Named Entity Recognition Only from Word Embeddings

Ying Luo, Hai Zhao [*] and **Junlang Zhan**

Department of Computer Science and Engineering, Shanghai Jiao Tong University
Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China
MoE Key Lab of Artificial Intelligence, AI Institute,
Shanghai Jiao Tong University, Shanghai, China
`kingln@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn,`
`longmr.zhan@sjtu.edu.cn`

Abstract

Deep neural network models have helped named entity recognition achieve amazing performance without handcrafting features. However, existing systems require large amounts of human annotated training data. Efforts have been made to replace human annotations with external knowledge (e.g., NE dictionary, part-of-speech tags), while it is another challenge to obtain such effective resources. In this work, we propose a fully unsupervised NE recognition model which only needs to take informative clues from pre-trained word embeddings. We first apply Gaussian Hidden Markov Model and Deep Autoencoding Gaussian Mixture Model on word embeddings for entity span detection and type prediction, and then further design an instance selector based on reinforcement learning to distinguish positive sentences from noisy sentences and then refine these coarse-grained annotations through neural networks. Extensive experiments on two CoNLL benchmark NER datasets (CoNLL-2003 English dataset and CoNLL-2002 Spanish dataset) demonstrate that our proposed light NE recognition model achieves remarkable performance without using any annotated lexicon or corpus.

1 Introduction

Named Entity (NE) recognition is a major natural language processing task that intends to identify words or phrases that contain the names of PER (Person), ORG (Organization), LOC (Location), etc. Recent advances in deep neural models allow us to build reliable NE recognition systems (Lample et al., 2016; Ma and Hovy, 2016; Liu et al.,

2018; Yang and Zhang, 2018; Luo et al., 2020; Luo and Zhao, 2020). However, these existing methods require large amounts of manually annotated data for training supervised models. There have been efforts to deal with the lack of annotation data in NE recognition, (Talukdar and Pereira, 2010) train a weak supervision model and use label propagation methods to identify more entities of each type; (Shen et al., 2017) employ Deep Active Learning to efficiently select the set of samples for labeling, thus greatly reduce the annotation budget; (Ren et al., 2015; Shang et al., 2018; Fries et al., 2017; Yang et al., 2018b; Jie et al., 2019) use partially annotated data or external resources such as NE dictionary, knowledge base, POS tags as a replacement of hand-labeled data to train distant supervision systems. However, these methods still have certain requirements for annotation resources. Unsupervised models have achieved excellent results in the fields of part-of-speech induction (Lin et al., 2015; Stratos et al., 2016), dependency parsing (He et al., 2018; Pate and Johnson, 2016), etc. Whereas the development of unsupervised NE recognition is still kept unsatisfactory. (Liu et al., 2019) design a Knowledge-Augmented Language Model for unsupervised NE recognition, they perform NE recognition by controlling whether a particular word is modeled as a general word or as a reference to an entity in the training of language models. However, their model still requires type-specific entity vocabularies for computing the type probabilities and the probability of the word under given type.

Early unsupervised NE systems relied on labeled seeds and discrete features (Collins and Singer, 1999), open web text (Etzioni et al., 2005; Nadeau et al., 2006), shallow syntactic knowledge (Zhang and Elhadad, 2013), etc. Word embeddings learned from unlabeled text provide representation with rich syntax and semantics and have shown to be valuable as features in unsupervised learning prob-

lems (Lin et al., 2015; He et al., 2018). In this work, we propose an NE recognition model with word embeddings as the unique feature source. We separate the entity span detection and entity type prediction into two steps. We first use Gaussian Hidden Markov Model (Gaussian-HMM) to learn the latent Markov process among NE labels with the IOB tagging scheme and then feed the candidate entity mentions to a Deep Autoencoding Gaussian Mixture Model (DAGMM) (Zong et al., 2018) for their entity types. We further apply BiLSTM and an instance selector based on reinforcement learning (Yang et al., 2018b; Feng et al., 2018) to refine annotated data. Different from existing distant supervision systems (Ren et al., 2015; Fries et al., 2017; Shang et al., 2018; Feng et al., 2018), which generate labeled data from NE lexicons or knowledge base which are still from human annotation, our model may be further enhanced by automatically labeling data with Gaussian-HMM and DAGMM.

The contribution of this paper is that we propose a fully unsupervised NE recognition model that depends on no external resources or annotation data other than word embeddings. The empirical results show that our model achieves remarkable results on CoNLL-2003 English and CoNLL-2002 Spanish benchmark datasets.

The rest of this paper is organized as follows. The next section introduces our proposed basic model in detail. Section 3 further gives a refinement model. Experimental results are reported in Section 4, followed by related work in Section 5. The last section concludes this paper.

2 Model

As shown in Figure 1, the first layer of the model is a two-class clustering layer, which initializes all the words in the sentences with 0 and 1 tags, where 0 and 1 represent non-NE and NE, respectively. The second layer is a Gaussian-HMM used to generate the boundaries of an entity mention with IOB tagging (Inside, Outside and Beginning). The representation of each candidate entity span is further fed into a Deep Autoencoding Gaussian Mixture Model (DAGMM) to identify the entity types.

