STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **12**, July 2024, pp 907–942.
Published online in International Academic Press (www.IAPress.org)

Systematic Literature Review on Named Entity Recognition: Approach, Method, and Application

Warto [1,2,*], Supriadi Rustad[2], Guruh Fajar Shidik[2], Edi Noersasongko[2], Purwanto[2], Muljono[2], De Rosal Ignatius Moses Setiadi[2]

[1] *Informatics, Faculty of Da'wah, UIN Saizu, Purwokerto, 53126, Indonesia*
[2] *Faculty of Computer Science, Dian Nuswantoro University, Semarang, 50131, Indonesia*

Abstract Named entity recognition (NER) is one of the preprocessing stages in natural language processing (NLP), which functions to detect and classify entities in the corpus. NER results are used in various NLP applications, including sentiment analysis, text summarization, chatbot, machine translation, and question answering. Several previous reviews partially discussed NER, for instance, NER reviews in specific domains, NER classification, and NER deep learning. This paper provides a comprehensive and systematic review on NER topic studies published from 2011 to 2020. The main contribution of this review is to present a comprehensive systematic literature review on NER from preprocessing techniques, datasets, application domains, feature extraction techniques, approaches, methods, and evaluation techniques. The result concludes that the deep learning approach and the Bi-directional long short-term memory with a conditional random field (Bi-LSTM-CRF) method are the most interesting methods among NER researchers. At the same time, medical and health are NER researchers' most popular domains. These developments have also led to an increasing number of public datasets in the medical and health fields. At the end of this review, we recommend some opportunities and challenges for NER research going forward.

Keywords Named entity recognition, Entity extraction, Entity detection, Entity classification, Natural

DOI: 10.19139/soic-2310-5070-1631

1. Introduction

Information extraction (IE) is one of the main problems in text mining, aims to find structured information from unstructured or semi-structured text [1,2]. In natural language processing (NLP) study, IE has two tasks, namely: 1) Named Entity Recognition (NER), and 2) Relationship Extraction (RE) [1,2]. Pirskorski and Yangarber [3], added two more tasks, which are 3) Co-Reference Resolution (CRR) and 4) Event Extraction (EE). NER presents challenges because text data, such as new vocabulary, are constantly changing [4]. In addition, NER also has an essential role in NLP because it is one of the preprocessing stages used is several tasks such as text summarization, machine translation, information retrieval, question answering, and chatbot [5].

Named Entity was first introduced in MUC-6 [6], as the forerunner of named entity recognition, which is used for identifying entities in free-text and classifying them into person (PER), location (LOC), and organization entities (ORG) [2,4,5,7–9]. Today, it can identify entities in various knowledge domains.

NER has several stages to complete its functions, as seen in Figure 1. The first stage of NER is making a dataset/corpus by taking a website or social media script. The popular term for this first stage is "scraping". Some methods and tools can be used for scraping, including Octoparse and Scrapingbee. The next step is

*Correspondence to: Warto (Email: warto@uinsaizu.ac.id). Informatics, Faculty of Da'wah, UIN Saizu, Purwokerto, 53126, Indonesia.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

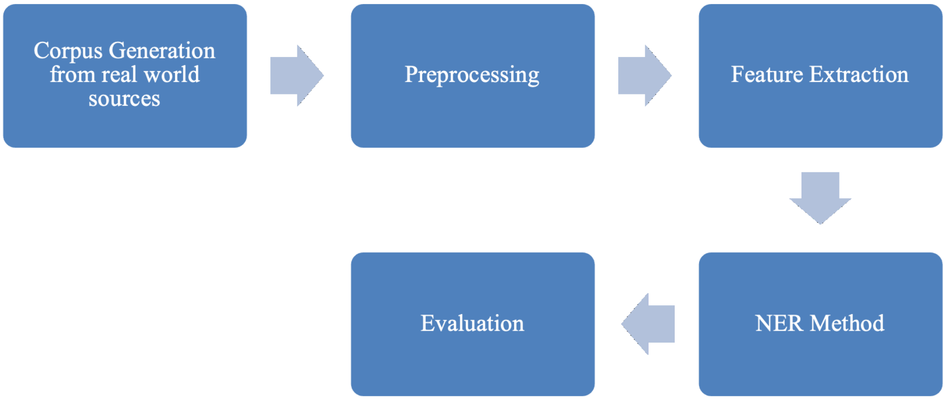

Figure 1. NER Phase.

preprocessing, which is helpful to clean the data from various noises. Several preprocessing steps are described in more detail in sub-section 3.3. The next stage is feature extraction which can be done using multiple techniques, as shown in subsection 3.6. Some extracted features include linguistic, orthographic, morphological, context, and lexicon features. Each feature has different uses, as explained by Eltayeb and Salim [10]. The next stage is the implementation of the NER method. The most widely used methods for NER work are discussed in detail in subsection 3.8. The final stage is to evaluate the performance of NER. The learning rate is also discussed in this article. The learning rate is an important discus-sion in developing deep learning as a method for determining parameters that control how much the model parameters change at each step during the training process, affecting the convergence speed and stability of the training, discussed in subsection 3.9. The evalua-tion methods used are Precision, Recall, and F-Score, discussed in subsection 3.10.

NER consists of two tasks:1) entity identification, 2) entity classification [11-13], where these tasks become structured data suppliers in the next NLP stage [7]. Entity identification is the process of determining words/phrases in the corpus (regardless of whether entities or not). It is followed by entity classification, which determines the entity type based on the defined entity class [13]. The entity classes commonly used are person (PER), location (LOC), and organization (ORG). Nadeau and Satoshi mentioned several strategies to solve NER problems: 1) a rule-based approach, 2) dictionaries-based, 3) supervised machine learning, 4) a combination of rule-based, dictionaries-based and supervised machine learning [12]. However, according to Akkasi and Varoglu, these methods have many drawbacks, making them challenging to implement, for example, the difficulty of a rule-based approach to explore updated rules and patterns in the newly created text [14].

Studies of NER have increased in recent years, and researchers have proposed various methods, approaches, datasets, feature extraction, preprocessing, and application domains. A large number of NER publications provide an opportunity to review this topic. Wen et al., [15], studied NER with an emphasis on a) rule-based and dictionary-based methods, b) statistical learning-based, c) hybrid method, and d) deep learning-based method. Other researchers conducted NER reviews in the chemical [10], clinical [16], biomedicals [17,18], and food domains [19]. Several NER reviews discuss method approaches, namely classification [20,21], decision tree [22], active learning [16], and unsupervised [23]. Thomas et al. [24] and Li et al. [25] conducted an NER review that specifically uses a deep learning approach. Dandhasi et al. [26] and Shalaan [27] reviewed NER publications specifically for Arabic. Several NER review articles indicate that NER is one of the popular topics in NLP topic research.

This article aims to discuss NER research more comprehensively than previous research. This NER review uses a systematic literature review (SLR) technique. We present the development of NER research from 2011 to 2020. The study includes a discussion of a) NER research trends from year to year; b) researchers and journals that publish many NER articles; c) dataset and preprocessing; d) application domain; e) type of feature and feature extraction technique; f) the approach used; g) the NER method and its performance evaluation technique. This

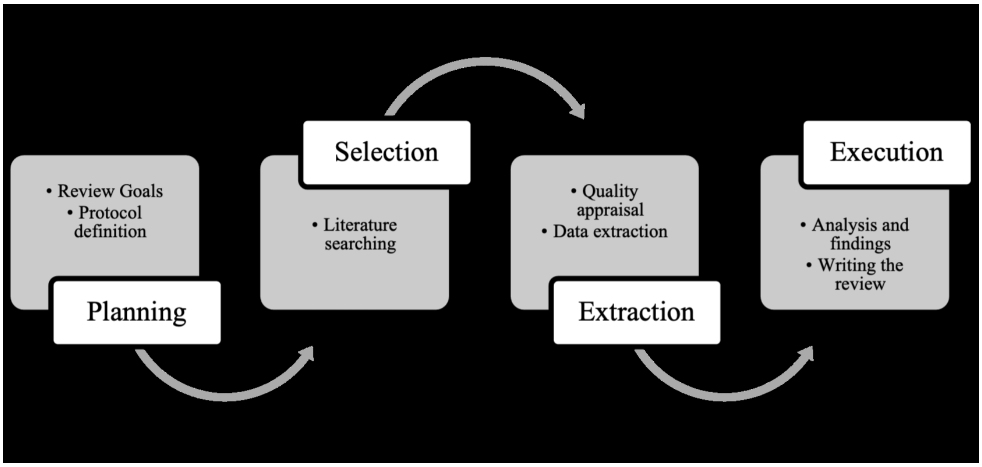

Figure 2. SLR Phase [29].

article begins with an introduction and continues with Section 2, which describes the review method, Section 3 presents the results and analysis; and Section 4 contains conclusions and suggestions.

2. Method

2.1. Review method

This NER review uses a systematic literature review technique adopted from medical and health science to computer science by Kitchenham [28]. The advantage of this SLR technique is that other researchers can later perform the same technique on the same topic. In general, SLR consists of four stages, namely (1) planning, (2) selection, (3) extraction, and (4) execution, as shown in Figure 2 The planning stage contains the determination of PICOC criteria, as shown in Table 1.

2.2. Research question

This review aims to answer several research questions (RQ), as seen in Table 2. This review aims to answer several research questions (RQ), as seen in Table 2. RQ1 and RQ2 identify published journals and productive authors contributing to NER research. RQ3-RQ5 discusses preprocessing, datasets, and NER application domains, RQ6-RQ8 discusses techniques feature extraction, NER approach, and method, and RQ9 identifies NER evaluation techniques.

Table 1. PICOC Criteria.

Population	Named Entity Recognition, Entity Extraction
Intervention	Approach or method preprocessing, dataset or corpus, feature extraction, feature decomposition, classification, and optimization, in NER
Comparison	-
Outcomes	Method performance
Context	Studies in general and domain-specific

Table 2. Research question.

RQ number	Research Question	Goal
RQ1	What journals publish NER articles?	Identify which journals publish the most NER articles.
RQ2	Who are the most prolific NER-themed writers?	Identify the authors who have published the most NER articles.
RQ3	What are the most widely used preprocessing methods?	Identify the preprocessing methods that NER researchers widely use.
RQ4	What datasets do the researchers use in NER?	Identify datasets that are popular and widely used for NER research.
RQ5	Where is the NER application domain used?	Identify popular application domains in NER research.
RQ6	What are the feature extraction methods widely used in NER?	Identify which NER researchers widely use feature extraction methods.
RQ7	What approaches do the researchers widely use to solve the NER problem?	Identify what approaches the researchers widely use to solve NER problems.
RQ8	What methods do the researchers widely implement to solve NER problems?	Identify widely implemented methods to solve NER problems.
RQ9	What evaluation techniques do researchers use widely to measure NER performance?	Identify widely used evaluation techniques to measure NER performance.

2.3. Search strategy

As shown in Figure 3, the search strategy begins with selecting a digital library. We chose IEEExplore, ACM, Springer, and ScienceDirect digital libraries. IEEExplore and ACM are the main choices because they are scientific associations with a high reputation in computers and information technology. Both associations are also publishers of reputable scientific journals. Meanwhile, Springer and ScienceDirect were chosen because they are publishers of scientific journals with a good reputation among scholars. Search for strings used Boolean OR and AND with structure (named entity recognition OR entity recognition OR entity recognition OR NER OR named entity OR extraction entity) AND (approach * OR technique * OR Method * OR procedure * OR way) AND (performance OR evaluation OR measure OR Assessment) AND (general OR general OR conventional OR application OR domain-specific). However, because the creators of the data search on the database indexer of scientific articles are different, the query adapts to the characteristics of the search engine. For example, ScienceDirect only allows a maximum of eight Boolean strings. The basic search was on English-language articles published from 2011-2022. This process of repairing this query, as illustrated in Figure 3, was repeated until the search results were more accurate. At this search stage, it produced 3108 article titles containing the keywords we had defined. A search of the ACM digital library returned 628 titles, IEEExplore 473 titles, ScienceDirect 721 titles, and Springer 1286 titles.

RQ1 and RQ2 identify published journals and productive authors contributing to NER research. RQ3-RQ5 discusses preprocessing, datasets, and NER application domains, RQ6-RQ8 discusses techniques feature extraction, NER approach, and method, and RQ9 identifies NER evaluation techniques.

2.4. Study selection

Literature search and selection were conducted in the second stage with source searches from the IEEEXplore, ACM, ScienceDirect, and Springer article indexers. At this stage, 2748 articles from journals and conferences were obtained, as shown in Figure 3. In the next step, a quality review was carried out based on the title and abstract.

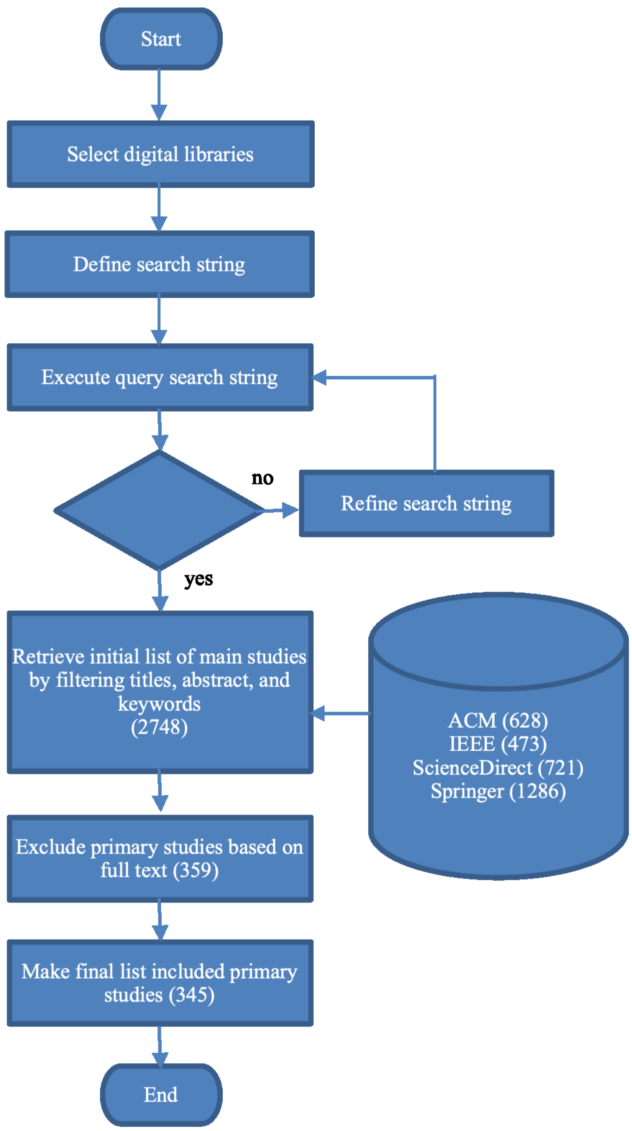

Figure 3. Journal search and selection phase.

Articles that not meet the criteria were excluded, while those that met the requirements were included and entered at the next stage. The included criteria are NER articles that list topics, problems, datasets, and methods used. The following criteria are articles published in the period 2011 to 2022.

Meanwhile, the exclude criteria are: a) articles that display unclear experimental results, b) only use private datasets, and c) articles not written in English. This exclude-include process resulted in 345 article titles which were then extracted based on nine RQs. The included criteria were further analyzed to find answers based on the nine research questions, as shown in Table 2.

2.5. Data extraction

After retrieving 345 articles, the next step was creating a matrix using a spreadsheet to extract data according to RQ1-RQ9. The data extraction process was carried out by reading the articles in detail. Then if data were found matched to one of the RQs, they were entered into the matrix. After the extraction process was complete, it processed the data for each RQ and visualized the graph for pattern analysis on each RQ.

3. Result and Analysis

3.1. Journals publishing NER articles

The number of journal manuscripts of NER has increased significantly during the last decade, with an average of 17 articles per year. A significant increase occurred in 2020 with as many as 61 titles and 2020 with 92 articles, as shown in Figure 4. Article publications increased sharply from 2018 to 2022, along with the popularity of the deep learning approach, seen from the many deep learning approaches published in 2018-2022. For example, in 2020 and 2021, from 61 and 92 titles, 53 and 87 used a deep learning approach (discussed in more detail in Section 3.7). NER publications until August 2022 have reached 63 articles.

NER topic articles were published in 134 journals, with the top ten journals shown in Figure 5. From 134 journals, 60% were in Q1, 30% were in Q2, and the remaining 9% were in Q3. IEEE Access became the journal that published the most articles about NER (32 titles), followed by the Journal of Biomedical Informatics (28 titles). A journal list can be considered for NER researchers to determine where NER articles should be published. The number of articles published in the Journal of Bioinformatics is the largest in the medical and health field. A journal

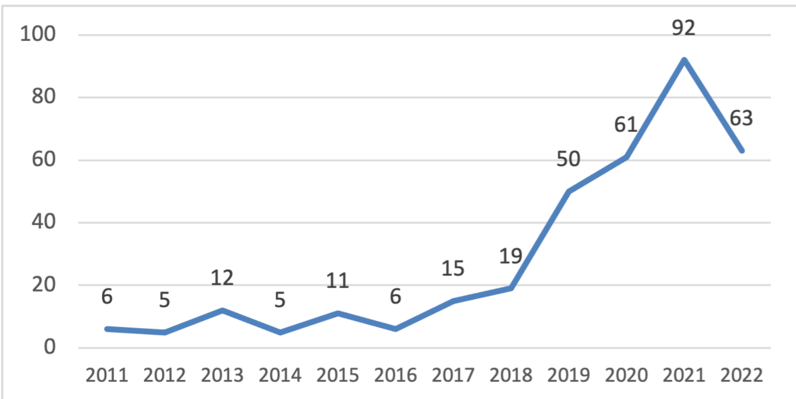

Figure 4. NER publications from 2011 to 2022.

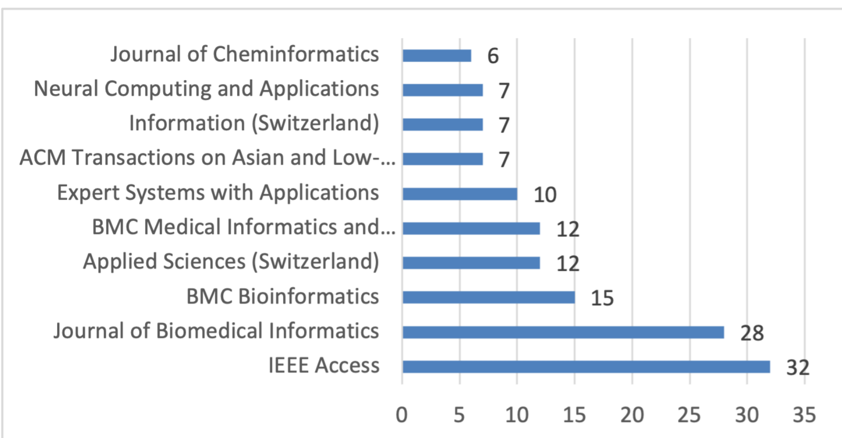

Figure 5. List of journals that publish NER articles.

list can be a consideration for NER researchers to publish articles on NER topics. Journal and publisher selection in publishing scientific papers are essential because many predatory journals are circulating lately, journals whose publication does not heed the publication code of ethics. The journals included in this review are reputable journals published by scientific associations. The novelty of each article can be scientifically justified. In addition, these journals are indexed by Scopus from Q1-Q3.

3.2. Active and influential researchers on NER topics

Researchers play an essential role in the sustainability of a research topic. A researcher with many publications on a case can be considered an expert in that field. Among the 190 curated articles, two names stand out, namely Asif Ekbal, with ten titles (seven titles as the first author [30–35], and three titles as the second author [36–38]. Sriparna Saha has ten (one title as the first author [36], and seven as the second author [30–35], and two as the third author [37,38]. There are at least ten articles in collaboration between Sriparna Saha and Asif Ekbal, two of them as the second or third authors. They combine many techniques of machine learning (genetic algorithm (GA), support vector machine (SVM), Conditional Random Field (CRF), maximum entropy (ME), memory-based learning (MBL), Hidden Markov model (HMM), and naïve Bayes (NB)) to improve their NER performance. Asif and Sriparna's NER research activities dominated from 2011 to 2016. In 2017, the two researchers began to decrease the intensity of publication of the NER topic, where the deep learning approach began to dominate. For example, articles still use a machine learning instead of deep learning [37].

Lin Sun published three titles as the first author of a deep learning approach.[39]–[41] As seen in Figure 6, several researchers active on the topic of NER are Akasi [14,42], Cho [43,44] Gao [45,46], Qiu [47,48] Tang [49,50], Yadav [37,38], and Naixin [51,52].

