Recent Advances in Named Entity Recognition: A Comprehensive Survey and Comparative Study

Imed Keraghel
Centre Borelli UMR9010
Université Paris Cité
Paris, France
imed.keraghel@u-paris.fr

Stanislas Morbieu
Kernix Software
Paris, France
smorbieu@kernix.com

Mohamed Nadif
Centre Borelli UMR9010
Université Paris Cité
Paris, France
mohamed.nadif@u-paris.fr

Abstract

Named Entity Recognition seeks to extract substrings within a text that name real-world objects and to determine their type (for example, whether they refer to persons or organizations). In this survey, we first present an overview of recent popular approaches, including advancements in Transformer-based methods and Large Language Models (LLMs) that have not had much coverage in other surveys. In addition, we discuss reinforcement learning and graph-based approaches, highlighting their role in enhancing NER performance. Second, we focus on methods designed for datasets with scarce annotations. Third, we evaluate the performance of the main NER implementations on a variety of datasets with differing characteristics (as regards their domain, their size, and their number of classes). We thus provide a deep comparison of algorithms that have never been considered together. Our experiments shed some light on how the characteristics of datasets affect the behavior of the methods we compare.

Keywords Named Entity Recognition · Information Extraction · Natural Language Processing · Large Language Models · Machine Learning

1 Introduction

Named Entity Recognition (NER) is a subfield of computer science and Natural Language Processing (NLP) that focuses on identifying and classifying entities in unstructured text into predefined categories, such as persons, geographical locations, and organizations [Grishman and Sundheim, 1996]. Over time, NER has expanded its scope beyond proper names to include more complex concepts [Mehmood et al., 2023], particularly in specialized domains such as biomedicine. For example, in the biomedical field, NER techniques are employed to identify entities such as genes, proteins, and diseases [Mesbah et al., 2018, Luo et al., 2023]. Consequently, NER has become a crucial component in various modern applications, including machine translation [Babych and Hartley, 2003], question-answering systems (QA) [Mollá et al., 2006], and information retrieval (IR) [Guo et al., 2009].

Historically, early NER systems relied on rule-based approaches with hand-crafted rules, lexicons, and spelling features [Rau, 1991, Mikheev et al., 1999, Farmakiotou et al., 2000]. These methods, while simple and interpretable, lacked flexibility and scalability. The introduction of machine learning techniques marked a significant shift in the field, allowing more adaptable and data-driven approaches [Chieu and Ng, 2003, Nadeau and Sekine, 2007, Konkol and Konopík, 2013, Shaalan, 2014, Eltyeb and Salim, 2014]. With the rise of neural networks, NER systems further improved, particularly with the adoption of deep learning methods, which enabled more sophisticated models capable of capturing complex patterns in text [Collobert, 2011, Zhao et al., 2016, Zhang and Yang, 2018]. Most recently, Transformer-based architectures have set new standards in NER performance, leading to breakthroughs in the field [Labusch et al., 2019, Jeong and Kim, 2022, Vacareanu et al., 2024, Shi and Kimura, 2024].

This progress is reflected in the substantial growth of NER research publications in the last three decades. As shown in Figure 1, the number of NER-related publications in the ACM Computing Surveys database has grown steadily. In the mid-1990s, NER publications were relatively few, reflecting the field's early focus on rule-based systems.

The introduction of statistical models such as hidden Markov models [Baum et al., 1970] and conditional random fields [Lafferty et al., 2001] around the year 2000 brought a surge in academic interest. More recently, the period 2018-2024 saw an explosion in NER publications, driven by the adoption of Transformer-based models that improved the performance of NER systems.

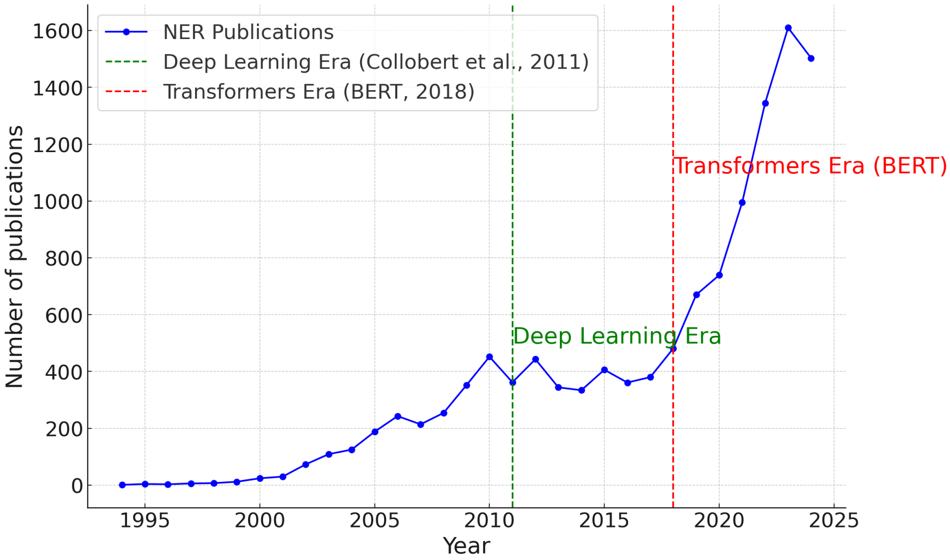

Figure 1: Growth of NER publications.

To track the evolution of NER and provide structured overviews of recent advances, several surveys have been published documenting the progression from rule-based methods to modern machine learning approaches [Nadeau and Sekine, 2007, Shaalan, 2014, Goulart et al., 2011, Marrero et al., 2013]. For example, Goulart et al. [2011] specifically reviewed advances in biomedical NER between 2007 and 2009, while Marrero et al. [2013] provided a broader theoretical and practical review of NER techniques. However, many of these early surveys predate the widespread adoption of deep learning and Transformer-based models. More recent surveys have shifted their focus to these modern approaches. For example, several reviews have concentrated on deep learning and Transformer-based methods [Pakhale, 2023, Jehangir et al., 2023, Li et al., 2020a, Nasar et al., 2021, Dai et al., 2023, Li et al., 2022]. Despite their depth, these surveys often overlook recent advances, such as the integration of Large Language Models (LLMs) and graph-based techniques. Research using graph-based methods, such as the work by Wang et al. [2022a], focuses on specific challenges such as nested entities but does not address flat-named entities. Similarly, biomedical NER surveys, such as Wang et al. [2023a], mention the use of pre-trained language models, but lack detailed analysis of their application in this domain.

A detailed review of NER in historical documentation by Ehrmann et al. [2023] examines the distinct challenges and strategies relevant to this field. This research provides significant insights into NER applied to historical texts, which often present unique issues such as language evolution, non-standardization, and inconsistent spelling. However, while it addresses these specific challenges in historical documents, the review does not comprehensively cover broader advancements in NER technologies, such as LLMs or graph-based approaches, nor does it extensively discuss strategies to manage limited annotations. Moreover, several recent works that use LLMs for NER have not yet been covered in existing surveys. For example, Wang et al. [2023b] improve few-shot NER in new domains by incorporating type-specific features, while Ashok and Lipton [2023] uses entity type definitions to enhance few-shot learning.

Another notable gap in the literature is the limited attention paid to methods that address low-resource settings, where annotated data is scarce. Producing annotated datasets is often expensive and time consuming, making it essential to develop methods that can perform effectively with limited data. To the best of our knowledge, no comprehensive survey has yet focused on techniques designed for datasets with scarce annotations. Furthermore, recent work has shown that Reinforcement Learning (RL) can help address the challenge of improving model performance in NER [Yang et al., 2023, Wan et al., 2020a]. However, this area remains underexplored in existing surveys, and only Pakhale [2023] briefly mentions the potential of RL in NER.

In this paper, our aim is to address these gaps by providing an all-encompassing review of NER techniques, from early rule-based approaches to the most recent methods, including those relying on LLMs and graph-based approaches. Our study also examines other learning paradigms such as RL. In addition, we focus on methods designed for datasets with scarce annotations and compare NER implementations across various datasets.

Our paper is structured as follows. We begin by defining the task of NER and explaining the different types of named entities. Next, in Section 4, we illustrate some NER applications. Significant methods are discussed in Section 5, with an emphasis on LLM, RL, and graph-based approaches. Section 6 covers techniques suitable for scenarios with limited annotated data. Section 7 reviews well-known tools for pre-trained models. Following the explanation of NER evaluation schemes in Section 8, we provide a variety of useful corpora to the research community in Section 9. For comparative analysis, Section 10 includes the application of the latest versions of five popular frameworks on selected datasets. The paper concludes with our findings and future perspectives in Section 12.

2 Task definition

NER is a specific task within NLP that involves identifying and categorizing named entities in a text corpus. These entities, defined as words or phrases, refer to real-world objects such as people, organizations, locations, temporal expressions, numerical values, and gene or protein identifiers in the biomedical field [Lee et al., 2004, Luo et al., 2023, Wu et al., 2024]. The main goal of NER is to detect and classify entities into predefined semantic categories.

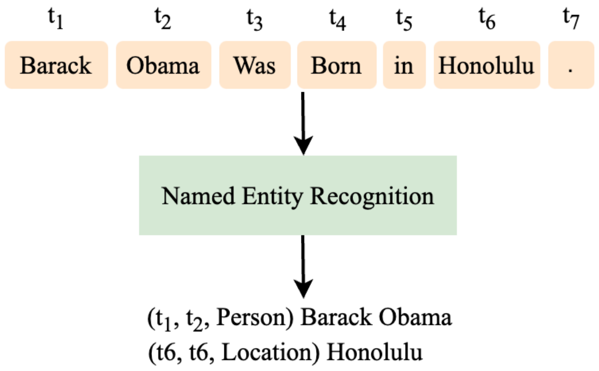

Figure 2: Given a sequence of tokens, NER outputs the boundaries of the named entities along with their associated category.

In a formal context, consider T a sequence of N tokens represented by $T = (t_1, t_2, \ldots, t_N)$. NER entails generating a set of tuples (I_s, I_e, ℓ), where s and e are integers confined to the interval $[1, N]$. Here, I_s and I_e denote respectively the start and end indices of mentions of a named entity, and ℓ indicates the category, from among a set of predefined categories, to which the entity belongs. For example, in the sentence “Barack Obama was born in Honolulu.”, NER would identify “Barack Obama” as a person’s name and “Honolulu” as a location, as shown in Figure 2.

3 Types of named entities

Named entities are categorized by their structural and contextual features. The three main types are nested, non-continuous, and continuous named entities.

3.1 Nested named entities

Nested named entities are entities contained within other entities. For example, in the sentence "Barack Obama was born in Honolulu, Hawaii." the term "Honolulu, Hawaii" is a location entity nested within another (Honolulu being the capital of the island state of Hawaii). Identifying such nested entities is crucial in specialized fields such as biomedical text mining, where entities frequently overlap and are intricately embedded within one another. The management of nested entities generally involves hierarchical models or multi-level tagging methods [Alex et al., 2007, Shibuya and Hovy, 2020, Wang et al., 2022a, Shen et al., 2021].

3.2 Non-continued named entities

Non-continued named entities take the form of singular, contiguous spans within the text. These entities constitute the most elementary form of named entities, each distinctly delineated and separable from others. For example, in the sentence "Google was founded by Larry Page and Sergey Brin." the entities "Google," "Larry Page," and "Sergey Brin" are non-continued named entities. Conventional NER systems are typically able to process such entities using sequence-labeled methods [Collobert, 2011, Liu et al., 2021, Wang et al., 2023b].

3.3 Continued named entities

Continued named entities are entities that span multiple, non-contiguous parts of the text. This can occur in cases where the entity is interrupted by other text, but still refers to the same real-world object. An example can be found in the sentence "The patient exhibited a productive cough with white or bloody sputum". Here, "cough white sputum" and "cough bloody sputum" are parts of the same symptom but are separated by other descriptive text. Recognizing continued entities requires models to understand context beyond immediate word sequences, often leveraging more advanced techniques such as attention mechanisms in Transformers [Pakhale, 2023].

4 Applications of NER

In this section, we present several illustrative examples of applications for NER.

- **Healthcare and clinical research**: NER is widely used in healthcare for extracting patient-related information from clinical notes, medical literature, and electronic health records (EHRs). By accurately identifying names of drugs, symptoms, diseases, and treatments, NER facilitates the aggregation of critical clinical data, which is essential for patient care and medical research. Liu [2023] highlighted the role of NER in improving EHR data extraction, and Jagannatha et al. [2019] introduced the first NLP challenge for extracting medication, indication, and adverse drug events from EHRs. NER can be used to pseudonymize clinical documents, ensuring that patient privacy is maintained while enabling the use of clinical data for research. A comprehensive approach combining rules with deep learning has been developed to address the challenges of de-identification in clinical data warehouses [Tannier et al., 2024].
- **Information extraction**: NER can be used to extract structured data from unstructured text, for example in retrieving the names of persons, organizations, or medical concepts. Studies such as [Nadeau and Sekine, 2007, Weston et al., 2019] have explored different ways of improving NER systems. To this end, Etzioni et al. [2005] examined the use of unsupervised learning approaches, while Weston et al. [2019] investigated the use of deep learning to increase the accuracy and efficiency of entity recognition. These studies are examples of the progress currently being made in increasing the precision of information extraction.
- **Information retrieval**: NER can significantly improve both traditional and conversational information retrieval (IR) by accurately identifying and classifying entities in user queries and responses. This improves the precision of search results and allows systems to better understand user intent. In IR, improvements come from precise identification of pertinent entities in both search queries and the resulting data [Banerjee et al., 2019, Cheng et al., 2021]. Research by Cowie and Lehnert [1996] and Etzioni et al. [2005] confirms that NER improves IR systems. In conversational systems, NER enables accurate understanding of context and user intent. For example, when a user inquires about the weather at a specific location, NER extracts the location entity, allowing the conversational assistant to provide an appropriate response. Studies such as [Jayarao et al., 2018, Cheng et al., 2019, Park et al., 2023] highlight the effectiveness of NER in improving the performance of virtual assistants-.
- **Document summarization**: Integrating NER into the process of document summarization significantly enhances the quality and pertinence of the resulting summaries. By accurately identifying and classifying key entities, NER ensures that the summary encapsulates crucial information regarding these entities. Prior research, such as [Khademi and Fakhredanesh, 2020, Liu et al., 2022, Roha et al., 2023], underscores the important role of NER in refining document summarization methods.
- **Social media monitoring**: Where it is integrated with social media monitoring, NER enables businesses to automatically identify and categorize mentions of entities like brands and persons. This can help track brand visibility, sentiment analysis, competitor insights, and crisis management. NER is also useful for spotting trends, assessing campaign effectiveness, and leveraging influencer marketing [Sufi et al., 2022]. Further studies by Ji et al. [2024] emphasize the value of NER in processing and analyzing social media data.
- **Named entity disambiguation**: NER can help disambiguate entities with the same term [Al-Qawasmeh et al., 2016]. For example, "Apple" can refer to the company or the fruit, and NER can determine the correct interpretation based on the surrounding context. Studies by Bunescu and Pasca [2006] and Dredze et al. [2010] further explore techniques for improving named entity disambiguation.

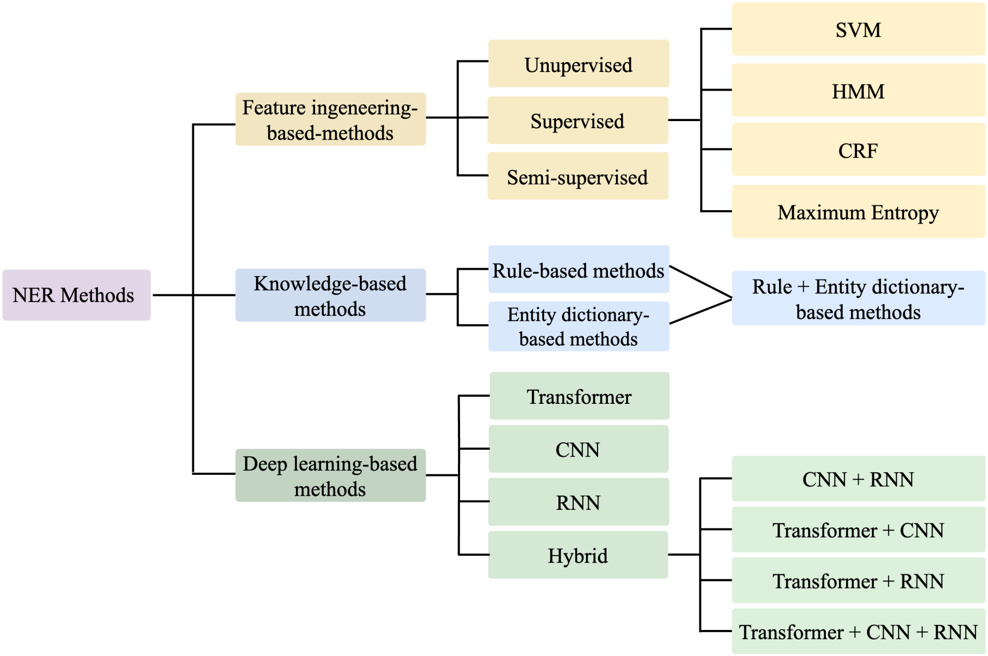

Figure 3: Main approaches to NER.

- **Question answering**: NER can play a role in answering questions that require pinpointing particular entities [Mollá et al., 2006]. For example, in a question like "When did Steve Jobs die?", NER can identify "Steve Jobs" as a person and extract the death date. Works by Mollá et al. [2006] and Zhang et al. [2016] demonstrate the value of NER in improving question answering systems.
- **Language translation**: NER can help improve the accuracy of machine translation by preserving named entities [Li et al., 2020b]. Additional work by Hkiri et al. [2017] supports the role of NER in refining language translation processes.

5 Methods

In this section, we explore the various approaches used for NER. These methodologies span a broad range of techniques, from knowledge-based systems to modern deep learning architectures. An overview of these approaches is illustrated in Figure 3.

5.1 Knowledge-based methods

Knowledge-based methods have been a foundational part of NER, particularly in the early stages of the field. These methods, which originate from linguistic principles, rely on predefined rules and lexical resources to identify named entities. For example, Borkowski [1966] presented an algorithm that uses rule-based lists and lexical markers, such as capitalization patterns, to identify company names. This rule-driven approach allows systems to detect entities based on consistent patterns, such as prefixes like "Mr." or "Ms." that signal the presence of a named entity. Another notable example is the *CasEN* transducer cascade [Maurel et al., 2011], designed specifically to recognize French-named entities.

These methods, which do not require annotated data, are primarily based on rules and gazetteers. Gazetteers serve as essential resources, providing a collection of domain-specific entities that improve recognition performance. Figure 4 shows a typical architecture used in knowledge-based NER systems. Architectures generally involve three key components: (1) a set of rules for identifying named entities, (2) optional gazetteers for additional context, and (3) an extraction engine that applies the rules and gazetteers to the input text. Despite the effort required to build these systems “manually”, they offer robust performance in well-defined domains [Btoush et al., 2016, Eftimov et al., 2017] with well-adapted gazetteers [Hanisch et al., 2005, Sekine and Nobata, 2004].

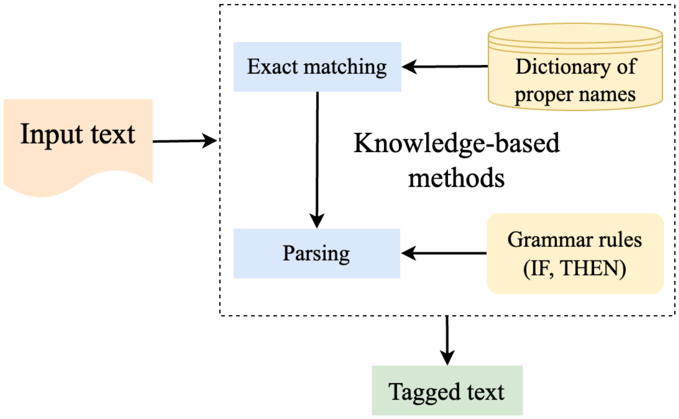

Figure 4: The architecture of knowledge-based methods for NER.

One of the key applications of knowledge-based methods is in biomedical NER, where systems such as *ProMiner* [Hanisch et al., 2005] have been developed to identify gene and protein names in texts. These systems use synonym gazetteers and specialized detection procedures to address challenges such as synonym ambiguity and case sensitivity. Similarly, Quimbaya et al. [2016] applied a gazetteer-based technique to electronic health records, obtaining an improvement in recall with minimal impact on precision (see Section 8.2).

