ToNER: Type-oriented Named Entity Recognition with Generative Language Model

Guochao Jiang[†], **Ziqin Luo**[†], **Yuchen Shi**[†], **Dixuan Wang**[†], **Jiaqing Liang**[†‡], **Deqing Yang**[†‡✉]

[†]School of Data Science, Fudan University

[‡]Shanghai Key Laboratory of Data Science

[†]{gcjiang22, zqluo22, ycshi21, dxwang23}@m.fudan.edu.cn

[‡]{liangjiaqing, yangdeqing}@fudan.edu.cn

Abstract

In recent years, the fine-tuned generative models have been proven more powerful than the previous tagging-based or span-based models on named entity recognition (NER) task. It has also been found that the information related to entities, such as entity types, can prompt a model to achieve NER better. However, it is not easy to determine the entity types indeed existing in the given sentence in advance, and inputting too many potential entity types would distract the model inevitably. To exploit entity types' merit on promoting NER task, in this paper we propose a novel NER framework, namely *ToNER* based on a generative model. In ToNER, a type matching model is proposed at first to identify the entity types most likely to appear in the sentence. Then, we append a multiple binary classification task to fine-tune the generative model's encoder, so as to generate the refined representation of the input sentence. Moreover, we add an auxiliary task for the model to discover the entity types which further fine-tunes the model to output more accurate results. Our extensive experiments on some NER benchmarks verify the effectiveness of our proposed strategies in ToNER that are oriented towards entity types' exploitation.

Keywords: Named Entity Recognition, Natural Language Generation, Information Extraction, Information Retrieval

1. Introduction

As one representative task of information extraction, *named entity recognition* (NER) (Li et al., 2022a) has been the critical component to achieve plenty of downstream applications, such as the construction of knowledge graph (Xu et al., 2017), information retrieval (Banerjee et al., 2019), question answering (Mollá et al., 2006), and recommendation (Madani and Ez-zahout, 2022). The primary objective of NER is to identify the span(s) of entity mention(s) within a given input sentence and subsequently categorize each identified entity.

The previous NER solutions include tagging-based models (Strubell et al., 2017; Devlin et al., 2019; Wang et al., 2019; Sharma and Daniel Jr, 2019; Lee et al., 2019) and span-based models (Yu et al., 2020a; Li et al., 2020, 2022b; Wang et al., 2022b; Fu et al., 2021; Li et al., 2021). In recent years, some researchers have employed the generative pre-trained language models such as T5 (Raffel et al., 2020), BART (Lewis et al., 2020) and GPT-3 (Brown et al., 2020) to achieve NER, given their powerful capability of natural language generation. According to the input requirement of generative models, the given sentence and the candidate entity types are simultaneously input into the model as the prompt to trigger the generation of NER results. As shown in Figure 1, besides the sentence "China says time right for Taiwan talks.", the candidate entity type LOC, ORG, PER and MISC regarded as the schema, are also input into the model as the prompt. Then, the model would generate the entity span 'China' and 'Taiwan' existing in the sentence, and simultaneously assign the correct type LOC from the schema for each of them.

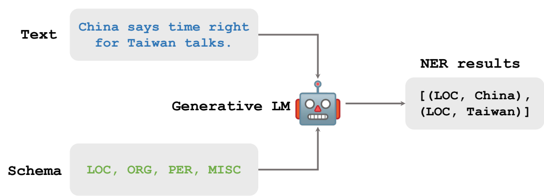


Figure 1: The standard inputs and outputs for a generative model to achieve NER. Besides the given sentence, all candidate entity types regarded as the schema are also input into the model as the prompt.

It has been found that leveraging the information of entity types can help the model recognize the entities in the sentence more accurately (Mo et al., 2023; Li and Qian, 2023). In general, there are many entity types in one NER corpus. It inevitably increases the difficulty of achieving accurate NER if too many types are input as the model's prompt. In addition, it is non-trivial to infer the entity types that are more likely to appear in the sentence in advance, and by now there are still no effective methods of discovering such entity types that can

LREC-COLING 2024, pages 16251–16262

be directly applied to generative models.

To address these problems, we propose a novel NER framework, namely *ToNER* (**T**ype-**o**riented **N**amed **E**ntity **R**cogition), which takes a generative model as the backbone and fully leverages the entity types to achieve enhanced NER. Specifically, we first introduce a small model to compute the matching degree between each candidate entity type and the input sentence, which is used to identify the types mostly likely to appear in the sentence. It helps the generative model concentrate on a limited number of credible types during achieving NER task. In addition, we add an additional multiple binary classification task to fine-tune the encoder in the generative model, so as to obtain optimal sentence representation which is benefit to generate more accurate NER results. Inspired by (Wang et al., 2022a; Lu et al., 2022; Wang et al., 2023), we further propose an auxiliary task for the generative model to recognize all entity types in the input sentence, which is different to the primary NER task but further fine-tunes the model to generate more accurate NER results.

Our main contributions in this paper are summarized as follows.

1. We propose a novel NER framework *ToNER* which successfully combines a generative pre-trained language model with a relatively small matching model to achieve more accurate NER.

2. We not only introduce a type matching model to discover the entity types most likely to appear in the input sentence, but also propose auxiliary learning tasks for fine-tuning the generative model, all of which can help ToNER obtain improved NER performance.

3. Our extensive experiments demonstrate that the proposed ToNER almost achieves the state-of-the-art (SOTA) performance on multiple representative NER benchmarks, and the effectiveness of each component in ToNER we propose is also justified, including the impact of adding Chain-of-Thoughta(CoT)-style explanations.

