



HAL
open science

A SURVEY ON RECENT ADVANCES IN NAMED ENTITY RECOGNITION

Imed Keraghel, Stanislas Morbieu, Mohamed Nadif

► **To cite this version:**

Imed Keraghel, Stanislas Morbieu, Mohamed Nadif. A SURVEY ON RECENT ADVANCES IN NAMED ENTITY RECOGNITION. 2024. hal-04488194v1

HAL Id: hal-04488194

<https://hal.science/hal-04488194v1>

Preprint submitted on 4 Mar 2024 (v1), last revised 19 Dec 2024 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A SURVEY ON RECENT ADVANCES IN NAMED ENTITY RECOGNITION

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, but we also look at graph- and transformer-based methods including Large Language Models (LLMs) that have not had much coverage in other surveys. 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 are never considered together. Our experiments shed some light on how the characteristics of datasets affect the behavior of the methods that we compare.

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

1 Introduction

Named Entity Recognition (NER) is a field of computer science and natural language processing (NLP) that deals with the identification and classification of named items in unstructured text. The items in question belong to predefined semantic types such as persons, locations, and organizations [Grishman and Sundheim, 1996a]. NER is today a key component in areas including machine translation [Babych and Hartley, 2003], question-answering [Mollá et al., 2006], and information retrieval [Guo et al., 2009].

A number of NER systems have been developed, particularly for English, but also for other languages, including Chinese [Liu et al., 2022] and French [Mikheev et al., 1999]. Early NER systems used algorithms based on handcrafted rules, lexicons, and spelling features [Rau, 1991]. Systems were subsequently developed that used algorithms based on machine learning [Nadeau and Sekine, 2007], neural networks [Collobert, 2011], and transformers [Labusch et al., 2019a].

Several NER surveys have been published. Among these, we might mention [Nadeau and Sekine, 2007, Shaalan, 2014], in which the authors present an overview of methods, from the rise of rule-based NER systems through to those based on machine learning. The authors in [Goulart et al., 2011] analyze relevant work in relation to biomedical texts over the period 2007-2009. Marrero et al. [2013] summarize the work on NER from a theoretical and practical point of view. An early study on neural network-based NER is available in [Sharnagat, 2014]. Deep learning methods are studied in-depth in [Li et al., 2020a]. However, the authors do not address the use of transformers because their use remained limited at the time of their study.

Surveys that include transformer-based methods are largely absent from the literature. Some recent studies include methods based on neural networks and transformers [Li et al., 2022], but these are far from complete since they do not include recent methods such as large language models (LLMs) and graph-based NER. The study Wang et al. [2022] focuses on graph-based methods specifically for nested named entities, but does not address flat named entities. Moreover, apart from one study, namely Li et al. [2020a] (which does not feature any mention of transformers), the

studies in question do not cover tools used for NER. Since tools are key components of practical interest, and since transformer-based methods have had a significant impact, we are seeking here to present a comprehensive survey of NER tools and frameworks, including popular ones that may still be relevant for low-resource languages or in restricted computational environments.

We also focus on methods intended to be trained on small datasets, an area that, to our knowledge, has not yet been addressed comprehensively in other works. Methods of this kind will often be appropriate where specific types of entities are to be retrieved and annotated data are costly to produce.

The outline of this paper is as follows. First, we define the task of NER and provide some examples of its applications. We then describe the main research approaches that have been developed in this field (see Section 4); in particular, we focus heavily on Large Language Models (LLMs) and graph-based approaches. Methods designed for contexts where only a small sample of annotated data is available are presented in Section 5. Section 6 is devoted to an overview of the best-known tools for pre-trained models. After describing the evaluation schemes used in NER (Section 7), we present corpora from a variety of fields that may be useful resources for the research community (Section 8). In Section 9, for purposes of comparison, we apply the latest versions of five popular frameworks on selected datasets. Finally, we give our conclusion and perspectives in Section 10.

2 Task definition

NER stands for Named Entity Recognition, which is a subtask of NLP focusing on identifying and classifying named entities within text. Named entities are specific words or phrases that refer to real-world objects such as people, organizations, locations, dates, quantities, gene and protein names in the biomedical domain. NER aims to locate and categorize these entities into predefined categories.

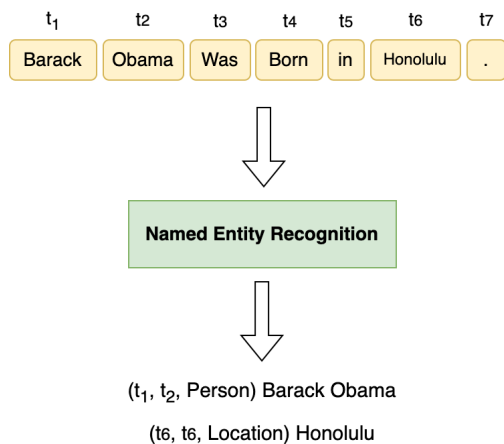


Figure 1: Given a sequence of tokens, NER outputs the boundaries of the named entities along with their associated types.

In a formal context, given a sequence of tokens denoted $T = (t_1, t_2, \dots, t_N)$, NER involves producing a collection of tuples (I_s, I_e, ℓ) , where s and e are both integers that lie within the interval $[1, N]$; I_s and I_e correspond to the beginning and the ending indices of a named entity mention respectively, and ℓ denotes the type of entity from a predefined set of categories. For example, in the sentence "Barack Obama was born in Honolulu." NER would identify "Barack Obama" as the person's name and "Honolulu" as the location, as illustrated in Figure 1.

3 Applications of NER

In this section, we provide a few examples of NER applications.

- **Information extraction:** NER can be used to extract structured data from unstructured text [Weston et al., 2019], such as retrieving the names of persons, organizations, and locations.
- **Information retrieval:** Using NER in the context of information retrieval enhances search outcomes [Banerjee et al., 2019]. This enhancement is achieved through the identification of relevant named entities within both search queries and the obtained results.

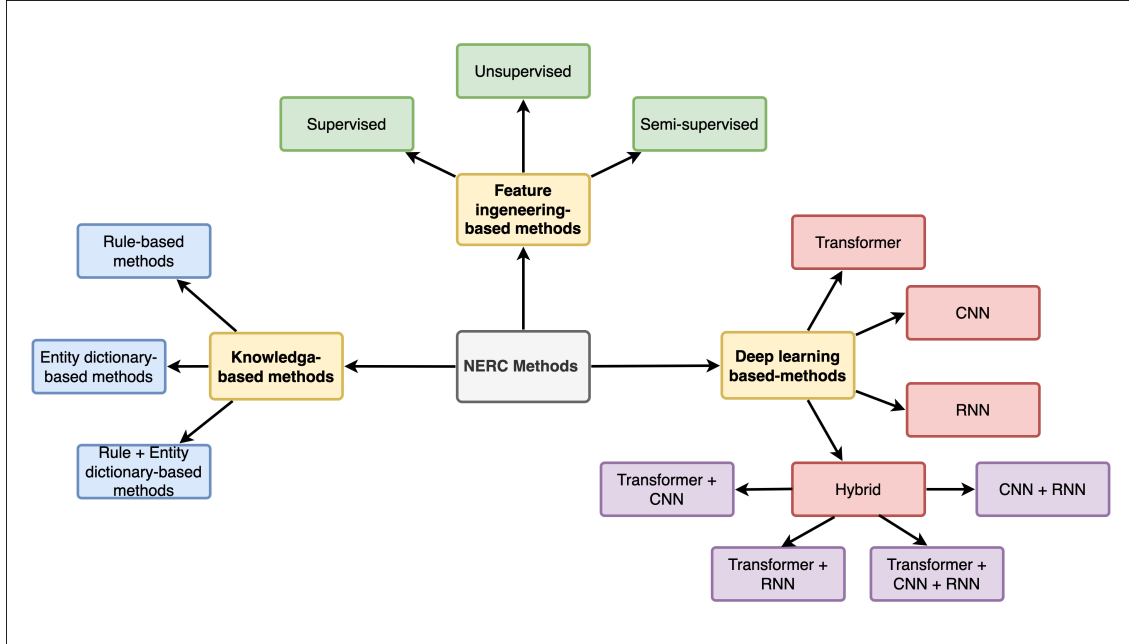


Figure 2: Main approaches to NER.

- **Document summarization:** The integration of NER into document summarization processes can enhance the quality and relevance of generated summaries. By identifying and categorizing key entities, NER can ensure that the summary captures the essential information related to these entities [Roha et al., 2023].
- **Social media monitoring:** Integrated with social media monitoring, NER empowers businesses to automatically identify and categorize mentions of entities like brands and persons. This aids in tracking brand visibility, sentiment analysis, competitor insights, and crisis management. NER also helps spot trends, assess campaign effectiveness, and leverage *influencer marketing* [Sufi et al., 2022].
- **Virtual assistants:** By identifying entities within user input, NER enables virtual assistants to comprehend context and offer tailored responses [Park et al., 2023]. For example, when a user inquires about the weather at a specific location, NER can extract the location entity, allowing the virtual assistant to furnish a relevant response.
- **Named entity disambiguation:** NER can help disambiguate entities with the same name [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.
- **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 Steve Jobs died?", NER can identify "steve jobs" as a person and extract the death date.
- **Language translation:** NER can help improve the accuracy of machine translation by preserving the named entities [Li et al., 2020b].

4 Methods

In this section, we explore various approaches that have been employed for NER. An overview of these approaches is presented in Figure 2.

4.1 Knowledge-based methods

Knowledge-based approaches emerged in the field of linguistics. For instance, Borkowski [1966] introduced an algorithm that employs rule-based lists, utilizing indicators like capital letters to detect company names. Lexical markers are elements that adjoin the named entity and, in so doing, reveal its presence (such as *Mr.* or *Ms.* in front of

names). Here we may cite, for instance, *CasEN* transducer cascade [Maurel et al., 2011], created to recognize French named entities.

Knowledge-based methods do not require annotated data and primarily depend on rules and dictionaries relying on lexical resources and domain-specific knowledge [Sekine and Nobata, 2004, Etzioni et al., 2005a], as depicted in Figure 3. They consist of three components: (1) a set of rules to extract named entities, (2) an optional set of entity dictionaries, and (3) an extraction engine that applies components (1) and (2) to the input text. These approaches require domain experts to provide a large number of rules using lexical markers [Zhang and Elhadad, 2013] or entity dictionaries [Etzioni et al., 2005a, Sekine and Nobata, 2004].

