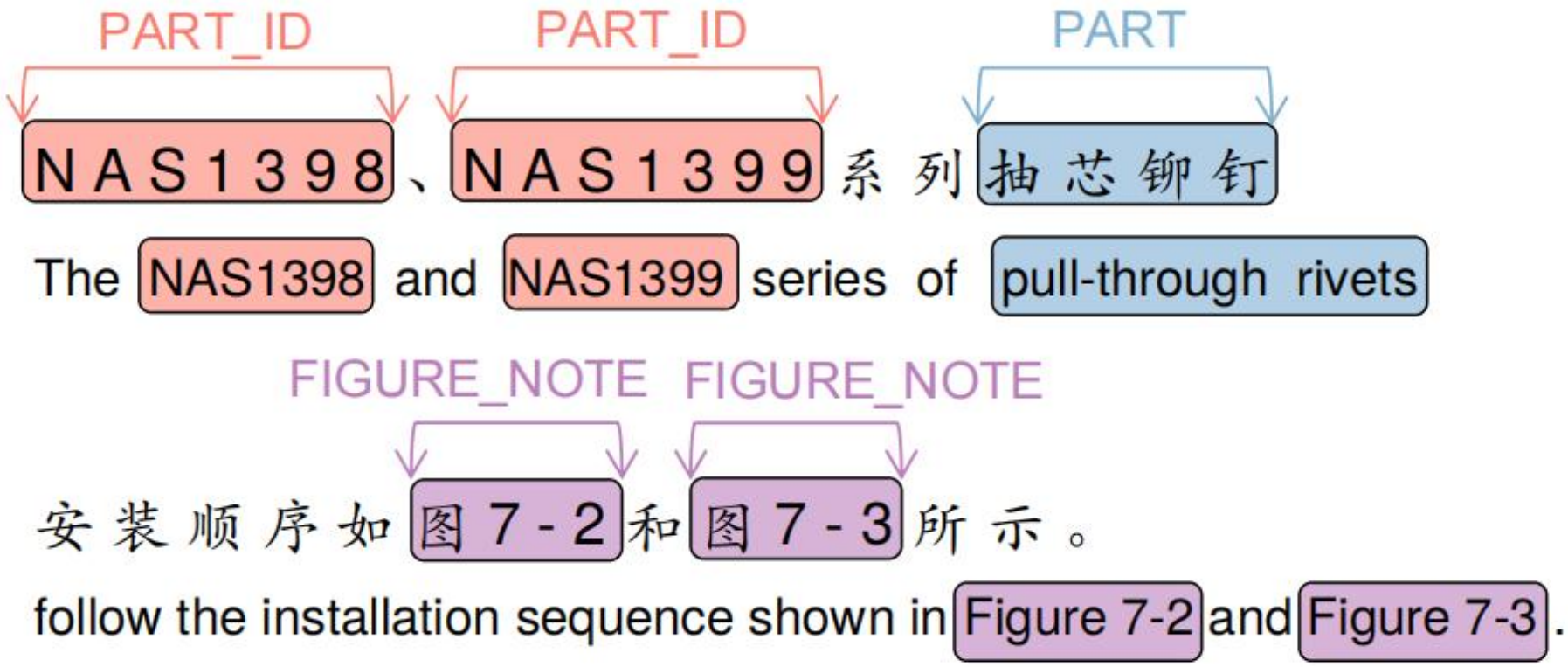
A Corpus and Method for Chinese Named Entity Recognition in Manufacturing

Ruiting Li[1,2], Peiyan Wang[1,2]◈ ,
Libang Wang[1,2], Danqingxin Yang[1,2], Dongfeng Cai[1,2]
1. School of Computer Science, Shenyang Aerospace University, Shenyang, China
2. Liaoning Professional Technology Innovation Center on Knowledge Engineering and Human-Computer Interaction, Shenyang, China
liruiting@stu.sau.edu.cn, wangpy@sau.edu.cn
{wanglibang, yangdanqingxin}@stu.sau.edu.cn, caidf@vip.163.com

Ruiting Li, at LREC-COLING 2024

Introduction

NER in the manufacturing specifications aims to locate and classify manufacturing specific named entities such as PART_ID, PART and FIGURE_NOTE.



Motivation

We pay special attention to Chinese NER in the manufacturing specifications.

Motivation

We pay special attention to Chinese NER in the manufacturing specifications.

- There is no publicly available corpus for Chinese NER in the manufacturing specifications.
  - Demands of strong background in domain-specific knowledge.
  - High cost of manual annotation.

Motivation

We pay special attention to Chinese NER in the manufacturing specifications.

- There is no publicly available corpus for Chinese NER in the manufacturing specifications.
  - Demands of strong background in domain-specific knowledge.
  - High cost of manual annotation.

- Conventional neural methods perform poorly in manufacturing.
  - Conventional neural methods rely on a large amount of training data.
  - Sufficient labeled data is unavailable in manufacturing.