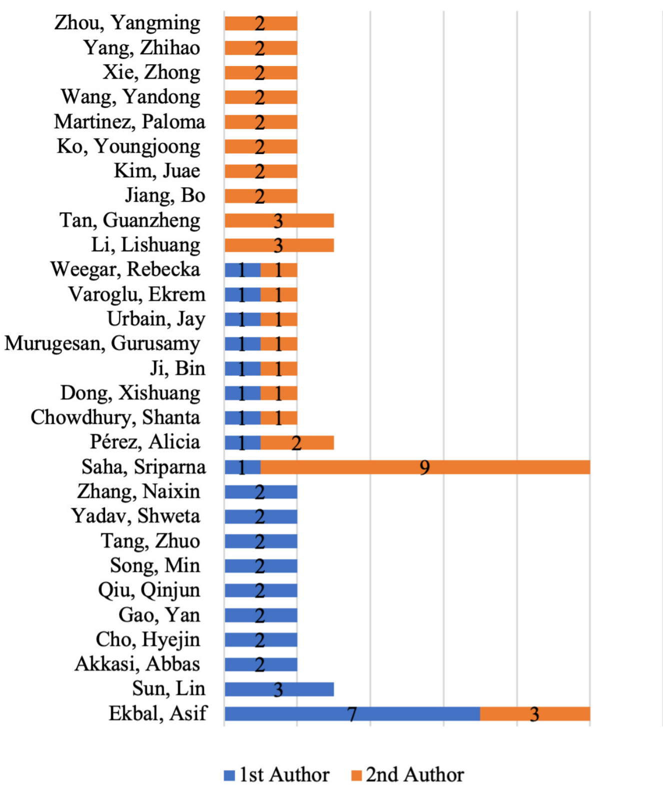

Figure 6. The authors with the most NER articles.

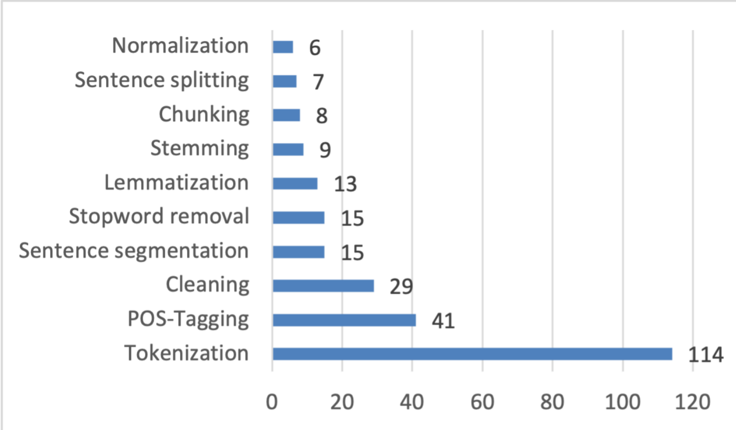

Figure 7. Various NER preprocessing techniques. Horizontal axes indicate several publications using these techniques.

Discussions about researchers who are influential on the topic of NER cannot only be based on the number of articles published by that person but can also use the number of citations. A search using the semanticscholar.org search engine shows that the author has the highest number of sources in articles on the topic of NER is Lample et al. [53]. Their paper was published in the Proceedings of NAACL-HLT 2016 under the title "Neural Architectures for Named Entity Recognition" was cited up to 2,446 times. Lample proposed two approaches: the Bi-LSTM neural architecture with CRF and segment labeling using a transition-based method inspired by the shift-reduce parser. He explored a new architecture that constructs chunks and labels a sequence of inputs using a similar algorithm to transition-based dependency [54]. It is called Stack-LSTM for "simplicity," in which the LSTM is augmented with a "stack pointer". Ritter et al. [55], wrote an article titled "Named Entity Recognition in Tweets: An Experimental Study", which was cited 1,231 times. They proposed that T-NER outperforms the state-of-the-art news-trained counterparts, reducing error by 41%. T-POS and T-CHUNK in segmenting Named entities perform much better than other POS tagging and Chunking tools.

3.3. *NER preprocessing techniques*

In the NLP research field, NER includes a preprocessing stage. Meanwhile, the NER topic also has a preprocessing step. We need to emphasize here that what we discuss in this article is not placing NER as a preprocessing stage in NLP, but NER as a computational process with several step ranging from preprocessing, representation, detection, and classification to evaluation stages. As shown in Figure 7, there are many preprocessing methods, shuch as tokenization, POS-Tagging, cleaning, sentence segmentation, stopword removal, lemmatization, chunking, stemming, and normalization. All these methods do not replace each other but are interrelated and complementary. Some NER researchers use several stages [56–60], others use just one preprocessing stage [42,61–64]. The use of various preprocessing techniques depends on the scope of the research, the dataset used, and the adjustment to the NER project being undertaken. For example, a study by Mu et al. [65], does not explicitly state the preprocessing stage because their research uses the CoNLL2003 "ready to classify" dataset.

Tokenization is the most widely used preprocessing technique, followed by part-of-speech (POS)-tagging and data cleaning, as shown in Figure 7. Tokenization is responsible for changing the sequence of characters in raw data into a word sequence (or token) [66]. Data cleaning is performed to remove noise that reduces NER performance, e.g., deleting "empty tokens" on bigram tokenization. NER researchers mostly do data cleaning to clean up data, including punctuation mark deletion, hashtags, URL removal, and excess space removal. The most common tokenization technique is white space to break raw text data into tokens. Tokenization is the most preferred choice in the NER preprocessing stage; it prepares the manuscript for NER annotation. The entity detection process is carried out on tokens generated by this preprocessing stage. Spaces-based tokenization is excellent for Roman-type-based languages like English and Indonesian.

A large dataset/corpus is generally performed segmentation, breaking the large corpus into smaller units [66]. Segmenting pieces can be either a paragraph or a sentence. Furthermore, common words such as a, an, any, are, by, etc., and, so, are removed using a stop-word removal technique, reducing the dataset volume so that computations can be done more quickly. Many NER researchers also carry out the preprocessing stage with lemmatization to find the root form of a word, for example, studied, spoke, spent, told, converted into a study, speak, spend, tell. Lemmatization is more widely used because the results are more accurate than stemming, which only removes additives using a rule-based method.

Tokenization has a significant effect on information extraction tasks, including NER. Khabsa and Giles compare three tokenizer techniques for accuracy in the chemical domain [67]. Among the three tokenizers, OSCAR4 gave the highest accuracy percentage of 87%, compared to ChemSpot and ChemXSeer, approximately 77.03% and 83.06%, respectively. Akkasi et al. [68] proposed ChemTok and compared it with other tokenization techniques (Whitespace, ChempSpot, and tmVar) on two datasets (DrugBank and Medline). They proved that the use of different tokenization affects NER performance. In testing using the DrugBank dataset with the SVM classification algorithm, ChemTok's tokenization can increase the F-Score value to 91.79. Meanwhile, WhiteSpace, ChemSpot, and tmVar only gave 82.85, 89.10, and 90.34 F-Score, respectively [68]. Besides tokenization, POS-Tagging is an option for the NER preprocessing step. Familiar entities, such as PER, LOC, and ORG, are nouns. POS-tagging can speed up and simplify recognizing entities based on nouns.

POS-Tagging conceptually has a similar task to NER: annotating the text. POS-Tagging performs text annotations based on part-of-speech such as nouns, pronouns, verbs, adverbs, prepositions, and others. Meanwhile, NER performs annotations based on certain entities such as a person, location, and organization. Many researchers use POS-Tag information to improve NER performance, as done by Atkinson and Bull [59], Cai et al. [69], Cho et al. [70], and Suárez-Paniagua et al. [71], Gaur et al. [72], use POS tag as a feature to identify entity based on proper noun tags. Entities are generally nouns, but, as revealed by Zhou et al. [73], who researched bugs in software, entities can also be adjectives or adverbs. By adding the POS-Tag feature to NER, you can add to the entity's treasury, such as in Zhou's system and software domain [73]. POS-Tag on ChER (chemical entity recognizer) performed by Navvaro et al. claims to improve the performance of NER compared to ChemSpot and MetaboliNER [74]. As shown in Figure 7, cleaning is also popularly used by NER researchers to clean raw data. Raw data from generic websites contain many HTML and XML tags, advertisements, and menus [75]. Raw data emails contain signatures and attachments. Meanwhile, raw PDF data contain many extra spaces between words, inconsistent parentheses, and inconsistent sentence segmentation [72]. Punctuation like (+, -, /, *, (,), _, & ,%, .,), ÷, ×,],], }, {, <, >, \, —, @, #) also need to be cleaned from the corpus [76]. These various noises must be cleaned to produce a clean corpus ready to be annotated. This cleaning step is also useful for reducing the dimensions of the corpus.

Sentence segmentation is a preprocessing stage that is useful for parsing the text in the document into sentences. One segment contains one sentence. Sentence detection is done using a period (.), a question mark (?), an exclamation mark (!), and a suspension point (...) [57]. Some researchers apply sentence segmentation using various tools such as Stanford parser,[77] Jieba segmentation system [40,78] and Apache OpenNLP Maxent Sentence Detector [56,79]. Other researchers do segmentation not based on sentences but words, as done by Sun et al. [40] and Song et al. [80], and Song et al. [80] combine word segmentation with POS-Tag to increase the accuracy of CRF-based entity recognition.

3.4. Dataset on NER topics

Research that uses the "ready to classify" dataset - for example, CoNLL2003, JNLPBA, BC2GM dataset - generally does not carry out the preprocessing stage. While NER research uses raw data, preprocessing steps are essential for entity detection and classification. Some studies that use raw data write down the preprocessing actions taken [73,77,81]. However, some others do not explicitly carry out the preprocessing stage [82–85]. As seen in Figure 8, more NER researchers use public datasets than private datasets (Figure 9). This trend is consistent yearly, and researchers using public datasets is increasing.

As shown in Figure 10, the most popular public dataset NER researchers use is CoNLL2003. It consists of three entities, namely (PER), location (LOC), and organization (ORG). CoNLL2003 dataset was still widely used until 2022, including by Gaur et al. [72], Wang et al. [86], Zhong et al. [87], Liu et al. [88], Xiaofeng et al. [65], and

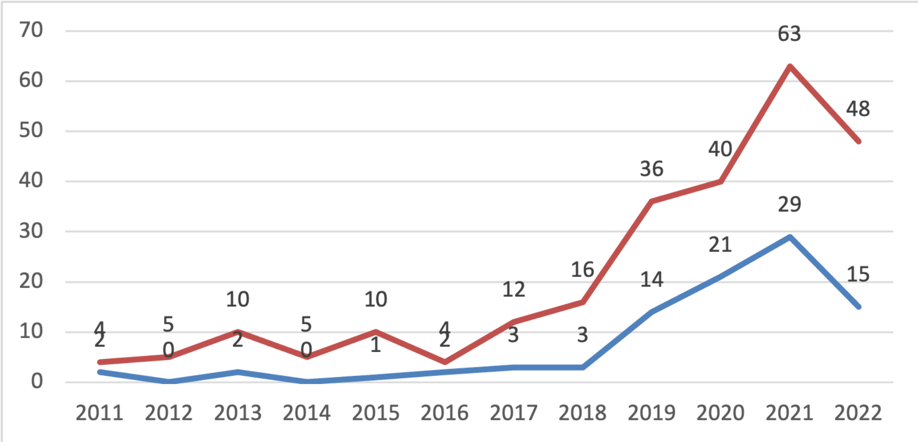

Figure 8. Use of private and public datasets from year to year.

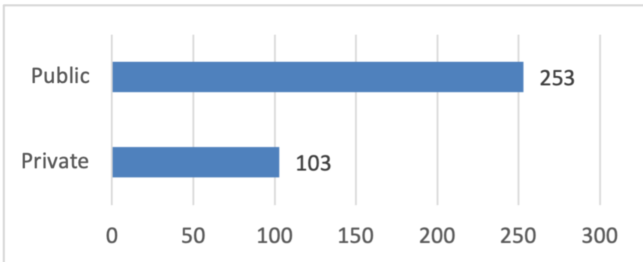

Figure 9. Use of public and private datasets.

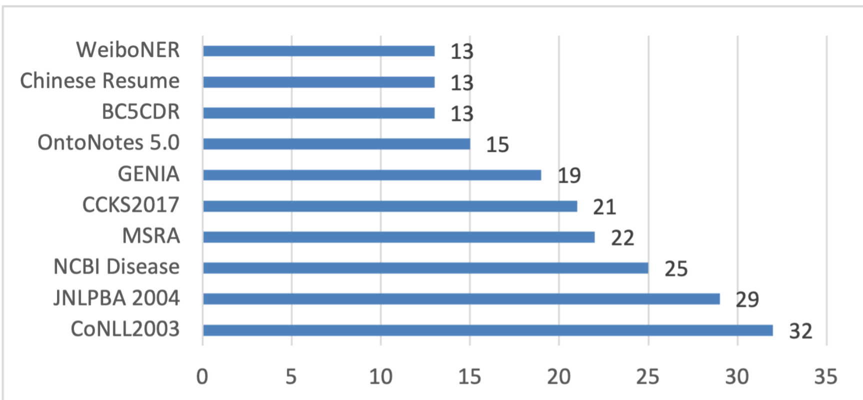

Figure 10. Popular NER datasets.

Chang et al. [89]. Its advantage is the gold-standard dataset used by NER research to benchmark the proposed NER performance evaluation [4]. Another CoNLL2003 dataset advantage is the balanced annotation of PER (10059), LOC (10.645), MISC (5062), and ORG (9323) entities [90]. Wang et al. [91] proposed the Adversarial Trained LSTM-CNN (ASTRAL) method to produce a robust NER system. The test results using the CoNLL2003 dataset showed the highest F-Score value of 93.32, compared to the OntoNote5.0 dataset of 89.44 and WNUT-17 of 49.72. The result seems influenced by the ratio between the size of the corpus and the number of entities. The OntoNotes5.0 dataset has more tokens and entity types but fewer entities than ConLL2003 and WNUT-17, as

Table 3. Dataset comparison statistic.

Dataset	Tokens	Entity frequency	Entity type
CoNLL-2003	23499	11.6%	4
OntoNotes 5.0	81828	7.5%	18
WNUT-17	3160	5.9%	6

Table 4. Data used based on languages.

Language	Amount
English	204
Chinese	86
Arabic	10
Indian	9
Spanish	7
Korean	7
Portuguese	3
Persian	3
Turkish	2
Indonesian	2

shown in Table 3. The WNUT-17 dataset comes from various sources: Twitter, comments on YouTube, Reddit, and Stackexchange. While OntoNotes 5.0 comes from the Linguistic Data Consortium [91]. Other experiments using the CoNLL2003 dataset were conducted by Lample et al. [53], Chiu and Nichols [92], Aguilar et al. [93], Peters et al. [94], Clark et al. [95], Devlin et al. [96], and Akbik et al. [97] and also showed much higher F-Score values than the OntoNote 5.0 and WNUT-17 datasets [91].

Numerous studies on the medical and health domain constitute many datasets in that domain, such as JNLPBA, CCKS, NCBI, BioCreative, and GENIA. The introduction of disease entities (NCBI), DNA cell entities (JNLPBA), disease symptom entities (CCKS), gene entities (BioCreative), and viral entities (GENIA) has become the benchmark for NER research in the medical and health fields. Recognition of these entities becomes input at later stages of NLP, such as disease detection and guidance on prescribing patients [16]. There are so many datasets in the medical and health domain that building the NER COVID-19 dataset is an opportunity and urgent to work on. The role of NER as a preprocessing stage in NLP research has a strategic position to be realized immediately.

Using public rather than private datasets shows enthusiasm for NER research on improving methods and algorithms. Private datasets are generally utilized for NER implementations in specific application domains, such as automotive [98], geology [48], agriculture [99], and law [83]. A public dataset is also chosen for benchmarking with a private dataset. Although some researchers use private datasets, they still use them as a comparison, as was done by Ekbal and Saha [34], Liu et al. [100], Kim et al. [101], and Kim et al. [102]. The proposed method can be more reliable when using public dataset testing.

As seen in Table 4, most of the datasets are in English, Chinese and Arabic. NER researchers mostly use English as a dataset for various reasons. Firstly, English is the international language with the most speakers. Second, preprocessing tools (stemming, POS-Tagging, lemmatization, parsing, stopword removal, chunking) are widely available in English. Third, because NER's primary mission is to develop methods, creating datasets in languages other than English is not a significant concern. Fourth, there are still few references or scientific publications that use non-English languages. As shown in Table 4, based on 345 articles, 204 articles used an English dataset, 86 articles used a Chinese dataset, ten articles used the Arabic dataset, nine titles in Indian, 7 in Spanish and Korean, respectively, and three articles in Portuguese and Persian. It is challenging for NER researchers to develop a corpus in low-resources languages.

3.5. NER research application domain

NER application covers both general and specific domains. NER, in a general domain, is responsible for recognizing entities such as a person (PER), location (LOC), and organization (ORG) recognition. As seen in Table 5, there are 137 of 190 titles in a general domain. NER in specific domains include medical and health, biomedical and chemical, network and security, biology, chemistry, geoscience, business and economics, history and culture, sport science, military, and agricultural. Entities in the farming domain are such pests and diseases [99], geology (e.g., rock, stratum, toponym) [48], sports science (e.g., competition name, level of competition, match time) [103], and military domain (e.g., weapons, mission, location, organization) [104]. However, there are some studies in several fields at once, such as those conducted by Zhong et al. [87], Johnson et al. [105], Jin et al. [106], Ekbal and Saha [107], Xu et al. [108],band Song et al.[109].

Publication in the medical and health domain reached 36%. Medical and health fields have become popular domains in NER research because of many medical records data in various hospitals. These data can support doctors in determining the right patient drug prescription [229]. Even if expert reviews, a massive medical record will be very long and tiring. Artificial intelligence has become a solution for processing patient data more quickly

Table 5. NER research domain.

Domain	Research
General	[12, 31, 32, 34–36, 39, 50–52, 64, 65, 72, 80, 84, 86–89, 91, 100, 101, 105, 106, 110–148]
Medical and health	[13, 30, 33, 37, 38, 43–46, 49, 56, 60–63, 69–71, 77–79, 85, 102, 149–203]
Biomedical and chemical	[14, 41, 204–207]
Network and security	[57, 208–213]
Biology	[59, 214–220]
Chemistry	[58, 74, 221–224]
Geoscience	[47,48, 225]
Business and economics	[40, 81, 226]
History and culture	[316-320]
Agriculture	[99, 321-323]
Law	[83, 227]
Social media	[108, 109]
Automotive and engineering	[98, 325-328]
Military	[104]
Neuroscience	[228]
Sport science	[103]
System and software	[73]

and accurately. NER in the medical and health domain identifies important entities such as disease symptoms and drug prescriptions [62, 63, 151, 203, 230].

NER researcher enthusiasm in the medical and health domain is because it directly contributes to human life. Modern human lifestyle is constantly changing, which affects their health. The COVID-19 pandemic increases the chance for NER researchers to identify entities related to symptoms, vaccines, organizations, number of sufferers, patient status, organizational entities, health worker entities, hospital location entities, and other related entities [231–233]. Its results can be valuable for different NLP tasks, such as text summarization, question answering, chatbots, machine translation, and information retrieval related to handling the COVID-19 pandemic.

Nozza et al. propose the L2AWE method as an ontological entity mapping schem.153 L2AWE (Learning To Adapt with Word Embeddings) aims to adapt the NER system trained on the source classification scheme to a given target. L2AWE does not need to retrain the underlying NER model to match the new target generic classification scheme. Because the amount of training data requires a large volume, Nozza et al. [153] use three pre-training models simultaneously, namely Wiki2Vec, GoogleNews (W2V), and BERT. Most domain adaptation methods focus on one domain without considering all annotations. When there is a shift between domains too large, the decline in NER performance is even more significant. This weakness is corrected by the method proposed by Li et al. taking multi-aspect relevance learning (MARL) [152]. The fine-tuning process uses BERT with the lost function formula, as shown in equation 1.

$$L_{NER} = -\frac{1}{m+n}\sum_{j=1}^{m+n} y_j log\hat{y}_j \tag{1}$$

Furthermore, by calculating the vector mean at the sentence level, the domain-level representation uses the cosine equation of the two domain-level vectors to find the domain distance.

$$Qs_i = \frac{1}{m_i}\sum_{j=i}^{m_i} BERT(x_j) \tag{2}$$

$$d_{s_a,s_b} = similarity(q_{s_a}, q_{s_b}) \tag{3}$$

The relevance learning sample level uses domain labels to design a binary domain discriminator classifier, determining whether the sample belongs to the target domain. That output value determines the proximity of the model to the target domain. The general entities can be the baseline in multidomain research by looking at the size of the public domain, as shown in Table 5

3.6. Features Extraction

NER dataset is a collection of texts or corpus. Computers cannot read a text like humans but can only read numbers. Therefore, a corpus needs to be represented in numbers. Various representation and feature extraction techniques are shown in Figure 11. Popular feature extraction techniques include word embedding and character embedding, POS-Tag and word2vec, n-gram, chunk, and skip-gram.