Entity dictionaries compile the most commonly occurring entities within a specific application domain. For example, the ProMiner system [Hanisch et al., 2005] addresses the challenge of identifying gene and protein names in biomedical text, which is complicated due the presence of synonyms and ambiguous terms. ProMiner employs a rule-based approach and a synonym dictionaries to detect potential name occurrences in biomedical text. It associates protein and gene database identifiers with these matches, focusing on multi-word names. The system has been extended to handle highly ambiguous and case-sensitive synonyms using specific detection procedures. The most likely database identifiers are linked to the detected synonyms in an abstract. The system’s performance was tested in the BioCreAtIvE competition, achieving promising results.

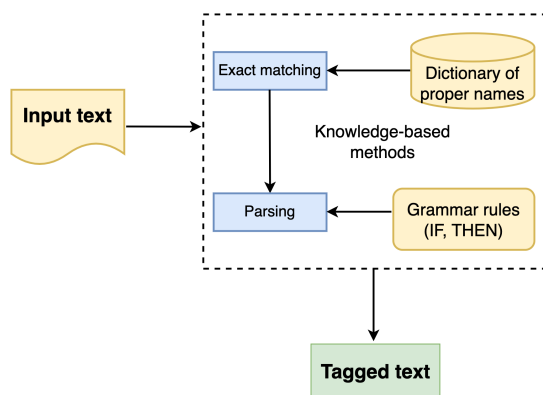


Figure 3: The architecture of knowledge-based methods for NER.

Quimbaya et al. [2016] introduced a method centered around entity dictionaries in electronic health records. The experimental findings demonstrate that this approach enhances the *recall* score while minimally affecting the *precision* score. Note that for this family of methods, *precision* (see Section 7.2) is generally high while *recall* is low due to domain, language-specific rules and incomplete dictionaries. Moreover, incorporating new rules and dictionaries is costly.

4.2 Feature engineering-based methods

Instead of manually crafting sets of rules, feature engineering-based methods are grounded in the principle of automating the extraction of named entities from data. Such techniques can be divided into three primary groups: unsupervised learning methods, supervised learning methods, and semi-supervised learning methods.

4.2.1 Unsupervised learning methods

Unsupervised learning methods do not require any data annotation beforehand. Different approaches exist and are mostly based on similarities in the data. The key idea behind these methods is to group *syntagmas* together based on their shared properties.

Distributions of words are harnessed in [Shinyama and Sekine, 2004a] in order to extract named entities. The authors of this article capitalized on the observation that named entities tend to appear together in multiple news articles, unlike common names. In [Bonney et al., 2011], the authors compute a semantic proximity score by comparing word distributions in documents linked to an entity and the entity type. Shinyama and Sekine [2004b] observe (like [Shinyama and Sekine, 2004a]) that named entities often appear synchronously in different news reports, whereas common nouns do not.

Nadeau et al. [2006] introduced an unsupervised system for building dictionaries and resolving the ambiguity of named entities. Inspired by previous works such as [Etzioni et al., 2005b], their approach involves the amalgamation of extracted dictionaries with publicly accessible ones, resulting in commendable performance.

4.2.2 Semi-supervised learning methods

Semi-supervised learning methods leverage a combination of labeled and unlabeled data to improve model performance. Unlike traditional supervised approaches that rely on labeled data, semi-supervised methods use the additional information available in unlabeled data to improve the NER system’s performance. They learn from the small dataset using a set of rules designed to learn extraction patterns based on a set of relevant markers. They then attempt to find other samples of named entities adjoined by these markers. This learning process is then applied to the new samples in order to discover new contextual markers. By repeating this process, a large number of entities should be found. Collins and Singer [1999] showed that a mere set of seven simple rules, coupled with unlabeled data, can be adequate for achieving effective supervision.

An approach that is commonly adopted in semi-supervised NER consists in using co-training [Kozareva et al., 2005], where multiple classifiers are trained on different data views, and they iteratively label unlabeled instances to build a more robust model. Another method is self-training [Gao et al., 2021], where the initial model is trained on the labeled data, and then used to predict the labels for the unlabeled data. The most confident predictions are added to the labeled data, and the process is iterated.

To sum up, semi-supervised learning methods offer several benefits, such as reducing the dependency on manually labeled datasets and improving performance in low-resource scenarios. Nonetheless, they also come with challenges, including the potential error propagation from the initial labeled data and the need for careful handling of noisy or incorrect predictions from the unlabeled data. However, when applied wisely, semi-supervised learning techniques can significantly enhance NER performance and address real-world challenges where labeled data might be limited or expensive to obtain.

4.2.3 Supervised learning methods

Supervised learning models involve learning rules from annotated data, where human intervention is required to label a set of samples. These labeled samples serve as guidance for the model during the learning process. A learning method is then used to train the model to recognize the distinctive characteristics of the named entities. Subsequently, the learning system generalizes this knowledge to produce a model capable of extracting named entities from new documents. The effectiveness of such method improves in line with the quantity and quality of the annotated data used for training. Using them, the NER task can be viewed as two sub-tasks, namely classification and sequence labeling. The aim is to reproduce an annotation scheme learned from labeled data. The most common algorithms or models in this category are the Hidden Markov Model (HMM) [Morwal et al., 2012, Bikel et al., 1999], the Maximum Entropy model (ME) [Borthwick, 1999, Lin et al., 2004], Support Vector Machines (SVM) [Isozaki and Kazawa, 2002, Makino et al., 2002], and Conditional Random Field (CRF) [McCallum and Li, 2003, Settles, 2004a].

Hidden Markov Model An HMM is a statistical model in which the system being modeled is assumed to be a Markov process. In our context, it is used to identify and classify named entities within a sequence of tokens. With this model, the observed tokens represent the visible states, while the different entity labels are treated as hidden states. The model assumes that the observed tokens depend only on the current hidden state, and can be used to infer the most likely sequence of named entity labels given the observed tokens. An HMM is mathematically described by five parameters as follows:

$$\text{HMM} = \{S, O, \pi, Tr, Em\} \quad (1)$$

where S refers to the number of hidden states (entity labels), O to the number of observations (tokens), π is the distribution of initial state probabilities, Tr is the transition probability matrix, and Em is the emission probability matrix.

The NER problem can be translated into an HMM problem and defined as

$$P(S|O) = P(EN|T),$$

which means that given a sequence of tokens T , the probability of obtaining the sequence of named entities (EN) given T is equivalent to calculating the probability of obtaining the sequence of hidden states S given the observations O . ? proposed **IdentiFinder**, a first HMM that learns to recognize and classify names, dates, times, and numerical quantities. Other works have followed this path [Morwal et al., 2012, Zhao, 2004].

Maximum Entropy ME models represent a statistical modeling approach employed in various NLP tasks, including NER. The fundamental concept underlying these models is to identify a probability distribution that includes potential outcomes, optimizing entropy while adhering to a predefined set of observed constraints. The resulting probability distribution has the highest entropy, is distinctive, aligns with the maximum-likelihood distribution, and takes on the form:

$$p(o|h) = \frac{1}{Z(h)} \prod_{j=1}^k \alpha_j^{f_j(h,o)} \quad (2)$$

where o refers to the outcome, h the context, and $Z(h)$ is a normalization function. The constraints are usually derived from the training data, and the model aims to assign higher probabilities to outcomes that are more likely given the observed data.

In the context of NER, a ME model could be trained to predict the named entity label of a token based on its surrounding context and other relevant features. The features may include information about the current token, its neighboring tokens, POS tags, and more. The model learns the weights associated with these features during training. In [Borthwick et al., 1998], the authors proposed a system called Maximum Entropy Named Entity (MENE) by applying the ME model and using a flexible object-based architecture, which allows it to make use of a broad range of knowledge sources in making its tagging decisions.

Conditional Random Fields CRFs are models used for labeling sequences, like sentences. They consider that neighboring items can affect each other’s labels, making them good for NER. They capitalize on their ability to model sequential dependencies between tokens within a sequence, making them particularly adept at capturing the intricate contextual relationships that define named entities. By integrating both local and global information, they can predict entity labels by considering the labels of neighboring tokens, effectively reducing labeling ambiguity. A CRF model is defined as follows:

$$P(Y|X) = \frac{1}{Z_0} \exp \left\{ \sum_{t=1}^T \sum_{k=1}^K \lambda_k f_k(y_{t-1}, y_t, x, t) \right\} \quad (3)$$

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, 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 converted into a CRF, where the observations are the processed strings and the labels are the possible named entities. Thus, finding the best sequence of named entities corresponds to a sequence of tokens in NER, which is equivalent to finding the best sequence of labels, i.e., $\text{argmax } P(Y = y|X = x)$. Shishtla et al. [2008] implemented a system that extracts information from research articles using CRF. They investigated regularization problems using Gaussian model and focused on the efficient use of feature space with CRF. Settles [Settles, 2004b] presented a framework for recognizing biomedical entities using CRF with a variety of features. He showed that a CRF with only simple orthographic features can achieve good performance.

Support Vector Machine SVMs are a class of machine learning algorithms commonly used for classification tasks. While SVMs are not as widely used for NER as some other approaches like 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. Their approach is an extension of Kudo’s chunking system [Kudo and Matsumoto, 2001]. In this system, each word in a sentence is classified sequentially either from the beginning or the end of the sentence.

4.3 Deep learning-based methods

The landscape of NER and deep learning has further evolved, with the most prevalent methodologies encompassing CNNs, RNNs, and their hybrid combinations. Additionally, these deep learning architectures are frequently combined with other supervised learning algorithms such as SVMs or CRFs. Collobert [2011] played a pioneering role by adopting deep learning techniques for NER. Their work showcased the application of a CNN not only for NER but also for other NLP tasks such as Semantic Role Labeling, POS tagging, and chunking. Of significant importance, these tasks were accomplished automatically, breaking away from the need for manually engineered features. This marked a pivotal moment as deep neural networks were harnessed for these tasks for the first time.

The overarching structure of deep learning-based approaches usually encompasses three fundamental stages (refer to Figure 4): first, the representation of data; followed by context encoding; and concluding with entity decoding. These stages will be elaborated upon in the subsequent sections.

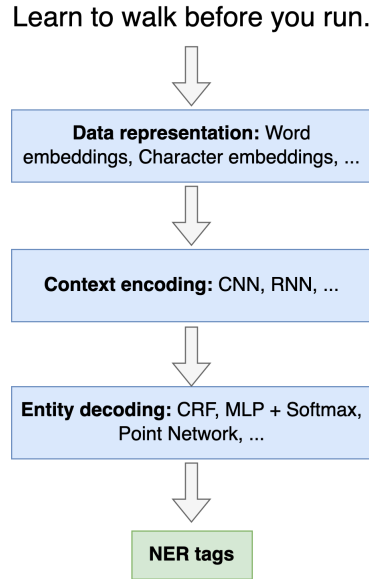


Figure 4: General steps for a deep learning-based NER model.

