Simple and Effective Few-Shot Named Entity Recognition with Structured Nearest Neighbor Learning

June 4, 2021

Content

- NER involves **indentification** of *proper names* in texts, and **classification** into a set of predefined categories of interest.
- Three universally accepted categories: Person, location, organisation
- Other common tasks: recognition of date/time expressions, measures (percent, money, weight etc), email adress etc
- Other domain-specific entities: names of Drugs, Genes, medical conditions, name of ships, bibliographic references etc

Introduction

What is NER?

When Sebastian Thrun PERSON started working on self-driving cars at Google ORG in 2007 DATE , few people outside of the company took him seriously. “I can tell you very senior CEOs of major American car companies would shake my hand and turn away because I wasn’t worth talking to,” said Thrun ORG , now the co-founder and CEO of online higher education startup Udacity, in an interview with Recode earlier this week DATE .

A little less than a decade later DATE , dozens of self-driving startups have cropped up while automakers around the world clamor, wallet in hand, to secure their place in the fast-moving world of fully automated transportation.

- **Few-shot learning (FSL)**, also referred to as **low-shot learning (LSL)** in few sources, is a type of machine learning problems where the training dataset contains limited information.
- As the dimension of input data is a factor that determines resource costs (e.g. time costs, computational costs etc.), companies can reduce data analysis/machine learning (ML) costs by using few-shot learning.
- In the context of NER, these fewshot classification methods can enable rapid building of NER systems for a new domain by labeling only a few examples per entity class.

- Firstly, NER is essentially a **structured learning problem**. It is crucial to model label dependencies instead of directly classifying each token independently using the existing few-shot classification approaches.
- Secondly, few-shot classification models typically learn to represent each semantic class by a prototype based on the labeled examples in its support set. However, for NER, unlike the entity classes, the Outside (O) class does not represent any unified semantic meaning.

Introduction

What's new in this paper?

- Propose a simple, yet effective method **STRUCTSHOT** for few-shot NER
- STRUCTSHOT uses a **nearest neighbor (NN) classifier** and a **Viterbi decoder** for prediction.
- Propose a more standard and reproducible evaluation setup for few-shot NER by using standard test sets and development sets from benchmark datasets of several domains

Problem Statement and Setup

Few-shot NER

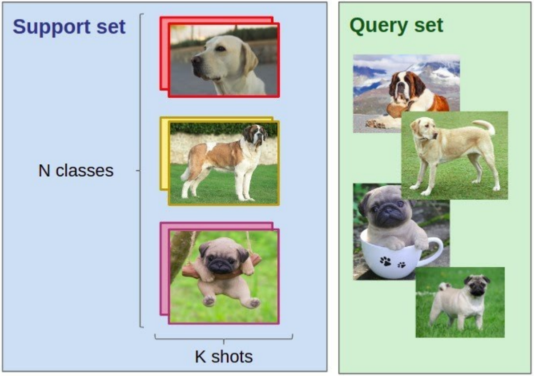
- Few-shot NER focuses on a specific NER setting where a system is trained on a annotations of one or more source domain $\{\mathcal{D}_{\mathcal{S}}^{(i)}\}$ and then tested on one or more target domains $\{\mathcal{D}_{\mathcal{T}}^{(i)}\}$ by only providing a few labeled examples per entity class.
- The target tag set $\mathcal{C}_{\mathcal{T}}^{(i)}$ can be different from any source tag set $\mathcal{C}_{\mathcal{S}}^{(i)}$.
- **K-shot NER:** Given an input sequence $x = \{x_t\}_{t=1}^{T}$ and K-shot support set for the target tag set $\mathcal{C}_{\mathcal{T}}$. Find the best tag sequence sequence $y = \{y_t\}_1^T$.
- The K-shot support set contains K entity examples for each entity class given by $\mathcal{C}_{\mathcal{T}}$.

Problem Statement and Setup

Few-shot NER

Support:	Former	prime	minister	Peres	to	Morocco	today
	O	O	O	PER	O	LOC	O
Target:	President	plans	to	visit	Japan	in	May
	O	O	O	O	LOC	O	O

Few-Shot NER



Few-shot in Computer Vision

Problem Statement and Setup

A standard evaluation setup

- Previous studies evaluated few-shots classification based on episode. An episode includes a sampled K-shot support set of labeled examples and a few sampled K-shot test sets.
- Authors propose a more realistic evaluation setting by sampling only the support sets and testing the model on the standard test sets from NER benchmarks.
- Using **Greedy sampling** for sampling support set.

Problem Statement and Setup

A standard evaluation setup

Algorithm 1: Greedy sampling

Require: # of shot K, labeled set $\mathbf{X}$ with tag set $\mathcal{C}$

Sort classes in $\mathcal{C}$ based on their freq. in $\mathbf{X}$
$S \leftarrow \phi$ //Initialize the support set
$\{\text{Count}_i \leftarrow 0\}$ //Initialize counts of entity classes in S
while $i < |\mathcal{C}|$ **do**
    while $\text{Count}_i < K$ **do**
        Sample $(\mathbf{x}, \mathbf{y}) \in \mathbf{X}$ s.t. $\mathcal{C}_i \in \mathbf{y}$, w/o replacement
        $S \leftarrow S \cup \{(\mathbf{x}, \mathbf{y})\}$
        Update $\{\text{Count}_j\}\ \forall \mathcal{C}_j \in \mathbf{y}$
    end while
end while
return S

Model

Nearest neighbor classification for few-shot NER (NNShot)

- NNShot simply computes a similarity score between a token x in the test example and all tokens $\{c'\}$ in the support set. It assigns the token x a tag c corresponding to the most similar token in the support set:

$$y^* = \underset{c \in \{1,\ldots,C\}}{\operatorname{argmin}} \; d_c(\hat{x})$$

$$d_c(\hat{x}) = \min_{x' \in \mathcal{S}_\mathcal{C}} d(\hat{x}, \hat{x}') = ||\hat{x} - \hat{x}'||_2^2$$

- Perform L2-normalization on the features before computing these distances.
- Pre-trained NER models as token embedders: BiLSTM and BERT.