What We Have Done

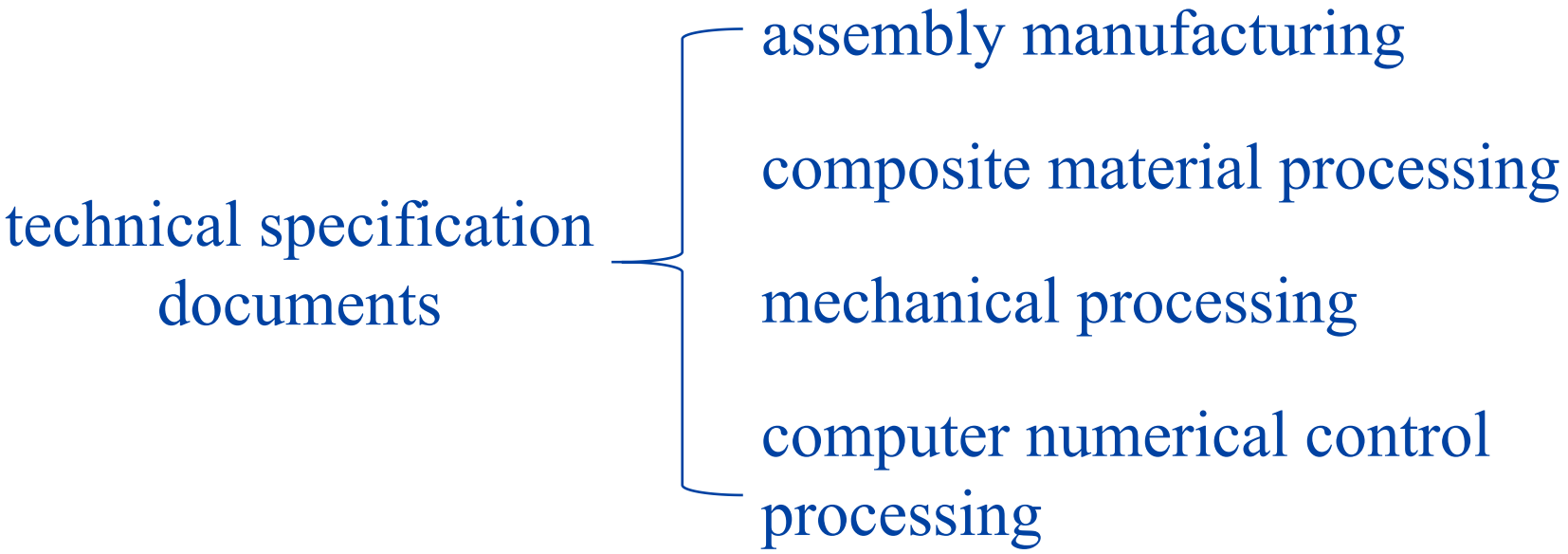
Our key contributions are the following:

- Annotation guidelines for NER corpus in manufacturing, including annotation instructions and definitions for 16 categories such as PART, PART_ID, FILE, and FIGURE_NOTE.

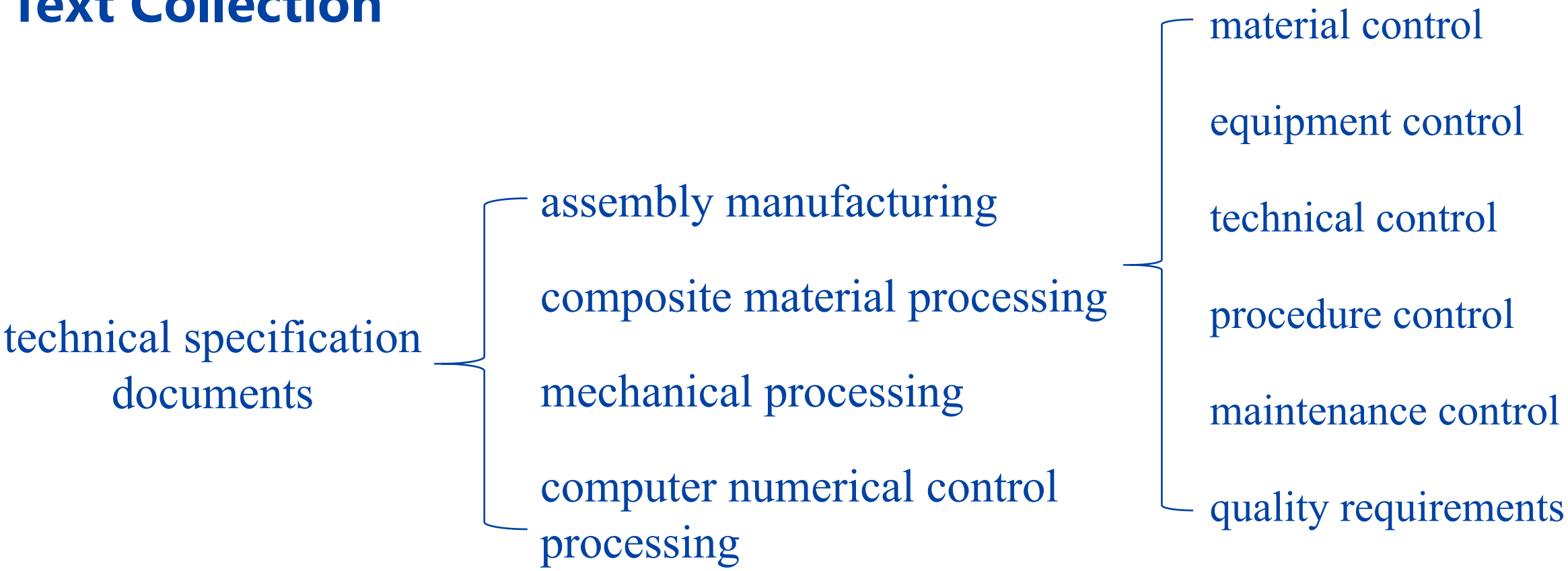
- A Chinese NER corpus named MS-NERC with 4,424 sentences and 16,383 entities.