Word embedding is converting words into numbers in the form of vectors or arrays [76]. Word embedding ranks at the top of feature extraction techniques. It relates to the deep learning approach that NER researchers have widely used in recent years. The deep learning approach uses word embedding as a feature extraction technique. It has several advantages such as: a) it can reduce the size of a vector or array to smaller dimensions when compared to one-hot-encoding [234]; b) it does not require tagging during the training stage [235]; c) it can give a richer semantic meaning [236]. However, word embedding also shows some drawbacks: out-of-vocabulary, antonymy, polysemy, and biased embedding [237]. Regarding this deficiency, it is an opportunity for future researchers to improve word embedding to provide maximum text vectorization performance.

The next feature is POS-Tagging. It is responsible for annotating words/tokens according to part-of-speech [238]. Part-of-speech is a class of words with grammatical properties that play a role in the syntax of similar sentences [237]. POS-Tagging has similarities to NER, which tag the corpus. Labels on POS Tags use nouns, verbs, adjectives, adverbs, exclamations, and others [239]. Labels on NER are defined based on entities such as a person (PER), location (LOC), organization (ORG), and others. POS tags can be implemented in any domain

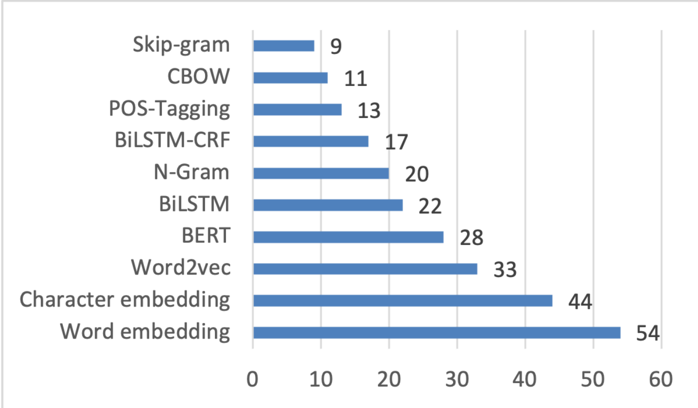

Figure 11. Various feature extractions on NER.

because the POS structure has the same pattern in all disciplines. NER is suitable for identifying entities in a particular domain because the entities can be defined during the corpus annotation process. For example, medical entities are different from entities in the automotive area. NER uses POS Tag to identify nouns in the corpus for determining the institution [72] person, location, etc. Lin et al. [240] state syntactic information, including POS Tag, explicitly as a contextual feature of each word in a sentence.

The number of NER researchers using word embedding, word2vec, character embedding, GloVe, and BERT feature extraction cannot be separated from the popularity of deep learning. BERT provides better word representation performance when compared to word2vec, GloVe, and ELMo [241]. Word2vec is a machine learning model for producing word vector representations based on the context of the words in the corpus. Word2Vec uses a neural network model to learn vector representations of words that reflect deep semantic relationships in the cor-pus. These representations are then used in various NLP tasks, including NER. Word2vec, first introduced by Mikolov in 2013, uses two architectures, Continuous Bag-of-Word (CBOW) and Skip-gram [242]. CBOW predicts the wt target word based on its context, while Skip-gram predicts the word context based on the target word. The vector representation of words is obtained from both architectures by updating the parameters through a neural network-based training process. The context here is the words around wt, namely wt-2, wt-1, wt+1, and wt+2, two words to the left of the target and two words to the right [66].

To predict w_t, CBOW uses the probability formula, as shown in formula 4.

$$P(w_t|w_{j(j|j-t|\leq l,j\neq t)}) = Softmax\left(M\left(\sum_{|j-t|\leq l,j\neq t} W_j\right)\right) \tag{4}$$

where, $P(w_t|w_{j(|j-t|\leq l,j\neq t)})$ is the word probability wt, l is the training context measure, M is the weighting matrix on $\mathbb{R}^{|V|m}$, V is the number of vocabulary words, and the m dimensions of the word vector. The CBOW model is optimized by minimizing the sum of the negative log probabilities, according to the formula 5.

$$\mathcal{L} = -\sum_t logP(w_t|w_{j(|j-t|\leq l,j\neq t)}) \tag{5}$$

We can adjust the number of window sizes as needed. If l increases the window size, the accuracy will increase, but the training time becomes longer, and vice versa.

The skip-gram model, opposite CBOW, predicts the word context based on the target word using formula 6.

$$P(w_j|w_t) = Softmax(Mw)(|j-t|\leq l, j\neq t) \tag{6}$$

where, $P(w_j|w_t)$ is the word context probability w_j given w_t, and M is the weighting matrix. The calculation of the loss function in the skip-gram uses the formula 7.

$$\mathcal{L} = -\sum_{t} \sum_{|j-t|\leq l, j\neq t} P(w_j|w_t) \tag{7}$$

The implementation of Word2vec in the proposed model aims to reap the advantages of Word2vec, namely: 1) computational efficiency because it only requires the context around the target word to make predictions, 2) being able to learn word representations in a large corpus by reducing the dimensions of the vector representation, and 3) being able to express semantic relationships between words in the form of vector operations, for example, ”king” - ”man” + ”women” will be predicted to become ”queen”.

With the increasing number of deep learning approaches implemented by NER researchers, BERT is a promising option. The advantage of BERT is that it can detect words in the OOV (out of vocabulary) category compared to Bi-directional long short-term memory with a conditional random field (BiLSTM-CRF) [196]. BERT is designed for in-depth two-way train representation by putting together the left and proper context across all layers. A pre-trained BERT can be well-tuned to create a competitive model for various tasks [244]. Since the appearance of word2vec in 2013, until now, it is still widely used by NER researchers to represent text features. Word2vec representations are great for capturing syntax and semantic forms of an order of language, and a relation-specific vector characterizes each relationship. However, with the increasingly widespread implementation of deep learning, word2vec is slowly shifted by the presence of BiLSTM and BERT, which are improvements from LSTM RNN. One of the advantages of BiLSTM is that it can store long memory from two directions [143]. and better capture the context of the sentence better when compared to vanilla RNN or LSTM. BiLSTM is a development of LSTM in two directions. The LSTM architecture has four gates: 1) learn gate, 2) forget gate, 3) remember gate, and 4) use gate, as shown in figure 12.

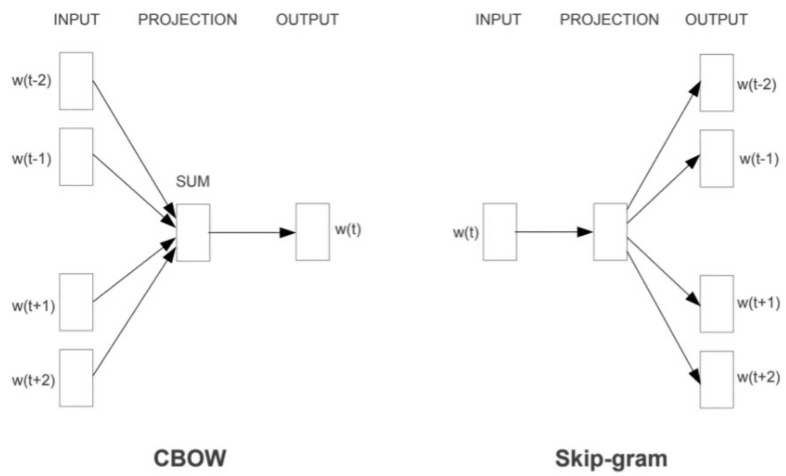

Figure 12. CBOW architecture predicts the current word based on context, and Skip-gram predicts surrounding words given the current word [243].

The learn gate serves as a memory to store short-term memory results coupled with input values. The forget gate serves as the long-term memory of the token sequence. The remember gate serves as a memory that stores short-term and long-term memory. This gate will later become a new long-term in the following sequence. The use gate is tasked with storing short-term and long-term memory, which subsequently becomes the new short-term memory for the next sequence.

Recurrent neural networks are designed to handle input sequences in variable sequences but cannot model the long-term dependencies of those input sequences. Although RNN output is based on previous calculations, long-term dependencies are still a challenging problem in RNN training [347]. LSTM is a type of RNN that mitigates this problem by keeping memory cells that serve as summaries of previous elements of the input sequence [348]. LSTM is a development of the hidden layer of the RNN into a more detailed unit called a cell. Inside the hidden layer, a cell consists of several gates that can be controlled to store or erase memory information throughout the sequential input [349]. Therefore, LSTM improves the ability to maintain remote context information. Longer contextual information can help models study semantics more precisely.

As shown in Figure 13, the elements in an LSTM cell have a gate it (input gate), ft (forget gate), and ot (output gate). While W and b are, respectively, weight values and biases in vectors. Forget gate f_t, with the output of the

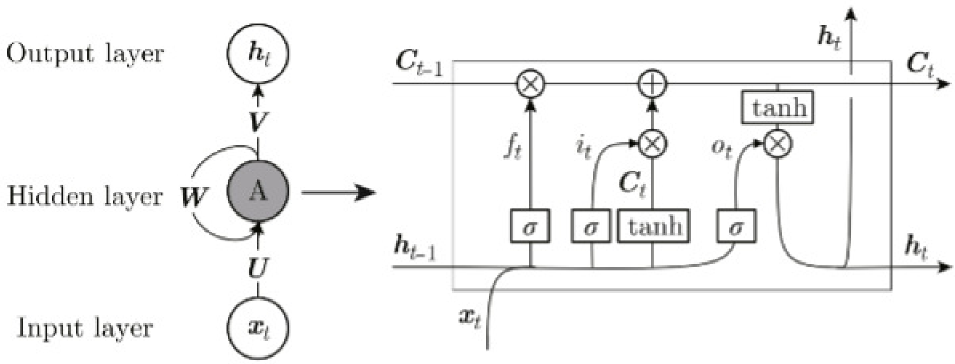

Figure 13. The smallest element in the LSTM, called a cell.

previous cell h_t and current input x_t in connection to prior circumstances $c_t - 1$ determine what proportions to forget some information. The output of the forget gate is calculated as a sigmoid value σ, as seen in equation 8.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{8}$$

The input gate determines which values should be updated considering the current input, x_t, candidate value c_t, and the previous state h_{t-1}. These values are updated with formulas 9, 10, and 11.

$$\begin{aligned} i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i), & (9)\\ \tilde{C}_t &= tanh(W_c)[h_{t-1}, x_t] + b_C), \text{and} & (10)\\ C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t & (11) \end{aligned}$$

Where the * is the multiplication element, the ht value is calculated using the output gate by combining currents C_t, h_{t-1}, and x_t, as seen in equations 12 and 13.

$$\begin{aligned} o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o), & (12)\\ h_t &= o_t \, tanh \, C_t & (13) \end{aligned}$$

The LSTM addresses significant gradient reductions during training procedures through gate mechanisms. BiLSTM consists of two directions LSTM, forward and backward. LSTM ahead performs the hidden state to the right ($\overrightarrow{h_1}$,$\overrightarrow{h_2}$,...,$\overrightarrow{h_n}$), with input from x_1 to x_n. Meanwhile, LSTM retreats to carry out the process of calculating the hidden state from the opposite direction ($\overleftarrow{h_1}$,$\overleftarrow{h_2}$,...,$\overleftarrow{h_n}$), with inputs in descending order from xn ke x1, as shown in figure 13. Depending on the application of the LSTM, in one case, it may require an output sequence corresponding to each element in the sequence, or a single output encapsulates the entire sequence. In the previous case, the sequence of output ($\overrightarrow{h_1}$,$\overrightarrow{h_2}$,...,$\overrightarrow{h_n}$) of LSTM is obtained by combining the hidden state of forwarding and backward LSTM for each element, $\overrightarrow{h_t} = (\overrightarrow{h_t}; \overleftarrow{h_t})$ for $t = 1, \ldots, n$. In another case, the output is obtained by combining the last hidden state of the forward and backward LSTM, $\overrightarrow{h} = (\overrightarrow{h_n}; \overleftarrow{h_n})$ [350].

3.7. NER approaches

Initially, researchers used traditional approaches, i.e., rule-based, dictionary-based, and knowledge-based. Currently, many studies use machine learning approaches, including deep learning. As the oldest approach, rule-based is now being abandoned because it has several disadvantages, such as depending on manual rules based on textual patterns [125,243]. The subsequent weakness is time-consuming and labor-intensive maintenance, especially if the linguist's knowledge and background are insufficient [125]. On the other hand, machine learning in NER can recognize entities better than that produced by traditional rule-based and dictionary-based [166].

Deep learning and machine learning are the most widely used, as shown in Figure 14. However, some researchers

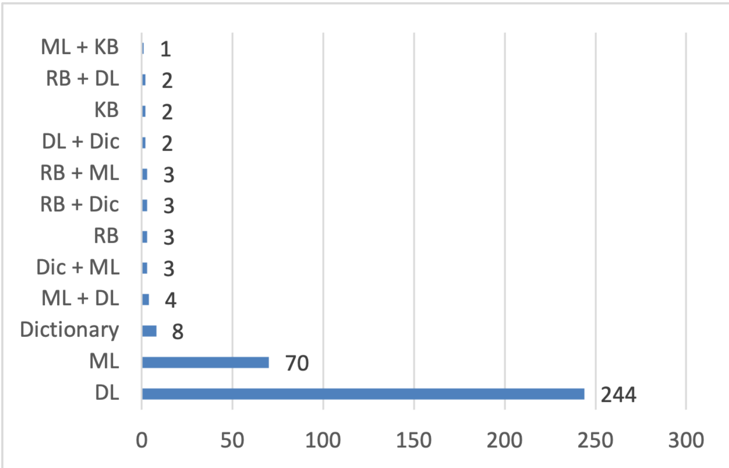

Figure 14. Various approaches in NER research.

combine rule-based or dictionary-based with machine learning [77, 110, 139, 155, 175, 178, 217], and deep learning [65, 156, 157, 179]. The knowledge-based approach is the least implemented approach, namely by Al-Jumaily et al. [133], Cabot et al. [61], followed by Munkhdalai et al. [204], which combines knowledge-based and machine learning. As shown in Table 6, each approach has its strengths and weaknesses. The combination of several approaches can improve performance, with the philosophy of maximizing the strengths and reducing the disadvantages of each. For example [155], combining dictionary-based, rule-based, and machine learning can improve the quality of NER data input, especially for Chinese electronic medical records (CEMR). Deep learning, a weakness in a few datasets, can be overcome using a dictionary-based method to improve performance; -4% compared to using only CRF. Performance is further enhanced by combining deep learning with rule-based to perform entity extraction. Here, researchers have vast opportunities to improve methods by combining several approaches and considering each approach's weaknesses and strengths, as shown in Table 6.

Xu et al. [157] proposed a hybrid model SBLC (Semantic Bi-Directional LSTM CRF) which combines rule-based and deep learning approaches. The rule-based approach used is Ab3P (Abbreviation Plus Pseudo-Precision) which is in charge of detecting abbreviations in the corpus [157]. The precision of each rule is estimated by applying randomized data (pseudo-precision). The rule-based approach embedded in deep learning improves NER performance with a precision value of 0.866, recall of 0.858, and F1-score of 0.862. The rule-based approach, although said to be the oldest approach, can provide improved performance. Hsu and Kao utilize curators to detect entities based on rule-based rankings. The results of the curation are then rule-based and then used to develop a dictionary-based NER system and machine learning using CRF. The test results show a significant increase in the F1 value from 0.54 to 0.61.

Neural network (NN), the pioneer of deep learning, has been found since the 1980s. However, the popularity slowed down because, at the time, computation speed was still slow, the dataset was small, and it was not thought about doing an in-depth training dataset with many layers [246]. NN, which later developed into a deep neural network (DNN), is currently a new approach widely used in NER research because it has many advantages. One

Table 6. Advantages and disadvantages of the NER approach.

Approaches	Advantage	Disadvantage
Knowledge base	It is a good performance for specific domains based on domain knowledge.	Use is limited to specific domains only. It takes a long time to build representative knowledge.
Dictionary-based	Fast process with precision results. By comparing the target entity with the vocabulary in the dictionary.	It requires great effort to build a dictionary. Weak on adaptability
Rule-based	Rules are built on the expertise of linguistic experts. They can deliver high performance.	Different expert knowledge can lead to ambiguity. Weak adaptability.
Machine learning-based	Continuous self- improvement. Quickly identifies trends and patterns. No human intervention.	High error-susceptibility. Time-consuming. The result depends on data quality.
Deep learning-based	Good performance for large datasets.	Computing is complex and takes a long time. Requires huge resources.

of the advantages of DNN is automatically generating features from a corpus [73]. DNN can also describe local data characteristics very well, and the merged layer can extract the most representative part of local features [88]. Increasingly large NER datasets with good annotation quality make deep learning performance better [101, 245], Murthy et al. [112] added the POS-Tag feature to improve NER performance with a deep learning approach. Combining deep learning and machine learning could provide better prospects for future NER research.

3.8. *Various NER methods*

The BiLSTM-CRF combination method is the most widely used choice, followed by the conditional random field (CRF) method, as shown in Figure 15. CRF method was first proposed by Lafferty et al. [248] for sequential data segmentation and labeling based on probabilistic models. CRF was widely used, including in the NER topic. The advantages of CRF are that it can estimate conditional probability distributions over labeled sequences and allows information about decision confidence to be used by other components in text processing [247, 248]. CRF method is also widely combined with other machine learning techniques, such as SVM [215,217], ME [32,36,107], and RNN [205,207].

Recurrent neural networks are designed to handle input sequences with variable sequences but cannot model the long-term dependencies of these input sequences. Even though the RNN output is based on previous calculations, long-term dependencies are still a challenging problem in RNN training [251]. LSTM is a type of RNN that reduces this problem by preserving memory cells that act as summaries of the previous elements of the input sequence [252]. LSTM is the development of the RNN hidden layer into a more detailed unit called a cell. Within the hidden layer, a cell consists of multiple gates that can be controlled to store or delete memory information along sequential input [157]. Therefore, LSTM improves the ability to maintain remote context information. Longer contextual information can help the model learn semantics more precisely.

As shown in Figure 15, the elements in an LSTM cell have i_t (input gate), f_t (forget gate), and o_t (output gate). While W and b are weighted and biased, respectively, in the vector. The forget gate f_t, with the previous cell output h_{t-1} and the current input x_t to the previous state c_{t-1} determines what proportion to forget some information. The output of the forget gate is calculated as the sigmoid value of $B\sigma$.

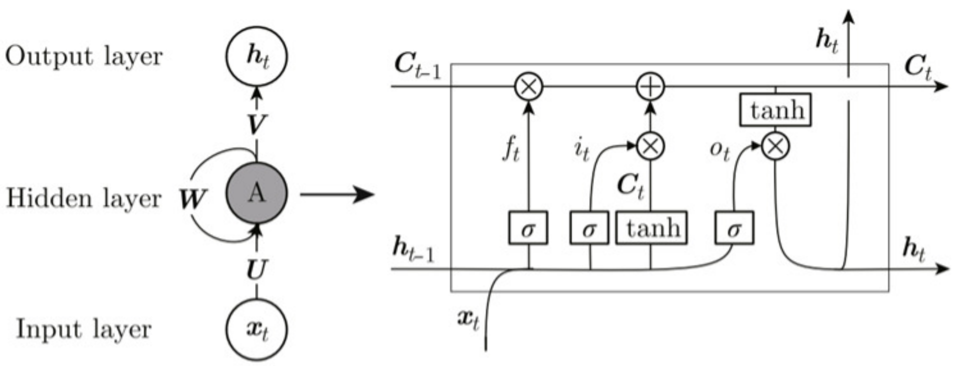

Figure 15. The smallest element in the LSTM is called a cell.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{14}$$

The input gate determines which value to update considering the current input x_t, the new candidate value c_t, and the previous state on h_{t-1}. Update these values using formulas 15 to 17.

$$\begin{aligned} i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i), && (15)\\ \tilde{C}_t &= tanh\,(W_c[h_(t-1), x_t] + b_C), \text{and} && (16)\\ C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t, && (17) \end{aligned}$$

where the * sign represents the multiplication element and the h_t output is calculated using the output gate by combining C_t, h_{t-1}, and x_t currents.

$$\begin{aligned} o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o), && (18)\\ h_t &= 0_t * tanh\, C_t && (19) \end{aligned}$$

This LSTM is also designed to overcome significant gradient reduction during the training procedure via the gate mechanism. BiLSTM consists of two forward and reverse LSTM directions. LSTM forward performs the hidden state calculation process to the right ($\overrightarrow{h_1}$,$\overrightarrow{h_2}$,...,$\overrightarrow{h_n}$), with input from x_1 to x_n. Whereas LSTM backward performs the hidden state calculation process from the opposite direction, namely to the left ($\overleftarrow{h_1}$,$\overleftarrow{h_2}$,...,$\overleftarrow{h_n}$), with inputs in reverse order from xn to x1. Depending on the implementation of the LSTM, in one case, it may require an output sequence that corresponds to every element in the sequence or a single output that encapsulates the entire sequence. In the previous case, the output sequence ($\overrightarrow{h_1}$,$\overrightarrow{h_2}$,...,$\overrightarrow{h_n}$) of the LSTM is obtained by combining the hidden state of the forward and backward LSTM for each element, i.e., $\overrightarrow{h_t}$=($\overrightarrow{h_t}$;$\overleftarrow{h_t}$) for $t = 1, \ldots, n$. For other cases, the output is obtained by combining the last hidden state of the forward and backward LSTM, i.e., $\overrightarrow{h} = (\overrightarrow{h_n}; \overleftarrow{h_n})$[252].