4.3.1 Data representation

NER requires data to be represented in a machine-understandable form. Textual embeddings can be used to satisfy this requirement. Several textual embedding techniques exist, which we separate into the two categories described below: word embeddings and character embeddings.

Word embeddings are type of word representations that convert words into vector representations in a high-dimensional space. These representations capture the semantic meaning and context of words based on their surrounding words in a given corpus. Worthy of mention are One-Hot encoding, TF-IDF [Ramos et al., 2003], Word2vec [Mikolov et al., 2013], GloVe [Pennington et al., 2014], fastText [Bojanowski et al., 2016] and more recently GPT [Radford et al., 2018].

Traditional word representations, like one-hot encodings, represent each word as a sparse binary vector, where only one element is set to 1 to indicate the presence of that word in the vocabulary. In NER, the use of one hot encoding involves automatically constructing training data, such as from social media messaging applications [Lee and Ko, 2020].

The TF-IDF weight is used to measure the importance of a word in a document relative to a corpus. The Term Frequency (TF) component assesses how often a word appears in a specific document, while the Inverse Document Frequency (IDF) component measures the rarity of a word across the entire corpus. By multiplying TF and IDF, TF-IDF highlights significant words in a document. TF-IDF can be employed in NER as a feature representation method [Karaa, 2011]. As named entities are often rare words, using TF-IDF, named entities that appear frequently in a specific document but rarely across the corpus are assigned higher TF-IDF scores, making them stand out in the feature vectors.

Furthermore, word embeddings are typically learned using unsupervised learning techniques like Word2Vec, GloVe, and fastText. These methods process large amounts of text data and learn word representations that capture semantic similarities and syntactic relationships between words. Several works, including [Collobert et al., 2011, Huang et al., 2015], use word embeddings for NER. Ma and Hovy [2016], for instance, compared the performance of their model using a variety of word embeddings including Word2vec and Glove. The role played by these embeddings in obtaining good results is highlighted in [Lample et al., 2016].

Finally, note that word embeddings may also be combined, as in [Dadas, 2019], where a knowledge base from Wikipedia is used to annotate named entities. In the work cited, the labels are transformed into one-hot vectors and concatenated with Word2Vec or ELMo [Peters et al., 2018a] word embeddings.

Character embeddings are type of word representations that capture the internal structure of words at the character level. Instead of representing words as vectors like traditional word embeddings, character embeddings represent each word as a sequence of vectors representing its constituent characters. The process of creating character embeddings involves breaking down each word into its characters and then representing each character as a vector. These vectors are combined to form the representation of the word, as illustrated in Figure 5 through the utilization of a CNN architecture.

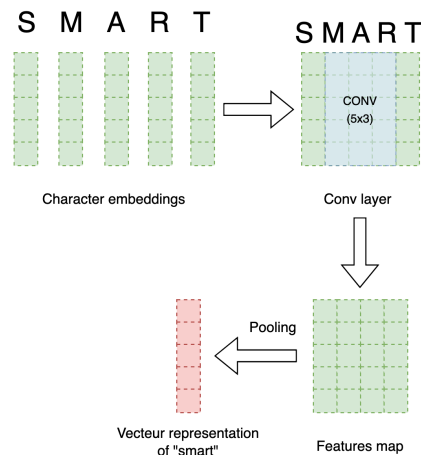


Figure 5: An example of character embeddings using a CNN).

Spelling variations are of great importance and reveal the presence of named entities. Representing individual characters can capture spelling variations by highlighting the syntax of words and their morphology. It can also generate representations of words that were not encountered during the learning phase, i.e. out of vocabulary, by combining different character vectors in order to represent a word.

Character embeddings are generated using various methods. A simple approach is one-hot encoding, where characters are represented as binary vectors with a 1 at the corresponding index. Character2Vec extends Word2Vec to the character level, predicting embeddings by considering neighboring characters. CNN-based character embeddings treat character sequences as images, extracting meaningful features through CNN filters. BiLSTM-based character embeddings use bidirectional LSTM networks to capture contextual information from character sequences.

Note that the CNNs and RNNs we describe in detail in the upcoming section, are used in [Ma and Hovy, 2016, Peters et al., 2017, 2018b] to compute a feature vector for each word. One of the findings in [Lample et al., 2016] is that recurrent models tend to favor later entries and that the resulting feature vectors encode suffixes more than they encode prefixes. For this reason, the authors of the cited work recommend Bi-LSTM to better capture the prefix.

4.3.2 Context encoding

To improve entity recognition, context encoding is a crucial aspect of NER systems aiming at capturing the contextual information of words in a sentence. Context encoding techniques focus on representing words in their surrounding context to take into account the dependencies and relationships between neighboring words. It employs feature vectors obtained using textual embeddings, which can be grounded in words, characters, or even a combination of both. In the sequel, we present popular context encoder architectures such as CNN and RNN. Other models like transformers will be presented separately in dedicated sections.

The CNN models have been devoted initially for images [O’Shea and Nash, 2015]. They use filters to detect patterns and features. Their use is extended to successfully to NLP tasks like NER, where the input text is typically represented as a sequence of word embeddings or characters. CNN is employed to capture contextual information and local patterns. The convolutional layers slide over the input sequence, and multiple filters are used to detect different linguistic patterns at many positions. These filters learn to recognize common patterns, such as suffixes, prefixes, and combinations of words that are indicative of named entities.

Several authors have used CNN for NER task. Collobert et al. [2011] proposed a sentence-based network for tagging words, considering the whole sentence. Each word is represented as a vector, and a convolutional layer generates local features around each word. The global feature vector is formed by combining these local features, with a fixed dimension regardless of sentence length. Global features are then passed to a tag decoder for predicting tags (c.f. Section 4.3.3). In [Gui et al., 2019], the authors proposed an approach for Chinese NER using a CNN with lexicons and a rethinking mechanism [Li et al., 2018]. Rather than relying on a single data pass for final decisions, the rethinking mechanism introduces feedback connections. These connections enable the network to reevaluate decisions by integrating high-level feedback into feature extraction. The authors showed that this method can model characters and potential words in parallel, and the rethinking mechanism can resolve word conflicts by refining high-level features iteratively. For biomedical NER tasks, Zhu et al. [2018] proposed a deep learning approach called GRAM-CNN. This

method utilizes local contexts based on n-gram character and word embeddings through CNN. By leveraging local information around a word, GRAM-CNN can automatically label words without requiring specific knowledge or feature engineering.

On the other hand, RNN have also been widely used for NER tasks. They are designed to process sequential data, making them well-suited for tasks where the context of each word is crucial for accurate labeling [Sherstinsky, 2020]. In the context of NER, the input text is represented as a sequence of embeddings, and each word is fed into the RNN one at a time. The RNN maintains a hidden state that captures information from the previous words. This hidden state is updated at each time step as the RNN processes each word, allowing it to remember important information. One of the challenges with RNN is the vanishing gradient problem, which limits their ability to capture long-term dependencies. To address this issue, variants of RNN like Long Short-Term Memory (LSTM) [Sherstinsky, 2020] and Gated Recurrent Unit (GRU) [Chung et al., 2014] have been introduced. LSTM and GRU networks use gating mechanisms that help them retain and update information more effectively over long sequences, making them better suited for NER tasks. Thereby, Huang et al. [2015] proposed a LSTM model for NER and demonstrated that incorporating a CRF layer as a tag decoder enhanced performance. In other domains, similar systems were applied by Chalapathy et al. [2016] for Drug NER, and Zhang and Yang [2018] for Chinese NER. RNN based on a Bi-LSTM architecture were used in [Huang et al., 2015]; this technique has subsequently been used by other researchers [Ma and Hovy, 2016, Lample et al., 2016].

4.3.3 Entity decoding

Tag decoder architectures play a vital role in NER systems for predicting entity labels for each work in a given sequence. Several tag decoder architectures have been proposed, each with its strengths and suitability for different NER scenarios. Some common tag decoder architectures for NER include: CRF, Multi-Layer Perceptron (MLP), and Pointer Networks [Vinyals et al., 2015]. The research conducted by [Lample et al., 2016, Ma and Hovy, 2016], demonstrated that employing CRF for entity decoding yields improved outcomes in NER. Many deep learning based NER models use a CRF layer as the tag decoder on top of an BiLSTM [Luo et al., 2018, Lin et al., 2019] or a CNN layer [Knobelreiter et al., 2017, Feng et al., 2020].

NER is typically approached as a sequence labeling task, where the goal is to assign entity tags to individual words. One simple method to achieve this is by using a MLP with a Softmax layer as the tag decoder. This formulation transforms the sequence labeling task into a multi-class classification problem, where each token's tag is independently predicted, see for instance [Gallo et al., 2008, Lin et al., 2019].

On the other hand, *Pointer Networks* are RNN devoted for tasks with large output spaces, like sequence-to-sequence tasks. The key idea is allowing the model to point directly to elements in the input sequence rather than relying on a fixed vocabulary. This flexibility is beneficial when dealing with output sequences containing items not present in the training data or with unknown output lengths. The model employs attention mechanisms to compute soft alignment scores between input and output elements, representing the probability of pointing to each input item for generating specific outputs. This approach has proven successful in many NLP tasks such as NER [Zhai et al., 2017].

4.4 Transformer-based language methods

Language models are a key component of NLP. They led to development of methods that can process, analyze, and generate natural text trained on a huge data set [OCDE, 2023]. The most recent of these methods are built upon the transformer architecture.

Let us remember that a transformer [Vaswani et al., 2017] is a neural network pre-trained on a large corpus of texts that aims to solve tasks sequence-to-sequence while easily handling long-distance dependencies. In general, the transformer model is based on the encoder-decoder architecture. The task of the encoder is to encode the output sequence into a fixed length vector called a context vector, which is then fed into a decoder. The decoder reads the encoder output along with the decoder output at the previous time steps to generate an output sequence. Figure 6 depicts the architectural representation of a transformer similar to one in the original paper [Vaswani et al., 2017]. Thereby, Transformer-based models belong to two broad families: encoder-based models and decoder-based models. In the following, we focus on encoder-based Transformer models, and will cover models based on decoders or a combination of both, including LLMs, in the next section.

