Nested Named Entity Recognition: A Survey

YU WANG, Nanjing University of Posts and Telecommunications
HANGHANG TONG, University of Illinois at Urbana-Champaign
ZIYE ZHU and YUN LI, Nanjing University of Posts and Telecommunications

With the rapid development of text mining, many studies observe that text generally contains a variety of implicit information, and it is important to develop techniques for extracting such information. Named Entity Recognition (NER), the first step of information extraction, mainly identifies names of persons, locations, and organizations in text. Although existing neural-based NER approaches achieve great success in many language domains, most of them normally ignore the nested nature of named entities. Recently, diverse studies focus on the nested NER problem and yield state-of-the-art performance. This survey attempts to provide a comprehensive review on existing approaches for nested NER from the perspectives of the model architecture and the model property, which may help readers have a better understanding of the current research status and ideas. In this survey, we first introduce the background of nested NER, especially the differences between nested NER and traditional (i.e., flat) NER. We then review the existing nested NER approaches from 2002 to 2020 and mainly classify them into five categories according to the model architecture, including early rule-based, layered-based, region-based, hypergraph-based, and transition-based approaches. We also explore in greater depth the impact of key properties unique to nested NER approaches from the model property perspective, namely entity dependency, stage framework, error propagation, and tag scheme. Finally, we summarize the open challenges and point out a few possible future directions in this area. This survey would be useful for three kinds of readers: (i) Newcomers in the field who want to learn about NER, especially for nested NER. (ii) Researchers who want to clarify the relationship and advantages between flat NER and nested NER. (iii) Practitioners who just need to determine which NER technique (i.e., nested or not) works best in their applications.

CCS Concepts: • **General and reference → Surveys and overviews**; • **Computing methodologies → Information extraction;**

Additional Key Words and Phrases: Nested named entity recognition, named entity recognition, information extraction, natural language processing, text mining

ACM Reference format:
Yu Wang, Hanghang Tong, Ziye Zhu, and Yun Li. 2022. Nested Named Entity Recognition: A Survey. *ACM Trans. Knowl. Discov. Data.* 16, 6, Article 108 (July 2022), 29 pages.
https://doi.org/10.1145/3522593

This work is partially supported by National Natural Science Foundation of China (No. 61772284), Postgraduate Research & Practice Innovation Program of Jiangsu Province (SJKY19_0766). Hanghang Tong is partially supported by NSF (1947135, 2134079, and 1939725).

Authors' addresses: Y. Wang, Z. Zhu, and Y. Li (corresponding author), Nanjing University of Posts and Telecommunications, No. 9, Wenyuan Road, Yadong New District, Nanjing, Jiangsu, 210023, China; emails: {2017070114, 2016070251, liyun}@njupt.edu.cn; H. Tong, University of Illinois at Urbana-Champaign, 201 North Goodwin Avenue, Urbana, Illinois, 61801-2302; email: htong@illinois.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2022 Association for Computing Machinery.

1556-4681/2022/07-ART108 $15.00
https://doi.org/10.1145/3522593

1 INTRODUCTION

Named Entity Recognition (**NER**) aims at recognizing name entities, which are words or phrases containing the names of predefined categories like location, organization, or medical code. NER has always been a classic and popular topic in **natural language processing** (**NLP**) and text mining, as it is an important pre-processing step for downstream tasks such as information extraction [26], text summarization [43], and question answering [19]. Normally, named entities have a complex nested structure; that is, a named entity can contain or embed other entities. Recognizing named entities with the nested structure, referred to as nested NER task, has become an emerging topic in the NER task and benefited various natural language applications.

The term "named entity" first appeared at the **Sixth Message Understanding Conference** (**MUC-6**) [17] held in November 1995, as a task of identifying the names of organizations, people, and geographic locations in text, as well as currency, time, and percentage expressions. Later on, the **Entity Detection and Tracking** (**EDT**) task from the **Automatic Content Extraction** (**ACE**) [51] proposed that it is necessary to recognize all mentions of an entity, whether a name, a description, or a pronoun, and then classify them into equivalence classes based on references to the same entity. Due to the growing interest in NER, various scientific events (e.g., CoNLL03 [45], IREX [8], and TREC Entity Track [3]) have devoted much effort to this topic. There will be some subtle differences in the named entities involved in different task scenarios. Formally, the named entities mentioned in the information extraction are real-world objects that can be denoted with proper names. Whereas in the field of NLP, named entities can simply be viewed as entity instances existing in text. Despite the diverse definitions of the named entity, a named entity generally is a word or a phrase that clearly identifies one item from a set of other items that have similar attributes. In contrast, the types of name entities to recognize are unified in the literature. Entities are commonly divided into generic named entities (i.e., person, organization, and location names in the general domain) and domain-specific named entities (e.g., gene, protein, and disease names in the biomedical domain).

In the ACE pilot study task definition,[1] the term "nested region names" was mentioned for the first time. The nested nature of named entities has since gained special attention. Entity mentions will frequently be nested; that is, they will contain mentions of other entities. Shen et al. [2003] [46] also raised the problem of the nested named entity and referred to it as "cascaded phenomena". Their statistics show that 16.57% of named entities in the GENIA corpus have nested annotations, and they stated that the cascaded phenomenon is the major factor limiting the overall performance of previous NER systems. With the gradual deepening research of NER, Alex et al. [2007] [1] suggested that the nesting can occur in three different ways, including (1) entities containing one or more shorter embedded entities, (2) entities with more than one entity category, and (3) coordinated entities (more broadly known as discontinuous entities). Lu and Roth [2015] [33] used the term "overlapping" to collectively represent both nested and overlapped structures. From the various definitions above, we can appreciate the complexity of nested named entities and thus consciously group them as a special kind of entity. In response, named entities are generally further subdivided into five types: flat named entities, nested named entities, multi-category named entities, overlapping named entities, and discontinuous named entities. In this survey, we mainly focus on the nested named entity, which is the most ordinary type other than flat named entities.

In recent years, a considerable amount of research has been proposed for the nested NER task and achieved state-of-the-art performance. To the best of our knowledge, there are no reviews or surveys in this field so far. This trend motivates us to conduct a survey to report on the current

[1] https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/edt-phase1-v2.2.pdf.

state of nested NER research. In this survey article, we give a comprehensive overview of eighteen years (from 2003 to 2020) of research in the nested NER field from two perspectives, the model architecture and the model property. To be specific, we first focus on the model architecture perspective to generalize all existing approaches into the following five main streams, (1) early rule-based approaches (from 2003 to 2006), (2) layered-based approaches (from 2004 to 2020), (3) region-based approaches (from 2007 to 2020), (4) hypergraph-based approaches (from 2015 to 2018), and (5) transition-based approaches (from 2009 to 2019). We then further categorize the above-mentioned nested NER approaches from the model property perspective, based on whether these approaches preserve entity dependency, belong to the single-stage framework, suffer from error propagation, or require multiple tag schemes. Finally, we summarize some open challenges and point out a few potential future directions in nested NER research. The main contributions of this survey article are highlighted as follows:

- To the best of our knowledge, we connect and systematize the existing nested NER research studies for the first time, as well as to give a comprehensive overview in this field.
- We systematically organize all existing approaches into five main streams, where we further present the pros and cons of each stream.
- We dig more deeply into several key properties unique to nested NER approaches, including entity dependency, stage framework, error propagation, and tag scheme.
- We present readers with the challenges faced by nested NER systems and outline future directions in this field.

The rest of the survey is organized as follows. Section 2 introduces the background of nested NER, including definitions, resources, evaluation metrics, and related surveys. Section 3 overviews two proposed categorizations of nested NER for convenience. Sections 4 and 5 interpret nested NER approaches from the model architecture and the model property, respectively. Section 6 lists the challenges and future directions. Finally, we conclude this survey in Section 7.

2 BACKGROUND

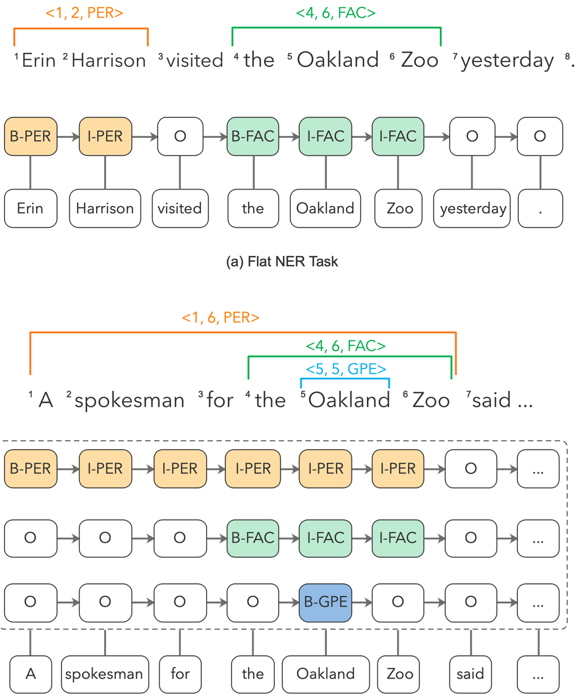
In this section, we first introduce the definitions of flat NER task and nested NER task, as well as some discussion and comparison of these two tasks. This allows the reader to understand and distinguish them more clearly. Furthermore, we especially summarize the major benchmark datasets widely used in these two tasks, considering that datasets serve as the primary ingredient in NER tasks. At the end of this section, we also briefly introduce the evaluation metrics and related surveys.

2.1 What is Flat NER?

The traditional NER task aims to identify named entities within the text that satisfy the following two assumptions:

(1) A named entity consists of contiguous tokens. That is, there is no case where entities are separated by non-entity words.
(2) These linear spans do not overlap with each other. That is, no token in the text can belong to more than one named entity mention.

Named entities that satisfy these assumptions are generally referred to as flat name entities. Correspondingly, the traditional NER task is essentially the flat NER task. The most common method for the flat NER task is to use sequence tagging techniques with a sequence tag scheme, which allows the model to classify individual tokens (i.e., words) and have the tokens combined together to identify named entities. Each token will be assigned with a tag consisting of a position indicator



(a) Flat NER Task

(b) Nested NER Task

Fig. 1. Tag schemes for flat NER and nested NER.

and an entity type, where the position indicator is used to represent the token's role in a named entity.

A common sequence tag scheme is the BIO (Beginning-Inside-Outside) tag scheme, while other schemes such as IO, BIOES, BIES, BMEWO, and BILOU are also widely used. Figure 1(a) illustrates a sentence using the BIO tag scheme for flat NER task, where two named entities are contained in the given sentence. "Erin" labeled "B-PER" and "Harrison" labeled "I-PER" are combined as a unit to identify a person entity "Erin Harrison", while the non-entity words (e.g., visited) are labeled as "O". Formally, the traditional flat NER system learns to produce the corresponding token-level annotation sequence $Y = \{y_1, y_2, \ldots, y_K\}$ for the given sentence $X = \{x_1, x_2, \ldots, x_K\}$ following a sequence tag scheme, where K is the number of the tokens in the sentence. Alternatively, named entities in the sentence can be indicated by a list of tuples. In this example, the tuple <1, 2, PER> points to the person entity "Erin Harrison". In practice, the sequence tag scheme is more commonly used for flat NER than the tuple tag scheme.

2.2 What is Nested NER?

In many practical applications, it is common that the named entities have a nested structure [13, 48]. Specifically, an entity could contain other entities or be a part of other entities, which breaks the second assumption mentioned above. As the example shown in Figure 1(b), the outer entity "the Oakland Zoo" contains an inner entity, i.e., "Oakland". In the AnCora corpus of Spanish and Catalan newspaper text, nearly half of entities are embedded within another entity. Named entities with the nested structure are also prevalent in specific domains. For example, approximately 17% of entities in the GENIA corpus, a biomedical domain corpus labeled with entity categories such as protein and DNA, are embedded. Consequently, the NER task is required for further recognizing named entities with nested structures (i.e., both outer entities and inner entities), rather than the longest outer entity only. The inherent complexity of nested entities makes the nested NER a more challenging task than traditional flat NER.