- An entity recognizer named Trainable States Transducer for modeling morphological patterns of named entities.

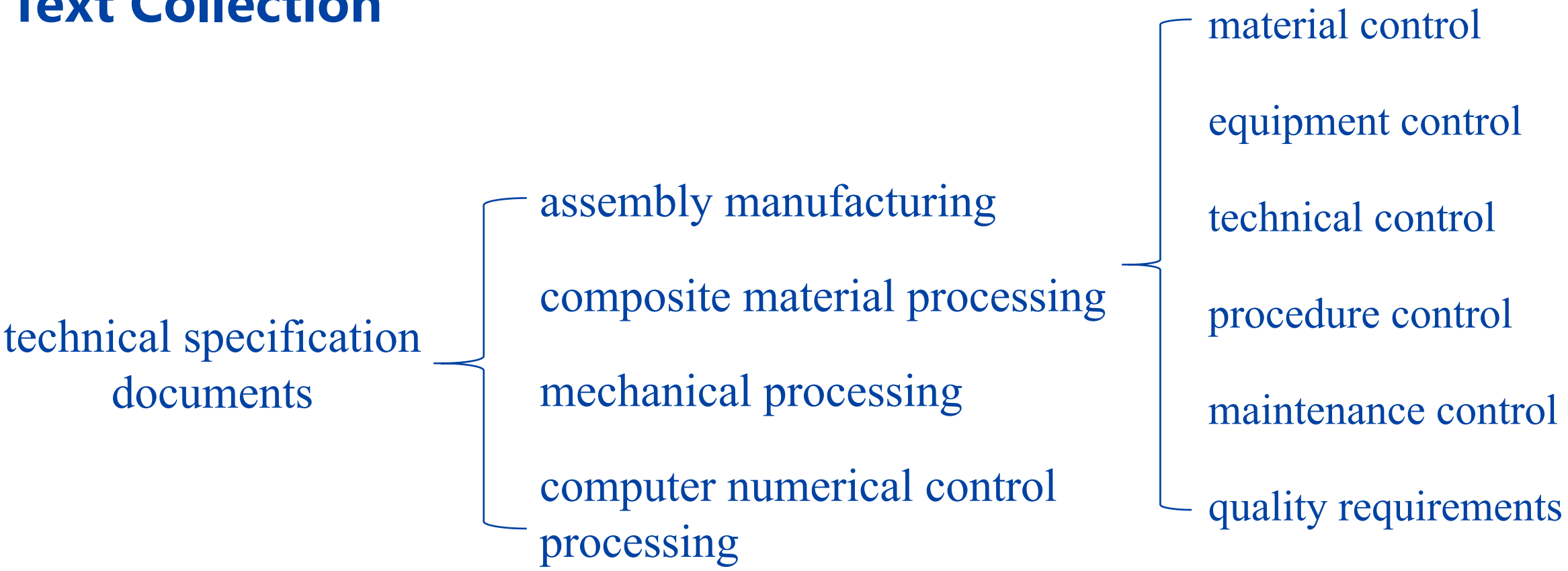
Text Collection



Text Collection



Text Collection



- Manufacturing parameters and manufacturing conditions. ✔
- Information related to specific products and enterprises. ✘
- A raw dataset of 4,424 sentences.

Annotation Guidelines

16 categories refer to the Fundamental Terminology of Mechanical Manufacturing (Hongyu et al.,2008).

Annotation Guidelines

16 categories refer to the Fundamental Terminology of Mechanical Manufacturing (Hongyu et al.,2008).

- PART_ID marks an identification number assigned to the part.
  (e.g., NAS1398).

- PART marks a basic component unit in manufacturing, including the combination of parts.
  (e.g., 抽芯铆钉 / pull-through rivet)