Transformer-based encoder models have had a significant impact on NLP, including on NER. They can be used either directly for NER as context encoder, or to provide inputs to other models such as CRF or LSTM. BERT [Devlin et al., 2018], for instance, is used in [Schweter and Baiter, 2019] as character representations for German language. Other authors [Labusch et al., 2019b] demonstrated that using BERT as a classifier outperforms other Bi-LSTM-CRF based methods, such as that proposed in [Riedl and Padó, 2018]. In [Boutalbi et al., 2022, Ait-Saada and Nadif, 2023a,b, Nadif and Role, 2021], the authors have studied the performance of BERT in unsupervised learning context.

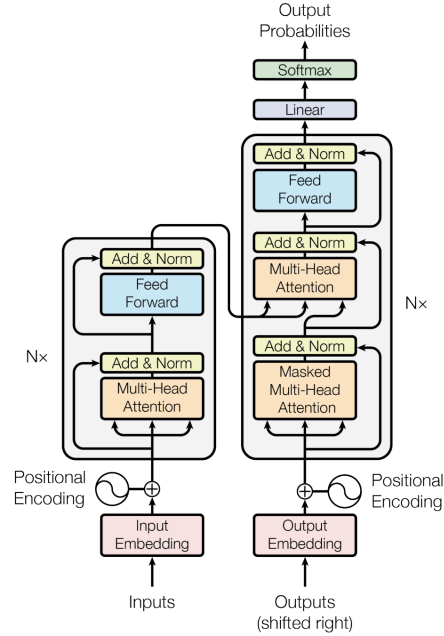


Figure 6: The Transformer architecture [Vaswani et al., 2017].

Several hybridized works combining BERT and RNN, such as by [Liu et al., 2021a]. They rely on BERT to extract the underlying features of text, a BiLSTM to learn a context representation of the text and a Multi-Headed Attention Mechanism (MHATT) to extract chapter-level features.

Transformer-based encoder models may be adapted and improved in various ways to make them more efficient for the purposes of NER. In [Yan et al., 2019] the transformer encoder is used, not only to extract word-level contextual information, but also to encode character-level information within a word. In [Choudhry et al., 2022], the authors proposed an approach for French language using adversarial adaptation to counter the lack of labelled NER datasets. They trained the models on labeled source datasets and used larger corpora from other domains to improve feature learning.

Several variants of BERT such as distilbert [Sanh et al., 2019] and Roberta [Liu et al., 2019a] have proven to be very effective and widely adopted for NER tasks. DistilBERT is a distilled version of BERT, created through a knowledge distillation process, which retains much of BERT’s language understanding capabilities while being significantly smaller in size and faster in computation. Roberta utilizes a larger pre-training corpus and employs a novel training approach called masked language model. This enables it to capture deeper contextual embeddings and representations, making it exceptionally skilled at understanding intricate linguistic structures and context-dependent information.

The paper of Abadeer [2020] explores the challenges posed by the large size and resource-intensive nature of BERT models in deploying them to production environments. To address this issue, the researchers fine-tune DistilBERT on medical text for the NER task involving Protected Health Information (PHI) and medical concepts (MC). Notably, for PHI NER, DistilBERT achieves nearly identical F1 scores to medical version of BERT while consuming almost half the runtime and disk space. However, in the detection of MC, DistilBERT’s F1 score is lower than medical BERT variant. The authors in the paper [Mehta and Varma, 2023] tried to solve a multilingual complex NER challenge by refining the basic XLM-Roberta model on datasets from several different languages.

4.5 Large language model-based methods

Large Language Models (LLMs) represent a novel category of deep learning models with the capability of performing a wide array of tasks, including translation, summarization, classification, and content generation. These models, encompassing transformer-based language models, are distinguished by their extensive parameter count, often numbering in the tens or hundreds of billions. They undergo training on substantial 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	Context	Outcome
GPT-NER [Wang et al., 2023]	Transforms sequence labeling to text generation	General NER tasks	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	Few-shot learning in NER	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	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	Task-Oriented Dialogue Systems	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	Historical texts in multiple languages	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	German medical texts	Outperforms rule-based baseline
Large code generation models [Li et al., 2023]	Utilizes generative LLMs of code for Information Extraction tasks	Information extraction tasks	Consistently outperforms fine-tuning moderate-size models and prompting NL-LLMs in few-shot settings

The basis of LLMs is to be found in the transformer decoder model, where multiple attention layers are stacked to create a highly complex neural network. The architectures and pre-training objectives employed in existing LLMs closely resemble those of smaller language models. The difference lies in the significantly increased size of both the model and the training data. Some LLMs, like T5 [Raffel et al., 2020], are hybrids, utilizing both encoder and decoder components of the Transformer architecture to enhance understanding and generation capabilities, allowing for more versatile and nuanced language processing.

The popularity of LLMs is due to their outstanding performance across various NLP tasks. They excel in tasks like text classification [Hegselmann et al., 2023], question answering [Robinson et al., 2022] (QA), text generation [Muennighoff et al., 2022], and machine translation [Hendy et al., 2023]. However, despite their impressive performance in numerous NLP tasks, LLMs have shown some limitations when applied to NER. This is primarily because NER is a sequence labeling task, while LLMs are originally designed for text generation.

To address the gap between LLMs and NER, Wang et al. [2023] introduced a method known as GPT-NER, which effectively transforms the sequence labeling task into a text generation task. With GPT-NER, the transformation involves representing the NER task as a text generation problem. For instance, the task of identifying location entities in the sentence "Paris is a city" is transformed by GPT-NER into the task of generating the text sequence "@@Paris## is a city," where the special tokens "@@" and "##" are used to mark the boundaries of named entities. In a series of experiments GPT-NER was able to produce promising results comparable to supervised approaches when applied to medium or large datasets, even outperforming supervised approaches when dealing with small datasets. This innovative approach successfully bridges the gap between LLMs and sequence labeling tasks, making them more suitable for NER and showing their potential for impressive performance in this area.

Ashok and Lipton [2023] proposed PromptNER, a few-shot learning algorithm for NER. PromptNER requires a set of entity type definitions in addition to annotated examples. Given a sentence, PromptNER prompts an LLM to produce a list of entities with corresponding explanations justifying their compatibility with the entity type definitions provided. The results show that PromptNER outperforms other few-shot and cross-domain methods on certain datasets.

Laskar et al. [2023] conducted an evaluation of ChatGPT on various NER tasks, providing insights into the strengths and limitations of LLMs in this area. The mixed results highlight the importance of contextualizing the use of LLMs according to the specificities of the task. Similarly, Hu et al. [2023] investigated ChatGPT for clinical entity recognition in a zero-shot context. The study compares the performance of ChatGPT against that of GPT-3 [Brown et al., 2020] and a BioClinicalBERT [Alsentzer et al., 2019] model specifically trained on synthetic clinical notes from MTSamples.

ChatGPT was shown to outperform GPT-3 in this particular setting. However, ChatGPT performs less well than the supervised BioClinicalBERT model.

Kim et al. [2023] explored the combined use of GPT-2 [Radford et al., 2019] and BERT for disambiguating named entities in dialogue systems, illustrating the effectiveness of a hybrid approach. GPT-2 serves as a generator during the training phase, and input sentences and target entities are used together. During inference, input sentences guide the evaluation to identify whether the target entity can be matched exactly. While the accuracy of the GPT-2-based approach is slightly lower than that of BERT, it illustrates the potential of combining LLMs for specific NER tasks.

Several studies have looked at how LLMs may be applied in specific contexts. For instance, De Toni et al. [2022] tested the multitask T0 model on historical texts, highlighting the challenges associated with NER in historical and multilingual contexts. They found that the model struggles with accurate zero-shot NER on historical documents due to various factors, including optical character recognition errors and linguistic variation. The accuracy of T0's predictions varied depending on the language of documents and on the historical period in which they originated. While the model underperformed compared to the state of the art in NER for historical documents, it demonstrated a particular ability to identify the language and publication date of the documents.

In contrast, Kammer et al. [2023] focused on the unique challenges presented by elliptical compound nominal phrases in German medical texts. Elliptical compound nominal phrases represent a distinctive linguistic construction where certain elements of sentences or coordinated segments are omitted to avoid repetition. For instance, the phrase "Vitamin C, Vitamin E, and Vitamin A" is condensed to "Vitamin C, E, and A". These phrases present challenges for entity extraction and disambiguation. To address this issue, a dataset of over 4,000 manually annotated ECCNPs (Elliptical Compound Coordinate Nominal Phrases) was created, and a generative model developed based on the Transformer architecture, which includes both encoder and decoder modules. This model demonstrated a high accuracy, significantly outperforming a rule-based method [Aepli and Volk, 2013]. Large Language Models (LLMs) like GPT-3.5, however, performed less well.

Li et al. [2023] introduced a novel approach for improving LLM performance in information extraction (IE), especially for NER and relationship extraction (RE), by advocating the use of code-based LLMs (Code-LLMs) such as Codex [Chen et al., 2021a] rather than natural language LLMs (NL-LLMs) such as GPT-3. This alternative approach involves reformulating IE tasks as code generation tasks, aligning structured outputs more efficiently with IE requirements. A key finding is that Code-LLMs with code-style prompts can be better than specially pre-trained models for IE and NL-LLMs in few-shot learning setups. Additionally, code-style prompts yield outputs with lower structural error rates, demonstrating the potential of Code-LLMs for specific sophisticated applications in IE.

Despite the recent enthusiasm for LLMs, it must be remembered that these models remain costly. Indeed, the main drawback of LLMs is the considerable cost of training them, which for now is a major impediment to their widespread adoption. The extensive computational resources and time required for training, given the enormous number of parameters in these models, make the process expensive and resource-intensive. As a result, LLMs are predominantly restricted to large organizations and research institutions with access to substantial computing power. However, although the high training cost limits the widespread availability and accessibility of LLMs, efforts are underway to address these challenges to make them more efficient and more cost-effective.

4.6 Graph-based methods

The work of Marcheggiani and Titov [2017] paved the way for the use of Graph Convolutional Networks (GCNs) [Kipf and Welling, 2016] in NLP. GCN-based methods consider each token as a node of a graph, with the edges representing contextual links to neighboring nodes (refer to Figure 7). A more detailed explanation of how these methods work is to be found in [Daigavane et al., 2021]. The authors of [Cetoli et al., 2017] looked at how GCN may be used in addressing the NER task, and the results that they obtained on the OntoNotes 5.0 dataset [Weischedel et al., 2013] show a significant improvement in performance. Liu et al. [Liu et al., 2019b] proposed an architecture based on GCN to create graph embeddings. These graph embeddings are then combined with token embeddings, and passed through a Bi-LSTM-CRF network to extract named entities. Harrando and Troncy [2021] suggested that NER, traditionally considered as a sequence-labeling problem, may be considered as a graph classification problem, where each token is represented as a node. Each node can then be associated with contextual information and other elements of external knowledge, such as morphological shape or a POS tag. This approach has shown promising results.

