Instance-Based Learning of Span Representations: A Case Study through Named Entity Recognition

Hiroki Ouchi[1,2] **Jun Suzuki**[2,1] **Sosuke Kobayashi**[2,3]
Sho Yokoi[2,1] **Tatsuki Kuribayashi**[2,4] **Ryuto Konno**[2] **Kentaro Inui**[2,1]
[1] RIKEN [2] Tohoku University [3] Preferred Networks, Inc. [4] Langsmith, Inc.
`hiroki.ouchi@riken.jp`
`{jun.suzuki,sosk,yokoi,kuribayashi,ryuto,inui}@ecei.tohoku.ac.jp`

Summary

- Our method builds **a feature space** where spans with the same class label are close to each other. At inference time, each span is assigned a class label based on its neighbor spans in the feature space.
  - This is the first work to investigate **instance-based learning of span representations**.
  - Through empirical analysis on NER, we demonstrate our instance-based method enables to build models that have high interpretability without sacrificing performance.

NER as span classification

Formally, given an input sentence of T words $X = (w_1, w_2, \ldots, w_T)$, we first enumerate possible spans $\mathcal{S}(X)$, and then assign a class label $y \in \mathcal{Y}$ to each span $s \in \mathcal{S}(X)$. We will write each span as $s = (a, b)$, where a and b are word indices in the sentence: $1 \le a \le b \le T$. Consider the following sentence.

$$\text{Franz}_1 \quad \text{Kafka}_2 \quad \text{is}_3 \quad \text{a}_4 \quad \text{novelist}_5$$
$$[\quad \texttt{PER} \quad]$$

$$\mathcal{S}(X) = \{(1,1), (1,2), (1,3), \ldots, (4,5), (5,5)\}$$

$$s = (1,2) \qquad y = \texttt{PER}$$

The probability that each span s is assigned a class label y is modeled by using softmax function:

$$\mathrm{P}(y|s) = \frac{\exp(\mathrm{score}(s, y))}{\sum_{y' \in \mathcal{Y}} \exp(\mathrm{score}(s, y'))} \ .$$

Typically, as the scoring function, the inner product between each label weight vector $\mathbf{w}_y$ and span feature vector $\mathbf{h}_s$ is used:

$$\mathrm{score}(s, y) = \mathbf{w}_y \cdot \mathbf{h}_s \ .$$

The score for the NULL label is set to a constant, $\mathrm{score}(s, y = \texttt{NULL}) = 0$, similar to logistic regression (He et al., 2018). For training, the loss function we minimize is the negative log-likelihood:

$$\mathcal{L} = - \sum_{(X,Y) \in \mathcal{D}} \sum_{(s,y) \in \mathcal{S}(X,Y)} \log \mathrm{P}(y|s) \ ,$$

Encoder and span representation

the encoder architecture proposed by Ma and Hovy (2016), which encodes each token of the input sentence $w_t \in X$ with word embedding and character-level CNN. The encoded token representations $\mathbf{w}_{1:T} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_T)$ are fed to bidirectional LSTM for computing contextual ones $\overrightarrow{\mathbf{h}}_{1:T}$ and $\overleftarrow{\mathbf{h}}_{1:T}$. From them, we create $\mathbf{h}_s^{\text{lstm}}$ for each span $s = (a, b)$ based on LSTM-minus (Wang and Chang, 2016). For flat NER, we use the representation $\mathbf{h}_s^{\text{lstm}} = [\overrightarrow{\mathbf{h}}_b - \overrightarrow{\mathbf{h}}_{a-1}, \overleftarrow{\mathbf{h}}_a - \overleftarrow{\mathbf{h}}_{b+1}]$. For nested NER, we use $\mathbf{h}_s^{\text{lstm}} = [\overrightarrow{\mathbf{h}}_b - \overrightarrow{\mathbf{h}}_{a-1}, \overleftarrow{\mathbf{h}}_a - \overleftarrow{\mathbf{h}}_{b+1}, \overrightarrow{\mathbf{h}}_a + \overrightarrow{\mathbf{h}}_b, \overleftarrow{\mathbf{h}}_a + \overleftarrow{\mathbf{h}}_b]$.[7] We then multiply $\mathbf{h}_s^{\text{lstm}}$ with a weight matrix $\mathbf{W}$ and obtain the span representation: $\mathbf{h}_s = \mathbf{W}\, \mathbf{h}_s^{\text{lstm}}$. For the scoring function in Equation 1 in the instance-based span model, we use the inner product between a pair of span representations: $\text{score}(s_i, s_j) = \mathbf{h}_{s_i} \cdot \mathbf{h}_{s_j}$.