Consistent with flat NER, sequence tag schemes can also be used for nested NER. Figure 1(b) shows an example of annotated nested named entities using two different tag schemes, namely the sequence tag scheme (e.g., BIO) and the tuple tag scheme. The choice of tag scheme is often closely related to the adopted solution. For instance, the layered-based approach generally applies a sequence tag scheme, while the region-based approach utilizes the tuple tag scheme. Due to the limitations in labeling the nested structures, sequence tag schemes always require multiple corresponding token-level annotation sequences to fully identify all the nested entities in one sentence (three corresponding annotation sequences are illustrated in this example). In contrast, the triples without redundant information are more concise (three corresponding annotation sequences are replaced by three triples in this example). Formally, given a sentence with K tokens, $X = \{x_1, x_2, \ldots, x_K\}$, the output of a nested NER system is a list of tuples $Y = \{< I_1^{head}, I_1^{tail}, t_1 >, < I_2^{head}, I_2^{tail}, t_2 >, \ldots, < I_m^{head}, I_m^{tail}, t_m >\}$, each of which specifies a entity mentioned in the sentence. Here, I_i^{head} and I_i^{tail} are the head index and the tail index of the ith named entity mention, respectively; t_i is the entity category from a predefined category set; m is the number of named entities in the given sentence.

2.3 Nested NER vs. Flat NER

Although the goal of both the nested NER task and the flat NER task is to identify name entities in sentences, the ideas for solving these two tasks are essentially distinct. We further discuss and compare these two tasks in this section.

Generalization. The traditional flat NER task aims to identify named entities from text that satisfy the two assumptions introduced above. In practice, numerous complex named entities break the two strict assumptions, mainly including nested, multi-category, overlapping, and discontinuous named entities. A brief introduction is as follows:

- Nested named entities: One named entity is completely contained by the other.
- Multi-category named entities: An extreme case of nested named entities is one on which a named entity has multiple entity categories.
- Overlapping named entities: Two named entities overlap, but no one is completely contained by the other.
- Discontinuous named entities: The name entity consists of discontinuous tokens.

Among them, the nested named entity is most common in various domains. The nested NER task will further recognize nested named entities at all levels in addition to the flat named entity at the top level. In all nested NER datasets (more details in Sections 2.4 and 2.5), a proportion of entities do not contain other entities. In other words, flat named entities are included in the nested NER datasets. Furthermore, all nested NER approaches theoretically have the ability to simultaneously recognize flat named entities, while most flat NER approaches cannot address the nested problem. Therefore, we consider that the nested NER is more generalized, and the flat NER can be regarded as a simplified task of nested NER.

Independence. Recent studies have demonstrated that the performance of traditional flat NER approaches will dramatically suffer when recognizing nested entities [23, 29]. Agreeing with Finkel and Manning [2009] [13], the problem of ignoring nested structures in most flat NER approaches is due to technological rather than ideological reasons. As shown in Figure 1(a), the flat named entity information contained in a sentence can be easily expressed by an annotation sequence. For this reason, considerable studies treat the flat NER problem as a sequence tagging problem;

Table 1. The Statistics of the Datasets

Corpus	Year	Data Sources	Language	#Tags	URL
Flat NER					
MUC-6	1995	Wall Street Journal	English	7	https://catalog.ldc.upenn.edu/LDC2003T13
MUC-7	1997	New York Time news	English	7	https://catalog.ldc.upenn.edu/LDC2001T02
CoNLL 2003	2003	Reuters news	English, German	4	https://www.clips.uantwerpen.be/conll2003/ner/
JNLPBA	2004	molecular biology	English	5	http://www.geniaproject.org/shared-tasks/bionlp-jnlpba-shared-task-2004
OntoNotes 5.0	2013	news, conversation, weblogs	English, Chinese, Arabic	89	https://catalog.ldc.upenn.edu/LDC2013T19
Nested NER					
ACE 2004	2004	news, conversations	English, Chinese, Arabic	7	https://catalog.ldc.upenn.edu/LDC2005T09
ACE 2005	2005	news, conversations, weblogs	English, Chinese, Arabic	7	https://catalog.ldc.upenn.edu/LDC2006T06
GENIA	2004	molecular biology	English	36	http://www.geniaproject.org/genia-corpus/pos-annotation
AnCora	2007	journalist texts	Spanish, Catalan	6	http://clic.ub.edu/corpus/en/ancora
GermEval 2014	2014	Wikipedia, news	German	4	https://sites.google.com/site/germeval2014ner/data
CNEC 2.0	2016	newspaper texts	Czech	46	https://ufal.mff.cuni.cz/cnec
TAC KBP 2017	2017	newswire, web documents	English, Chinese, Spanish	5	https://tac.nist.gov/2017/index.html
NNE	2019	Wall Street Journal	English	114	https://github.com/nickyringland/nested_named_entities

accordingly, the sequence tagging model is applied to assign one label (e.g., B-Location, I-Person) to each token in the input sentence. In contrast, the nested named entity information contained in a sentence is difficult to represent through a simple annotation sequence (the example in Figure 1(b) contains three annotation sequences), making standard tagging techniques unsuitable for recognizing nested entities. More importantly, we consider that nested entities contain two key properties, namely partly shared entity boundary and token with multiple entity categories. In order to address this issue, the researchers attempt to reduce the dependence on the sequence tagging model and gradually propose various innovative architectures. Extensive experiments have verified that approaches sensitive to the properties of nested entities are obviously superior to the traditional NER models. This reflects the independence of the nested NER task from the traditional NER task. As the most common entity type (besides flat entities), it has practical importance to handle and explore nested NER task independently. Following the independence of nested NER, this article concentrates on the nested NER task and reviews numerous nested NER approaches to enlighten and guide researchers and practitioners in this area.

2.4 Flat NER Datasets

In the following, we summarize major benchmark datasets that are being widely used in flat NER task. Each tagged corpus is a collection of documents that contain annotations of one or more entity categories.

MUC-6, MUC-7 are part of the MUC corpus, which is focused on tasks related to information extraction, starting in 1987. There are in total seven conferences in this series from MUC-1 to MUC-7. In particular, MUC-6 and MUC-7 extend previous versions by means of adding tasks of NER.

CoNLL 2003 is a classic dataset and contains annotations for Reuters news in English and Frankfurter Rundschau in German. The English dataset involves 22,117 sentences corresponding to 302,811 tokens in four entity categories, including Locations, Organizations, Persons, and MISC (Miscellaneous).

JNLPBA dataset is originally from the GENIA corpus (more details in Section 2.5). The entire GENIA dataset is used for training, and new data is annotated for testing. Only the flat top-most entities are present in this dataset.

OntoNotes 5.0 is a corpus of large-scale, accurate, and integrated annotation of multiple levels of the shallow semantic structure in text provided by the OntoNotes project [42]. The dataset comprises various genres in three languages: roughly 1.5 million words of English, 800K of Chinese, and 300K of Arabic. Especially, the English NER data is tagged with a set of 18 proper name entity categories, consisting of 89 sub-categories.

The above-mentioned datasets with their data sources and number of entity categories (i.e., #Tags) are reported in Table 1. The datasets before 2005 were mainly constructed by annotating news articles with a small number of entity categories. Since then, more datasets have been constructed on diverse kinds of text sources, including Wikipedia articles, conversations, and user-generated text (e.g., tweets), while the number of entity categories has increased significantly (e.g., 89 in OntoNotes).

2.5 Nested NER Datasets

In this section, we describe major benchmark datasets that are being widely used in literature for training and eventual evaluation of proposed nested NER techniques.

ACE 2004, ACE 2005 belong to ACE Corpus, which contains entities, relations, and events for English, Chinese and Arabic. In ACE, the entities are limited to the following seven categories: Person, Organization, Facility, Location, GPE (geo-political entity), Weapon, and Vehicle. The entire English dataset of ACE 2004 contains 8,507 sentences with 27,749 entities, and about 40% of the sentences contain nested entities. In terms of the English dataset of ACE 2005, it contains 9,341 sentences with 30,931 entities, and about 35% of the sentences contain nested entities (the max nesting depth is 6).

GENIA corpus is a richly-annotated corpus for bio-text mining that contains 2,000 MEDLINE abstracts. The corpus contains 18,546 sentences corresponding to 55,740 tokens, annotated with five kinds of biological entities (DNA, RNA, Protein, Cell line, and Cell category), 36 fine-grained subcategories, and with parts of speech.

AnCora is annotated with parts of speech, parse trees, semantic roles, and word senses, including Spanish and Catalan portions. The entity categories present are Person, Location, Organization, Date, Number, and Other. The corpus annotators made a distinction between strong and weak entities. They define strong named entities as a word, a number, a date, or a string of words that refer to a single individual entity in the real world. Weak named entities consist of a noun phrase and must contain a strong entity. Weak entities are very prevalent; 47.1% of entities are embedded.

GermEval 2014 is a new German dataset sampled from German Wikipedia and News Corpora. The dataset covers over 31,000 sentences corresponding to over 590,000 tokens and contains 12 classes of named entities: four main classes (Person, Location, Organisation, and Other) with their subclasses, annotated at two levels (inner and outer chunks). The entire dataset contains over 41,000 named entities, about 7.8% of them embedded in other named entities, about 11.8% are derivations (deriv) and about 5.6% are parts of named entities concatenated with other words (part).

CNEC 2.0 is a corpus of 8,993 Czech sentences with manually annotated 35,220 Czech named entities, classified according to a two-level hierarchy of 46 named entities.

TAC KBP 2017 is a part of the TAC Knowledge Base Population (KBP), which is a set of evaluation tracks for populating knowledge bases from unstructured text. All KBP 2017 tasks are in trilingual (English, Chinese, and Spanish). Approximately 500 core documents are annotated with entities, relations, and events (ERE) according to the guidelines for Rich ERE. Five major coarse-grained entity categories are labeled in Rich ERE: Person, GPE, Location, Organization, and Facility. There are no entity subcategories.

NNE is a nested named entity dataset in English that is built on top of the full Wall Street Journal portion of the **Penn Treebank** (**PTB**). The entire dataset comprises 279,795 mentions of 114 entity categories with up to 6 layers of nesting. About 60% top-level entity mentions contain averaging 2.25 structural mentions. In addition, 19,144 out of 260,386 total spans are assigned multiple categories.

Table 1 presents a brief overview of various nested NER datasets over the course of time, along with the data source, languages involved, entity categories, and so on. As shown in Table 1, all nested NER datasets are constructed based on formal documents (e.g., news or science articles) rather than user-generated text. In addition, some researchers have paid attention to the nested

NER problem in minority languages (e.g., CNEC 2.0 in Czech). Among these datasets, ACE 2005 and GENIA datasets are the most widely used in many recent nested NER studies (more details in Table 2).

2.6 Evaluation Metrics

NER models, both flat NER and nested NER, are generally evaluated by comparing their predictions with manual annotations. This section presents a brief introduction to the widely used quantified comparisons, including exact match and relaxed match.

Exact Match Evaluation. The NER task involves two essential steps, namely correctly detecting the entity boundaries and correctly determining their entity categories. Regarding the exact match evaluation, a named entity is considered correctly recognized only if the detected boundaries and categories are both consistent with the manual annotations. Precision (also called positive predictive value) is the fraction of correct entities among the recognized entities, while Recall (known as sensitivity) is the fraction of correct entities that were recognized. Usually, Precision and Recall scores are not discussed in isolation. F-score or F-measure provides a single score that balances both the concerns of Precision and Recall in one number. To be specific, Precision, Recall, and F-score are calculated based on the number of **true positives** (**TP**), **false positives** (**FP**), and **false negatives** (**FN**), which are defined as follows:

- **True Positive** (TP): entities that are recognized by NER and match ground truth.
- **False Positive** (FP): entities that are recognized by NER but do not match ground truth.
- **False Negative** (FN): entities annotated in the ground truth that are not recognized by NER.