CRF is a conditionally trained model that can efficiently function with various non-independent features. CRFs have specific properties of considering surrounding instances, unlike discrete classifiers. When predicting label order for input sample sequences, CRF assumes contextual features [21]. CRF usually calculates the transition probability between labels and the likelihood of the tag's entire sequence [253].

CRF is combined with Bi-LSTM by putting it at the end of the NER algorithm to estimate that the entity label sequence is correct [254]. For example, given input token $x = \{x_1, \ldots, x_T\}$ and the score of the output matrix score [fil], score for labeling output $y = \{y_1, \ldots, y_T\}$ is given by: $s(x,y) = \Sigma_{t=1}^{T}(A_{y_{t-1},y_t} + f_{t,y_t})$, where A is the matrix of $L \times L$ parameters for transition between output labels. CRF then produces correct labeling possibilities by normalizing this score across all possible output labeling: $log\,P(y|x) = s(x,y) - logsumexp\,s(x,y')$. The log normalization referred to here is: $\underset{y'}{logsumexp\,s(x,y')} = log\,\Sigma_{y'}\,exp\,s(x,y')$ [248]. CRF layer is designed to

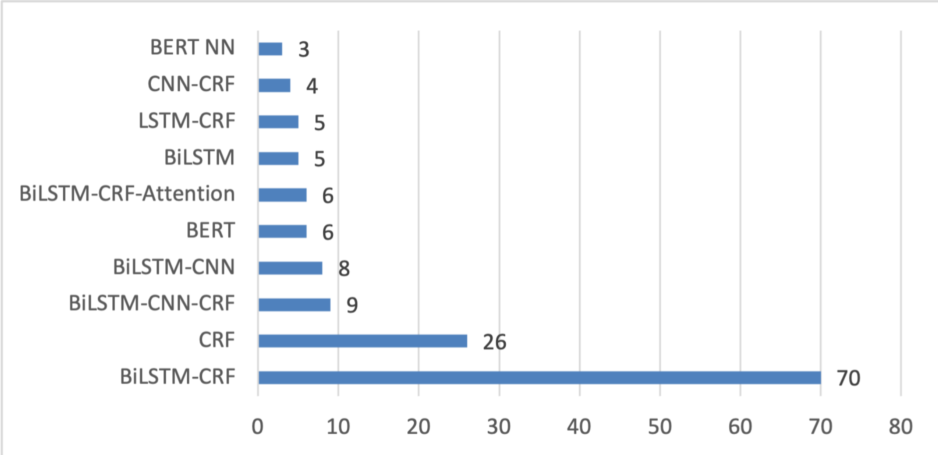

Figure 16. Several methods for NER task.

select the best tag sequence from all possible tag sequences considering the output of Bi-LSTM and correlation between adjacent tags [255]. The CRF input is sequential data, which considers the previous context when making predictions on the data state. This behavior modeling uses the feature function, which has several input values, namely:

1. vector input set, X;
2. position i of the predicted data state;
3. data label i-1 in state x; And
4. label data state i on x.

where the feature function is defined as $f(X, i, l_{i-1}, l_i)$. The purpose of feature functions is to express some characteristic of the sequence represented by the data state. Each feature function is based on the label of the previous word and the current word and is in the form of 0 or 1. Then assign each feature function a set of weights (lambda), with the CRF distribution probability learning algorithm as formulas 20 and 21.

$$P(Y, X, \lambda) = \frac{1}{Z(X)} exp\{\sum_{i=1}^{n} \sum_{j} \lambda_j f_i(X, i, y_{i-1}, y_i)\} \tag{20}$$

where

$$Z(X) = \sum_{y^{\epsilon} y} \sum_{i=1}^{n} \sum_{j} \lambda_j f_i(X, i, y'_{i-1}, y'_i) \tag{21}$$

Calculating the lambda parameter uses the maximum estimation likelihood by taking the negative log of the distribution to make it easier to calculate the partial derivatives, as in Formula 22, the negative log-likelihood of the CRF probability distribution.

$$\begin{aligned} L(y, X, \lambda) &= -log\left\{\prod_{k=1}^{m} P(y^k|x^k, \lambda)\right\} \\ &= -\sum_{k-1}^{m} log\left[\frac{1}{Z(x_m) exp\left\{\sum_{i=1}^{n} \lambda_j f_j(X^m, i, y_{i-1}^k, y_i^k)\right\}}\right] \end{aligned} \tag{22}$$

To apply maximum likelihood to a negative log function, you can use argmin, because minimizing negatives get the maximum value. The lambda partial derivative function can be used to find the minimum value, as in formulas 23 and 24.

$$\frac{\partial L(X,y,\lambda)}{\partial \lambda} = \frac{-1}{m}\sum_{i=1}^{n} F_j(y^k,x^k) + \sum_{k=1}^{m} p(y|x^k,\lambda)F_j(y,x^k) \tag{23}$$

where

$$F_j(y,x) = \sum_{i=1}^{n} f_i(X,i,y_{i-1},y_i) \tag{24}$$

The derivative formula above is used as a step in the gradient descent stage, where the gradient updates the parameter values iteratively until it reaches a convergent value. Where is the updated gradient descent formula used in CRF as in Formula 25

$$\lambda = \lambda + \alpha \left[\sum_{k=1}^{m} F_j(y^k,x^k) + \sum_{k=1}^{m} p(y|x^k,\lambda)F_j(y,x^k)\right] \tag{25}$$

Thus, in general, CRF consists of three steps: 1. define the required feature functions (formulas 20 and 21), 2. initialize the weights to random values (formulas 22 and 23), and 3. apply gradient descent iteratively so that the parameter values, in this case, the lambda values, reach convergence (formulas 24 and 25).

The CRF function is similar to logistic regression because it uses a conditional probability distribution with input data in sequences or sequences. As stated in the definition, the first stage of probability calculation is related to the CRF model and input/output order needed to calculate the conditional probabilities of the marginal distribution $P(Y_i = y_{i|x})$ at a node and the marginal distribution $P(Y_{i-1} = y_{i-1}, Y_i = y_{i|x})$. In the second stage, based on the CRF parametric formula, two parameters λ_k and μ_j need to be studied using a training dataset to obtain a $\hat{P}(Y|X)$ model, which aims to find the maximum/highest probability in a particular sequence. After training the CRF parameters based on the training dataset, we get a pleasant basic model $\hat{P}(Y|X)$. In this third phase, we need to calculate the output order $\hat{Y}$ that maximizes the conditional probability $\hat{P}(Y|X)$ based on the input x. The algorithm to determine the output generally uses Viterbi [256].

3.9. Learning Rate

Learning rate is one of the methods used to control how many changes will be made to the model parameters in each iteration during the training process. The learning rate determines how fast or slow the model learns from the training data [257]. If the learning rate is too low, the model will learn slowly and need more iterations to achieve optimal results. However, if the learning rate is too large, the model may ”skip” the minimum point in the optimization process and fail to achieve good convergence.

Determining the right learning rate can use several techniques: trial and error, adaptive, and automatic. The trial and error method for determining the learning rate is by manually determining the learning rate value, then looking at the performance of NER. Researchers can start with a large learning rate, for example, 0.1, and gradually reduce it to see changes in model performance. If the model does not converge or the results are bad, the learning rate can be reduced gradually. The second technique is adaptive, where the learning rate value is adjusted based on information from the gradient [258]. This adaptive model can use Adam's optimization algorithm, which combines the gradient's first and second moment estimates to adapt the learning rate. Besides Adam's algorithm, we can use Adagrad and RMSProp. The third way is to use an automatic tuning algorithm such as Grid Search or Random Search to find the optimal learning rate that combines with other hyperparameters. Like the experiments conducted by Smith, the learning rate that provides the most optimal performance is Nesterov [259], followed later by Adam, RMSProp, AdaGrad, and AdaDelta [257]. Meanwhile, the comparison of learning rates carried out by Warto shows that the optimal learning rate for NER is the Adam algorithm [260], compared to SGD and AdaDelta. These results differ highly depending on the dataset type and experimental device used. Nesterov introduced a method for accelerating gradients in four steps [259]. First, at each iteration t, we have the model parameter w and the function's gradient concerning that parameter, which is notated in Formula 26.

$$\nabla J_t(w) \tag{26}$$

The second step is updating the model parameters using the momentum obtained from the previous iteration, using the formula 27.

$$\tilde{w}_{t+1} = w_t + \mu \dot{v}_t \tag{27}$$

where μ is the momentum which is a hyperparameter that controls the extent to which momentum is used in the update, usually a value between 0 and 1.

The third step calculates the gradient at the current position using the formula 28.

$$\tilde{\nabla}_t(w) = \nabla J_t(\tilde{w}_{t+1}) \tag{28}$$

It then calculates the final update of the model parameters using the gradients in formula 29.

$$w_{t+1} = \tilde{w}_{t+1} - \eta \cdot \tilde{\nabla} J_t(w) \tag{29}$$

where η is the learning rate which controls how much changes are made to the model parameters.

AdaGrad individually adapts the learning rates of all model parameters by scaling them inversely to the square root of the sum of all historical squared values of the gradients [261]. AdaGrad uses a vector v to store the sum of the squares of the previous gradients, as in the formula 30.

$$v_t v_{t-1} + (\nabla J_t(w))^2 \tag{30}$$

Then the adaptive learning rate is calculated for each wi parameter using the square root of the previous gradient accumulation, using the formula **??**.

$$Learning\, Rate_i = \frac{(learning\, rate)}{\sqrt{(v_{t,i} + \epsilon)}} \tag{31}$$

Where $v_t{,}i$ is the i$t - i$ element of the v_t vector, namely the accumulation of the previous gradient squares for the ith parameter. At the same time, ϵ is a small value added to maintain numerical stability, for example, 10^{-8}. Then the parameters are updated using the gradient adaptive learning rate using the formula 32.

$$w_{t+1} = w_t - learning\, rate_i \cdot J_t(w) \tag{32}$$

AdaGrad calculates and updates the learning rate based on accumulating previous gradients. This technique effectively adjusts the learning rate for each parameter based on the previous gradient history.

RMSProp modifies AdaGrad to perform better in settings by converting the accumulated gradients into an exponentially weighted moving average. RMSProp uses the vector v to store the exponential average of the squares of the previous gradients, which is denoted in the formula 33.

$$v_t = \beta v_{t-1} + (1 - \beta)(\nabla J_t(w))^2 \tag{33}$$

Where $\nabla J_t(w)^2$ is the square of the gradient $\nabla J_t(w)$, β is the hyperparameter that governs the contribution of the latest gradient in the calculation of vector v, the value is between 0.9 – 0.99. The adaptive learning rate is calculated for each parameter wi using the square root of the previous gradient exponential average, as in Formula 32.

Adam (Adaptive Moment Estimation) was introduced by Kingma and Ba in 2014 as an alternative to determining the learning rate. Adam's method combines the momentum and RMSProp methods to calculate model parameter changes [262]. First, Adam calculates the momentum mt based on the gt gradient using Formula 34.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{34}$$

Then calculate RMSProp vt based on the squared gradient g_t^2 using the Formula 35.

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{35}$$

Calculation of the adjusted momentum estimate $\tilde{m}_t$ and the adjusted RMSProp estimate $\tilde{v}_t$, as shown in formulas 36 and 37.

$$\tilde{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{36}$$

$$\tilde{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{37}$$

Where β_1 and β_2 are the momentum reduction factor and RMSProp, by using the momentum estimation and RMSProp estimation as formulas 36 and 37, Adam can adjust the learning rate adaptively for each model parameter so that it can help the algorithm quickly learn when approaching the minimum loss function and slow down when approaching the optimum point.

The next RMSProp variant is AdaDelta, which Zeiler developed with additional adjustments [263]. AdaDelta replaces the global learning rate with the difference from the previous model parameters to calculate the adaptive learning rate. RMSProp $E[g^2]_t$ calculation is based on the g_t^2 gradient, as in formula 38.

$$E[g^2]_t = \rho E[g^2]_{t-1} + (1 - \rho) g_t^2 \tag{38}$$

where ρ is the RMSProp reduction factor.

Then the adjusted model parameter changes are calculated using the formula 39.

$$\delta w_t = \frac{-\sqrt{(E[\delta w^2]_(t-1) + \epsilon)}}{\sqrt{(E[g^2]_t + \epsilon) \cdot g_t}} \tag{39}$$

Then the model updates the adjusted model parameters, namely. $w_{t+1} = w_t + \delta w_t$. Of the various learning rate optimization methods described above, the Adam and SGD methods are the most widely used by NER researchers. Adam can provide faster convergence and be adaptive to changes in learning rate. In contrast, SGD is more efficient for large datasets with lower computational speeds.

3.10. NER performance evaluation technique

An evaluation of the proposed NER method or technique is necessary to measure the performance of the proposed method or technique. Articles that do not demonstrate the results of evaluating the performance of the proposed approach are not included in this review. Most papers we reviewed used precision, recall, and F1-score or F-measure evaluation techniques. A combination of these three measurements is the most preferred choice, as shown in Figure 17. However, many researchers also measure evaluation using only F-Score. It is understandable because precision and recall are the first steps to calculating F-score. In other words, F-score is obtained by calculating precision and recall, as shown in Eq. 27.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{40}$$

$$Recall = \frac{TruePositive}{TruePostiive + FalseNegative} \tag{41}$$

$$F1Score = \frac{2 \times recall \times precision}{precision + recall} \tag{42}$$

positive (TP) is positive data, and the predictive results are positive. True negative (TN) is negative data, and the prediction results also show negative. TP and TN state that the classification model recognizes tuples correctly, meaning that positive tuples are recognized as positive tuples, and negative tuples are recognized as negative tuples. False positive (FP) is negative data, but the prediction results are positive. False negatives (FN) are positive data, but the prediction results are negative. FP and FN stated that the classification model was misclassified, meaning that positive tuples were recognized as negative tuples, and negative tuples were recognized as positive tuples. Precision functions to measure accuracy, namely the percentage of tuples labeled positive by the classifier and the label positive.

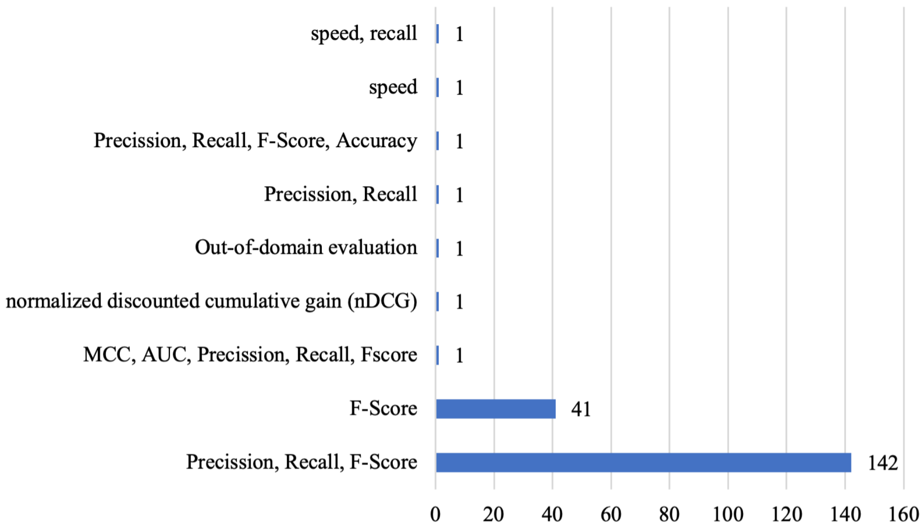

Figure 17. Various NER evaluation methods.

		Positive	Negative	
Actual	Positive	True Positive (TP)	False Negative (FN)	P
	Negative	False Positive (FP)	True Negative (TN)	N
		P’	F’	

Figure 18. Confusion matrix table.

Meanwhile, recall is used to measure completeness, the percentage of tuples labeled positive by the classifier as positive. F-Score is the harmonic mean of precision and recall. So that when the researcher only shows the F-Score value, he has calculated precision and recall values as seen in Eq. 40 and Eq. 41

Deng et al. [111] used speed evaluation to measure the speed of the filtering algorithm and the pruning technique he proposed in the dictionary-based entity extraction process. The study’s proposed method proved to be faster than the NGPP method proposed by Wang et al. [264] and ISH by Chakrabarti [265].

4. Discussion

NER is a preprocessing stage in NLP research. According to Larose [266], the preprocessing step contributes 60% of all time and effort for the entire data mining process. This large percentage opens up further research opportunities that are increasingly challenging. The current development of the NER topic cannot be separated from the deep learning trend, starting from corpus generation, preprocessing, feature extraction, etc. Deep learning promises better performance compared to other machine learning methods. However, the computational complexity of deep learning is resource-intensive. With the development of future computational speed, computational complexity constraints can be overcome.

English is the largest language dataset available for NER research because English is the most widely spoken language globally. NER research in other languages include Arabic [64, 117, 125, 132, 255], Chinese [51, 52, 105, 156, 159, 180, 256], German [269], Korean [115, 130, 196, 258] and Indian [35, 112, 119, 141]. Among the NER studies in Bahasa Indonesia have been conducted by Wintaka et al. [271], Syachrul et al. [272], Azalia,[273], Wilie et al. [274], Wibawa and Purwarianti [275], and Gunawan et al. [276] NER research on Bahasa Indonesia is up-and-coming because Indonesian speakers in 2021 will reach 260 million people. Indonesia is also one of the countries with active social media users worldwide.

Some researchers are concerned with bilingual and multilingual NER, including Li et al. [113], Xie et al. [277], Sabty et al. [278], Zafarian et al. [279], Winata et al. [280], Nguyen et al. [281], Dao et al. [282], Arkhipov et al. [283], Ni and Florian [284]. There are still few who research this area, even though languages in this world are prodigious. For example, Arkhiphov et al. [283] studied multilingual NER by performing NER tuning using the CRF-BERT model in Russian, Bulgarian, Czechoslovak, and Polish. Ni and Florian [284], utilized the Wikipedia corpus in various languages to improve NER in English, Spanish, Portuguese, Dutch, German, and Japanese. Multilingual NER research on Asian languages can also be challenging because the countries with the largest population are in Asia, such as China, India, and Indonesia.

NER performance with a machine learning approach depends on choosing the right features to create reliable learning model [21]. Preprocessing techniques such as parsing, stemming, chunking, lemmatizing, and stopword removal are the mainstays of machine learning-based NER preprocessing. As shown in Figure 7, these techniques are becoming less and less used and have shifted to tokenization and POS-Tagging. This trend parallels the increasing deep learning popularity, which can reduce long-time feature engineering [25]. The learning rate is an important parameter in deep learning algorithms that determines how many steps it takes to update model parameters during training. Choosing the right learning rate affects the success of training and the final performance of the deep learning model. If the learning rate is too high or too low, the training process is at risk of failure, and it can take a very long time to achieve good results [285].

CoNLL2003 dataset is still the benchmark for NER research, using the begin-inside-outside (B-I-O) annotation scheme. In addition to these schemes, there are several other annotation schemes, including inside-outside (IO), inside-outside-end (IOE), inside-outside-begin-end-single (IOBES), begin-inside (BI), and begin-inside-end-single (BIES). Research on annotation schemes conducted by Alshammari and Alanazi [146], shows that selecting annotation schemes affects NER performance. However, this performance is also influenced by the corpus language factor [146]. The opportunity to research these annotation schemes in multiple languages can also present challenges.