Instance-based span model

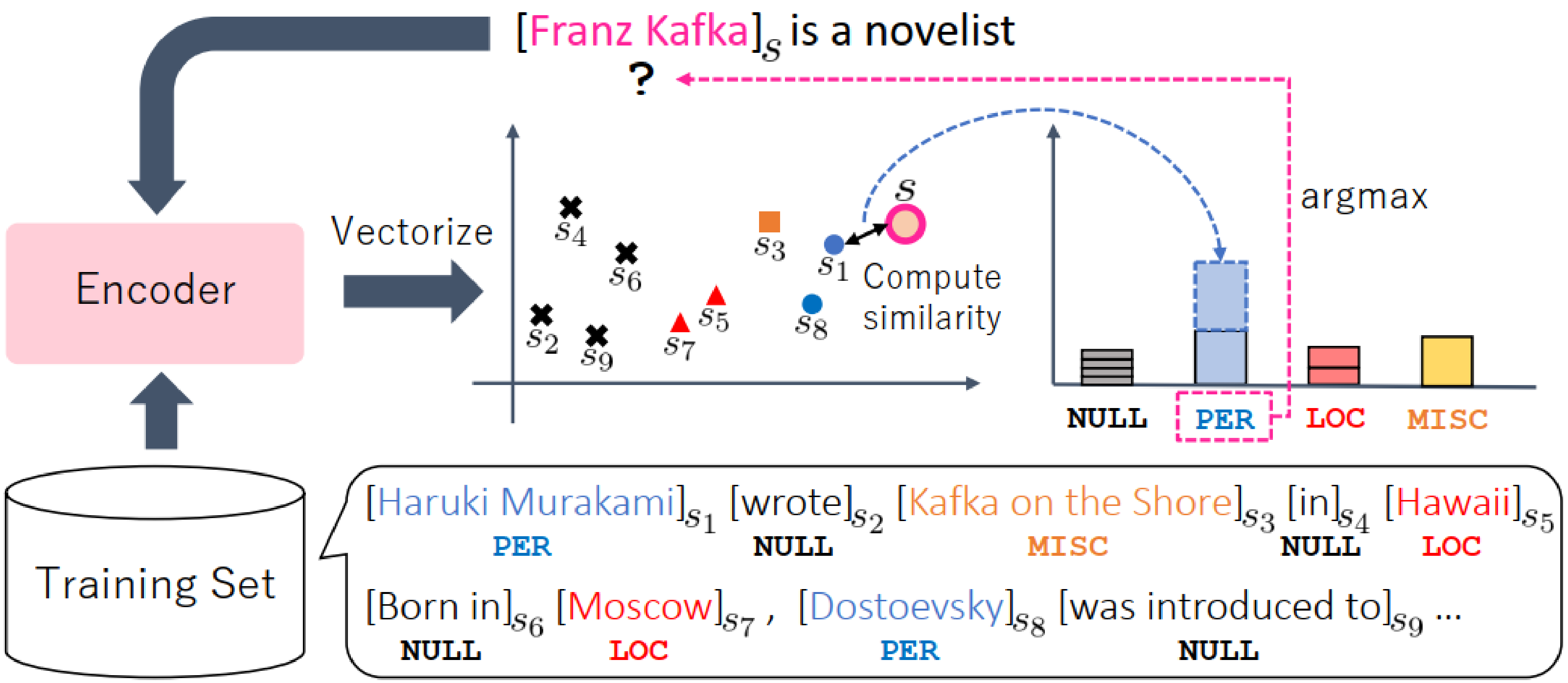


Figure 1: Illustration of our instance-based span model. An entity candidate "Franz Kafka" is used as a query and vectorized by an encoder. In the vector space, similarities between all pairs of the candidate (s) and the training instances ($s_1, s_2, \ldots, s_9$) are computed, respectively. Based on the similarities, the label probability (distribution) is computed, and the label with the highest probability PER is assigned to "Franz Kafka."