Based on this, Precision and Recall are calculated as

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{2}$$

And the traditional F-measure or balanced F-score (F1 score) is the harmonic mean of Precision and Recall,

$$\text{F1} = \frac{2}{\text{Precision}^{-1} + \text{Recall}^{-1}}. \tag{3}$$

The highest possible value of an F1 score is 1.0, indicating perfect Precision and Recall, and the lowest possible value is 0, if either the Precision or the Recall is zero. The widespread used F1 score gives equal importance to Precision and Recall.

Considering that most NER systems need to distinguish multiple entity categories, it is required to evaluate the performance with more than two entity classes (multiclass classification). In this setup, the final score is obtained by macro-averaging (taking all classes as equally important) or micro-averaging (biased by class frequency). Macro-averaged F-score is suitable for accessing a NER system performance overall across the sets of data. On the other hand, micro-averaged F-score can be a useful measure when the dataset varies in size.

Relaxed Match Evaluation. The relaxed match evaluation, also known as MUC evaluation, scores NER systems from two perspectives, namely the ability to predict the correct entity category and the ability to detect entity boundary information. In terms of an entity, a correct category will be recorded if its predicted entity category is the reference (true) category regardless of its detected boundary (as long as there is an overlap); a correct boundary will be recorded regardless of its category assignment. ACE further designs a more complex entity evaluation procedure to consider a few issues like partial match and errors in entity category, subcategory, and class. However, this

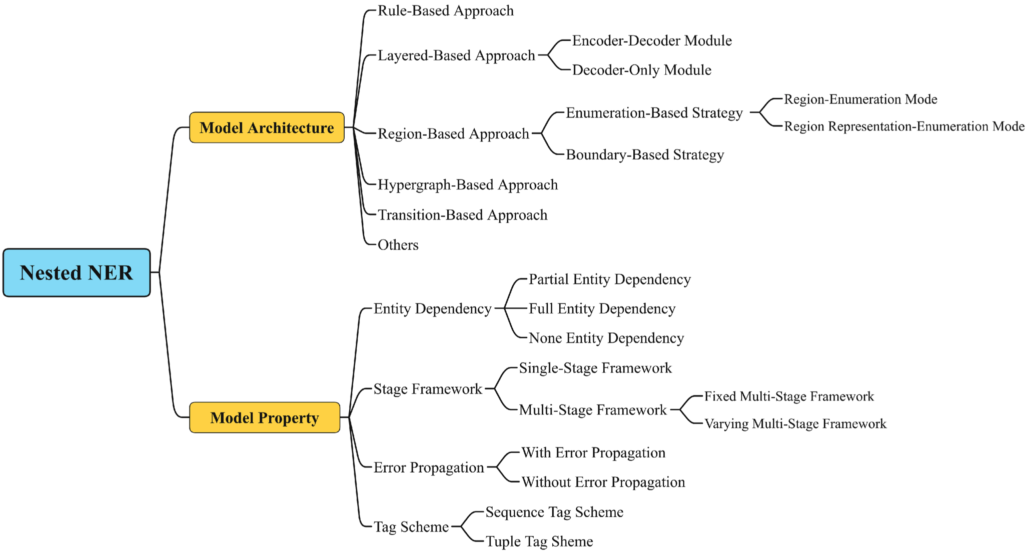


Fig. 2. The categorizations of nested NER from the perspectives of model architecture and model property.

complex evaluation method is not widely used in recent NER studies due to its high complexity and low versatility.

2.7 Related Surveys

Although we present the first comprehensive survey for the nested NER task, there are several survey papers related to the NER task (mainly for flat NER task) so far. Specifically, Nadeau and Sekine [2007] [38] presented the first survey of the NER task. They investigated early research in the NER field (from 1991 to 2006), including handcrafted rule-based algorithms and early machine learning techniques. Ling and Weld [2012] [31] introduced fine-grained entity recognition and formulated the tagging problem as multiclass, multilabel classification. Then Marrero et al. [2013] [36] analyzed the NER task from a theoretical and practical point of view. Saju and Shaja [2017] [44] provided a short survey on extracting named entities from new domains using big data analytics. Moreover, Dai [2018] [7] also provided a short survey on complex named entities, including nested entities, overlapping entities, discontinuous entities, and so on. Yadav and Bethard [2018] [64] presented a short survey of recent advances in NER, especially for the input representation. Goyal et al. [2018] [16] surveyed the development and progress made in the NER task. Li et al. [2020] [25] provided a comprehensive review on existing deep learning techniques for the NER task.

3 OVERVIEW

The survey aims to give readers a comprehensive understanding of nested NER from the perspectives of model architecture and model property. In this section, we briefly outline the two mutually complementary categorizations of nested NER for convenience. In Sections 4 and 5, we will interpret nested NER approaches in more detail from each categorization. Specifically, over 30 representative nested NER approaches are introduced. Figure 2 summarizes the above-mentioned categorizations of nested NER approaches.

All existing nested NER approaches are first introduced according to their model architectures that allows readers to grasp how the approaches work. As shown in Figure 2, nested NER approaches can be categorized into five main streams from the model architecture perspective:

early rule-based, layered-based, region-based, hypergraph-based, and transition-based approaches. Early rule-based nested NER approaches mainly rely on rule-based post-processing. Layered-based approaches commonly treat nested NER task as multiple flat NER task, and thus obey a cascade structure in which basic modules (i.e., the layer) are connected in series. They can be further divided into two subcategories according to the composition of basic modules, that is, encoder-decoder and decoder-only modules. Region-based nested NER approaches transform the nested NER task as a multiclass or multinomial classification task, and finally classify each potential region into one of the numerous classes. Depending on the progressing strategy, there exist two paradigms of region-based approaches, including enumeration-based and boundary-based strategies. Hypergraph-based approaches leverage the hypergraph to represent the nested structure of the entities in the sentence, benefiting from the hyperarcs in the hypergraph to naturally express that tokens belong to different entities. Transition-based nested NER approaches are mainly inspired by transition-based parsers. Such approaches parse a sentence from left to right, building a tree (or a forest) based on greedy decoding one action at a time. The second perspective focuses on the model property to understand the impact of the key properties unique to nested NER approaches and explore the connections between them. Specifically, we categorize existing studies based on whether they preserve entity dependency, belong to the single-stage framework, suffer from error propagation, or require multiple tag schemes. The proposed perspective is summarized in Figure 2. First, entity dependency refers to the dependency between entities (especially nested entities) in the sentence. Capturing entity dependency in sentences will be more effective and in line with human intuition for recognizing entities. Existing approaches have varying degrees of capturing the entity dependency, namely partial, full, and none. Second, different from flat NER approaches, many nested NER approaches decompose the nested NER task into multiple stages or processes (e.g., a two-stage framework including boundary detection stage and category prediction stage). In such settings, apart from single-stage approaches, other multi-stage nested NER approaches present various decomposition strategies for addressing the nesting problem. Third, error propagation is another key property of the nested NER task. Take the transition-based approach as an example, this kind of approach generally leverages the greedy decoding strategy to predict the next action, thereby leading to error propagation. Thus, it is essential to determine nested NER approaches that can avoid this issue and further analyze the possible causes of this issue. Finally, we present the tag scheme used by each approach and find that a small number of approaches use multiple tag schemes for their different targets. We believe that this categorization is beneficial for readers to gain insight into the key properties unique to nested NER approaches, and thus to use these studies as the basis of future investigations into nested NER.

4 MODEL ARCHITECTURE PERSPECTIVE

In this section, we categorize existing nested NER approaches from the first perspective, focusing on the model architecture. Clarifying which specific architectures nested NER approaches leveraged to learn and express the nested structure from input sentences allows readers to grasp how the approaches work. In this section, over 30 representative nested NER approaches are introduced in proper order according to this perspective. Additionally, we summarize these nested NER approaches (except for rule-based approaches) in Table 2 according to the model architecture and list the input representation, feature encoder, tag decoder, and the performance in F1-score on a few representative benchmark datasets.

4.1 Early Rule-Based Approach

Early approaches for addressing nested NER rely on hand-craft rules and rule-based post-processing. Shen et al. [2003] [46] presented the first rule-based solution to recognize biomedical

Table 2. Summary of Recent Work on Nested NER from the Model Architecture Perspective

Model	Approach Type	Input Representation	Feature Encoder	Tag Decoder	F1 Score
Zhang et al. [2004] [66]	Layered-based	Randomly	Feature-based	HMM	GENIA:66.5%
Alex et al. [2007] [1]	Layered-based	Randomly	Feature-based	CRF	GENIA:67.9%
Ju et al. [2018] [22]	Layered-based	Word2vec	BiLSTM	CRF	ACE2005:72.2%
Fisher and Vlachos [2019] [14]	Layered-based	Glove/ELMo/BERT	Self-designed Network	Self-designed Network	ACE2005:74.6%/78.9%/82.4%
Fei et al. [2020] [11]	Layered-based	Glove	BiLSTM, Attention	CRF	ACE2005:75.7%
Wang et al. [2020] [58]	Layered-based	Glove/BERT+ALBERT	BiLSTM	LSTM-CNN-Linear	ACE2005:79.4%/86.3%
Shibuya and Hovy [2020] [47]	Layered-based	Glove/+BERT/+BERT+Flair	BiLSTM	CRF	ACE2005:76.8%/84.0%/84.3%
Byrne [2007] [4]	Region-based	Randomly	Feature-based	CandC classifier	Private corpus:77.3%
Xu et al. [2017] [63]	Region-based	FOFE	FFNN	FFNN	KBP2015:72.2%
Sohrab and Miwa [2018] [49]	Region-based	Word2vec	BiLSTM	FFNN(ReLU)-Softmax	GENIA:77.1%
Lin et al. [2019] [30]	Region-based	SENNA/BERT	BiLSTM, Attention	FFNN-sigmoid	ACE2005:75.2%/80.1%
Xia et al. [2019] [61]	Region-based	Glove+ELMo	BiLSTM, Self-Attention	FFNN-Softmax	ACE2005:78.2%
Zheng et al. [2019] [68]	Region-based	Word2vec	BiLSTM	FFNN(ReLU)-Softmax	GENIA:74.7%
Ouchi et al. [2020] [40]	Region-based	Glove/BERT	BiLSTM	-	GENIA:74.2%/73.9%
Sun et al. [2020] [52]	Region-based	Glove/BERT	Self-Attention, Deep Residual CNN	FFNN-Softmax	ACE2005:77.5%/86.9%
Long et al. [2020] [32]	Region-based	Glove/BERT	BiLSTM	FFNN-Softmax	ACE2005:76.5%/84.6%
Tan et al. [2020] [53]	Region-based	Glove/BERT	BiLSTM	FFNN-Softmax	ACE2005:75.6%/83.9%
Yu et al. [2020] [65]	Region-based	fastText+BERT	BiLSTM	Biaffine-Softmax	ACE2005:85.4%
Wang et al. [2020] [59]	Region-based	Word2vec	BiLSTM, Self-Attention	Biaffine-Softmax, CRF	GENIA:76.2%
Lu and Roth [2015] [33]	Hypergraph-based	Randomly	Feature-based	Mention Hypergraphs	ACE2005:62.5%
Wang and Lu [2018] [56]	Hypergraph-based	Glove	BiLSTM	Segmental Hypergraphs	ACE2005:74.5%
Katiyar and Cardie [2018] [23]	Hypergraph-based	Word2vec	BiLSTM	LSTM-Softmax	ACE2005:70.2%
Finkel and Manning [2009] [13]	Transition-based	Randomly	Feature-based	CRF-CFG parser	GENIA:70.3%
Wang et al. [2018] [57]	Transition-based	Glove	BiLSTM, Stack-LSTM	FFNN-Softmax	ACE2005:73.0%
Marinho et al. [2019] [35]	Transition-based	Word2vec	LSTM	FFNN-Softmax	GENIA:73.0%
Gu [2006] [18]	Other	Randomly	Feature-based	SVM	–
Muis and Lu [2017] [37]	Other	Randomly	Feature-based	Mention Separator	ACE2005:61.3%
Straková et al. [2019] [50]	Other	Word2vec/+BERT+Flair	BiLSTM	CRF, LSTM-Softmax	ACE2005:75.4%/84.3%
Lin et al. [2019] [29]	Other	Glove	BiLSTM, CNN	FFNN-Softmax	ACE2005:74.9%
Li et al. [2020] [27]	Other	BERT	-	Softmax, Sigmoid	ACE2005:86.9%
Luo and Zhao [2020] [34]	Other	Glove	BiLSTM, Bi-GCN	CRF	ACE2005:75.1%

