Revisiting CoNNL-2003 with Classical Machine Learning

Matthew A. Hernandez
Linguistics Department
University of Arizona
mah8@arizona.edu

Abstract

This paper presents a brief survey on classical machine learning algorithms in the context of the CoNNL-2003 Shared Task. A Named Entity Recognition (NER) system is built to recognize and classify objects in a body of text into predefined categories. The paper includes a principled framework that motivates the use of machine learning. Finally, the paper includes an analysis between generative and discriminative machine learning algorithms with various decoding methods for inference.

1 Introduction

The Named Entity Recognition and Classification (NERC) task refers to the process that identifies phrases in text into various categories. Common entity types include `ORGANIZATION (ORG)`, `PERSON (PER)`, `LOCATION (LOC)`, and `MISCELLANEOUS (MISC)`. Tagging is the process of labeling each word in a sentence with its respective named entity (NE). As an example:

The [`The University of Arizona ORG`] is a public land-grant research university.

The University of Arizona is tagged as `ORGANIZATION` and the remaining words are outside the entity. Note the dominant categorization is subjective as one can argue *Arizona* is also a `LOCATION`. Approaches to NERC generally use BIO notation which separates the beginning (B-) and inside (I-) of entities. Otherwise the label outside (O) is used.

NER remains a relevant Natural Language Processing task that benefits several extrinsic tasks such as question answering and information extraction. The traditional schema largely ignores nested entities in favor of flat structures that are easier to identity but frames the problem with missing information. Our paper is not interested in domain-specific application, however the loss of information can become disadvantageous for some researchers [1]. State-of-the-art (SOTA) methods make use of bidirectional Long Short-Term Memory (LSTM) networks with a Conditional Random Field (CRF) layer [2].

The paper describes a baseline & benchmark methodology for developing a NER system in the context of the CoNNL-2003 Shared Task [3]. The scope of this paper will focus on the English dataset[1] and compare the Spanish corpus from the CoNNL-2002 Shared Task [4].

The layout of the paper is as follows. Section 2 gives a formal description of the algorithms. Section 3 implements two baselines to motivate machine learning and assess performance. Section 4 provides a discussion of our various systems. Finally, Section 5 describes limitations of our work.

[1]The German corpus is available only via paid membership.

2 Methods

2.1 HMM Taggers

Suppose $X = (X_1, ..., X_n)$ is a sequence of random variables taking discrete values in finite set $T = (t_1, ..., t_k)$, the state space. A first-order Markov model is predicated on the Markov assumption where the probability of the next time step is conditioned only on the current state:

$$P(X_{n+1} = t|X_1, ..., X_n) = P(X_{n+1} = t|X_{n-1})$$

We will assume the state variables X_n are NE tags whose state sequence is not directly observable. Given a sequence of n observations $W = (w_1, ..., w_n)$, we are interested in the best sequence of tags that corresponds to n observations:

$$\hat{t}_1^n = \underset{\hat{t}_1^n}{\text{argmax}}\, P(t_1^n|w_1^n) \tag{1}$$

This probability cannot be computed easily, therefore we apply Bayes' rule and simplify the equation by recognizing the denominator is constant for each sentence:

$$\hat{t}_1^n = \underset{\hat{t}_1^n}{\text{argmax}}\, P(t_1^n)P(w_1^n|t_1^n)$$

The equation and joint probability is made explicit in Equation (2):

$$P(X, W) = \hat{t}_1^n \approx \underset{\hat{t}_1^n}{\text{argmax}} \prod_{i=1}^{T} P(t_i|t_{i-1})P(w_i|t_i) \tag{2}$$

The joint probability is factorized into likelihood and prior probabilities that is easily computable via maximum likelihood estimation (MLE) [2]. The presence of unknown words at prediction time is problematic without smoothing. While the model's probabilities are ameliorated with smoothing, it remains brittle for entity recognition because it assumes independence between features and doesn't allow for new features to be easily incorporated.

2.2 Maximum-Entropy Taggers

The motivation for better modeling is clear: we want to utilize global, local, and dependent features for our representation at each time step. The Maximum-entropy model directly computes the conditional probability from Equation (1). The likelihood and prior are now replaced by a single term:

$$\hat{t}_1^n \approx \underset{\hat{t}_1^n}{\text{argmax}} \prod_{i=1}^{T} P(t_i|t_{i-1}, w_i) \tag{3}$$

Earlier we informally stated Equation (1) is not analytically computable. Therefore, we resort to a point approximation method know as Gradient Descent for learning the parameters[3] of the model [5]. Equation (4) is now possible:

$$\hat{t}_1^n \approx \underset{\hat{t}_1^n}{\text{argmax}} \prod_{i}^{T} \frac{1}{Z(W, t_{i-1})} exp\{\mathbf{W} \cdot f(t_{i-1}, w_i)\} \tag{4}$$

In practice, we condition on several features $\vec{x}_i$ where $f(t_{i-1}, w_i)$ becomes $f(\vec{x_i})$. Lastly, Equation (4) is a Multinomial Logistic Regression in disguise. Thus our final equation is:

$$\hat{t}_1^n \approx \underset{\hat{t}_1^n}{\text{argmax}} \prod_{i}^{T} softmax(\mathbf{W}\vec{x}_i + \vec{b}) \tag{5}$$

[2] MLE for discrete variables counts how many times each event occurs and normalizes the counts into a probability distribution. Our implementation makes use of Python dictionaries to compute the transition and emission probabilities.