2. Related Work

Named Entity Recognition The task of Named Entity Recognition (NER) aims to identify spans expressing entities from text (Tjong Kim Sang and De Meulder, 2003), including three tasks: flat NER, nested NER, and discontinuous NER. Nested NER includes overlapping entities, while entities in discontinuous NER may include multiple nonadjacent spans. Overall, NER models can be divided into three types: those based on token sequence labeling, span classification, and seq2seq generation. In token-level models (Ratinov and Roth, 2009; Straková et al., 2019; Dai et al., 2020), each token is tagged as BIO or BILOU, and decoded using Conditional Random Fields (CRF) or other methods. In the category of span-level classification methods (Wang et al., 2020; Yu et al., 2020b), the text within the span is considered as a whole and classified using a classification model to determine whether it is an entity. Some methods based on hypergraphs (Lu and Roth, 2015; Wang and Lu, 2018) also fall into this category. In seq2seq generative models, the extraction target is encoded as a text sequence. Various methods (Cui et al., 2021; Yan et al., 2021) have explored different forms of input text and target coding, which we will introduce more in the next section.

Generative Methods for NER Benefiting from the development of generative Pretrained Language Models (PLMs), more and more work has adopted a sequence-to-sequence (seq2seq) approach to complete NER tasks. (Cui et al., 2021) models the NER task as a template filling task, using PLMs to fill in candidate spans and entity categories in pre-written templates. However, enumerating all possible spans is time-consuming. (Yan et al., 2021) transforms flat NER, nested NER, and discontinuous NER into a unified entity span sequence generation problem, and proposes a pointer-based framework based on BART to infer entity boundaries and categories simultaneously. Building on this, (Chen et al., 2021) introduces prompt-tuning to the attention mechanism of BART for low-resource scenarios. (Wang et al., 2022a) introduces task instructions and answer options in the input sentence, and directly extracts the required entity as the target output, by instructing the tuning of the T5 model. Compared to these works, our model adopts, and emphasizes, the role of entity type, and designs targeted auxiliary tasks.

Multi-task Learning in Information Extraction Many previous works (Wang et al., 2022a, 2023) have validated that introducing relevant intermediate tasks or auxiliary tasks in information extraction tasks can enhance the overall performance of the model. (Lu et al., 2022) models IE as a unified text-to-structure task. Besides the main extraction task, Universal Information Extraction (UIE)(Lu et al., 2022) also introduces an auxiliary learning task for the intermediate structured extraction language. (Wang et al., 2022a), besides the main NER task, introduces two auxiliary tasks - entity extraction and entity typing - which respectively enhance the model's ability to capture entity boundaries, and understand entity category information. Through ablation experiments, these two auxiliary tasks have been found to improve NER performance, especially in low-resource NER settings. (Wang et al., 2023) also validates that auxiliary tasks can provide additional information that complements the

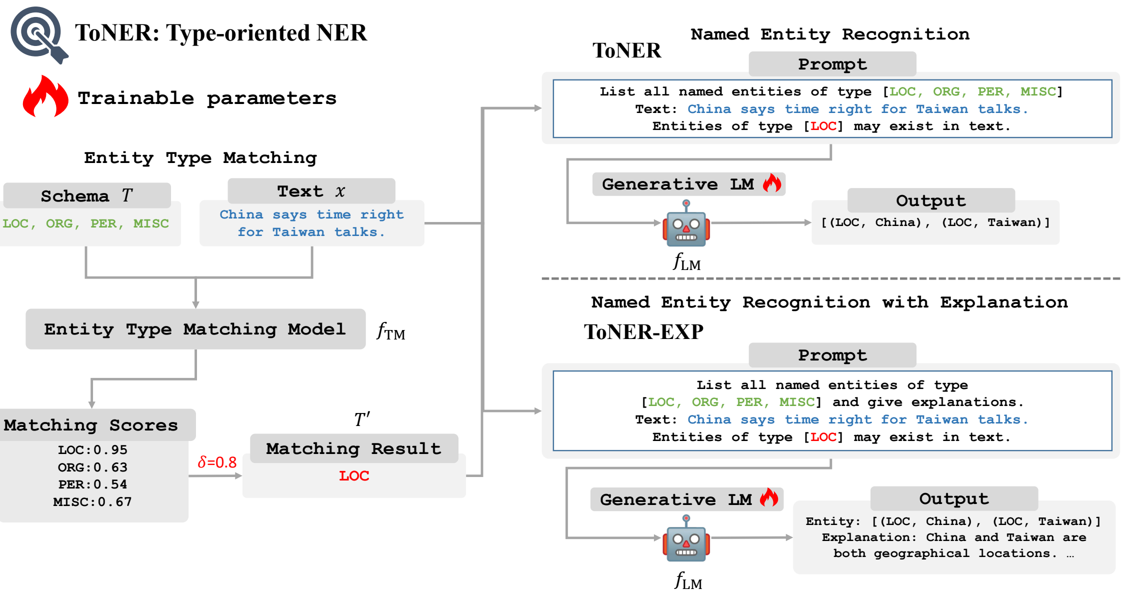


Figure 2: The pipeline of our proposed ToNER. ToNER achieves the NER task mainly with a generative model f_{LM}, which focuses more on the entity type LOC filtered out by the matching model f_{TM}. In the figure, the schema part is represented by green characters, the text to be extracted is represented by blue characters, and the entity type matching results are represented by red characters.

main task. For Named Entity Recognition, relationship extraction, and event extraction tasks, they designed span extraction and entity typing tasks, an entity pair extraction task and a relationship classification task, and a trigger extraction task and an argument extraction task, respectively.