nested entities (referred to as cascaded entities in their article). Their approach contains four basic patterns corresponding to types of nested entities, namely (1) ⟨entity1⟩ head noun → ⟨entity2⟩, (2) ⟨entity1⟩ ⟨entity2⟩ → ⟨entity3⟩, (3) modifier ⟨entity1⟩ → ⟨entity2⟩, and (4) ⟨entity1⟩ word ⟨entity2⟩ → ⟨entity3⟩. They also leveraged the **Hidden Markov Models** (**HMMs**) [10] and integrated various features, including simple deterministic features, morphological features, POS features, and semantic trigger features, to recognize flat entities. HMM is a generative statistical model that assigns a probable target sequence to each input sentence following the Viterbi algorithm [55]. HMM is able to capture the locality of phenomena, thereby improving its evaluation performance. Subsequently, Zhou et al. [2004] [70] and Zhou [2006] [69] proposed similar approaches based on the above rules. To be specific, Zhou et al. [2004] [70] also presented a HMM-based NER model, while enhancing the rule-based post-processing to automatically extract rules from training data for nested NER task. From the GENIA corpus, they observed six useful patterns (two extend patterns were added except the above four patterns) of nested entity name constructions. Zhou [2006] [69] further proposed a NER approach called **Mutual Information Independence Model** (**MIIM**) in the biomedical domain. In MIIM, he also presented the rule-based post-processing for addressing nested entities, while leveraging the same patterns mentioned above.

4.2 Layered-Based Approach

The layered-based approach is an intuitive solution for nested NER. These models generally contain multiple layers (or levels) according to the hierarchical nature of the structure in nested named entities. Each layer is used to identify a group of named entities, which can be entities at a certain level or with a certain length. Since the layer module is the basic unit in layered-based approaches, we divide them into two subcategories according to the composition of the layer module, including (1) encoder-decoder module and (2) decoder-only module. Specifically, the encoder-decoder module refers to the basic unit containing a feature encoder and a tag decoder. The decoder-only module refers to the basic unit only containing a tag decoder, which means that these approaches only encode the sentence once before tag decoding. We present these two representative layered-based architectures in Figures 3 and 4, respectively.

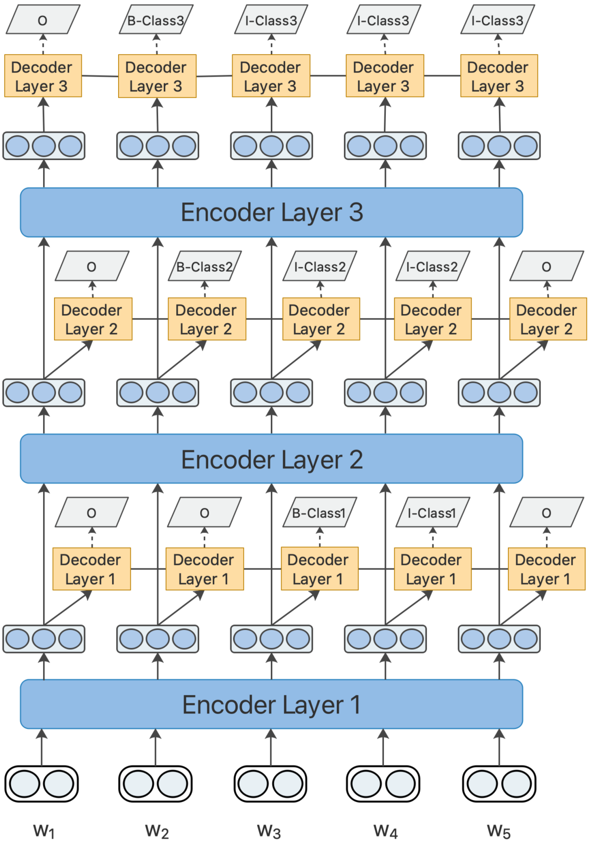


Fig. 3. The first representative layered-based architecture (i.e., encoder-decoder module), where each layer contains one encoder layer and one decoder layer. The encoders in higher layer can obtain entity information from the encoders in lower layers.

4.2.1 Encoder-Decoder Module. The architecture of layered-based approaches with the encoder-decoder module is depicted in Figure 3. Zhang et al. [2004] [66] proposed a HMM-based approach with the layered structure to recognize named entities with nested structures. Specifically, two HMM models need to be trained in their model, one of which is used to recognize short embedded entities, and another one is to iteratively extend short entities. Alex et al. [2007] [1] introduced three models based on **Conditional Random Fields** (**CRFs**) [24], which can reduce the nested NER problem into one or more sequence tagging problems. Different from the HMM, the CRF is a probabilistic graphical model which can easily work with a vast amount of non-independent features [24]. They separately built inside-out and outside-in layered CRFs for addressing nested NER, both of which can use the current guesses as to the input to the next layer. They also cascaded separate CRFs of each entity category by using output from the previous CRFs as features of the subsequent CRFs, yielding the best performance in their work. These two approaches are instructive for the follow-up layered-based researches. In particular, recognizing nested entities in an inside-out manner is widely adopted in existing studies. Ju et al. [2018] [22] proposed the first neural layered model to identify nested entities by dynamically stacking flat NER layers in the inside-out manner. Specifically, each flat NER layer is a simple sequence tagging model that contains one **Bidirectional Long Short-Term Memory** (**BiLSTM**) [41] encoder and one CRF

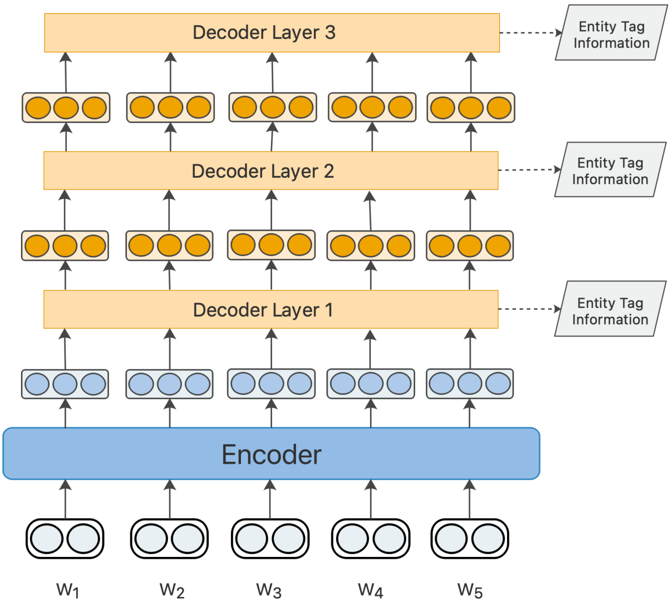


Fig. 4. Another representative layered-based architecture (i.e., decoder-only module). In this architecture, we only have one shared encoder layer, and multiple decoder layers to capture entities from different layers.

decoder. LSTM [21] is a variant of **Recurrent Neural Networks (RNNs)** [15] that incorporates a memory cell to remember the past information for a long period of time. Their model merges the output of the encoder in the current flat NER layer to construct new representations for detected entities and subsequently feeds the new representations into the next flat NER layer. This strategy allows their model to identify outer entities by leveraging the knowledge of their corresponding inner entities. The number of stacked layers depends on the level of entity nesting from the input sequences. Similar to the work of Ju et al. [2018] [22], Fei et al. [2020] [11] proposed a dispatched attention model with multitask learning for nested NER, where each task recognizes entities at a corresponding nesting level. Their approach first employs a BiLSTM to encode the given sentence into a sequence representation. Beyond that, each layer module (referred to as each task in their article) also contains one position- and syntax-aware attention-based encoder and one CRF decoder. The attention mechanism [2] leveraged by the encoder can dynamically decide how much information to use from a word or character level component. They further designed the dispatched attention mechanism, which can transfer knowledge in the inside-out order, to capture information across layers (i.e., tasks). Alternatively, Fisher and Vlachos [2019] [14] introduced another novel neural model that first merges tokens and/or entities into entities forming nested structures, and then labels each of them independently. Their approach can smoothly merge token embeddings in a sentence into entity embeddings across multiple levels. Moreover, these entity embeddings are used to label the entities identified. The above-mentioned three neural approaches all first identify inner entities, and then further identify outer entities in the inside-out manner.

4.2.2 Decoder-Only Module. The architecture of layered-based approaches with the decoder-only module is shown in Figure 4. Wang et al. [2020] [58] presented a neural layered model, called Pyramid, to handle nested NER task in an inside-out manner. In particular, the Pyramid model consists of a stack of interconnected layers, each of which predicts whether a text region of a certain length is an entity. Besides the normal pyramid, they additionally designed an inverse pyramid

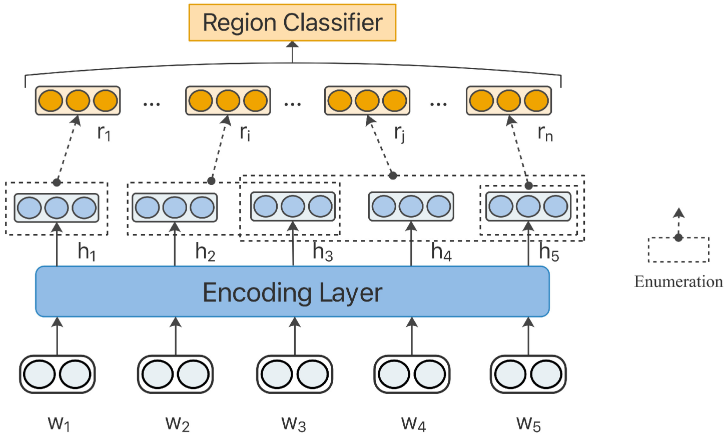


Fig. 5. The first representative region-based architecture (i.e., enumeration-based strategy). The model with this architecture learns representations of all enumerated regions from the input sentence, and then classifies them into corresponding entity categories. The dotted line represents the enumeration process. It is worth mentioning that several studies enumerate regions before encoding sentences.

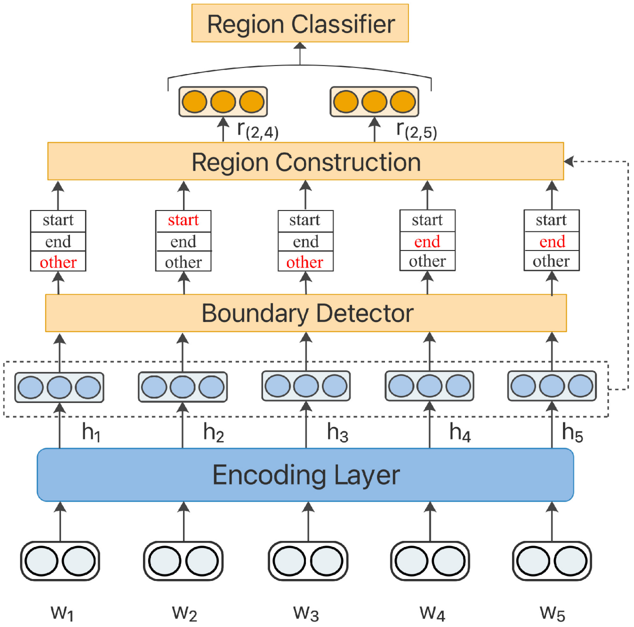


Fig. 6. Another representative region-based architecture (i.e., boundary-based strategy). The model with this architecture establishes representations of candidate regions (possibly entities) by leveraging boundary information, and subsequently accomplishes the entity classification.

to allow bidirectional interactions between adjacent layers. In this way, their model will recognize named entities by their length, without layer disorientation and error propagation. Unlike the previous inside-out approaches, Shibuya and Hovy [2020] [47] proposed a CRF-based decoding approach that can iteratively recognize entities in an outside-in manner (i.e., from outermost ones to inner ones) without structural ambiguity. First, they encoded the input sentence based on the BiLSTM model and leveraged the output from the last hidden layer to represent tokens in the sentence. And then, they constructed a separate CRF for each named entity category decoding and extracted outermost entities and inner entities without re-encoding. For each entity category, the corresponding CRF first decodes a tag sequence over the entire sentence to extract outermost entities (called the best path in their article). Their model further recursively extracts inner entities (called the 2nd best path in their article) based on the previously extracted entities, until no multi-token entities are detected in each region.