2.1 Clustering

The objective of training word embeddings is to let words with similar context occupy close spatial positions. (Seok et al., 2016) conduct experiments on the nearest neighbors of NEs and discover that similar NEs are more likely to be their neighbors, since NEs are more similar in position in the corpus and syntactically and semantically related. Based on the discoveries above, we perform K-Means clustering algorithm on the word embeddings of the whole vocabulary. According to the clusters, we assign words in the cluster with fewer words 1 tags, and the other cluster 0 tags (according to the statics of (Jie et al., 2019), the proportion of NEs is very small on CoNLL datasets), and generate a coarse NE dictionary using the words with 1 tag.

2.2 Gaussian HMM

Hidden Markov model is a classic model for NE recognition (Zhou and Su, 2002; Zhao, 2004), since hidden transition matrix exists in the IOB format of the NE labels (Sarkar, 2015). We follow the Gaussian hidden Markov model introduced by (Lin et al., 2015; He et al., 2018). Given a sentence of length l, we denote the latent NE labels as $z = \{z_i\}_{i=1}^{l}$, the cluster embeddings as $v = \{v_i\}_{i=1}^{l}$, observed (pre-trained) word embeddings as $x = \{x_i\}_{i=1}^{l}$, transition parameters as θ. The joint distribution of observations and latent labels is given as following:

$$p(z, x, v; \theta) = \prod_{i=1}^{l} p(z_i|z_{i-1}; \theta)p(x_i|z_i)p(v_i|z_i)$$

(1)

where $p(z_i|z_{i-1}; \theta)$ is the multinomial transition probability, $p(x_i|z_i)$ is the multivariate emission probability, which represents the probability of a particular label generating the embedding at position i.

Cluster features (0, 1 tags) carry much word-level categorization information and can indicate the distribution representation, which we map to 3-dimension cluster embeddings $v \in \mathbb{R}^{2 \times 3}$. We initialize $v^{2 \times 3}$ as $[[1, 0, 0], [0, 0.5, 0.5]]$ (corresponding to O, I, B tag, respectively), which means that if the cluster tag of a word is 0, we initialize the word with all the probability of being O tag, otherwise it will be half of the probability of being B or I tag. $p(v_i|z_i)$ is obtained through this lookup table, and we fine-tune the cluster embeddings during the training.

.

Gaussian emissions Given a label $z \in \{B, I, O\}$, we adopt multivariate Gaussian distribution with mean μ_z and covariance matrix Σ_z as the emission probability. The conditional probability density is
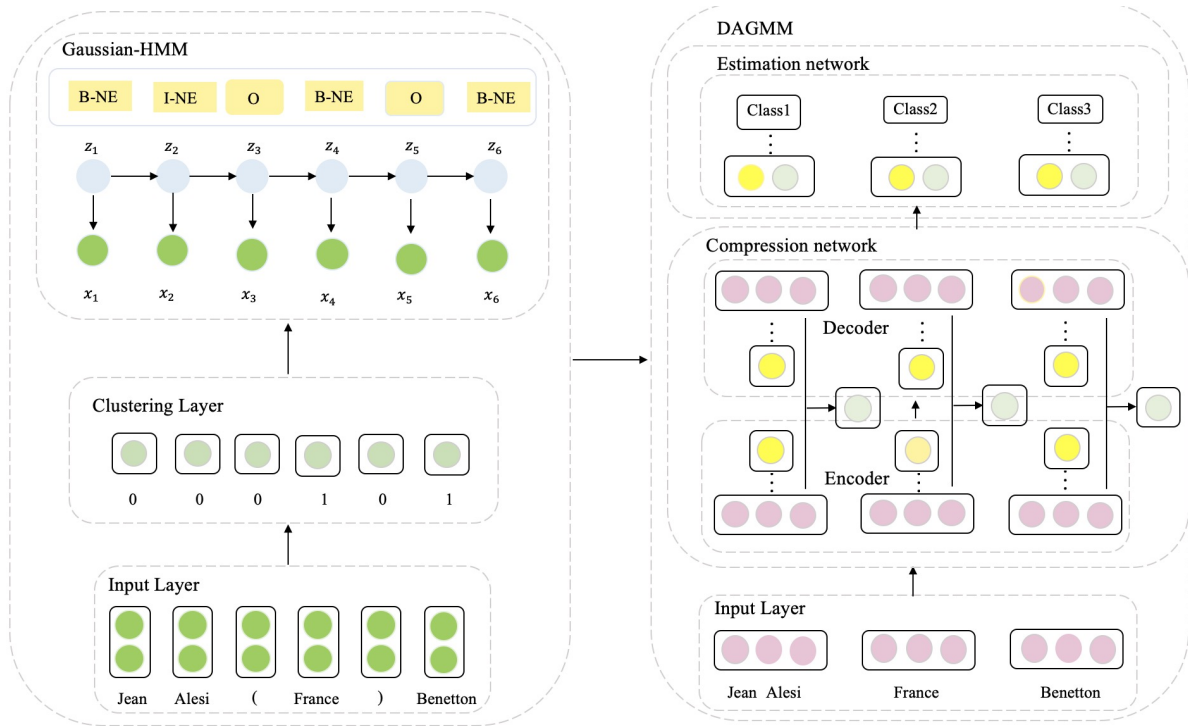
Figure 1: Architecture of the unsupervised NE recognition model. The left part is designed for entity span detection and the right part is used for entity type prediction.

in a form as:

$$p(x; \mu_z, \Sigma_z) = \frac{exp(-\frac{1}{2}(x - \mu_z)^T \Sigma_z^{-1}(x - \mu_z))}{\sqrt{(2\pi)^d |\Sigma_z|}} \quad (2)$$

where d is the dimension of the embeddings, $|\cdot|$ denotes the determinant of a matrix. The equation assumes that embeddings of words labeled as z are concentrated around the point μ_z, and the concentration is attenuated according to the covariance matrix Σ_z.