3. Methodology

In this section, we first introduce the basic framework of achieving NER with a generative model, and then present our special designs in our ToNER to obtain enhanced NER performance.

3.1. Named Entity Recognition with A Generative Model

Formally, we denote the generative language model as f_{LM}, the input token sequence as $x = \{x_1, x_2, \cdots, x_m\}$, and the input instruction as $\mathcal{I}$. In addition, the output (generated) token sequence is denoted as $y = f_{\text{LM}}(x) = \{y_1, y_2, \cdots, y_n\}$. For the classic auto-regressive generative model, the sampling probability of the model generating y is formularized as

$$\mathbb{P}(y|\mathcal{I}, x) = \prod_{t=1}^{n} \mathbb{P}(y_t|\mathcal{I}, x, y_{<t}). \tag{1}$$

In our ToNER, we input the following prompt into the generative model to achieve NER,

List all named entities of type [T]
Text: x

Wherein T is the list of candidate entity types, i.e., the input schema.

Using generative models to achieve information extraction generally requires the model to output the results according to a given format. In ToNER, the generative model's outputs follow the format as
[(type$_1$, entity$_1$), (type$_2$, entity$_2$), ..., (type$_l$, entity$_l$)]
Among them, type$_i (1 \leq i \leq l)$ is the type assigned to the extracted (generated) entity span entity$_i$.

According to the generative model's rule of generating tokens, the loss of generating y is as follows,

$$\mathcal{L}_g = -\sum_{t=1}^{n} \log \mathbb{P}(y_t|\mathcal{I}, x, y_{<t}). \tag{2}$$

3.2. Entity Type Matching Model

As we introduced before, a predefined (candidate) list of entity types should be input as the schema into the generative model, to trigger the generation of NER. With such a prompt, the model needs to fully understand the semantics of each given entity type, based on which it then assigns the correct type for each generated entity span. This procedure implies that, too many candidate entity types would hinder the model from assigning the correct types for the entities in the sentence. As a result, reducing the entity types deserving to be cared about is the key to enhance the model's NER performance.

To this end, we introduce an entity type matching model in our ToNER, denoted as f_{TM}, which computes the semantic similarity, i.e., the matching

t_i	D_{t_i}
LOC	location: Names that are locations.
PER	person: Names of people.
ORG	organization: Companies, agencies, institutions, etc.
MISC	miscellaneous: Names of miscellaneous entities that do not belong to person, organization and location.

Table 1: The descriptions for some representative entity types in CoNLL2003 dataset. These descriptions provide the matching model with richer semantic information of entity types.

degree between each type and the sentence based on their semantic representation. Thus, the entity types most likely to appear in the sentence can be identified to reduce the number of entity types on which the model should concentrate.

Formally, suppose the original candidate entity type list (schema) is $T = \{t_1, t_2, \cdots, t_k\}$. Given that information of encoding $t_i (1 \leq i \leq k)$ is not sufficient to compute the accurate matching degree between t_i and x, we incorporate an additional description for t_i, of which the token sequence is denoted as D_{t_i}. Obviously, D_{t_i} contains more richer semantic information of t_i. Taking the dataset CoNLL2003 as an example, we list the descriptions for some representative entity types in Table 1, which were provided in the original paper (Tjong Kim Sang and De Meulder, 2003) .

Specifically, we adopt a BERT-like architecture for Entity Type Matching Model's encoder, denoted as E_{TM}, which converts a piece of input text into a representation through average pooling the last hidden state of each token in the text. Given x and a candidate t, the whole entity type matching model f_{TM} outputs the semantic similarity between x and t as

$$f_{\text{TM}}(x, t) = \frac{E_{\text{TM}}(x)^\top E_{\text{TM}}(D_t)}{\|E_{\text{TM}}(x)\|_2 \|E_{\text{TM}}(D_t)\|_2}, \tag{3}$$

where $E_{\text{TM}}(x) \in \mathbb{R}^d$ is x's semantic representation generated by the encoder. Thus, the entity types in T with the semantic similarity score higher than the threshold δ are retained as the possible types in the sentence, which constitute a new schema denoted as T'.

Next, with T' we modify the prompt of the generative model as follows,

```
List all named entities of type [T]
Text: x
Entities of type [T'] may exist in
text.
```

With such a prompt, the generative model can focus more on the types in T' rather than T, which reduces the difficulty of achieving NER with the model. We still list the original schema T in the prompt to ensure the model does not miss the correct entity types not in T'.

The pipeline of incorporating the entity types filtered out based on the matching model into the generative model is shown in Figure 2.

In order to train f_{TM}, we have collected sufficient samples from the original NER benchmark. Formally, suppose the sets of entity types that are mentioned and not mentioned in x are denoted as $\mathcal{P}_x$ and $\mathcal{N}_x$, respectively, then inspired by SimCSE (Gao et al., 2021), we propose the following the loss to train f_{TM},

$$\mathcal{L}_m = - \sum_{t^+ \in \mathcal{P}_x} \log \frac{e^{f_{\text{TM}}(x, t^+)/\tau}}{\sum_{t \in \mathcal{P}_x \bigcup \mathcal{N}_x} e^{f_{\text{TM}}(x,t)/\tau}}, \tag{4}$$

where τ is a hyperparameter of temperature .