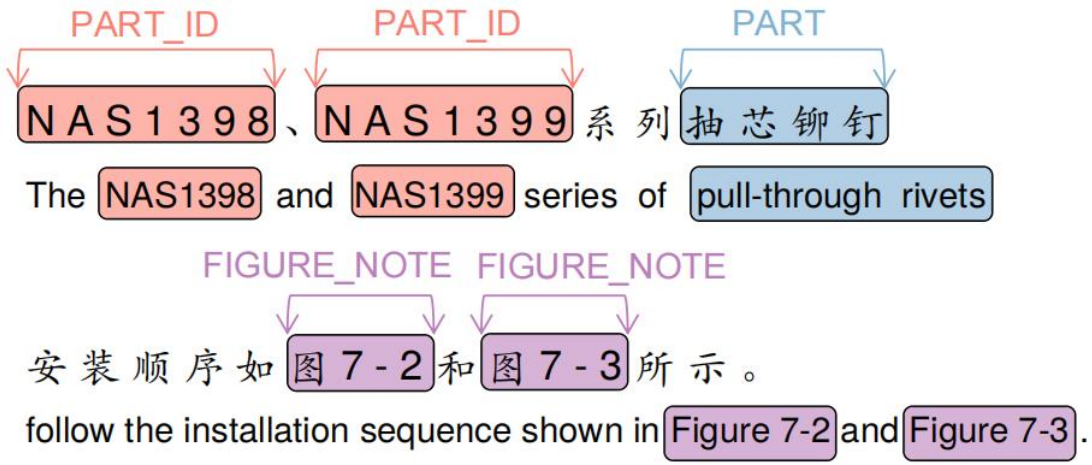


Figure 1: Examples of the sentence with named entities and translations in the manufacturing specifications.

Annotation Guidelines

Three specific instructions:

- The entities must be specific rather than generalized .
  (e.g., ’铆钉’ /rivet instead of ’零件’/part)

- The entities should not be accompanied by conjunctions and punctuation marks indicating juxtaposition, except in the case of notes in parentheses.
  (e.g., ’1.02mm(0.040in.)’)

- The entities are annotated based on their maximum span without nesting.

Annotation Process

Three experts for annotating:

- Pre-annotation:
  - Iterative discussions to refine the guidelines.

Annotation Process

Three experts for annotating:

- Pre-annotation:
  - Iterative discussions to refine the guidelines.

- Formal annotation:
  - 4,424 technical specification sentences are divided into three groups.
  - Each group is assigned to a different annotator, with a 15% overlap for duplicate assessment.

Annotation Process

Three experts for annotating:

- Pre-annotation:
  - Iterative discussions to refine the guidelines.

- Formal annotation:
  - 4,424 technical specification sentences are divided into three groups.
  - Each group is assigned to a different annotator, with a 15% overlap for duplicate assessment.

- Formal annotation commences once a Cohen's Kappa (Cohen, 1960) score exceeding 0.6 is achieved. Our inter-annotator agreement is 0.68.

MS-NERC Statistics

The corpus comprises a total of 4,424 sentences and 16,383 entities.

Category	Number	Max	Min	Mean
ACCESSORY	1,861	12	1	3.31
ACCESSORY_ID	876	17	3	7.87
ATTRIBUTE	2,479	19	1	3.54
ATTRIBUTE_VA	1,357	50	1	10.85
FIGURE_NOTE	396	6	3	4.21
FILE	556	12	3	6.82
HOLE	418	12	1	2.14
MATERIAL	481	13	1	3.27
OPERATION	1,933	9	1	2.63
PART	2,407	13	1	3.27
PART_AR	1,396	14	1	3.85
PART_ID	202	15	2	8.75
PART_NU	131	4	1	1.89
REDUNDANT	164	6	1	2.27
TABLE_NOTE	296	6	3	4.05
TOOL	1,430	18	1	3.65

Table 1: Entity statistics in *MS-NERC*.

MS-NERC Statistics

The corpus comprises a total of 4,424 sentences and 16,383 entities.

Category	Number	Max	Min	Mean
ACCESSORY	1,861	12	1	3.31
ACCESSORY_ID	876	17	3	7.87
ATTRIBUTE	2,479	19	1	3.54
ATTRIBUTE_VA	1,357	50	1	10.85
FIGURE_NOTE	396	6	3	4.21
FILE	556	12	3	6.82
HOLE	418	12	1	2.14
MATERIAL	481	13	1	3.27
OPERATION	1,933	9	1	2.63
PART	2,407	13	1	3.27
PART_AR	1,396	14	1	3.85
PART_ID	202	15	2	8.75
PART_NU	131	4	1	1.89
REDUNDANT	164	6	1	2.27
TABLE_NOTE	296	6	3	4.05
TOOL	1,430	18	1	3.65

Table 1: Entity statistics in *MS-NERC*.

Category	Example	REs
ACCESSORY_ID	RQI5275	RQI(\d){4}
ATTRIBUTE_VA	-28.6m	-(\d+).(\d+)m
FIGURE_NOTE	图6-20	图(\d+)-(\d+)
FILE	RQX3001	RQX(\d){4}
PART_ID	NAS1252	NAS(\d+)
TABLE_NOTE	表7-1	表(\d+)-(\d+)

Table 2: Examples of entities and regular expressions.

Regular Expressions and Finite State Transducer

PART_ID PART_ID FIGURE_NOTE FIGURE_NOTE

NAS1398、NAS1399系列抽芯铆钉安装顺序如图7-2和图7-3所示。

The NAS1398 and NAS1399 series of pull-through rivets follow the installation sequence shown in Figure 7-2 and Figure 7-3.

