Revisiting CoNNL-2003 with Classical Machine Learning

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

Abstract

We present 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. We include a principled framework that motivates the use of machine learning by creating two baseline systems. Finally, the paper includes an analysis between generative and discriminative machine learning algorithms.

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

Note the 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. The named entity above is now:

[`The B-ORG`] [`University I-ORG`] [`of I-ORG`] [`Arizona I-ORG`]

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; we are not interested in domain-specific application however. State-of-the-art (SOTA) methods make use of bidirectional Long Short-Term Memory (LSTM) networks with a Conditional Random Field (CRF) layer [1].

The paper describes a baseline & benchmark methodology for developing a NER system in the context of the CoNNL-2003 Shared Task [2]. This paper will focus on the English dataset[1] and compare the Spanish corpus from the CoNNL-2002 Shared Task [3].

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 and the overall framework of the task. 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 the finite state space $T = (t_1, ..., t_k)$. A first-order Markov model is predicated on the Markov assumption, where the chain of probabilities is conditioned on the current state:

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

We will assume the state variables X_n are NE tags whose 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 $\hat{t}_1^n$ 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 the joint probability are 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 (emission) and prior (transition) probabilities that are easily computable via maximum likelihood estimation.[2]

Unobserved items in the training data will have zero probabilities that eliminate valid sequences from the output. Therefore, we ameliorate this sparsity with smoothing. However, the model is inherently probabilistic and is less flexible if we want to model additional features that maintain the distribution.

2.2 Maximum-Entropy Taggers

The motivation for better modeling is clear: we want to utilize global and local features for prediction. For example, knowing whether a word is capitalized may be a useful predictor for the PER tag. We look to maximum-entropy modeling, a framework for incorporating information for classification to achieve this. We can now model Equation (3) explicitly:

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

Each feature is a constraint λ_i (a Lagrange multiplier) in the model. The solution to the constrained optimization problem is in fact equivalent to the probability distribution of a multinomial logistic regression[3] under MLE [4]. In practice however, we condition on a vector of several features $\mathbf{x}$ where $f(t_{i-1}, w_i)$ becomes $f(\mathbf{x})$. Thus our final equation is:

$$\hat{t}_1^n \approx \underset{\hat{t}_1^n}{\text{argmax}} \prod_{i}^{T} \text{softmax}(\mathbf{W}\mathbf{x}_i + \mathbf{b}) \tag{4}$$

[2]This estimation technique counts how many times each event occurs and normalizes the counts into a probability distribution.

[3]$\mathbf{W}$ is a $k \times f$ matrix for the model with k labels and f features. The matrix is not to be confused with the word sequence W. In the literature this sequence is referred to as O for observations.

Equation (4) is not analytically tractable; therefore, we resort to gradient descent, a point approximation for learning the parameters of the model [5]. The interested reader may visit Berkeley's 2024 course offering on sequence modeling for more information.[4]

2.3 Feature Functions

Our feature representation consists of a small set of indicator functions, gazetters and Word2Vec embeddings. A gazetteer refers to a compiled list of known entities [6]. In Section 4 we detail how the inclusion of each feature contributes to the performance of the system.

2.4 Decoding

The decoding strategy for the HMM is the Viterbi algorithm that finds the most optimal state by considering all possible transitions between states. Greedy search is the strategy used for MEMMs that makes a hard decision at each time step.

3 Results

3.1 Baselines

Two rule-based systems were computed for the English and Spanish corpora with the seqeval package [7]. The `AlwaysNonEntity` is a naive baseline that labels all tokens as 'O' or as non-entities, see Table 1. The less naive baseline `SingleEntity` is based on a lookup table where only the beginning (B-) of entities are added, see Table 2.

Model	Dev		Test	
	Acc	F1	Acc	F1
`AlwaysNonEntity`	83.3	0.0	82.5	0.0
`SingleEntity`	86.5	40.0	84.9	**35.4**

Table 1: Results on English dev and test set (Accuracy and Micro F1).

Model	Test	
	Acc	F1
`AlwaysNonEntity`	88.0	0.0
`SingleEntity`	74.5	**16.9**

Table 2: Results on Spanish test set.

As evidenced with the `AlwaysNonEntity` baseline, the token-level accuracy has a modest 80% but is meaningless for NEs. The improved baseline `SingleEntity` labels entities only if they appear in the training data and is the baseline for our system. The English dataset is split into train, development, and test. The Spanish is only split into training and test sets. See Appendix 6.3 for information on the spread of the English labels.