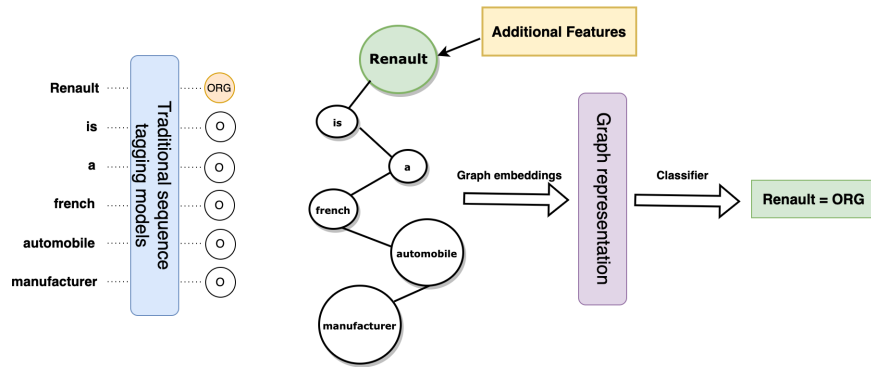


Figure 7: Left side: Conventional model for sequence tagging. Right side: Every word within a sentence transforms into a node of a graph connected to surrounding words and additional features like grammatical characteristics. This graph is subsequently encoded and fed into a classifier to predict entity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

5 Low-resource NER

Recent methods based on neural networks and transformers have shown good results for NER. These methods need to be trained on a sufficient quantity of data; where there is a scarcity of data, this has a severe negative impact on performance. However, labeled data is sometimes difficult to obtain. In this section, we present some ways in which this problem may be addressed.

5.1 Transfer learning

Transfer learning involves taking knowledge that has been obtained while solving one task, and applying this knowledge to a related task [Zhuang et al., 2020]. This technique has been shown to improve model performance in areas including image classification [Shaha and Pawar, 2018], speech recognition [Wang and Zheng, 2015], and time series classification [Fawaz et al., 2018].

Transfer learning in NER involves pretraining a model on a large amount of general text data and then fine-tuning it on a smaller dataset specifically for the target NER task. This approach leverages the knowledge learned from the general data to improve performance on the specific task. A number of studies have been carried out on transfer learning. Lee et al. [2017] looked at transfer learning using RNN in the anonymization of health data. Authors like [Francis et al., 2019, Liu et al., 2021b] explored the use of transformers for NER, and showed that transfer learning significantly improves performance. Recently, Fabregat et al. [2023] proposed several architectures based on a Bi-LSTM and a CRF in order to detect biomedical named entities.

5.2 Data augmentation

Data augmentation artificially increases the amount of training data by creating modified copies of a dataset using existing data. This includes making small changes to data [Dai and Adel, 2020, Sawai et al., 2021, Duong and Nguyen-Thi, 2021] (synonym replacement, random deletion, random insertion, random swap, back-translation, lexical substitution, etc.) or using generative methods to create new data [Sharma et al., Keraghel et al., 2020].

The application of data augmentation techniques to NLP has been done in areas including, for example, 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 makes predictions about words, and not about sentences. Therefore, applying transformations to words may change their labels. Because of this difficulty, there has been very little interest in using data augmentation for NER. However, some authors have attempted to adapt data augmentation techniques for this purpose. In [Dai and Adel, 2020], different simple data augmentation strategies (word replacement, named entity replacement, etc.) are applied with the aim of improving model performance, especially when the dataset contains very few examples. Other authors, including Sharma et al., use more complex strategies such as paraphrasing to generate new data.

Data augmentation is a vital technique in training NER models, especially when dealing with limited datasets. Traditional methods include paraphrasing, synonym replacement, and back-translation. In the future, however, data augmentation in NER could be significantly facilitated and enhanced by leveraging LLMs. 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 like GPT3Mix [Yoo et al., 2021], makes it possible to create more robust and more accurate NER models by enriching training data with a wide range of linguistic variations and contextual scenarios.

5.3 Active learning

Active learning is a form of semi-supervised learning. The main idea in active learning is that if a learner (the learning algorithm) is able to choose the data that it wishes to learn from, it can perform better than it otherwise would using traditional learning schemes. One of the main challenges of active learning is to determine what constitutes the most informative data, and how the learner can recognize this data. The most common approach today is uncertainty sampling [Settles, 2009], in which the model selects the examples for which its current prediction is the least reliable.

Where active learning is successfully applied to NLP, either performance may be improved using the same amount of data, or a similar performance may be maintained while reducing the amount of data and annotation required to develop an effective model. In the field of deep learning research, pioneering adoptions of active learning have produced promising results [Siddhant and Lipton, 2018]. As regards NER, approaches based on deep learning have been proposed in several works [Shen et al., 2017, Yan et al., 2022].

5.4 Few-shot learning

Few-shot learning aims to build accurate machine learning models with less training data. This technique can be implemented by applying transformations to the data, by applying changes in the algorithms, or by 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. Applying changes in the algorithms involves using pre-trained models as feature extractor, or refining the already trained model in relation to the new data by continuing the back-propagation of the gradient. Using dedicated algorithms involves using networks that do not learn from a single instance, but from pairs or triplets of instances. A larger database can therefore be obtained. In the context of named entities, studies including [Fritzler et al., 2019, Hou et al., 2020] proposed adapting prototypical networks [Snell et al., 2017] for NER. However, their use of this technique failed to achieve good performances. The authors in [Yang and Katiyar, 2020] propose a few-shot learning method based on nearest neighbors and structured inference, with an approach that goes beyond classical meta-learning approaches. Cui et al. [2021] treat the NER task as a language template classification problem. Their approach outperforms traditional sequence labeling methods. A growing number of works are recognizing the potential of few-shot learning in NER [Hofer et al., 2018, Huang et al., 2020].

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

5.5 Zero-shot learning

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

Zero-shot learning can be applied in NER as a way of detecting new types of named entities. Aly et al. [2021] propose an architecture using textual descriptions. ZERO [Van Hoang et al., 2021] performs zero-shot learning and few-shot learning by incorporating external knowledge in the form of semantic representations of words. In [Yang et al., 2022], multilingual sequence translation is proposed as a possible solution in the case of low-resource languages where labeled data is scarce or absent. Multilingual sequence translation acts as a bridge by transferring knowledge from the source language to a language possessing large amounts of annotated data. Additionally, 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.

6 Software frameworks

In this section we present the best-known and most commonly used NER frameworks today:

- **OpenAI** [OpenAI] offers a range of AI tools, including GPT models, for text generation, question answering, and more. Although its focus was not originally on NER, OpenAI is now beginning to explore this domain. The API is known for its flexibility and user-friendliness, with an additional emphasis on safe, ethical AI use.

Table 2: 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/

- **spaCy** [Honnibal and Montani, 2017] is a free open-source library for advanced natural language processing (NLP) in Python. It is designed to make it easy to construct systems for information extraction or general-purpose natural language processing. spaCy offers a number of analysis operations 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 a specific dataset. These models are based on convolutional networks (CNNs) and transformers.
- **NLTK** [Bird, 2006] is a suite of Python modules dedicated to NLP. NLTK integrates more than 50 corpora and lexical resources such as WordNet, as well as a suite of word processing libraries for textual analysis operations such as tokenization, POS tagging, sentiment analysis, topic segmentation, and speech recognition. Unlike spaCy, which includes algorithms adapted to different problems and which are managed by the library, NLTK gives users the freedom to choose from 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 allowing the separation of text into tokens, POS taggings, and training models for NER (based on CRF). Unfortunately, these features are not available for all languages, and each language has its own specific characteristics. For example, NER is available only for English, Spanish, German, and Chinese.
- **Apache OpenNLP** [Kwartler, 2017] is a library that supports the most common NLP tasks, such as NER, language detection, POS tagging, chunking, etc. Unlike other frameworks, which detect all NER-named entities using the same model, OpenNLP offers a specialized model (based on maximum entropy) for each type of named entity.
- **Polyglot** [Al-Rfou et al., 2015] is a NLP pipeline for Java. It can handle a much wider range of languages than other frameworks. For NER it includes models for 40 different languages.
- **Flair** [Akbik et al., 2019] is a free, open-source library allowing the creation of an NLP pipeline supporting multilingual applications. Different language models, such as Flair [Akbik et al., 2019] or BERT [Vaswani et al., 2017] can be used alone or in combination. Flair has its own embeddings, but also offers ELMo [Peters et al., 2018a] and BERT embeddings.
- **Hugging Face** [Wolf et al., 2020] provides open-source NLP technologies. It offers two categories of services: either free services or paid services aimed at businesses. This framework owes its popularity to its transformers library, which has an API providing access to several pre-trained models. Hugging Face also offers a collaborative platform allowing users to 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, 2022] is a Python library for training and tuning NER models implemented in Pytorch. It has a web application with an intuitive interface allowing predictions to be visualized.

Packages like Apache OpenNLP, Stanford CoreNLP, and spaCy are also accessible in languages other than Python. For instance, openNLP¹ is an R package that leverages the capabilities of the Apache OpenNLP library, originally Java-based, by acting as an interface within the R environment. Similarly, the spacyr² package connects R with spaCy. Notably, the spacyr package facilitates NER using spaCy’s pre-trained language models. Other solutions like the reticulate³ package make it easier to achieve interoperability between R and Python enabling Python libraries such as Hugging Face to be accessible within R.

7 Evaluation of NER systems

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

7.1 Annotation schemes

To segment and label entities, different types of annotation schemes have been proposed. These annotation schemes or encodings are used to determine the type and position of entities in the sentence. Considered as a sequence-labeling task, the purpose of NER is to find the corresponding sequence tag for a given textual sequence. In existing annotation schemes, the first token of a named entity is tagged as B (Begin). Where a named entity comprises more than one token, the last token is tagged as either E (End) or I (Inside), depending on the schema, with any intermediate tokens being tagged as I. Tokens that are not part of a named entity are tagged as O (Outside). A tag schema can include any or all of these four tags. Common schemes are: BIO, IO, IOE, IOBES, IE and BIES.