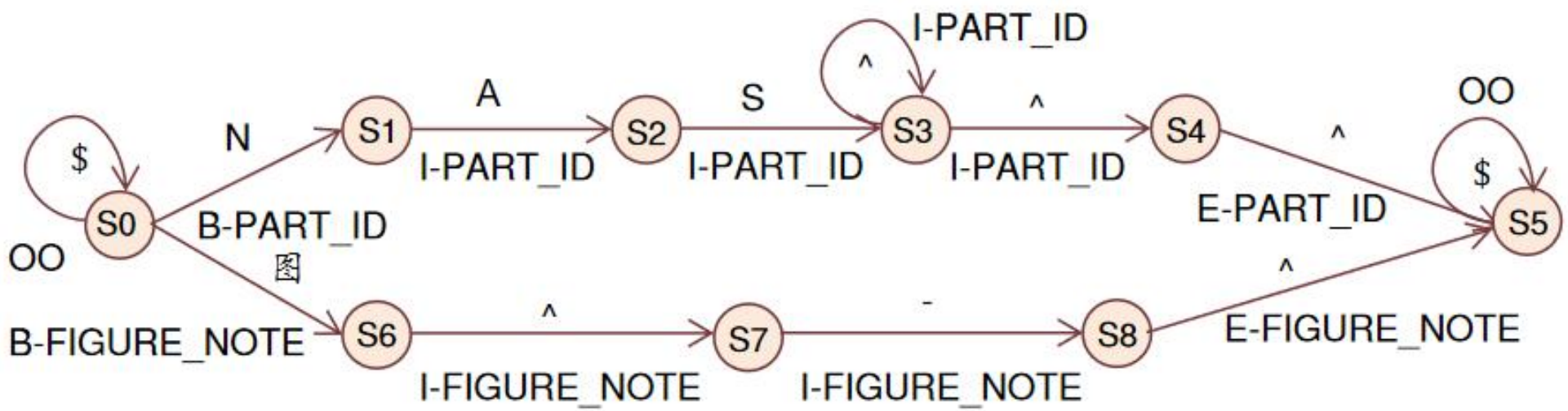
Category	Example	REs
ACCESSORY_ID	RQI5275	RQI(\d){4}
ATTRIBUTE_VA	-28.6m	-(\d+).(\d+)m
FIGURE_NOTE	图6-20	图(\d+)-(\d+)
FILE	RQX3001	RQX(\d){4}
PART_ID	NAS1252	NAS(\d+)
TABLE_NOTE	表7-1	表(\d+)-(\d+)

Regular Expressions and Finite State Transducer

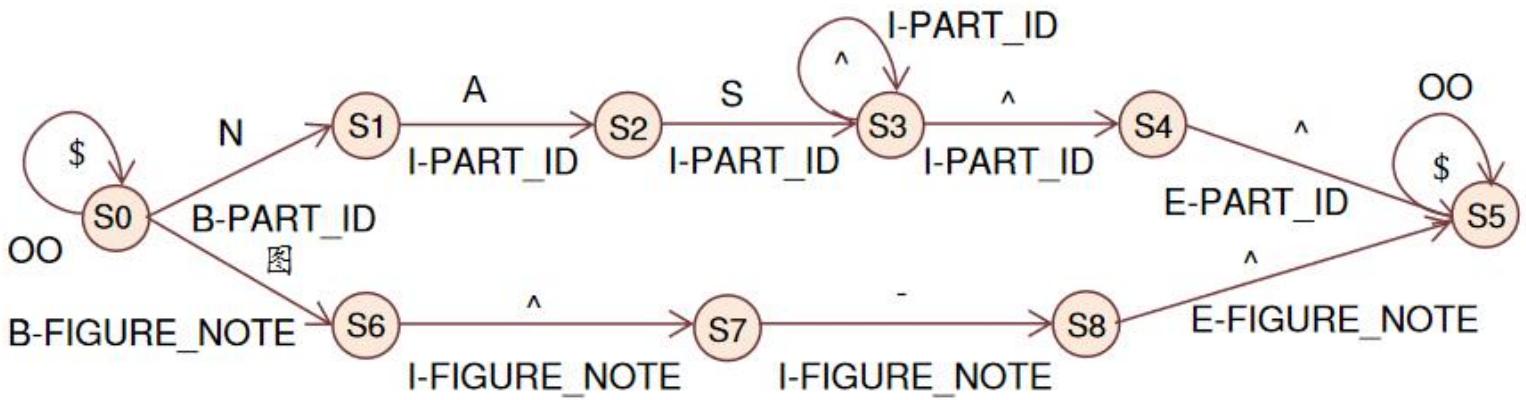
PART_ID PART_ID FIGURE_NOTE FIGURE_NOTE

NAS1398、NAS1399系列抽芯铆钉安装顺序如图7-2和图7-3所示。

The NAS1398 and NAS1399 series of pull-through rivets follow the installation sequence shown in Figure 7-2 and Figure 7-3.