Other studies have applied knowledge-based methods in a variety of domains. For example, a rule-based approach was used to extract dietary recommendations from unstructured text, demonstrating the flexibility and domain-specific applicability of such methods [Eftimov et al., 2017]. Hybrid methods that combine rule-based approaches with machine learning have been shown to improve the accuracy of entity recognition in complex texts such as biomedical literature [Zhang and Elhadad, 2013]. A comprehensive approach that combined rules and deep learning techniques for pseudonymizing clinical documents [Tannier et al., 2024] serves as a good illustration of the importance and challenges of deidentification in clinical data warehouses.

An interesting extension to knowledge-based approaches involves the use of factored sequence labeling to extract methodology components from AI research articles. Ghosh et al. [2023] propose a data-driven factored sequence labeling approach that leverages both ontology-based and data-driven techniques. The ontology-based technique uses predefined categories from knowledge bases such as PaperswithCode, while the data-driven variant employs clustering on sentence embeddings to dynamically identify emerging methodologies. This combined approach allows for more precise extraction of scientific concepts, particularly in dynamic fields where terminology evolves rapidly. The factorized method captures the dependencies between methodology names and their contexts, thus overcoming the limitations of traditional rule-based approaches, especially for newly introduced concepts.

For this family of methods, *precision* is typically high, while *recall* tends to be low due to domain- or language-specific rules and incomplete gazetteers. Moreover, the process of introducing new rules and gazetteers is costly. These methods, while resource-intensive, provide robust performance in well-defined domains and continue to be an essential component of NER systems. They remain relevant today, despite having been around for some time, and recent studies continue to use them, as indicated by [Wu et al., 2022a, Mengliev et al., 2023].

5.2 Feature-engineering-based methods

As NER evolved, the need to automate the extraction of named entities led to the development of methods based on feature engineering. The idea behind these approaches was to reduce manual rule-setting by focusing on identifying key features that could be used for NER tasks within the text. Feature-engineering-based methods can be broadly categorized into unsupervised, supervised, and semi-supervised learning methods, each with its own advantages and limitations.

5.2.1 Unsupervised learning methods

Unsupervised methods, which do not rely on labeled training data, attempt to identify patterns directly from the input data. This approach is particularly valuable when labeled data is scarce or unavailable, as it allows the system to uncover hidden structures within the text. A fundamental concept in unsupervised methods is the grouping of *syntagmas*, or linguistic units, based on shared properties. For example, in the phrase "the black cat," the words collectively form a noun syntagm. Such syntagms can reveal patterns in text that can be leveraged to extract named entities without prior annotation. In unsupervised learning, syntagms can be identified and grouped based on their shared characteristics, such as word order, syntactic roles, allowing the model to discern patterns and structures within the data without prior labeling.

Shinyama and Sekine [2004] demonstrated this principle by utilizing word distributions to identify named entities, particularly in news articles where named entities frequently co-occur. This method capitalizes on the tendency of named entities to appear together in multiple documents, distinguishing them from common nouns. Bonnefoy et al. [2011], calculated a semantic proximity score by comparing word distributions in documents linked to an entity and the type of entity. Nadeau et al. [2006] introduced an unsupervised system to build gazetteers and to resolve the ambiguity of named entities. Inspired by previous works such as [Etzioni et al., 2005], their approach involves amalgamating extracted gazetteers with publicly accessible gazetteers, and the resulting performance has been commendable. However, unsupervised learning methods face several limitations. First, the absence of supervision makes it difficult to assess the accuracy of the extracted named entities, where labeled data could potentially provide a clearer assessment of accuracy. Second, although word distribution methods capture named entities well, they may fail in disambiguating entities with similar surface forms but with different meanings. Finally, the complexity of syntagmatic structures means that unsupervised methods might overlook nuanced semantic differences between entities, leading to less precise groupings or associations.

5.2.2 Semi-supervised learning methods

These methods use labeled and unlabeled data to improve the effectiveness of the model. Unlike traditional supervised approaches that rely exclusively on labeled data, semi-supervised methods leverage the additional information available in unlabeled data to improve the performance of NER systems. Such methods learn from a small labeled dataset using a set of rules designed to identify extraction patterns based on a set of relevant markers. They then attempt to find other samples of named entities that are adjacent to these markers. Subsequently, the learning process is applied to the new samples to discover additional contextual markers. Repeating the process may then lead to the identification of a large number of entities. Collins and Singer [1999] demonstrated that a set of seven seed rules, coupled with unlabeled data, can be sufficient to improve model performance in a semi-supervised context.

One approach that is frequently adopted in semi-supervised NER involves co-training [Kozareva et al., 2005], where multiple classifiers are trained on different data views, and these classifier then serve to iteratively label unlabeled instances in order to build a more robust model. An alternative approach involves self-training [Gao et al., 2021], where the initial model is trained on the labeled data and then used to generate the labels for the unlabeled data. The most confident predictions are added to the labeled data and the process is iterated.

Semi-supervised learning methods offer numerous advantages, such as reducing the dependence on annotated datasets and improving outcomes in low-resource contexts. However, they also present challenges, among which we may mention the risk of propagating errors from the initial labeled data, and the requirement that handling noisy or inaccurate predictions from the unlabeled data is handled meticulously. Despite these challenges, when used judiciously, semi-supervised learning approaches can significantly improve NER performance, and they can have clear benefits in the real world, where labeled data may be scarce or costly to acquire.

5.2.3 Supervised learning methods

These methods rely on patterns derived from labeled data, which require human effort to annotate a set of samples. The labeled samples provide a basis for the model's learning process. The effectiveness of supervised learning methods depends on the quantity and quality of the labeled data used for training.

In the context of NER, the task can be divided into two main subtasks: classification and sequence labeling. During classification, the model learns to identify which words or phrases belong to specific categories of named entities. Sequence labeling, on the other hand, involves assigning a label to each token in a sentence, indicating whether it is part of a named entity and what type it is.

Several prominent models have been developed for NER tasks. These include the Hidden Markov Model (HMM) [Baum et al., 1970], which uses probabilistic methods to predict the sequence of labels [Zhou and Su, 2002, Morwal et al., 2012, Bikel et al., 1999, Zhao, 2004]. The Maximum Entropy model (MaxEnt) [Berger et al., 1996] focuses on finding the most likely label for a given word or phrase based on its context [Borthwick, 1999, Curran and Clark, 2003, Chieu and Ng, 2003, Lin et al., 2004, Borthwick et al., 1998]. Support Vector Machines (SVM) are used for classification tasks, separating named entities from non-named entities with a clear margin [Cortes and Vapnik, 1995, Makino et al., 2002, Ekbal and Bandyopadhyay, 2010, Ju et al., 2011]. Finally, Conditional Random Fields (CRF)

[Lafferty et al., 2001] consider the context of the entire sentence to make predictions about named entities [McCallum and Li, 2003, Settles, 2004, Nongmeikapam et al., 2011, Shishtla et al., 2008].

Hidden Markon Model (HMM) An HMM is a probabilistic model in which the system is assumed to operate as a Markov process. In the context of NER, HMM is used to identify and categorize named entities within a sequence of tokens. In this model, the observed tokens correspond to the observable states, while the various entity labels are regarded as hidden states. The model assumes that the observed tokens depend solely on the current hidden state, which allows the most probable sequence of named entity labels to be inferred based on the observed tokens. Mathematically, an HMM is described by five parameters:

$$\text{HMM} = \{S, O, \pi, T, E\} \tag{1}$$

where S represents the number of hidden states (entity labels), O represents the number of observations (tokens), π is the initial state probability distribution, T is the transition probability matrix, and E is the emission probability matrix. The NER problem can be reframed as an HMM problem and expressed as:

$$P(S|O) = P(N|T). \tag{2}$$

This equation posits that given a sequence of tokens T, the probability of identifying the sequence of named entities N conditional on T is the same as the probability of identifying the sequence of hidden states S conditional on observations O. Bikel et al. [1999] introduced IdentiFinder, an early HMM-based system that effectively learns to identify and categorize names, dates, times, and numerical quantities. The methodology was further developed in subsequent studies, such as [Zhou and Su, 2002, Morwal et al., 2012, Zhao, 2004].

Maximum Entropy (MaxEnt) MaxEnt models correspond to an advanced statistical method that is often used in NLP tasks, including NER. The core principle of these models is to determine a probability distribution over potential outcomes by maximizing entropy, subject to a predefined set of empirical constraints. The resulting distribution, characterized by having the highest entropy allowable under the observed constraints, is uniquely determined and coincides with the maximum likelihood distribution. It can be expressed as follows:

$$P(O|H) = \frac{1}{Z(H)} \prod_{j=1}^{k} \alpha_j^{f_j(H,O)} \tag{3}$$

where O refers to the outcome, H the context, $Z(H)$ is a normalization function, and α_j represents the weights corresponding to the features $f_j(H, O)$. The constraints are typically derived from the training data, and the model seeks to assign higher probabilities to the outcomes that are more likely given the observed data. In the context of NER, a MaxEnt model can be trained to predict the label of a named entity for a given token by considering its surrounding context and other relevant features. These features may include information about the current token, its neighboring tokens, and Part-of-Speech (POS) tags, among other elements. During training, the model learns the weights assigned to these features in order to optimize its predictions.

One of the first systems to use the MaxEnt model was presented by Borthwick et al. [1998], giving it the name *Maximum Entropy Named Entity* (*MENE*). A versatile object-based architecture allowed the integration of a wide range of knowledge sources in order to make tagging decisions, and demonstrated of the effectiveness of MaxEnt models in handling NER tasks using diverse contextual information.

Conditional Random Fields (CRF) CRFs are probabilistic models widely used for sequence labeling tasks, such as sentence annotation. CRFs take account of the interdependence of neighboring elements, which makes these models exceptionally effective for NER. By capturing sequential dependencies among tokens within a sequence, CRFs adeptly encapsulate the complex contextual relationships characteristic of named entities. Through the integration of both local and global information, CRFs facilitate the prediction of entity labels by considering adjacent token labels, thereby significantly reducing labeling ambiguity. A CRF model is expressed as follows:

$$P(Y|X) = \frac{1}{Z_0} \exp\left\{ \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_{t-1}, y_t, x, t) \right\} \tag{4}$$

where Z_0 is the normalization factor for all possible sequences of states (labels), f_k are feature functions, each representing the occurrence of a specific combination of observations and associated labels, y_{t-1} is the label of the previous word, y_t is the label of the current word, and x_t is the word at position t in the observed sequence. λ_k are the model parameters and can be interpreted as the importance or reliability of the information provided by the binary function.

The NER problem can be formulated as a CRF, where the observations are processed strings, and the labels correspond to the possible named entities. The best sequence of named entities will thus correspond to some sequence of tokens, and finding this best sequence of named entities is equivalent to finding the best sequence of labels, i.e., argmax $P(Y = y|X = x)$. Shishtla et al. [2008] implemented a system that extracts information from research articles using CRF. They investigated regularization problems using the Gaussian model and focused on the efficient use of feature space with CRF. Settles [Settles, 2004] presented a framework to recognize biomedical entities using CRF with a variety of features. They demonstrated that a CRF with only simple orthographic features could achieve good performances.

Support Vector Machine (SVM) SVMs are a class of machine learning algorithms commonly used for classification tasks. Although SVMs are not as widely used for NER as some other approaches, such as CRF models, they can still be applied effectively with appropriate feature engineering and considerations. Yamada et al. [2002] introduced a SVM-based NER system for Japanese, based on Kudo's system [Kudo and Matsumoto, 2001].

In an SVM-based NER system, each word in a sentence is classified sequentially, either from the beginning or the end of the sentence. To handle contextual dependencies, these systems typically incorporate a variety of features that capture the context around each word. These features can include:

- **Word-level features:** The actual word, its part-of-speech (POS) tag, and orthographic features (such as capitalization, presence of digits, etc.) [Asahara and Matsumoto, 2003].
- **Window features:** Information from surrounding words within a fixed-size window (e.g., the previous and next two words) [Ju et al., 2011].
- **Sentence-level features:** Global features that provide information about the sentence structure, such as sentence length or the position of the word in the sentence [Takeuchi and Collier, 2002].
- **Lexicon features:** Features derived from external knowledge bases or gazetteers that help in identifying named entities based on predefined lists [Ekbal and Bandyopadhyay, 2010].

These features, taken together, allow the SVM to consider the broader context of each word, which is crucial for accurately identifying the named entities. Moreover, by sequentially classifying each word and potentially applying post-processing steps such as Viterbi decoding [Viterbi, 1967], SVM-based systems, though not optimized for sequence prediction, can achieve effective NER performance through careful feature engineering and contextual post-processing steps.

Thus, while SVMs might not model sequence dependencies in themselves as well as CRFs do, careful feature engineering and a strategic use of contextual information can nevertheless enable them to perform NER tasks competently.

Combined techniques NER systems will often use separate supervised approaches in combination. Srihari [2000] described a hybrid strategy based on MaxEnt, HMM, and custom grammatical rules. Rule-based tagging is used for predictable patterns such as time and monetary expressions, while statistical models such as HMM handle variable entities such as names, locations, and organizations. MaxEnt is used in conjunction with features enriched by external gazetteers to improve tagging accuracy.

Srivastava et al. [2011] presented a combined NER system for Hindi that integrates CRF, MaxEnt, and rule-based methods. This solution addresses language challenges that are peculiar to Hindi, including the absence of capitalization and significant morphological complexity. A voting mechanism merges outputs from CRF and MaxEnt models with custom linguistic rules. Chiong and Wei [2006] introduced a sequential hybrid system that uses MaxEnt to initially label named entities within a corpus, providing training data for HMM to finalize the tagging. This method takes advantage of MaxEnt's strength in managing sparse data and the sequential modeling capabilities of HMM. Tests on the British National Corpus demonstrate levels of precision and recall that compare favorably with individual statistical models.

Hybrid methods have proven particularly valuable in low-resource languages, where they can effectively leverage both linguistic rules and machine learning models to improve NER accuracy. For example, a hybrid NER system for Punjabi, proposed by Bajwa and Kaur [2015], integrates rule-based methods with HMM. Developed without an existing dataset, the system involved manual tagging to create training and testing data under linguistic supervision. Two versions of NER were introduced: one using only HMM, and another combining HMM with hand-crafted rules. Similarly, to address the lack of existing resources for Arabic, Shaalan and Oudah [2014] proposed a hybrid NER system combining rule-based methods with machine learning approaches. This system recognizes 11 types of entity, including Person, Location, and Organization, by integrating decision trees [Quinlan, 1986], SVM, and logistic regression classifiers [Cox, 1958].

5.3 Deep-learning-based methods

With the rise of deep learning, NER systems have seen significant advances in both accuracy and flexibility. Deep learning methods, which rely on neural networks, have proven particularly effective in automatically learning representations of entities from large datasets. These methods often employ architectures such as Convolutional Neural Networks (CNNs) [LeCun et al., 1998] and Recurrent Neural Networks (RNNs) [Bengio et al., 1994] to capture both local and sequential patterns within the text.

The introduction of neural probabilistic language models by Bengio et al. [2003] laid the foundation for deep learning in NLP. By demonstrating the power of distributed word representations, Bengio's work showed that neural networks could capture word similarities in high-dimensional spaces, a capability that would later be crucial for deep learning-based NER systems. Building on this foundation, Collobert [2011] applied deep learning to NER, proving that CNNs could be used successfully for a wide range of NLP tasks, including NER, semantic role labeling, and chunking.

Deep learning methods for NER generally follow a three-step process, as shown in Figure 5: data representation, context encoding, and entity decoding. Each of these steps plays a critical role in ensuring that the model can accurately identify and classify entities within a text.

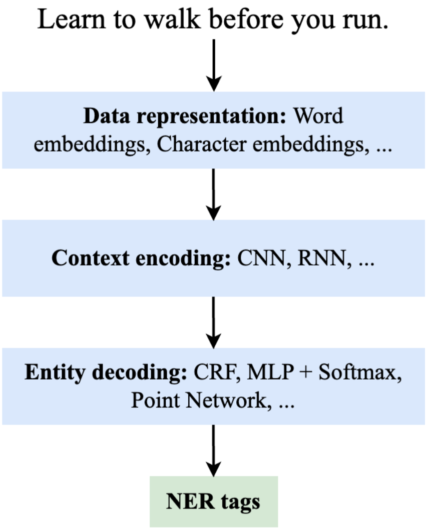
Learn to walk before you run.
Data representation: Word embeddings, Character embeddings, ...
Context encoding: CNN, RNN, ...
Entity decoding: CRF, MLP + Softmax, Point Network, ...
NER tags

Figure 5: Overview of a deep learning NER model pipeline.

5.3.1 Data representation

Before a deep learning model can process text, the data must be transformed into a format that the model can understand. This requirement can be met through the use of textual embeddings. Numerous techniques for textual embeddings in the context on NER exist, and they can be grouped into two main types: word embeddings and character embeddings.

Word embeddings transform words into dense (or sparse) vectors of real numbers in a high-dimensional space. The resulting vectors represent each word's semantic meaning based on surrounding words in a corpus. Traditional methods include techniques such as One-Hot encoding and TF-IDF [Ramos et al., 2003]. Static word embeddings, such as Word2Vec [Mikolov, 2013], GloVe [Pennington et al., 2014], and fastText [Bojanowski et al., 2016], provide a fixed vector for each word regardless of its context. Contextual embeddings, based on transformers such as GPT [Radford, 2018] and BERT [Devlin et al., 2018], capture the meaning of words based on their specific context within a sentence, allowing for more nuanced and flexible word representations.

Traditional methods for word representation focus on encoding words based on their frequency or presence in a corpus. Although semantic relationships between words are fundamental, these traditional approaches often struggle to capture these relationships, treating each term independently of its context. Two widely used traditional techniques for NER are One-Hot Encoding and TF-IDF. One-Hot encodings represent each word as a sparse binary vector with one component set to 1, indicating the word's presence in the vocabulary. In NER, One-Hot encoding is used to automatically generate training data from sources such as social media [Lee and Ko, 2020]. TF-IDF is a method that evaluates the importance of a word in a document relative to a collection of documents. It combines two metrics: Term Frequency (TF), which counts how often a word appears in a document, and Inverse Document Frequency (IDF), which measures how rare the word is across the document set. Multiplying TF by IDF highlights important terms that appear frequently in a document but are rare in the general corpus. In NER, TF-IDF can be used to represent features [Karaa, 2011]. The fact that named entities, often rare, receive higher TF-IDF scores as a result of their frequent occurrence within a document but their rare occurrence across the corpus helps them stand out in feature vectors, thus improving entity recognition.

Modern word embeddings like Word2Vec and GloVe produce dense vector representations, where similar terms have close vectors. These embeddings are derived from large corpora in an unsupervised manner. Word2Vec uses Continuous Bag-of-Words (CBOW), which predicts a target word from its context, and Skip-gram, which predicts context words from a target word. Both architectures are trained with noise contrastive estimation (NCE), a type of negative sampling. NCE increases the likelihood of the target word in context and decreases the likelihood of noise words, effectively teaching the model to distinguish between true contexts and artificially generated noise [Mikolov et al., 2013]. This contrastive approach leads to dense embeddings that are highly effective for capturing semantic similarities, as it forces the model to focus on discriminative features during training, enhancing the quality of the resulting vector representations. GloVe, on the other hand, uses matrix factorization of the word co-occurrence matrix, capturing the corpus's general statistics for efficient, high-performance embeddings [Pennington et al., 2014]. Unlike Word2Vec, which is context-based, GloVe emphasizes global word co-occurrence to produce vectors that represent the global relationships between words.

Numerous studies, including [Collobert et al., 2011, Huang et al., 2015], have used word embeddings for NER. For example, Ma and Hovy [2016] evaluated the performance of their NER system using different word embeddings such as Word2Vec and GloVe. The importance of these embeddings to achieve good performance is presented in [Lample, 2016].

Finally, we note that word embeddings can also be combined, as in [Dadas, 2019, Das et al., 2017], where a Wikipedia knowledge base is used to annotate named entities. In [Dadas, 2019], the labels are transformed into One-Hot vectors and concatenated with Word2Vec or ELMo [Peters et al., 2018] word embeddings.

Character embeddings represent words as sequences of character vectors, capturing their internal structure. Unlike traditional embeddings, they vectorize characters and combine them to form word representations, as Figure 6 illustrates in the case of a CNN architecture.