Medical and health domain is the largest sector in NER research, as shown in Table 3. Transportation, sports, advertising, military, agriculture, automotive, and geoscience domain can be promising alternatives to NER research. We argue that more research application domains imply NER is more advantageous. Therefore, there are still extensive opportunities for NER researchers to explore new fields. Some of the new areas include civil engineering, education, animal husbandry, marine, geography, forestry, and psychology. The challenge for NER in new domain applications lies in the new entities that can be detected. More and more implementation of NER research in new domains will increasingly add specific entities to that domain. In the future, a universal entity library can be built for cross-domain NER research and present a versatile NER platform. The big data trend is currently supporting the development of multi-domain NER.

The increasingly rapid development of big data, deep learning, and cloud computing has challenged NER researchers to offer a universal entity library containing various cross-domain entities. But the biggest challenge is regarding entity ambiguity. At the entity detection step, the same entity in one domain may have different meanings in other domains. For example, ”Willian P. Hobby” can be interpreted as a political figure or an airport in Texas. In the political domain, “Willian P. Hobby” is recognized as a “Person” entity, but in the transportation domain, it is recognized as an airport, a “Location” entity.

The rapid development of big data opens up opportunities for NER research. The NER dataset is not only derived from a text but can also be images, sounds, and videos. Zheng et al. [181], researched NER with text and image datasets [181]. He proposed AGBAN (Adversarial Gated Bilinear Attention Neural Network), which combines the extraction of image and text features. With this combination method, it can improve the overall performance of NER. As Zheng et al. do, most multimodal NER researchers use microblog social media datasets. The social media user added an image to the status he made on social media to corroborate the message he wanted to convey. The uploaded image and the status update are related to the status. This background then motivated the multimodal NER researchers to strengthen NER’s performance by extracting not only the text but on the accompanying image.

Multimodal NER research has been conducted by Asgari-Chenaghlu et al. [116], Tian et al. [170], Seunghwan et al. [388], and Liu et al.[146]. The datasets they use are all status posts on social media. NER's multimodal opportunities are still wide open, for example, in medicine and health, business and economics, and others.

5. Conclusion and Future Works

This review aims to map NER studies based on preprocessing data, datasets, application domains, feature extraction, approaches, methods, and NER evaluation. By looking at a graph of a significant increase from 2011 to 2020, the NER topic in the future will still be a trend in the NLP area. Named entity ambiguity is another challenge for research on NER. Research conducted by Zhou et al. [216] can be a starting point to continue the entity ambiguity topic because entity ambiguity can reduce NER performance. The combination of deep learning and other machine learning approaches is promising for future NER research.

NER for low-resource language is a future challenge in NER research. Most of NER's research has been in English, Chinese, and Arabic. There is little research and dataset for NER in Bahasa Indonesia, including NERgrit and NERP. It is a challenge, especially for researchers in Indonesia, because Indonesian speakers reach more than 270 million people. The preprocessing stage has a large portion of NER research. A good preprocessing stage will produce a quality dataset. Exploration of the preprocessing stage using various approaches could be an interesting opportunity in the future. For example, at the tokenization stage, you can use sub-word tokenization such as Byte-Pair Encoding (BPE) or WordPiece to deal with uncommon words or unusual entities.

The rapid development of big data opens up extensive opportunities to develop multimodal NER. Multimodal NER datasets can come from various types, including images, audio, and video. The development of multimodal NER applications can be state-of-the-art to provide an accurate data supply on IoT devices. IoT wearable devices are required to recognize various entities around them in text, audio, and three-dimensional visual objects. Identifying these IoT device entities will improve further if a multimodal NER module is embedded. This prediction is in line with what was conveyed by Middleton et al., that this trend is likely to continue and could foresee the progress made in approaches that exploit new sources of information such as personal mobile devices and data connection from the Internet of Things (IoT) [389]. The more context available to associate with a user's post, the better the chances of getting the correct location extraction and disambiguation.

NER's research related to the COVID-19 pandemic is wide open with the availability of public datasets from various related institutions, for example, WHO, scientific publication publishers, health authorities of each country, hospitals, and others. Entity recognition in the COVID-19 dataset can help perform sentiment analysis applications, chatbots, speech recognition, and machine translation dedicated to helping solve pandemics. The future chances are related to integrating NLP techniques in several ways, for example, morphological processing, syntactic analysis, or determination of POS-tags with the preprocessing stage to obtain richer features.

Acknowledgement

We would like to thank Dian Nuswantoro University, Semarang, Indonesia, and the UIN Saizu Purwokerto, Central Java, Indonesia, for providing insights and expertise that were very helpful in this research.

REFERENCES

1. C. C. Aggarwal and C. Zhai, An Introduction to Text Mining, in Mining Text Data, C. C. Aggarwal and C. Zhai, Eds., Boston, MA: Springer US, 2012, pp. 1–10. doi: 10.1007/978-1-4614-3223-4_1.
2. J. Jiang, Information Extraction from Text, in Mining Text Data, C. C. Aggarwal and C. Zhai, Eds., Boston, MA: Springer US, 2012, pp. 11–41. doi: 10.1007/978-1-4614-3223-4_2.
3. J. Piskorski and R. Yangarber, Information Extraction: Past, Present and Future, in Multi-source, Multilingual Information Extraction and Summarization, T. Poibeau, H. Saggion, J. Piskorski, and R. Yangarber, Eds., in Theory and Applications of Natural Language Processing. , Berlin, Heidelberg: Springer, 2013, pp. 23–49. doi: 10.1007/978-3-642-28569-1_2.

4. I. Augenstein, L. Derczynski, and K. Bontcheva, Generalisation in named entity recognition: A quantitative analysis, Comput. Speech Lang., vol. 44, pp. 61–83, Jul. 2017, doi: 10.1016/j.csl.2017.01.012.
5. A. Goyal, V. Gupta, and M. Kumar, Analysis of Different Supervised Techniques for Named Entity Recognition, in Advanced Informatics for Computing Research, A. K. Luhach, D. S. Jat, K. B. G. Hawari, X.-Z. Gao, and P. Lingras, Eds., in Communications in Computer and Information Science. Singapore: Springer, 2019, pp. 184–195. doi: 10.1007/978-981-15-0108-1_18.
6. Appendix C: Named Entity Task Definition (v2.1), in Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995, Columbia, 1995. Accessed: Sep. 19, 2021. Online. Available: https://aclanthology.org/M95-1024
7. M. Marrero, J. Urbano, S. Sánchez-Cuadrado, J. Morato, and J. M. Gómez-Berbís, Named Entity Recognition: Fallacies, challenges and opportunities, Comput. Stand. Interfaces, vol. 35, no. 5, pp. 482–489, Sep. 2013, doi: 10.1016/j.csi.2012.09.004.
8. M. S. Salleh, S. A. Asmai, H. Basiron, and S. Ahmad, A Malay named entity recognition using conditional random fields, in 2017 5th International Conference on Information and Communication Technology (ICoIC7), Melaka, Malaysia: IEEE, May 2017, pp. 1–6. doi: 10.1109/ICoICT.2017.8074647.
9. L. Derczynski et al., Analysis of named entity recognition and linking for tweets, Inf. Process. Manag., vol. 51, no. 2, pp. 32–49, Mar. 2015, doi: 10.1016/j.ipm.2014.10.006.
10. S. Eltyeb and N. Salim, Chemical named entities recognition: a review on approaches and applications, J. Cheminformatics, vol. 6, p. 17, 2014, doi: 10.1186/1758-2946-6-17.
11. D. Nadeau, P. D. Turney, and S. Matwin, Unsupervised Named-Entity Recognition: Generating Gazetteers and Resolving Ambiguity, in Advances in Artificial Intelligence, vol. 3060, A. Y. Tawfik and S. D. Goodwin, Eds., in Lecture Notes in Computer Science, vol. 3060. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 266–277. doi: 10.1007/11766247_23.
12. S. J. Pan, Z. Toh, and J. Su, Transfer joint embedding for cross-domain named entity recognition, ACM Trans. Inf. Syst., vol. 31, no. 2, pp. 1–27, May 2013, doi: 10.1145/2457465.2457467.
13. L. Li, W. Fan, and D. Huang, A Two-Phase Bio-NER System Based on Integrated Classifiers and Multiagent Strategy, IEEE/ACM Trans. Comput. Biol. Bioinform., vol. 10, no. 4, pp. 897–904, Jul. 2013, doi: 10.1109/TCBB.2013.106.
14. A. Akkasi and E. Varoglu, Improving Biochemical Named Entity Recognition Using PSO Classifier Selection and Bayesian Combination Methods, IEEE/ACM Trans. Comput. Biol. Bioinform., vol. 14, no. 6, pp. 1327–1338, Nov. 2017, doi: 10.1109/TCBB.2016.2570216.
15. Y. Wen, C. Fan, G. Chen, X. Chen, and M. Chen, A Survey on Named Entity Recognition, in Communications, Signal Processing, and Systems, vol. 571, Q. Liang, W. Wang, X. Liu, Z. Na, M. Jia, and B. Zhang, Eds., in Lecture Notes in Electrical Engineering, vol. 571. , Singapore: Springer Singapore, 2020, pp. 1803–1810. doi: 10.1007/978-981-13-9409-6_218.
16. Y. Chen, T. A. Lasko, Q. Mei, J. C. Denny, and H. Xu, A study of active learning methods for named entity recognition in clinical text, J. Biomed. Inform., vol. 58, pp. 11–18, Dec. 2015, doi: 10.1016/j.jbi.2015.09.010.
17. J. A. Lossio-Ventura, C. Jonquet, M. Roche, and M. Teisseire, Biomedical term extraction: overview and a new methodology, Inf. Retr. J., vol. 19, no. 1–2, pp. 59–99, Apr. 2016, doi: 10.1007/s10791-015-9262-2.
18. U. Kanimozhi and D. Manjula, A Systematic Review on Biomedical Named Entity Recognition, in Data Science Analytics and Applications, vol. 804, S. R and M. Sharma, Eds., in Communications in Computer and Information Science, vol. 804. , Singapore: Springer Singapore, 2018, pp. 19–37. doi: 10.1007/978-981-10-8603-8_3.
19. G. Popovski, B. K. Seljak, and T. Eftimov, A Survey of Named-Entity Recognition Methods for Food Information Extraction, IEEE Access, vol. 8, pp. 31586–31594, 2020, doi: 10.1109/ACCESS.2020.2973502.
20. D. Nadeau and S. Sekine, A survey of named entity recognition and classification, Lingvisticae Investig., vol. 30, no. 1, pp. 3–26, Aug. 2007, doi: 10.1075/li.30.1.03nad.
21. A. Goyal, V. Gupta, and M. Kumar, Recent Named Entity Recognition and Classification techniques: A systematic review, Comput. Sci. Rev., vol. 29, pp. 21–43, Aug. 2018, doi: 10.1016/j.cosrev.2018.06.001.
22. V. T. Chakaravarthy, V. Pandit, S. Roy, P. Awasthi, and M. K. Mohania, Decision trees for entity identification: Approximation algorithms and hardness results, ACM Trans. Algorithms, vol. 7, no. 2, pp. 1–22, Mar. 2011, doi: 10.1145/1921659.1921661.
23. X. Liao and Z. Zhao, Unsupervised Approaches for Textual Semantic Annotation, A Survey, ACM Comput. Surv., vol. 52, no. 4, pp. 1–45, Sep. 2019, doi: 10.1145/3324473.
24. A. Thomas and S. Sangeetha, Deep Learning Architectures for Named Entity Recognition: A Survey, in Advanced Computing and Intelligent Engineering, B. Pati, C. R. Panigrahi, R. Buyya, and K.-C. Li, Eds., in Advances in Intelligent Systems and Computing. Singapore: Springer, 2020, pp. 215–225. doi: 10.1007/978-981-15-1081-6_18.
25. J. Li, A. Sun, J. Han, and C. Li, A Survey on Deep Learning for Named Entity Recognition, IEEE Trans. Knowl. Data Eng., pp. 1–1, 2020, doi: 10.1109/TKDE.2020.2981314.
26. A. Dandashi, J. Al Jaam, and S. Foufou, Arabic named entity recognition—a survey and analysis: 9th KES International Conference on Intelligent Interactive Multimedia Systems and Services, IIMSS 2016, Intell. Interact. Multimed. Syst. Serv. 2016, pp. 83–96, 2016, doi: 10.1007/978-3-319-39345-2_8.
27. K. Shaalan, A Survey of Arabic Named Entity Recognition and Classification, Comput. Linguist., vol. 40, no. 2, pp. 469–510, Jun. 2014, doi: 10.1162/COLI_a_00178.
28. B. A. Kitchenham, Systematic review in software engineering: where we are and where we should be going, in Proceedings of the 2nd international workshop on Evidential assessment of software technologies - EAST '12, Lund, Sweden: ACM Press, 2012, p. 1. doi: 10.1145/2372233.2372235.
29. C. Okoli and K. Schabram, A Guide to Conducting a Systematic Literature Review of Information Systems Research, SSRN Electron. J., 2010, doi: 10.2139/ssrn.1954824.
30. A. Ekbal and S. Saha, Stacked ensemble coupled with feature selection for biomedical entity extraction, Knowl.-Based Syst., vol. 46, pp. 22–32, Jul. 2013, doi: 10.1016/j.knosys.2013.02.008.
31. A. Ekbal and S. Saha, Simultaneous feature and parameter selection using multiobjective optimization: application to named entity recognition, Int. J. Mach. Learn. Cybern., vol. 7, no. 4, pp. 597–611, Aug. 2016, doi: 10.1007/s13042-014-0268-7.

32. A. Ekbal and S. Saha, Combining feature selection and classifier ensemble using a multiobjective simulated annealing approach: application to named entity recognition, Soft Comput., vol. 17, no. 1, pp. 1–16, Jan. 2013, doi: 10.1007/s00500-012-0885-6.
33. A. Ekbal, S. Saha, and U. K. Sikdar, Biomedical named entity extraction: some issues of corpus compatibilities, SpringerPlus, vol. 2, no. 1, p. 601, Dec. 2013, doi: 10.1186/2193-1801-2-601.
34. A. Ekbal and S. Saha, Weighted Vote-Based Classifier Ensemble for Named Entity Recognition: A Genetic Algorithm-Based Approach, ACM Trans. Asian Lang. Inf. Process., vol. 10, no. 2, pp. 1–37, Jun. 2011, doi: 10.1145/1967293.1967296.
35. A. Ekbal and S. Saha, A multiobjective simulated annealing approach for classifier ensemble: Named entity recognition in Indian languages as case studies, Expert Syst. Appl., vol. 38, no. 12, pp. 14760–14772, Nov. 2011, doi: 10.1016/j.eswa.2011.05.004.
36. S. Saha and A. Ekbal, Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition, Data Knowl. Eng., vol. 85, pp. 15–39, May 2013, doi: 10.1016/j.datak.2012.06.003.
37. S. Yadav, A. Ekbal, and S. Saha, Feature selection for entity extraction from multiple biomedical corpora: A PSO-based approach, Soft Comput., vol. 22, no. 20, pp. 6881–6904, Oct. 2018, doi: 10.1007/s00500-017-2714-4.
38. S. Yadav, A. Ekbal, and S. Saha, Information theoretic-PSO-based feature selection: an application in biomedical entity extraction, Knowl. Inf. Syst., vol. 60, no. 3, pp. 1453–1478, Sep. 2019, doi: 10.1007/s10115-018-1265-z.
39. L. Sun, F. Ji, K. Zhang, and C. Wang, Multilayer ToI Detection Approach for Nested NER, IEEE Access, vol. 7, pp. 186600–186608, 2019, doi: 10.1109/ACCESS.2019.2961118.
40. L. Sun, K. Zhang, Y. Sun, F. Weng, and J. Zhang, ETIP: a lengthy nested NER problem for Chinese insurance policy analysis, Pattern Anal. Appl., vol. 23, no. 4, pp. 1755–1765, Nov. 2020, doi: 10.1007/s10044-020-00885-6.
41. L. Sun, Y. Sun, F. Ji, and C. Wang, Joint Learning of Token Context and Span Feature for Span-Based Nested NER, IEEEACM Trans. Audio Speech Lang. Process., vol. 28, pp. 2720–2730, 2020, doi: 10.1109/TASLP.2020.3024944.
42. A. Akkasi, Sentence-based undersampling for named entity recognition using genetic algorithm, Iran J. Comput. Sci., vol. 1, no. 3, pp. 165–174, Sep. 2018, doi: 10.1007/s42044-018-0014-5.
43. H. Cho and H. Lee, Biomedical named entity recognition using deep neural networks with contextual information, BMC Bioinformatics, vol. 20, no. 1, p. 735, Dec. 2019, doi: 10.1186/s12859-019-3321-4.
44. H. Cho, W. Choi, and H. Lee, A method for named entity normalization in biomedical articles: application to diseases and plants, BMC Bioinformatics, vol. 18, no. 1, p. 451, Dec. 2017, doi: 10.1186/s12859-017-1857-8.
45. Y. Gao, L. Gu, Y. Wang, Y. Wang, and F. Yang, Constructing a Chinese electronic medical record corpus for named entity recognition on resident admit notes, BMC Med. Inform. Decis. Mak., vol. 19, no. S2, p. 56, Apr. 2019, doi: 10.1186/s12911-019-0759-2.
46. Y. Gao, Y. Wang, P. Wang, and L. Gu, Medical Named Entity Extraction from Chinese Resident Admit Notes Using Character and Word Attention-Enhanced Neural Network, Int. J. Environ. Res. Public. Health, vol. 17, no. 5, p. 1614, Mar. 2020, doi: 10.3390/ijerph17051614.
47. Q. Qiu, Z. Xie, L. Wu, and L. Tao, GNER: A Generative Model for Geological Named Entity Recognition Without Labeled Data Using Deep Learning, Earth Space Sci., vol. 6, no. 6, pp. 931–946, Jun. 2019, doi: 10.1029/2019EA000610.
48. Q. Qiu, Z. Xie, L. Wu, L. Tao, and W. Li, BiLSTM-CRF for geological named entity recognition from the geoscience literature, Earth Sci. Inform., vol. 12, no. 4, pp. 565–579, Dec. 2019, doi: 10.1007/s12145-019-00390-3.
49. Z. Tang, L. Jiang, L. Yang, K. Li, and K. Li, CRFs based parallel biomedical named entity recognition algorithm employing MapReduce framework, Clust. Comput., vol. 18, no. 2, pp. 493–505, Jun. 2015, doi: 10.1007/s10586-015-0426-z.
50. Z. Tang, B. Wan, and L. Yang, Word-Character Graph Convolution Network for Chinese Named Entity Recognition, IEEEACM Trans. Audio Speech Lang. Process., vol. 28, pp. 1520–1532, 2020, doi: 10.1109/TASLP.2020.2994436.
51. N. Zhang, F. Li, G. Xu, W. Zhang, and H. Yu, Chinese NER Using Dynamic Meta-Embeddings, IEEE Access, vol. 7, pp. 64450–64459, 2019, doi: 10.1109/ACCESS.2019.2916816.
52. N. Zhang, G. Xu, Z. Zhang, and F. Li, MIFM: Multi-Granularity Information Fusion Model for Chinese Named Entity Recognition, IEEE Access, vol. 7, pp. 181648–181655, 2019, doi: 10.1109/ACCESS.2019.2958959.
53. G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, Neural Architectures for Named Entity Recognition, in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California: Association for Computational Linguistics, 2016, pp. 260–270. doi: 10.18653/v1/N16-1030.
54. M. Ballesteros, C. Dyer, and N. A. Smith, Improved Transition-Based Parsing by Modeling Characters instead of Words with LSTMs, ArXiv150800657 Cs, Aug. 2015, Accessed: Sep. 18, 2021. Online. Available: http://arxiv.org/abs/1508.00657
55. A. Ritter, S. Clark, Mausam, and O. Etzioni, Named Entity Recognition in Tweets: An Experimental Study, in Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, UK.: Association for Computational Linguistics, Jul. 2011, pp. 1524–1534. Accessed: Sep. 18, 2021. Online. Available: https://aclanthology.org/D11-1141
56. T. Eftimov, B. Koroušić Seljak, and P. Korošec, A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations, PLOS ONE, vol. 12, no. 6, p. e0179488, Jun. 2017, doi: 10.1371/journal.pone.0179488.
57. M. Dias, J. Boné, J. C. Ferreira, R. Ribeiro, and R. Maia, Named Entity Recognition for Sensitive Data Discovery in Portuguese, Appl. Sci., vol. 10, no. 7, p. 2303, Mar. 2020, doi: 10.3390/app10072303.
58. W. Hemati and A. Mehler, LSTMVoter: chemical named entity recognition using a conglomerate of sequence labeling tools, J. Cheminformatics, vol. 11, no. 1, p. 3, Dec. 2019, doi: 10.1186/s13321-018-0327-2.
59. J. Atkinson and V. Bull, A multi-strategy approach to biological named entity recognition, Expert Syst. Appl., vol. 39, no. 17, pp. 12968–12974, Dec. 2012, doi: 10.1016/j.eswa.2012.05.033.
60. B. Bhasuran, G. Murugesan, S. Abdulkadhar, and J. Natarajan, Stacked ensemble combined with fuzzy matching for biomedical named entity recognition of diseases, J. Biomed. Inform., vol. 64, pp. 1–9, Dec. 2016, doi: 10.1016/j.jbi.2016.09.009.
61. C. Cabot, S. Darmoni, and L. F. Soualmia, Cimind: A phonetic-based tool for multilingual named entity recognition in biomedical texts, J. Biomed. Inform., vol. 94, p. 103176, Jun. 2019, doi: 10.1016/j.jbi.2019.103176.
62. L. Gligic, A. Kormilitzin, P. Goldberg, and A. Nevado-Holgado, Named entity recognition in electronic health records using transfer learning bootstrapped Neural Networks, Neural Netw., vol. 121, pp. 132–139, Jan. 2020, doi: 10.1016/j.neunet.2019.08.032.