Parameter Tensors for TST



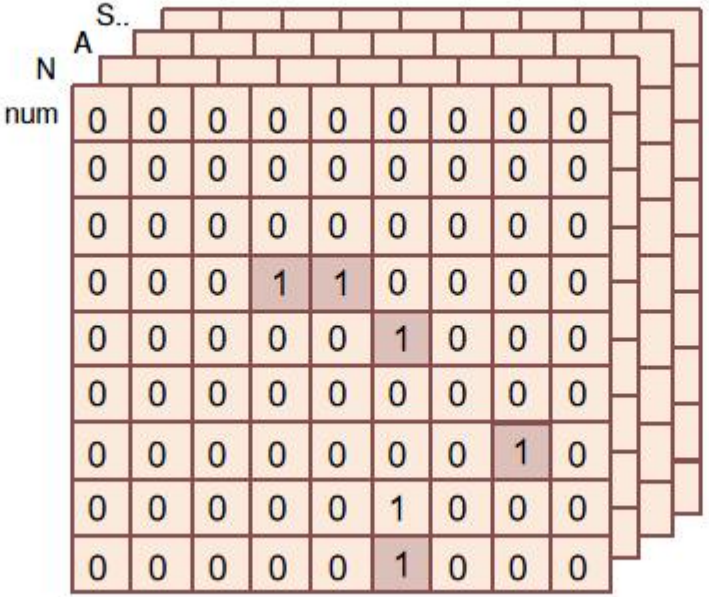
Parameter of TST

1	0	0	0	0	0	0	0	0

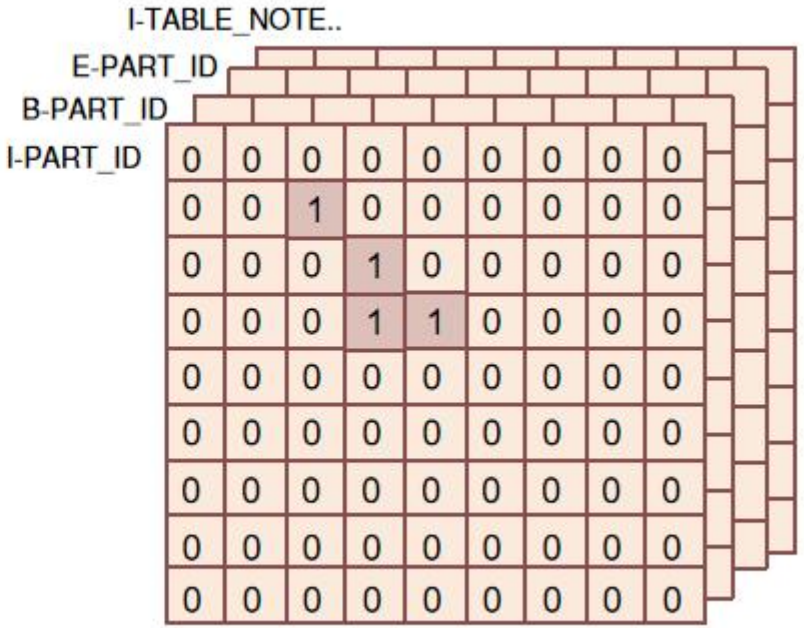
s

0	0	0	0	0	1	0	0	0

m



w_i



w_o

TST Inference

Given a sentence $x = (x_1, x_2, ..., x_n)$, and TST.

Algorithm 1 Inference in *TST*

Input: $x = (x_1, x_2, ..., x_n)$,
$\mathcal{N} = \langle \mathcal{S}, \mathcal{I}, \mathcal{O}, W_i, W_o, s, m \rangle$.
Output: the label scores f_t of x_t.
Step 1: Let $\odot$ denote hadamard product, $\otimes$ denote outer product.
Let $\alpha_0 = s^{\mathsf{T}}, \beta_n = m^{\mathsf{T}}$.
Step 2: calculate forward score α_t
for $t = 1 \rightarrow n$ **do**
  $\alpha_t = W_i[x_t] \cdot \alpha_{t-1}$
end
Step 3: calculate backward score β_t
for $t = n \rightarrow 1$ **do**
  $\beta_t = W_i^{\mathsf{T}}[x_t] \cdot \beta_{t+1}$
end
Step 4: calculate bidirectional score bi_scores
for $t = 1 \rightarrow n$ **do**
  $bi\_scores = \alpha_t \otimes \beta_t$
end
$F_t = \sum_{l_j \in \mathcal{O}} (bi\_scores \odot W_i[x_t]) \odot W_o[l_j]$
$f_t = \sum_{k=1}^{|\mathcal{S}|} F_t[:, k]$
return f_t

TST Inference

Given a sentence $x = (x_1, x_2, ..., x_n)$, and TST.

Algorithm 1 Inference in *TST*

Input: $x = (x_1, x_2, ..., x_n)$,
$\mathcal{N} = \langle \mathcal{S}, \mathcal{I}, \mathcal{O}, W_i, W_o, s, m \rangle$.
Output: the label scores f_t of x_t.
Step 1: Let $\odot$ denote hadamard product, $\otimes$ denote outer product.
Let $\alpha_0 = s^{\mathsf{T}}, \beta_n = m^{\mathsf{T}}$.
Step 2: calculate forward score α_t
for $t = 1 \rightarrow n$ **do**
  $\alpha_t = W_i[x_t] \cdot \alpha_{t-1}$
end
Step 3: calculate backward score β_t
for $t = n \rightarrow 1$ **do**
  $\beta_t = W_i^{\mathsf{T}}[x_t] \cdot \beta_{t+1}$
end
Step 4: calculate bidirectional score bi_scores
for $t = 1 \rightarrow n$ **do**
  $bi\_scores = \alpha_t \otimes \beta_t$
end
$F_t = \sum_{l_j \in \mathcal{O}} (bi\_scores \odot W_i[x_t]) \odot W_o[l_j]$
$f_t = \sum_{k=1}^{|\mathcal{S}|} F_t[:, k]$
return f_t

$$P_t = priority\left(W_p\left(f_t + f_t'\right) + b_p\right)$$

TST Inference

Given a sentence $x = (x_1, x_2, ..., x_n)$, and TST.