3.3. Fine-tuning Encoder with Type Classification

For a model with an encoder-decoder architecture, the encoder is its critical component since the model's results are generated mainly based on the representations learned by the encoder. As we know, the generative pre-trained language models are pre-trained through the task different to NER, although they can directly achieve NER task. Thus, we believe that fine-tuning the encoder in ToNER with the task more related to NER would help the encoder generate refined representations in terms of improved NER. Since we have found that the entity types existing in the sentence are helpful, we propose a multiple binary classification task as an auxiliary learning task of ToNER to train a better encoder, resulting in more accurate generations of NER.

Formally, suppose $h(x)$ is x's representation which is generated by the encoder through the average pooling upon the hidden states of all tokens in x. Then, we adopt a neural classifier c to map $h(x)$ to a k-dimensional vector as $c(h(x)) = [p_1, p_2, \cdots, p_k] \in \mathbb{R}^k$, where $p_i (1 \leq i \leq k)$ is the logit corresponding to the candidate entity types t_i, indicating whether t_i appears in x or not. The loss of this multiple binary classification is as follows,

$$\begin{aligned} \mathcal{L}_c = \log \left(1 + \sum_{i=1}^{k} \mathbb{1}\{t_i \in \mathcal{P}_x\} e^{p_i}\right) + \\ \log \left(1 + \sum_{i=1}^{k} \mathbb{1}\{t_i \in \mathcal{N}_x\} e^{-p_i}\right), \end{aligned} \tag{5}$$

where $\mathbb{1}\{\}$ is indicator function.

Thus, the overall training loss for ToNER is

$$\mathcal{L} = \mathcal{L}_g + \lambda \mathcal{L}_c, \tag{6}$$

where λ is the controlling parameter.

3.4. Improving NER by An Auxiliary Task

Many previous works (Wang et al., 2022a; Lu et al., 2022; Wang et al., 2023) have verified that for information extraction tasks, adding relevant intermediate tasks or auxiliary tasks (such as entity typing, entity extraction and relation extraction) in the instruction fine-tuning stage is beneficial to improve the model's performance on the primary extraction task. Inspired by them, we also add an auxiliary task in the instruction fine-tuning stage to explicitly encourage the model recognize the entity types that may exist in the input sentence.

Similar to the instruction prompt of NER, we construct the following prompt to ask the generative model to list all entity types in the sentence.

```
List all entity types in the text
from type [T]
Text: x
```

To construct this auxiliary task's training samples, we randomly select some training samples from the datasets, each of which only takes the entity types as its label (model output). Obviously, if the generative model can accomplish this auxiliary task well, it can also generate satisfactory NER results since these two tasks are very correlated.

3.5. Achieving NER with CoT-style Explanations

Chain-of-Thought (CoT) Prompting (Wei et al., 2022b) has been widely used to improve the performance of large language models (LLMs) on various NLP tasks. A recent study (Wadhwa et al., 2023) has found that using CoT-style explanations generated by LLMs to fine-tune the relatively small generative models can help improve the small models' performance on relation extraction. It inspires us to investigate whether adding CoT-style explanation besides outputting the standard NER results could improve our ToNER's performance.

To this end, for each training sample in the datasets we used an LLM (such as ChatGPT[1] or GPT4 (OpenAI, 2023)) to generate the explanations for its NER result. Next, we used these augmented samples with CoT-style explanations to fine-tune the generative model, and thus get a advanced variant of ToNER, denoted as ToNER-EXP. In the following experiments, we will compare the performance of ToNER and ToNER-EXP. We list two instances of such training samples from CoNLL2003 dataset as follows.

[1] https://openai.com/blog/chatgpt/

```
Input List all named entities of
type [organization, person, miscel-
laneous, location] and give explana-
tions.
Text: - Wellington newsroom 64 4
4734 746
Entities of type [location] may ex-
ist in text.
Output Entity: [(location, Welling-
ton)]
Explanation: 'Wellington' is la-
beled as 'location' because it
refers to a specific location,
which is the capital city of New
Zealand.
=====================================
Input List all named entities of
type [organization, person, miscel-
laneous, location] and give explana-
tions.
Text: The bank said there were con-
cerns fiscal consolidation would
unduly restrict growth, but evidence
was ambiguous.
Entities of type [] may exist in
text.
Output Entity: []
Explanation: No entity in the text
belongs to any pre-defined entity
type.
```

4. Experiment

In this section, we evaluate the performance of our ToNER with some previous NER models on several NER datasets, and further analyze the experiment results.

4.1. Datasets

We conducted our experiments on the following five NER benchmarks.

1. **CoNLL2003** (Tjong Kim Sang and De Meulder, 2003) is a collection of news wire articles from the Reuters Corpus, which contains 4 entity types including `LOC`, `ORG`, `PER` and `MISC`.
2. **OntoNotes 5.0** (Hovy et al., 2006) is a large corpus comprising various genres of text (news, conversational telephone speech, weblogs, usenet newsgroups, broadcast, talk shows). We only considered its English samples in our experiments.
3. **JNLPBA** (Collier et al., 2004) is a biomedical dataset from the GENIA version 3.02, which contains 5 entity types including `DNA`, `RNA`, `cell_type`, `cell_line` and `protein`. It was created with a controlled search on MEDLINE.
4. **ACE2004** (Alexis et al., 2004) and **ACE2005** (Walker et al., 2006) are two nested NER datasets,

Para.	Value	Comment
batch size	32 or 8	32 for Flan-T5-large and Flan-T5-xl, 8 for Flan-T5-xxl
max length	512	max token length for input and output
lr	3e-5	learning rate for ToNER
λ	0.1	the controlling parameter of $\mathcal{L}_c$ in Eq. 6
τ	0.05	temperature coefficient for matching model loss 4
δ	0.8/0.7/0.6	type matching threshold of CoNLL2003/JNLPBA/OntoNotes 5.0, ACE2004, ACE2005

Table 2: Some important hyperparameter settings for ToNER implementation.