63. Y. Zhang et al., Psychiatric symptom recognition without labeled data using distributional representations of phrases and on-line knowledge, J. Biomed. Inform., vol. 75, pp. S129–S137, Nov. 2017, doi: 10.1016/j.jbi.2017.06.014.
64. M. Khalifa and K. Shaalan, Character convolutions for Arabic Named Entity Recognition with Long Short-Term Memory Networks, Comput. Speech Lang., vol. 58, pp. 335–346, Nov. 2019, doi: 10.1016/j.csl.2019.05.003.
65. M. Xiaofeng, W. Wei, and X. Aiping, Incorporating token-level dictionary feature into neural model for named entity recognition, Neurocomputing, vol. 375, pp. 43–50, Jan. 2020, doi: 10.1016/j.neucom.2019.09.005.
66. C. C. Aggarwal, Machine Learning for Text. Springer International Publishing, 2018. doi: 10.1007/978-3-319-73531-3.
67. M. Khabsa and C. L. Giles, Chemical entity extraction using CRF and an ensemble of extractors, J. Cheminformatics, vol. 7, no. S1, p. S12, Dec. 2015, doi: 10.1186/1758-2946-7-S1-S12.
68. A. Akkasi, E. Varoğlu, and N. Dimililer, ChemTok: A New Rule Based Tokenizer for Chemical Named Entity Recognition, BioMed Res. Int., vol. 2016, pp. 1–9, 2016, doi: 10.1155/2016/4248026.
69. X. Cai, S. Dong, and J. Hu, A deep learning model incorporating part of speech and self-matching attention for named entity recognition of Chinese electronic medical records, BMC Med. Inform. Decis. Mak., vol. 19, no. S2, p. 65, Apr. 2019, doi: 10.1186/s12911-019-0762-7.
70. H.-C. Cho, N. Okazaki, M. Miwa, and J. Tsujii, Named entity recognition with multiple segment representations, Inf. Process. Manag., vol. 49, no. 4, pp. 954–965, Jul. 2013, doi: 10.1016/j.ipm.2013.03.002.
71. V. Suárez-Paniagua, R. M. Rivera Zavala, I. Segura-Bedmar, and P. Martínez, A two-stage deep learning approach for extracting entities and relationships from medical texts, J. Biomed. Inform., vol. 99, p. 103285, Nov. 2019, doi: 10.1016/j.jbi.2019.103285.
72. B. Gaur, G. S. Saluja, H. B. Sivakumar, and S. Singh, Semi-supervised deep learning based named entity recognition model to parse education section of resumes, Neural Comput. Appl., vol. 33, no. 11, pp. 5705–5718, 2020, doi: 10.1007/s00521-020-05351-2.
73. C. Zhou, B. Li, and X. Sun, Improving software bug-specific named entity recognition with deep neural network, J. Syst. Softw., vol. 165, p. 110572, Jul. 2020, doi: 10.1016/j.jss.2020.110572.
74. R. Batista-Navarro, R. Rak, and S. Ananiadou, Optimising chemical named entity recognition with pre-processing analytics, knowledge-rich features and heuristics, J. Cheminformatics, vol. 7, no. S1, p. S6, Dec. 2015, doi: 10.1186/1758-2946-7-S1-S6.
75. S. Bordag, C. Hänig, and C. Beutenmüller, A Structuralist Approach for Personal Knowledge Exploration Systems on Mobile Devices, in Text Mining, C. Biemann and A. Mehler, Eds., in Theory and Applications of Natural Language Processing. , Cham: Springer International Publishing, 2014, pp. 115–136. doi: 10.1007/978-3-319-12655-5_6.
76. H. Khalil, T. Osman, and M. Miltan, Extracting Arabic Composite Names Using Genitive Principles of Arabic Grammar, ACM Trans. Asian Low-Resour. Lang. Inf. Process., vol. 19, no. 4, pp. 1–16, Jul. 2020, doi: 10.1145/3382187.
77. Y.-Y. Hsu and H.-Y. Kao, Curatable Named-Entity Recognition Using Semantic Relations, IEEE/ACM Trans. Comput. Biol. Bioinform., vol. 12, no. 4, pp. 785–792, Jul. 2015, doi: 10.1109/TCBB.2014.2366770.
78. L. Li, J. Zhao, L. Hou, Y. Zhai, J. Shi, and F. Cui, An attention-based deep learning model for clinical named entity recognition of Chinese electronic medical records, BMC Med. Inform. Decis. Mak., vol. 19, no. S5, p. 235, Dec. 2019, doi: 10.1186/s12911-019-0933-6.
79. G. Murugesan, S. Abdulkadhar, B. Bhasuran, and J. Natarajan, BCC-NER: bidirectional, contextual clues named entity tagger for gene/protein mention recognition, EURASIP J. Bioinforma. Syst. Biol., vol. 2017, no. 1, p. 7, Dec. 2017, doi: 10.1186/s13637-017-0060-6.
80. S. Song, N. Zhang, and H. Huang, Named entity recognition based on conditional random fields, Clust. Comput., vol. 22, no. S3, pp. 5195–5206, May 2019, doi: 10.1007/s10586-017-1146-3.
81. R. Sivashankari and B. Valarmathi, NLP-MTFLR: Document-Level Prioritization and Identification of Dominant Multi-word Named Products in Customer Reviews, Arab. J. Sci. Eng., vol. 43, no. 2, pp. 843–855, Feb. 2018, doi: 10.1007/s13369-017-2773-y.
82. Q. Zhao, D. Wang, J. Li, and F. Akhtar, Exploiting the concept level feature for enhanced name entity recognition in Chinese EMRs, J. Supercomput., vol. 76, no. 8, pp. 6399–6420, 2019, doi: 10.1007/s11227-019-02917-3.
83. W. Huang, D. Hu, Z. Deng, and J. Nie, Named entity recognition for Chinese judgment documents based on BiLSTM and CRF, EURASIP J. Image Video Process., vol. 2020, no. 1, p. 52, Dec. 2020, doi: 10.1186/s13640-020-00539-x.
84. J. Wang, C. Lin, M. Li, and C. Zaniolo, Boosting approximate dictionary-based entity extraction with synonyms, Inf. Sci., vol. 530, pp. 1–21, Aug. 2020, doi: 10.1016/j.ins.2020.04.025.
85. L. Li and Y. Jiang, Integrating Language Model and Reading Control Gate in BLSTM-CRF for Biomedical Named Entity Recognition, IEEE/ACM Trans. Comput. Biol. Bioinform., vol. 17, no. 3, pp. 841–846, 2018, doi: 10.1109/TCBB.2018.2868346.
86. Y. Wang, Y. Li, Z. Zhu, H. Tong, and Y. Huang, Adversarial Learning for Multi-task Sequence Labeling with Attention Mechanism, IEEEACM Trans. Audio Speech Lang. Process., vol. 28, pp. 1–1, 2020, doi: 10.1109/TASLP.2020.3013114.
87. X. Zhong, E. Cambria, and A. Hussain, Extracting Time Expressions and Named Entities with Constituent-Based Tagging Schemes, Cogn. Comput., vol. 12, no. 4, pp. 844–862, Jul. 2020, doi: 10.1007/s12559-020-09714-8.
88. X. Liu, N. Yang, Y. Jiang, L. Gu, and X. Shi, A parallel computing-based Deep Attention model for named entity recognition, J. Supercomput., vol. 76, no. 2, pp. 814–830, Feb. 2020, doi: 10.1007/s11227-019-02985-5.
89. H.-S. Chang, S. Vembu, S. Mohan, R. Uppaal, and A. McCallum, Using error decay prediction to overcome practical issues of deep active learning for named entity recognition, Mach. Learn., vol. 109, no. 9–10, pp. 1749–1778, Sep. 2020, doi: 10.1007/s10994-020-05897-1.
90. E. F. T. K. Sang and F. De Meulder, Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition, arXiv:cs/0306050, Jun. 2003, Accessed: Sep. 18, 2021. Online. Available: http://arxiv.org/abs/cs/0306050
91. J. Wang, W. Xu, X. Fu, G. Xu, and Y. Wu, ASTRAL: Adversarial Trained LSTM-CNN for Named Entity Recognition, Knowl.-Based Syst., vol. 197, p. 105842, Jun. 2020, doi: 10.1016/j.knosys.2020.105842.
92. J. P. C. Chiu and E. Nichols, Named Entity Recognition with Bidirectional LSTM-CNNs, Trans. Assoc. Comput. Linguist., vol. 4, pp. 357–370, Dec. 2016, doi: 10.1162/tacl_a_00104.
93. G. Aguilar, F. AlGhamdi, V. Soto, M. Diab, J. Hirschberg, and T. Solorio, Named Entity Recognition on Code-Switched Data: Overview of the CALCS 2018 Shared Task, in Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching, Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 138–147. doi: 10.18653/v1/W18-3219.

94. M. Peters et al., Deep Contextualized Word Representations, in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 2227–2237. doi: 10.18653/v1/N18-1202.
95. K. Clark, M.-T. Luong, C. D. Manning, and Q. V. Le, Semi-Supervised Sequence Modeling with Cross-View Training, ArXiv Prepr. ArXiv180908370, Sep. 2018, Accessed: Sep. 19, 2021. Online. Available: http://arxiv.org/abs/1809.08370
96. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, ArXiv Prepr. ArXiv181004805, May 2019, Accessed: Sep. 19, 2021. Online. Available: http://arxiv.org/abs/1810.04805
97. A. Akbik, D. Blythe, and R. Vollgraf, Contextual String Embeddings for Sequence Labeling, in Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1638–1649. Accessed: Sep. 19, 2021. Online. Available: https://aclanthology.org/C18-1139
98. S. Bratus, A. Rumshisky, A. Khrabrov, R. Magar, and P. Thompson, Domain-specific entity extraction from noisy, unstructured data using ontology-guided search, Int. J. Doc. Anal. Recognit. IJDAR, vol. 14, no. 2, pp. 201–211, Jun. 2011, doi: 10.1007/s10032-011-0149-5.
99. X. Guo et al., Chinese agricultural diseases and pests named entity recognition with multi-scale local context features and self-attention mechanism, Comput. Electron. Agric., vol. 179, p. 105830, Dec. 2020, doi: 10.1016/j.compag.2020.105830.
100. C. Liu, C. Fan, Z. Wang, and Y. Sun, An Instance Transfer-Based Approach Using Enhanced Recurrent Neural Network for Domain Named Entity Recognition, IEEE Access, vol. 8, pp. 45263–45270, 2020, doi: 10.1109/ACCESS.2020.2974022.
101. J. Kim, Y. Ko, and J. Seo, Construction of Machine-Labeled Data for Improving Named Entity Recognition by Transfer Learning, IEEE Access, vol. 8, pp. 59684–59693, 2020, doi: 10.1109/ACCESS.2020.2981361.
102. J. Kim, Y. Ko, and J. Seo, A Bootstrapping Approach With CRF and Deep Learning Models for Improving the Biomedical Named Entity Recognition in Multi-Domains, IEEE Access, vol. 7, pp. 70308–70318, 2019, doi: 10.1109/ACCESS.2019.2914168.
103. X. Seti, A. Wumaier, T. Yibulayin, D. Paerhati, L. Wang, and A. Saimaiti, Named-Entity Recognition in Sports Field Based on a Character-Level Graph Convolutional Network, Information, vol. 11, no. 1, p. 30, Jan. 2020, doi: 10.3390/info11010030.
104. F. Liao, L. Ma, J. Pei, and L. Tan, Combined Self-Attention Mechanism for Chinese Named Entity Recognition in Military, Future Internet, vol. 11, no. 8, p. 180, Aug. 2019, doi: 10.3390/fi11080180.
105. S. Johnson, S. Shen, and Y. Liu, CWPC_BiAtt: Character–Word–Position Combined BiLSTM-Attention for Chinese Named Entity Recognition, Information, vol. 11, no. 1, p. 45, Jan. 2020, doi: 10.3390/info11010045.
106. Y. Jin, J. Xie, W. Guo, C. Luo, D. Wu, and R. Wang, LSTM-CRF Neural Network With Gated Self Attention for Chinese NER, IEEE Access, vol. 7, pp. 136694–136703, 2019, doi: 10.1109/ACCESS.2019.2942433.
107. A. Ekbal and S. Saha, Multiobjective optimization for classifier ensemble and feature selection: an application to named entity recognition, Int. J. Doc. Anal. Recognit. IJDAR, vol. 15, no. 2, pp. 143–166, Jun. 2012, doi: 10.1007/s10032-011-0155-7.
108. J. Xu, H. He, X. Sun, X. Ren, and S. Li, Cross-Domain and Semisupervised Named Entity Recognition in Chinese Social Media: A Unified Model, IEEEACM Trans. Audio Speech Lang. Process., vol. 26, no. 11, pp. 2142–2152, Nov. 2018, doi: 10.1109/TASLP.2018.2856625.
109. Y. Song, S. Jeong, and H. Kim, Semi-automatic construction of a named entity dictionary for entity-based sentiment analysis in social media, Multimed. Tools Appl., vol. 76, no. 9, pp. 11319–11329, May 2017, doi: 10.1007/s11042-016-3445-8.
110. T. Iwakura, H. Takamura, and M. Okumura, A Named Entity Recognition Method Based on Decomposition and Concatenation of Word Chunks, ACM Trans. Asian Lang. Inf. Process., vol. 12, no. 3, pp. 1–18, Aug. 2013, doi: 10.1145/2499955.2499958.
111. D. Deng, G. Li, J. Feng, Y. Duan, and Z. Gong, A unified framework for approximate dictionary-based entity extraction, VLDB J., vol. 24, no. 1, pp. 143–167, Feb. 2015, doi: 10.1007/s00778-014-0367-9.
112. R. Murthy, M. M. Khapra, and P. Bhattacharyya, Improving NER Tagging Performance in Low-Resource Languages via Multilingual Learning, ACM Trans. Asian Low-Resour. Lang. Inf. Process., vol. 18, no. 2, pp. 1–20, Feb. 2019, doi: 10.1145/3238797.
113. L. Li, P. Wang, D. Huang, and L. Zhao, Mining English-Chinese Named Entity Pairs from Comparable Corpora, ACM Trans. Asian Lang. Inf. Process., vol. 10, no. 4, pp. 1–19, Dec. 2011, doi: 10.1145/2025384.2025387.
114. L. Zhao, X. Qiu, Q. Zhang, and X. Huang, Sequence Labeling With Deep Gated Dual Path CNN, IEEEACM Trans. Audio Speech Lang. Process., vol. 27, no. 12, pp. 2326–2335, Dec. 2019, doi: 10.1109/TASLP.2019.2944563.
115. H. Kim and H. Kim, Integrated Model for Morphological Analysis and Named Entity Recognition Based on Label Attention Networks in Korean, Appl. Sci., vol. 10, no. 11, p. 3740, May 2020, doi: 10.3390/app10113740.
116. W. Khan, A. Daud, F. Alotaibi, N. Aljohani, and S. Arafat, Deep recurrent neural networks with word embeddings for Urdu named entity recognition, ETRI J., vol. 42, no. 1, pp. 90–100, Feb. 2020, doi: 10.4218/etrij.2018-0553.
117. M. Ali, G. Tan, and A. Hussain, Bidirectional Recurrent Neural Network Approach for Arabic Named Entity Recognition, Future Internet, vol. 10, no. 12, p. 123, Dec. 2018, doi: 10.3390/fi10120123.
118. A. Saimaiti, L. Wang, and T. Yibulayin, Learning Subword Embedding to Improve Uyghur Named-Entity Recognition, Information, vol. 10, no. 4, p. 139, Apr. 2019, doi: 10.3390/info10040139.
119. A. P. Ajees, K. Manju, and S. Mary Idicula, An Improved Word Representation for Deep Learning Based NER in Indian Languages, Information, vol. 10, no. 6, p. 186, May 2019, doi: 10.3390/info10060186.
120. B. Peng, J. Wu, H. Yuan, Q. Guo, and D. Tao, ANEEC: A Quasi-Automatic System for Massive Named Entity Extraction and Categorization, Comput. J., vol. 56, no. 11, pp. 1328–1346, Nov. 2013, doi: 10.1093/comjnl/bxs114.
121. H. Zhu, C. He, Y. Fang, and W. Xiao, Fine Grained Named Entity Recognition via Seq2seq Framework, IEEE Access, vol. 8, pp. 53953–53961, 2020, doi: 10.1109/ACCESS.2020.2980431.
122. M. Al-Smadi, S. Al-Zboon, Y. Jararweh, and P. Juola, Transfer Learning for Arabic Named Entity Recognition With Deep Neural Networks, IEEE Access, vol. 8, pp. 37736–37745, 2020, doi: 10.1109/ACCESS.2020.2973319.
123. I. Ashrafi et al., Banner: A Cost-Sensitive Contextualized Model for Bangla Named Entity Recognition, IEEE Access, vol. 8, pp. 58206–58226, 2020, doi: 10.1109/ACCESS.2020.2982427.
124. D. Li, X. Zhang, and X. Wu, Integrated Chinese Segmentation, Parsing and Named Entity Recognition, Chin. J. Electron., vol. 27, no. 4, pp. 756–760, Jul. 2018, doi: 10.1049/cje.2018.05.014.

