly used to have a rough idea about the performances of our models.

	# of Entities
LOC	593
ORG	637
PER	349

TABLE III
INDIAN NEWS TEST CORPUS1

There are several issues about this corpus (Daha sonra detayli anlatabilirim):

[3]https://catalog.ldc.upenn.edu/LDC2006T06

[4]http://anonymous.4open.science/repository/1b9e0e44-54af-403b-81de-3a983b9261d3/

[5]https://proycon.github.io/folia/

- Annotation is done for only single occurrence of a place name entity.
- Almost none of the entities are capitalized. Thus we capitalized the entities manually for this set. Yet this approach can be misleading.
- Another major issue is that our dataset is taken from a specific time period and related to a specific topic (news related to political events). This makes the domain of the test set is quite specific and requires a specific training corpus.

D. 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 we annotated 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.

Note: Burada facility tanimi ve loc tanimini tam ogrenip yazmaliyim.

Main issue is related to the Facility type we have in our Indian News guideline. The definitions in the annotation guideline state 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 (Koc University, X 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 corpos by only changing the FAC type entities to LOC and ORG accordingly. This method requires a lot of manual labor and the benefits is 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.

   Above mentioned issue 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.

IV. TOOLS

This section describes the tools we have used in this project. We will briefly introduce the tools and give general information about them. Then the following subsections will give detailed about each tool separately.

The tools we have used are **Wapiti**[6][6] and **NeuroNer**[7]. 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. All algorithms are quite similar and use the Markov Models. CRF can be considered as a more flexible version of HMMs as it allows any kinds of feature functions.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.
NeuroNER is a tool designed by Dernoncourt et al.[3] specifically for Named Entity Recognition task. It comes together with a model already trained on Conll2003 English dataset 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 very successful methods in NER task in various domains and languages. Following subsections explain each tool in detail separately.

A. Wapiti Toolkit

This section explains the Wapiti toolkit. The section is divided into segments devoted to a specific aspect about the tool. We also give the details related to the experiments done using this toolkit here but the results will be discussed in the results section.

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 which correspond 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. FigureIV-A.1 gives an example from the Conll dataset in this specific format. Note that there are features apart from the words themselves embedded inside the corpus. We excluded these features during training to keep our models simple for now. Annotating the Indian News in the same way is costly and beyond the scope of this paper.

The training and test data must be annotated in the same way. For example BIO metric must be used in both sets. Since the model works by exact matching the features and

[6]https://wapiti.limsi.fr/
[7]https://github.com/Franck-Dernoncourt/NeuroNER

```
-DOCSTART- -X- -X- O

EU NNP B-NP B-ORG
rejects VBZ B-VP O
German JJ B-NP B-MISC
call NN I-NP O
to TO B-VP O
boycott VB I-VP O
British JJ B-NP B-MISC
lamb NN I-NP O
. . O O

Peter NNP B-NP B-PER
Blackburn NNP I-NP I-PER

BRUSSELS NNP B-NP B-LOC
1996-08-22 CD I-NP O
```

Fig. 1. Example Data format of Conll used by Wapiti and NeuroNER

labels must be consistent in test and training sets.

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 very 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 outputs 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 accurate if the test set and the training set is labeled differently.

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. 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) 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. Future work includes using only hand-crafted domain dependent (Indian News) suffix and prefix patterns for a window size of 2. 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.

B. NeuroNER

NeuroNER [3] 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 F-1 score for the frequently used Conll corpus.

NeuroNER is publicly available and comes together with pretrained 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 word and character 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. [4].

=== Wapitiye kiyasla NeuroNER'i kisa tuttum=== daha sonra uzatabilirim

V. TRAINING

We have trained our models on the Conll and ACE corpora. Even though the domain of the training and testing corpora is News articles, the overlapping entities are very scarce because the test set is from Indian News.

We started with reimplementing the previous models on datasets they are already implemented. First we achieved the previously achieved result for our newly trained models on the Conll and ACE corpora. Then we changed the configurations and combined the datasets to have various different models available.

A. Wapiti Models

First, we trained several models using Wapiti on Conll corpus. First model is trained using the recommended configuration in the website of Wapiti. Then second model we trained does not use L-1 regularization which makes the trained model more complicated and takes relatively more time to train. Finally we used a sentence-splitted version of the Conll corpus. Then we tested these models on the two test sets called 'a' and 'b'. Results are given in TableIV. We named each trained model in the order of appearance in the description above(Conll1-2-3).

Next, we trained models using the ACE dataset alone. This dataset is relatively small and we consider these models as baseline for ACE. For training we used 90% of the ACE corpus and testing used the remaining 10%. We trained two different models with 5 and 1 for L1-regularization penalty. Results are shown in TableIV with names ACE1 and ACE2.