The joint distribution over a sequence of observations x, cluster sequence v and the latent label sequence z is:

$$p(z, x, v; \theta, \mu_t, \Sigma_t) = $$
$$\prod_{i=1}^{l} p(z_i|z_{i-1}; \theta) p(x; \mu_z, \Sigma_z) p(v_i|z_i) \quad (3)$$

We use forward algorithm to calculate the probability of x which we maximize during training.

We present two techniques to refine the output of Gaussian-HMM.

Single-word NEs We check the experimental results of Gaussian-HMM and discover that they perform well on the recognition of multi-word NEs, but inferiorly on single-word NEs, which incorrectly gives many false-positive labels, so we

need to do further word-level discrimination. For a single-word NE identified by the above model, if it is less than half of the probability of being marked as an NE in the corpus and does not appear in the coarse NE dictionary generated in the clustering step, then we modify it to a non-NE type. Through this modification, the precision is greatly improved without significantly reducing the recall.

High-Quality phrases Another issue of the above models is the false-negative labels, a long NE may be divided into several short NEs, in which case we need to merge them with phrase matching. We adopt a filter to determine high quality phrases according to word co-occurrence information in the corpus:

$$\frac{p(word_{last}, word_{current})}{p(word_{last}) * p(word_{current})} * n > T \quad (4)$$

where $p(\cdot)$ represents the frequency of one word appearing in the corpus, n is the total number of words and T is the threshold, which is set as the default value in word2vec[1] for training phrase embeddings. The intuition behind this is that if the ratio of the co-occurrence frequency of two adjacent words to their respective frequencies is greater than the threshold, then we consider that these two

---
[1]https://code.google.com/archive/p/word2vec

words are likely to form a phrase. Being aware of these high-quality phrases, we expect to enhance the recall of our model.

After obtaining the candidate entity span mentions, we represent them by separating words in them into two parts, the boundary and the internal (Sohrab and Miwa, 2018). The boundary part is important to capture the contexts surrounding the region, we directly take the word embedding as its representation. For the internal part, we simply average the embedding of each word to treat them equally. In summary, given the word embeddings x, we obtain the representation u of $NE(i, j)$ as follows:

$$u = NE(i, j) = [x_i; \frac{1}{j - i + 1} \sum_{k=i}^{j} x_k; x_j] \quad (5)$$

2.3 DAGMM

After obtaining the candidate entity mentions, we need to further identify their entity types. Gaussian Mixture Model (GMM) is adopted to learn the distribution of each entity type. Experimental result of (Zong et al., 2018) suggested to us that it is more efficient to perform density estimation in the low-dimensional space, in which case the distribution of words are denser and more suitable for GMM. Therefore, we adopt Deep Autoencoding Gaussian Mixture Model (DAGMM) (Zong et al., 2018) to identify NE types. DAGMM consists of two major components: compression network utilizes a deep autoencoder to perform dimension reduction and concatenate the reduced low-dimensional representation and the reconstruction error features as the representations for the estimation network; The estimation network takes the low-dimension representation as input, and uses GMM to perform density estimation.

Compression network contains an encoder function for dimension reduction and a decode function for reconstruction, both of which are multi-layer perceptron (MLP), and we use tanh function as the activation function. Given NE representation u, the compression generates its low-dimensional representation t as follows.

$$\begin{align} t_e = MLP(u; \theta_e) \quad & u' = MLP(t_e; \theta_d) \\ t_r = f(u, u') \quad & t = [t_e, t_r] \end{align} \quad (6)$$

where θ_e and θ_d are respectively the parameters of the encoder and decoder, u' is the reconstruction counterpart of u, $f(\cdot)$ denotes the reconstruction error, we take the concatenation of relative

Euclidean distance and cosine similarity as t_r in our experiment. t is then fed into the input layer of estimation network. Intuitively, we need to make the reconstruction error low to ensure that the low-dimensional representations preserve the key information of the NE representations. Thus the reconstruction error is taken as part of the loss function and is designed as the L_2-norm.