Model	**CoNLL2003**			**OntoNotes 5.0**		
	P	R	F1	P	R	F1
Strubell et al. (2017)	-	-	90.65	-	-	86.84
Devlin et al. (2019)	-	-	92.80	90.01	88.35	89.16
Yu et al. (2020a)	92.91	92.13	92.52	90.01	89.77	89.89
Li et al. (2020)	92.33	**94.61**	93.04	**92.98**	89.95	91.11
Yan et al. (2021)	92.61	93.87	93.24	89.99	90.77	90.38
Li et al. (2022b)	92.71	93.44	93.07	90.03	90.97	90.50
Wang et al. (2023)	-	-	92.94	-	-	90.19
Shen et al. (2023)	92.99	92.56	92.78	90.31	91.02	90.66
$\text{ToNER}_{\text{large}}$	**93.55**	<u>94.11</u>	**93.83**	<u>91.16</u>	<u>91.35</u>	<u>91.25</u>
ToNER_{xl}	<u>93.53</u>	93.65	<u>93.59</u>	91.11	**91.50**	**91.30**

Table 3: All models' NER performance on CoNLL2003 and OntoNotes 5.0. **Bold** and <u>underline</u> denote the best and second best scores, respectively.

which contain 7 entity types including `PER`, `ORG`, `LOC`, `GPE`, `WEA`, `FAC` and `VEH`, and are generally used to evaluate the more complicate task of overlapped NER. We followed the same setup as the previous work (Katiyar and Cardie, 2018; Lin et al., 2019).

4.2. Baselines

We compared our ToNER with the following NER baselines belonging to different families, including the tagging-based methods (Strubell et al., 2017; Devlin et al., 2019; Wang et al., 2019; Sharma and Daniel Jr, 2019; Lee et al., 2019), the span-based methods (Yu et al., 2020a; Li et al., 2020, 2022b; Wang et al., 2022b; Fu et al., 2021; Li et al., 2021), and the generation-based methods (Yan et al., 2021; Wang et al., 2023; Shen et al., 2023; Fei et al., 2021).

4.3. Implementation Details

We report the entity-level micro Precision (P), Recall (R) and F1 scores of all compared models in the following result figures and tables. To construct ToNER, we selected Flan-T5 (Chung et al., 2022) as the generative model in our framework. We used AdamW (Loshchilov and Hutter, 2017) as our models' optimizer. We also selected GTE-large (Li et al., 2023) as the entity type matching model and fine-tuned it with the AdamW of 1 epoch, where the learning rate is 8e-6, and the weight decay is 1e-3. The CoT-style explanations were generated by GPT4. Other important hyperparameters are listed in Table 2, which were decided based on our tuning studies. We conducted our experiments on eight NVIDIA Tesla A100 GPU with 80GB of GPU memory.

4.4. Overall Performance Comparisons

For the baselines' NER performance on the different datasets, we directly report their results published on the previous paper. Thus different baselines are reported in the result Table 3 $\sim$ 5 corresponding to different datasets, where the best scores and the second best scores are bold and underlined, respectively. Specifically, we compared our ToNER with Flan-T5-large and Flan-T5-xl with the baselines, which are denoted as $\text{ToNER}_{\text{large}}$ and ToNER_{xl}, respectively.

From the tables, we find that on CoNLL2003, OntoNotes 5.0 and JNLPBA, our $\text{ToNER}_{\text{large}}$ or ToNER_{xl} achieves the best F1. Since CoNLL2003 and OntoNotes 5.0 are both NER datasets in general fields, and JNLPBA is the dataset of the biological field, it shows that our ToNER is more effective than the baselines in both general fields and the special field. We also compared our framework's performance with the baselines on ACE2004 and ACE2005 for the task of overlapped NER, of which the results are listed in Table 5. Although the baseline Shen et al. (2023) achieves the best F1 on this task, our ToNER_{xl} can also obtain quite competi-

Model	JNLPBA		
	P	R	F1
Fu et al. (2021)	-	-	74.49
Wang et al. (2022b)	-	-	77.03
Sharma and Daniel Jr (2019)	-	-	77.03
Lee et al. (2019)	72.68	**83.21**	77.59
$\text{ToNER}_{\text{large}}$	**77.88**	79.55	78.71
ToNER_{xl}	76.31	82.09	**79.09**

Table 4: Results for JNLPBA. **Bold** and underline denote the best and second best scores.

Model	ACE2004			ACE2005		
	P	R	F1	P	R	F1
Yu et al. (2020a)	87.30	86.00	86.70	85.20	85.60	85.40
Li et al. (2020)	85.05	86.32	85.98	**87.16**	86.59	86.88
Yan et al. (2021)	87.27	86.41	86.84	83.16	86.38	84.74
Li et al. (2022b)	87.33	87.71	87.52	85.03	**88.62**	86.79
Shen et al. (2023)	88.11	**88.66**	**88.39**	86.15	87.72	**86.93**
$\text{ToNER}_{\text{large}}$	88.39	85.29	86.81	84.74	84.68	84.71
ToNER_{xl}	**90.03**	86.24	88.09	86.66	86.71	86.68

Table 5: Results for ACE2004 and ACE2005. **Bold** and underline denote the best and second best scores.

tive performance.