125. M. N. A. Ali, G. Tan, and A. Hussain, Boosting Arabic Named-Entity Recognition With Multi-Attention Layer, IEEE Access, vol. 7, pp. 46575–46582, 2019, doi: 10.1109/ACCESS.2019.2909641.
126. R. Agerri and G. Rigau, Robust multilingual Named Entity Recognition with shallow semi-supervised features, Artif. Intell., vol. 238, pp. 63–82, Sep. 2016, doi: 10.1016/j.artint.2016.05.003.
127. G. Bekoulis, J. Deleu, T. Demeester, and C. Develder, Joint entity recognition and relation extraction as a multi-head selection problem, Expert Syst. Appl., vol. 114, pp. 34–45, Dec. 2018, doi: 10.1016/j.eswa.2018.07.032.
128. E. Fersini, E. Messina, G. Felici, and D. Roth, Soft-constrained inference for Named Entity Recognition, Inf. Process. Manag., vol. 50, no. 5, pp. 807–819, Sep. 2014, doi: 10.1016/j.ipm.2014.04.005.
129. O. Ghiasvand and R. J. Kate, Learning for clinical named entity recognition without manual annotations, Inform. Med. Unlocked, vol. 13, pp. 122–127, 2018, doi: 10.1016/j.imu.2018.10.011.
130. S.-H. Na, H. Kim, J. Min, and K. Kim, Improving LSTM CRFs using character-based compositions for Korean named entity recognition, Comput. Speech Lang., vol. 54, pp. 106–121, Mar. 2019, doi: 10.1016/j.csl.2018.09.005.
131. M. Konkol, T. Brychcín, and M. Konopík, Latent semantics in Named Entity Recognition, Expert Syst. Appl., vol. 42, no. 7, pp. 3470–3479, May 2015, doi: 10.1016/j.eswa.2014.12.015.
132. M. N. A. Ali and G. Tan, Bidirectional Encoder–Decoder Model for Arabic Named Entity Recognition, Arab. J. Sci. Eng., vol. 44, no. 11, pp. 9693–9701, Nov. 2019, doi: 10.1007/s13369-019-04068-2.
133. H. Al-Jumaily, P. Martínez, J. L. Martínez-Fernández, and E. Van der Goot, A real time Named Entity Recognition system for Arabic text mining, Lang. Resour. Eval., vol. 46, no. 4, pp. 543–563, Dec. 2012, doi: 10.1007/s10579-011-9146-z.
134. K.-Y. Cui, P.-J. Ren, Z.-M. Chen, T. Lian, and J. Ma, Relation Enhanced Neural Model for Type Classification of Entity Mentions with a Fine-Grained Taxonomy, J. Comput. Sci. Technol., vol. 32, no. 4, pp. 814–827, Jul. 2017, doi: 10.1007/s11390-017-1762-7.
135. C. de Pablo-Sánchez, I. Segura-Bedmar, P. Martínez, and A. Iglesias-Maqueda, Lightly supervised acquisition of named entities and linguistic patterns for multilingual text mining, Knowl. Inf. Syst., vol. 35, no. 1, pp. 87–109, Apr. 2013, doi: 10.1007/s10115-012-0502-0.
136. B. Desmet and V. Hoste, Fine-grained Dutch named entity recognition, Lang. Resour. Eval., vol. 48, no. 2, pp. 307–343, Jun. 2014, doi: 10.1007/s10579-013-9255-y.
137. Z. Li, D. Feng, D. Li, and X. Lu, Learning to select pseudo labels: a semi-supervised method for named entity recognition, Front. Inf. Technol. Electron. Eng., vol. 21, no. 6, pp. 903–916, Jun. 2020, doi: 10.1631/FITEE.1800743.
138. R. Lima, B. Espinasse, and F. Freitas, OntoILPER: an ontology- and inductive logic programming-based system to extract entities and relations from text, Knowl. Inf. Syst., vol. 56, no. 1, pp. 223–255, Jul. 2018, doi: 10.1007/s10115-017-1108-3.
139. H. Moradi, F. Ahmadi, and M.-R. Feizi-Derakhshi, A Hybrid Approach for Persian Named Entity Recognition, Iran. J. Sci. Technol. Trans. Sci., vol. 41, no. 1, pp. 215–222, Mar. 2017, doi: 10.1007/s40995-017-0209-x.
140. D. Peng, Y. Wang, C. Liu, and Z. Chen, TL-NER: A Transfer Learning Model for Chinese Named Entity Recognition, Inf. Syst. Front., vol. 22, no. 6, pp. 1291–1304, Dec. 2020, doi: 10.1007/s10796-019-09932-y.
141. R. Sharma, S. Morwal, B. Agarwal, R. Chandra, and M. S. Khan, A deep neural network-based model for named entity recognition for Hindi language, Neural Comput. Appl., vol. 32, no. 20, pp. 16191–16203, Oct. 2020, doi: 10.1007/s00521-020-04881-z.
142. M. Song, H. Yu, and W.-S. Han, Developing a hybrid dictionary-based bio-entity recognition technique, BMC Med. Inform. Decis. Mak., vol. 15, no. S1, p. S9, Dec. 2015, doi: 10.1186/1472-6947-15-S1-S9.
143. Z. Wang, X. Cui, L. Gao, Q. Yin, L. Ke, and S. Zhang, A hybrid model of sentimental entity recognition on mobile social media, EURASIP J. Wirel. Commun. Netw., vol. 2016, no. 1, p. 253, Dec. 2016, doi: 10.1186/s13638-016-0745-7.
144. C.-L. Chou, C.-H. Chang, Y.-H. Lin, and K.-C. Chien, On the Construction of Web NER Model Training Tool based on Distant Supervision, ACM Trans. Asian Low-Resour. Lang. Inf. Process., vol. 19, no. 6, pp. 1–28, Nov. 2020, doi: 10.1145/3422817.
145. M. S. H. Ameur, R. Belkebir, and A. Guessoum, Robust Arabic Text Categorization by Combining Convolutional and Recurrent Neural Networks, ACM Trans. Asian Low-Resour. Lang. Inf. Process., vol. 19, no. 5, pp. 1–16, Aug. 2020, doi: 10.1145/3390092.
146. N. Alshammari and S. Alanazi, The impact of using different annotation schemes on named entity recognition, Egypt. Inform. J., vol. 22, no. 3, pp. 295–302, Sep. 2021, doi: 10.1016/j.eij.2020.10.004.
147. J. T. Zhou, H. Zhang, D. Jin, X. Peng, Y. Xiao, and Z. Cao, RoSeq: Robust Sequence Labeling, IEEE Trans. Neural Netw. Learn. Syst., pp. 1–11, 2019, doi: 10.1109/TNNLS.2019.2911236.
148. C. Gong, Z. Li, Q. Xia, W. Chen, and M. Zhang, Hierarchical LSTM with char-subword-word tree-structure representation for Chinese named entity recognition, Sci. China Inf. Sci., vol. 63, no. 10, p. 202102, Oct. 2020, doi: 10.1007/s11432-020-2982-y.
149. M. Song, W. C. Kim, D. Lee, G. E. Heo, and K. Y. Kang, PKDE4J: Entity and relation extraction for public knowledge discovery, J. Biomed. Inform., vol. 57, pp. 320–332, Oct. 2015, doi: 10.1016/j.jbi.2015.08.008.
150. G. Crichton, S. Pyysalo, B. Chiu, and A. Korhonen, A neural network multi-task learning approach to biomedical named entity recognition, BMC Bioinformatics, vol. 18, no. 1, p. 368, Dec. 2017, doi: 10.1186/s12859-017-1776-8.
151. X. Dong et al., Deep learning for named entity recognition on Chinese electronic medical records: Combining deep transfer learning with multitask bi-directional LSTM RNN, PLOS ONE, vol. 14, no. 5, p. e0216046, May 2019, doi: 10.1371/journal.pone.0216046.
152. X. Li, C. Fu, R. Zhong, D. Zhong, T. He, and X. Jiang, A hybrid deep learning framework for bacterial named entity recognition with domain features, BMC Bioinformatics, vol. 20, no. S16, p. 583, Dec. 2019, doi: 10.1186/s12859-019-3071-3.
153. M. Gao, Q. Xiao, S. Wu, and K. Deng, An Improved Method for Named Entity Recognition and Its Application to CEMR, Future Internet, vol. 11, no. 9, p. 185, Aug. 2019, doi: 10.3390/fi11090185.
154. E. Batbaatar and K. H. Ryu, Ontology-Based Healthcare Named Entity Recognition from Twitter Messages Using a Recurrent Neural Network Approach, Int. J. Environ. Res. Public. Health, vol. 16, no. 19, p. 3628, Sep. 2019, doi: 10.3390/ijerph16193628.
155. X. Chen, C. Ouyang, Y. Liu, and Y. Bu, Improving the Named Entity Recognition of Chinese Electronic Medical Records by Combining Domain Dictionary and Rules, Int. J. Environ. Res. Public. Health, vol. 17, no. 8, p. 2687, Apr. 2020, doi: 10.3390/ijerph17082687.
156. Y. Wang, S. Ananiadou, and J. Tsujii, Improving clinical named entity recognition in Chinese using the graphical and phonetic feature, BMC Med. Inform. Decis. Mak., vol. 19, no. S7, p. 273, Dec. 2019, doi: 10.1186/s12911-019-0980-z.

157. K. Xu, Z. Zhou, T. Gong, T. Hao, and W. Liu, SBLC: a hybrid model for disease named entity recognition based on semantic bidirectional LSTMs and conditional random fields, BMC Med. Inform. Decis. Mak., vol. 18, no. S5, p. 114, Dec. 2018, doi: 10.1186/s12911-018-0690-y.
158. C. Lyu, B. Chen, Y. Ren, and D. Ji, Long short-term memory RNN for biomedical named entity recognition, BMC Bioinformatics, vol. 18, no. 1, p. 462, Dec. 2017, doi: 10.1186/s12859-017-1868-5.
159. B. Ji et al., A hybrid approach for named entity recognition in Chinese electronic medical record, BMC Med. Inform. Decis. Mak., vol. 19, no. S2, p. 64, Apr. 2019, doi: 10.1186/s12911-019-0767-2.
160. S. Chowdhury et al., A multitask bi-directional RNN model for named entity recognition on Chinese electronic medical records, BMC Bioinformatics, vol. 19, no. S17, p. 499, Dec. 2018, doi: 10.1186/s12859-018-2467-9.
161. R. Patra and S. K. Saha, A Kernel-Based Approach for Biomedical Named Entity Recognition, Sci. World J., vol. 2013, pp. 1–7, 2013, doi: 10.1155/2013/950796.
162. W. Yoon, C. H. So, J. Lee, and J. Kang, CollaboNet: collaboration of deep neural networks for biomedical named entity recognition, BMC Bioinformatics, vol. 20, no. S10, p. 249, May 2019, doi: 10.1186/s12859-019-2813-6.
163. Z. Zhao et al., Disease named entity recognition from biomedical literature using a novel convolutional neural network, BMC Med. Genomics, vol. 10, no. S5, p. 73, Dec. 2017, doi: 10.1186/s12920-017-0316-8.
164. Y. Lou, T. Qian, F. Li, and D. Ji, A Graph Attention Model for Dictionary-Guided Named Entity Recognition, IEEE Access, vol. 8, pp. 71584–71592, 2020, doi: 10.1109/ACCESS.2020.2987399.
165. J. Qiu, Y. Zhou, Q. Wang, T. Ruan, and J. Gao, Chinese Clinical Named Entity Recognition Using Residual Dilated Convolutional Neural Network With Conditional Random Field, IEEE Trans. NanoBioscience, vol. 18, no. 3, pp. 306–315, Jul. 2019, doi: 10.1109/TNB.2019.2908678.
166. K. Li et al., Hadoop Recognition of Biomedical Named Entity Using Conditional Random Fields, IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 11, pp. 3040–3051, Nov. 2015, doi: 10.1109/TPDS.2014.2368568.
167. G. Wu, G. Tang, Z. Wang, Z. Zhang, and Z. Wang, An Attention-Based BiLSTM-CRF Model for Chinese Clinic Named Entity Recognition, IEEE Access, vol. 7, pp. 113942–113949, 2019, doi: 10.1109/ACCESS.2019.2935223.
168. D. Kim et al., A Neural Named Entity Recognition and Multi-Type Normalization Tool for Biomedical Text Mining, IEEE Access, vol. 7, pp. 73729–73740, 2019, doi: 10.1109/ACCESS.2019.2920708.
169. H. Wei et al., Named Entity Recognition From Biomedical Texts Using a Fusion Attention-Based BiLSTM-CRF, IEEE Access, vol. 7, pp. 73627–73636, 2019, doi: 10.1109/ACCESS.2019.2920734.
170. A. Casillas, N. Ezeiza, I. Goenaga, A. Pérez, and X. Soto, Measuring the effect of different types of unsupervised word representations on Medical Named Entity Recognition, Int. J. Med. Inf., vol. 129, pp. 100–106, Sep. 2019, doi: 10.1016/j.ijmedinf.2019.05.022.
171. J. Chae et al., Identifying non-elliptical entity mentions in a coordinated NP with ellipses, J. Biomed. Inform., vol. 47, pp. 139–152, Feb. 2014, doi: 10.1016/j.jbi.2013.10.002.
172. Y. Chen et al., Named entity recognition from Chinese adverse drug event reports with lexical feature based BiLSTM-CRF and tri-training, J. Biomed. Inform., vol. 96, p. 103252, Aug. 2019, doi: 10.1016/j.jbi.2019.103252.
173. M. Cho, J. Ha, C. Park, and S. Park, Combinatorial feature embedding based on CNN and LSTM for biomedical named entity recognition, J. Biomed. Inform., vol. 103, p. 103381, Mar. 2020, doi: 10.1016/j.jbi.2020.103381.
174. M. Gridach, Character-level neural network for biomedical named entity recognition, J. Biomed. Inform., vol. 70, pp. 85–91, Jun. 2017, doi: 10.1016/j.jbi.2017.05.002.
175. J. Guo, C. Blake, and Y. Guan, Evaluating automated entity extraction with respect to drug and non-drug treatment strategies, J. Biomed. Inform., vol. 94, p. 103177, Jun. 2019, doi: 10.1016/j.jbi.2019.103177.
176. I. J. Unanue, E. Z. Borzeshi, and M. Piccardi, Recurrent neural networks with specialized word embeddings for health-domain named-entity recognition, J. Biomed. Inform., vol. 76, pp. 102–109, Dec. 2017, doi: 10.1016/j.jbi.2017.11.007.
177. S. Santiso, A. Pérez, A. Casillas, and M. Oronoz, Neural negated entity recognition in Spanish electronic health records, J. Biomed. Inform., vol. 105, p. 103419, May 2020, doi: 10.1016/j.jbi.2020.103419.
178. J. Urbain, Mining heart disease risk factors in clinical text with named entity recognition and distributional semantic models, J. Biomed. Inform., vol. 58, pp. S143–S149, Dec. 2015, doi: 10.1016/j.jbi.2015.08.009.
179. Q. Wang, Y. Zhou, T. Ruan, D. Gao, Y. Xia, and P. He, Incorporating dictionaries into deep neural networks for the Chinese clinical named entity recognition, J. Biomed. Inform., vol. 92, p. 103133, Apr. 2019, doi: 10.1016/j.jbi.2019.103133.
180. M. Yin, C. Mou, K. Xiong, and J. Ren, Chinese clinical named entity recognition with radical-level feature and self-attention mechanism, J. Biomed. Inform., vol. 98, p. 103289, Oct. 2019, doi: 10.1016/j.jbi.2019.103289.
181. S. Zhang and N. Elhadad, Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts, J. Biomed. Inform., vol. 46, no. 6, pp. 1088–1098, Dec. 2013, doi: 10.1016/j.jbi.2013.08.004.
182. S. Zhao, Z. Cai, H. Chen, Y. Wang, F. Liu, and A. Liu, Adversarial training based lattice LSTM for Chinese clinical named entity recognition, J. Biomed. Inform., vol. 99, p. 103290, Nov. 2019, doi: 10.1016/j.jbi.2019.103290.
183. A. Pérez, R. Weegar, A. Casillas, K. Gojenola, M. Oronoz, and H. Dalianis, Semi-supervised medical entity recognition: A study on Spanish and Swedish clinical corpora, J. Biomed. Inform., vol. 71, pp. 16–30, Jul. 2017, doi: 10.1016/j.jbi.2017.05.009.
184. A. Akkasi, E. Varoğlu, and N. Dimililer, Balanced undersampling: a novel sentence-based undersampling method to improve recognition of named entities in chemical and biomedical text, Appl. Intell., vol. 48, no. 8, pp. 1965–1978, Aug. 2018, doi: 10.1007/s10489-017-0920-5.
185. M. Basaldella, L. Furrer, C. Tasso, and F. Rinaldi, Entity recognition in the biomedical domain using a hybrid approach, J. Biomed. Semant., vol. 8, no. 1, p. 51, Dec. 2017, doi: 10.1186/s13326-017-0157-6.
186. Y. Qin and Y. Zeng, Research of Clinical Named Entity Recognition Based on Bi-LSTM-CRF, J. Shanghai Jiaotong Univ. Sci., vol. 23, no. 3, pp. 392–397, Jun. 2018, doi: 10.1007/s12204-018-1954-5.
187. B. Tang, X. Wang, J. Yan, and Q. Chen, Entity recognition in Chinese clinical text using attention-based CNN-LSTM-CRF, BMC Med. Inform. Decis. Mak., vol. 19, no. S3, p. 74, Apr. 2019, doi: 10.1186/s12911-019-0787-y.
188. R. Weegar, A. Pérez, A. Casillas, and M. Oronoz, Recent advances in Swedish and Spanish medical entity recognition in clinical texts using deep neural approaches, BMC Med. Inform. Decis. Mak., vol. 19, no. S7, p. 274, Dec. 2019, doi: 10.1186/s12911-019-0981-y.

189. L. Yang and Y. Zhou, Exploring feature sets for two-phase biomedical named entity recognition using semi-CRFs, Knowl. Inf. Syst., vol. 40, no. 2, pp. 439–453, Aug. 2014, doi: 10.1007/s10115-013-0637-7.
190. L. Yao, C. Sun, Y. Wu, X. Wang, and X. Wang, Biomedical named entity recognition using generalized expectation criteria, Int. J. Mach. Learn. Cybern., vol. 2, no. 4, pp. 235–243, Dec. 2011, doi: 10.1007/s13042-011-0022-3.
191. S. M. Yimam, C. Biemann, L. Majnaric, Š. Šabanović, and A. Holzinger, An adaptive annotation approach for biomedical entity and relation recognition, Brain Inform., vol. 3, no. 3, pp. 157–168, Sep. 2016, doi: 10.1007/s40708-016-0036-4.
192. M. Liu, Y. Zhang, W. Li, and D. Ji, Joint Model of Entity Recognition and Relation Extraction with Self-attention Mechanism, ACM Trans. Asian Low-Resour. Lang. Inf. Process., vol. 19, no. 4, pp. 1–19, Jul. 2020, doi: 10.1145/3387634.
193. R. Catelli, F. Gargiulo, V. Casola, G. De Pietro, H. Fujita, and M. Esposito, Crosslingual named entity recognition for clinical de-identification applied to a COVID-19 Italian data set, Appl. Soft Comput., vol. 97, p. 106779, Dec. 2020, doi: 10.1016/j.asoc.2020.106779.
194. Y. Song, S. Tian, and L. Yu, A Method for Identifying Local Drug Names in Xinjiang Based on BERT-BiLSTM-CRF, Autom. Control Comput. Sci., vol. 54, no. 3, pp. 179–190, May 2020, doi: 10.3103/S0146411620030098.
195. Y. Tian, W. Shen, Y. Song, F. Xia, M. He, and K. Li, Improving biomedical named entity recognition with syntactic information, BMC Bioinformatics, vol. 21, no. 1, p. 539, Dec. 2020, doi: 10.1186/s12859-020-03834-6.
196. Y.-M. Kim and T.-H. Lee, Korean clinical entity recognition from diagnosis text using BERT, BMC Med. Inform. Decis. Mak., vol. 20, no. S7, p. 242, Sep. 2020, doi: 10.1186/s12911-020-01241-8.
197. L. Akhtyamova, P. Martinez, K. Verspoor, and J. Cardiff, Testing Contextualized Word Embeddings to Improve NER in Spanish Clinical Case Narratives, IEEE Access, vol. 8, pp. 164717–164726, 2020, doi: 10.1109/ACCESS.2020.3018688.
198. D. Zhang et al., Improving Distantly-Supervised Named Entity Recognition for Traditional Chinese Medicine Text via a Novel Back-Labeling Approach, IEEE Access, vol. 8, pp. 145413–145421, 2020, doi: 10.1109/ACCESS.2020.3015056.
199. G. Wen, H. Chen, H. Li, Y. Hu, Y. Li, and C. Wang, Cross domains adversarial learning for Chinese named entity recognition for online medical consultation, J. Biomed. Inform., vol. 112, p. 103608, Dec. 2020, doi: 10.1016/j.jbi.2020.103608.
200. C. Wang et al., Chinese medical named entity recognition based on multi-granularity semantic dictionary and multimodal tree, J. Biomed. Inform., vol. 111, p. 103583, Nov. 2020, doi: 10.1016/j.jbi.2020.103583.
201. C. Wu, G. Luo, C. Guo, Y. Ren, A. Zheng, and C. Yang, An attention-based multi-task model for named entity recognition and intent analysis of Chinese online medical questions, J. Biomed. Inform., vol. 108, p. 103511, Aug. 2020, doi: 10.1016/j.jbi.2020.103511.
202. B. Ji et al., Research on Chinese medical named entity recognition based on collaborative cooperation of multiple neural network models, J. Biomed. Inform., vol. 104, p. 103395, Apr. 2020, doi: 10.1016/j.jbi.2020.103395.
203. I. Lerner, N. Paris, and X. Tannier, Terminologies augmented recurrent neural network model for clinical named entity recognition, J. Biomed. Inform., vol. 102, p. 103356, Feb. 2020, doi: 10.1016/j.jbi.2019.103356.
204. T. Munkhdalai, M. Li, K. Batsuren, H. A. Park, N. H. Choi, and K. H. Ryu, Incorporating domain knowledge in chemical and biomedical named entity recognition with word representations, J. Cheminformatics, vol. 7, no. S1, p. S9, Dec. 2015, doi: 10.1186/1758-2946-7-S1-S9.
205. Y. Chen et al., A Boundary Assembling Method for Nested Biomedical Named Entity Recognition, IEEE Access, vol. 8, pp. 214141–214152, 2020, doi: 10.1109/ACCESS.2020.3040182.
206. S. Dadas and J. Protasiewicz, A Bidirectional Iterative Algorithm for Nested Named Entity Recognition, IEEE Access, vol. 8, pp. 135091–135102, 2020, doi: 10.1109/ACCESS.2020.3011598.
207. S. Gajendran, D. Manjula, and V. Sugumaran, Character level and word level embedding with bidirectional LSTM – Dynamic recurrent neural network for biomedical named entity recognition from literature, J. Biomed. Inform., vol. 112, p. 103609, Dec. 2020, doi: 10.1016/j.jbi.2020.103609.
208. H. Zhang, Y. Guo, and T. Li, Multifeature Named Entity Recognition in Information Security Based on Adversarial Learning, Secur. Commun. Netw., vol. 2019, pp. 1–9, Feb. 2019, doi: 10.1155/2019/6417407.
209. F. Yi, B. Jiang, L. Wang, and J. Wu, Cybersecurity Named Entity Recognition Using Multi-Modal Ensemble Learning, IEEE Access, vol. 8, pp. 63214–63224, 2020, doi: 10.1109/ACCESS.2020.2984582.
210. M. W. Al-Nabki, E. Fidalgo, E. Alegre, and L. Fernández-Robles, Improving named entity recognition in noisy user-generated text with local distance neighbor feature, Neurocomputing, vol. 382, pp. 1–11, Mar. 2020, doi: 10.1016/j.neucom.2019.11.072.
211. G. Kim, C. Lee, J. Jo, and H. Lim, Automatic extraction of named entities of cyber threats using a deep Bi-LSTM-CRF network, Int. J. Mach. Learn. Cybern., vol. 11, no. 10, pp. 2341–2355, Oct. 2020, doi: 10.1007/s13042-020-01122-6.
212. M. M. Tikhomirov, N. V. Loukachevitch, and B. V. Dobrov, Recognizing Named Entities in Specific Domain, Lobachevskii J. Math., vol. 41, no. 8, pp. 1591–1602, Aug. 2020, doi: 10.1134/S199508022008020X.
213. P. Ma, B. Jiang, Z. Lu, N. Li, and Z. Jiang, Cybersecurity named entity recognition using bidirectional long short-term memory with conditional random fields, Tsinghua Sci. Technol., vol. 26, no. 3, pp. 259–265, Jun. 2021, doi: 10.26599/TST.2019.9010033.
214. M. Li, T. Munkhdalai, X. Yu, and K. H. Ryu, A Novel Approach for Protein-Named Entity Recognition and Protein-Protein Interaction Extraction, Math. Probl. Eng., vol. 2015, pp. 1–10, 2015, doi: 10.1155/2015/942435.
215. F. Zhu and B. Shen, Combined SVM-CRFs for Biological Named Entity Recognition with Maximal Bidirectional Squeezing, PLoS ONE, vol. 7, no. 6, p. e39230, Jun. 2012, doi: 10.1371/journal.pone.0039230.
216. H. Zhou, S. Ning, Z. Liu, C. Lang, Z. Liu, and B. Lei, Knowledge-enhanced biomedical named entity recognition and normalization: application to proteins and genes, BMC Bioinformatics, vol. 21, no. 1, p. 35, Dec. 2020, doi: 10.1186/s12859-020-3375-3.
217. R. Danger, F. Pla, A. Molina, and P. Rosso, Towards a Protein–Protein Interaction information extraction system: Recognizing named entities, Knowl.-Based Syst., vol. 57, pp. 104–118, Feb. 2014, doi: 10.1016/j.knosys.2013.12.010.
218. H. Fei, Y. Ren, and D. Ji, Dispatched attention with multi-task learning for nested mention recognition, Inf. Sci., vol. 513, pp. 241–251, Mar. 2020, doi: 10.1016/j.ins.2019.10.065.
219. A. Lourenço et al., Correction: A linear classifier based on entity recognition tools and a statistical approach to method extraction in the protein-protein interaction literature, BMC Bioinformatics, vol. 13, no. 1, p. 180, Dec. 2012, doi: 10.1186/1471-2105-13-180.
220. X. Wang, Y. Li, T. He, X. Jiang, and X. Hu, Recognition of bacteria named entity using conditional random fields in Spark, BMC Syst. Biol., vol. 12, no. S6, p. 106, Nov. 2018, doi: 10.1186/s12918-018-0625-3.