$$L(u_i, u_i') = \|u_i - u_i'\|_2^2 \quad (7)$$

Estimation network contains an MLP to predict the mixture membership for each instance and a GMM with unknown mixture-component distribution ϕ, mixture means μ and covariance matrix Σ for density prediction. During the training phase, the estimation network estimates the parameters of GMM and evaluates the likelihood for the instances. Given the low-dimensional representation t and the number of entity types K as the number of mixture components, MLP maps the representation to the K-dimension space:

$$\begin{align} m &= MLP(t; \theta_m) \\ \hat{\gamma} &= softmax(m) \end{align} \quad (8)$$

where θ_m is the parameter of MLP, $\hat{\gamma}$ is a K-dimension vector for the soft mixture-component membership prediction. The estimation network estimates the parameters of GMM as follows ($\forall 1 \leq k \leq K$),

$$\begin{align} \hat{\phi}_k &= \sum_{i=1}^{N} \frac{\hat{\gamma}_{ik}}{N}, \hat{\mu}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} t_i}{\sum_{i=1}^{N} \hat{\gamma}_{ik}} \\ \hat{\Sigma}_k &= \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik}(t_i - \hat{\mu}_k)(t_i - \hat{\mu}_k)^T}{\sum_{i=1}^{N} \hat{\gamma}_{ik}} \end{align} \quad (9)$$

where $\hat{\gamma}_i$ is the membership prediction for t_i, and $\hat{\phi}_k, \hat{\mu}_k, \hat{\sigma}_k$ are mixture probability, mean, covariance for component k in GMM, respectively.

The likelihood for the instance is inferred by

$$E(t) = -log(\sum_{k=1}^{K} \hat{\phi}_k \frac{exp(-\frac{1}{2}(t - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1}(t - \hat{\mu}_k))}{\sqrt{(2\pi)^d |\hat{\Sigma}_k|}}) \quad (10)$$

To avoid the diagonal entries in covariance matrices degenerating to 0, we penalize small values on the diagonal entries by

$$p(\hat{\Sigma}) = \sum_{k=1}^{K} \sum_{j=1}^{d} \frac{1}{\hat{\Sigma}_{kjj}} \quad (11)$$
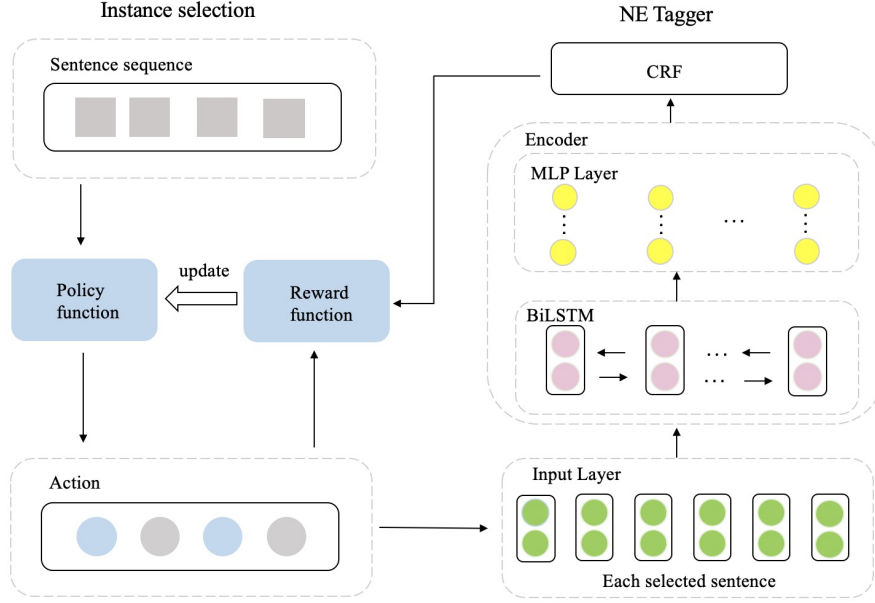
Figure 2: The framework of the reinforcement learning model, which consists of two parts. The left instance selector filters sentences according to a policy function, and then the selected sentences are used to train a better NE tagger. The instance selector updates its parameters based on the reward computed from NE tagger.

where d is the dimension of the t.

During training, we minimize the joint objective function:

$$J(\theta_e, \theta_d, \theta_m) = \frac{1}{N} \sum_{i=1}^{N} L(u_i, u_i')$$
$$+ \frac{\lambda_1}{N} \sum_{i=1}^{N} E(t_i) + \lambda_2 P(\hat{\Sigma}) \quad (12)$$

where λ_1 and λ_2 are two user-tunable parameters.

The final output is the result of K (the number of entity types) classification. We can only identify whether a word is an NE and whether several NEs are of the same category, since the entity type names as any other user-defined class/cluster/type names are just a group of pre-defined symbols by subjective naming. Therefore, following most work of unsupervised part-of-speech induction such as (Lin et al., 2015), we use matching to determine the corresponding entity category of each class, just for evaluation.

3 Refinement

The annotations obtained from the above procedure are noisy, we apply Reinforcement Learning (RL) (Feng et al., 2018; Yang et al., 2018b) to distinguish positive sentences from noisy sentences and refine these coarse-grained annotations. The RL model has two modules: an NE tagger and an instance selector.

3.1 NE tagger

Given the annotations generated by the above model, we take it as the noisy annotated label to train the NE tagger. Following (Lample et al., 2016; Yang et al., 2018a; Yang and Zhang, 2018), we employ bi-directional Long Short-Term Memory network (BiLSTM) for sequence labeling. In the input layer, we concatenate the word-level and character-level embedding as the joint word representation. We employ BiLSTM as the encoder, the concatenation of the forward and backward network output features $[\overrightarrow{h_k}, \overleftarrow{h_k}]$ is fed into an MLP, and then feed the output of MLP to a CRF layer.