Instance-based span model

Formally, within the neighbourhood component analysis framework (Goldberger et al., 2005), we define the *neighbor span probability* that each span $s_i \in \mathcal{S}(X)$ will select another span s_j as its neighbor from candidate spans in the training set:

$$\mathrm{P}(s_j|s_i, \mathcal{D}') = \frac{\exp(\mathrm{score}(s_i, s_j))}{\sum_{s_k \in \mathcal{S}(D')} \exp(\mathrm{score}(s_i, s_k))} \ . \quad (1)$$

Here, we exclude the input sentence X and its ground-truth labels Y from the training set $\mathcal{D}$: $\mathcal{D}' = \mathcal{D} \setminus \{(X, Y)\}$, and regard all other spans as candidates: $\mathcal{S}(\mathcal{D}') = \{s \in \mathcal{S}(X') | (X', Y') \in \mathcal{D}'\}$. The scoring function returns a similarity between the spans s_i and s_j. Then we compute the probability that a span s_i will be assigned a label y_i:

$$\mathrm{P}(y_i|s_i) = \sum_{s_j \in \mathcal{S}(\mathcal{D}', y_i)} \mathrm{P}(s_j|s_i, \mathcal{D}') \ . \qquad (2)$$

Here, $\mathcal{S}(\mathcal{D}', y_i) = \{s_j \in \mathcal{D}' | \ y_i = y_j\}$, so the equation indicates that we sum up the probabilities of the neighbor spans that have the same label as the span s_i. The loss function we minimize is the negative log-likelihood:

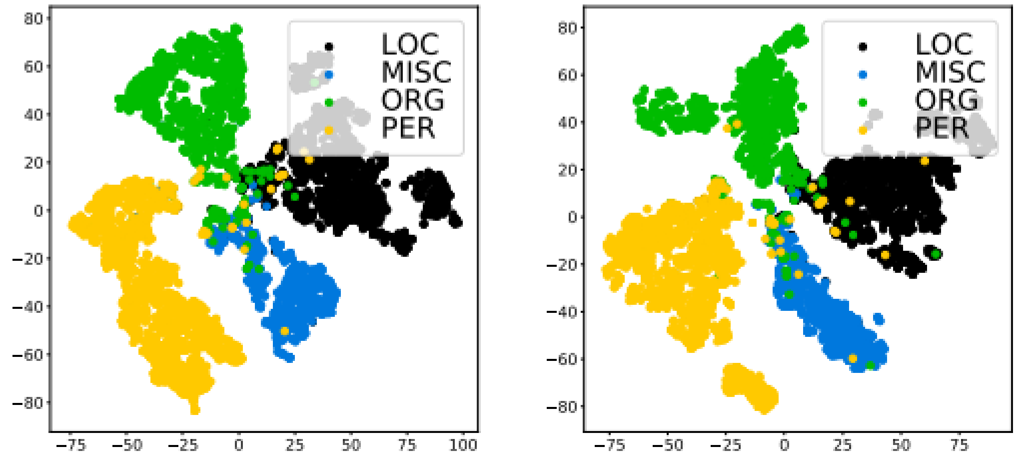
$$\mathcal{L} = - \sum_{(X,Y) \in \mathcal{D}} \ \sum_{(s_i, y_i) \in \mathcal{S}(X,Y)} \log \mathrm{P}(y_i|s_i) \ ,$$

where $\mathcal{S}(X, Y)$ is a set of pairs of a span s_i and its ground-truth label y_i. At inference time, we predict $\hat{y}_i$ to be the class label with maximal marginal probability:

$$\hat{y}_i = \arg\max_{y \in \mathcal{Y}} \mathrm{P}(y|s_i) \ ,$$

where the probability $\mathrm{P}(y|s_i)$ is computed for each of the label set $y \in \mathcal{Y}$.

A.2 Feature space visualization



(a) Classifier-based (b) Instance-based

Figure 3: Visualization of entity span features computed by classifier-based and instance-based models.

Experiment

- Datasets:
    - GENIA (Kim et al.,2003)(DNA RNA Protein Cell-line Cell_type)
    - CoNLL-2003 dataset (LOC PER ORG MISC)
- Baseline:
    - a classifier-based span model

	Classifier-based	Instance-based
	GloVe	
Flat NER	90.68 ±0.25	90.73 ±0.07
Nested NER	73.76 ±0.35	74.20 ±0.16
	BERT	
Flat NER	90.48 ±0.18	90.48 ±0.07
Nested NER	73.27 ±0.19	73.92 ±0.20

Table 1: Comparison between classifier-based and instance-based span models. Cells show the F_1 scores and standard deviations on each test set.