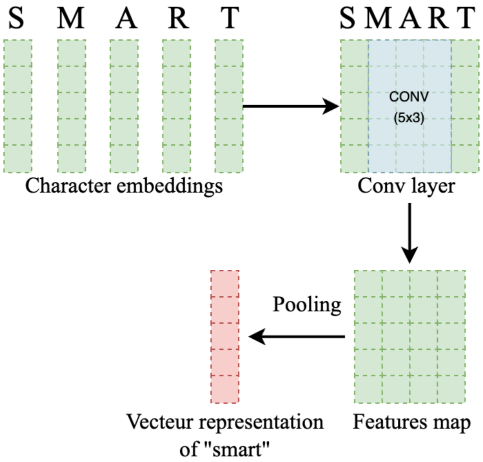

Figure 6: A CNN-based illustration of character embeddings.

Spelling variations play a crucial role, as they can reveal the presence of named entities. By representing individual characters, it is possible to capture differences in spelling indicative of a word's syntax and morphology. Character embeddings can also create representations for words not seen during training, known as out-of-vocabulary (OOV) words, by merging various character vectors to form the word.

Character embeddings can be created using several methods. One-hot encoding is a technique that represents characters as binary vectors with a 1 at the index corresponding to the character. Chars2Vec[1], inspired by Word2Vec, generates embeddings by predicting characters based on their neighboring characters. CNN-based embeddings treat character sequences as images, using CNN filters to extract meaningful features. BiLSTM-based embeddings use bidirectional

[1] https://github.com/IntuitionEngineeringTeam/chars2vec?tab=readme-ov-file

Long Short-Term Memory [Hochreiter, 1997, Graves and Graves, 2012] networks to capture contextual information from character sequences.

CNN and RNN, which we describe in the following section, are used in [Ma and Hovy, 2016, Peters et al., 2017, Zhou et al., 2023a] to calculate feature vectors for each word. One of the conclusions drawn in [Lample, 2016] is that recurrent models tend to prioritize later elements, leading to feature vectors that represent suffixes more strongly than prefixes. Consequently, the authors suggest the use of BiLSTM to more effectively capture the prefix information.

5.3.2 Context encoding

Once the data is represented as embeddings, the next step is context encoding, which focuses on capturing the relationships and dependencies between words in a sequence. Textual embeddings, based on words, characters, or both, serve as the foundation for various encoding architectures. Among the most widely used are CNNs, which capture local patterns, and RNNs, which handle sequential dependencies. However, newer models like Transformers, which offer a more advanced understanding of contextual relationships, will be discussed separately in Section 5.4.

CNN models were initially used for images [O'Shea and Nash, 2015], employing filters to detect patterns. They have since been successfully extended to NLP tasks like NER, representing text as word embeddings or characters. This transition from image processing to text is an illustration of CNNs' flexibility in its ability to capture local patterns in different types of data. CNN captures contextual information and local patterns by sliding convolutional layers over the input and using multiple filters to recognize common patterns such as suffixes, prefixes, and word combinations that are indicative of named entities.

A number of researchers have used CNN for NER tasks. Collobert et al. [2011] introduced a sentence-based network for word tagging, taking into account the whole sentence. Each word is encoded as a vector, and a convolutional layer extracts local features around every word. A global feature vector is created by merging these local features into a fixed dimension, independent of the length of the sentence. This approach of combining local features allows the network to capture both granular and holistic patterns within the sentence, a critical aspect for NER. The global features are then placed in a tag decoder for tag prediction (refer to Section 5.3.3). In [Gui et al., 2019a], the authors presented a CNN-based method for Chinese NER that incorporates lexicons and a rethinking mechanism [Li et al., 2018]. Instead of making a final decision in one pass, the rethinking mechanism includes feedback connections. These feedback loops enhance the model's decision-making by allowing it to reassess its predictions, which helps in refining difficult cases. The connections allow the network to reassess decisions by incorporating high-level feedback into feature extraction. The authors demonstrated that the method can simultaneously model characters and potential words, and the rethinking mechanism can resolve word conflicts by iteratively refining high-level features. In specialized fields such as biomedical NER, CNN-based models have also shown significant promise. For biomedical NER tasks, Zhu et al. [2018] proposed a deep learning method named GRAM-CNN. The approach leverages local contexts from n-gram character and word embeddings through CNN. Using the local context around each word, GRAM-CNN can autonomously label words without the need for specific knowledge or feature engineering.

In contrast, RNNs have been extensively employed for NER applications. While CNNs excel at capturing local patterns, RNNs are better suited for tasks that require understanding the sequential nature of text. These networks are adept at handling sequential data, making them ideal for tasks in which understanding the context of each word is vital for precise labeling [Sherstinsky, 2020]. For NER tasks, the input text is encoded as a series of embeddings, each word being sequentially put into the RNN. The RNN maintains a hidden state that encapsulates information from preceding words. As the RNN processes each word sequentially, the hidden state is continually updated to retain crucial information. However, RNNs are not without their challenges. RNNs face the challenge of the vanishing gradient problem, which hinders their ability to maintain long-term dependencies. RNN variations such as LSTM [Sherstinsky, 2020] and Gated Recurrent Unit (GRU) [Chung et al., 2014] were developed to address the problem. Both LSTM and GRU networks incorporate gating mechanisms that enhance their ability to retain and manage information over extended sequences, thereby improving their effectiveness for NER tasks.

It is for this reason that Huang et al. [2015] introduced an LSTM model for NER and showed that adding a CRF layer as a tag decoder can improve performance. This combination of LSTM for sequence modeling and CRF for decoding proved to be a highly effective strategy. In other areas, similar approaches were used by Chalapathy et al. [2016] for Drug NER and Zhang and Yang [2018] for Chinese NER. An RNN based on a BiLSTM framework was used in [Huang et al., 2015]; the same approach has since been adopted by other authors [Ma and Hovy, 2016, Lample, 2016, Huang et al., 2020a]. This widespread adoption underscores the robustness and flexibility of RNN-based models for a variety of NER applications.

5.3.3 Entity decoding

The final phase of the deep learning pipeline is entity decoding, where the model assigns entity labels to each word in the input sequence. A variety of architectures are commonly used for this task, including CRF, Multi-Layer Perceptron (MLP) and Pointer Networks [Vinyals et al., 2015]. Each architecture has its strengths, depending on the complexity of the NER task.

CRF, for example, is a probabilistic model that assigns labels by considering the entire sequence of tokens rather than making independent predictions for each one. It excels in modeling dependencies between adjacent labels, which is crucial for ensuring coherent tag sequences. By learning to assign higher probabilities to valid label sequences (such as recognizing that a "Person" label is likely to be followed by a "Location"), CRF provides a robust method for NER. Studies by Lample [2016] and Ma and Hovy [2016] show that incorporating a CRF layer into deep learning models, such as BiLSTM [Luo et al., 2018, Lin et al., 2019] or CNN [Knobelreiter et al., 2017, Feng et al., 2020], enhances performance. This combination leverages CRF's ability to ensure consistent sequence labeling while benefiting from the feature extraction power of neural networks. Many advanced NER models integrate a CRF layer with either BiLSTM or CNN for feature extraction. The BiLSTM captures the sequential context by processing the input in both forward and backward directions, ensuring that the model considers the entire sentence before making predictions. CNN, on the other hand, focuses on local patterns such as prefixes, suffixes, and word combinations, which are key in identifying entities. The CRF layer is then used to refine the final output, ensuring consistency across the sequence. This combination allows the model to capture both local features and long-range dependencies, making it more accurate and reliable.

In contrast, MLP, a simpler architecture, assigns entity labels independently to each word. The MLP transforms the sequence labeling task into a multiclass classification problem by using a softmax layer to predict the tag of each token separately. Although this approach is easier to implement and can work well for basic NER tasks, it lacks the ability to capture relationships between adjacent tokens. This limitation means that MLP models, although effective for straightforward cases, may underperform in more complex contexts where understanding token dependencies is critical, as noted by Gallo et al. [2008], Lin et al. [2019].

For more dynamic tasks, Pointer Networks represent a different approach. They are designed to handle variable-length output sequences and use attention mechanisms to directly point to elements in the input sequence rather than relying on a fixed set of output labels. This approach provides greater flexibility, particularly when dealing with sequences that involve unknown or variable output lengths, which is often the case in NER tasks. By computing soft alignment scores between input and output elements, Pointer Networks can dynamically adjust to different sequence structures, making them highly effective in tasks that require adaptable and context-specific output [Zhai et al., 2017, Li et al., 2019, Skylaki et al., 2020].

Finally, recent innovations in NER, such as the approach introduced by Fei et al. [2021], focus on token-to-token interactions rather than traditional sequential tagging. This method shifts the emphasis from token-level prediction to understanding the relationships between pairs of tokens, which allows for a deeper understanding of entity relations within a sentence. Using a multiview contrastive learning framework that aligns both semantic and relational spaces across languages, this model improves multilingual NER tasks. This evolution in NER methodologies highlights the importance of capturing more complex relationships between tokens, offering improved performance across different languages and contexts.

5.4 Transformer-based language methods

Transformer-based models have significantly reshaped NLP tasks, and in particular NER, by allowing the modeling of complex contextual relationships within text. As described in Vaswani et al. [2017], the Transformer model utilizes self-attention mechanisms to effectively capture and integrate contextual information. Transformer-based encoder models, such as BERT, have significantly influenced NER. These architectures undergo extensive pre-training on large corpora, followed by fine-tuning on domain-specific NER datasets. BERT, in particular, has demonstrated high efficacy across various NER tasks due to its ability to produce richly contextualized embeddings. Empirical studies have shown that using BERT as a classifier consistently outperforms traditional BiLSTM-CRF architectures [Riedl and Padó, 2018, Schweter and Baiter, 2019, Oralbekova et al., 2024].

Several BERT derivatives, notably DistilBERT [Sanh et al., 2019] and Robustly Optimized BERT Pre-training Approach (RoBERTa) [Liu et al., 2019a], have demonstrated strong performance in NER tasks [Abadeer, 2020, Mehta et al., 2021, Su et al., 2022, Höfer and Mottahedin, 2023]. DistilBERT, a smaller and faster variant of BERT, retains a substantial portion of BERT's language understanding capabilities. On the other hand, RoBERTa, trained on a larger dataset and benefiting from improved training techniques, generates deeper contextual embeddings and achieves superior performance across various NLP tasks, including NER. Decoding-enhanced BERT with Disentangled Attention (DeBERTa) [He et al., 2020a] offers an alternative model that has demonstrated substantial improvements in NER and various other NLP tasks. DeBERTa integrates disentangled attention mechanisms and an enhanced mask decoder, allowing for more efficient capture of word dependencies and increased model robustness. This sophisticated pre-training has enabled DeBERTa to outperform both BERT and RoBERTa on several evaluation benchmarks. PLTR [Wang et al., 2023c] employs pre-trained models such as RoBERTa to generate contextual embeddings for NER tasks, specifically avoiding recent LLMs like GPT due to their high computational costs and relatively poor performance in NER. By incorporating prompts and type-related features, PLTR improves the model's ability to generalize across domains. Another notable approach is TDMS-IE, proposed by Hou et al. [2019]. This method automates the extraction of information such as tasks, datasets, and evaluation metrics from scientific papers using BERT. By combining ontology-based and data-driven techniques, TDMS-IE excels at identifying emerging methodologies, which makes it particularly useful for constructing leaderboards in NLP research.

Generalist Model for NER using Bidirectional Transformer (GLiNER) [Zaratiana et al., 2023] constitutes a novel paradigm that integrates global contextual information into NER. The model employs a global attention mechanism that enables it to adeptly utilize long-distance entity relationships. This capability is especially advantageous in dealing with complex NER tasks that require contextual understanding that span multiple sentences. Additionally, Transformer-based models have been adapted for specific languages and domains. For example, Choudhry et al. [2022] proposed an approach for French using adversarial adaptation to overcome the lack of labeled NER datasets. By training the models on labeled source datasets and utilizing larger corpora from other domains, they succeeded in improving feature learning.

Although Transformer-based models such as BERT, RoBERTa, DeBERTa, and GLiNER have shown excellent performances in NER, they are computationally intensive. The motivation behind attempts to create more efficient models such as DistilBERT is obtaining a better balance between performance and resource requirements. Recent advances also include the development of multilingual models such as mBERT [Devlin et al., 2018], which can handle multiple languages and be fine-tuned for specific languages using bilingual gazetteers or additional datasets, enhancing performance in low-resource settings. Additionally, models such as Language Understanding with Knowledge-based Embeddings (LUKE) incorporate entity information into word embeddings, improving entity recognition by using innovative masking and self-attention techniques [Yamada et al., 2020].

These models can also be combined with other architectures to further enhance NER performance. For example, combining BERT with LSTM networks helps capture both long-term dependencies and contextual information [Souza et al., 2019, Wan et al., 2020b, He and Chen, 2021, Chen et al., 2021a]. BERT provides robust contextual embeddings, which LSTM processes to model sequential dependencies more effectively. Integrating BERT with BiLSTM improves the model's ability to capture dependencies in both forward and backward directions, improving the accuracy of NER tasks by leveraging comprehensive context from both directions [Dai et al., 2019, Chang et al., 2021, Xu et al., 2021, Lee et al., 2022, Shi and Kimura, 2024]. Using CNN with BERT can capture local patterns in the text, as the convolutional layers detect local features in the BERT embeddings, which can be particularly useful for identifying entities in specific contexts or within fixed-size windows of text.

Combining BiLSTM and CNN architectures within the same NER framework can also significantly enhance performance. In the hybrid approach, BERT embeddings are first fed into BiLSTM layers to capture long-term dependencies and bidirectional context. The output of the BiLSTM layers is then processed through CNN layers to identify local patterns and features. The combination leverages the strengths of the two architectures, that is to say BiLSTM's ability to model sequential and contextual information and CNN's efficiency in detecting local patterns. The dual architecture improves the overall performance of the NER by capturing a wide range of dependencies and contextual cues from the text [Wu et al., 2022b, Chen et al., 2022].

It is, moreover, possible to effectively integrate other RNN architectures such as GRU into NER frameworks. GRUs offer a simpler and more computationally efficient alternative to LSTM while still maintaining the ability to capture long-term dependencies [Alsaaran and Alrabiah, 2021]. Integrating BERT embeddings with GRU networks can yield similar benefits as with LSTM networks. Combining GRUs with CNN layers can further enhance the ability to detect local patterns and features in the text, resulting in improved NER performance.

5.5 Large language model-based methods

5.5.1 Principles and applications of LLMs in NER

LLMs constitute an advanced category of deep learning architectures that have the capacity to perform various tasks, including, but not limited to, translation, summarization, classification, and content generation. These models are characterized by their substantial numbers of parameters, which will often extend into the tens or hundreds of billions. They are trained on large datasets, such as GPT [Brown et al., 2020], BloomZ [Muennighoff et al., 2022], and LlaMA [Touvron et al., 2023].

Table 1: Summary of studies on LLMs in NER

Study	Approach	Outcome
GPT-NER [Wang et al., 2023b]	Transforms sequence labeling to text generation	Comparable to fully supervised baselines, better in low-resource and few-shot setups
PromptNER [Ashok and Lipton, 2023]	Uses entity type definitions for few-shot learning	State-of-the-art performance on few-shot NER, significant improvements on various datasets
ChatGPT Evaluation [Laskar et al., 2023]	Evaluates ChatGPT on various NER tasks	Impressive in several tasks, but far from solving many challenging tasks
Injecting comparison skills in TOD Systems [Kim et al., 2023]	Compares properties of multiple entities	Effectively addresses ambiguity handling in database search results
Zero-Shot on historical texts with T0 [De Toni et al., 2022]	Explores zero-shot abilities for NER	Shows potential for historical languages lacking labeled datasets, error-prone in naive approach
Resolving ECCNPs [Kammer et al., 2023]	Proposes a generative encoder-decoder Transformer	Outperforms rule-based baseline
Large code generation models [Li et al., 2023]	Uses generative LLMs of code for Information Extraction tasks	Consistently outperforms fine-tuning moderate-size models and prompting NL-LLMs in few-shot settings
UniversalNER [Zhou et al., 2023b]	Targeted distillation from LLMs	Broad coverage of entity types, suitable for clinical applications
Self-Improving Zero-Shot NER [Xie et al., 2023]	Unlabeled corpus for self-improvement	Enhanced zero-shot capabilities through self-annotated pseudo-demonstrations
GL-NER [Zhu et al., 2024]	Generation-aware LLM with label-injected instructions	Improves few-shot learning performance with novel prompt template and masking-based loss
E2DA [Zhang et al., 2024a]	Combines exogenous and endogenous data augmentation	Significantly improves performance in low-resource contexts
GPT-4 and Claude v2 [Chebbi et al., 2024]	LLMs applied to dynamic entity extraction	Adapts to new entity types in dynamic environments
CALM [Luiggi et al., 2024]	Generates additional context for entities offline	Creates relevant context to improve low-resource settings

LLMs are based on the Transformer decoder architecture, in which a multitude of attention mechanisms are orchestrated in layers to form an intricate neural network. The structural designs and pre-training paradigms implemented in current LLMs exhibit strong parallels with those used in smaller-scale language models. The primary distinction is the markedly increased size of both the model parameters and the training corpus. Some LLMs, such as T5 [Raffel et al., 2020], operate as hybrids, incorporating the encoder and decoder modules of the Transformer to increase comprehension and generative functionalities.

LLMs are lauded for their exceptional performance in various NLP tasks, including text classification [Hegselmann et al., 2023], question answering [Robinson et al., 2022], text generation [Muennighoff et al., 2022], and machine translation [Hendy et al., 2023]. However, their application to NER, a sequence labeling task, has revealed some limitations, as LLMs were originally designed for text generation.

To address the gap between LLMs and NER, various innovative methods have been developed. One notable method is GPT-NER [Wang et al., 2023b], which converts sequence labeling into a text generation task. For example, rather than directly identifying entities, the task of marking a location entity such as "Paris" is reformulated as generating a modified sentence: "@@Paris## is a city," where the special tokens "@@" and "##" indicate entity boundaries. This transformation allows LLMs to perform sequence labeling in a more natural text generation format, yielding promising results, especially in few-shot and low-resource settings, where training data are limited. Another significant development in NER involves using few-shot learning approaches, which allow models to learn from a minimal number of examples. PromptNER [Ashok and Lipton, 2023] shows how entity type definitions in prompts enable LLMs to list entities with explanations. This method has shown state-of-the-art performance in few-shot NER, demonstrating its ability to generalize across domains with minimal training data. Moreover, research including [Hu et al., 2023, Laskar et al., 2023] has investigated the potential of ChatGPT in zero-shot or few-shot clinical NER scenarios. Although initially developed for general text generation, ChatGPT has demonstrated performance on par with specialized models such as BioClinicalBERT [Alsentzer et al., 2019], although it faces certain challenges in more complex tasks.

To further advance few-shot learning, GL-NER [Zhu et al., 2024] was introduced as a generation-aware LLM specifically designed for few-shot NER. GL-NER employs a novel prompt template that incorporates label-injected instructions, enabling it to either generate entity names or to signal "does not exist" when no entity is present. In addition, it uses a masking-based loss optimization strategy that significantly improves few-shot learning performance over traditional prompt-based methods. This approach helps to tackle the inherent challenges of few-shot NER, where labeled examples are scarce.

Hybrid approaches have also been explored to combine the strengths of different LLMs. For example, BERT and GPT-2 have been used together in order to disambiguate named entities in dialogue systems [Kim et al., 2023]. In this setup, GPT-2 acts as a generator during the training phase, while BERT performs the evaluation during inference. This combination allows the model to effectively address ambiguity and entity comparison in real-time dialogue, showing the potential of integrating different LLM architectures to tackle specific NER tasks.

In more specialized contexts, such as historical and multilingual NER, an evaluation has been done of the T0 multitask model [De Toni et al., 2022]. This study highlights the unique challenges presented by historical texts, including language variations and inconsistent spellings. Although the model showed potential, particularly in its ability to identify languages and publication dates, it struggled with zero-shot NER in these specialized domains, where labeled data are often scarce. In medical domains, a successful handling of complex language constructs is crucial. [Kammer et al., 2023] proposed a generative Transformer model to address the challenge of elliptical compound nominal phrases (ECCNPs) in German medical texts. This method exhibited a significant improvement over rule-based systems, demonstrating the power of LLMs to handle specialized language constructions and to increase accuracy in medical NER tasks.

In the realm of information extraction, [Li et al., 2023] explored the use of code-based LLMs (Code-LLMs) for tasks traditionally tackled by natural language LLMs. By reframing information extraction tasks as code generation problems, Code-LLMs like Codex [Chen et al., 2021b] have shown better performance than traditional NL-LLMs in few-shot setups. Code-based LLMs show the potential of using LLMs trained in different modalities (such as code) to improve performance in specific NLP applications such as NER.