Then we merged the 2 datasets together and trained models on this larger corpus. During merging we used all the Conll corpus (including the training validation and test sets) and 70% of the ACE corpus. For testing we used 20% of the ACE corpus. We again trained several models with slightly different configurations. Testing on the ACE test set show that all models have similar performance and no configuration have significant superiority over others. These models are shown in TableIV with names starting with ConllACE.

B. NeuroNER Models

We started with reimplementing the reported state-of-the-art result for Conll corpususing NeuroNER. We have trained a NER model using the exact same configurations with the pretrained model. We achieved the same performance scores for the Conll test set(Table V). This model is our baseline NeuroNER model. For the following trained models we always used the ACE test set which is the 20% of the ACE corpus to measure the performance. The results on the ACE test set are shown in Table VI. Each models name in the table is given according to appearance in this paper. First we tested this state-of-the-art model on ACE test set. The performance dropped significantly compared to the results obtained for Conll test set(Neuro1).

	Precision	**Recall**	**F-1 Score**
Conll1-a	0.846	0.872	0.859
Conll1-b	0.802	0.813	0.808
Conll2-a	0.914	0.916	0.915
Conll2-b	0.872	0.853	0.862
Conll3-a	0.914	0.920	0.917
Conll3-b	0.852	0.862	0.857
ACE1	0.665	0.838	0.741
ACE2	0.688	0.845	0.758
ConllACE1	0.841	0.808	0.824
ConllACE2	0.835	0.818	0.826
ConllACE3	0.826	0.818	0.822

TABLE IV
RESULTS FOR TRAINING WITH WAPITI FOR LOC TYPE ENTITIES

	Precision	**Recall**	**F-1 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 V
RESULTS FOR THE PRETRAINED MODEL OF NEURONER ON CONLL DATASET

Then we trained a new model using ACE training set only. The results show that doing the training and testing on the same domain (ACE corpus) gives better results (Neuro2). Finally we trained new models on the merged corpus (100% of Conll and 70% of ACE) and tested on the 20% of ACE(the remainder 10% of ACE is used as the validation set). Best results are achieved using this model trained on the merged corpus (Neuro3). These results are better than the models trained using Wapiti as well.

	Precision	**Recall**	**F-1 Score**
Neuro1	0.664	0.600	0.630
Neuro2	0.803	0.867	0.834
Neuro3	0.841	0.862	0.851

TABLE VI
RESULTS FOR THE NEURONER MODELS ON ACE TEST SET FOR LOC AND ORG TYPE ENTITIES

Table VII shows the results for the same models for only LOC type entities which is more relevant to our objective in this project. These results show that the best model for ACE still performs better when we only consider the place names.

Until this point, we tested our models on the ACE and Conll 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 the Indian News Domain. We compare the results obtained using each model and using baseline lookup table models.

	Precision	Recall	F-1 Score
Neuro1	0.715	0.633	0.672
Neuro2	0.832	0.914	0.871
Neuro3	0.857	0.922	0.888

TABLE VII
RESULTS FOR THE NEURONER MODELS ON ACE TEST SET FOR ONLY LOC TYPE ENTITIES

VI. RESULTS AND DISCUSSION

In this section we give the results we obtained for our models on Indian News Domain. We start with baseline models as simple as lookup tables and continue with giving results for our various trained models.

NOTE: Buraya simdilik 100 dokumanlik folyada yapilanlari anlatarak basliyorum. Daha sonra yeni test seti gelirse eklicem.

First dataset we have done tests on is the dataset described in Table III. As a baseline method we use lookup table programs. The gazetteers are obtained from the GeoNames website[8]. First we used a gazetteer of size 11097 place names taken from (Baseline1). Second baseline lookup table model uses a larger gazetteer of size 130830 place names. Using this larger gazetteer increased the recall as expected(Baseline2). Yet since we did a case free matching the Indian names which are regular words in English like 'said' and 'will' are predicted wrongly as NEs. This decreased the precision score significantly. Observation of the data reveal that apart from the mistakes mentioned above many of the LOC type entities in this dataset is left unannotated causing the precision to drop. (bu son kisim discussiona da konabilir)

Initial explorations of our trained models is done on the same dataset starting with one of the models trained using Wapiti(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 5 copies of the smaller gazetteer into the training corpus and the result is given as Wapiti2. We obtained a slight increase in the F-1 score. Then we appended 2 copies of the larger gazetteer into the training corpus and trained another Wapiti model (Wapiti3). This also resulted in a slight increase in the performance of the Wapiti trained models.

Then we tested the model that we trained using NeuroNer on Conll corpus and 70% of ACE combined (Neuro3). The model is observed to perform poorly compared to the pretrained NeuroNer model. Our main goal is to outperform the pretrained models since they are already available(Pretrained modeli yazdim ama emin olamadim koymaliyim miyim diye).