Model

Structured nearest neighbor learning (STRUCTSHOT)

- STRUCTSHOT only makes use of its Viterbi decoder during inference.

	O	I	I-Other
O	0.7	0.3	/
I	0.3	0.5	0.2

Abstract transition probabilities

	O	I-PER	I-LOC	I-ORG
O	0.7	0.1	0.1	0.1
I-PER	0.3	0.5	0.1	0.1
I-LOC	0.3	0.1	0.5	0.1
I-ORG	0.3	0.1	0.1	0.5

Target transition probabilities

Model

Structured nearest neighbor learning (STRUCTSHOT)

- Estimates the abstract **transition probabilities**:

$$p(Y|X) = \frac{N(X \to Y)}{N(. \to Y)}$$

- **Emission probabilities** $p(y = c|x)$ for each token in the test example from NNShot:

$$p(y = c|x) = \frac{e^{-d_c(\hat{x})}}{\sum_{c'} e^{-d_{c'}(\hat{x})}}$$

- Finally:

$$\hat{y}^* = \underset{y}{\operatorname{argmax}} \prod_{t=1}^{T} \overbrace{p\left(y_t \mid x\right)}^{\text{emission}} \overbrace{p\left(y_t \mid y_{t-1}\right)}^{\text{transition}}$$

2 situations:

- Tag set extension
- Domain transfer

Dataset	Domain	# Class	# Sent	# Entity
OntoNotes	General	18	76,714	104,151
CoNLL'03	News	4	20,744	35,089
I2B2'14	Medical	23	140,817	29,233
WNUT'17	Social	6	5,690	3,890

Experiments

Result: one-shot NER

System	Tag Set Extension				Domain Transfer			
	Group A	Group B	Group C	Ave.	CoNLL	I2B2	WNUT	Ave.
BiLSTM-based systems								
Prototypical Network	4.0±1.6	5.4±1.9	5.2±1.5	4.9	18.7±9.2	2.2±1.0	5.5±2.7	8.8
NNShot (ours)	15.7±7.1	25.1±7.1	22.7±7.1	21.2	46.4±11.7	7.5±2.9	6.9±3.2	20.3
StructShot (ours)	18.9±9.4	31.9±5.1	22.0±3.4	24.3	53.1±9.9	10.5±2.6	10.4±4.4	24.7
BERT-based systems								
SimBERT	8.3±1.4	9.0±3.8	8.4±1.8	8.6	15.7±3.7	7.7±0.8	4.9±1.2	9.4
Prototypical Network	18.7±4.7	24.4±8.9	18.3±6.9	20.5	53.0±7.2	7.6±3.5	14.8±4.9	25.1
PrototypicalNet+P&D	18.5±4.4	24.8±9.3	20.7±8.4	21.3	56.0±7.3	7.9±3.2	18.8±5.3	27.6
NNShot (ours)	27.2±3.5	**32.5**±14.4	**23.8**±10.2	27.8	61.3±11.5	16.6±2.1	21.7±6.3	33.2
StructShot (ours)	**27.5**±4.1	32.4±14.7	**23.8**±10.2	**27.9**	**62.3**±11.4	**22.1**±3.0	**25.3**±5.3	**36.6**

Experiments

Result: five-shot NER

System	Tag Set Extension				Domain Transfer			
	Group A	Group B	Group C	Ave.	CoNLL	I2B2	WNUT	Ave.
BiLSTM-based systems								
Prototypical Network	7.4±2.7	21.8±7.6	18.2±5.6	15.8	47.6±9.0	5.9±1.1	8.8±3.3	20.8
NNShot (ours)	24.5±5.4	35.2±7.4	33.8±6.3	31.2	62.0±6.1	8.4±2.7	12.4±4.2	27.6
StructShot (ours)	26.1±6.0	46.1±6.5	38.0±1.8	36.7	63.8±6.9	13.7±0.8	15.1±4.9	30.9
BERT-based systems								
SimBERT	10.1±0.8	23.0±6.7	18.0±3.5	17.0	28.6±2.5	9.1±0.7	7.7±2.2	15.1
Prototypical Network	27.1±2.4	38.0±5.9	38.4±3.3	34.5	65.9±1.6	10.3±0.4	19.8±5.0	32.0
PrototypicalNet+P&D	29.8±2.8	41.0±6.5	38.5±3.3	36.4	67.1±1.6	10.1±0.9	23.8±3.9	33.6
NNShot (ours)	44.7±2.3	53.9±7.8	53.0±2.3	50.5	74.3±2.4	23.7±1.3	23.9±5.0	40.7
StructShot (ours)	**47.4**±3.2	**57.1**±8.6	**54.2**±2.5	**52.9**	**75.2**±2.3	**31.8**±1.8	**27.2**±6.7	**44.7**

Thanks for listening!