Targeted distillation techniques have also gained attention as a way to improve NER in specific domains. UniversalNER [Zhou et al., 2023b] distills the knowledge from LLMs to handle open-domain NER tasks, sampling inputs from large and diverse corpora and using ChatGPT to generate a wide variety of entity types. This makes UniversalNER suitable for applications such as clinical NER, where entity diversity is crucial. Similarly, Self-Improving Zero-Shot NER [Xie et al., 2023] introduces a framework for improving performance by leveraging an unlabeled corpus. Through self-annotated pseudo-demonstrations, the model continuously improves its zero-shot capabilities.

Among efforts seeking to improve NER through contextual information, CALM [Luiggi et al., 2024] involves generating additional context for entities offline. This approach leverages LLMs to create relevant context, especially useful in low-resource settings where the availability of annotated data is limited. [Chebbi et al., 2024] applied similar LLM-based techniques in the agricultural sector, using models such as GPT-4 and Claude v2 to monitor agricultural commodities. These models extracted and classified key entities from unstructured sources such as market reports and trade documents. Using prompt-based few-shot learning, LLMs were able to adapt to new entity types in dynamic environments without extensive domain-specific retraining.

Finally, advanced data augmentation techniques have been explored to further improve NER performance. [Zhang et al., 2024a] introduced E2DA, which combines exogenous and endogenous data augmentation. Exogenous augmentation uses LLMs to generate additional data based on specific instructions, increasing the diversity of the data set, while endogenous augmentation exploits semantic relationships within the data to maximize the use of meaningful features. This method significantly improves NER performance in low-resource contexts.

5.5.2 Pros and Cons of LLMs for NER

Recent advances in LLM-based NER approaches underscore several advantages. First, versatility and adaptability stand out as key strengths. LLMs' ability to perform few-shot and zero-shot learning is particularly advantageous in contexts where labeled data are scarce or unavailable. This capability allows these models to generalize with minimal data input, as demonstrated by few-shot models such as PromptNER and UniversalNER, which have shown good performance. This adaptability makes LLMs highly suitable for domains where annotated data are often costly or difficult to obtain.

Another important advantage is contextual understanding. The transformer-based architecture of LLMs enables these models to capture complex contextual relationships, an essential feature for NER tasks in specialized domains, such as medical or historical texts. For example, models such as Resolving ECCNPs [Kammer et al., 2023], customized for German medical texts, effectively handle unique language constructs and domain-specific terminology. This ability to understand nuanced context improves LLM performance in scenarios where accurate interpretation of specialized language is crucial.

Finally, LLMs demonstrate significant potential in multi-domain applications. Through prompt engineering and self-improvement techniques, these models have shown promising results across a range of applications, from clinical NER to information extraction in sectors like agriculture. This adaptability is particularly beneficial in fields that require dynamic knowledge, where entity types and contexts frequently change. LLMs such as those used in agricultural commodity monitoring demonstrate the practical value of adapting to new entity types without extensive retraining, which adds to their appeal in diverse applications.

However, these models are not without their challenges. A primary drawback is that LLMs are resource-intensive. The high computational and memory demands for training and deploying these models make them costly and often inaccessible for smaller-scale applications or institutions with limited resources. The large-scale infrastructure required to run models with billions of parameters may limit LLM adoption in resource-constrained environments, despite their advantages in performance. Additionally, LLMs exhibit prompt sensitivity. Many LLM-based NER approaches, such as PromptNER and UniversalNER, rely heavily on prompt engineering. Crafting effective prompts requires substantial fine-tuning and domain expertise, as small changes in prompt design can lead to significant variations in model performance. This dependence on prompt quality introduces a degree of unpredictability and limits the scalability of LLMs, particularly in settings where consistent outputs across tasks are crucial. Lastly, LLMs face limitations in complex or ambiguous contexts. Although generally effective in standard NER tasks, models like ChatGPT struggle with more intricate scenarios, especially when dealing with nuanced or domain-specific entity types. These challenges are pronounced in tasks that require a deep understanding of specific domain knowledge, where generic LLMs may not capture subtle distinctions without extensive domain-specific training. This limitation suggests a need for specialized adaptations or hybrid models to effectively address complex or ambiguous NER tasks.

5.6 Reinforcement learning

RL has emerged as a promising approach for increasing the performance and adaptability of NER systems. Agents are trained to make sequences of decisions through the rewarding of desirable actions and the penalizing of undesirable ones. In the context of NER, RL can optimize the identification and classification of named entities by learning from interactions with data and progressively improving the model's performance through rewards and penalties.

Combining RL with entity triggers, Yang et al. [2023] proposed a Gaussian Prior Reinforcement Learning framework (GPRL) to learn the order of recognition of the entity and use the boundary positions of nested entities. GPRL converts nested NER into an entity triplet sequence generation task using BART [Shen et al., 2021] with a pointer mechanism [Straková et al., 2019]. To improve nested entity recognition, a Gaussian prior adjustment is applied to the probability of the boundary of the entity predicted by the pointer network. The recognition order is modeled as an RL process, optimizing the network by maximizing triplet generation rewards.

Another significant approach is to integrate distant supervision and RL. Distant supervision addresses the scarcity of labeled data by using external resources to automatically generate annotated data sets, but it is a process that will often introduce noise [Wan et al., 2020a]. RL helps mitigate the noise by employing confidence calibration strategies to refine the model's predictions, thereby improving the overall performance of NER systems.

It is also possible to combine RL with adversarial training. Adversarial training consists in training models to distinguish between true and false positives in a competitive setting, which can improve resilience to noisy and incomplete annotations. In the biomedical domain, RL improves the recognition of complex medical entities, handling a wide range of biomedical terms and variations, thus improving the extraction of meaningful entities from medical texts [Peng et al., 2021].

Studies such as [Yang et al., 2018] have shown that RL can effectively handle incomplete and noisy annotations. For example, by integrating partial annotation learning, RL reduces the impact of unknown labels. An RL-based instance selector can filter out noisy annotations, further refining the model's accuracy.

The SKD-NER model is an example of advanced techniques in continual learning for NER [Chen and He, 2023]. This model addresses the problem of catastrophic forgetting, where a model trained to identify new entities loses its ability to recognize previously learned ones. SKD-NER uses knowledge distillation to maintain memory and applies RL strategies during this process. It fine-tunes soft labeling and distillation losses produced by the teacher model, effectively mitigating catastrophic forgetting during continual learning.

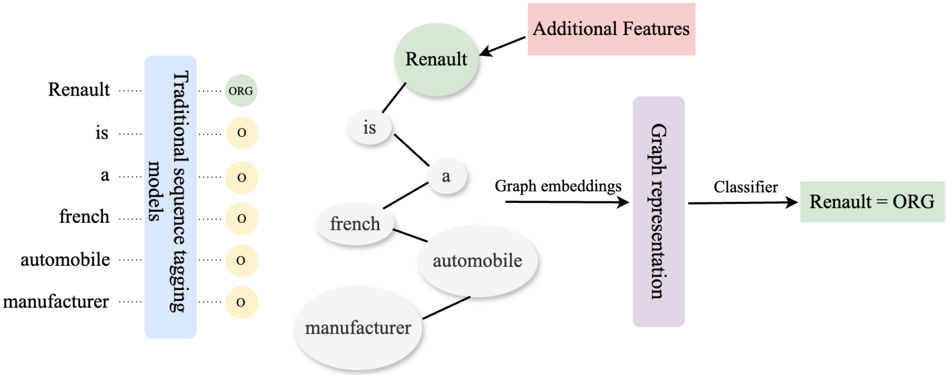

Figure 7: Left side: Conventional model for sequence tagging. Right side: Every word within a sentence transforms into a node of a graph connected to surrounding words and additional features like grammatical characteristics. This graph is subsequently encoded and fed into a classifier to predict entity tags.

5.7 Graph-based methods

The use of Graph Convolutional Networks (GCNs) in NLP allows sequences to be represented as graph structures. In these methods, each token in a sentence is treated as a node, and edges represent relationships between them (see Figure 7). This approach was first used in NLP by Marcheggiani and Titov [2017], leveraging GCNs as proposed by Kipf and Welling [2016]. Cetoli et al. [2017] demonstrated that applying GCNs to NER tasks was able to improve performance significantly, particularly on the OntoNotes 5.0 dataset [Weischedel et al., 2013]. This representation transforms sequential data into richer structures, capturing contextual dependencies effectively.

Graph embeddings have been widely adopted to strengthen token representation in NER. Liu et al. [2019b] proposed a GCN-based architecture that generates embeddings from graph structures. These graph embeddings were then combined with token embeddings and processed through a BiLSTM-CRF network to predict NER tags. Similarly, Harrando and Troncy [2021] viewed the NER task as a graph classification problem, where each token in a sequence serves as a node, and features such as morphological shape or POS-tag are incorporated to enrich the node representation. These approaches exhibited better token representations and led to better entity recognition outcomes.

Further advances, particularly for Chinese NER, have led to the development of specialized graph-based networks, such as the Polymorphic Graph Attention Network (PGAT) by Wang et al. [2022b]. PGAT improves character representation by dynamically integrating lexicon information. Another approach, the Lexicon-Based Graph Neural Network (LGN) by Gui et al. [2019b], constructs connections among characters, leveraging both local compositional structures and global sentence semantics. This approach demonstrated significant performance improvements across various Chinese NER datasets. Furthermore, the Multi-Graph Collaborative Network (MGCN) of Zhang et al. [2022] addresses challenges like boundary confusion and irrelevant lexical words, outperforming existing models by constructing multiple relationships between lexical words and characters in order to better capture entity boundaries.

Multimodal NER presents additional challenges, such as dealing with both textual and visual data. Zhao et al. [2022] introduced the Relation-enhanced GCN (R-GCN) for Multimodal NER (MNER), using inter-modal and intra-modal relation graphs to effectively integrate text and image information. This method allows for better identification of entities in cases where information is spread across multiple modalities. Another challenge is embedding newly emerging entities in existing knowledge graphs. The VN Network introduced by He et al. [2020b] addresses this problem by generating "virtual neighbors" for unseen entities. This approach involves creating connections based on logical rules and utilizing Graph Neural Networks (GNNs) to aggregate information from these synthetic neighbors, thus improving the representation of new entities.

5.8 Annotation schemes

NER relies on precise annotation schemes to effectively segment and label entities within a text. These schemes serve as the foundation for determining what constitutes an entity and where it begins and ends within a sentence. Choosing the appropriate annotation scheme is crucial for ensuring that the model can accurately perform the sequence-labeling task. Different schemes are designed to handle the complexity of identifying multi-token entities and distinguishing between entity and non-entity tokens.

In most annotation schemes, the first token of a named entity is tagged as B (Begin), marking the start of the entity. If the entity spans multiple tokens, the intermediate tokens are labeled as I (Inside), while the last token can be tagged as either E (End) or I, depending on the specific schema. Tokens that do not belong to any named entity are assigned the tag O (Outside). Annotation schemes vary in their use of combinations of tags in order to handle different contexts. Commonly used schemes include BIO (Begin, Inside, Outside), IO (Inside, Outside), IOE (Inside, Outside, End), IOBES (Inside, Outside, Begin, End, Single), IE (Inside, End), and BIES (Begin, Inside, End, Single). In some implementations, the End tag is denoted as L (Last) rather than E. Each of these schemes offers different advantages depending on the nature of the entities and the complexity of the text being analyzed.

- The IO scheme is the simplest method in which each token receives either an I or an O tag. The I tag represents named entities, while the O tag represents other words. One limitation of this schema is its inability to differentiate between consecutive entity names of the same type.
- The BIO schema, widely used and adopted in the CoNLL Conference, assigns one of three tags to each token: B for the start of a named entity, I for inside tags within the entity, and O for outside tags indicating non-entity words.
- The IOE schema is similar to BIO, but instead of marking the start of a named entity (B), it denotes the end of the entity (E).
- IOBES serves as an extension of the IOB scheme, offering more information regarding the boundaries of named entities. It uses five tags: B for the beginning of an entity, I for inside tags within the entity, E for the end of an entity, S for single-token entities, and O for non-entity words outside named entities.
- The IE scheme functions similarly to IOE, with the difference that it labels the end of non-entity words as E-O and the rest as I-O.
- The BIES scheme is an extension of IOBES. It utilizes tags such as B-O for the beginning of non-entity words, I-O for inside tags within non-entity words, E-O for the end of non-entity words, and S-O for single non-entity tokens located between two entities.

Table 2: Comparison of different annotation schemes on a sample sentence, where "PER" denotes a person and "ORG" represents an organization.

Words	**IO**	**BIO**	**IOE**	**IE**	**IOBES**	**BIES**
Emma	I-PER	B-PER	I-PER	I-PER	B-PER	B-PER
Charlotte	I-PER	I-PER	I-PER	I-PER	I-PER	I-PER
Duerre	I-PER	I-PER	I-PER	I-PER	I-PER	I-PER
Watson	I-PER	I-PER	E-PER	E-PER	E-PER	E-PER
was	O	O	O	I-O	B-O	B-O
born	O	O	O	I-O	I-O	I-O
in	O	O	O	E-O	E-O	E-O
Paris	I-ORG	B-ORG	I-ORG	I-ORG	S-ORG	S-ORG
.	O	O	O	I-O	B-O	S-O

For a comparison of these annotation schemes, we refer the reader to Table 2, which illustrates the application of the different schemes to an example sentence.

It is worth noting that the choice of tag scheme can also affect NER performance. For example, Alshammari and Alanazi [2021] found that the IO scheme outperforms other schemes for articles written in Arabic. Similarly, Chen et al. [2021c] demonstrated that the IO scheme is more suitable for steel e-commerce data compared to the BIO and BIEO schemes.

6 Low-resource NER

Although neural networks and Transformer-based models have shown remarkable success in NER, their effectiveness is often tied to the availability of large annotated datasets. However, in many real-world scenarios, especially for low-resource languages or specialized domains, such datasets are either limited or nonexistent. This presents a significant challenge for NER systems, as lack of sufficient data can hinder model training and performance.

To address these challenges, several strategies have been proposed to overcome data scarcity issues in low-resource NER settings. Using these techniques, NER models can be adapted to perform effectively even when faced with limited labeled data, ensuring that entity recognition can be applied in a wider range of contexts and languages.

6.1 Transfer learning

Transfer learning applies knowledge from one task in order to improve performance on a related task. It is a technique that has proven useful in areas such as image classification [Shaha and Pawar, 2018], speech recognition [Wang and Zheng, 2015], and time series classification [Fawaz et al., 2018]. In NER, transfer learning is implemented by first pre-training a model on a large corpus of generic text data, followed by fine-tuning on a smaller dataset specifically designed for the target NER application. This approach utilizes the broad knowledge gained from the generic dataset so as to obtain better performance on the specific task at hand. One example of transfer learning is BERTweet, which is a BERT-based model pre-trained on Twitter data that achieves good performances on Twitter text classification, as well as on POS-tagging and named-entity recognition [Nguyen et al., 2020].

Numerous studies have examined the application of transfer learning to NER. Lee et al. [2018] looked at transfer learning through RNN in the context of anonymization of health data. Studies such as Francis et al. [2019] have explored the implementation of Transformers for NER, demonstrating that transfer learning significantly improves performance. More recently, Fabregat et al. [2023] proposed a range of architectures based on BiLSTM and CRF to detect biomedical named entities using negation-based transfer learning techniques. Specifically, this approach incorporates negation detection as a pre-training step, where weights relating to negation triggers and scopes are transferred, resulting in improvements in both NER and Relation Extraction (RE) tasks.

6.2 Data augmentation

Data augmentation is a technique that is used to artificially increase the size of a training dataset by creating modified versions of existing data. This can involve applying small transformations to the original data, such as synonym replacement, random deletion, insertion, swap, back-translation, or lexical substitution [Dai and Adel, 2020, Sawai et al., 2021, Duong and Nguyen-Thi, 2021]. Generative models can also be used to synthesize entirely new examples, further enriching the training set [Sharma et al., Keraghel et al., 2020].

The application of data augmentation techniques to NLP has been explored in various areas, including text classification [Dai and Adel, 2020, Karimi et al., 2021], machine translation [Sawai et al., 2021], and sentiment analysis [Duong and Nguyen-Thi, 2021]. However, unlike other NLP tasks, NER involves making predictions about words rather than sentences. This brings with it an additional challenge, since applying transformations to words can alter their labels, and this complicates the use of data augmentation for NER. Despite this difficulty, adaptations of data augmentation techniques for NER have been done. For example, Dai and Adel [2020] applied simple strategies like word replacement and named entity replacement to improve model performance, particularly for datasets containing very few examples. In a similar study, Chen et al. [2020] applied Local Additivity based Data Augmentation (LADA), which generates new samples by combining similar ones. LADA has two variants: Intra-LADA and Inter-LADA. Intra-LADA creates new sentences by exchanging words within a single sentence and interpolating between these new sentences, whereas Inter-LADA combines different sentences to construct new data. More complex strategies, such as paraphrasing, have also been employed to generate new data [Sharma et al.].

Data augmentation is a vital technique in training NER models, especially when dealing with limited datasets. In the future, data augmentation in NER might potentially be made much easier and more effective through the use of LLM. LLMs can generate realistic and diverse text data, which can be used to augment existing datasets for NER training. This approach, as exemplified by techniques such as GPT3Mix [Yoo et al., 2021], allows for the creation of more accurate NER models by enriching training data with a wide range of linguistic variations and contextual scenarios.

6.3 Active learning

Active learning is a form of semi-supervised learning in which the learning algorithm can select the data from which it wants to learn, potentially improving its performance with respect to traditional learning approaches. One of the primary challenges in active learning is determining which data points are the most informative. The most widely used strategy today is uncertainty sampling [Settles, 2009], where the model selects examples for which its current predictions are the least certain. This strategy is effective because it focuses the learning on the most ambiguous examples, which helps refine the model's understanding.

When active learning is applied successfully to NLP, it can either improve model performance with the same amount of labeled data, or alternatively maintain similar performance while reducing the amount of data and annotation necessary. In deep learning research, active learning techniques have shown promising results. For example, Siddhant and Lipton [2018] explored the use of uncertainty-based strategies such as Least Confident (LC) and Monte Carlo Dropout Bayesian Active Learning (DO-BALD), demonstrating the ability of these techniques to reduce the annotation requirement in deep learning models. Shen et al. [2017] introduced Maximum Normalized Log-Probability (MNLP), an improvement over LC that normalizes uncertainty scores by sequence length, which is particularly effective in sequence-labeling tasks like NER.

In the context of NER, several deep learning approaches have adopted active learning strategies to improve model performance with limited annotations. For example, [Yan et al., 2023] investigated approaches that combine uncertainty- and diversity-based sampling to efficiently select the most informative examples for sequence labeling. Their work illustrates how gradient embeddings and clustering techniques, such as weighted k-means++, can be used to achieve a balance between informative and diverse sample selection, thus enhancing learning efficiency in deep active learning setups.

6.4 Few-shot learning

Few-shot learning aims to build accurate machine learning models with minimal training data. This technique can be implemented by applying transformations to the data, applying changes to the algorithms, or using dedicated algorithms [Wang et al., 2020]. Applying transformations to the data involves generating new data from the training data using data augmentation or a generative network. Changes in the algorithms involve using pre-trained models as feature extractors or refining already trained models with new data through continued backpropagation. Dedicated algorithms involve networks that learn from pairs or triplets of instances rather than single instances, thereby leveraging a larger training set.

In the context of named entities, studies such as Fritzler et al. [2019] and Hou et al. [2020] have proposed the adaptation of prototypical networks [Snell et al., 2017] for NER. However, these implementations were unable to achieve optimal performance. Yang and Katiyar [2020] introduced a few-shot learning method based on nearest neighbors and structured inference that is shown to be superior to classical meta-learning approaches. Cui et al. [2021] approached the NER task as a language template classification problem, outperforming traditional sequence-labeling methods. An increasing number of works are recognizing the potential of few-shot learning in NER [Hofer et al., 2018, Huang et al., 2020b, He et al., 2023].

The emergence of LLMs, such as those utilized in PromptNER [Ashok and Lipton, 2023], has further advanced few-shot learning in NER. These LLMs employ prompt-based methods and Chain-of-Thought prompting, significantly enhancing adaptability and performance in few-shot settings without extensive dataset requirements.

In this context, the CoTea framework [Yang et al., 2024] represents a novel approach for low-resource NER, utilizing a divide-and-conquer strategy with two collaborative teacher models to improve training with minimal labeled data. CoTea leverages external knowledge and employs a mining refinery mechanism to iteratively improve label quality, thereby reducing noise and increasing performance, achieving competitive results even in extremely low-resource settings.