Second model is trained only using Conll corpus and the result is given in Table VIII. This model (Neuro4) performs significantly better than the previous NeuroNer model. Third NeuroNer model tested on this Indian News domain is trained using a combination (70% of ACE and Conll training corpus is used for training) of ACE 2005 and Conll 2003 corpora(Neuro5).

	Precision	Recall	F-1 Score
Baseline1	0.290	0.358	0.320
Baseline2	0.428	0.149	0.221
Wapiti1	0.607	0.216	0.318
Wapiti2	0.567	0.258	0.355
Wapiti3	0.567	0.298	0.391
Neuro1	0.407	0.516	0.455
Neuro4	0.711	0.749	0.730
Neuro5	0.779	0.729	0.753

TABLE VIII
RESULTS FOR ALL THE MODELS ON THE TEST SET1 FROM INDIAN NEWS (100 FOLIA DOCUMENTS)

A. Results on the Updated Test Set

First results are obtained using the test set that contained inconsistencies for the Place Name Recognition task. We adjusted the entity tags according to the Conll and ACE specifications 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 solving the inconsistencies increase the performance. The results are included in Table IX

	Precision	Recall	F-1 Score
Wapiti-6	0.458	0.692	0.552
Neuro6	0.700	0.879	0.780

TABLE IX
RESULTS FOR THE UPDATED TEST SET(100 FOLIA DOCUMENTS)

B. Discussion

=== bu yorum en azindan ilk modeller icin dogru belki baska yere konulabilir===
The results show the importance of using domain specific external knowledge. This phenomena becomes even more important when the change is not only in the domain but also in the way the language is being used. The English in Indian News is structured differently causing our models trained on Conll and ACE corpora to fail to detect the important patterns and wrongly detect not important patterns in Indian English. This issue strongly suggests training models in the same domain.

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

Error Analysis

We have used conventional metrics frequently used in IR and NLP tasks. We used the same methodology to evaluate the test results on Conll-2003 English test set and the Indian News test-set1. Yet there are certain issues mentioned earlier about the Indian News test set that make the performance results go down significantly. Thus we found it important to analyze the errors of our trained models manually. The analysis reveals that most of the errors of our models are caused by the inconsistencies and problems related to the dataset itself. In this part we first explain how we did the error analysis and then show in detail the specific issues related to the dataset.

For the error analysis we have investigated the errors of the model on the training and the test datasets. Since we will be using a different 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 Neuro1 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 an entity or not. An example line from the Indian News test-set1 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 0.85x$366 = 310$ additional true positives in total. The estimate performance scores after the analysis is given in the TableX. As the table shows the precision jumps around 40% after the analysis.

	Precision	Recall	F-1 Score
Neuro1	0.820	0.643	0.720

TABLE X
ESTIMATED RESULTS 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 Entity 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.

Error analysis successfully showed that the models can have high precision values for the Place Name Recognition task without suffering from low recall scores.

VII. FUTURE WORK

Future work includes using more patterns for Wapiti such as occurrence in gazetteer or having certain prefix and suffixes. Words that occur frequently around LOC type entities can be included as features as well. Finally we can make use of the POS tag of the words as features.

Gazetteer! We are planning to make use of gazetteers. We already used them in our baseline models for comparing performances. Two main ways to make use of them:

1) Use gazetteers to annotate unannotated corpus. Then use this annotated corpus to train and retrain our models.
2) Binary feature for feature based models (inside gazetteer or not).

We are also planning to use word embeddings trained in the Indian News domain. For now we are using the pretrained word embeddings. Since the domain is very different we have the Out-of-Vocabulary(OOV) problem. Also the embeddings do not reflect the co-occurrence relations in this domain. Since the only input to the NeuroNer models is the word and character embeddings we believe that this will increase the performance significantly.

Data is the most important aspect 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 is suitable for our purposes[9]. We believe that by incorporating a domain specific corpus from India will significantly increase the performance of our models.

We are also planning to further develop these to not only detect place names but detect the place name that is most related to the event mentioned in the document.

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

In terms of the data we use, future work will include taking into account MISC tagged entities to capture location-related information.

========== BURDAYIM ==========

Uzunluk olarak sikinti olacagini zannetmiyorum. Elimmizde 10 sayfayi dolduracak icerik rahatlikla var. Hatta future work olarak dusundugumuz kisimlari ekleyebilirsek kisaltmaya gitmem gerekebilir.

VIII. CONCLUSIONS

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

APPENDIX

Appendixes should appear before the acknowledgment.

ACKNOWLEDGMENT

The preferred spelling of the word acknowledgment in America is without an e after the g. Avoid the stilted expression, One of us (R. B. G.) thanks . . . Instead, try R. B. G. thanks. Put sponsor acknowledgments in the unnumbered footnote on the first page.

References are important to the reader; therefore, each citation must be complete and correct. If at all possible, references should be commonly available publications.

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

[3] 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.

[4] 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.

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

[6] 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.

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

[8] 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.

[9] 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.