[3] The weight parameter $\mathbf{W}$ is not the same as the word sequence W. The literature often refers to this sequence as O for observations.

3 Results

3.1 Baselines

Two baseline systems `AlwaysNonEntity` and `SingleEntity` were computed for the English and Spanish corpora (Tables 1 & 2). As evidenced with the `AlwaysNonEntity` baseline, the token-level accuracy has a modest 80% but is meaningless for NEs. The improved baseline[4] `SingleEntity` labels entities only if they appear in the training data.

Model	Dev		Test	
	Acc	F1	Acc	F1
`AlwaysNonEntity`	83.2	0.0	82.2	0.0
`SingleEntity`	86.2	40.0	84.8	**35.3**

Table 1: English dev and test set

Model	Test	
	Acc	F1
`AlwaysNonEntity`	88.0	0.0
`SingleEntity`	74.4	**16.9**

Table 2: Spanish test set

Therefore, we adopt exact-match (macro-F1) as stated by Tjong Kim Sang and De Meulder [6], "precision is the percentage of NEs that are correct. Recall is the percentage of NEs in the corpus. A NE is correct only if it is an exact match of the corresponding entity."

The baselines suggest NEs in Spanish w.r.t the dataset contain longer phrases when compared to English.

3.2 Experiments

We applied both Hidden Markov and Maximum-Entropy models to the NERC task. For the English dataset we tested several variants of each algorithm by using various add-α smoothing (α = 1 or > 1) values and used grid search to find the optimal values for regularization, epochs, and training rate.

Run Description	**Decoding**	**Test**	**LOC**	**MISC**	**ORG**	**PER**	**Overall**
1. MEMM-sk	Greedy	Val	91.	80.	76.	90.	**84.53**
		Test	84.	72.	73.	85.	**77.81**
2. (Me)MM	Greedy	Val	89.	78.	75.	90.	83.11
		Test	82.	70.	69.	85.	75.64
3. HM(Me)	Viterbi	Val	86.	82.	71.	76.	79.01
		Test	80.	72.	62.	57.	68.45

Table 3: English evaluation on tuned models.The (Me) caption indicates the model is built from scratch. Models with the -sk suffix are imported from Scikit-learn.

While the Spanish dataset was evaluated exclusively by an HMM, the results in Table 4 suggest a first-order model is an appropriate benchmark for the task. We obtained an $F_{\beta=1}$ score of 67.5% which is an improvement over baselines and the transformation-based model proposed by Villalba and Guzmán [7] ($F_{\beta=1}$: 60%).[5]

Run Description	**Decoding**	**Test**	**LOC**	**MISC**	**ORG**	**PER**	**Overall**
1. HM(Me)	Greedy	Test	71.	37.	72.	69.	**67.5**

Table 4: Spanish evaluation on tuned model. The (Me) caption indicates model is built from scratch.

[4]The less naive baseline is based on a lookup table where only beginning (B-) of entities are added.

[5]Brill's Tagger was the first to use transformation-based learning for part-of-speech tagging. The results compare our tagger with this approach as it is based on linguistic intuition. Modern approaches often ignore linguistic cues and use advanced machine learning models to do the heavy lifting.

The maximum-entropy model was introduced to incorporate hand-crafted features that better represent each time step. Our representation contains a small set of hand-crafted features and word embeddings. We used fastText embeddings that are composed of subword information which is useful as new entities are often not present in static embeddings; fastText embeddings also achieve the best results over Word2Vec embeddings [8].

We compared the results of the MEMM imported from scikit-Learn and our implementation in Table 3. We are satisfied that the performance of these two models is almost identical suggesting our implementation is on par with scikit-learn's version (except for runtime). Concerning the optimization problem the weights were updated using Gradient Descent instead of Generalized Iterative Scaling (GIS), the original parameter estimation for ME models [9].

Traditionally, the decoding strategy for HMM uses the Viterbi algorithm which finds the most optimal state by maximizing the probability of sensible transitions. That is, at each time step the most probable sequence considers all possible transitions between states. Greedy search is another decoding strategy that makes a hard decision at each time step and is much faster as it evaluates only one possible label per token.

The MEMM performed the best empirically from greedy search and had better generalization properties as the validation score did not drop drastically for the test data. Moreover the HMM had a near 10% drop suggesting that the model is overfitting to the data. One possible explanation is that the large value for the add-α smoothing pushes for a uniform probability which is not useful for prediction. We must consider that the HMM is a first-order model where a second-order model may yield better results.

