Identification of Different Feature Sets for NER tagging using CRFs and its impact

Vijay Sundar Ram R and Pattabhi R.K. Rao and Sobha Lalitha Devi
AU-KBC Research Centre
MIT Campus of Anna University
Chennai, India

***Abstract*- This paper presents a study of the impact of different types of language modeling by selecting different feature matrices in the Conditional Random Fields (CRFs) learning algorithm for Named Entity tagging. We have come up with four different feature matrices and identified features at word, phrase and sentence level. It is identified that the language model which has the structural feature is better than the models with other features.**

I. INTRODUCTION

In this paper, we present a study on how the performance of the Named Entity Recognition (NER) using Conditional Random Fields (CRFs) varies according to different features and feature matrices. Named Entity tagging is a labeling task. Given a text document, named entities such as Person names, Organization names, Location names, Product names are identified and tagged. Identification of named entities is important in several higher language technology systems such as information extraction, machine translation systems.

Named Entity Recognition was one of the tasks defined in MUC 6. Several techniques have been used for Named Entity tagging. A survey on Named Entity Recognition was done by David Nadaeu[6]. The techniques used include rule based technique by Krupka [9], using maximum entropy by Borthwick [4], using Hidden Markov Model by Bikel [3] and hybrid approaches such as rule based tagging for certain entities such as date, time, percentage and maximum entropy based approach for entities like location and organization [16]. There was also a bootstrapping approach using concept based seeds [14] and using maximum entropy markov model [7]. Alegria et al, [1], have developed NER for Basque, where NER was handled as classification task. In their study, they have used several classification techniques based on linguistic information and machine learning algorithms. They observe that different feature sets having linguistic information give better performance.

Lafferty [11] came up with Conditional Random Fields (CRFs), a probabilistic model for segmenting and labeling sequence data and showed it to be successful with POS tagging experiment. Sha and Pereira [17] used CRFs for shallow parsing tasks such as noun phrase chunking. McCallum and Li [12] did named entity tagging using CRFs, feature induction and web enhanced lexicons. CRFs based Named Entity tagging was done for Chinese by Wenliang Chen [21]. CRFs are widely used in biological and medical domain named entity tagging such as work by Settles [18] in biomedical named entity recognition task and Klinger's [8] named entity tagging using a combination of CRFs. The Stanford NER software [10], uses linear chain CRFs in their NER engine. Here they identify three classes of NERs viz., Person, Organization and Location. Here they have used distributional similarity features in their engine, but this utilizes large amount of system memory. This paper discusses different feature sets used and their impacts in CRFs for NER.

The paper is further organized as follows. In Section 2 we have described our approach for identifying the suitable feature matrix. Section 3 presents the different experiments, results obtained and discussion on the performance of each experiment. Section 4 concludes the paper.

II. OUR APPROACH

In this work we have used a machine learning technique for identification of named entities. Here we did four different experiments by varying the feature matrix given to the training algorithm of the machine learning approach to study the performance and to choose the best feature set for identifying the named entities.

We have used Conditional Random Fields (CRFs) for the task of identifying the named entities. CRFs is undirected graphical model, where the conditional probabilities of the output are maximized for a given input sequence. CRFs is one of the techniques best suited for sequence labeling task. Hidden Markov Model (HMM), Maximum Entropy Markov Model (MEMM) and CRFs are well suited for sequence labeling task. MEMM and CRFs allows linguistic rules or conditions to be incorporated into machine learning algorithm. HMM [15] does not allow the words in the input sentence to show dependency among each other. MEMM [2] shows a label bias problem because of its stochastic state transition nature. CRFs, overcomes these problems and performs better than the other two.

A. Conditional Random Fields

CRFs make a first order Markov independence assumption and can be viewed as conditionally trained probabilistic finite state automata.

Now let $o=(o_1,\ldots,o_T)$ be some observed input data sequence, such as a sequence of words in a text document, (the values on T input nodes of the graphical model). Let S be a set of FSM states, each of which is associated with a

label, $l \in L$, (such as PERSON). Let $s=(s_1,\ldots,s_T)$ be some sequence of states, (the values on T output nodes).