Algorithm 1 Inference in *TST*

Input: $x = (x_1, x_2, ..., x_n)$,
$\mathcal{N} = \langle \mathcal{S}, \mathcal{I}, \mathcal{O}, W_i, W_o, s, m \rangle$.
Output: the label scores f_t of x_t.
Step 1: Let $\odot$ denote hadamard product, $\otimes$ denote outer product.
Let $\alpha_0 = s^{\mathsf{T}}, \beta_n = m^{\mathsf{T}}$.
Step 2: calculate forward score α_t
for $t = 1 \rightarrow n$ **do**
  $\alpha_t = W_i[x_t] \cdot \alpha_{t-1}$
end
Step 3: calculate backward score β_t
for $t = n \rightarrow 1$ **do**
  $\beta_t = W_i^{\mathsf{T}}[x_t] \cdot \beta_{t+1}$
end
Step 4: calculate bidirectional score bi_scores
for $t = 1 \rightarrow n$ **do**
  $bi\_scores = \alpha_t \otimes \beta_t$
end
$F_t = \sum_{l_j \in \mathcal{O}} (bi\_scores \odot W_i[x_t]) \odot W_o[l_j]$
$f_t = \sum_{k=1}^{|\mathcal{S}|} F_t[:, k]$
return f_t

$$P_t = priority\left(W_p\left(f_t + f_t'\right) + b_p\right)$$

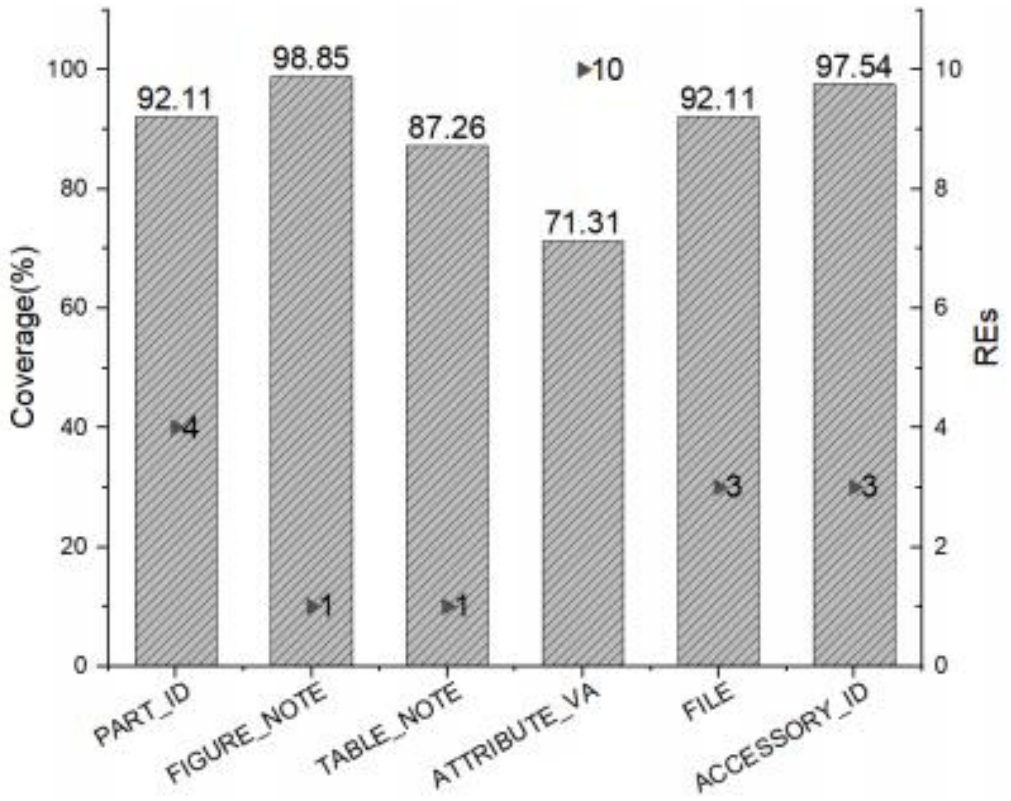
$$P_t' = \left(\max\left(p_1', p_k'\right), p_2', p_3', \ldots, p_{k-1}'\right)$$

$$l_t = \underset{1 \le j \le k-1}{argmax}\, P_t'(j)$$

Dataset and Regular Expressions

- MS-NERC: 70% sentences as the training set, 10% as the development set, and 20% as the testing set.