6.5 Zero-shot learning

Zero-shot learning uses a pre-trained model to assign classes to elements that the model has never encountered previously [Larochelle et al., 2008, Lampert et al., 2013, Ding et al., 2017]. This approach has been explored for linking entities [Wu et al., 2020] and typing named entities [Obeidat et al., 2019] (i.e., attributing a semantic label to a given entity).

Zero-shot learning can be applied in NER to detect new types of named entities. Aly et al. [2021] proposed an architecture using textual descriptions. The ZERO model [Van Hoang et al., 2021] performs both zero-shot and few-shot learning by incorporating external knowledge through semantic representations of words. Yang et al. [2022] proposed multilingual sequence translation as a solution for low-resource languages, where labeled data is scarce or absent. This method acts as a bridge by transferring knowledge from a source language to a target language with ample annotated data. Furthermore, the rise of prompt-based learning methods, as detailed in [De Toni et al., 2022], has introduced a new paradigm in training and fine-tuning LLMs for applications like NER, enhancing the capabilities of zero-shot learning in this area. In a recent study [González-Gallardo et al., 2023], ChatGPT was evaluated for its zero-shot capabilities in NER on historical documents. The study found that, while ChatGPT has some capacity to identify entities, it struggles to cope with difficulties such as inconsistencies in annotation guidelines, complex entities, code-switching, and the accessibility of historical archives. Other references are explained in Section 5.5.

7 Software frameworks

NER has evolved significantly, and today a number of software frameworks provide robust tools to build and deploy NER systems. These frameworks simplify the development process by offering pre-built models, customizable pipelines, and support for various languages and domains. In this section, we present some of the most widely recognized and commonly used NER frameworks, highlighting their key features and capabilities.

- **OpenAI** [OpenAI] offers a range of AI tools, including GPT models, for text generation, question answering, and more. Although not originally focused on NER, OpenAI models can be adapted for NER tasks through fine-tuning or prompt engineering. The API is known for its flexibility and user-friendliness, with an additional emphasis on safe, ethical AI use.
- **spaCy** [Honnibal and Montani, 2017] is a free open-source library for advanced NLP in Python. It is designed to make it easy to construct systems for information extraction or general-purpose NLP. spaCy offers multiple analysis tools, such as tokenization, classification, POS tagging, and NER. In addition to the entities included by default, spaCy allows the addition of new classes by training the models on new data. A variety of pre-trained models are available, which can either be used directly for tasks such as NER or re-trained on specific datasets. These models are based on CNNs or Transformers.
- **NLTK** [Bird, 2006] is a suite of Python modules for NLP, integrating more than 50 corpora and lexical resources such as WordNet, as well as a suite of tools for text analysis, including tokenization, POS tagging, sentiment analysis, topic segmentation, and speech recognition. Unlike spaCy, which includes built-in algorithms tailored to various tasks, NLTK provides flexibility by allowing users to choose among a wide range of algorithms.
- **Stanford CoreNLP** [Manning et al., 2014] is a library developed by the associated research group at Stanford University. It is a set of natural language analysis tools written in Java, supporting tokenization, POS tagging, and training models for NER (based on CRF). NER features are only available for specific languages: English, Spanish, German, and Chinese, as each language has its unique set of characteristics.
- **Apache OpenNLP** [Baldridge, 2005] is a library that supports common NLP tasks such as NER, language detection, POS tagging, and chunking. Unlike other frameworks, which may use a single model for all entity types, OpenNLP provides a specialized maximum entropy model for each named entity type.
- **Polyglot** [Al-Rfou et al., 2015] is an NLP pipeline for Python. It can handle a much wider range of languages than other frameworks, supporting NER in over 40 languages.
- **Flair** [Akbik et al., 2019] is a free, open-source library that enables the creation of NLP pipelines for multilingual applications. Flair allows the stacking of embeddings, meaning users can combine different embeddings (such as Flair, ELMo, and BERT) to improve NER performance. It supports various language models, including Flair embeddings, ELMo, and BERT.
- **Hugging Face** [Wolf et al., 2020] provides open-source NLP technologies. It offers both free and paid services aimed at businesses. The framework is particularly known for its Transformers library, which offers an API for accessing numerous pre-trained models, as well as the Datasets library, which simplifies managing datasets for NLP tasks. Hugging Face also includes a collaborative platform where users can create, train, and share their deep learning models.
- **Gate** [Cunningham, 2002] is a tool written in Java. It is used by a number of NLP communities for different languages. Gate provides an information extraction system, known as ANNIE, which is able to recognize several types of entities (people, places, and organizations).
- **TNER** [Ushio and Camacho-Collados, 2021] is a Python library for training and tuning NER models implemented in PyTorch. It features a web application with an intuitive interface that allows users to visualize predictions.
- **GliNER** [Zaratiana et al., 2023] is a specialized NER model that leverages bidirectional Transformers, such as DeBERTa, for NER. Unlike traditional autoregressive models, GliNER supports parallel processing of entity spans, making it more efficient in resource-limited scenarios. It is designed to identify a wide variety of entity types by matching entity embeddings with text spans in a shared latent space.

Packages like Apache OpenNLP, Stanford CoreNLP, and spaCy are also accessible in languages other than Python. For example, openNLP[2] is an R package that takes advantage of the capabilities of the Apache OpenNLP library, originally Java-based, by acting as an interface within the R environment. Similarly, the spacyr[3] package connects R to spaCy. Notably, the spacyr package facilitates NER using spaCy's pre-trained language models. Other solutions like the reticulate[4] package make it easier to achieve interoperability between R and Python, enabling Python libraries such as Hugging Face to be accessible within R.

[2] https://cran.r-project.org/web/packages/openNLP/index.html

[3] https://cran.r-project.org/web/packages/spacyr/index.html

[4] https://cran.r-project.org/web/packages/reticulate/index.html

8 Evaluation of NER systems

Evaluation of NER systems requires an annotation scheme, an evaluation strategy, and metrics. Each of these requirements is discussed below.

8.1 Evaluation strategies: exact or relaxed evaluation

The evaluation of NER systems is based on comparing predictions with a gold standard and typically employs one of two strategies: exact evaluation or relaxed evaluation.

- **Exact Evaluation**: In this approach, both the boundaries and the type of the named entity must match the gold standard accurately. This stringent method requires a perfect alignment between the predicted entity and the reference and is commonly used in the CoNLL-2003 evaluation [Tjong Kim Sang and De Meulder, 2003].
- **Relaxed Evaluation**: This approach allows partial credit, giving points when either the type or the boundaries are correct, even if both are not. Relaxed evaluation is often used in the MUC [Grishman and Sundheim, 1996] and ACE [Doddington et al., 2004] standards.

8.2 Metrics

Classical metrics such as precision, recall, and F1 score are often used for evaluating named entities:

- **Precision**: The proportion of named entities correctly recognized by the model in relation to the total number of named entities recognized. This metric reflects how many of the entities identified by the model are actually correct.
$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$
where TP is the number of True Positives and FP is the number of False Positives.
- **Recall**: The proportion of relevant named entities correctly retrieved by the model in comparison to the total number of relevant named entities in the dataset. This metric indicates how many of the actual entities the model successfully identified.
$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$
where FN is the number of False Negatives.
- **F1 score**: Reflects a model's effectiveness in detecting named entities by balancing precision and recall. It is calculated as the harmonic mean of precision and recall, providing a single measure that combines both metrics.
$$\text{F1} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{7}$$

These metrics can be computed for each class of entities and can be aggregated when considering more than one type of entity:

- **Macro-average**: The metric (e.g., F1 score) is computed for each class separately, and the macro-average is the mean of these values. This approach treats all classes equally, regardless of their frequency in the dataset.
- **Micro-average**: This method gives equal weight to each individual sample by pooling all predictions across classes before calculating the metrics.

To go beyond these aggregated metrics, [Fu et al., 2020] proposed a new evaluation method involving a set of attributes possessed by entities (such as length or density). They found that models often have a better correlation with some attributes than with others, providing deeper insight into model performance.

9 NER datasets

Named entities often belong to broad categories, such as persons, locations, and organizations. However, categories can be much narrower than this: for example, they might correspond to books, periodicals, magazines, etc. Table 9 provides an overview of several English NER datasets, with between one and 505 types of entities in various domains such as medical data, news, social media, and more.

Table 3: Datasets for English NER. Datasets highlighted in gray are those selected for our study.

Dataset	Year	Domain	Tags	URL
MUC-6	1995	News	7	https://cs.nyu.edu/~grishman/muc6.html
MUC-7	1997	News	7	https://catalog.ldc.upenn.edu/LDC2001T02
NIST-IEER	1999	News	3	https://www.nist.gov/el/intelligent-systems-division-73500/ieee-1588
CoNLL-2002	2002	News	4	https://www.clips.uantwerpen.be/conll2002/ner/‘
CoNLL-2003	2003	News	4	https://www.clips.uantwerpen.be/conll2003/ner/
GENIA	2003	Medical	5	http://www.geniaproject.org/genia-corpus
NCBI Disease	2014	Medical	1	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3951655/
i2b2-2014	2015	Medical	32	https://www.i2b2.org/NLP/DataSets/Main.php
BC5CDR	2016	Medical	2	https://biocreative.bioinformatics.udel.edu/tasks/biocreative-v/track-3-cdr/
MedMentions	2019	Medical	128	https://github.com/chanzuckerberg/MedMentions
BioNLP2004	2004	Bioinformatics	5	https://www.ncbi.nlm.nih.gov/research/bionlp/Data/
ACE 2004	2005	Various	7	https://catalog.ldc.upenn.edu/LDC2005T09
ACE 2005	2006	Various	7	https://catalog.ldc.upenn.edu/LDC2006T06
OntoNotes 5.0	2013	Various	18	https://catalog.ldc.upenn.edu/LDC2013T19
MultiCoNER	2022	Various	33	https://multiconer.github.io/
WikiGold	2009	Wikipedia	4	https://aclanthology.org/W09-3302
WiNER	2012	Wikipedia	4	https://github.com/ghaddarAbs/WiNER
WikiFiger	2012	Wikipedia	112	https://orkg.org/paper/R163134
Few-NERD	2021	Wikipedia	66	https://github.com/thunlp/Few-NERD
HYENA	2012	Wikipedia	505	https://aclanthology.org/C12-2133.pdf
WikiAnn	2017	Wikipedia	3	https://aclanthology.org/P17-1178/
WNUT 2017	2017	Social media	6	https://noisy-text.github.io/2017/emerging-rare-entities.html
MalwareTextDB	2017	Malware	4	https://statnlp-research.github.io/resources/
SciERC	2018	Scientific	6	http://nlp.cs.washington.edu/sciIE/
HIPE-2022-data	2022	Historical	3	https://github.com/hipe-eval/HIPE-2022-data
MITMovie	2013	Queries	12	http://groups.csail.mit.edu/sls/
MITRestaurant	2013	Queries	8	http://groups.csail.mit.edu/sls/
FIN	2015	Financial	4	https://aclanthology.org/U15-1010/

In the remainder of our survey and in our experiments, we make use of the following datasets obtained from various sources:

- **CoNLL-2003**: This dataset consists mainly of news articles from Reuters.
- **OntoNotes 5.0**: A comprehensive dataset comprising various genres of texts including phone conversations, newswires, newsgroups, broadcast news, broadcast conversations, weblogs, and religious texts.
- **WNUT2017**: This dataset includes texts from various sources, such as tweets, Reddit comments, YouTube comments, and StackExchange.
- **BioNLP2004**: A biomedical dataset comprising 2000 abstracts from the MEDLINE database, annotated for NER.
- **FIN**: A dataset containing financial documents released by the US Securities and Exchange Commission, annotated specifically for financial named entities [Alvarado et al., 2015].
- **NCBI Disease**: This dataset provides disease names and concept annotations drawn from the NCBI Disease Corpus, with a focus on biomedical-named entities.
- **BC5CDR**: A dataset consisting of articles with annotations for chemicals, diseases, and their relationships.
- **MITRestaurant**: A collection of annotated online restaurant reviews for entity recognition in the culinary domain.
- **Few-NERD**: This dataset contains a collection of Wikipedia articles and news reports. Due to its large size, we trained the models on a limited portion (20% of the data, which represents 32,941 samples).
- **MultiCoNER**: A large multilingual dataset covering three domains: Wiki sentences, questions, and search queries.

The characteristics of these datasets are provided in Table 4.

10 Experiments

In this section, we describe the methodology used to assess the performance of the chosen algorithms.

10.1 Datasets

Our experiments were carried out on ten datasets from various domains, as shown in Table 4. These datasets differ in size and class count, allowing for various evaluation scenarios.

Table 4: Characteristics of the selected datasets used in our comparative study, with "#" indicating the number of samples in each split.

Corpus	#Train	#Test	#Validation	tags
CoNLL-2003	14,041	3,453	3,250	4
OntoNotes 5.0	59,924	8,262	8,528	18
WNUT2017	2,395	1,287	1,009	6
BioNLP2004	16,619	3,856	1,927	5
FIN	1,018	305	150	4
NCBI Disease	5,433	941	924	1
BC5CDR	5,228	5,865	5,330	2
MITRestaurant	6,900	1,521	760	8
Few-NERD	131,767	37,648	18,824	66
MultiCoNER	16,778	249,980	871	33

The corpora were formatted using the CoNLL-U standard, following the BIO tagging scheme, where (a) each word line includes annotations for individual words, and (b) blank lines denote sentence boundaries. Since each framework requires its own data representation format, we converted the original CoNLL-U format into the appropriate formats for each framework.

For GPT-4, custom prompts were designed for each dataset to highlight the key categories of named entities specific to that dataset. These prompts also included multiple examples to provide context. The box below showcases sample prompts for the BIONLP2004 dataset.

Example of a prompt used for the dataset BIONLP2004

Identify the named entities in the following sentence, categorizing them according to these definitions:

- **"DNA"**:
  - Entities that refer to specific DNA sequences or are related to DNA in a biological context.
  - Examples: BRCA1 gene, CD14 5'-Upstream Sequence, or nonoptimal binding sites.
- **"Protein"**:
  - Entities representing specific proteins or related to proteins in a biological sense.
  - Examples: c-myb, hemoglobin, or E-box-binding repressor.
- **"Cell_type"**:
  - Entities referring to specific types of cells in a biological context.
  - Examples: neuron, T-cell, or hepatocyte.
- **"Cell_line"**:
  - Entities representing specific cell lines used in biological research.
  - Examples: Monocytic U937 cells, Jurkat cell line, or MCF-7.
- **"RNA"**:
  - Entities that are specific RNA sequences or related to RNA in a biological context.
  - Examples: mRNA, siRNA, or c-jun mRNA.

Format the output in JSON with keys corresponding to each entity type. List the identified named entities under each key, grouping multiple entities of the same type into a single list. Ensure a clear separation and precise identification of each entity to avoid ambiguity. Return only the JSON response with no additional explanations.

In our study, we adopted an experimental approach that diverges slightly from the method proposed in the GPT-NER paper Wang et al. [2023b]. While that study demonstrated an effective technique, it relied on distinct prompts for each category of named entities, querying the model individually. This approach, though successful, simplifies the model's task by reducing the need to disambiguate between different entity types. To better simulate real-world conditions, we opted for a less guided strategy that challenges the model's general capability to accurately identify and classify named entities in more complex and varied scenarios. This approach enhances the evaluation by requiring greater context understanding and entity disambiguation, which are essential to the model's overall effectiveness in practical applications.

10.2 Models

Our selection of frameworks was guided by three primary criteria: open-source availability, free access, and the ability to train models on custom datasets. Based on these factors, we included Flair Akbik et al. [2019], Stanford CoreNLP Manning et al. [2014], spaCy Honnibal and Montani [2017], GliNER Zaratiana et al. [2023], OpenAI OpenAI, and

Hugging Face Wolf et al. [2020] in our analysis. For frameworks that support multiple algorithms, we selected several representative models.

For spaCy, we evaluated three models, each illustrating different architectures and capability levels: a small CNN (“`en_core_web_sm`”), a large CNN (“`en_core_web_lg`”), and a Transformer model based on basic RoBERTa Liu et al. [2019a] (“`en_core_web_trf`”).

In the case of Hugging Face, we chose six models: the basic BERT Devlin et al. [2018] architecture in both lowercase and uppercase configurations (“`BERT-base-cased`” and “`BERT-base-uncased`”), its distilled variant Sanh et al. [2019] (“`DistilBERT-base-cased`”), a large RoBERTa-based model (“`FacebookAI/xlm-roberta-large`”), and both small and large versions of DeBERTa He et al. [2020a] (“`Microsoft/DeBERTa-v3-base`” and “`Microsoft/DeBERTa-v3-large`”).

For OpenAI, we used the GPT-4o[5] model, while for GliNER, we selected the large architecture based on DeBERTa (“`urchade/gliner_large`”), both in its pre-trained and fine-tuned versions.

Turning to Flair, we leveraged the standard architecture, which combines an LSTM-CRF network with Flair embeddings. A grid search determined the optimal hyperparameters, testing hidden layer sizes of 64, 128, and 256, and learning rates of 0.05, 0.1, 0.15, and 0.2. The chosen configuration, which achieved the highest F1 score on the validation set, used a learning rate of 0.1 and a hidden layer size of 256.

For the Hugging Face models, we used the TNER library Ushio and Camacho-Collados [2021] to fine-tune each model, with entity decoding managed by CRF Lafferty et al. [2001]. Hyperparameter tuning involved a grid search with learning rates of 10^{-4} and 10^{-5}, both with and without weight decay set to 0.01. Training was carried out over 10 epochs, with an interim evaluation at 5 epochs, a batch size of 16, and gradient accumulation steps set to 2. Additionally, configurations included both CRF and non-CRF models, with a warmup step ratio of 0.1.

For both Stanford CoreNLP and spaCy, we applied their default settings to maintain consistency with standard configurations.

Lastly, for GliNER, we fine-tuned the large model (“`urchade/gliner_large`”) on each dataset using customized hyperparameters. The main model was trained with a learning rate of 5×10^{-6} and weight decay set to 0.01. We used a linear learning rate scheduler with a warm-up ratio of 0.1. Training was carried out over 10 epochs with a batch size of 16, and focal loss parameters were set to alpha = 0.75 and gamma = 2. Step-based evaluations were performed every 300 steps.

10.3 Evaluation and statistical analysis

To evaluate the results, we use an exact evaluation. We select the F1 score as our primary metric because it includes the other two metrics mentioned in Section 8.2, specifically precision and recall.

To assess the differences between the models, we perform the Friedman test [Friedman, 1940]. The Friedman test is a non-parametric statistical test used to detect differences in treatments across multiple attempts. The models are ranked according to their F1 scores, and the null hypothesis of the Friedman test is that the medians of these rankings are equal across the models.

The Friedman test ranks each block together and compares the rankings between columns. The test statistic is given by:

$$Q = \frac{12}{nk(k+1)} \sum_{j=1}^{k} R_j^2 - 3n(k+1) \tag{8}$$

where:

- n is the number of blocks (different sets of data being compared),
- k is the number of models,
- R_j is the sum of the rankings for the j-th treatment.

When the null hypothesis of the Friedman test is rejected (given a chosen significance threshold, e.g., $\alpha = 0.1$), we apply Nemenyi’s method [Nemenyi, 1963] to identify the model pairs that differ significantly. Nemenyi’s post-hoc test compares the average rankings of executions to determine if the differences are statistically significant. The critical difference (CD) is computed as follows, to determine if the differences in average rankings between model pairs are

[5] https://openai.com/index/hello-gpt-4o/

statistically significant:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6n}} \tag{9}$$

where q_α is the critical value from the studentized range distribution, based on the selected significance level α.

11 Results and discussion

Table 5 provides a comparison of the selected NER frameworks based on their macro-averaged F1 scores across ten datasets. The best and second-best performances are highlighted in bold and underlined, respectively. Our experiments reveal key performance trends influenced by dataset size, domain specificity, and the comparative effectiveness of pre-trained versus fine-tuned NER models. In the following, we delve into the details of these results and their implications for model performance across various datasets.

Table 5: Comparison of NER frameworks in terms of Macro-averaged F1 score. Best and second-best scores are respectively in bold and underlined.

