Few-shot Named Entity Recognition: Definition, Taxonomy and Research Directions

VINCENZO MOSCATO, MARCO POSTIGLIONE, and GIANCARLO SPERLÍ, University of Naples Federico II, Italy

Recent years have seen an exponential growth (+98% in 2022 w.r.t. the previous year) of the number of research articles in the *few-shot learning* field, which aims at training machine learning models with extremely limited available data. The research interest toward few-shot learning systems for Named Entity Recognition (NER) is thus at the same time increasing. NER consists in identifying mentions of pre-defined entities from unstructured text, and serves as a fundamental step in many downstream tasks, such as the construction of Knowledge Graphs, or Question Answering. The need for a NER system able to be trained with few-annotated examples comes in all its urgency in domains where the annotation process requires time, knowledge and expertise (e.g., healthcare, finance, legal), and in low-resource languages. In this survey, starting from a clear definition and description of the few-shot NER (FS-NER) problem, we take stock of the current state-of-the-art and propose a taxonomy which divides algorithms in two macro-categories according to the underlying mechanisms: model-centric and data-centric. For each category, we line-up works as a story to show how the field is moving toward new research directions. Eventually, techniques, limitations, and key aspects are deeply analyzed to facilitate future studies.

CCS Concepts: • **Computing methodologies** → **Natural language processing**; **Information extraction**; **Learning paradigms;**

Additional Key Words and Phrases: Few-shot learning, Named Entity Recognition

ACM Reference format:
Vincenzo Moscato, Marco Postiglione, and Giancarlo Sperlí. 2023. Few-shot Named Entity Recognition: Definition, Taxonomy and Research Directions. *ACM Trans. Intell. Syst. Technol.* 14, 5, Article 94 (October 2023), 46 pages.
https://doi.org/10.1145/3609483

1 INTRODUCTION

Despite the ever-increasing amount of data, the difficulty or even impossibility to share private information (for example, in healthcare), and the manual efforts required to build high-quality datasets, impose the necessity to find alternative ways to training machine learning models with massive data [86, 101, 153, 181].

Named Entity Recognition (**NER**) is the task of identifying mentions of entities from unstructured text and classifying their type (e.g., person, organization, disease, drug). It is the fundamental task for many downstream applications — such as question answering [3], dialogue systems [16],

This project has been funded by PNRR MUR project PE0000013-FAIR.
Authors' address: V. Moscato, M. Postiglione, and G. Sperlí, University of Naples Federico II, Via Claudio 21, Naples, 80125, Italy; emails: {vincenzo.moscato, marco.postiglione, giancarlo.sperli}@unina.it.

2157-6904/2023/10-ART94 $15.00
https://doi.org/10.1145/3609483

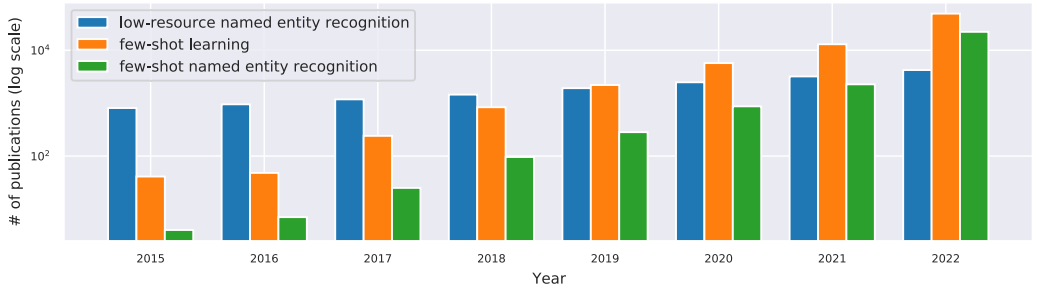

Fig. 1. Evolution of the number of total publications whose title, abstract, and/or keywords refer to few-shot learning during the last years. Data retrieved from Google Scholar[1] (Feb 11th, 2023) by using the queries indicated in the legend.

and knowledge graph construction [2] — and, as a consequence, poor performance could degrade the quality of the overall system. The recent advancements in **Natural Language Processing** (**NLP**) [11, 30, 114] have enabled the achievement of remarkable performance improvements when large labeled training corpora are available.

However, real-world applications — which are not limited to the English language, where most of publicly available datasets reside — would often require the annotation of large amounts of documents to reach comparable results to the current literature, and this is highly expensive, time-consuming and prone to human errors, especially when domain knowledge is required to produce high-quality results. Furthermore, *inconsistencies* between annotation schemes of different datasets lead to models which, having being trained on a given dataset, do not work properly on another even if it refers to the same context [75].