4.3 Region-Based Approach

The region-based approaches generally treat the nested NER task as a multiclass classification problem, and various strategies have been designed to obtain the representation of potential regions (i.e., subsequences) before classification. We mainly divide existing region-based approaches into two categories based on the progressing strategy, including (1) enumeration-based strategy and (2) boundary-based strategy. To be specific, the enumeration-based strategy refers to the region-based approach that learns representations of all enumerated regions from the input sentence, and then classifies them into corresponding entity categories. The boundary-based strategy refers to the region-based approach that establishes representations of candidate regions (possibly entities) by leveraging boundary information, and subsequently accomplishes the entity classification. We respectively present the architectures of two progressing strategies in Figures 5 and 6.

4.3.1 Enumeration-Based Strategy. As shown in Figure 5, some region-based approaches leverage enumeration manner to obtain representations of all regions before region classification. Specifically, a small portion of region-based approaches with enumeration strategy first enumerate all regions from sentence, and then learn their corresponding region representations. However, more studies suggest to first encode the input sentence to contextual token-level representations, and then utilize token representations to enumerate all region representations. Therefore, we further classify approaches based on enumeration strategy on the basis of their enumeration objects, including (1) region-enumeration mode and (2) region representation-enumeration mode.

Region-Enumeration Mode. In this mode, all approaches first enumerate all regions from the input sentence, and then learn their corresponding region representations. Byrne [2007] [4] first presented such an enumeration-based nested NER model, which transforms the input sentence to potential entity subsequences by concatenating adjacent tokens. According to the analysis of the distribution of entity string lengths in this work, 97.10% of entities contain only 6 tokens or fewer, though the longest is 25 tokens. Based on this, he chose the maximum entity length to 6 and labeled each concatenated subsequence using the CandC tagger [6]. In recent years, Xu et al. [2017] [63] also proposed a neural nested NER approach by local detection. Similar to Byrne [2007] [4], Xu et al. [2017] [63] enumerated all regions (up to a certain length) in a sentence at first. The major difference in this work is that each region (or called word segment in their article) will be determined whether it is a valid named entity individually based on the region itself and its left and right contexts in the sentence. Since the region and its context are of varying lengths in natural languages, they proposed to use the **fixed-size ordinally-forgetting encoding** (**FOFE**) method [67] to fully encode them. The FOFE method can uniquely and losslessly encode a variable-length sequence of words into a fixed-size representation, so that easily be input to a feedforward neural network.

Region Representation-Enumeration Mode. In this mode, all approaches first encode the input sentence to a token-level representation sequence, and then enumerate all region representations from the sequence. Some of these approaches utilize additional features (e.g., gazetteer feature) or multiple feature extractors to improve the overall performance. Sohrab and Miwa [2018] [49] proposed a neural exhaustive model for the nested NER, where they first encoded the input sentence to token representations by BiLSTM, and each region representation is obtained based on the token representations belonging to this region. The number of possible regions depends on the predefined maximum entity size. Specifically, the strategy of representing the region is to average the region inside (token) representations and concatenate it with region boundary (token) representations. Finally, region representations are fed to an activation layer and a softmax output layer to classify each region into a specific entity category or non-entity. Moreover, Lin et al. [2019] [30] proposed the **Gazetteer-Enhanced Attentive Neural Network** (**GEANN**) that models both the candidate region and its corresponding contextual information. They followed the work of Sohrab and Miwa [2018] [49] to obtain the region representation for each region. Then, they designed an attentive context encoder that outputs the region corresponding contextual representation to explicitly model the association between a region and its context. Finally, the region representation concatenated with its contextual representation is fed into a **Multi-Layer Perceptron** (**MLP**) classifier to accomplish the entity prediction. Furthermore, they devised an auxiliary gazetteer network to learn the name knowledge in a context-free way (i.e., only using gazetteer). Ouchi et al. [2020] [40] proposed an instance-based nested NER approach, which treats the NER as a multiclass classification problem. This approach also needs to enumerate all region representations from the contextual encoding sequence first, and then assign a category label to each of them. They further built a feature space in which regions with the same category label are close to each other. At inference time, each region is assigned a category label based on its similar regions in the training set, where it is easy to understand how much each training instance contributes to the predictions. Sun et al. [2020] [52] proposed an end-to-end region-based model TCSF, which can jointly learn the token context and region feature in sentences. Similar to the above approaches, they enumerated all region representations from the contextual token-level sequence. Inspired by the Transformer [54], they further designed a region relation network that models the region set (i.e., all region representations of the sentence) to produce a relation representation for each region. Recently, some studies exploit the region filter to drop out some regions that are unlikely to be entities before region classification. For example, Xia et al. [2019] [61] proposed a multi-grained NER model called MGNER, which needs to detect all possible regions before categorizing regions based on context. The MGNER consists of two sub-networks: a detector and a classifier. The detector detects all the possible entity regions (i.e., discards regions that will not be entities) after the sequential feature extractor, while the classifier aims to classify detected possible entity regions into predefined entity categories. In addition, contextual information and a self-attention [54] are utilized throughout the framework to improve the NER performance. The most benefit of the self-attention mechanism is that it can associate different positions of a single sequence in order to compute a representation of the same sequence. Long et al. [2020] [32] further introduced a hierarchical region learning framework to automatically generate a tree hierarchy of candidate regions and incorporate structure information into the region representation for better classification. They defined the hierarchical region generation based on the word coherence measure, while the higher this measure, the more coherent adjacent words.

4.3.2 *Boundary-Based Strategy.*

Instead of enumerating all regions from sentence, many studies suggest that the candidate boundary (which may belong to the entity) should be detected first, and then the learning of the region representation can use the obtained boundary information. As shown in Figure 6, the most benefit of this boundary-based strategy is that we no longer need to enumerate all regions in a sentence. Zheng et al. [2019] [68] proposed the first boundary-aware neural model for nested NER. Their model precisely locates entities by detecting candidate boundaries using sequence tagging models. Based on the candidate boundaries, their model utilizes the boundary-relevant regions to predict entity category labels. Subsequently, Tan et al. [2020] [53] presented another nested NER approach using a different boundary detection manner. After

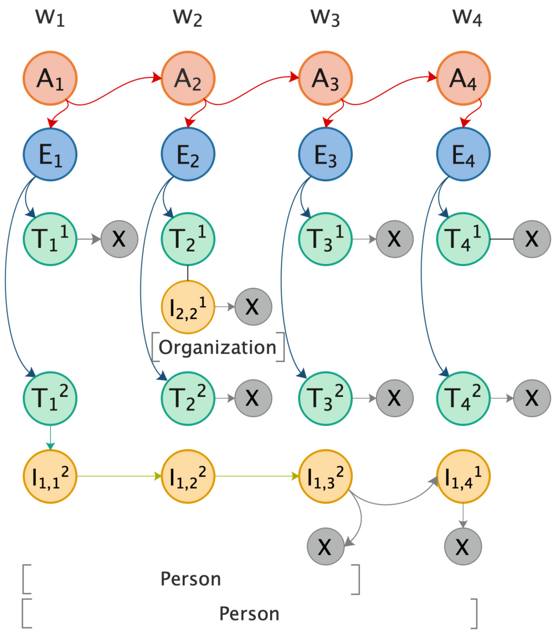


Fig. 7. A hypergraph model structure [56]. This model leverages hyperarcs to express the nested structure. A_i encodes all such entities that start with the ith or a later token (word). E_i encodes all entities that start exactly with the ith token. T_i^k represents all entities of category k starting with the ith token. $I_{i,j}^k$ represents all entities of category k that contain the jth token and start with the ith token. X marks the end of an entity.

encoding the input sentence, they fed the contextual representation into two token-wise classifiers, which will predict each token whether it is the first or the end token of an entity. Finally, they summarized the region representation according to the results of boundary detection and predicted its entity category label. Recently, Yu et al. [2020] [65] used ideas from graph-based dependency parsing to recognize named entities. Specifically, they first leveraged the BiLSTM to obtain the contextual representation, and then applied two separate MLPs to create the start/end representations. Their model utilizes the pair of start/end representations (i.e., boundary information) as the corresponding region representation, and simply employs a biaffine model to predict named entities in the sentence. Furthermore, Wang et al. [2020] [59] specially introduced a biaffine-based head-tail detector to determine whether each pair of tokens in the sentence is the boundary of an entity. In addition, they further constructed a token interaction tagger to characterize the internal token connection within the head-tail pair. Finally, a region classifier is employed to integrate the head-tail detector and token interaction tagger for entity recognition.

4.4 Hypergraph-Based Approach

The hypergraph-based approaches leverage the hypergraph structure to express the nested structure, and the hyperarc in a hypergraph is able to precisely tag the token in one sentence that belongs to different named entities. Lu and Roth [2015] [33] firstly presented a novel directed hypergraph (referred to as mention hypergraph) for both boundary detection and category prediction. The proposed hypergraph structure consists of five types of nodes, which are used to compactly represent many entities of different semantic categories and boundaries. Moreover, the hyperarc will naturally address the nested structure problem as it can connect two or more nodes. These paths altogether form a unique sub-hypergraph of the original hypergraph to express all nested

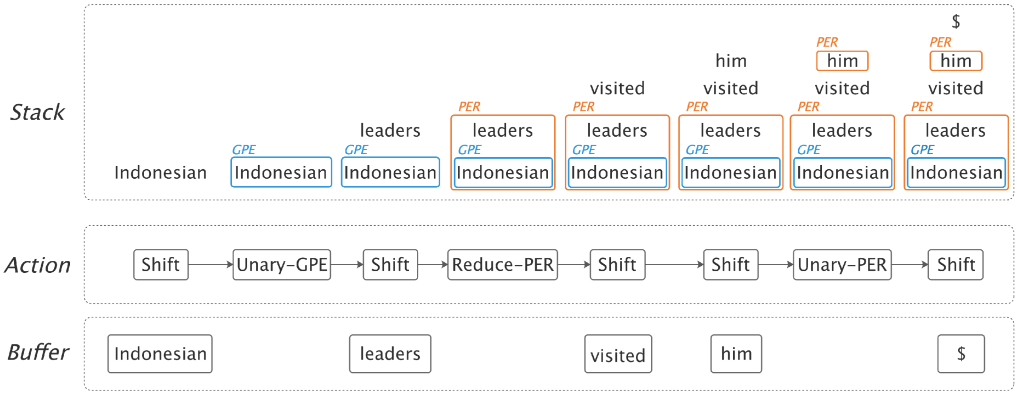


Fig. 8. The structure of a transition-based model [57]. In this example, we artificially design a stack set, an action set, and a buffer set, and then parse an input sentence "Indonesian leaders visited him" from left to right, where "$" denotes the end of sentence.

entities in a sentence. In order to address the structural ambiguity issue of Lu and Roth [2015] [33], Wang and Lu [2018] [56] proposed a neural segmental hypergraph model using neural networks to obtain distributed feature representations. We present their hypergraph structure in Figure 7. Their model has a $O(cmn)$ time complexity (m is the number of entity categories, n is the number of words in a sentence, and c is the maximal number of words for each entity). Particularly, the segmental feature is introduced into hypergraph for addressing the structural ambiguity issue [33]. Different from the strategy of constructing the hypergraph in the above two approaches, Katiyar and Cardie [2018] [23] introduced another hypergraph structure based on the BILOU tag scheme (an extended version of BIO tag scheme), and proposed a standard LSTM-based sequence tagging model to learn a hypergraph representation for nested entities in the input sentence. In their hypergraph structure, each node is the BILOU tag corresponding to the input token. There are two types of directed edges in this hypergraph, including simple edges for which there is only one head node for every tail node and hyperarcs that connect more than one head node to a tail node. They treated the hypergraph construction procedure as a multilabel assignment process, and greedily decoded the hypergraph from left to right to find the most likely sub-hypergraph.