		CoNLL-2003	OntoNotes	WNUT2017	FIN	BioNLP2004	NCBI Disease	BC5CDR	MITRestaurant	Few-NERD	MultiCoNER
Frameworks	**Algorithms**	**Macro-averaged F1 score**									
Stanford CoreNLP	CRF	86.96	66.61	13.18	56.70	**79.95**	**90.94**	88.80	74.35	47.07	20.00
Flair	LSTM-CRF	90.35	80.08	38.07	**69.22**	71.64	85.97	**90.51**	79.19	60.28	55.81
GliNER	GliNER-L pre-trained	48.73	28.41	36.77	15.59	45.53	64.16	68.65	35.09	34.63	27.73
	GliNER-L fine-tuned	90.70	**83.15**	**54.38**	65.46	73.13	89.23	90.10	80.42	**64.96**	63.19
spaCy	en_core_web_sm	80.54	68.56	9.72	54.08	66.85	78.71	80.28	74.86	39.27	36.82
	en_core_web_lg	81.66	69.68	9.57	56.52	65.95	78.54	80.25	75.57	40.20	35.36
	en_core_web_trf	90.36	80.79	39.35	53.43	71.34	86.68	87.26	78.48	60.81	**63.82**
Hugging Face	FacebookAI/xlm-RoBERTa-large	90.85	80.63	41.04	46.33	70.35	86.49	87.91	80.45	61.05	58.68
	DistilBERT-base-cased	88.08	77.66	24.11	42.08	67.25	84.41	83.99	77.85	57.66	54.34
	BERT-base-uncased	88.87	77.64	33.78	39.67	69.31	85.89	85.06	79.44	58.29	59.99
	BERT-base-cased	89.37	79.52	33.69	38.43	67.60	86.59	85.21	77.32	59.56	55.88
	Microsoft/DeBERTa-v3-base	**91.36**	80.13	42.96	46.94	71.21	88.03	89.20	79.47	60.44	56.46
	Microsoft/DeBERTa-v3-large	90.98	82.66	44.81	47.28	71.61	86.70	89.20	**81.06**	61.83	61.94
OpenAI	GPT-4o	65.17	59.12	44.73	36.70	45.32	63.46	67.62	51.72	50.87	40.90

Starting with large general-domain datasets such as CoNLL-2003, OntoNotes, Few-NERD, and MultiCoNER, Transformer-based models such as GliNER-L (fine-tuned), RoBERTa, and DeBERTa demonstrate outstanding performance. This can probably be attributed to their high parameter counts, which enable them to capture complex data relationships when trained on extensive datasets. A key insight from this analysis is the notable impact of GliNER-L's comprehensive pre-training and fine-tuning processes. Compared to DeBERTa from Microsoft, GliNER-L exhibits improved performance, especially on domain-specific datasets. This improvement is likely due to GliNER-L's structured approach, which involves sequential stages: pre-training as a language model, followed by NER training, and fine-tuning on domain-specific datasets. These stages allow GliNER-L to capture nuanced entity relationships and effectively adapt to NER tasks.

Shifting the focus to domain-specific datasets, particularly in the biomedical field (BioNLP2004, NCBI Disease, and BC5CDR), a different trend emerges. Traditional models such as CRF and LSTM-CRF remain competitive, often closely matching or even surpassing Transformer-based models. This may be due to the specialized terminology in biomedical texts, which is not well represented in the general datasets typically used for Transformer pre-training. However, fine-tuning on domain-specific data, as demonstrated with GliNER-L, significantly enhances performance by refining embeddings to better capture the contextual nuances of specialized terms. This trend is illustrated in Table 6, which compares the performance of CRF, LSTM-CRF, GliNER-L (fine-tuned) and Microsoft's DeBERTa on key types of entity in biomedical datasets.

In the BC5CDR dataset, which includes both "Chemical" and "Disease" entities, LSTM-CRF achieves the highest F1 score for the "Chemical" entity (94.23), outperforming both GliNER-L and DeBERTa. This suggests that LSTM-CRF may have an edge in identifying chemical entities, likely due to its structured feature handling. For the entity "Disease",

Table 6: Performance comparison of top models on biomedical datasets (BC5CDR, BioNLP2004, NCBI Disease) for key entity types, showing F1, precision, and recall scores for each model

	Model	CRF			LSTM-CRF			GliNER-L			DeBERTa (large)		
		F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall
Dataset	entity_type												
BC5CDR	Chemical	89.83	99.0	82.22	94.23	94.39	94.08	93.63	93.44	93.83	92.57	92.05	93.09
	Disease	87.77	99.41	78.56	86.79	86.13	87.45	86.56	84.56	88.65	85.84	84.18	87.57
BioNLP2004	cell_line	67.78	64.79	71.05	60.83	54.52	68.8	62.97	54.91	73.8	62.25	53.61	74.2
	cell_type	79.36	94.54	68.38	76.3	81.7	71.58	76.81	78.24	75.43	76.12	80.1	72.51
	DNA	81.17	89.32	74.38	73.4	72.23	74.62	74.39	71.23	77.84	73.61	70.51	76.99
	Protein	86.37	93.96	79.92	76.67	71.01	83.32	78.46	71.36	87.13	77.19	70.83	84.8
	RNA	85.05	93.0	78.36	70.97	67.69	74.58	73.0	66.21	81.36	68.86	60.65	79.66
NCBI Disease	Disease	90.94	99.9	83.39	85.97	86.1	85.83	89.23	86.73	91.88	86.65	86.42	86.88

CRF shows strong precision performance (99.41), but GliNER-L achieves the highest recall (88.65), indicating its ability to capture more disease-related entities even though its overall F1 score is slightly lower than LSTM-CRF.

In the BioNLP2004 dataset, which covers a broader range of entity types ("cell_line", "cell_type", "DNA", "Protein", and "RNA"), CRF consistently achieves high scores, especially in precision across entities like "cell_type" (94.54), "DNA" (89.32), "Protein" (93.96), and "RNA" (93.0). For "Protein" and "RNA" entities, GliNER-L shows competitive recall values (87.13 and 81.36, respectively), indicating its strength in capturing complex biomedical terms despite not leading in all F1 scores. LSTM-CRF and DeBERTa display relatively balanced scores across categories, though GliNER-L has a slight edge in recall for multi-word biomedical terms.

In the NCBI Disease dataset, which focuses solely on the "Disease" entity, CRF leads with the highest F1 (90.94) and precision (99.9), underscoring its effectiveness in datasets with specialized terminology. However, GliNER-L achieves the highest recall (91.88), highlighting its ability to identify a broader range of disease-related mentions within the dataset.

Turning to smaller datasets such as FIN and WNUT2017, we observe that LSTM-CRF models tend to outperform Transformer-based models. This difference may stem from the tendency of overparameterized models, such as Transformers, to overfit when trained on limited data. For example, on the FIN dataset, LSTM-CRF achieves the highest F1 score (69.22), significantly outperforming Transformer models, which require careful hyperparameter tuning and well-structured validation sets to reduce overfitting. Notably, GliNER-L fine-tuned remains competitive even on these smaller datasets, consistently ranking among the top three models. This suggests that GliNER-L's structured pre-training and fine-tuning processes enhance its robustness and adaptability across datasets of varying sizes.

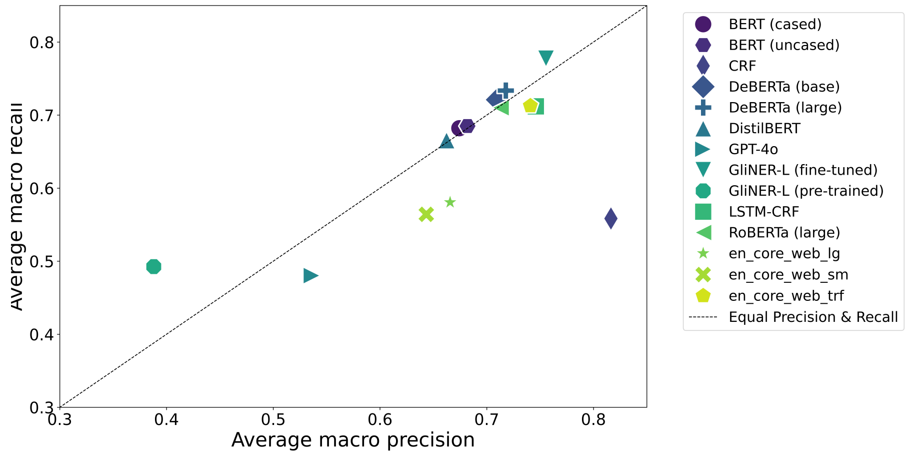

Figure 8: Average precision and recall scores by model. Each point represents a model's mean precision and recall across all datasets, with the dashed line indicating a balance between precision and recall.

Figure 8 represents the average scores of each model in all datasets. Each point shows the mean precision and recall for a model, with the dashed line indicating a perfect balance between the two metrics. Here are some insights:

- **Precision/recall balance:** Models located near the central line, such as DeBERTa variants and LSTM-CRF, exhibit a well-balanced trade-off between precision and recall. This balance makes them particularly suitable for tasks where an equilibrium between these metrics is crucial, ensuring both high precision and coverage in entity detection.
- **Precision-focused models:** Models positioned above the reference line, such as CRF and CNNs (“`en_core_web_sm`” and “`en_core_web_lg`”), demonstrate higher precision than recall, making them ideal for applications where minimizing false positives is a priority.
- **Recall-focused model:** GliNER-L pre-trained model favors recall, demonstrating considerably higher recall than precision, though it still falls significantly behind the other models in overall performance.

Apart from GliNER-L pre-trained, which leans significantly toward recall, the other transformer models generally achieve a balanced approach, without a strong bias toward either precision or recall. GliNER-L pre-trained has allowed it to recognize a broad range of named entities, which boosts its recall by capturing many possible entities. However, this recall comes at the expense of precision, as the model often makes errors. To achieve effective performance, GliNER-L requires fine-tuning on specific datasets. In contrast, the CNN and CRF models appear to favor precision, likely due to their structural design and training objectives. For example, CRF models are optimized to model dependencies between labels in sequence data, enabling them to capture context and dependencies effectively. This makes CRFs more precise in identifying specific, well-defined entities, as they are less prone to false positives. Similarly, CNNs focus on capturing local patterns in the data, which can make them precise in NER tasks.

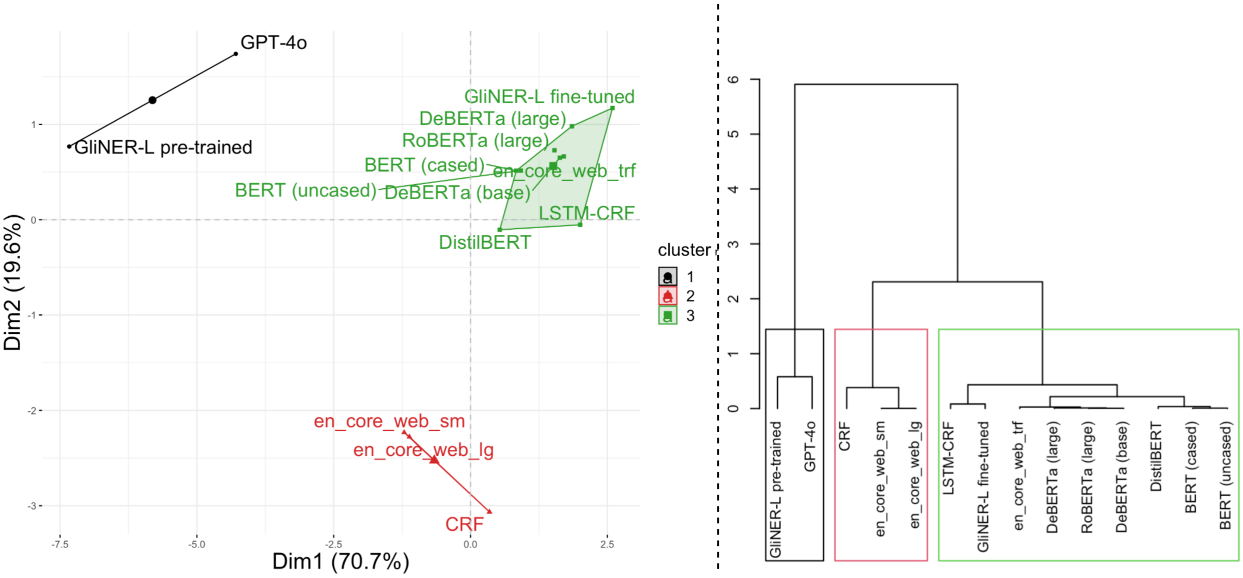

Figure 9: PCA plot (left) and hierarchical clustering dendrogram (right) of NER models based on macro-averaged F1 scores across all datasets.

Furthermore, Figure 9 presents a PCA plot (left) and hierarchical clustering (right), which illustrate performance similarities between frameworks. In the PCA plot, each point represents a model, with colors indicating clusters based on performance similarity. The first two dimensions capture most of the variance, showing that Transformer-based models, such as DeBERTa (base and large) and GliNER-L fine-tuned, cluster closely, suggesting consistent performance across datasets. In contrast, isolated points such as GPT-4o and GliNER-L pre-trained exhibit lower performance, particularly without fine-tuning. Hierarchical clustering (right) reinforces these insights. Traditional models, such as CRF and CNNs (both “`en_core_web_sm`” and “`en_core_web_lg`”), form a distinct cluster, highlighting their competitiveness but limited flexibility compared to Transformers. Fine-tuned Transformers, especially GliNER-L, stand out, underscoring the advantages of domain-specific training.

Statistical analysis supports these findings. A Friedman test applied to all entity types in our datasets reveals statistically significant differences between model medians (p-value = 0.000), confirming the significance at an alpha level of 5%. The results of Nemenyi’s post hoc test (Figure 10) show that certain groups of models have similar performance levels, as indicated by the connecting horizontal lines. For example, models such as GliNER-L fine-tuned and DeBERTa (large) achieve the best rankings, indicating superior performance, and are not significantly different from each other.

This suggests that these models perform similarly on the evaluated datasets. In particular, among the Transformer-based models, the DeBERTa variants appear to be the most effective for NER.

Additionally, another cluster of models, including GliNER-L pre-trained, CNNs (“`en_core_web_sm`” and “`en_core_web_lg`”), and GPT-4o, also exhibit statistically similar performance, although with higher mean ranks, indicating relatively lower performance compared to the top performing models. Toward the right of the figure, models such as CRF, DistilBERT, and BERT exhibit the highest ranks, indicating inferior performance in comparison to the other models. Our experiments have shown that while LLMs perform exceptionally well in many NLP tasks, their general nature often leads to a lack of specialized effectiveness for NER, which can impede precise entity recognition, particularly in noisy or dynamic contexts. Evaluations highlight a performance gap for NER tasks Han et al. [2023], likely due to limited task-specific learning and explicit understanding, which may lead to a “lack of specialty” in NER. Moreover, the large, sometimes undisclosed, number of parameters in LLMs poses challenges for fine-tuning to meet the demands of NER. Consequently, directly addressing NER tasks using only LLMs remains difficult. However, combining LLMs with simpler and fine-tuned NER models offers a promising solution. Approaches such as LinkNER Zhang et al. [2024b] leverage a hybrid model in which the NER component handles common entities, while the LLM, guided by uncertainty estimation, addresses complex or ambiguous cases, enhancing robustness and flexibility in open domain contexts. Similarly, the Super In-Context Learning (SuperICL) approach Xu et al. [2023] integrates smaller specialized plug-in models to provide task-specific predictions and confidence scores, refining LLM outputs for more accurate final predictions. Together, LinkNER and SuperICL demonstrate that combining the broad contextual knowledge of LLMs with the focused precision of smaller models offers a promising path to advance NER, particularly in challenging open-domain settings with unseen entities and noisy data.

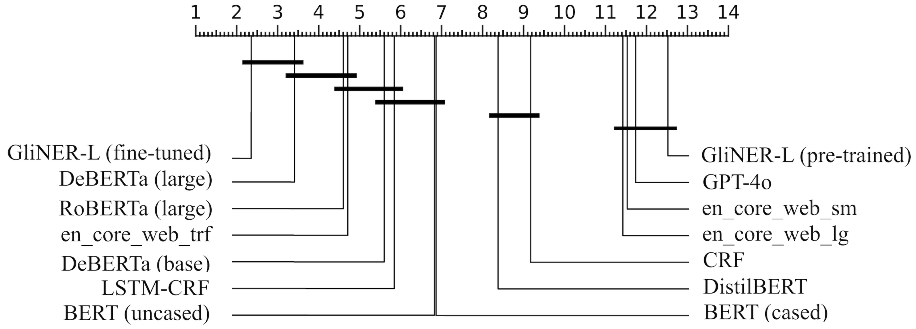

Figure 10: The critical difference (CD) diagram based on F1 score. The plot shows the mean rankings of the different models on 10 datasets. The lower the ranking, the better the performance of a model. A horizontal line indicates no significant difference between the models crossed by the line.

12 Conclusion and perspectives

This article presents a comprehensive survey on recent advances in NER within a classification framework. We focus on recent methods, including LLMs, graph-based approaches, reinforcement learning techniques, and strategies to train models on small datasets. To assess these approaches, we evaluated popular frameworks across datasets with varying characteristics.

Transformer-based architectures, especially on larger datasets, have demonstrated strong performance due to their substantial parameterization and adaptability. However, our analysis indicates that, despite the overall success of Transformers, models like GPT-4o do not consistently achieve top rankings for NER tasks. This may be attributed to the challenges in accurately disambiguating and detecting composite named entities, which are crucial in NER. In contrast, GliNER-L has shown remarkable consistency and robustness across various datasets, making it a reliable choice for a wide range of applications. Moreover, in the case of smaller datasets or specialized domains, such as biomedical texts, traditional approaches such as CRF or LSTM-CRF can outperform Transformers, as these simpler models often handle specific terminology and limited data more effectively.

Looking ahead, the potential of LLMs to enhance NER should not be underestimated. Despite current limitations in handling specialized NER tasks, the rapid advancement of language models presents an exciting opportunity to integrate these technologies into more refined NER systems. Promising directions to advance NER include hybrid approaches that combine LLMs with specialized, fine-tuned NER models (such as LinkNER or SuperICL). By blending the broad contextual understanding of LLMs with the precision of specialized NER models, these strategies hold promise to tackle NER in dynamic, open-domain settings that involve unseen entities and noisy data. Future research should further investigate these hybrid approaches, along with fine-tuning techniques and preprocessing methods designed to enhance LLM adaptability for NER. Such strategies could improve the applicability and accuracy of NER systems across diverse datasets and challenging contexts.

References

Ralph Grishman and Beth M Sundheim. Message understanding conference-6: A brief history. In *COLING*, 1996.

Tahir Mehmood, Ivan Serina, Alberto Lavelli, Luca Putelli, and Alfonso Gerevini. On the use of knowledge transfer techniques for biomedical named entity recognition. *Future Internet*, 15(2):79, 2023.

Sepideh Mesbah, Christoph Lofi, Manuel Valle Torre, Alessandro Bozzon, and Geert-Jan Houben. Tse-ner: An iterative approach for long-tail entity extraction in scientific publications. In *The Semantic Web–ISWC 2018*, pages 127–143, 2018.

Ling Luo, Chih-Hsuan Wei, Po-Ting Lai, Robert Leaman, Qingyu Chen, and Zhiyong Lu. Aioner: all-in-one scheme-based biomedical named entity recognition using deep learning. *Bioinformatics*, 39(5):btad310, 2023.

Bogdan Babych and Anthony Hartley. Improving machine translation quality with automatic named entity recognition. In *EAMT*, 2003.

Diego Mollá, Menno Van Zaanen, and Daniel Smith. Named entity recognition for question answering. In *ALTA*, pages 51–58, 2006.

Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. In *SIGIR*, pages 267–274, 2009.

Lisa F Rau. Extracting company names from text. In *Conference on Artificial Intelligence Application*, pages 29–30, 1991.

Andrei Mikheev, Marc Moens, and Claire Grover. Named entity recognition without gazetteers. In *EACL*, pages 1–8, 1999.

Dimitra Farmakiotou, Vangelis Karkaletsis, John Koutsias, George Sigletos, Constantine D Spyropoulos, and Panagiotis Stamatopoulos. Rule-based named entity recognition for greek financial texts. In *Proceedings of the Workshop on Computational lexicography and Multimedia Dictionaries (COMLEX 2000)*, pages 75–78, 2000.

Hai Leong Chieu and Hwee Tou Ng. Named entity recognition with a maximum entropy approach. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 160–163, 2003.

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

Michal Konkol and Miloslav Konopík. Crf-based czech named entity recognizer and consolidation of czech ner research. In *Text, Speech, and Dialogue: 16th International Conference, TSD 2013, Pilsen, Czech Republic, September 1-5, 2013. Proceedings 16*, pages 153–160. Springer, 2013.