4.5. Ablation Study

In order to further justify the effectiveness of each component we propose in ToNER, we compared ToNER with its ablated variants. Specifically, the basic variant only uses the generative model Flan-T5-large and Flan-T5-xl. Then, we added the type matching model f_{TM}, the type classification (TC) task and the auxiliary of type recognition (TR) into this basic variant in turn. Due to space limitation, table 6 only lists the performance of $\text{ToNER}_{\text{large}}$ and ToNER_{xl} along with their corresponding three ablated variants on CoNLL2003. As well, each variant's performance improvement rate relative to the preceding variant is also listed. This table's results obviously show that either f_{TM}, TC or TR can improve ToNER's performance. The ablation studies on other datasets also support this conclusion.

4.6. Threshold Selection of Entity Type Matching

We also investigated the type matching model f_{TM}'s capability of discriminating between the positive text-type pairs and the negative text-type pairs, since it is a key part of ToNER to improve NER performance. Figure 4 displays the distributions of all text-type pairs' similarity score's in the five datasets, which were computed by the fine-tuned f_{TM} according to Eq. 3. The distributions show that f_{TM} can well discriminate between the positive pairs and the negative pairs, based on which we can select the best threshold δ, as shown in the five sub-figures.

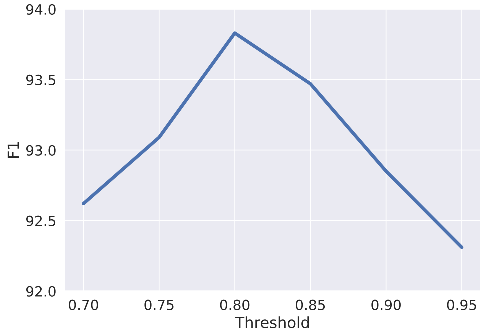


Figure 3: The performance of $\text{ToNER}_{\text{large}}$ using different threshold δ on CoNLL2003.

Model	F1
Flan-T5-large (Chung et al., 2022)	87.11
Flan-T5-large+f_{TM}	91.18(+4.67%)
Flan-T5-large+f_{TM}+TC	92.23(+1.15%)
Flan-T5-large+f_{TM}+TC+TR ($\text{ToNER}_{\text{large}}$)	**93.83**(+1.73%)
Flan-T5-xl (Chung et al., 2022)	89.05
Flan-T5-xl+f_{TM}	91.21(+2.16%)
Flan-T5-xl+f_{TM}+TC	92.13(+0.92%)
Flan-T5-xl+f_{TM}+TC+TR (ToNER_{xl})	**93.59**(+1.46%)

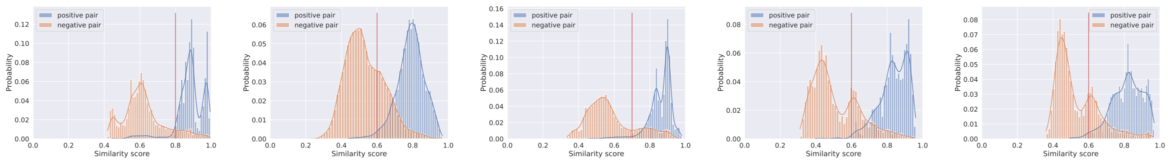
Table 6: Ablation study results of $\text{ToNER}_{\text{large}}$ and ToNER_{xl} on CoNLL2003. f_{Tm} means the entity type matching model in Section 3.2. TC means the entity type classification for fine-tuning Encoder in Section 3.3. TR means the auxiliary entity type recognition task in Section 3.4.

We also tested δ's impact on ToNER's performance. Figure 3 depicts $\text{ToNER}_{\text{large}}$'s F1 score on CoNLL2003 as δ varies from 0.7 to 0.95. It shows that $\delta = 0.8$ is the best setting for this dataset.

4.7. Impacts of CoT-style Explanations and Model Size

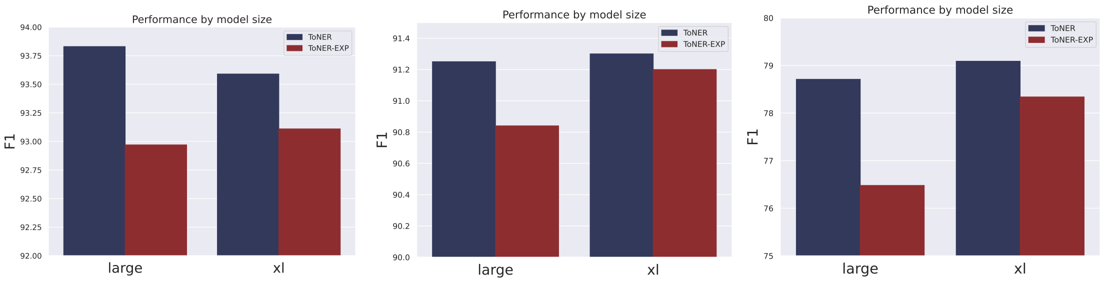
It has been found that CoT only has a positive effect on sufficiently large models (typically containing 10B or more parameters) but not on small models (Wei et al., 2022c), since CoT is an emergent ability (Wei et al., 2022a). In order to explore the impact of adding CoT explanations to achieve NER task, we compared the performance of ToNER and ToNER-EXP in the `large` and `xl` setting, as shown in Figure 5. For these datasets, the model performance improvement of ToNER-EXP is higher than that of ToNER when the model size changes from `large` to `xl`. This suggests that the model's ability to generate CoT-style explanations may gradually increase as the number of parameters increases. Increasing model size not only helps direct NER performance but also improves CoT-style explanation and helps generate NER results indirectly.