3.2 Experiments

We compare ME models and HMM to the NERC task. 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_1 score of 67.5% which is an improvement over baselines and the transformation-based model proposed by Villalba and Guzmán [8] (F_1: 60%).[5]

Run Description	Decoding	Test	LOC	MISC	ORG	PER	Overall
1. HM(Me)	Greedy	Test	71.1	37.3	72.5	69.9	**67.9**

Table 3: Spanish evaluation on tuned ($\alpha = 100$) model. The (Me) caption indicates model is built from scratch.

For the English dataset we tested several variants by using various add-α smoothing ($\alpha = 1$ or > 1) values and used pairwise grid search to find optimal values for regularization, epochs,

[4] https://www.cse.iitd.ac.in/ mausam/courses/csl772/autumn2014/lectures/09-memm.pdf

[5] Brill's Tagger is an instance of transformation-based learning for another sequence task, part-of-speech tagging. It is used for comparison because it is based on linguistic intuition.

and learning rate. We compared the results of the MEMM imported from scikit-learn and our implementation in Table 3.

Run Description	Decoding	Test	LOC	MISC	ORG	PER	Overall
MEMM-sk	Greedy	Val	89.3	79.1	75.2	89.3	**84.5**
λ=0.1, η=0.1, epochs=15		Test	84.0	72.2	70.4	83.5	**78.3**
(Me)MM	Greedy	Val	89.7	79.4	74.5	88.0	84.2
λ=0.1, η=0.1, epochs=15		Test	84.0	70.4	68.8	82.7	77.6
HM(Me)	Viterbi	Val	85.7	82.5	71.4	76.1	79.1
α=100		Test	80.3	71.6	62.4	57.3	68.2

Table 4: English evaluation on tuned models. The (Me) caption indicates the model was built from scratch and -sk indicates it was imported from scikit-learn. λ is regularization, η is the learning rate, and α is the smoothing value.

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. The MEMM performed and generalized the best with greedy search as the validation score did not drop drastically for the test data. We report the final results for the MEMM in Tables 5 & 6.

English val	Precision	Recall	F_1
LOC	91%	88%	89
MISC	91%	77%	79
ORG	74%	76%	75
PER	91%	88%	89
Overall	86%	83%	84.5

Table 5: Development evaluation

English test	Precision	Recall	F_1
LOC	84%	84%	84
MISC	72%	72%	72
ORG	70%	71%	70
PER	86%	81%	83
Overall	80%	78%	78.3

Table 6: Test evaluation

4 Conclusion

We described the performance of our algorithms and their variants in Tables 3 & 4. The ME approach clearly shows discriminative modeling gives good results; it easily outperforms the `AlwaysNonEntity` and `SingleEntity` baselines and beats a first-order HMM as well.[6]

The discussion of generative and discriminative modeling by Ng and Jordan [9] also suggests logistic regression will eventually catch up and overtake the performance of a generative classifier given enough data. This implication may be extended to deep neural networks that already hold SOTA performance.

	Validation	Test	Time (s)	# Parameters
Features	75.7%	68.3%	91	129,754
Feats + Gazetteers	79.5%	71.5%	86	129,756
Feats + Gazetteers + $\mathbf{E}_i$	84.6%	78.4%	111	130,056
Feats + Gazetteers + $\mathbf{E}_i$ + $\mathbf{E}_{i-1}$	84.5%	**78.3**%	159	130,356
Feats + Gazetteers + $\mathbf{E}_i$ + $\mathbf{E}_{i-1}$ + $\mathbf{E}_{i+1}$	83.3%	77.3%	190	130,656

Table 7: Effect of ME model size with different feature sets. 26 hand-crafted features were used. The embeddings $\mathbf{E}$ of previous, current, and next tokens were used.

Therefore, we conclude that advancements in NLP (i.e., embeddings) have improved our model (see Table 7) with little feature engineering and can reach competitive performance with tuning or different modeling, see Appendix 6.5 for a comparison of model size. We are not interested in performance however, and propose the CoNNL-2003 dataset is dated and we should instead access the generalization of these models or evaluate nested entities. We may also consider the upper bound of new named entities in our evaluation.

[6]The HMM had a near 10% drop suggesting the model is overfitting to the data. One possible explanation is large α smoothing values push for uniform probability that is not useful for prediction.

5 Limitations

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

NER datasets are subject to neologisms, temporal drift, and SOTA methods often ignore linguistic intuitions [10]. That is to say, it is unclear how well models trained on CoNNL-2003 will perform on modern data. 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, but our work did not attempt a second-order (trigram) model which may have yielded better result. The inclusion of a second-order model may have cemented the preference for discriminative modeling versus generative modeling. However, 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 severely penalizes entities that are not an exact-match and may not be the best metric for evaluating the performance. We did not compare other evaluation metrics, such as partial credit, but deem this evaluation to be task-dependent.