Khaled Shaalan. A survey of arabic named entity recognition and classification. *Computational Linguistics*, 40(2): 469–510, 2014.

Safaa Eltyeb and Naomie Salim. Chemical named entities recognition: a review on approaches and applications. *Journal of cheminformatics*, 6:1–12, 2014.

Ronan Collobert. Deep learning for efficient discriminative parsing. In *AISTATS*, pages 224–232. JMLR Workshop and Conference Proceedings, 2011.

Zhehuan Zhao, Zhihao Yang, Ling Luo, Yin Zhang, Lei Wang, Hongfei Lin, and Jian Wang. Ml-cnn: A novel deep learning based disease named entity recognition architecture. In *IEEE-BIBM)*, pages 794–794, 2016.

Yue Zhang and Jie Yang. Chinese ner using lattice lstm. *arXiv preprint arXiv:1805.02023*, 2018.

Kai Labusch, Preußischer Kulturbesitz, Clemens Neudecker, and David Zellhöfer. Bert for named entity recognition in contemporary and historical german. In *KONVENS*, pages 8–11, 2019.

Yuna Jeong and Eunhui Kim. Scideberta: Learning deberta for science technology documents and fine-tuning information extraction tasks. *IEEE Access*, 10:60805–60813, 2022.

Robert Vacareanu, Enrique Noriega-Atala, Gus Hahn-Powell, Marco A Valenzuela-Escárcega, and Mihai Surdeanu. Active learning design choices for ner with transformers. In *LREC-COLING*, pages 321–334, 2024.

Yuning Shi and Masaomi Kimura. Bert-based models with attention mechanism and lambda layer for biomedical named entity recognition. In *ICMLC*, pages 536–544, 2024.

Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970.

John Lafferty, Andrew McCallum, Fernando Pereira, et al. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Icml*, volume 1, page 3. Williamstown, MA, 2001.

Rodrigo Rafael Villarreal Goulart, Vera Lúcia Strube de Lima, and Clarissa Castellã Xavier. A systematic review of named entity recognition in biomedical texts. *JBCS*, 17:103–116, 2011.

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

Kalyani Pakhale. Comprehensive overview of named entity recognition: Models, domain-specific applications and challenges. *arXiv preprint arXiv:2309.14084*, 2023.

Basra Jehangir, Saravanan Radhakrishnan, and Rahul Agarwal. A survey on named entity recognition—datasets, tools, and methodologies. *Natural Language Processing Journal*, 3:100017, 2023.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70, 2020a.

Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. Named entity recognition and relation extraction: State-of-the-art. *CSUR*, 54(1):1–39, 2021.

Chenquan Dai, Xiaobin Zhuang, and Jiaxin Cai. A survey on deep learning for chinese medical named entity recognition. In *Proceedings of the 2023 9th International Conference on Computing and Artificial Intelligence*, pages 472–476, 2023.

Xiaole Li, Tianyu Wang, Yadan Pang, Jin Han, and Jin Shi. Review of research on named entity recognition. In *Artificial Intelligence and Security*, pages 256–267, 2022.

Yu Wang, Hanghang Tong, Ziye Zhu, and Yun Li. Nested named entity recognition: a survey. *TKDD*, 16(6):1–29, 2022a.

Benyou Wang, Qianqian Xie, Jiahuan Pei, Zhihong Chen, Prayag Tiwari, Zhao Li, and Jie Fu. Pre-trained language models in biomedical domain: A systematic survey. *CSUR*, 56(3):1–52, 2023a.

Maud Ehrmann, Ahmed Hamdi, Elvys Linhares Pontes, Matteo Romanello, and Antoine Doucet. Named entity recognition and classification in historical documents: A survey. *CSUR*, 56(2):1–47, 2023.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*, 2023b.

Dhananjay Ashok and Zachary C Lipton. Promptner: Prompting for named entity recognition. *arXiv preprint arXiv:2305.15444*, 2023.

Yawen Yang, Xuming Hu, Fukun Ma, Aiwei Liu, Lijie Wen, S Yu Philip, et al. Gaussian prior reinforcement learning for nested named entity recognition. In *ICASSP*, pages 1–5, 2023.

Jing Wan, Haoming Li, Lei Hou, and Juaizi Li. Reinforcement learning for named entity recognition from noisy data. In Xiaodan Zhu, Min Zhang, Yu Hong, and Ruifang He, editors, *Natural Language Processing and Chinese Computing*, pages 333–345, Cham, 2020a. Springer International Publishing. ISBN 978-3-030-60450-9.

Ki-Joong Lee, Young-Sook Hwang, Seonho Kim, and Hae-Chang Rim. Biomedical named entity recognition using two-phase model based on svms. *Journal of biomedical informatics*, 37(6):436–447, 2004.

Kangjie Wu, Liqian Xu, Xinxiang Li, Youhua Zhang, Zhenyu Yue, Yujia Gao, and Yiqiong Chen. Named entity recognition of rice genes and phenotypes based on bigru neural networks. *Computational Biology and Chemistry*, 108:107977, 2024.

Beatrice Alex, Barry Haddow, and Claire Grover. Recognising nested named entities in biomedical text. In *Biological, translational, and clinical language processing*, pages 65–72, 2007.

Takashi Shibuya and Eduard Hovy. Nested named entity recognition via second-best sequence learning and decoding. *TACL*, 8:605–620, 2020.

Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang, and Weiming Lu. Locate and label: A two-stage identifier for nested named entity recognition. *arXiv preprint arXiv:2105.06804*, 2021.

Zihan Liu, Feijun Jiang, Yuxiang Hu, Chen Shi, and Pascale Fung. Ner-bert: A pre-trained model for low-resource entity tagging. *arXiv preprint*, 2021.

Jiang Liu. Detection of adverse drug events using electronic health records. 2023.

Abhyuday Jagannatha, Feifan Liu, Weisong Liu, and Hong Yu. Overview of the first natural language processing challenge for extracting medication, indication, and adverse drug events from electronic health record notes (made 1.0). *Drug safety*, 42:99–111, 2019.

Xavier Tannier, Perceval Wajsbürt, Alice Calliger, Basile Dura, Alexandre Mouchet, Martin Hilka, and Romain Bey. Development and validation of a natural language processing algorithm to pseudonymize documents in the context of a clinical data warehouse. *Methods of Information in Medicine*, 2024.

Leigh Weston, Vahe Tshitoyan, John Dagdelen, Olga Kononova, Amalie Trewartha, Kristin A Persson, Gerbrand Ceder, and Anubhav Jain. Named entity recognition and normalization applied to large-scale information extraction from the materials science literature. *Journal of chemical information and modeling*, 59(9):3692–3702, 2019.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134, 2005.

Partha Sarathy Banerjee, Baisakhi Chakraborty, Deepak Tripathi, Hardik Gupta, and Sourabh S Kumar. A information retrieval based on question and answering and ner for unstructured information without using sql. *Wireless Personal Communications*, 108:1909–1931, 2019.

Xiang Cheng, Mitchell Bowden, Bhushan Ramesh Bhange, Priyanka Goyal, Thomas Packer, and Faizan Javed. An end-to-end solution for named entity recognition in ecommerce search. In *AAAI*, volume 35, pages 15098–15106, 2021.

Jim Cowie and Wendy Lehnert. Information extraction. *Communications of the ACM*, 39(1):80–91, 1996.

Pratik Jayarao, Chirag Jain, and Aman Srivastava. Exploring the importance of context and embeddings in neural ner models for task-oriented dialogue systems. *arXiv preprint arXiv:1812.02370*, 2018.

Minhao Cheng, Wei Wei, and Cho-Jui Hsieh. Evaluating and enhancing the robustness of dialogue systems: A case study on a negotiation agent. In *NAACL*, pages 3325–3335, 2019.

Cheoneum Park, Seohyeong Jeong, and Juae Kim. Admit: Improving ner in automotive domain with domain adversarial training and multi-task learning. *Expert Systems with Applications*, 225:120007, 2023.

Mohammad Ebrahim Khademi and Mohammad Fakhredanesh. Persian automatic text summarization based on named entity recognition. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, pages 1–12, 2020.

Jing Liu, Xiaodong He, and Jianfeng Gao. Summarization with named entity recognition and salience detection. In *ACL*, pages 5100–5110, 2022.

Vishal Singh Roha, Naveen Saini, Sriparna Saha, and Jose G Moreno. Moo-cmds+ ner: Named entity recognition-based extractive comment-oriented multi-document summarization. In *European Conference on Information Retrieval*, pages 580–588. Springer, 2023.

Fahim K Sufi, Imran Razzak, and Ibrahim Khalil. Tracking anti-vax social movement using ai-based social media monitoring. *IEEE Transactions on Technology and Society*, 3(4):290–299, 2022.

Yuanze Ji, Bobo Li, Jun Zhou, Fei Li, Chong Teng, and Donghong Ji. Cmner: A chinese multimodal ner dataset based on social media. *arXiv preprint arXiv:2402.13693*, 2024.

Omar Al-Qawasmeh, Mohammad Al-Smadi, and Nisreen Fraihat. Arabic named entity disambiguation using linked open data. In *ICICS*, pages 333–338, 2016.

Razvan Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, pages 9–16, 2006.

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, Tim Finin, et al. Entity disambiguation for knowledge base population. In *ACL*, 2010.

Yuanzhe Zhang, Kang Liu, Shizhu He, Guoliang Ji, Zhanyi Liu, Hua Wu, and Jun Zhao. Question answering over knowledge base with neural attention combining global knowledge information. *arXiv preprint arXiv:1606.00979*, 2016.

Zhen Li, Dan Qu, Chaojie Xie, Wenlin Zhang, and Yanxia Li. Language model pre-training method in machine translation based on named entity recognition. *IJAIT*, 29(07n08):2040021, 2020b.

Emna Hkiri, Souheyl Mallat, and Mounir Zrigui. Arabic-english text translation leveraging hybrid ner. In *Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation*, pages 124–131, 2017.

Casimir George Borkowski. *A system for automatic recognition of names of persons in newspaper texts*. IBM Watson Research Center, 1966.

Denis Maurel, Nathalie Friburger, Jean-Yves Antoine, Iris Eshkol, and Damien Nouvel. Cascades de transducteurs autour de la reconnaissance des entités nommées. *Revue TAL*, 52(1):69–96, 2011.

Mohammad Hjouj Btoush, Abdulsalam Alarabeyyat, and Isa Olab. Rule based approach for arabic part of speech tagging and name entity recognition. *IJACSA*, 7(6), 2016.

Tome Eftimov, Barbara Koroušić Seljak, and Peter Korošec. A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations. *PloS one*, 12(6):e0179488, 2017.

Daniel Hanisch, Katrin Fundel, Heinz-Theodor Mevissen, Ralf Zimmer, and Juliane Fluck. Prominer: rule-based protein and gene entity recognition. *BMC bioinformatics*, 6:1–9, 2005.

Satoshi Sekine and Chikashi Nobata. Definition, dictionaries and tagger for extended named entity hierarchy. In *LREC*, pages 1977–1980. Lisbon, Portugal, 2004.

Alexandra Pomares Quimbaya, Alejandro Sierra Múnera, Rafael Andrés González Rivera, Julián Camilo Daza Rodríguez, Oscar Mauricio Muñoz Velandia, Angel Alberto Garcia Peña, and Cyril Labbé. Named entity recognition over electronic health records through a combined dictionary-based approach. *Procedia Computer Science*, 100: 55–61, 2016.

Shaodian Zhang and Noémie Elhadad. Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. *JBI*, 46(6):1088–1098, 2013.

Madhusudan Ghosh, Debasis Ganguly, Partha Basuchowdhuri, and Sudip Kumar Naskar. Extracting methodology components from ai research papers: A data-driven factored sequence labeling approach. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3897–3901, 2023.

Lang-Tao Wu, Jia-Rui Lin, Shuo Leng, Jiu-Lin Li, and Zhen-Zhong Hu. Rule-based information extraction for mechanical-electrical-plumbing-specific semantic web. *Automation in Construction*, 135:104108, 2022a.

Davlatyor B Mengliev, Vladimir B Barakhnin, Mukhammadjon Atakhanov, Bahodir B Ibragimov, Mukhriddin Eshkulov, and Bobur Saidov. Developing rule-based and gazetteer lists for named entity recognition in uzbek language: geographical names. In *APEIE*, pages 1500–1504, 2023.

Yusuke Shinyama and Satoshi Sekine. Named entity discovery using comparable news articles. In *COLING*, pages 848–853, 2004.

Ludovic Bonnefoy, Patrice Bellot, and Michel Benoit. An unsupervised measure of the degree of belonging of an entity to a type. TALN, 2011.

David Nadeau, Peter D Turney, and Stan Matwin. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *CAIAC*, pages 266–277. Springer, 2006.

Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–110, 1999. URL `https://aclanthology.org/W99-0613`.

Zornitsa Kozareva, Boyan Bonev, and Andres Montoyo. Self-training and co-training applied to spanish named entity recognition. In *Mexican International conference on Artificial Intelligence*, pages 770–779. Springer, 2005.

Shang Gao, Olivera Kotevska, Alexandre Sorokine, and J Blair Christian. A pre-training and self-training approach for biomedical named entity recognition. *PloS one*, 16(2):e0246310, 2021.

GuoDong Zhou and Jian Su. Named entity recognition using an hmm-based chunk tagger. In *ACL*, pages 473–480, 2002.

Sudha Morwal, Nusrat Jahan, and Deepti Chopra. Named entity recognition using hidden markov model (hmm). *IJNLC*, 1, 2012.

Daniel M Bikel, Richard Schwartz, and Ralph M Weischedel. An algorithm that learns what's in a name. *Machine learning*, 34:211–231, 1999.

Shaojun Zhao. Named entity recognition in biomedical texts using an hmm model. In *NLPBA/BioNLP*, pages 87–90, 2004.

Adam Berger, Stephen A Della Pietra, and Vincent J Della Pietra. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71, 1996.

Andrew Eliot Borthwick. *A maximum entropy approach to named entity recognition*. New York University, 1999.

James R Curran and Stephen Clark. Language independent ner using a maximum entropy tagger. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 164–167, 2003.

Yi-Feng Lin, Tzong-Han Tsai, Wen-Chi Chou, Kuen-Pin Wu, et al. A maximum entropy approach to biomedical named entity recognition. In *International Conference on Data Mining in Bioinformatics*, pages 56–61, 2004.

Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. Nyu: Description of the mene named entity system as used in muc-7. In *MUC-7*, 1998.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

Takaki Makino, Yoshihiro Ohta, Jun'ichi Tsujii, et al. Tuning support vector machines for biomedical named entity recognition. In *ACL*, pages 1–8, 2002.

Asif Ekbal and Sivaji Bandyopadhyay. Named entity recognition using support vector machine: A language independent approach. *IJECE*, 4(3):589–604, 2010.

Zhenfei Ju, Jian Wang, and Fei Zhu. Named entity recognition from biomedical text using svm. In *2011 5th international conference on bioinformatics and biomedical engineering*, pages 1–4. IEEE, 2011.

Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *HLT-NAACL*, page 188–191, 2003.

Burr Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *NLP-BA/BioNLP*, pages 107–110, 2004.

Kishorjit Nongmeikapam, Tontang Shangkhunem, Ngariyanbam Mayekleima Chanu, Laisuhram Newton Singh, Bishworjit Salam, and Sivaji Bandyopadhyay. Crf based name entity recognition (ner) in manipuri: A highly agglutinative indian language. In *2011 2nd NCETACS*, pages 1–6. IEEE, 2011.

Praneeth M Shishtla, Karthik Gali, Prasad Pingali, and Vasudeva Varma. Experiments in telugu ner: A conditional random field approach. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*, 2008.

Hiroyasu Yamada, Taku Kudo, and Yuji Matsumoto. Japanese named entity extraction using support vector machine. 43(1):44–53, 2002.

Taku Kudo and Yuji Matsumoto. Chunking with support vector machines. In *Second meeting of the North American chapter of the Association for Computational Linguistics*, 2001.

Masayuki Asahara and Yuji Matsumoto. Japanese named entity extraction with redundant morphological analysis. In *NAACL*, pages 8–15, 2003.

Koichi Takeuchi and Nigel Collier. Use of support vector machines in extended named entity recognition. In *COLING*, 2002.

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

Rohini K Srihari. A hybrid approach for named entity and sub-type tagging. In *Sixth applied natural language processing conference*, pages 247–254, 2000.

Shilpi Srivastava, Mukund Sanglikar, and DC Kothari. Named entity recognition system for hindi language: a hybrid approach. *IJCL*, 2(1):10–23, 2011.

Raymond Chiong and Wang Wei. Named entity recognition using hybrid machine learning approach. In *2006 5th IEEE International Conference on Cognitive Informatics*, volume 1, pages 578–583, 2006.

Kanwalpreet Singh Bajwa and Amardeep Kaur. Hybrid approach for named entity recognition. *IJCA*, 118(1), 2015.

Khaled Shaalan and Mai Oudah. A hybrid approach to arabic named entity recognition. *Journal of Information Science*, 40(1):67–87, 2014.

J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986.

David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 20(2):215–232, 1958.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. jmlr, vol. 3, no, 2003.

Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer, 2003.

Tomas Mikolov. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 3781, 2013.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint*, 2016.

Alec Radford. Improving language understanding by generative pre-training. 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Seungwook Lee and Youngjoong Ko. Named-entity recognition using automatic construction of training data from social media messaging apps. *IEEE Access*, 8:222724–222732, 2020.

Wahiba Ben Abdessalem Karaa. Named entity recognition using web document corpus. *arXiv preprint arXiv:1102.5728*, 2011.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *NeurIPS*, 26, 2013.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537, 2011.

Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *ACL*, pages 1064–1074, 2016.

Guillaume Lample. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

Sławomir Dadas. Combining neural and knowledge-based approaches to named entity recognition in polish. In *ICAISC*, pages 39–50. Springer, 2019.

Arjun Das, Debasis Ganguly, and Utpal Garain. Named entity recognition with word embeddings and wikipedia categories for a low-resource language. *TALLIP*, 16(3):1–19, 2017.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL*, pages 2227–2237, 2018.

S Hochreiter. Long short-term memory. *Neural Computation MIT-Press*, 1997.

Alex Graves and Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012.

Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. In *ACL*, pages 1756–1765, 2017.

Yi Zhou, Xiao-Qing Zheng, and Xuan-Jing Huang. Chinese named entity recognition augmented with lexicon memory. *JCST*, 38(5), 2023a.

Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

Tao Gui, Ruotian Ma, Qi Zhang, Lujun Zhao, Yu-Gang Jiang, and Xuanjing Huang. Cnn-based chinese ner with lexicon rethinking. In *ijcai*, volume 2019, 2019a.

Xin Li, Zequn Jie, Jiashi Feng, Changsong Liu, and Shuicheng Yan. Learning with rethinking: Recurrently improving convolutional neural networks through feedback. *Pattern Recognition*, 79:183–194, 2018.

Qile Zhu, Xiaolin Li, Ana Conesa, and Cécile Pereira. Gram-cnn: a deep learning approach with local context for named entity recognition in biomedical text. *Bioinformatics*, 34(9):1547–1554, 2018.

Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Raghavendra Chalapathy, Ehsan Zare Borzeshi, and Massimo Piccardi. An investigation of recurrent neural architectures for drug name recognition. *arXiv preprint arXiv:1609.07585*, 2016.

Wenming Huang, Dengrui Hu, Zhenrong Deng, and Jianyun Nie. Named entity recognition for chinese judgment documents based on bilstm and crf. *EURASIP Journal on Image and Video Processing*, 2020:1–14, 2020a.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. *NeurIPS*, 28, 2015.

Ling Luo, Zhihao Yang, Pei Yang, Yin Zhang, Lei Wang, Hongfei Lin, and Jian Wang. An attention-based bilstm-crf approach to document-level chemical named entity recognition. *Bioinformatics*, 34(8):1381–1388, 2018.

Ying Lin, Liyuan Liu, Heng Ji, Dong Yu, and Jiawei Han. Reliability-aware dynamic feature composition for name tagging. In *ACL*, pages 165–174, 2019.

Patrick Knobelreiter, Christian Reinbacher, Alexander Shekhovtsov, and Thomas Pock. End-to-end training of hybrid cnn-crf models for stereo. In *CVPR*, pages 2339–2348, 2017.

Naiqin Feng, Xiuqin Geng, and Lijuan Qin. Study on mri medical image segmentation technology based on cnn-crf model. *IEEE Access*, 8:60505–60514, 2020.