Linear-chain CRFs thus define the conditional probability of a state sequence given as follows

$$P_\Lambda(\mathrm{s}|\mathrm{o}) = \frac{1}{Z_\mathrm{o}} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k f_k(s_{t-1}, s_t, \mathrm{o}, t)\right),$$

where Z_0 a normalization factor over all state sequences, $f_k(s_{t-1},s_t,o,t)$ is an arbitrary feature function over its arguments, and λ_k (ranging from $-\infty$ to ∞) is a learned weight for each feature function. A feature function may, for example, be defined to have value 0 in most cases, and have value 1 if and only if s_{t-1} is state #1 (which may have label OTHER), and s_t is state #2 (which may have PERSON or PRODUCT or TITLE label), and the observation at position t in o is a proper noun or an adjective. Higher λ weights make their corresponding FSM transitions more likely, so the weight λ_k in the above example should be positive since the word appearing is any NE category (such as LOCATION or PRODUCT-COMPONENT) and it is likely to be the starting of a named entity.

We have used an open source toolkit for linear chain CRFs called as CRF++ [19].

B. Feature Matrix

The training of the CRFs requires iterative scaling techniques, where a quasi-Newton method such as L-BFGs is used. The input data for the task is processed with part-of-speech tagging (POS) and chunking. Part-of-Speech tagging is done using the Brill's Tagger [5] and text chunking is done using fn-TBL [13]. In the shallow processed text, named entities are manually annotated using a NE tagset, containing Person, Organization, Location, Facility, Product, Product Component, Output, and Output Quality as tags. This processed data is used as the training data.

The choice of features is as important as the choice of technique for obtaining a good Named Entity Recognizer [20]. The selection of the set of feature can improve the results. Here we have presented the NE annotated input in four different forms of feature matrix.

Type1

The complete sentence is represented in the feature matrix. Consider the sentence in the example 1.

(1) "I love point-and-shoots and have no desire at this point to get DSLR".

The feature matrix for this type 1 would be as shown below.

Feature Matrix of Type 1, for example 1

I	PRP	B-NP	B-PERSON
love	VBP	B-VP_act	o
point-and-shoots	NNS	B-NP	o
and	CC	o	o
.	.	.	.
.	.	.	.
get	VB	I-VP	o
DSLR	NN	B-NP	B-PRODUCT
.	.	o	o

Type 2

The feature matrix for the second type is built by taking only the Noun Phrases (NPs) from the training data. From the example 1, we obtain six sequences, because there are six noun phrases in this sentence. A sample of the feature matrix of type 2 is shown below.

Feature Matrix of Type 2, for example 1

I	PRP	B-NP	B-PERSON
point-and-shoots	NNS	B-NP	o
no	DT	B-NP	o
desire	NN	I-NP	o
this	DT	B-NP	o
desire	NN	I-NP	o
this	DT	B-NP	o
point	NN	I-NP	o
DSLR	NN	B-NP	B-PRODUCT

Type 3

Named Entities (NEs) with one preceding word and one following word, is considered from the processed input text to build the feature matrix for the third type. A window of size three is taken. Considering the example 1, we have two named entities, 'I', which has PERSON, NE tag and 'DSLR', having PRODUCT, NE tag. Here for this example we obtain two sequences in the feature matrix as shown below.

Feature Matrix of Type 3, for example 1

:	:	o	o
I	PRP	B-NP	B-PERSON
love	VBP	B-VP_act	o
get	VB	I-VP_act	o
DSLR	NN	B-NP	B-PRODUCT
,	,	o	o

Type 4

In this type, a window of size five is considered. NEs is taken along with two preceding and two following words. Here we provide more contextual information by increasing the size of the window to five. Considering the sentence in example 1, the feature matrix consists of two sequences, where each sequence has one more word added to the left and right of the NE, comparing to the feature matrix of type 3. The sample of feature matrix of type 4 is shown below.

Feature Matrix of Type 4, for example 1

camera NN B-NP I-TITLE
: : o o
I PRP B-NP B-PERSON
love VBP B-VP_act o
point-and-shoots NNS B-NP o
to TO B-VP_act o
get VB I-VP_act o
DSLR NN B-NP B-PRODUCT
, , o o
and CC o o

C. Features of CRFs

The set of features used are word, phrase and structure level. The word level features are words or tokens that occur in the first column of the feature matrix. The word level features are current word, previous to previous word, previous word, next word, next to next word.