4.5 Transition-Based Approach

Transition-based nested NER approaches are mainly inspired by transition-based dependency parsers [39]. As shown in Figure 8, such approaches parse a sentence from left to right, building a tree based on greedy decoding one action at a time. Finkel and Manning [2009] [13] proposed a discriminative constituent parser for recognizing nested named entities. The parser aims to extract a constituency-based parsing tree from a sentence that represents its nested structure. In addition, the parser is similar to the chart-based PCFG parser [12], except that instead of putting probabilities over rules, it puts clique potentials over local subtrees. Recently, Wang et al. [2018] [57] introduced a neural transition-based approach to model the nested structure of entities. They first mapped the sentence with nested entities to the forest structure, where each entity corresponds to a constituent of the forest. Their system then learns to construct the forest structure in a bottom-up manner through an action sequence. To be specific, the system consists of three types of transition action (i.e., SHIFT, REDUCE, UNARY) and employs a stack to partially store processed nested elements. The system's state is defined as $[S, i, A]$ that denotes stack, buffer front index, and action history, respectively. In each step, one of three types of actions will be applied to change the system's state. In addition, Marinho et al. [2019] [35] introduced another neural transition-based approach, the **Hierarchical and Nested Named Entity Recognition** (**HNNER**) model, to handle different levels of nested entities. Specifically, they designed the transition system based on a word stack, a word buffer, a mention stack, and an output buffer. Moreover, they considered four types of possible system actions, including OUT, SHIFT, TRANSITION, and REDUCE. They further proposed a set of modifier classes that introduce certain concepts that change the meaning of an entity, such as absence, or uncertainty about a given entity.

4.6 Other Approaches

Apart from the above-mentioned approaches, there are a few distinctive and particular studies and explorations on the nested NER, and it is difficult to classify them into the five types described above. In earlier studies, Gu [2006] [18] treated the nested NER task as a binary classification problem and solved it using **Support Vector Machine** (**SVM**) [5]. The SVM is a widely used supervised model that can perform well for handling high dimensional input space problem in many text mining tasks. In this solution, he used two tag schemes to set the class label for each token, including outmost labeling and inner labeling. Muis and Lu [2017] [37] developed a gap-based tag schema to capture nested structures of entities. They defined the mention separators to represent nested named entities. At each gap between two words, they considered eight possible types of mention separators based on the combination of three cases, including (1) the entity is starting at the next word (S), (2) the entity is ending at the previous word (E), and (3) the entity is continuing to the next word (C). Since each mention separators sequence can only present the entity information in the same entity category, they supported multiple entity categories by using multiple sequences, one for each entity category. Strakova et al. [2019] [50] proposed two neural network architectures for nested NER. For their models, they need to concatenate the nested entity multiple labels into one multilabel. The first model predicts the label of each token with a standard LSTM-CRF model. In the second model, the nested NER task will be viewed as a sequence-to-sequence task, in which the input sequence is the tokens and the output sequence is the labels. The decoder predicts labels for each token, until a special label "$\langle eow \rangle$" (end of the word) is predicted and the decoder moves to the next token. Lin et al. [2019] [29] proposed **Anchor-Region Networks** (**ARNs**), a sequence-to-nuggets architecture for recognizing nested entities. The ARNs first identifies anchor words (i.e., possible head words) of all named entities, and then recognizes the entity boundaries for each anchor word by exploiting regular phrase structures. Furthermore, they also designed Bag Loss, an objective function that can train ARNs in an end-to-end manner without using any anchor word annotation. Li et al. [2020] [27] proposed to formulate the NER task as a **Machine Reading Comprehension** (**MRC**) task. This formulation naturally tackles the nested structure problem. Specifically, they transformed the tagging-style annotated NER dataset to a set of {QUESTION, ANSWER, CONTEXT} tuples. For the question generation, they used a template-based procedure, which can take annotation guideline notes as references. Furthermore, they designed and adopted one region selection strategy for the answer representation. They first used two binary classifiers, one to predict whether each token is the start index or not, the other to predict whether each token is the end index or not. They further used another binary classification model to match each predicted start index with its corresponding end index since there could be multiple entities of the same category. Luo and Zhao [2020] [34] proposed a novel bipartite flat-graph (BiFlaG) network for nested NER, which contains a flat NER module and a graph module. The BiFlaG can jointly learn flat entities and effectively captures the bidirectional interaction (inside-out and outside-in directions) between them. Their model first uses the flat NER module to recognize the outermost entities and construct entity graphs. And then, each entity graph is fed to the subsequent graph module, which aims to effectively learn the dependencies of inner entities and improve outermost entity prediction. From these approaches, we can observe that most of them (e.g., Gu [2006] [18], Muis and Lu [2017] [37], Strakova et al. [2019] [50], Li et al. [2020] [27]) need to design unique tag schemes to facilitate the model to capture the nested structure of entities. Alternatively, some models additionally consider some auxiliary knowledge (e.g., anchor words, graph structure) to solve the nesting problem of entities.

4.7 Summary and Discussion

As shown in Table 2, we summarize all nested NER approaches according to the model architecture, along with their respective input representation, encoder-decoder, and performance. Apart from early rule-based approaches, the other four main model architectures have achieved great success in nested NER task. We sum up these four types of approaches in general, as follows. The layered-based approach is inspired by the sequence tagging approaches and identifies entities in the sentence layer by layer. The region-based approach also decomposes the nested NER, that is, produces all possible regions in the sentence and classifies them independently. The hypergraph-based approach presents a novel data structure to express the nested structure from the input sentence, which regards identifying the entities in the sentence as finding the best sub-hypergraph in the complete mention hypergraph. The transition-based approach is mainly inspired by the transition-based parsing and treats the nested NER task as a sequential action decision process. Furthermore, we infer the following three observations based on Table 2.

First, deep neural models have already been widely used by the existing approaches to address the nested NER task. Earlier approaches from all four model architectures are normally based on feature engineering and machine learning algorithms. Recently, deep neural models have become dominant and have various usages depending on different architectures. For example, as we can be seen from Table 2, many layered-based approaches only leverage the neural model as the sequential feature extractor (i.e., encoder). In the case of region-based, it is also feasible to build a neural-based classifier (i.e., decoder) to predict which entity category the region belongs to.

Second, we can observe that approaches from all four model architectures can achieve state-of-the-art performance by comparing all existing approaches. Thus, it is hard to determine which architecture is the optimal solution and most appropriate for nested NER task. Furthermore, we observe that a few of the existing approaches incorporate the characteristics from more than one model architecture. For example, Wang et al. [2020] [58] presented a layered model, and each model layer enumerates all regions of a certain length and predicts the entity category for them. Long et al. [2020] [32] proposed a region-based model, while the hierarchical region generation is from bottom to up recursively (i.e., similar to the layered-based approach). In our opinion, future efforts can be spared to explore how to fuse the existing model architectures.

Third, recent nested NER studies mainly introduce pre-trained language models as (or as a supplement to) the input representation layer of their model. The overall experimental results of various nested NER approaches shown in Table 2 demonstrate that the pre-trained language model embedding can extremely improve the performance of the proposed approaches, especially on the ACE 2005 dataset. Pre-trained language model representation has become one of the most effective auxiliary means to the nested NER task. With the tremendous growth of representation learning, pre-trained language models have been used in a wide range of NLP fields with huge success [9, 28, 62].

5 MODEL PROPERTY PERSPECTIVE

From this section, we illustrate the above-mentioned nested NER approaches around the second perspective, the model property. We analyze various key properties to nested NER approaches conscientiously, and further explore the connections between them.

Table 3. Summary of Recent Work on Nested NER from the Model Property Perspective

Model	Entity Dependency	Stage Framework	Error Propagation	Tag Scheme
Layered-based Approach				
Zhang et al. [2004] [66]	None	Two-stage	None	Sequence
Alex et al. [2007] [1]	Partial (Inside-out/Outside-in)	Varying multi-stage	Yes	Sequence
Ju et al. [2018] [22]	Partial (Inside-out)	Varying multi-stage	Yes	Sequence
Fisher and Vlachos [2019] [14]	Partial (Inside-out)	Two-stage	Yes	Sequence
Fei et al. [2020] [11]	Partial (Inside-out)	Varying multi-stage	Yes	Sequence
Wang et al. [2020] [58]	None	Varying multi-stage	None	Sequence
Shibuya and Hovy [2020] [47]	Partial (Outside-in on each category)	Varying multi-stage	Yes	Sequence
Region-based Approach				
Byrne [2007] [4]	None	Single-stage	None	Tuple
Xu et al. [2017] [63]	None	Single-stage	None	Tuple
Sohrab and Miwa [2018] [49]	None	Single-stage	None	Sequence
Lin et al. [2019] [30]	None	Single-stage	None	Tuple
Xia et al. [2019] [61]	None	Two-stage	Yes	Tuple
Zheng et al. [2019] [68]	None	Two-stage	Yes	Sequence+Tuple
Ouchi et al. [2020] [40]	None	Single-stage	None	Tuple
Sun et al. [2020] [52]	Full	Single-stage	None	Tuple
Long et al. [2020] [32]	Partial (Inside-out)	Two-stage	Yes	Tuple
Tan et al. [2020] [53]	None	Two-stage	Yes	Tuple
Yu et al. [2020] [65]	None	Single-stage	None	Tuple
Wang et al. [2020] [59]	None	Three-stage	Yes	Sequence+Tuple
Hypergraph-based Approach				
Lu and Roth [2015] [33]	Full	Single-stage	None	Tuple
Wang and Lu [2018] [56]	Full	Single-stage	None	Tuple
Katiyar and Cardie [2018] [23]	Partial (Left-to-right)	Single-stage	None	Sequence
Transition-based Approach				
Finkel and Manning [2009] [13]	Partial (Left-to-right)	Single-stage	Yes	Sequence
Wang et al. [2018] [57]	Partial (Left-to-right)	Single-stage	Yes	Sequence
Marinho et al. [2019] [35]	Partial (Left-to-right)	Single-stage	Yes	Sequence
Other Approaches				
Gu [2006] [18]	None	Two-stage	Yes	Tuple
Muis and Lu [2017] [37]	Partial (Global on each category)	Varying multi-stage	None	Sequence
Straková et al. [2019] [50]	None	Single-stage	Yes	Sequence
Lin et al. [2019] [29]	None	Two-stage	Yes	Tuple
Li et al. [2020] [27]	None	Varying multi-stage	None	Tuple
Luo and Zhao [2020] [34]	Partial (Outside-in)	Two-stage	Yes	Sequence+Tuple

5.1 Entity Dependency

Entity dependency refers to the dependency between different entities in a sentence. In particular, the entity may be included in another entity in the nested entity setting. Take "A spokesman for the Oakland Zoo said ..." as an example, the outer entity "the Oakland Zoo" contains an inner entity "Oakland". It is effective for identifying entities to leverage such dependence between different entities, especially true for entities with nested structure. The targeted learning of the model to the entity dependency is consistent with the process of human entity recognition. From Table 3, we can observe that many existing approaches have the ability to capture the entity dependency. We categorize the existing approaches depending on the varying degrees of captured entity dependency, as follows.