In order to further explore the effect of further increasing the model size, we selected the CoNLL2003 dataset for further exploration. Specifically, besides $\text{ToNER}_{\text{large}}$ and ToNER_{xl}, we further

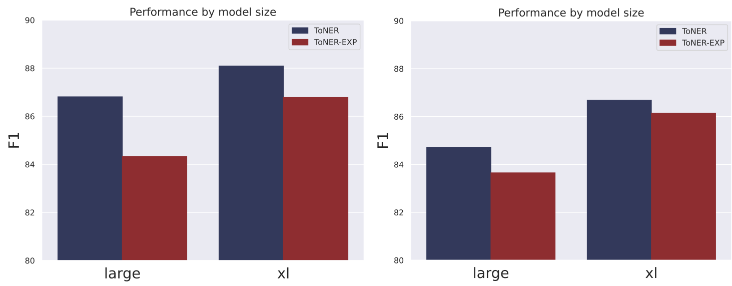


(a) CoNLL2003 (b) OntoNotes 5.0 (c) JNLPBA (d) ACE2004 (e) ACE2005

Figure 4: Similarity score distributions of all text-type pairs computed by the fine-tuned type matching model. The red vertical line represents the threshold δ we selected.



(a) CoNLL2003 (b) OntoNotes 5.0 (c) JNLPBA



(d) ACE2004 (e) ACE2005

Figure 5: The performance of ToNER and ToNER-EXP with different model size on different datasets.

Model	**CoNLL2003**		
	P	R	F1
ToNER$_{large}$	<u>93.55</u>	**94.11**	**93.83**
ToNER$_{large}$-EXP	93.18 (-0.40%)	92.77 (-1.42%)	92.97 (-0.92%)
ToNER$_{xl}$	93.53	<u>93.65</u>	93.59
ToNER$_{xl}$-EXP	93.10 (-0.46%)	93.13 (-0.56%)	93.11 (-0.51%)
ToNER$_{xxl}$	92.74	92.28	92.52
ToNER$_{xxl}$-EXP	**93.93** (+1.28%)	93.47 (+1.29%)	<u>93.70</u> (+1.28%)

Table 7: ToNER and ToNER-EXP's performance on CoNLL2003. **Bold** and <u>underline</u> denote the best and second best scores.

considered ToNER$_{xxl}$, which use the Flan-T5 versions of 780M, 3B, 11B parameters, respectively. Table 7 lists their performance on CoNLL2003, including the relative performance improvement rate of each version's ToNER-EXP to its corresponding ToNER. From the table we find that, only for the xxl version, ToNER-EXP can improve NER performance, verifying the previous finding that CoT's effectiveness on large models rather than small models. When the generative model's scale is large enough, the CoT-style explanations can fine-tune the model to better utilize its rich knowledge to understand the input texts correctly, thus improving NER performance further.

Another interesting observation is that, ToNER$_{large}$ has the best R score and F1 score although it only has the fewest parameters. Instead, further increasing the parameters of the generative model degrades the performance. This could be attributed to that, fine-tuning a large model for optimal performance necessitates a broader and larger dataset. The available dataset can not meet this requirement. Our experiments on other datasets have the similar results.

5. Conclusion

In this paper, we propose a novel NER framework *ToNER* based on a generative language model. In ToNER we further employ an entity type matching model to discover the entity types mostly likely to appear in the sentence, which are input into the generative model for more concentrations. Additional classification learning objectives are also designed to fine-tune the generative model, to improve ToNER's performance further. At the same time, we also explored the impact of generating CoT-style explanations for model outputs. Our experiments on five NER datasets illustrate the advantages of ToNER over the previous models.

Limitations

We only explore the feasibility of the Encoder-Decoder architecture model as a base model for ToNER. For generative language models, more existing options are based on Decoder-only. This limitation highlights the potential for future work to explore different model architectures to understand named entity recognition.

Acknowledgements

This work was supported by the Chinese NSF Major Research Plan (No.92270121), Youth Fund (No.62102095), Shanghai Science and Technology Innovation Action Plan (No.21511100401). The computations in this research were performed using the CFFF platform of Fudan University.

Bibliographical References

Partha Sarathy Banerjee, Baisakhi Chakraborty, Deepak Tripathi, Hardik Gupta, and Sourabh S. Kumar. 2019. A information retrieval based on question and answering and ner for unstructured information without using sql. *Wirel. Pers. Commun.*, 108(3):1909–1931.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Xiang Chen, Lei Li, Shumin Deng, Chuanqi Tan, Changliang Xu, Fei Huang, Luo Si, Huajun Chen, and Ningyu Zhang. 2021. Lightner: A lightweight tuning paradigm for low-resource NER via pluggable prompting. *CoRR*, abs/2109.00720.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models. *arXiv e-prints*, page arXiv:2210.11416.

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 2021*, pages 1835–1845, Online. Association for Computational Linguistics.

Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cecile Paris. 2020. An effective transition-based model for discontinuous NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5860–5870, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Hao Fei, Donghong Ji, Bobo Li, Yijiang Liu, Yafeng Ren, and Fei Li. 2021. Rethinking boundaries: End-to-end recognition of discontinuous mentions with pointer networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12785–12793.