Phrase level features include words, POS tags and chunk or phrase information. Phrase level or chunk level features are

(a) current word's POS and chunk information,
(b) current word's POS,
(c) previous word's POS and next word's POS.

The following are sample rules learnt by CRF from phrase level features

Rule P1:

-1 w1 DT
0 w2 NNP PRODUCT

This rule describes if the previous word's POS is determiner (DT) and current word has POS tag as 'NNP' then the current word is tagged as PRODUCT

Rule P2:

-1 w1 JJ NP OUTPUT-QUALITY
0 w2 NN NP OUTPUT-QUALITY

The above rule describes if the previous word's POS is adjective (JJ) and current word's POS is 'NN' then both current word and the previous word will be tagged as 'OUTPUT-QUALITY'.

Structure level features includes features such as

i) Current word given the Current word's POS tag
ii) and Previous word,
iii) Current word given the previous word and its chunk information
iv) Current word given the next word and its POS tag,
v) Current word's chunk information given previous word and its chunk information.
vi) Current word's POS tag given the previous word's NE tag and the next word's NE tag.

Here we consider these to be dependent on each other and find the conditional probabilities. The sample rules learned by CRF engine from the structural features are described below.

Rule S1:

-1 consumes VP
0 w1 NP OUTPUT-QUALITY

The above rule describes if the previous word is 'consumes' which is a verb phrase (VP) then the current word which is a noun phrase (NP) will be tagged as OUTPUT-QUALITY.

Rule S2:

-2 rate/rates/rated VP
-1 w1 NP PROD-COMP
0 w2 NP OUTPUT-QUALITY

The above rule describes if the previous to previous word is 'rate' or 'rates' or 'rated', which is a VP and if the previous word is a NP having the NE tag as Product Component (PROD-COMP) then the current word which is a NP will be tagged as OUTPUT-QUALITY

Rule S3:

-2 w1 NP PERSON
-1 purchased VP
0 w2 NP PRODUCT

If the previous to previous word is a NP with NE tag as PERSON and the previous word is 'purchased' which is a VP then the current word which is a NP will be tagged as PRODUCT.

Using the feature matrices built from input training sentences. The different language models are built by CRFs training algorithm.

Thus the language model LM1 is built from feature matrix of Type1. Here the model learns the structure of the complete sentence, both the structures, where NEs and non- NEs occur. The occurrence of non NEs is more. The language model LM2 is built by training the feature matrix formed by type 2. Here the NEs occurring inside the NPs are learned and rest are not seen by the CRF engine. Using the feature matrix built by type 3, which contains a sequence of window of size three, language model LM3 is formed. This has contextual information of the NEs. The language model LM4 is built using the feature matrix of type 4, which is formed using NE, with a window of size five.

We have performed four different experiments, to study how the performance of the named entity recognizer varies when different language models and different features are used. The experiments and their results are described in the following section.

III. Experiments, Results and Discussion

The training data consists of 1000 documents describing user reviews on different electronic goods such as mobiles, camcorders, notebooks. These documents were obtained from

online trading websites such as Amazon, eBay. The training data consisted of 3107 unique NEs and the number of occurrence of NEs is 24345. The test data consists of similar type of documents. This consists of 456 non-unique NEs. This test data consisted of 94 NEs which were not in the training data. This constitutes 20.6% out-of vocabulary words (OOV words). The test data consisted of 300 unique NEs. The 94 not seen NEs had no repetition, they were unique. Here we have performed CRFs training using the four different feature matrices to build different language models. In the first experiment, language model LM1, is built using CRFs by taking the full sentences in the training data as sequences. The LM1 is taken as the baseline language model. The second experiment, language model LM2 is built by taking the Noun Phrases (NPs) as sequences. In the third experiment, language model LM3 is built by taking NE, with a window of three. In the fourth experiment, language model LM4 is built by NE, with a window of five. The table 1 below, show the results obtained, by doing NE identification on test data using the four different language models.