Partial Entity Dependency. For nested entities, the dependencies between entities are bidirectional, that is, from the inner entity to the outer entity (i.e., inside-out direction) and from the outer entity to the inner entity (i.e., outside-in direction). Most current approaches can only capture and transfer knowledge about entity dependency in one direction. To be specific, in terms of layered-based approaches, most of them can transfer the learned entity dependency in a certain direction. For example, the proposed models of Ju et al. [2018] [22] and Fisher and Vlachos [2019] [14] can learn entity dependency in inside-out direction, while the proposed models of Shibuya and Hovy [2020] [47] and Luo and Zhao [2020] [34] can learn entity dependency in out-inside direction. Besides the layered-based approaches, the transition-based approaches [13, 35, 57] can utilize knowledge of the inner entity in predicting the outer entity (i.e., inside-out direction), because they parse the sentence (i.e., the sequential decision of transitions) from left to right.

Full Entity Dependency. Only a few of the existing approaches can fully consider the dependence between entities in sentences. The representative approaches are hypergraph-based approaches [33, 56] that can naturally consider all the entities in a sentence by using a global optimization strategy. For example, Lu and Roth [2015] [33] can represent each of model outputs with a single fully-observed structure, which can be globally optimized with standard gradient-based methods. In addition, the region-based approach proposed by Sun et al. [2020] [52] incorporates a span relation network to capture the relationships between all possible spans in the sentence.

None Entity Dependency. Many approaches recognize entities without considering the dependence between entities, mainly based on regions. It is because existing region-based approaches mostly require to independently classify each region or boundary. In addition, there are some approaches, such as Lin et al. [2019] [29] and Li et al. [2020] [27], also do not consider the entity dependency.

5.2 Stage Framework

Generally speaking, the mainstream models to solve the flat NER problem are single-stage models (e.g., LSTM-CRF, CNN-LSTM-CRF), which complete entity detection and category prediction simultaneously. Extending to the nested NER task, many solutions including more than one stage have been proposed, in addition to continuing with the single-stage framework. For such approaches, the recognition process is artificially decomposed into several sequential operating stages or processes to improve the overall performance. Accordingly, we divide the existing approaches into single-stage and multi-stage frameworks.

Single-Stage Framework. The nested NER approaches with the single-stage framework have the ability to jointly detect entity boundaries and predict entity categories in only a single stage. Among the existing approaches, all hypergraph-based approaches [23, 33, 56] and transition-based approaches [13, 35, 57] belong to single-stage framework. In addition, some region-based approaches that first enumerate regions [29, 40, 49, 63] are classified as single-stage frameworks. These approaches only need to classify all possible regions (optionally considering its context) from the sentence. The single-stage nested NER approaches share the common characteristic of high demand for computing power to enhance the relatively slow training speed.

Multi-Stage Framework. The nested NER approaches with the multi-stage framework have two or more successive operating stages. Normally, such solutions artificially decompose the recognition process into more than one stage to improve the overall performance. It is worth noting that the number of stages of these approaches can be fixed or varying.

- **Fixed Multi-Stage Framework.** The fixed multi-stage framework approach for solving nested NER problem consists of a fixed number of stages. In general, these approaches deal with nested NER task by transforming this task into several subtasks that can be processed step by step, and the number of subtasks is fixed. A representative decomposition strategy is to decompose the nested NER task into detection and classification tasks [53, 61, 68]. The only exception is that Wang et al. [2020] [59] decomposed the nested NER task into three subtasks: detection, tagging, and classification. The other decomposition strategies are to decompose nested NER tasks into region fusion and category tagging tasks [14], or flat NER and graph-based nested NER tasks [34].
- **Varying Multi-Stage Framework.** The unfixed multi-stage approaches also decompose the nested NER task, while the number of subtasks is uncertain after decomposition. The representative decomposition is based on (i) nested levels, or (ii) entity categories. The nested

level-based approaches generally decompose the nested NER task into multiple flat NER tasks, such as Alex et al. [2007] [1] and Ju et al. [2018] [22]. Alternatively, the entity category-based method divides the entire nested NER task into multiple nested NER tasks with a single category, such as Muis and Lu [2017] [37] and Shibuya and Hovy [2020] [47].

From Table 3, we can observe that employing a multi-stage framework is currently a prevalent manner to solve nested NER task, which also indicates the complexity of the nested NER task. Combined with Table 2, we further observe that the model performance is not directly related to the stage framework property. In other words, there is no clear optimal one among single-stage, fixed multi-stage, and unfixed multi-stage frameworks for the nested NER task.

5.3 Error Propagation

Through the above categorizations, we observe that most of the existing models have more than one stage or consider entity dependency. However, these models are likely to suffer from the error propagation problem, i.e., errors in the previous decisions are propagated to the next decisions. In a nutshell, entity recognition consists of multiple consecutive decisions, which is prone to error propagation. Therefore, we consider error propagation as an important property of nested NER models and categorize existing approaches accordingly.

With Error Propagation. The nested NER models that are divided into two or more stages and there is an order relationship between the stages will raise the error propagation problem. The approaches that may cause error propagation for this reason include part of region-based approaches [14, 32, 53, 59, 61, 68] and most layered-based approaches [1, 22, 34, 47]. We take the layered-based approach as an example to explain in detail the causes of error propagation. Layered-based models commonly first extract the inner entities and feed them into the next layer to extract outer entities. Intuitively, extracting wrong entities from the previous layer will affect the performance of the next layer. Moreover, when an outer entity is extracted first, the inner one will be ignored by the model. Thus, most layered-based approaches suffer from error propagation. In addition, all transition-based approaches [13, 35, 57], which use a greedy training strategy, will raise error propagation (although they belong to the single-stage framework). For transition-based models, during training, the current action depends on the golden previous actions, while during testing, the entire action sequence is generated by the model. For this reason, an erroneous action prediction will further deviate from the predictions of follow-up actions. We also consider such accumulated discrepancy in transition-based approaches as error propagation.

Without Error Propagation. In all existing approaches, there are mainly two types of approaches that can avoid error propagation. The first type is hypergraph-based approaches [33, 56] that leverage graph-based global optimization strategies to complete boundary detection and category prediction at the same time. The second type is single-stage region-based approaches [4, 40, 63], which independently determine the entity category of each region. In particular, Wang et al. [2020] [58] also successfully avoided error propagation since each layer in their model predicts whether each region of a certain length is a complete entity mention.

After the above analysis, we further observe that considering entity dependence does not necessarily lead to error propagation, such as hypergraph-based approaches [33, 56]. Error propagation does not necessarily exist in multi-stage approaches, such as Muis and Lu [2017] [37] (independent between stages). More importantly, the current research results cannot verify that the model with error propagation will perform worse than the model without error propagation. We believe that the error propagation in the model will definitely bring some negative effects. So far, the performance level of the nested NER approach generally has room for improvement, thus many studies choose to improve performance first after weighing error propagation and performance improvement.

5.4 Tag Scheme

In the Background section, we introduce two types of tag schemes for the NER task, including the sequence tag scheme and the tuple tag scheme. The sequence tag schemes produce multiple sequences for expressing entities with different levels in sentences, and the tuple scheme uses a list of tuples to indicate all entities in sentences. In the existing nested NER approaches, both types of tag schemes are extremely pervasive. The approaches applying the sequence tag scheme mainly treat nested NER task as sequence tagging or sequence generation tasks, such as multiple sequence tagging tasks (i.e., layered-based approaches), and sequential decision tasks (i.e., transition-based approaches). The approaches using the tuple tag scheme mainly regard the nested NER task as a classification task, such as all region-based approaches. In addition, a small number of approaches require both the sequence tag scheme and the tuple tag scheme, neither of which is indispensable. For example, in the multi-stage region-based approaches [59, 68], multiple stages may require different tag schemes for their different targets. This indicates that different types of tag schemes have individuality and complement each other.

6 CHALLENGES AND FUTURE DIRECTIONS

As we expected, the nested NER task has drawn widespread attention of all parties and in-depth research because of its unique features. Based on the summary of over 30 studies, we have discussed the advantage and disadvantage of various kinds of nested NER approaches. We now would like to discuss the challenge and potential future directions for the nested NER task.

6.1 Challenges

Learning the nested structure. Benefiting from the powerful word-level representation, the nested NER system significantly improves the overall performance. The pre-trained language model representation has become one of the most effective auxiliary means to the nested NER task. This fact also conveys an optimistic message that without pre-trained language model embeddings, there is still remaining a large space worth exploring in future nested NER research. We consider that in addition to adopting or integrating powerful pre-trained language models as a complement, researchers also need to propose more effective and innovative methods for the nested nature of entities. This puts forward a requirement for researchers to face this peculiar nature of the NER task without hesitation.

Some researchers explored and treated the entire nested NER problem as other learning problems (e.g., multi-layer sequence tagging problem and syntactic structure parsing problem) to handle the nested structure in our task. Therefore, it is the most challenging to propose a nested NER model that can precisely express the nested structure, including the boundary information of entities and the dependency information between entities.

Alternatively, some studies complete nested NER in two steps, namely entity boundary detection and entity category prediction. In the case where different named entities partly share the same boundary word (e.g., a person entity “a spokesman for the Oakland Zoo” and a facility entity “the Oakland Zoo” share the same boundary word “Zoo”), this makes the nested NER model challenging to cope with the step of boundary detection. Furthermore, detecting all boundaries independently ignores the entity dependency even if different entities partly share a boundary, where the dependency between entities is one of the most important factors in nested structure. Thus, learning the nested structure is a consistent challenge in this task, even if the nested NER task is decomposed.

Data annotation. Data annotation remains time-consuming and expensive in the NER task. As shown in Table 1, there are relatively few nested NER datasets, and there are not many nested instances (that is, sentences containing nested structures) in each dataset. Specifically, 17% of the entities in the GENIA corpus are embedded within another entity, and 35% of sentences contain nested named entities in ACE 2005 corpora. More importantly, such datasets generally come from formal documents (e.g., news, biological domain), which makes the nested NER approaches difficult to apply to real-world scenarios. Compared to flat NER datasets, the quality of data annotation is more concerned in nested NER datasets. Another challenge for data annotation is that the current data formats, including the BIO tag scheme and the triple scheme, cannot well present the nested entity structure. We believe that the nested NER model will learn the nested structure more efficiently and effectively, if we label such type of information when entities are embedded within another entity.

6.2 Future Directions

A United architecture for nested NER. Based on the above discussion, we have at least four major architectures for nested NER task, including layered-based, region-based, hypergraph-based, and transition-based approaches. Meanwhile, we also categorize existing approaches based on entity dependency, stage framework, error propagation, and tag scheme. According to this, we can find that each architecture or category approach has its own advantages and shortcomings. It is worth noting that there are several explorations to propose a combination model that integrates the advantages from multiple types of approaches. We consider that the Pyramid model proposed by Wang et al. [2020] [58] is a successful instance, where the Pyramid model fuses the characteristics of region-based approaches though it is a layered-based approach. Therefore, it is worth exploring one united architecture for nested NER, which will have advantages in both theoretical and experimental results.

Deep transfer learning for nested NER. While we focus on developing more advanced architecture to address the nested NER problem, we also expect more other technologies to gain significant improvements for this task. Deep transfer learning has achieved great success in the NLP field in recent years. For addressing the flat NER problem, some studies [60] proposed the multitask models to jointly train on the flat NER task and its related tasks (e.g., the Chunking task and the Part-of-Speech task). These multitask models leverage the related task to extract and transfer shared knowledge, so that improves the overall performance both on flat NER task and its related tasks. Therefore, we believe that such a deep transfer manner can also perform well in nested NER task. Besides, one interesting transfer direction might be the granularity knowledge transfer, which means transferring knowledge from flat NER task to nested NER task. Different from the joint training manner, we desire the granularity knowledge transfer model can leverage the coarse-grained knowledge from a flat NER model (could be a sequence tagging architecture) to improve the overall performance of the nested NER model.