- The IO scheme is the simplest method applicable to this task, where each token in the dataset 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 by 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 IOB, but instead of marking the start of the named entity (B), it denotes the end of the entity (E).
- IOBES serves as an alternative to the IOB scheme, offering more information regarding the boundaries of named entities. It uses four 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 much like IOE, with the difference lying in the labeling of 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.

For a comparison of these annotation schemes, refer to Table 3 which illustrates the application of each scheme to a single sentence example.

It is worth noting that the tag scheme chosen may also affect NER performance; for example, Alshammari and Alanazi [2021] found that IO outperforms other schemes in the case of articles written in Arabic. The study [Chen et al., 2021b] shows that IO scheme is more suitable for the steel E-commerce data than BIO and BIEO schemes.

¹<https://cran.r-project.org/web/packages/openNLP/index.html>

²<https://cran.r-project.org/web/packages/spacyr/index.html>

³<https://cran.r-project.org/web/packages/reticulate/index.html>

Table 3: Comparison of different annotation schemes on a single 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

7.2 Evaluation strategies: exact or relaxed evaluation

The evaluation of NER systems is based on comparing predictions with a gold standard. In making this comparison, either of two strategies may be employed, namely exact evaluation or relaxed evaluation. In exact evaluation, the contours of the named entity and its type must exactly match the gold standard. In contrast, relaxed assessment is based on a scoring system where any entity with the correct type earns a certain number of points, even though the contours may not be correct. Similarly, entities also earn points where the contours are correct, but not the type. The MUC [Grishman and Sundheim, 1996b] and ACE [Doddington et al., 2004] evaluations are based on relaxed evaluation methods. On the other hand, CoNLL-2003 [Sang and De Meulder, 2003] uses exact evaluation, which is the strategy most commonly used today for evaluating NER systems.

7.3 Metrics

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

- Precision corresponds to the proportion of named entities successfully recognized by a model in relation to the total number of named entities.

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4)$$

where TP is the number of True Positives and FP is the number of False Positives.

- Recall measures the number of relevant named entities retrieved by a model against the total number of relevant named entities.

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (5)$$

where FN is the number of False Negatives.

- F1-score reflects a model's ability to effectively detect named entities, with a trade-off between precision and recall.

$$\text{F1} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (6)$$

The above metrics can be computed for each class of entities. They can be aggregated when more than one type of entity is being considered.

- Macro-average: the metric (for instance, F1-score) is computed for each class separately, with the macro-average corresponding to the mean of these values.
- Micro-average: it gives equal weight to each sample.

To go beyond these aggregated metrics, the authors in [Fu et al., 2020] proposed a new evaluation method, which involves defining 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.

8 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 8 gives an

overview of 26 datasets for English NER, with between one and 505 types of entities in various domains: medical data, news, social media, etc.

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

- **CoNLL-2003**: a dataset consisting mainly of news articles from Reuters.
- **OntoNotes 5.0**: a dataset comprising various genres of texts (phone conversations, newswires, newsgroups, broadcast news, broadcast conversations, weblogs, and religious texts).
- **WNUT2017**: a dataset comprising various genres of texts (tweets, reddit comments, YouTube comments, and StackExchange).
- **BioNLP2004**: a medical dataset that provides 2000 MEDLINE abstracts.
- **FIN**: a dataset containing a set of financial data made public through the US Security and Exchange Commission. This dataset has been annotated by [Alvarado et al., 2015].
- **NCBI Disease**: a dataset containing disease names and concept annotations from the NCBI Disease Corpus.
- **BC5CDR**: a dataset consisting of three separate sets of articles with chemicals, diseases, and their relations.
- **MITRestaurant**: a dataset containing a set of online restaurant reviews.
- **Few-NERD**: a dataset containing a set of Wikipedia articles and news reports. Since this dataset is very large, we trained the models on a limited part of it (20% of the data which represents 32 941 samples).
- **MultiCoNER**: a large multilingual dataset that covers three domains: Wiki sentences, questions, and search queries.

The characteristics of these datasets are given in Table 4.

Table 4: Description of the datasets used in our comparative study. “#” refers to the number of samples.

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

9 Experiments

In this section, we detail the protocol adopted to evaluate the effectiveness of the selected algorithms.

9.1 Settings

This section describes the experimental conditions used for our study.

9.1.1 Datasets

Our experiments were conducted on ten datasets with texts from various domains (Table 4). They have different sizes and different numbers of classes, enabling us to test the models in a variety of circumstances.

9.1.2 Models

We select the frameworks based on their availability (free and open source) and their ability to train models on custom datasets. Consequently, those retained were Flair, Stanford CoreNLP, Apache OpenNLP, spaCy, and Hugging Face. NLTK was excluded from the analysis because it comes with the Stanford Named Entities tagger. For frameworks that contain more than one algorithm, we choose several models. For spaCy the three models chosen were three networks: a small CNN (en_core_web_sm), a large CNN (en_core_web_lg), and a transformer based on basic RoBERTa [Liu

et al., 2019a]. For Hugging Face, we choose four models: basic BERT architecture, with and without lowercasing, the distilled version of BERT, and a wide RoBERTa architecture. For OpenAI, we select the most recent and updated version of GPT, known as GPT-4-1106-preview.

9.1.3 Material conditions

All experiments were conducted on an Amazon p3.2xlarge VM with Amazon Linux 2 and Python 3.10. The instance has 8 vCPUs, 16 GB of GPU (Nvidia Tesla v100) and 64 GB of memory.

9.2 Method

Experiments involved the following three steps:

9.2.1 Data format and annotation scheme

Each corpus was encoded in the CoNLL-U format in accordance with the BIO scheme where: a) word lines contain the annotation of each word, 2) blank lines mark sentence boundaries. Since each framework uses its own format for data representation, the initial CoNLL-U format then had to be converted to the required framework-specific formats. For GPT-4, dedicated prompts were used, each tailored to specific datasets. Every prompt outlines the primary categories of the named entities targeted, accompanied by a few examples. Figure 8 presents an example of a prompt with a response formatted in JSON.

In our study, we deliberately adopted an experimental approach that slightly diverges from the one presented in the GPT-NER paper Wang et al. [2023]. Although the method used in that prior research demonstrated effectiveness, it was based on the use of specific prompts for each named entity type, querying the model separately. While effective, this technique tends to simplify the task for the model by reducing the need for disambiguation between different entity types. Our choice was to employ a less directed strategy, aiming to assess the model’s overall ability to accurately identify and categorize named entities in more complex and varied situations. This approach more closely resembles real-world usage conditions, where entity disambiguation and context understanding play a crucial role in the model’s overall performance.

9.2.2 Models and hyper-parameters

Flair For Flair models, we took the standard architecture based on an LSTM-CRF network with Flair embeddings. The hyper-parameters were obtained via a grid search where `hidden_layer_size` is in (64, 128, 256) and `learning_rate` in (0.05, 0.1, 0.15, 0.2). The values giving the best F1-score on the validation set were selected. This gave a `learning_rate` of 0.1 and a `hidden_layer_size` of 256, these values also being used in the pre-trained version of the chosen architecture.

Hugging Face For the Hugging Face models, CRF was used to decode the entities (c.f Section 4.3.3). The same grid search method was used with `learning_rate` in (0.001, 0.0001, 0.00001), with and without `max_grad_norm`. The number of epochs was set to 10 and `weight_decay` to 0.01.

Apache OpenNLP The default hyper-parameter value was selected: only the number of iterations can be changed, so we kept the default value (i.e. 300).

Stanford CoreNLP and spaCy The values of the pre-trained models were selected.

9.2.3 Evaluation

To assess the quality of the results we use a strategy of exact evaluation. Our chosen metric is F1-score, since this reflects the two other metrics discussed in section 7.3 (namely, precision and recall). In the case of Apache OpenNLP, results for some datasets are absent, since only three types of named entities are detected by this framework (persons, organizations, and locations), and datasets not containing these three types were therefore not included in the evaluation.

To assess the differences between the models, we perform Friedman test [Friedman, 1940]. The models are ordered according to their F1-score and the null hypothesis can be stated as: "the median is equal". When the null hypothesis is rejected (we choose 0.1 as threshold of significance), we apply Nemenyi’s method [Nemenyi, 1963] to determine the groups that differ. We replace missing F1-score values with zeros, specifically for Apache OpenNLP.

- You**
 Prompt: Please identify the named entities in the sentence provided, classifying them as either "CHM" for chemical entities or "DIS" for disease entities. "CHM" refers to chemical substances, including drugs, compounds, biomolecules, and more, like "Aspirin" or "Carbon Dioxide". "DIS" pertains to medical conditions or diseases, such as "Diabetes" or "Glioblastoma". Present your findings in JSON format, with keys "CHM" and "DIS", listing the identified entities under the appropriate category.

Sentence for analysis: "In this model of chronic renal failure."

- ChatGPT**
 Based on the provided sentence, here are the identified named entities classified as "CHM" and "DIS" in JSON format:

```

json
{
  "CHM": [],
  "DIS": ["chronic renal failure"]
}

```

In this sentence, "chronic renal failure" is a disease entity (DIS). There are no chemical entities (CHM) mentioned in the sentence.

Figure 8: An example of a prompt for detecting named entities in the NCBI Disease dataset.

9.3 Results and discussion

The results of our experiments are reported in Table 5. The main findings are the following.

Table 5: Comparison of NER frameworks in terms of Macro-averaged F1-score. Best and second-best scores are respectively in bold and underlined. The dashes represent missing values because Apache OpenNLP does not allow the addition of other named entity categories beyond the four standard types: people, organizations, locations, and miscellaneous.