As we observe the results, in the LM1 model, the learning algorithm learns many rules, from the training data, this makes an overfit, due to which false positives is more and hence the precision is less. In the LM2 model, even though the number of false positives is reduced, and the precision increases slightly, the recall does not increase significantly. In this model, the disambiguation of the NE tags is poor, the learning algorithm does not get any context information, since only NPs are presented to the learning algorithm during training. This does not handle the OOV words. In the LM3 model, we observe that the precision and recall increase significantly. Since in this model the NEs and the preceding and following words are presented to the learning algorithm during training, this gives contextual information, and this learns only the structure of the sentence, where NE can occur. This reduces the number of irrelevant rules, which confuses the learning algorithm. So we obtain better results compared to the first two models. In LM4, we introduce more contextual information to the feature matrix, by considering a window of size five. This helps in learning the structure of the sentence, where the NEs occur more precisely, which increases the recall. Since the feature matrix has the NE with a window of five, more relevant rules are learnt by the system. This reduces the false positives and increases the true positives. The precision increases. As in this model, the structure of sentence, where NE occurs, OOV words also identified. The recall in this model also increases. Hence we obtain a Precision of 96.4% and Recall of 90.1%. The F-measure for this LM4 model is 93.14%.

A. Role of Different Features in Learning

In the table 2 below, the results obtained on using different set of features while learning for the LM4 model are shown.

The word level features and chunk level features help in obtaining rules based on the syntactic information in sentence. The structural features help in learning the sentence structures, where the NEs can occur in a sentence. As this task of NE identification does not require learning of the complete structural information of the sentence.

As we observe in the table 2, when word level features alone are used, the precision is high, but not the recall, because, here the algorithm does not learn sentence structures, and is completely dependent on the words. Hence does not handle out-of-vocabulary (OOV) words. In practice, the real time data consists of OOV words. When we use chunk level features along with the word level features, it is observed that the precision decrease, but the recall increases significantly. This can be explained by the fact that, using chunk level feature, makes the learning algorithm to infer from the POS tags and chunk information, and not just the words alone.

When the structural features are used along with the word and chunk level features, we find that the recall increase significantly, without deteriorating the precision. When the structural features are used, the conditional probabilities calculated are considering the context of the NE, hence this creates a context based model, and makes the CRFs learn the sentence structure well. This in turn helps in handling the 60% of the OOV words.

In the table 3, we find that two NE classes Output and OutputQuality have less recall compared to other NE classes. The occurrence of the NE tags 'Product components', 'Output' and 'OutputQuality' are more ambiguous. For example Nokia N73 is a product NE and its feature such as wifi, 4 mega pixel. mp3 player are tagged as the 'Output' and in the case of a camera, 4 mega pixel is tagged as a Product component. This creates ambiguity, while building the training model. Also these NEs, does not occur enough number of times for the CRFs to learn well. This affects the recall. It was also observed that for the tags Product-Component and Output, the inter annotator agreement is low. This resulted in recall and precision to be lower for both these NE classes. This shows how the inter annotator agreement affects the performance of named entity recognizer.

TABLE I
RESULTS ON USING DIFFERENT LANGUAGE MODELS

Models	Total NEs	NEs Identified	NEs Identified correct	Precision (%)	Recall (%)	F-Measure (%)
LM1	456	388	348	89.7	76.3	82.46
LM2	456	397	362	91.2	79.3	84.83
LM3	456	402	375	93.3	82.2	87.39
LM4	456	426	411	96.4	90.1	93.14

TABLE II
ROLE OF DIFFERENT SET OF FEATURES

Features taken	Total NEs	NEs identified	Correct NEs	Precision (%)	Recall (%)	F-Measure (%)
Word Level	456	320	315	98.4	69.1	81.19
+Chunk level	456	380	352	92.6	77.2	84.20
+Structural Features	456	426	411	96.4	90.1	93.14

TABLE III
NE TAG WISE RESULT BY USING LM4 MODEL

NE Tag	Total NEs	NEs Identified	Correct NEs	Precision (%)	Recall (%)
Person	77	75	73	97.3	94.8
Product	122	115	111	96.5	90.9
Product Component	143	137	133	97.1	93.0
Output	54	48	45	93.7	83.3
Output Quality	60	51	49	96.1	81.7

IV. CONCLUSION

In this work we study the performance of the NE identification task using CRFs by building four different language models by varying the feature matrix constructed from the NE annotated and preprocessed input sentences. The language model, LM4, NEs with a window of five, performs the best of all four. We obtain an F-measure of 93.14%.