4 Conclusion

We described our algorithms for tagging by implementing two models from scratch and compare their performance against a model imported from scikit-learn. The maximum-entropy approach clearly suggests discriminative modeling gives good results (see Table 5) with word embeddings; it easily outperforms the `AlwaysNonEntity` and `SingleEntity` baseline and beats a basic first-order HMM as well.

English val	Precision	Recall	$F_{\beta=1}$
LOC	94%	88%	91
MISC	83%	78%	80
ORG	79%	74%	76
PER	89%	91%	90
Overall	86%	83%	84

English test	Precision	Recall	$F_{\beta=1}$
LOC	85%	82%	84
MISC	73%	72%	72
ORG	73%	70%	73
PER	85%	85%	85
Overall	80%	78%	79

Table 5: Results for the development and test evaluations for the English task.

The inclusion of word embeddings yielded competitive performance but suggest more hand-crafted rules are necessary for better results. We hypothesize with a better feature template our model can achieve near SOTA results. The word embeddings were incorporated to prevent high dimensional features in our model but we still used hand-crafted features that increased the dimensionality of our model. We are not interested in language-specific feature engineering however, and conclude that more complex machine learning (i.e., neural networks and conditional random fields) is better suited for this task.

5 Limitations

The task of named entity recognition assumes thousands of sentences are annotated with named entities, implying that new datasets require extensive annotation. New NER datasets are not often created. Therefore the CoNNL-2003 dataset is adopted as the de-facto evaluation dataset for English NER. This is a severe limitation of our work does not perform extrinsic evaluation to assess the quality of our model and if it can improve a downstream task.

The task not only requires extensive annotation, these datasets are subject to neologisms, temporal drift, and SOTA methods often ignore linguistic intuitions [10]. It is however unclear how well models trained on CoNNL-2003 will perform on modern data but Liu and Ritter [11] have shown NER models have very different generalization. This implication suggests certain models (which we did not attempt) are better suited for the task.

We declared that the first-order Hidden Markov Model is a solid baseline for the English and Spanish dataset, however our work did not attempt a second-order (trigram) model which may have yielded better results. The work proposed by Mayfield, McNamee, and Piatko [12] used Brant's TnT Tagger (Trigrams & Tags) as a baseline for their system achieving 82.90% on the validation set and 75.54% on the test set.

The strict evaluation (exact-match) severely penalizes entities that are not perfect and may not be the best metric for evaluating the performance of a NER system. This is another severe limitation for our work as it does not compare other evaluation metrics, such as partial credit when an entity is identified, may be more appropriate.

The performance of our model greatly improves after implementing a "hack" that cleans up sequences of entities that are not sensible. For example, the sequence `[B-ORG I-PER]` is converted to `[B-ORG I-ORG]`. A better performing NER system would likely not resort to this hack. Lastly, we did not provide a deep analysis of the benefits of using the fastText embeddings over the Word2Vec embeddings.

6 Appendices

- Derivation for softmax?

- Grid Search

- Full performance results?

References

[1] M. Saef Ullah Miah, Junaida Sulaiman, Talha Bin Sarwar, Saima Sharleen Islam, Mizanur Rahman, and Md. Samiul Haque. Medical Named Entity Recognition (MedNER): A Deep Learning Model for Recognizing Medical Entities (Drug, Disease) from Scientific Texts. In: *IEEE EUROCON 2023 - 20th International Conference on Smart Technologies*. 2023, pp. 158–162.

[2] Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. A Neural Layered Model for Nested Named Entity Recognition. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Ed. by Marilyn Walker, Heng Ji, and Amanda Stent. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 1446–1459.

[3] 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*. 2003, pp. 142–147.

[4] Erik F. Tjong Kim Sang. Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In: *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*. 2002.

[5] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In: *Proceedings of the Twenty-First International Conference on Machine Learning*. ICML '04. Banff, Alberta, Canada: Association for Computing Machinery, 2004, p. 116.

[6] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: *Proceedings of CoNLL-2003*. Edmonton, Canada, 2003.

[7] Diego Alexander Huérfano Villalba and Elizabeth León Guzmán. *Named Entity Extraction with Finite State Transducers*. 2020.

[8] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. Enriching Word Vectors with Subword Information. In: *CoRR* abs/1607.04606 (2016).

[9] Andrew McCallum, Dayne Freitag, and Foster Provost. Maximum Entropy Markov Models for Information Extraction and Segmentation. In: *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*. Morgan Kaufmann. 2000, pp. 591–598.

[10] Haris Riaz, Razvan-Gabriel Dumitru, and Mihai Surdeanu. *ELLEN: Extremely Lightly Supervised Learning For Efficient Named Entity Recognition*. 2024.

[11] Shuheng Liu and Alan Ritter. Do CoNLL-2003 Named Entity Taggers Still Work Well in 2023? In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 8254–8271.

[12] James Mayfield, Paul McNamee, and Christine Piatko. Named Entity Recognition using Hundreds of Thousands of Features. In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*. 2003, pp. 184–187.