Frameworks	Algorithms	Macro-averaged F1-score									
		CoNLL-2003	OntoNotes	WNUT2017	FIN	BioNLP2004	NCBI Disease	BC5CDR	MITRestaurant	Few-NERD	MultiCoNER
Apache OpenNLP Stanford CoreNLP Flair	Maximum Entropy	80.00	67.83	-	<u>63.24</u>	-	-	-	-	-	-
	CRF	85.18	63.87	8.34	<u>55.25</u>	73.26	86.10	85.22	70.57	45.13	19.39
	LSTM-CRF	<u>90.35</u>	80.10	38.07	74.23	<u>71.64</u>	<u>86.21</u>	90.27	78.33	<u>60.03</u>	56.27
spaCy	CNN-small	81.26	69.30	9.01	55.12	65.92	77.92	80.83	75.62	40.55	35.63
	CNN-large	85.64	69.60	9.78	54.71	66.17	79.15	79.66	76.39	40.01	35.82
	roberta-base	89.92	<u>81.04</u>	<u>41.84</u>	63.18	66.56	87.05	87.08	79.09	59.15	55.21
Hugging Face	xlm-roberta-large	91.46	81.57	43.92	48.68	71.43	85.25	<u>87.41</u>	80.12	61.59	58.15
	distilbert-base-cased	88.12	77.63	25.45	43.74	69.63	84.42	<u>84.03</u>	77.67	58.62	55.17
	bert-base-uncased	88.89	76.99	32.12	46.84	70.50	85.64	85.78	<u>79.18</u>	58.16	59.96
	bert-base-cased	90.09	79.55	33.32	39.53	69.46	85.27	85.14	78.48	59.48	56.64
OpenAI	GPT-4	62.74	33.61	18.82	36.70	41.32	57.46	55.67	41.38	44.96	33.61

On datasets with a large training set (OntoNotes, CoNLL-2003, Few-NERD and MultiCoNER), the architectures giving the best results are those based on transformers. Our hypothesis is that transformers are over-parameterized and are thus favored by large amounts of training data. However, the GPT model seems to not work correctly when it comes to recognizing named entities; it does not even appear in the top 3, and this is the case across all datasets. This can be explained by the fact that the model struggles to disambiguate named entities that can be associated with multiple classes. In addition, the GPT model also faces difficulties in detecting composite named entities, such as disease names. Indeed, these composite entities often consist of multiple words and can be particularly challenging for the model to recognize accurately. Training a supervised model becomes even more crucial when dealing with complex and composite named entities, as it allows for better disambiguation and classification in such cases.

For BioNLP2004, which also has a large training set, LSTM-CRF and CRF have slightly better results than Hugging Face’s transformers. Our intuition is that biomedical data has little in common with the data used to pre-train the transformers, so the pre-training does not greatly contribute to their performance. However, on medical datasets containing very few kinds of named entities (NCBI Disease and BC5CDR), transformers appear among the most efficient architectures along with the one based on LSTM-CRF, with a slight difference in favor of LSTM-CRF. This performance improvement over the BioNLP2004 dataset may be related to the reduced number of named entity types in these datasets.

On a dataset such as FIN with a small training set, transformers can be seen to perform less well: LSTM-CRF obtains far better results, with an F1-score of 74.23, compared to 63.18 for the best transformer, namely RoBERTa-base. The difference in performance is even more marked in the case of xlm-roberta-large, with an F1-score of 48.68, although this transformer excels on the other datasets. For FIN only three of the four types of entities are present in the validation set, which explains this drop in performance of transformers: they are sensitive to hyper-parameter tuning and require well-formed validation sets.

On MITRestaurant and WNUT2017 (which is also a small dataset), xlm-roberta-large performs the best. Since RoBERTa is trained on Common Crawl, and since the type of data in these two datasets is abundant on the web, the pre-training is likely to be of considerable benefit.

It is also interesting to note that LSTM-CRF performs well on all datasets, thus making it a good choice independently of considerations regarding the type of data.

It will also be remarked that the bert-base-cased and bert-base-uncased models do not yield the same results, despite the fact that they were trained under the same conditions. With bert-base-uncased, text is lowercased before the tokenization step, while with bert-base-cased it does not undergo any transformation. On the CoNLL-2003, OntoNotes, MultiCoNER and Few-NERD datasets, the uncased model gives better results, while the cased model performs better on the others.

Based on Friedman test, the null hypothesis is rejected ($p\text{-value}=4.32e\text{-}10$), and one can assume a statistically significant difference between the median values: a common behavior among the models has little probability.

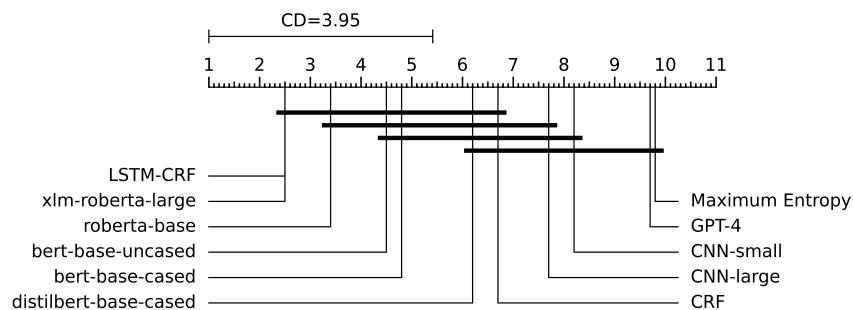


Figure 9: The critical difference (CD) diagram based on F1-score. The plot shows the mean ranks of each model on 10 datasets. The lower the rank, the better the performance of a model. A horizontal line indicates no significant difference between the models crossed by the line.

Nemenyi’s post-hoc test results are given in Figure 9. One would assume that there are no significant differences between LSTM-CRF and all the transformers used in this study. One can also assume that LSTM-CRF and xlm-roberta-large have the same medians. However, no significant difference is found between the bert-base-cased and bert-base-uncased models.

10 Conclusion and perspectives

This article proposes a comprehensive survey on recent advances on named entity recognition in classification context. In particular, we reviewed recent methods such as LLMs, graph-based approaches, and methods for training models on small datasets. We evaluated the most popular frameworks and tools on datasets with different characteristics.

Transformer-based architectures, particularly on larger datasets, have demonstrated good performance due to their substantial parameterization and adaptability. However, our analysis suggests that despite the overall success of transformers, specific models like GPT, and by extension, GPT-4, do not consistently rank at the top for NER tasks. This is attributed to challenges in disambiguating and accurately detecting composite named entities, crucial tasks in NER. In contrast, LSTM-CRF models have shown remarkable consistency and robustness across various datasets, making them a reliable choice for a wide range of applications. The specific needs of biomedical datasets highlight the importance of domain-specific modeling and the limits of general pre-training typically employed in transformer models.

Looking to the future, the role of LLMs models in enhancing NER cannot be understated. Despite the current limitations noted in models like GPT for specific NER tasks, the rapid evolution of language models suggests a promising avenue for incorporating such technologies into more refined and accurate entity recognition systems. GPT-4's ability to understand and generate nuanced text, combined with continual learning and adaptation, presents opportunities to overcome the current challenges noted with composite and domain-specific entities.

Further research should aim to harness and adapt the capabilities of GPT-4 and similar models for improving performance in NER. This involves not only fine-tuning these models for specific NER tasks but also exploring innovative training techniques, data augmentation strategies, and model architectures that leverage the contextual understanding and flexibility of LLMs. Additionally, a deeper dive into preprocessing techniques and their impact on various models will provide a more comprehensive understanding of how to optimize performance across different scenarios.

References

- Ralph Grishman and Beth M Sundheim. Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996a.
- Bogdan Babych and Anthony Hartley. Improving machine translation quality with automatic named entity recognition. In *International EAMT workshop on MT and other language technology tools, Improving MT through other language technology tools, Resource and tools for building MT at EACL*, 2003.
- Diego Mollá, Menno Van Zaanen, and Daniel Smith. Named entity recognition for question answering. In *Proceedings of the Australasian language technology workshop 2006*, pages 51–58, 2006.
- Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. In *SIGIR*, pages 267–274, 2009.
- Pan Liu, Yanming Guo, Fenglei Wang, and Guohui Li. Chinese named entity recognition: The state of the art. *Neurocomputing*, 473:37–53, 2022.
- Andrei Mikheev, Marc Moens, and Claire Grover. Named entity recognition without gazetteers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8, 1999.
- Lisa F Rau. Extracting company names from text. In *Conference on Artificial Intelligence Application*, pages 29–30, 1991.
- David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, 2007.
- Ronan Collobert. Deep learning for efficient discriminative parsing. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 224–232. JMLR Workshop and Conference Proceedings, 2011.
- Kai Labusch, Preußischer Kulturbesitz, Clemens Neudecker, and David Zellhöfer. Bert for named entity recognition in contemporary and historical german. In *Proceedings of the 15th conference on natural language processing, Erlangen, Germany*, pages 8–11, 2019a.
- Khaled Shaalan. A Survey of Arabic Named Entity Recognition and Classification. *Computational Linguistics*, 40(2): 469–510, 06 2014. ISSN 0891-2017. doi:10.1162/COLI_a_00178. URL https://doi.org/10.1162/COLI_a_00178.
- Rodrigo Rafael Villarreal Goulart, Vera Lúcia Strube de Lima, and Clarissa Castellã Xavier. A systematic review of named entity recognition in biomedical texts. *Journal of the Brazilian Computer Society*, 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.
- Rahul Sharnagat. Named entity recognition: A literature survey. *Center For Indian Language Technology*, pages 1–27, 2014.
- 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.
- 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, Ziyue Zhu, and Yun Li. Nested named entity recognition: A survey. *ACM Trans. Knowl. Discov. Data*, 16(6), jul 2022. ISSN 1556-4681. doi:10.1145/3522593. URL <https://doi.org/10.1145/3522593>.
- 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.
- 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.
- Vishal Singh Roha, Naveen Saini, Sriparna Saha, and Jose G Moreno. Moo-cmnd+ 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.
- 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.
- Omar Al-Qawasmeh, Mohammad Al-Smadi, and Nisreen Fraihat. Arabic named entity disambiguation using linked open data. In *2016 7th International Conference on Information and Communication Systems (ICICS)*, pages 333–338. IEEE, 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. *International Journal on Artificial Intelligence Tools*, 29(07n08): 2040021, 2020b.
- Casimir George Borkowski. A system for automatic recognition of names of persons in newspaper texts. 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. URL <https://hal.archives-ouvertes.fr/hal-00682805>.
- Satoshi Sekine and Chikashi Nobata. Definition, dictionaries and tagger for extended named entity hierarchy. In *LREC'04*, 2004.
- 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, 2005a. ISSN 0004-3702. doi:<https://doi.org/10.1016/j.artint.2005.03.001>. URL <https://www.sciencedirect.com/science/article/pii/S0004370205000366>.
- Shaodian Zhang and Noémie Elhadad. Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. *Journal of Biomedical Informatics*, 46(6):1088–1098, 2013. ISSN 1532-0464. doi:<https://doi.org/10.1016/j.jbi.2013.08.004>. URL <https://www.sciencedirect.com/science/article/pii/S1532046413001196>. Special Section: Social Media Environments.
- Daniel Hanisch, Katrin Fundel, Heinz-Theodor Mevissen, Ralf Zimmer, and Juliane Fluck. Prominer: rule-based protein and gene entity recognition. *BMC bioinformatics*, 6(1):1–9, 2005.
- Alexandra Pomares Quimbaya, Alejandro Sierra Múnica, Rafael Andrés González Rivera, Julián Camilo Daza Rodríguez, Oscar Mauricio Muñoz Velandía, Angel Alberto García Peña, and Cyril Labbé. Named entity recognition over electronic health records through a combined dictionary-based approach. *Procedia Computer Science*, 100: 55–61, 2016.
- Yusuke Shinyama and Satoshi Sekine. Named entity discovery using comparable news articles. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 848–853, Geneva, Switzerland, aug 23–aug 27 2004a. COLING. URL <https://aclanthology.org/C04-1122>.