221. P. Corbett and J. Boyle, Chemlistem: chemical named entity recognition using recurrent neural networks, J. Cheminformatics, vol. 10, no. 1, p. 59, Dec. 2018, doi: 10.1186/s13321-018-0313-8.
222. B. Kolluru, L. Hawizy, P. Murray-Rust, J. Tsujii, and S. Ananiadou, Using Workflows to Explore and Optimise Named Entity Recognition for Chemistry, PLoS ONE, vol. 6, no. 5, p. e20181, May 2011, doi: 10.1371/journal.pone.0020181.
223. I. Korvigo, M. Holmatov, A. Zaikovskii, and M. Skoblov, Putting hands to rest: efficient deep CNN-RNN architecture for chemical named entity recognition with no hand-crafted rules, J. Cheminformatics, vol. 10, no. 1, p. 28, Dec. 2018, doi: 10.1186/s13321-018-0280-0.
224. L. Luo et al., A neural network approach to chemical and gene/protein entity recognition in patents, J. Cheminformatics, vol. 10, no. 1, p. 65, Dec. 2018, doi: 10.1186/s13321-018-0318-3.
225. R. Fan, L. Wang, J. Yan, W. Song, Y. Zhu, and X. Chen, Deep Learning-Based Named Entity Recognition and Knowledge Graph Construction for Geological Hazards, ISPRS Int. J. Geo-Inf., vol. 9, no. 1, p. 15, Dec. 2019, doi: 10.3390/ijgi9010015.
226. Y. Yao and A. Sun, Mobile phone name extraction from internet forums: a semi-supervised approach, World Wide Web, vol. 19, no. 5, pp. 783–805, Sep. 2016, doi: 10.1007/s11280-015-0361-1.
227. H. Huang, H. Wang, and D. Jin, A Low-Cost Named Entity Recognition Research Based on Active Learning, Sci. Program., vol. 2018, pp. 1–10, Dec. 2018, doi: 10.1155/2018/1890683.
228. M. Shardlow et al., A Text Mining Pipeline Using Active and Deep Learning Aimed at Curating Information in Computational Neuroscience, Neuroinformatics, vol. 17, no. 3, pp. 391–406, Jul. 2019, doi: 10.1007/s12021-018-9404-y.
229. A. Rajkomar, J. Dean, and I. Kohane, Machine Learning in Medicine, N. Engl. J. Med., vol. 380, no. 14, pp. 1347–1358, Apr. 2019, doi: 10.1056/NEJMra1814259.
230. Y. Wang et al., Supervised methods for symptom name recognition in free-text clinical records of traditional Chinese medicine: An empirical study, J. Biomed. Inform., vol. 47, pp. 91–104, Feb. 2014, doi: 10.1016/j.jbi.2013.09.008.
231. T. H. Truong, M. H. Dao, and D. Q. Nguyen, COVID-19 Named Entity Recognition for Vietnamese, in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online: Association for Computational Linguistics, 2021, pp. 2146–2153. doi: 10.18653/v1/2021.naacl-main.173.
232. N. Colic, L. Furrer, and F. Rinaldi, Annotating the Pandemic: Named Entity Recognition and Normalisation in COVID-19 Literature, in Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020, Online: Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.nlpcovid19-2.27.
233. X. Wang, X. Song, B. Li, Y. Guan, and J. Han, Comprehensive Named Entity Recognition on CORD-19 with Distant or Weak Supervision, ArXiv200312218 Cs, Apr. 2020, Accessed: Sep. 19, 2021. Online. Available: http://arxiv.org/abs/2003.12218
234. O. Levy, Y. Goldberg, and I. Dagan, Improving Distributional Similarity with Lessons Learned from Word Embeddings, Trans. Assoc. Comput. Linguist., vol. 3, pp. 211–225, Dec. 2015, doi: 10.1162/tacl_a_00134.
235. G. Lev, B. Klein, and L. Wolf, In Defense of Word Embedding for Generic Text Representation, in Natural Language Processing and Information Systems, vol. 9103, C. Biemann, S. Handschuh, A. Freitas, F. Meziane, and E. Métais, Eds., in Lecture Notes in Computer Science, vol. 9103. , Cham: Springer International Publishing, 2015, pp. 35–50. doi: 10.1007/978-3-319-19581-0_3.
236. M. Li, Q. Lu, Y. Long, and L. Gui, Inferring Affective Meanings of Words from Word Embedding, IEEE Trans. Affect. Comput., vol. 8, no. 4, pp. 443–456, Oct. 2017, doi: 10.1109/TAFFC.2017.2723012.
237. U. Kamath, J. Liu, and J. Whitaker, Deep Learning for NLP and Speech Recognition. Cham: Springer International Publishing, 2019. doi: 10.1007/978-3-030-14596-5.
238. F. M. Hasan, N. UzZaman, and M. Khan, Comparison of different POS Tagging Techniques (n-gram, HMM and Brill's tagger) for Bangla, in Advances and Innovations in Systems, Computing Sciences and Software Engineering, K. Elleithy, Ed., Dordrecht: Springer Netherlands, 2007, pp. 121–126. doi: 10.1007/978-1-4020-6264-3_23.
239. A. A. Farzindar and D. Inkpen, Natural Language Processing for Social Media, Third Edition, Synth. Lect. Hum. Lang. Technol., vol. 13, no. 2, pp. 1–219, Apr. 2020, doi: 10.2200/S00999ED3V01Y202003HLT046.
240. B. Y. Lin, F. Xu, Z. Luo, and K. Zhu, Multi-channel BiLSTM-CRF Model for Emerging Named Entity Recognition in Social Media, in Proceedings of the 3rd Workshop on Noisy User-generated Text, Copenhagen, Denmark: Association for Computational Linguistics, 2017, pp. 160–165. doi: 10.18653/v1/W17-4421.
241. X. Li, H. Zhang, and X.-H. Zhou, Chinese clinical named entity recognition with variant neural structures based on BERT methods, J. Biomed. Inform., vol. 107, p. 103422, Jul. 2020, doi: 10.1016/j.jbi.2020.103422.
242. A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
243. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, Distributed Representations of Words and Phrases and their Compositionality. arXiv, Oct. 16, 2013. Accessed: Sep. 30, 2023. Online. Available: http://arxiv.org/abs/1310.4546
244. C. Sun and Z. Yang, Transfer Learning in Biomedical Named Entity Recognition: An Evaluation of BERT in the PharmaCoNER task, in Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, Hong Kong, China: Association for Computational Linguistics, 2019, pp. 100–104. doi: 10.18653/v1/D19-5715.
245. J. Gao, M. Li, A. Wu, and C.-N. Huang, Chinese Word Segmentation and Named Entity Recognition: A Pragmatic Approach, Comput. Linguist., vol. 31, no. 4, pp. 531–574, 2005, doi: 10.1162/089120105775299177.
246. A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
247. S. Francis, J. V. Landeghem, and M.-F. Moens, Transfer Learning for Named Entity Recognition in Financial and Biomedical Documents, Information, vol. 10, no. 8, p. 248, Jul. 2019, doi: 10.3390/info10080248.
248. J. D. Lafferty, A. McCallum, and F. C. N. Pereira, Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, in Proceedings of the Eighteenth International Conference on Machine Learning, in ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Jun. 2001, pp. 282–289.
249. S. Ananiadou et al., Named Entity Recognition for Bacterial Type IV Secretion Systems, PLoS ONE, vol. 6, no. 3, p. e14780, Mar. 2011, doi: 10.1371/journal.pone.0014780.

250. Y. Tsuruoka, J. Tsujii, and S. Ananiadou, Fast Full Parsing by Linear-Chain Conditional Random Fields, in Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), Athens, Greece: Association for Computational Linguistics, Mar. 2009, pp. 790–798. Accessed: Sep. 19, 2021. Online. Available: https://aclanthology.org/E09-1090
251. Y. Qin and Y. Zeng, Research of Clinical Named Entity Recognition Based on Bi-LSTM-CRF, J. Shanghai Jiaotong Univ. Sci., vol. 23, no. 3, pp. 392–397, Jun. 2018, doi: 10.1007/s12204-018-1954-5.
252. F. Dernoncourt, J. Y. Lee, and P. Szolovits, NeuroNER: an easy-to-use program for named-entity recognition based on neural networks, in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Copenhagen, Denmark: Association for Computational Linguistics, 2017, pp. 97–102. doi: 10.18653/v1/D17-2017.
253. Y. Qin, G. Shen, W. Zhao, Y. Chen, M. Yu, and X. Jin, A network security entity recognition method based on feature template and CNN-BiLSTM-CRF, Front. Inf. Technol. Electron. Eng., vol. 20, no. 6, pp. 872–884, Jun. 2019, doi: 10.1631/FITEE.1800520.
254. N. Greenberg, T. Bansal, P. Verga, and A. McCallum, Marginal Likelihood Training of BiLSTM-CRF for Biomedical Named Entity Recognition from Disjoint Label Sets, in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 2824–2829. doi: 10.18653/v1/D18-1306.
255. S. Misawa, M. Taniguchi, Y. Miura, and T. Ohkuma, Character-based Bidirectional LSTM-CRF with words and characters for Japanese Named Entity Recognition, in Proceedings of the First Workshop on Subword and Character Level Models in NLP, Copenhagen, Denmark: Association for Computational Linguistics, 2017, pp. 97–102. doi: 10.18653/v1/W17-4114.
256. A. Viterbi, Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, IEEE Trans. Inf. Theory, vol. 13, no. 2, pp. 260–269, 1967.
257. L. N. Smith, Cyclical learning rates for training neural networks, presented at the 2017 IEEE winter conference on applications of computer vision (WACV), IEEE, 2017, pp. 464–472.
258. K. Wang, T. Sun, and Y. Dou, An adaptive learning rate schedule for SIGNSGD optimizer in neural networks, Neural Process. Lett., pp. 1–14, 2022.
259. Y. E. Nesterov, A method of solving a convex programming problem with convergence rate O/bigl(k^2/bigr), in Doklady Akademii Nauk, Russian Academy of Sciences, 1983, pp. 543–547.
260. Warto, Muljono, Purwanto, and E. Noersasongko, Capitalization Feature and Learning Rate for Improving NER Based on RNN BiLSTM-CRF, in Proceedings - 2022 IEEE International Conference on Cybernetics and Computational Intelligence, CyberneticsCom 2022, 2022. doi: 10.1109/CyberneticsCom55287.2022.9865660.
261. J. Duchi, E. Hazan, and Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization., J. Mach. Learn. Res., vol. 12, no. 7, 2011.
262. D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, ArXiv Prepr. ArXiv14126980, 2014.
263. M. D. Zeiler, Adadelta: an adaptive learning rate method, ArXiv Prepr. ArXiv12125701, 2012.
264. W. Wang, C. Xiao, X. Lin, and C. Zhang, Efficient approximate entity extraction with edit distance constraints, in Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, in SIGMOD '09. New York, NY, USA: Association for Computing Machinery, Jun. 2009, pp. 759–770. doi: 10.1145/1559845.1559925.
265. K. Chakrabarti, S. Chaudhuri, V. Ganti, and D. Xin, An efficient filter for approximate membership checking, in Proceedings of the 2008 ACM SIGMOD international conference on Management of data, in SIGMOD '08. New York, NY, USA: Association for Computing Machinery, Jun. 2008, pp. 805–818. doi: 10.1145/1376616.1376697.
266. D. T. Larose, Discovering Knowledge in Data: An Introduction to Data Mining. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2004. doi: 10.1002/0471687545.
267. M. Althobaiti, U. Kruschwitz, and M. Poesio, Combining Minimally-supervised Methods for Arabic Named Entity Recognition, Trans. Assoc. Comput. Linguist., vol. 3, pp. 243–255, Dec. 2015, doi: 10.1162/tacl_a_00136.
268. P. Cao, Y. Chen, K. Liu, J. Zhao, and S. Liu, Adversarial Transfer Learning for Chinese Named Entity Recognition with Self-Attention Mechanism, in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 182–192. doi: 10.18653/v1/D18-1017.
269. M. Riedl and S. Padó, A Named Entity Recognition Shootout for German, in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 120–125. doi: 10.18653/v1/P18-2020.
270. W. Lee, K. Kim, E. Y. Lee, and J. Choi, Conditional random fields for clinical named entity recognition: A comparative study using Korean clinical texts, Comput. Biol. Med., vol. 101, pp. 7–14, Oct. 2018, doi: 10.1016/j.compbiomed.2018.07.019.
271. D. C. Wintaka, M. A. Bijaksana, and I. Asror, Named-Entity Recognition on Indonesian Tweets using Bidirectional LSTM-CRF, Procedia Comput. Sci., vol. 157, pp. 221–228, 2019, doi: 10.1016/j.procs.2019.08.161.
272. R. M. Syachrul M.A.K., M. A. Bijaksana, and A. F. Huda, Person Entity Recognition for the Indonesian Qur'an Translation with the Approach Hidden Markov Model-Viterbi, Procedia Comput. Sci., vol. 157, pp. 214–220, Jan. 2019, doi: 10.1016/j.procs.2019.08.160.
273. F. Y. Azalia, M. A. Bijaksana, and A. F. Huda, Name Indexing in Indonesian Translation of Hadith using Named Entity Recognition with Naïve Bayes Classifier, Procedia Comput. Sci., vol. 157, pp. 142–149, 2019, doi: 10.1016/j.procs.2019.08.151.
274. B. Wilie et al., IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding, ArXiv200905387 Cs, Oct. 2020, Accessed: Sep. 19, 2021. Online. Available: http://arxiv.org/abs/2009.05387
275. A. S. Wibawa and A. Purwarianti, Indonesian Named-entity Recognition for 15 Classes Using Ensemble Supervised Learning, Procedia Comput. Sci., vol. 81, pp. 221–228, 2016, doi: 10.1016/j.procs.2016.04.053.
276. W. Gunawan, D. Suhartono, F. Purnomo, and A. Ongko, Named-Entity Recognition for Indonesian Language using Bidirectional LSTM-CNNs, Procedia Comput. Sci., vol. 135, pp. 425–432, 2018, doi: 10.1016/j.procs.2018.08.193.
277. J. Xie, Z. Yang, G. Neubig, N. A. Smith, and J. Carbonell, Neural Cross-Lingual Named Entity Recognition with Minimal Resources, ArXiv180809861 Cs, Sep. 2018, Accessed: Sep. 19, 2021. Online. Available: http://arxiv.org/abs/1808.09861
278. C. Sabty, M. Elmahdy, and S. Abdennadher, Named Entity Recognition on Arabic-English Code-Mixed Data, in 2019 IEEE 13th International Conference on Semantic Computing (ICSC), Newport Beach, CA, USA: IEEE, Jan. 2019, pp. 93–97. doi: 10.1109/ICOSC.2019.8665500.

279. A. Zafarian, A. Rokni, S. Khadivi, and S. Ghiasifard, Semi-supervised learning for named entity recognition using weakly labeled training data, in 2015 The International Symposium on Artificial Intelligence and Signal Processing (AISP), Mashhad, Iran: IEEE, Mar. 2015, pp. 129–135. doi: 10.1109/AISP.2015.7123504.
280. G. I. Winata, C.-S. Wu, A. Madotto, and P. Fung, Bilingual Character Representation for Efficiently Addressing Out-of-Vocabulary Words in Code-Switching Named Entity Recognition, in Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching, Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 110–114. doi: 10.18653/v1/W18-3214.
281. L. H. B. Nguyen, D. Dinh, and P. Tran, An Approach to Construct a Named Entity Annotated English-Vietnamese Bilingual Corpus, ACM Trans. Asian Low-Resour. Lang. Inf. Process., vol. 16, no. 2, pp. 1–17, Dec. 2016, doi: 10.1145/2990191.
282. T.-A. Dao, H.-T. Truong, L. Nguyen, and D. Dinh, Building a Named Entity Annotated Bilingual English-Vietnamese Corpus, in 2018 10th International Conference on Knowledge and Systems Engineering (KSE), Ho Chi Minh City: IEEE, Nov. 2018, pp. 61–66. doi: 10.1109/KSE.2018.8573411.
283. M. Arkhipov, M. Trofimova, Y. Kuratov, and A. Sorokin, Tuning Multilingual Transformers for Language-Specific Named Entity Recognition, in Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing, Florence, Italy: Association for Computational Linguistics, 2019, pp. 89–93. doi: 10.18653/v1/W19-3712.
284. J. Ni and R. Florian, Improving Multilingual Named Entity Recognition with Wikipedia Entity Type Mapping, in Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas: Association for Computational Linguistics, 2016, pp. 1275–1284. doi: 10.18653/v1/D16-1135.
285. Warto, Muljono, Purwanto, and E. Noersasongko, Improving Named Entity Recognition in Bahasa Indonesia with Transformer-Word2Vec-CNN-Attention Model., Int. J. Intell. Eng. Syst., vol. 16, no. 4, 2023.