Semi-supervised Learning for nested NER. As we mentioned before, the data annotation cost for the nested NER task is much higher than the flat NER task. As data annotation is one important challenge for our task, one interesting research direction is developing nested NER approaches (e.g., semi-supervised learning architectures and even unsupervised learning architectures) to adapt to the current environment. In addition, all approaches we present in this survey are strong supervised models. In flat NER task, there are many semi-supervised approaches, but they are hardly applied in nested NER task since such approaches are mainly based on sequence tagging architecture. For example, He and Sun [2017] [20] proposed a unified model which can learn from out-of-domain corpora and in-domain unannotated texts, where the self-training learning function is based on the sentence confidence and decision boundary. We consider that exploring the semi-supervised models for nested NER can further enhance the possibility of applying nested NER systems into real-world downstream natural language applications.

A united tag scheme for nested NER. As discussed in Section 5.4, different types of nested NER approaches adopt unique tag schemes. For example, region-based approaches generally leverage the tuple tag scheme as the entity label information, while layered-based approaches retain the sequence tag scheme, which is more commonly used in the flat NER task. The original intention of the tag scheme is to present the entity label information in the sentence. In order to accurately and comprehensively present the label information of more complex nested entities, the choice of tag scheme is particularly critical. In practice, neither of the above tag schemes can fully express the nested structure, because the dependency relationship between inner and outer entities in the same nesting structure has to be abandoned in the tagging process. To the best of our knowledge, no existing work separately focuses on the design of precise tag schemes for nested NER. We expect a breakout in this research direction in the future.

7 CONCLUSION

To the best of our knowledge, we presented a first comprehensive survey of the nested NER task. This survey aims at reviewing all studies of nested NER for more than 18 years to help researchers quickly get a quick overview of this field. We first give clear definitions about nested NER, a brief background of this task, and a detailed comparison with flat NER task. Second, we present an overview of the approaches employed to solve the nested NER problem from the perspectives of model architecture and model property. We provide these approaches in a tabular form for each perspective and show their technical details for easy comparison and summarization. Finally, although the state-of-the-art for nested NER has improved a lot in the last decade, we further present the challenges and future directions in this task. We believe the nested NER will be a new fundamental task in the NLP field and widely used to solve other challenging natural language applications.

REFERENCES

[1] Beatrice Alex, Barry Haddow, and Claire Grover. 2007. Recognising nested named entities in biomedical text. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (Workshop)*. 65–72.

[2] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*. 1–15.

[3] Krisztian Balog, Pavel Serdyukov, and Arjen P. De Vries. 2010. Overview of the TREC 2010 entity track. In *Proceedings of the TREC*.

[4] Kate Byrne. 2007. Nested named entity recognition in historical archive text. In *Proceedings of the International Conference on Semantic Computing*. 589–596.

[5] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning* 20, 3 (1995), 273–297.

[6] James R. Curran and Stephen Clark. 2003. Language independent NER using a maximum entropy tagger. In *Proceedings of the 7th Conference on Natural Language Learning*. 164–167.

[7] Xiang Dai. 2018. Recognizing complex entity mentions: A review and future directions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Workshop)*. 37–44.

[8] Gianluca Demartini, Tereza Iofciu, and Arjen P. De Vries. 2009. Overview of the INEX 2009 entity ranking track. In *Proceedings of the International Workshop of the Initiative for the Evaluation of XML Retrieval*.

[9] Xin Luna Dong, Hannaneh Hajishirzi, Colin Lockard, and Prashant Shiralkar. 2020. Multi-modal information extraction from text, semi-structured, and tabular data on the web. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3543–3544.

[10] Sean R. Eddy. 1996. Hidden markov models. *Current Opinion in Structural Biology* 6, 3 (1996), 361–365.

[11] Hao Fei, Yafeng Ren, and Donghong Ji. 2020. Dispatched attention with multi-task learning for nested mention recognition. *Information Sciences* 513 (2020), 241–251.

[12] Jenny Rose Finkel, Alex Kleeman, and Christopher D. Manning. 2008. Efficient, feature-based, conditional random field parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 959–967.

[13] Jenny Rose Finkel and Christopher D. Manning. 2009. Nested named entity recognition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 141–150.

[14] Joseph Fisher and Andreas Vlachos. 2019. Merge and label: A novel neural network architecture for nested NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 5840–5850.

[15] Christoph Goller and Andreas Kuchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of the International Conference on Neural Networks*. 347–352.

[16] Archana Goyal, Vishal Gupta, and Manish Kumar. 2018. Recent named entity recognition and classification techniques: A systematic review. *Computer Science Review* 29 (2018), 21–43.

[17] Ralph Grishman and Beth M. Sundheim. 1996. Message understanding conference-6: A brief history. In *Proceedings of the 16th International Conference on Computational Linguistics*. 466–471.

[18] Baohua Gu. 2006. Recognizing nested named entities in GENIA corpus. In *Proceedings of the BioNLP Workshop on Linking Natural Language Processing and Biology*. 112–113.

[19] Poonam Gupta and Vishal Gupta. 2012. A survey of text question answering techniques. *International Journal of Computer Applications* 53, 4 (2012), 1–8.

[20] Hangfeng He and Xu Sun. 2017. A unified model for cross-domain and semi-supervised named entity recognition in chinese social media. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. 3216–3222.

[21] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[22] Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. 2018. A neural layered model for nested named entity recognition. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1446–1459.

[23] Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 861–871.

[24] John Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*. 282–289.

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

[26] Peipei Li, Haixun Wang, Hongsong Li, and Xindong Wu. 2018. Employing semantic context for sparse information extraction assessment. *ACM Transactions on Knowledge Discovery from Data* 12, 5 (2018), 1–36.

[27] 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*. 5849–5859.

[28] Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. Bond: Bert-assisted open-domain named entity recognition with distant supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1054–1064.

[29] 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*. 5182–5192.

[30] Hongyu Lin, Yaojie Lu, Xianpei Han, Le Sun, Bin Dong, and Shanshan Jiang. 2019. Gazetteer-enhanced attentive neural networks for named entity recognition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. 6233–6238.

[31] Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*. 94–100.

[32] Xinwei Long, Shuzi Niu, and Yucheng Li. 2020. Hierarchical region learning for nested named entity recognition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: Findings*. 4788–4793.

[33] Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 857–867.

[34] 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*. 6408–6418.

[35] Zita Marinho, Alfonso Mendes, Sebastiao Miranda, and David Nogueira. 2019. Hierarchical nested named entity recognition. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*. 28–34.

[36] Mónica Marrero, Julián Urbano, Sonia Sánchez-Cuadrado, Jorge Morato, and Juan Miguel Gómez-Berbís. 2013. Named entity recognition: Fallacies, challenges and opportunities. *Computer Standards & Interfaces* 35, 5 (2013), 482–489.

[37] Aldrian Obaja Muis and Wei Lu. 2017. Labeling gaps between words: Recognizing overlapping mentions with mention separators. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2608–2618.

[38] David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes* 30, 1 (2007), 3–26.

[39] Joakim Nivre and Ryan McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 950–958.

[40] Hiroki Ouchi, Jun Suzuki, Sosuke Kobayashi, Sho Yokoi, Tatsuki Kuribayashi, Ryuto Konno, and Kentaro Inui. 2020. Instance-based learning of span representations: A case study through named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* 6452–6459.

[41] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24, 4 (2016), 694–707.

[42] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning: Shared Task.* 1–40.

[43] Jipeng Qiang, Ping Chen, Wei Ding, Tong Wang, Fei Xie, and Xindong Wu. 2019. Heterogeneous-length text topic modeling for reader-aware multi-document summarization. *ACM Transactions on Knowledge Discovery from Data* 13, 4 (2019), 1–21.

[44] C. Janarish Saju and A. S. Shaja. 2017. A survey on efficient extraction of named entities from new domains using big data analytics. In *Proceedings of the 2nd International Conference on Recent Trends and Challenges in Computational Models.* 170–175.

[45] 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 7th Conference on Natural Language Learning.*

[46] Dan Shen, Jie Zhang, Guodong Zhou, Jian Su, and Chew Lim Tan. 2003. Effective adaptation of hidden markov model-based named entity recognizer for biomedical domain. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (Workshop).* 49–56.

[47] Takashi Shibuya and Eduard Hovy. 2020. Nested named entity recognition via second-best sequence learning and decoding. *Transactions of the Association for Computational Linguistics* 8 (2020), 605–620.

[48] Carlos N. Silla and Alex A. Freitas. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery* 22, 1–2 (2011), 31–72.

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

[50] 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.* 5326–5331.

[51] Stephanie Strassel and Alexis Mitchell. 2003. Multilingual resources for entity extraction. In *Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics (Workshop).* 49–56.

[52] Lin Sun, Yuxuan Sun, Fule Ji, and Chi Wang. 2020. Joint learning of token context and span feature for span-based nested NER. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), 2720–2730.

[53] Chuanqi Tan, Wei Qiu, Mosha Chen, Rui Wang, and Fei Huang. 2020. Boundary enhanced neural span classification for nested named entity recognition. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence.* 9016–9023.

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st Conference on Neural Information Processing Systems.* 6000–6010.

[55] Andrew Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory* 13, 2 (1967), 260–269.

[56] Bailin Wang and Wei Lu. 2018. Neural segmental hypergraphs for overlapping mention recognition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing.* 204–214.

[57] Bailin Wang, Wei Lu, Yu Wang, and Hongxia Jin. 2018. A neural transition-based model for nested mention recognition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing.* 1011–1017.

[58] 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.* 5918–5928.

[59] Yu Wang, Yun Li, Hanghang Tong, and Ziye Zhu. 2020. HIT: Nested named entity recognition via head-tail pair and token interaction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing.* 6027–6036.

[60] Yu Wang, Yun Li, Ziye Zhu, Hanghang Tong, and Yue Huang. 2020. Adversarial learning for multi-task sequence labeling with attention mechanism. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), 2476–2488.

[61] Congying Xia, Chenwei Zhang, Tao Yang, Yaliang Li, Nan Du, Xian Wu, Wei Fan, Fenglong Ma, and S. Yu Philip. 2019. Multi-grained named entity recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.* 1430–1440.

[62] Canwen Xu, Feiyang Wang, Jialong Han, and Chenliang Li. 2019. Exploiting multiple embeddings for chinese named entity recognition. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management.* 2269–2272.

[63] Mingbin Xu, Hui Jiang, and Sedtawut Watcharawittayakul. 2017. A local detection approach for named entity recognition and mention detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. 1237–1247.

[64] Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*. 2145–2158.

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

[66] Jie Zhang, Dan Shen, Guodong Zhou, Jian Su, and Chew-Lim Tan. 2004. Enhancing HMM-based biomedical named entity recognition by studying special phenomena. *Journal of Biomedical Informatics* 37, 6 (2004), 411–422.

[67] Shiliang Zhang, Hui Jiang, Mingbin Xu, Junfeng Hou, and Li-Rong Dai. 2015. The fixed-size ordinally-forgetting encoding method for neural network language models. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. 495–500.

[68] Changmeng Zheng, Yi Cai, Jingyun Xu, Ho-fung Leung, and Guandong Xu. 2019. A boundary-aware neural model for nested named entity recognition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. 357–366.

[69] Guodong Zhou. 2006. Recognizing names in biomedical texts using mutual information independence model and SVM plus sigmoid. *International Journal of Medical Informatics* 75, 6 (2006), 456–467.

[70] Guodong Zhou, Jie Zhang, Jian Su, Dan Shen, and Chewlim Tan. 2004. Recognizing names in biomedical texts: A machine learning approach. *Bioinformatics* 20, 7 (2004), 1178–1190.

Received August 2021; revised December 2021; accepted February 2022