We have performed experiments to study the impact of various features on the performance of the NER. Here we have selected three different types of features, word level, chunk or phrase level and structural level. We identify that the best performance is obtained when all the three types of features are used together in learning. If only word level features are used, NER does not handle OOV words, when both chunk level and word level features are used, the learning algorithm does not learn the sentence structures effectively.

We also observe the how the inter annotator agreement plays a vital role in the performance of the NERs using CRFs. It is observed that when the inter annotator agreement is low, the training data consists of ambiguous tagging and this creates ambiguity for the learning algorithm. Hence the performance gets negatively affected.

REFERENCES

[1] Alegria I, Arregi O, Ezeiza N, Fernandez I. Lessons from the Development of Named Entity Recognizer for Basque, *Natural Language Processing*, 36,2006. pp. 25 – 37.

[2] Berger A, Della Pietra S and Della Pietra V. A Maximum Entropy Approach to Natural Language Processing, *Computational Linguistics*, 22(1), 1996

[3] Bikel D M. Nymble: a high-performance learning name-finder, *In Proceedings of the Fifth Conference on Applied Natural Language Processing*, 1997.pp.194-201.

[4] Borthwick A, Sterling J, Agichtein E and Grishman R. Description of the MENE named Entity System, *In Proceedings of the Seventh Machine Understanding Conference (MUC-7)*, 1998.

[5] Brill, Eric. Some Advances in transformation Based Part of Speech Tagging, *In the Proceedings of the Twelfth International Conference on Artificial Intelligence (AAAI- 94)*, Seattle, WA, 1994.

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

[7] Jenny Finkel, Shipra Dingare, Huy Nguyen, Malvina Nissim, Gail Sinclair and Christopher Manning. Exploiting Context for Biomedical Entity Recognition: from Syntax to the Web. *In the Proceedings of Joint Workshop on Natural Language Processing in Biomedicine and its Applications, (NLPBA)*, Geneva, Switzerland, 2004.

[8] Roman Klinger, Christoph M. Friedrich, Juliane Fluck, Martin Hofmann-Apitius. Named Entity Recognition with Combinations of Conditional Random Fields. *In Proceedings of 2nd Biocreative Challenge Evaluation Workshop*, CNIO, Madrid,Spain, 2007.pp. 89-92

[9] Krupka G R and Hausman K. Iso Quest Inc: Description of the NetOwl Text Extraction System as used for MUC-7. *In Proceedings of Seventh Machine Understanding Conference (MUC 7)*, 1998.

[10] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *In the proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, 2005 pp. 363-370.

[11] John Lafferty, Andrew McCallum, Fernando Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *In Proceedings of the Eighteenth International Conference on Machine Learning (ICML-2001)*, 2001.pp.282-289.

[12] Andrew McCallum and Wei Li. Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons, *In Proceedings of Seventh Conference on Natural Language Learning (CoNLL)*, 2003.

[13] G. Ngai and R. Florian. Transformation-Based Learning in the Fast Lane, *In the Proceedings of the NAACL'2001*, Pittsburgh, PA, 2001.pp.40-47

[14] C. Niu, W. Li, Rohini K. Srihari. Bootstrapping for Named Entity Tagging using Concept-based Seeds. *In Proceedings of HLT-NAACL'03, Companion Volume*, Edmonton, 2003.pp.73-75.

[15] Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *In Proceedings of the IEEE*, 77(2), 1989.pp.257– 286.

[16] Rohini K Srihari, C.Niu and W.Li. Hybrid Approach for Named Entity and Sub-type Tagging. *In Proceedings of Applied Natural Language Processing Conference*, Seattle, 2000.pp.247-254.

[17] Fei Sha and Fernando Pereira. Shallow Parsing with Conditional Random Fields, *In the Proceedings of HLT-NAACL 03*, 2003. pp.213-220

[18] Settles B. Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets. *In Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA)*, Geneva, Switzerland, 2004.pp.104-107.

[19] Taku Kudo. CRF++, an open source toolkit for CRFs, http://crfpp.sourceforge.net, 2005.

[20] Tjong Kim Sang, Erik. F.; De Meulder, F. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. I*n Proceedings of Conference on Natural Language Learning*. 2003

[21] Wenliang Chen, Yujie Zhang and Hitoshi Isahara. Chinese Named Entity Recognition with Conditional Random Fields. *In Proceedings of Fifth SIGHAN Workshop on Chinese Language Processing*, Sydney, 2006.pp.118-121.