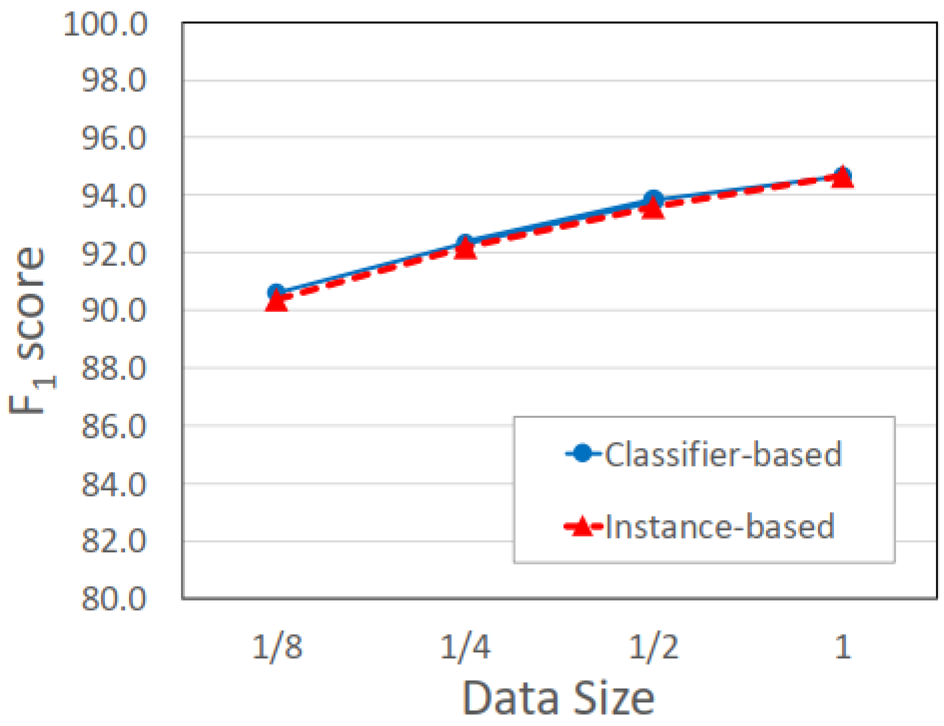


Figure 2: Performance on the CoNLL-2003 development set for different amounts of the training set.

QUERY		... [Tom Moody] took six for 82 but ...
Classifier-based		
1	PER	... [Billy Mayfair] and Paul Goydos and ...
2	NULL	... [Billy Mayfair and Paul Goydos] and ...
3	NULL	... [Billy Mayfair and Paul Goydos and] ...
4	NULL	... [Billy] Mayfair and Paul Goydos and ...
5	NULL	... [Ducati rider Troy Corser] , last year ...
Instance-based		
1	PER	[Ian Botham] began his test career ...
2	PER	... [Billy Mayfair] and Paul Goydos and ...
3	PER	... [Mark Hutton] scattered four hits ...
4	PER	... [Steve Stricker] , who had a 68 , and ...
3	PER	... [Darren Gough] polishing off ...

Table 2: Example of span retrieval. An entity candidate "Tom Moody" in the CoNLL-2003 development set used as a query for retrieving five nearest neighbors from the training set.

QUERY		... spokesman for [Air France] 's ... Pred: LOC Gold: ORG
1	LOC	... [Colombia] turned down American 's ...
2	LOC	... involving [Scotland] , Wales , ...
3	LOC	... signed in [Nigeria] 's capital Abuja ...
4	LOC	... in the West Bank and [Gaza] .
5	LOC	... on its way to [Romania] ...

Table 3: Example of an error by the instance-based span model. Although the gold label is ORG (Organization), the wrong label LOC (Location) is assigned.

4.2 Quantitative analysis

We report averaged F_1 scores across five different runs of the model training with random seeds.

Overall F_1 scores We investigate whether or not our instance-based span model can achieve competitive performance with the classifier-based span model. Table 1 shows F_1 scores on each test set.[10] Consistently, the instance-based span model yielded comparable results to the classifier-based span model. This indicates that our instance-based learning method enables to build NER models without sacrificing performance.

Effects of training data size Figure 2 shows F_1 scores on the CoNLL-2003 development set by the models trained on full-size, $1/2$, $1/4$ and $1/8$ of the training set. We found that (i) performance of both models gradually degrades when the size of the training set is smaller and (ii) both models yield very competitive performance curves.

4.3 Qualitative analysis

To better understand model behavior, we analyze the instance-based model using GloVe in detail.

Examples of retrieved spans The span feature space learned by our method can be applied to various downstream tasks. In particular, it can be used as a span retrieval system. Table 2 shows five nearest neighbor spans of an entity candidate "Tom Moody." In the classifier-based span model, person-related but non-entity spans were retrieved. By contrast, in the instance-based span model, person (PER) entities were consistently retrieved.[11] This tendency was observed in many other cases, and we confirmed that our method can build preferable feature spaces for applications.

Errors analysis The instance-based span model tends to wrongly label spans that includes location or organization names. For example, in Table 3, the wrong label LOC (Location) is assigned to "Air France" whose gold label is ORG (Organization).