Ignazio Gallo, Elisabetta Binaghi, Moreno Carullo, and Nicola Lamberti. Named entity recognition by neural sliding window. In *IAPR*, pages 567–573. IEEE, 2008.

Feifei Zhai, Saloni Potdar, Bing Xiang, and Bowen Zhou. Neural models for sequence chunking. In *AAAI*, volume 31, 2017.

Jing Li, Deheng Ye, and Shuo Shang. Adversarial transfer for named entity boundary detection with pointer networks. In *IJCAI*, pages 5053–5059, 2019.

Stavroula Skylaki, Ali Oskooei, Omar Bari, Nadja Herger, and Zac Kriegman. Named entity recognition in the legal domain using a pointer generator network. *arXiv preprint arXiv:2012.09936*, 2020.

Hao Fei, Donghong Ji, Bobo Li, Yijiang Liu, Yafeng Ren, and Fei Li. Rethinking boundaries: End-to-end recognition of discontinuous mentions with pointer networks. In *AAAI*, volume 35, pages 12785–12793, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

Martin Riedl and Sebastian Padó. A named entity recognition shootout for german. In *ACL*, pages 120–125, 2018.

Stefan Schweter and Johannes Baiter. Towards robust named entity recognition for historic German. In *Proceedings of the 4th Workshop on Representation Learning for NLP*, pages 96–103. ACL, 2019.

Dina Oralbekova, Orken Mamyrbayev, Sholpan Zhumagulova, and Nurdaulet Zhumazhan. A comparative analysis of lstm and bert models for named entity recognition in kazakh language: A multi-classification approach. In *MSBC*, pages 116–128, 2024.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, et al. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint*, 2019a.

Macarious Abadeer. Assessment of distilbert performance on named entity recognition task for the detection of protected health information and medical concepts. In *Proceedings of the 3rd clinical natural language processing workshop*, pages 158–167, 2020.

Shreyansh Mehta, Mansi Radke, and Sagar Sunkle. Named entity recognition using knowledge graph embeddings and distilbert. In *NLPIR*, pages 146–150, 2021.

Ming-Hsiang Su, Chin-Wei Lee, Chi-Lun Hsu, and Ruei-Cyuan Su. Roberta-based traditional chinese medicine named entity recognition model. In *Proceedings of the 34th Conference on Computational Linguistics and Speech Processing (ROCLING 2022)*, pages 61–66, 2022.

Antonia Höfer and Mina Mottahedin. Minanto at semeval-2023 task 2: Fine-tuning xlm-roberta for named entity recognition on english data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1127–1130, 2023.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020a.

Zihan Wang, Ziqi Zhao, Zhumin Chen, Pengjie Ren, Maarten de Rijke, and Zhaochun Ren. Generalizing few-shot named entity recognizers to unseen domains with type-related features. *arXiv preprint arXiv:2310.09846*, 2023c.

Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, and Debasis Ganguly. Identification of tasks, datasets, evaluation metrics, and numeric scores for scientific leaderboards construction. *arXiv preprint arXiv:1906.09317*, 2019.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. Gliner: Generalist model for named entity recognition using bidirectional transformer. *arXiv preprint arXiv:2311.08526*, 2023.

Arjun Choudhry, Pankaj Gupta, Inder Khatri, Aaryan Gupta, Maxime Nicol, Marie-Jean Meurs, and Dinesh Kumar Vishwakarma. Transformer-based named entity recognition for french using adversarial adaptation to similar domain corpora. *arXiv preprint arXiv:2212.03692*, 2022.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. Luke: Deep contextualized entity representations with entity-aware self-attention. *arXiv preprint arXiv:2010.01057*, 2020.

Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. Portuguese named entity recognition using bert-crf. *arXiv preprint arXiv:1909.10649*, 2019.

Tianyu Wan, Wenhui Wang, and Hui Zhou. Research on information extraction of municipal solid waste crisis using bert-lstm-crf. In *NLPIR*, pages 205–209, 2020b.

Bingjun He and Jianfeng Chen. Named entity recognition method in network security domain based on bert-bilstm-crf. In *2021 IEEE 21st International Conference on Communication Technology (ICCT)*, pages 508–512. IEEE, 2021.

YuXuan Chen, Jianwei Ding, Dashuang Li, and Zhouguo Chen. Joint bert model based cybersecurity named entity recognition. In *Proceedings of the 2021 4th International Conference on Software Engineering and Information Management*, pages 236–242, 2021a.

Zhenjin Dai, Xutao Wang, Pin Ni, Yuming Li, Gangmin Li, and Xuming Bai. Named entity recognition using bert bilstm crf for chinese electronic health records. In *2019 12th international congress on image and signal processing, biomedical engineering and informatics (cisp-bmei)*, pages 1–5. IEEE, 2019.

Yuan Chang, Lei Kong, Kejia Jia, and Qinglei Meng. Chinese named entity recognition method based on bert. In *2021 IEEE international conference on data science and computer application (ICDSCA)*, pages 294–299. IEEE, 2021.

Lei Xu, Shuang Li, Yuchen Wang, and Lizhen Xu. Named entity recognition of bert-bilstm-crf combined with self-attention. In *Web Information Systems and Applications*, pages 556–564, 2021.

Lung-Hao Lee, Chien-Huan Lu, and Tzu-Mi Lin. Ncuee-nlp at semeval-2022 task 11: Chinese named entity recognition using the bert-bilstm-crf model. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1597–1602, 2022.

Xiaojun Wu, Tianqi Zhang, Sheng Yuan, and Yuanfeng Yan. One improved model of named entity recognition by combining bert and bilstm-cnn for domain of chinese railway construction. In *2022 7th international conference on intelligent computing and signal processing (ICSP)*, pages 728–732, 2022b.

Peng Chen, Meng Zhang, Xiaosheng Yu, and Songpu Li. Named entity recognition of chinese electronic medical records based on a hybrid neural network and medical mc-bert. *BMC Medical Informatics and Decision Making*, 22 (1):315, 2022.

Norah Alsaaran and Maha Alrabiah. Arabic named entity recognition: A bert-bgru approach. *Comput. Mater. Contin*, 68(1):471–485, 2021.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 33:1877–1901, 2020.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*, 2022.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(140):1–67, 2020.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Xiangji Huang. A systematic study and comprehensive evaluation of chatgpt on benchmark datasets. *arXiv preprint arXiv:2305.18486*, 2023.

Yongil Kim, Yerin Hwang, Joongbo Shin, Hyunkyung Bae, and Kyomin Jung. Injecting comparison skills in task-oriented dialogue systems for database search results disambiguation. In *ACL*, pages 4047–4065, 2023.

Francesco De Toni, Christopher Akiki, Javier De La Rosa, Clémentine Fourrier, Enrique Manjavacas, Stefan Schweter, and Daniel Van Strien. Entities, dates, and languages: Zero-shot on historical texts with t0. *arXiv preprint arXiv:2204.05211*, 2022.

Niklas Kammer, Florian Borchert, Silvia Winkler, Gerard De Melo, and Matthieu-P Schapranow. Resolving elliptical compounds in german medical text. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 292–305, 2023.

Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuanjing Huang, and Xipeng Qiu. Codeie: Large code generation models are better few-shot information extractors. *arXiv preprint arXiv:2305.05711*, 2023.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. Universalner: Targeted distillation from large language models for open named entity recognition. *arXiv preprint arXiv:2308.03279*, 2023b.

Tingyu Xie, Qi Li, Yan Zhang, Zuozhu Liu, and Hongwei Wang. Self-improving for zero-shot named entity recognition with large language models. *arXiv preprint arXiv:2311.08921*, 2023.

Xingyu Zhu, Feifei Dai, Xiaoyan Gu, Bo Li, Meiou Zhang, and Weiping Wang. Gl-ner: Generation-aware large language models for few-shot named entity recognition. In *ICANN*, pages 433–448, 2024.

Xinghua Zhang, Gaode Chen, Shiyao Cui, Jiawei Sheng, Tingwen Liu, and Hongbo Xu. Exogenous and endogenous data augmentation for low-resource complex named entity recognition. In *SIGIR*, pages 630–640, 2024a.

Abir Chebbi, Guido Kniesel, Nabil Abdennadher, and Giovanna Dimarzo. Enhancing named entity recognition for agricultural commodity monitoring with large language models. In *Proceedings of the 4th Workshop on Machine Learning and Systems*, pages 208–213, 2024.

Tristan Luiggi, Tanguy Herserant, Thong Tran, Laure Soulier, and Vincent Guigue. Calm: Context augmentation with large language model for named entity recognition. In *International Conference on Theory and Practice of Digital Libraries*, pages 273–291, 2024.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. Tabllm: Few-shot classification of tabular data with large language models. In *AISTATS*, pages 5549–5581. PMLR, 2023.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. Leveraging large language models for multiple choice question answering. *arXiv preprint arXiv:2210.12353*, 2022.

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. How good are gpt models at machine translation? a comprehensive evaluation. *arXiv preprint arXiv:2302.09210*, 2023.

Yan Hu, Iqra Ameer, Xu Zuo, Xueqing Peng, Yujia Zhou, Zehan Li, Yiming Li, Jianfu Li, Xiaoqian Jiang, and Hua Xu. Zero-shot clinical entity recognition using chatgpt. *arXiv preprint arXiv:2303.16416*, 2023.

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*, 2019.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021b.

Jana Straková, Milan Straka, and Jan Hajič. Neural architectures for nested ner through linearization. *arXiv preprint arXiv:1908.06926*, 2019.

Shi Peng, Yong Zhang, Yuanfang Yu, Haoyang Zuo, and Kai Zhang. Named entity recognition based on reinforcement learning and adversarial training. In *Knowledge Science, Engineering and Management*, pages 191–202, 2021.

Yaosheng Yang, Wenliang Chen, Zhenghua Li, Zhengqiu He, and Min Zhang. Distantly supervised ner with partial annotation learning and reinforcement learning. In *ACL*, pages 2159–2169, 2018.

Yi Chen and Liang He. Skd-ner: Continual named entity recognition via span-based knowledge distillation with reinforcement learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6689–6700, 2023.

Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. In *EMNLP*, pages 1506–1515, 2017.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint*, 2016.

Alberto Cetoli, Stefano Bragaglia, Andrew O'Harney, and Marc Sloan. Graph convolutional networks for named entity recognition. In *TLT*, pages 37–45, 2017.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Sanh, et al. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23, 2013.

Xiaojing Liu, Feiyu Gao, Qiong Zhang, and Huasha Zhao. Graph convolution for multimodal information extraction from visually rich documents. *NAACL*, pages 32–39, 2019b.

Ismail Harrando and Raphaël Troncy. Named entity recognition as graph classification. In *European Semantic Web Conference*, pages 103–108, 2021.

Yuke Wang, Ling Lu, Yang Wu, and Yinong Chen. Polymorphic graph attention network for chinese ner. *Expert Systems with Applications*, 203:117467, 2022b.

Tao Gui, Yicheng Zou, Qi Zhang, Minlong Peng, Jinlan Fu, Zhongyu Wei, and Xuan-Jing Huang. A lexicon-based graph neural network for chinese ner. In *(EMNLP-IJCNLP)*, pages 1040–1050, 2019b.

Yingqi Zhang, Wenjun Ma, and Yuncheng Jiang. Mgcn: A novel multi-graph collaborative network for chinese ner. In Wei Lu, Shujian Huang, Yu Hong, and Xiabing Zhou, editors, *Natural Language Processing and Chinese Computing*, pages 618–630, Cham, 2022. Springer International Publishing. ISBN 978-3-031-17120-8.

Fei Zhao, Chunhui Li, Zhen Wu, Shangyu Xing, and Xinyu Dai. Learning from different text-image pairs: a relation-enhanced graph convolutional network for multimodal ner. In *ACM MULTIMEDIA*, pages 3983–3992, 2022.

Yongquan He, Zihan Wang, Peng Zhang, Zhaopeng Tu, and Zhaochun Ren. Vn network: Embedding newly emerging entities with virtual neighbors. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 505–514, 2020b.

Nasser Alshammari and Saad Alanazi. The impact of using different annotation schemes on named entity recognition. *Egyptian Informatics Journal*, 22(3):295–302, 2021.

Maojian Chen, Xiong Luo, Hailun Shen, Ziyang Huang, and Qiaojuan Peng. A novel named entity recognition scheme for steel e-commerce platforms using a lite bert. *CMES*, 129(1), 2021c.

Manali Shaha and Meenakshi Pawar. Transfer learning for image classification. In *ICECA*, pages 656–660. IEEE, 2018.

Dong Wang and Thomas Fang Zheng. Transfer learning for speech and language processing. In *APSIPA*, pages 1225–1237. IEEE, 2015.

Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Transfer learning for time series classification. In *Big Data*, pages 1367–1376, 2018.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. Bertweet: A pre-trained language model for english tweets. *arXiv preprint arXiv:2005.10200*, 2020.

Ji Young Lee, Franck Dernoncourt, and Peter Szolovits. Transfer learning for named-entity recognition with neural networks. In *LREC 2018*, 2018.

Sumam Francis, Jordy Van Landeghem, and Marie-Francine Moens. Transfer learning for named entity recognition in financial and biomedical documents. *Information*, 10(8):248, 2019.

Hermenegildo Fabregat, Andres Duque, Juan Martinez-Romo, and Lourdes Araujo. Negation-based transfer learning for improving biomedical named entity recognition and relation extraction. *Journal of Biomedical Informatics*, page 104279, 2023.

Xiang Dai and Heike Adel. An analysis of simple data augmentation for named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3861–3867, 2020.

Ranto Sawai, Incheon Paik, and Ayato Kuwana. Sentence augmentation for language translation using gpt-2. *Electronics*, 10(24):3082, 2021.

Huu-Thanh Duong and Tram-Anh Nguyen-Thi. A review: preprocessing techniques and data augmentation for sentiment analysis. *Computational Social Networks*, 8(1):1–16, 2021.

Saket Sharma, Aviral Joshi, Namrata Mukhija, Yiyun Zhao, et al. Systematic review of effect of data augmentation using paraphrasing on named entity recognition. In *NeurIPS*.

Abdenacer Keraghel, Khalid Benabdeslem, and Bruno Canitia. Data augmentation process to improve deep learning-based ner task in the automotive industry field. In *IJCNN*, pages 1–8, 2020.

Akbar Karimi, Leonardo Rossi, and Andrea Prati. Aeda: An easier data augmentation technique for text classification. In *EMNLP 2021*, pages 2748–2754, 2021.

Jiaao Chen, Zhenghui Wang, Ran Tian, Zichao Yang, and Diyi Yang. Local additivity based data augmentation for semi-supervised ner. *arXiv preprint arXiv:2010.01677*, 2020.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyeong Park. Gpt3mix: Leveraging large-scale language models for text augmentation. *arXiv preprint arXiv:2104.08826*, 2021.

Burr Settles. Active learning literature survey. 2009.

Aditya Siddhant and Zachary C Lipton. Deep bayesian active learning for natural language processing: Results of a large-scale empirical study. In *EMNLP*, pages 2904–2909, 2018.

Yanyao Shen, Hyokun Yun, Zachary C Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep active learning for named entity recognition. In *SIGREP*, pages 252–256, 2017.

Chengxi Yan, Xuemei Tang, Hao Yang, and Jun Wang. A deep active learning-based and crowdsourcing-assisted solution for named entity recognition in chinese historical corpora. *Aslib Journal of Information Management*, 75(3): 455–480, 2023.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *CSUR*, 53(3):1–34, 2020.

Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. Few-shot classification in named entity recognition task. In *ACM/SIGAPP*, pages 993–1000, 2019.

Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In *ACL*, pages 1381–1393, 2020.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *NeurIPS*, 30, 2017.

Yi Yang and Arzoo Katiyar. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *EMNLP*, pages 6365–6375, 2020.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. Template-based named entity recognition using bart. In *ACL-IJCNLP*, pages 1835–1845, 2021.

Maximilian Hofer, Andrey Kormilitzin, Paul Goldberg, and Alejo Nevado-Holgado. Few-shot learning for named entity recognition in medical text. *arXiv preprint*, 2018.

Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, et al. Few-shot named entity recognition: A comprehensive study. *arXiv preprint*, 2020b.

Kai He, Rui Mao, Yucheng Huang, Tieliang Gong, Chen Li, and Erik Cambria. Template-free prompting for few-shot named entity recognition via semantic-enhanced contrastive learning. *IEEE transactions on neural networks and learning systems*, 2023.

Zhiwei Yang, Jing Ma, Kang Yang, Huiru Lin, Hechang Chen, Ruichao Yang, and Yi Chang. Cotea: Collaborative teaching for low-resource named entity recognition with a divide-and-conquer strategy. *Information Processing & Management*, 61(3):103657, 2024.

Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. Zero-data learning of new tasks. In *AAAI*, volume 1, page 3, 2008.

Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):453–465, 2013.

Zhengming Ding, Ming Shao, and Yun Fu. Low-rank embedded ensemble semantic dictionary for zero-shot learning. In *CVPR*, pages 2050–2058, 2017.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. In *EMNLP*, pages 6397–6407, 2020.

Rasha Obeidat, Xiaoli Fern, Hamed Shahbazi, and Prasad Tadepalli. Description-based zero-shot fine-grained entity typing. In *NAACL*, pages 807–814, 2019.

Rami Aly, Andreas Vlachos, and Ryan McDonald. Leveraging type descriptions for zero-shot named entity recognition and classification. In *ACL*, pages 1516–1528, 2021.

Nguyen Van Hoang, Soeren Hougaard Mulvad, Dexter Neo Yuan Rong, and Yang Yue. Zero-shot learning in named-entity recognition with external knowledge. *arXiv preprint*, 2021.

Jian Yang, Shaohan Huang, Shuming Ma, Yuwei Yin, Li Dong, Dongdong Zhang, Hongcheng Guo, Zhoujun Li, and Furu Wei. Crop: Zero-shot cross-lingual named entity recognition with multilingual labeled sequence translation. In *EMNLP 2022*, pages 486–496, 2022.

Carlos-Emiliano González-Gallardo, Emanuela Boros, Nancy Girdhar, Ahmed Hamdi, Jose G Moreno, and Antoine Doucet. Yes but.. can chatgpt identify entities in historical documents? In *2023 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 184–189, 2023.

OpenAI. Openai: Ai tools and models. `https://www.openai.com`, 2024. Accessed: 2024-01-04.

Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. 2017.

Steven Bird. Nltk: the natural language toolkit. In *COLING/ACL*, pages 69–72, 2006.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL*, pages 55–60, 2014.

Jason Baldridge. The opennlp project. *URL: http://opennlp.apache.org/index.html,(accessed 1 February 2024)*, 1, 2005.

Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. Polyglot-ner: Massive multilingual named entity recognition. In *SIAM, SDM*, pages 586–594. SIAM, 2015.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL*, pages 54–59, 2019.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, et al. Transformers: State-of-the-art natural language processing. In *EMNLP*, pages 38–45, 2020.

Hamish Cunningham. Gate: A framework and graphical development environment for robust nlp tools and applications. In *ACL*, pages 168–175, 2002.

Asahi Ushio and Jose Camacho-Collados. T-NER: An all-round python library for transformer-based named entity recognition. In *EACL*, pages 53–62. ACL, 2021. URL `https://www.aclweb.org/anthology/2021.eacl-demos.7`.

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *NAACL 2003*, pages 142–147, 2003. URL `https://aclanthology.org/W03-0419`.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon, 2004.

Jinlan Fu, Pengfei Liu, and Graham Neubig. Interpretable multi-dataset evaluation for named entity recognition. In *EMNLP*, pages 6058–6069, November 2020.

Julio Cesar Salinas Alvarado, Karin Verspoor, and Timothy Baldwin. Domain adaption of named entity recognition to support credit risk assessment. In *ALTA*, pages 84–90, 2015.

Milton Friedman. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92, 1940.

Peter Bjorn Nemenyi. *Distribution-free multiple comparisons.* Princeton University, 1963.

Ridong Han, Tao Peng, Chaohao Yang, Benyou Wang, Lu Liu, and Xiang Wan. Is information extraction solved by chatgpt? an analysis of performance, evaluation criteria, robustness and errors. *arXiv preprint arXiv:2305.14450*, 2023.

Zhen Zhang, Yuhua Zhao, Hang Gao, and Mengting Hu. Linkner: Linking local named entity recognition models to large language models using uncertainty. In *WWW*, pages 4047–4058, 2024b.

Canwen Xu, Yichong Xu, Shuohang Wang, Yang Liu, Chenguang Zhu, and Julian McAuley. Small models are valuable plug-ins for large language models. *arXiv preprint arXiv:2305.08848*, 2023.