CRF (Lafferty et al., 2001) has been included in most sota models, which captures label dependencies by adding transition scores between adjacent labels. During the decoding process, the Viterbi algorithm is used to search the label sequence with the highest probability. Given a sentence of length l, we denote the input sequence $x = \{x_1, ..., x_l\}$, where x_i stands for the ith word in sequence x. For $y = \{y_1, ..., y_l\}$ being a predicted sequence of labels for x. We define its score as

$$score(x, y) = \sum_{i=0}^{l-1} T_{y_i, y_{i+1}} + \sum_{i=1}^{l} P_{i, y_i} \quad (13)$$

where $T_{y_i,y_{i+1}}$ represents the transmission score from the y_i to y_{i+1}, P_{i,y_i} is the score of the ith tag of the ith word from the BiLSTM.

A softmax over all possible tag sequences in the sentences generates a probability for the sequence y:

$$p(y|x) = \frac{e^{score(x,y)}}{\sum_{\tilde{y} \in Y} e^{score(x,\tilde{y})}} \qquad (14)$$

where Y is the set of all possible tag sequences. During the training, we consider the maximum log-likelihood of the correct NE tag sequence. While decoding, we predict the optimal sequence which achieves the maximum score:

$$y^* = \arg\max_{\tilde{y} \in Y} score(x, \tilde{y}) \qquad (15)$$

3.2 Instance Selector

The instance selection is a reinforcement learning process, where the instance selector acts as the agent and interacts with the environment (sentences) and the NE tagger, as shown in Figure 2. Given all the sentences, the agent takes an action to decide which sentence to select according to a policy network at each state, and receives a reward from the NE tagger when a batch of N sentences have been selected.

State representation. We follow the work of (Yang et al., 2018b) and represent the state s_j as the concatenation of the serialized vector representations from BiLSTM and the label scores from the MLP layer.

Policy network. The agent makes an action a_j from set of $\{0, 1\}$ to indicate whether the instance selector will select the jth sentence. We adopt a logistic function as the policy function:

$$\begin{aligned} A(s_j, a_j) = &a_i\sigma(W * s_j + b) \\ &+ (1 - a_j)(1 - \sigma(W * s_j + b)) \end{aligned} \qquad (16)$$

where W and b are the model parameters, and $\sigma(\cdot)$ stands for the logistic function.

Reward. The reward function indicates the ability of the NE tagger to predict labels of the selected sentences and only generates a reward when all the actions of the selected N sentences have been completed,

$$r = \frac{1}{N}\left(\sum_{x,y \in \tilde{H}} \log p(y|x)\right) \qquad (17)$$

where $\tilde{H}$ represents the set of selected N sentences.

Training During the training phase, we optimize the policy network to maximize the reward of the selected sentences. The parameters are updated as follows,

$$\Theta = \Theta + \alpha \sum_{j=1}^{N} r\nabla_{\Theta} \log A(s_j, a_j) \qquad (18)$$

where α is the learning rate and Θ is the parameter of the instance selector.

We train the NE tagger and instance selector iteratively. In each round, the instance selector first selects sentences from the training data, and then the positive sentences are used to train the NE tagger, the tagger updates the reward to the selector to optimize the policy function. Different from the work of (Yang et al., 2018b), we relabel the negative sentences by the NE tagger after each round, and merge them with the positive sentences for the next selection.

4 Experiments

We conduct experiments [2] on two standard NER datasets: CoNLL 2003 English dataset (Tjong Kim Sang and De Meulder, 2003) and CoNLL 2002 Spanish dataset (Tjong Kim Sang, 2002) that consist of news articles. These datasets contain four entity types: LOC (location), MISC (miscellaneous), ORG (organization), and PER (person). We adopt the standard data splitting and use the micro-averaged F_1 score as the evaluation metric.

4.1 Setup

Pre-trained Word Embeddings. For the CoNLL 2003 dataset, we use the pre-trained $50D$ SENNA embeddings released by (Collobert et al., 2011) and $100D$ GloVe (Pennington et al., 2014) embeddings for clustering and training, respectively. For CoNLL 2002 Spanish dataset, we train $64D$ GloVe embeddings with the minimum frequency of occurrence as 5, and the window size of 5.

Parameters and Model Training. For DAGMM, the hidden dimensions for compression network and estimation network are $[75, 15]$ and 10, respectively. For NE Tagger, we follow the work of (Yang and Zhang, 2018) and use the default experimental settings. We conduct optimization with the stochastic gradient descent, the learning rate is initially set to 0.015 and will shrunk by 5% after each epoch. The number of selected sentences at each time is set

---

[2]Code is available at: https://github.com/cslydia/uNER.