The largest limitation of our work is that the performance of our model improves after implementing a post-processing step disallows sequences 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.

6 Appendix

6.1 Micro-Averaged F1

We adopt micro-F1 to better represent class imbalances as stated by Tjong Kim Sang and De Meulder [13], "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 reported "Overall" metric is simply the micro-averaged F_1 measure.

6.2 Timing Experiments

The MEMM requires less time training than the (Me)MM implementation. The results are comparable for the two models therefore the final results are from the scikit-learn model to produce the comparison of features. Training the MEMM takes around 2 minutes and nearly 16 hours for the (Me)MM version. All timing experiments were performed on a single core of an Apple M1 with 8GB of RAM.

6.3 Dataset split

The English dataset was preprocessed into training, development, and test sets, see Figure 2. Additionally, the number of tokens per entity type (including O) highlight the unbalanced nature of the CoNNL-2003 dataset and of NER in general.

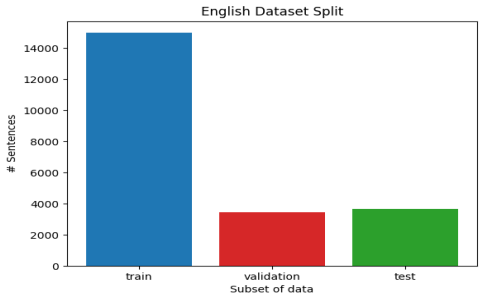

Figure 1: Dataset split

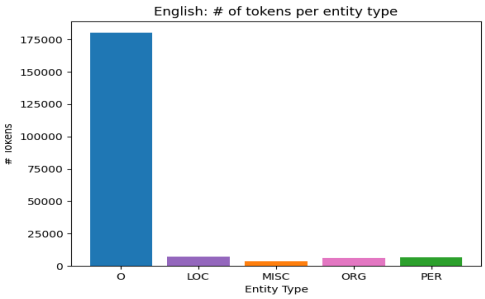

Figure 2: Number of tokens

Figure 3: Comparison of dataset split and number of tokens

6.4 Grid Search

Logistic regression with gradient descent has at least three hyperparameters that need to be tuned: a regularization constant λ, a learning rate η, and the number of epochs. The array of reasonable values is:

$$\lambda \in \{1e-04, 1e-05, 1e-06, 1e-07\}$$

$$\eta \in \{1e-01, 1e-02, 1e-03, 1e-04\}$$

$$\text{epochs} \in \{5, 10, 15, 20\}$$

6.5 Effect of Model Size

The motivation for Maximum-entropy modeling is clear, the addition of features allows for a model to recognize difficult entities. This increase in features, and in effect the parameters, improved the performance of the system. That is, the combination of hand-crafted features, gazetteers, and different embeddings each contributed to the performance.

The performance neared a plateau after including previous, current, and next word embeddings. We did not try further embeddings to respect the first-order Markovian assumption.

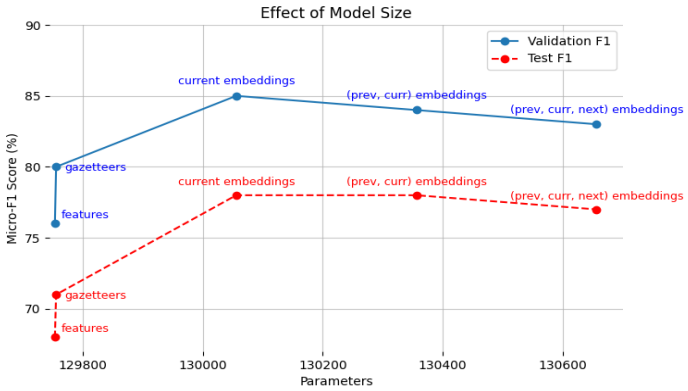

Figure 4: Inclusion of feature set

References

[1] 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.

[2] 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.

[3] 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.

[4] Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. A Maximum Entropy Approach to Natural Language Processing. In: *Computational Linguistics* 22.1 (1996). Ed. by Julia Hirschberg, pp. 39–71.

[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] Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. Towards Improving Neural Named Entity Recognition with Gazetteers. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 5301–5307.

[7] Hiroki Nakayama. *seqeval: A Python framework for sequence labeling evaluation*. Software available from https://github.com/chakki-works/seqeval. 2018.

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

[9] Andrew Ng and Michael Jordan. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In: *Advances in Neural Information Processing Systems*. Ed. by T. Dietterich, S. Becker, and Z. Ghahramani. Vol. 14. MIT Press, 2001.

[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.

[13] 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.