- Ludovic Bonnefoy, Patrice Bellot, and Michel Benoit. Mesure non-supervisée du degré d'appartenance d'une entité à un type. TALN 2011, juin 2011. URL https://www.lirmm.fr/TALN2011/PDF_court/Bonnefoy_taln11_submission_120.pdf.
- Yusuke Shinyama and Satoshi Sekine. Named entity discovery using comparable news articles. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 848–853, 2004b.
- David Nadeau, Peter D Turney, and Stan Matwin. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *Advances in Artificial Intelligence: 19th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2006, Québec City, Québec, Canada, June 7-9, 2006. Proceedings 19*, pages 266–277. Springer, 2006.
- 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, 2005b.
- 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*, 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.
- Sudha Morwal, Nusrat Jahan, and Deepti Chopra. Named entity recognition using hidden markov model (hmm). *International Journal on Natural Language Computing (IJNLC) Vol, 1*, 2012.
- Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. An algorithm that learns what's in a name. *Mach. Learn.*, 34(1–3):211–231, feb 1999. ISSN 0885-6125. doi:10.1023/A:1007558221122. URL <https://doi.org/10.1023/A:1007558221122>.
- Andrew Eliot Borthwick. *A maximum entropy approach to named entity recognition*. New York University, 1999.
- 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.
- Hideki Isozaki and Hideto Kazawa. Efficient support vector classifiers for named entity recognition. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.
- Takaki Makino, Yoshihiro Ohta, Jun'ichi Tsujii, et al. Tuning support vector machines for biomedical named entity recognition. In *ACL-02 workshop on NLP in the biomedical domain*, pages 1–8, 2002.
- Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *HLT-NAACL 2003 - Volume 4*, page 188–191, 2003.
- Burr Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pages 107–110, Geneva, Switzerland, August 28th and 29th 2004a. COLING. URL <https://aclanthology.org/W04-1221>.
- Shaojun Zhao. Named entity recognition in biomedical texts using an hmm model. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications (NLPBA/BioNLP)*, pages 87–90, 2004.
- Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. Nyu: Description of the mene named entity system as used in muc-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998*, 1998.
- 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.
- Burr Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications (NLPBA/BioNLP)*, pages 107–110, 2004b.
- 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.

- 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, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL <https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf>.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint*, 2016.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 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 Abdesslem Karaa. Named entity recognition using web document corpus. *arXiv preprint arXiv:1102.5728*, 2011.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12(null):2493–2537, nov 2011. ISSN 1532-4435.
- 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 *Proceedings of ACL*, pages 1064–1074, 2016.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, 2016.
- Sławomir Dadas. Combining neural and knowledge-based approaches to named entity recognition in polish. In *Artificial Intelligence and Soft Computing*, pages 39–50, 2019.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018a. Association for Computational Linguistics. doi:10.18653/v1/N18-1202. URL <https://aclanthology.org/N18-1202>.
- Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi:10.18653/v1/P17-1161. URL <https://aclanthology.org/P17-1161>.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018b. Association for Computational Linguistics. doi:10.18653/v1/N18-1202. URL <https://aclanthology.org/N18-1202>.
- 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, 2019.
- 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, mar 2020. doi:10.1016/j.physd.2019.132306. URL <https://doi.org/10.1016%2Fj.physd.2019.132306>.

- 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.
- Yue Zhang and Jie Yang. Chinese ner using lattice lstm. *arXiv preprint arXiv:1805.02023*, 2018.
- Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. *Advances in neural information processing systems*, 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 *Proceedings of the 57th annual meeting of the association for computational linguistics*, 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 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 *2008 The Eighth IAPR International Workshop on Document Analysis Systems*, pages 567–573. IEEE, 2008.
- Feifei Zhai, Saloni Potdar, Bing Xiang, and Bowen Zhou. Neural models for sequence chunking. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.
- OCDE. Ai language models. (352), 2023. doi:<https://doi.org/https://doi.org/10.1787/13d38f92-en>. URL <https://www.oecd-ilibrary.org/content/paper/13d38f92-en>.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- 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.
- 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. doi:10.18653/v1/W19-4312. URL <https://aclanthology.org/W19-4312>.
- Kai Labusch, Clemens Neudecker, and David Zellhöfer. Bert for named entity recognition in contemporary and historic german. In *KONVENS*, 2019b.
- Martin Riedl and Sebastian Padó. A named entity recognition shootout for German. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 120–125, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi:10.18653/v1/P18-2020. URL <https://aclanthology.org/P18-2020>.
- Rafika Boutalbi, Mira Ait-Saada, Anastasiia Iurshina, Steffen Staab, and Mohamed Nadif. Tensor-based graph modularity for text data clustering. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2227–2231, 2022.
- Mira Ait-Saada and Mohamed Nadif. Contextual word embeddings clustering through multiway analysis: A comparative study. In *International Symposium on Intelligent Data Analysis*, pages 1–14. Springer, 2023a.
- Mira Ait-Saada and Mohamed Nadif. Is anisotropy truly harmful? a case study on text clustering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1194–1203, 2023b.
- Mohamed Nadif and François Role. Unsupervised and self-supervised deep learning approaches for biomedical text mining. *Briefings in Bioinformatics*, 22(2):1592–1603, 2021.
- Jian Liu, Lei Gao, Sujie Guo, Rui Ding, Xin Huang, Long Ye, Qinghua Meng, Asef Nazari, and Dhananjay Thiruvady. A hybrid deep-learning approach for complex biochemical named entity recognition. *Knowledge-Based Systems*, 221:106958, 2021a.
- Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. Tener: adapting transformer encoder for named entity recognition. *arXiv preprint*, 2019.

- 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.
- 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, Online, November 2020. Association for Computational Linguistics. doi:10.18653/v1/2020.clinicalnlp-1.18. URL <https://aclanthology.org/2020.clinicalnlp-1.18>.
- Rahul Mehta and Vasudeva Varma. Llm-rm at semeval-2023 task 2: Multilingual complex ner using xlm-roberta. *arXiv preprint arXiv:2305.03300*, 2023.
- 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. *Advances in neural information processing systems*, 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. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL <http://jmlr.org/papers/v21/20-074.html>.
- 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*, 2023.
- Dhananjay Ashok and Zachary C Lipton. Promptner: Prompting for named entity recognition. *arXiv preprint arXiv:2305.15444*, 2023.
- 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 *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4047–4065, 2023.
- Francesco De Toni, Christopher Akiki, Javier De La Rosa, Clémentine Fourier, 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.
- 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 *International Conference on Artificial Intelligence and Statistics*, 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.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- Noëmi Aepli and Martin Volk. Reconstructing complete lemmas for incomplete german compounds. In *Language Processing and Knowledge in the Web: 25th International Conference, GSCL 2013, Darmstadt, Germany, September 25-27, 2013. Proceedings*, pages 1–13. Springer, 2013.
- 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*, 2021a.
- Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv preprint*, 2017.
- Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint*, 2016.
- Ameya Daigavane, Balaraman Ravindran, and Gaurav Aggarwal. Understanding convolutions on graphs. *Distill*, 6(9): e32, 2021.
- Alberto Cetoli, Stefano Bragaglia, Andrew D O’Harney, and Marc Sloan. Graph convolutional networks for named entity recognition. *arXiv preprint*, 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. *arXiv preprint*, 2019b.
- Ismail Harrando and Raphaël Troncy. Named entity recognition as graph classification. In *The Semantic Web: ESWC 2021 Satellite Events: Virtual Event, June 6–10, 2021, Revised Selected Papers 18*, pages 103–108. Springer, 2021.
- Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, et al. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.
- Manali Shaha and Meenakshi Pawar. Transfer learning for image classification. In *2018 second international conference on electronics, communication and aerospace technology (ICECA)*, pages 656–660. IEEE, 2018.
- Dong Wang and Thomas Fang Zheng. Transfer learning for speech and language processing. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (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 *international conference on big data (Big Data)*, pages 1367–1376, 2018.
- Ji Young Lee, Franck Dernoncourt, and Peter Szolovits. Transfer learning for named-entity recognition with neural networks. *arXiv preprint*, 2017.
- 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.
- Zihan Liu, Feijun Jiang, Yuxiang Hu, Chen Shi, and Pascale Fung. Ner-bert: A pre-trained model for low-resource entity tagging. *arXiv preprint*, 2021b.
- 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. *arXiv preprint*, 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 2022 Workshop on Synthetic Data for Empowering ML Research*.

- 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. *arXiv preprint*, 2021.
- 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. *arXiv preprint*, 2018.
- Yanyao Shen, Hyokun Yun, Zachary C Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep active learning for named entity recognition. *arXiv preprint*, 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*, (ahead-of-print), 2022.
- Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.
- Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. Few-shot classification in named entity recognition task. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 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. *arXiv preprint*, 2020.
- Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.
- Yi Yang and Arzoo Katiyar. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. *arXiv preprint*, 2020.
- Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. Template-based named entity recognition using bart. *arXiv preprint*, 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*, 2020.
- 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2050–2058, 2017.
- Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. *arXiv preprint*, 2019.
- Rasha Obeidat, Xiaoli Fern, Hamed Shahbazi, and Prasad Tadepalli. Description-based zero-shot fine-grained entity typing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 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. *arXiv preprint*, 2022.
- 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 *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, 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.
- Ted Kwartler. *The OpenNLP Project*, pages 237–269. 05 2017. ISBN 9781119282013. doi:10.1002/9781119282105.ch8.
- 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 *Conference of the North American chapter of ACL*, 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. *arXiv preprint*, 2022.
- 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-Computer Modeling in Engineering & Sciences*, 129(1), 2021b.
- Ralph Grishman and Beth Sundheim. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996b. URL <https://aclanthology.org/C96-1079>.
- George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC’04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA). URL <http://www.lrec-conf.org/proceedings/lrec2004/pdf/5.pdf>.
- Erik F Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint*, 2003.
- 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 *Proceedings of the Australasian Language Technology Association Workshop 2015*, 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.