		EN			SP		
		Pre	Rec	F_1	Pre	Rec	F_1
(Lample et al., 2016)	LSTM-CRF	91.0	90.8	90.9	85.7	85.8	85.8
(Jie et al., 2019)	IA-Training[0]	89.0	90.1	89.5	81.3	82.7	82.0
(Liu et al., 2019)	Dict [1]	-	-	72	-	-	-
	Dict $+P(\tau\|y)$[2]	-	-	76	-	-	-
(Shang et al., 2018)	Dict-Training[3]	75.18	79.71	77.38	22.11	70.89	33.71
	Handcraft	23.45	26.38	24.83	-	-	-
	SENNA	7.09	7.0	7.036	-	-	-
Ours	basic[4]	62.57	56.83	60.76	45.35	53.41	49.05
	LSTM-CRF	73.15	60.02	65.94	49.99	56.76	53.16
	LSTM-CRF + RL[5]	74.25	63.51	68.64	50.61	58.36	54.31

Table 1: Main results of NE recognition on CoNLL 2003 English (EN) and CoNLL 2002 Spanish (SP) datasets. Superscript annotations: 0: represents incomplete annotations in training data. 1: type-specific entity vocabularies extracted from WikiText-2. 2: a prior type information which was pre-computed from entity popularity information. 3: these three represent the lexicon extracted from training data, human annotated lexicon from Wikipedia corpus and SENNA lexicon. 4: Our basic ouput from GMM without refinement. 5: +RL: add reinforcement learning with instance selector.

as 10. Dropout (Srivastava et al., 2014) of a ratio 0.5 is applied for embeddings and hidden states.

4.2 Compared Methods

Supervised benchmarks on each dataset are represented to show the gap between supervised and our unsupervised model without any annotation data or external resources. LSTM-CRF (Lample et al., 2016) is the state-of-the-art supervised NE recognition model.

(Jie et al., 2019) propose an approach to tackle the incomplete annotation problem. This work introduces q distribution to model missing labels instead of traditionally uniform distribution for all possible complete label sequences, and uses k-fold cross-validation for estimating q. They report the result of keeping 50% of all the training data and removing the annotations of the rest entities together with the O labels for non-NEs.

(Liu et al., 2019) proposes a Knowledge-Augmented Language Model (KALM), which recognizes NEs during training language models. Given type-specific entity vocabularies and the general vocabulary, KALM computes the entity probability of the next word according to its context. This work extracts 11,123 vocabularies from WikiText-2 as the knowledge base. WikiText-2 is a standard language modeling dataset and covers 92.80% of

entities in CoNLL 2003 dataset.

Category	SENNA	Handcraft
Location	36,697	213,318
Miscellaneous	4,722	-
Organization	6,440	11,749
Person	123,283	80,050
Total	171,142	305,117

Table 2: Number of entries for each category in lexicons for (Shang et al., 2018) for comparisons with our model, which need no lexicon.

(Shang et al., 2018) propose a distant supervision NE recognition model AutoNER using domain-specific dictionaries. This work designs a *Tie or Break* tagging scheme that focuses on the ties between adjacent tokens. Accordingly, AutoNER is designed to distinguish Break from Tie while skipping unknown positions. The authors report their evaluation results on datasets from a specific domain and their method needs necessary support from an NE lexicon. For better comparisons, we use the lexicon from the training data, the SENNA lexicon presented by (Collobert et al., 2011) and our handcraft lexicon [3] as the domain-specific dictionary to re-implement their work on CoNLL-2003 English dataset, the size of each category

---

[3]This dictionary is mainly based on Wikipedia corpus.

	LOC	MISC	ORG	PRR	overall
basic	0.75	0.67	0.64	0.83	0.72
$P(\tau|y)$	0.81	0.67	0.65	0.88	0.76
Ours	0.68	0.45	0.60	0.83	0.69

Table 3: Comparisons with (Liu et al., 2019) on CoNLL-2003 for each entity type.

	EN			SP		
	Pre	Rec	F_1	Pre	Rec	F_1
Cluster	0.43	0.53	0.47	0.27	0.69	0.39
HMM	0.80	0.72	0.76	0.54	0.70	0.63

Table 4: Main results for entity span detection. Cluster is the result before sending to Gaussian-HMM, HMM is short for Gaussian-HMM.

in each lexicon is shown in Table 2. Due to the resource constraints, we only extract the lexicon in training data without labeling a larger dictionary for wider comparisons for CoNLL-2002.

4.3 Results and Comparisons

We present F_1, precision, and recall scores on both datasets in Table 1. All the models compared in Table 1 besides ours need extra resources to some extent, like partially annotated training data, NE dictionary, etc. While our model achieves comparable results without using any resources mentioned above. We compare the prediction results for each entity type with (Liu et al., 2019) in Table 3, and it is shown that our model performs well in LOC, ORG and PER types. These NEs have specific meanings, and more similar in position and length in the corpus, thus their word embeddings can better capture semantic and syntactic regularities, and thus better represent the words, while MISC includes various entity types which may bring significant confusion on learning type patterns. While (Liu et al., 2019) better regularize the type information from NE dictionaries and re-trained type information.

Though (Shang et al., 2018) achieves better results when using golden NE dictionary for English, they perform poorly on SENNA and our manually annotated dictionary. Specially, when using the gold NE dictionary for training Spanish dataset, the result is especially unsatisfactory. According to our statistics, over half of the MISC NEs in CoNLL 2002 Spanish training data are labeled as other types in the same dataset, while the ratio

is 28% in CoNLL 2003 English dataset, thus the results differs a lot in the two datasets. Our models achieve much better performance than those of (Shang et al., 2018) by more than doubling their F_1 scores in the general NE dictionary (SENNA and human-labeled Wikipedia dictionary). Besides, our unsupervised NE recognition method is shown more general and gives a more stable performance than the distant supervision model in (Shang et al., 2018), which highly relies on the quality of the support dictionary and the domain relevance of the dictionary to the corpus.