Jinlan Fu, Xuanjing Huang, and Pengfei Liu. 2021. SpanNER: Named entity re-/recognition as span prediction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7183–7195, Online. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 861–871, New Orleans, Louisiana. Association for Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Fei Li, ZhiChao Lin, Meishan Zhang, and Donghong Ji. 2021. A span-based model for joint overlapped and discontinuous named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4814–4828, Online. Association for Computational Linguistics.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2022a. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Jingye Li, Hao Fei, Jiang Liu, Shengqiong Wu, Meishan Zhang, Chong Teng, Donghong Ji, and Fei Li. 2022b. Unified named entity recognition as word-word relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10965–10973.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.

Yongqi Li and Tieyun Qian. 2023. Type-aware decomposed framework for few-shot named entity recognition. *arXiv preprint arXiv:2302.06397*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards General Text Embeddings with Multi-stage Contrastive Learning. *arXiv e-prints*, page arXiv:2308.03281.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2019. Sequence-to-nuggets: Nested entity mention detection via anchor-region networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5182–5192, Florence, Italy. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 857–867, Lisbon, Portugal. Association for Computational Linguistics.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5755–5772. Association for Computational Linguistics.

Rabie Madani and Abderrahmane Ez-zahout. 2022. A review-based context-aware recommender systems: Using custom ner and factorization machines. *International Journal of Advanced Computer Science and Applications*, 13(3).

Ying Mo, Hongyin Tang, Jiahao Liu, Qifan Wang, Zenglin Xu, Jingang Wang, Wei Wu, and Zhoujun Li. 2023. Multi-task transformer with relation-attention and type-attention for named entity recognition. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Diego Mollá, Menno van Zaanen, and Daniel Smith. 2006. Named entity recognition for question answering. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 51–58, Sydney, Australia.

OpenAI. 2023. Gpt-4 technical report.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.

Shreyas Sharma and Ron Daniel Jr. 2019. Bioflair: Pretrained pooled contextualized embeddings for biomedical sequence labeling tasks. *arXiv preprint arXiv:1908.05760*.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. DiffusionNER: Boundary diffusion for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3875–3890, Toronto, Canada. Association for Computational Linguistics.

Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested NER through linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331, Florence, Italy. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2670–2680, Copenhagen, Denmark. Association for Computational Linguistics.

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.

Somin Wadhwa, Silvio Amir, and Byron Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15566–15589, Toronto, Canada. Association for Computational Linguistics.

Bailin Wang and Wei Lu. 2018. Neural segmental hypergraphs for overlapping mention recognition. In *Conference on Empirical Methods in Natural Language Processing*.

Jue Wang, Lidan Shou, Ke Chen, and Gang Chen. 2020. Pyramid: A layered model for nested named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5918–5928, Online. Association for Computational Linguistics.

Liwen Wang, Rumei Li, Yang Yan, Yuanmeng Yan, Sirui Wang, Wei Wu, and Weiran Xu. 2022a. Instructionner: A multi-task instruction-based generative framework for few-shot ner.

Xiao Wang, Shihan Dou, Limao Xiong, Yicheng Zou, Qi Zhang, Tao Gui, Liang Qiao, Zhanzhan Cheng, and Xuanjing Huang. 2022b. MINER: Improving out-of-vocabulary named entity recognition from an information theoretic perspective. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5590–5600, Dublin, Ireland. Association for Computational Linguistics.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, et al. 2023. Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. CrossWeigh: Training named entity tagger from imperfect annotations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5154–5163, Hong Kong, China. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022c. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Bo Xu, Yong Xu, Jiaqing Liang, Chenhao Xie, Bin Liang, Wanyun Cui, and Yanghua Xiao. 2017. Cn-dbpedia: A never-ending chinese knowledge extraction system. In *Advances in Artificial Intelligence: From Theory to Practice - 30th International Conference on Industrial Engineering* *and Other Applications of Applied Intelligent Systems, IEA/AIE 2017, Arras, France, June 27-30, 2017, Proceedings, Part II*, volume 10351 of *Lecture Notes in Computer Science*, pages 428–438. Springer.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various NER subtasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5808–5822, Online. Association for Computational Linguistics.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020a. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online. Association for Computational Linguistics.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020b. Named entity recognition as dependency parsing. In *Annual Meeting of the Association for Computational Linguistics*.

Language Resource References

Alexis, Mitchell and Stephanie, Strassel and Shudong, Huang and Ramez, Zakhary. 2004. *ACE 2004 Multilingual Training Corpus*. European Language Resources Association (ELRA), ISLRN 789-870-824-708-5.

Collier, Nigel and Ohta, Tomoko and Tsuruoka, Yoshimasa and Tateisi, Yuka and Kim, Jin-Dong. 2004. *Introduction to the Bio-entity Recognition Task at JNLPBA*. COLING. PID https://aclanthology.org/W04-1213.

Hovy, Eduard and Marcus, Mitchell and Palmer, Martha and Ramshaw, Lance and Weischedel, Ralph. 2006. *OntoNotes: The 90% Solution*. Association for Computational Linguistics, ISLRN 151-738-649-048-2.

Tjong Kim Sang, Erik F. and De Meulder, Fien. 2003. *Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition*. PID https://aclanthology.org/W03-0419.

Walker, Christopher and Strassel, Stephanie and Medero, Julie and Maeda, Kazuaki. 2006. *ACE 2005 Multilingual Training Corpus*. European Language Resources Association (ELRA), ISLRN 458-031-085-383-4.