- Regular Expressions:



Zero-shot Experiment

Type	*TST*	REs
ACCESSORY_ID	96.81	97.08
ATTRIBUTE_VA	63.41	56.63
FIGURE_NOTE	96.97	98.20
FILE	92.98	88.98
PART_ID	58.06	75.51
TABLE_NOTE	97.67	97.67
micro-average	82.05	78.76

Table 6: F1 scores (%) for the initial *TST* and REs in zero-shot.

Rich-resource Experiment

Three baselines:

- PER (Jia et al., 2022): a specialized NER method in manufacturing.
- BILSTM (Siami-Namini et al., 2019) is one of the prominent deep learning models employed for addressing sequence-related tasks.
- TENER (Yan et al., 2019) utilizes the Transformer architecture to model information for NER tasks.

Rich-resource Experiment

	P	R	F1
TST	**65.96**	61.27	**63.53**
PER	58.07	**67.2**	62.3
BILSTM	55.28	64.67	59.61
TENER	53.37	63.98	58.19

Table 7: Micro-average F1 scores (%) for *TST* and baselines in rich-resource.

Rich-resource Experiment

	P	R	F1
TST	**65.96**	61.27	**63.53**
PER	58.07	**67.2**	62.3
BILSTM	55.28	64.67	59.61
TENER	53.37	63.98	58.19

Table 7: Micro-average F1 scores (%) for *TST* and baselines in rich-resource.

Few-shot Experiment

Four baselines:

- PER (Jia et al., 2022): a specialized NER method in manufacturing.
- Template-based BART (Cui et al., 2021): a template-based method for exploiting the few-shot learning potential of generative pretrained language models to sequence labeling.
- Prompt-Slot-Tagging (Hou et al., 2022) reversely predict slot values based on provided slot types. This approach incorporates training by considering the relationships between different slot types.
- NNShot (Yang and Katiyar, 2020) is a method based on nearest neighbor learning and structured inference. This approach uses a supervised NER model trained on the source domain, as a feature extractor.

Few-shot Experiment

- N-shot: N entities of each entity category.
- N%-sen: N% sentences extracted from the training set.

Few-shot Experiment

- N-shot: N entities of each entity category.
- N%-sen: N% sentences extracted from the training set.

	20-shot	50-shot	3%-sen	6%-sen
init-*TST*		29.51		
TST	26.7	**31.24**	**25.77**	**36.7**
PER	0.61	4.8	0.87	6.81
Template-based BART	12.52	15.68	13.87	17.54
Prompt Slot Tagging	16.38	17.08	17.44	23.2
NNShot	**27.2**	30.5	25.25	29.37

Table 8: Micro-average F1 scores (%) for *TST* and baselines in few-shot.

Few-shot Experiment

- N-shot: N entities of each entity category.
- N%-sen: N% sentences extracted from the training set.

	20-shot	50-shot	3%-sen	6%-sen
init-*TST*		29.51		
TST	26.7	**31.24**	**25.77**	**36.7**
PER	0.61	4.8	0.87	6.81
Template-based BART	12.52	15.68	13.87	17.54
Prompt Slot Tagging	16.38	17.08	17.44	23.2
NNShot	**27.2**	30.5	25.25	29.37

Table 8: Micro-average F1 scores (%) for *TST* and baselines in few-shot.

Few-shot Experiment

- N-shot: N entities of each entity category.
- N%-sen: N% sentences extracted from the training set.

	20-shot	50-shot	3%-sen	6%-sen
init-*TST*		29.51		
TST	26.7	**31.24**	**25.77**	**36.7**
PER	0.61	4.8	0.87	6.81
Template-based BART	12.52	15.68	13.87	17.54
Prompt Slot Tagging	16.38	17.08	17.44	23.2
NNShot	**27.2**	30.5	25.25	29.37

Table 8: Micro-average F1 scores (%) for *TST* and baselines in few-shot.

Bibliographical References

Ding Hongyu, Xi Daoyun, Zhang Xiufen, Han Linlin, Xiao Chengxaing, and Yunfeng Wang. 2008. Fundamental Terminology of Mechanical Manufacturing Processes. General Administration of Quality Supervision, Inspection and Quarantine of the People's Republic of China;Standardization Administration of China.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. Educational and psychological measurement, 20(1):37–46.

Meng Jia, Peiyan Wang, Guiping Zhang, and Dongfeng Cai. 2022. Named entity recognition for process text. Journal of Chinese Information Processing, 36(3):54–63.

Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. 2019. The performance of LSTM and bilstm in forecasting time series. In Proceedings of the 2019 IEEE International Conference on Big Data (IEEE BigData), pages 3285–3292, Los Angeles, CA, USA.

Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. 2019. TENER: adapting transformer encoder for named entity recognition. CoRR,abs/1911.04474.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In Findings of the Association for Computational Linguistics (ACL/IJCNLP), pages 1835–1845, Online.

Yutai Hou, Cheng Chen, Xianzhen Luo, Bohan Li, and Wanxiang Che. 2022. Inverse is better! fast and accurate prompt for few-shot slot tagging. In Findings of the Association for Computational Linguistics (ACL), pages 637–647, Dublin, Ireland.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6365–6375, Online

Thanks for your time!