We acknowledge that there still exists a gap between our unsupervised NE recognition model with the sota supervised model (Lample et al., 2016; Jie et al., 2019), but the applicability of unsupervised models and the robustness of resource dependence are unreachable by supervised models.

Table 4 lists the results of entity span detection. Our Gaussian-HMM absorbs informative clue from clustering, and greatly improves the results of entity span detection. For the English dataset, we apply SENNA embedding, which is trained on English Wikipedia and Reuters RCV1 corpus, thus the result of clustering becomes better, leading to a better result of Gaussian-HMM. While for the Spanish dataset, the embedding is trained on Wikipedia corpus only, which has little connection with the CoNLL-2002 datasets, thus the result is slightly lower. Overall, unsupervised modeling based on word embeddings may be more general and robust than dictionary-based and corpus-based modeling.

4.4 Discussion

Our model is good at dealing with common NEs, because their word embeddings well represent meanings, thus leading to a better prediction. However, our model is not very satisfactory in dealing with nested NEs. For example, *South Africa* and *Africa* can be taken as NEs respectively, and *south* is recognized as *O* labels in most of the other cases, thus in this case, our model makes a bias prediction, and only recognizes *Africa*. Table 5 shows an example of a positive instance and a negative instance before RL and after RL. During the training process, the instance selector takes action to select the first instance for training a silver NE Tagger. Then the second instance is relabeled after one epoch, and merged with the first instance for the next turn. We can discover that the NE Tagger learns the effective features of the ORG type, and can modify

Intance 1										
Instance	Newcombe	was	quoted	as	saying	in	Sydney	's	Daily	Telegraph
Before RL	B-PER	O	O	O	O	O	B-LOC	O	B-ORG	I-ORG
After RL	B-PER	O	O	O	O	O	B-LOC	O	B-ORG	I-ORG
golden label	B-PER	O	O	O	O	O	B-LOC	O	B-ORG	I-ORG

Instance 2									
Instance	Thursday	's	overseas	edition	of	the	People	's	Daily
Before RL	O	O	O	O	O	O	O	O	O
After RL	O	O	O	O	O	O	B-ORG	I-ORG	I-ORG
golden label	O	O	O	O	O	O	B-ORG	I-ORG	I-ORG

Table 5: Example of of two instances before and after Reinforcement Learning (RL).

the wrong labels in the second instance.

Using Pre-trained Languages Models. We have also tried language models such as ELMo and BERT as encoders for Gaussian-HMM, but their sparse characteristics in high-dimensional space are not conducive to Gaussian modeling. Unsupervised models have fewer parameters and simpler training phase, thus there is no guarantee that the language model will retain its key properties when it is reduced to low dimensions. We further add the pre-trained language model BERT as the additional embeddings for the NE Tagger to refine the output of Gaussian-HMM and DAGMM, which slightly improves our result to 69.99 for CoNLL-2003 English NER and 56.66 for CoNLL-2002 Spanish NER.

5 Related work

Deep neural network models have helped peoples released from handcrafted features in a wide range of NLP tasks (Zhang et al., 2019; Li et al., 2018a,b, 2019; Zhou and Zhao, 2019; Xiao et al., 2019; Zhang et al., 2020a,b,c). LSTM-CRF (Lample et al., 2016; Ma and Hovy, 2016) is the most state-of-the-art model for NE recognition. In order to reduce the requirements of training corpus, distant supervised models (Shang et al., 2018; Yang et al., 2018b; Ren et al., 2015; He, 2017; Fries et al., 2017) have been applied to NE recognition. Recently, (Liu et al., 2019) proposed a Knowledge-Augmented Language Model, which trains language models and at the same time compute the probability of the next word being different entity types according to the context given type-specific entity/general vocabularies. Unlike these existing approaches, our study focuses on unsupervised NE recognition learning without any extra resources.

Noisy data is another important factor affecting the neural network models, reinforcement learning has been applied to many tasks, (Feng et al., 2018) use reinforcement learning for Relation Classification from Noisy Data. (Yang et al., 2018b) show how to apply reinforcement learning in NE recognition systems by using instance selectors, which can tell high-quality training sentences from noisy data. Their work inspires us to use reinforcement leaning after obtaining coarse annotated data from Gaussian-HMM.

6 Conclusion

This paper presents an NE recognition model with only pre-trained word embeddings and achieves remarkable results on CoNLL 2003 English and CoNLL 2002 Spanish benchmark datasets. The proposed approach yields, to the best of our knowledge, first fully unsupervised NE recognition work on these two benchmark datasets without any annotation data or extra knowledge base.

References

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*.

Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement learning for relation classification from noisy data. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Jason Fries, Sen Wu, Alex Ratner, and Christopher Ré. 2017. Swellshark: A generative model for biomedical named entity recognition without labeled data. *arXiv preprint arXiv:1704.06360*.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2018. Unsupervised learning of syntactic structure with invertible neural projections. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1292–1302.

Wenqi He. 2017. Autoentity: automated entity detection from massive text corpora.

Zhanming Jie, Pengjun Xie, Wei Lu, Ruixue Ding, and Linlin Li. 2019. Better modeling of incomplete annotations for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 729–734.

John D. Lafferty, Andrew Mccallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270.

Zuchao Li, Jiaxun Cai, Shexia He, and Hai Zhao. 2018a. Seq2seq dependency parsing. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3203–3214, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Zuchao Li, Shexia He, Jiaxun Cai, Zhuosheng Zhang, Hai Zhao, Gongshen Liu, Linlin Li, and Luo Si. 2018b. A unified syntax-aware framework for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2401–2411, Brussels, Belgium. Association for Computational Linguistics.

Zuchao Li, Shexia He, Hai Zhao, Yiqing Zhang, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2019. Dependency or span, end-to-end uniform semantic role labeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6730–6737.

Chu-Cheng Lin, Waleed Ammar, Chris Dyer, and Lori Levin. 2015. Unsupervised pos induction with word embeddings. *Computer Science*.

Angli Liu, Jingfei Du, and Veselin Stoyanov. 2019. Knowledge-augmented language model and its application to unsupervised named-entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1142–1150.

Liyuan Liu, Jingbo Shang, Xiang Ren, Frank Fangzheng Xu, Huan Gui, Jian Peng, and Jiawei Han. 2018. Empower sequence labeling with task-aware neural language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Ying Luo, Fengshun Xiao, and Hai Zhao. 2020. Hierarchical contextualized representation for named entity recognition. In *the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2020)*.

Ying Luo and Hai Zhao. 2020. Bipartite flat-graph network for nested named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6408–6418.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074.

David Nadeau, Peter D Turney, and Stan Matwin. 2006. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *Conference of the Canadian society for computational studies of intelligence*, pages 266–277. Springer.

John K Pate and Mark Johnson. 2016. Grammar induction from (lots of) words alone. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 23–32, Osaka, Japan. The COLING 2016 Organizing Committee.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Xiang Ren, Ahmed El-Kishky, Chi Wang, Fangbo Tao, Clare R Voss, and Jiawei Han. 2015. Clustype: Effective entity recognition and typing by relation phrase-based clustering. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 995–1004.

Kamal Sarkar. 2015. A hidden markov model based system for entity extraction from social media english text at fire 2015. *arXiv preprint arXiv:1512.03950*.

Miran Seok, Hye-Jeong Song, Chan-Young Park, Jong-Dae Kim, and Yu-seop Kim. 2016. Named entity recognition using word embedding as a feature. *International Journal of Software Engineering and Its Applications*, 10(2):93–104.

Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. 2018. Learning named entity tagger using domain-specific dictionary. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2054–2064.

Yanyao Shen, Hyokun Yun, Zachary C Lipton, Yakov Kronrod, and Animashree Anandkumar. 2017. Deep active learning for named entity recognition. In *International Conference on Learning Representations*.

Mohammad Golam Sohrab and Makoto Miwa. 2018. Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*.

Karl Stratos, Michael Collins, and Daniel Hsu. 2016. Unsupervised part-of-speech tagging with anchor hidden markov models. *Transactions of the Association for Computational Linguistics*, 4:245–257.

Partha Pratim Talukdar and Fernando Pereira. 2010. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1473–1481.

Erik F. Tjong Kim Sang. 2002. 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)*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. 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*, pages 142–147.

Fengshun Xiao, Jiangtong Li, Hai Zhao, Rui Wang, and Kehai Chen. 2019. Lattice-based transformer encoder for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3090–3097.

Jie Yang, Shuailong Liang, and Yue Zhang. 2018a. Design challenges and misconceptions in neural sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3879–3889.

Jie Yang and Yue Zhang. 2018. NCRF++: An open-source neural sequence labeling toolkit. In *Proceedings of ACL 2018, System Demonstrations*, pages 74–79.

Yaosheng Yang, Wenliang Chen, Zhenghua Li, Zhengqiu He, and Min Zhang. 2018b. Distantly supervised NER with partial annotation learning and reinforcement learning. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2159–2169, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Shaodian Zhang and Noémie Elhadad. 2013. Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. *Journal of biomedical informatics*.

Shuailiang Zhang, Hai Zhao, Yuwei Wu, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2020a. DCMN+: Dual co-matching network for multi-choice reading comprehension. In *the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2020)*.

Zhuosheng Zhang, Yafang Huang, and Hai Zhao. 2019. Open vocabulary learning for neural Chinese Pinyin IME. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1584–1594.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020b. Semantics-aware BERT for language understanding. In *the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2020)*.

Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, and Hai Zhao. 2020c. SG-Net: Syntax-guided machine reading comprehension. In *the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2020)*.

Shaojun Zhao. 2004. Named entity recognition in biomedical texts using an HMM model. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pages 87–90, Geneva, Switzerland. COLING.

GuoDong Zhou and Jian Su. 2002. Named entity recognition using an HMM-based chunk tagger. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 473–480.

Junru Zhou and Hai Zhao. 2019. Head-Driven Phrase Structure Grammar parsing on Penn Treebank. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2396–2408.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. 2018. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*.