To tackle these issues, a widespread interest toward few-shot learning has emerged [140, 142, 144] and various books and surveys on this theme have been published [59, 124, 141, 156, 157, 170, 178]. *Meta learning* and *transfer learning*, for example, are training paradigms which allow to train models so as parameters can be easily adapted to new tasks, and to initialize models with parameters from other domains or languages, respectively. Anyway, most of the works related to the few-shot research area are *data-centric*, i.e., they try to maximize the value derivable from data. For example, *active learning* aims at selecting the most informative examples from an unsupervised data source to be annotated, so as to get the best possible result from model training; on the other hand, *distant supervision* and *self learning* leverage unsupervised data sources to increase the size of the training set by using heuristics or the model itself to annotate examples.

Reviewing the rich and articulate state-of-the-art in the few-shot learning field, we found that extensions of current few-shot methodologies and their applications to the NER task are currently starting to exponentially increase (refer to Figure 1 for an overview of the number of published articles in recent years). The need to explore few-shot learning in the context of NER emerges in the recent survey by Li et al. [76] as a consequence of the ineffectiveness of transfer learning, since due to the variations in language traits and annotated texts, consequently a model developed for one dataset may not perform well on texts from other datasets. Furthermore, the variety of theoretical methods to improve the model generalization ability for few-shot settings and the increasing volume of research impose the need to take stock of the current state-of-the-art to understand the value inherent in each method. However, to the best of our knowledge, at the current state-of-the-art there is only one benchmarking study from Huang et al. [57] that investigates three common schemes for few-shot learning extended and applied to NER: meta-learning,

[1] https://scholar.google.com/

supervised pre-training with noisy data extracted from web sources and self-training with unlabeled samples. While the above-mentioned benchmarking study has an extremely high value to serve as a baseline for future works, it does not capture the wide and dynamic landscape of few-shot NER works.

In this survey, we start from a precise definition of ***Few-Shot Named Entity Recognition*** (**FS-NER**) and its contextualization in the current research scenario. Then, we propose a taxonomy to divide works into two macro-categories: model-centric and data-centric. The former focuses on model architectures, i.e., how to build and train models so that they can achieve high performance in few-shot settings; the latter focuses on data operations allowing to improve or increment the size of the available training corpora. In each sub-category, works are described in chronological order so as to provide insights on how later techniques address issues from earlier methods. Thus, we suggest promising directions for the further development of this research field.

The remainder of this work is structured as follows: Section 2 provides the definition of NER and the few-shot setting we are considering; Section 3 illustrates the algorithm taxonomy by first describing the key idea at the foundation of each category and then going more deeply to practical implementations in literature; Section 4 provides a benchmark study of 7 well-known few-shot learning methods for NER, while Section 5 illustrates final considerations and suggests paths for future work.

2 FEW-SHOT NER: WHAT, WHY, WHERE, AND HOW

In this Section, we use some questions from the Kipling method[2] to provide an overview of FS-NER. Specifically, we provide answers for questions listed as follows:

- *What? (Problem definition)* — in Section 2.1 we formalize the few-shot NER problem in the context of machine learning and few-shot learning sub-field. Contextually, we will outline differences making few-shot NER a more challenging task worth of an in-depth analysis.
- *Why? (The need for FS-NER)* — we will provide a clear and concise example, which shows the reasons behind the hype toward few-shot NER in Section 2.2.
- *Where? (Applications)* — in Section 2.3 we will describe real-world scenarios where FS-NER methodologies are needed.
- *How? (Taxonomy)* — we propose a taxonomy to categorize works on FS-NER, which will be described and analyzed in Section 3 and Section 4.

2.1 Problem Definition

Few-Shot NER (**FS-NER**) is a sub-field of **Few-Shot Learning** (**FSL**), which in turn can be considered as a sub-area in machine learning. We will first describe foundation definitions and then we will provide a formal definition of FS-NER. Contextually, we will discuss relatedness and differences with other NLP tasks.

Definition 1 (Machine Learning [95, 97]). A computer program is said to learn from experience E with respect to some classes of task T and performance measure P if its performance can improve with E on T measured by P.

Based on Definition 1, NER can be identified as the task T and E is a corpus of sentences annotated with entity mentions, so that a performance measure P (usually $Precision$, $Recall$ and/or $F1$) of the machine learning model can be improved. Formally, NER is defined as in Definition 2.

[2] https://projectofhow.com/methods/the-kipling-method/

Definition 2 (Named Entity Recognition [76]). Given a sequence of tokens (i.e., a sentence) $s = [t_1, t_2, \ldots, t_N]$, NER outputs a list of tuples $[I_s, I_e, t]$ representing named entities mentioned in s. Here, $I_s \in [1, N]$ and $I_e \in [1, N]$ are the indexes of start and end characters of the named entity mention, while t is the entity type.

It should be noted that the definition provided above is restricted to continuous spans, which represents a frequent occurrence. However, during the application of NER to real-world tasks, named entities that are overlapping or situated in discontinuous text spans may arise. These scenarios have been extensively investigated in various studies [37, 98, 154].

As a result of its peculiar characteristics, NER is a token-level classification task, where models make their predictions token by token. It may be considered as the parallel task of image segmentation in computer vision, where classification is on a pixel-by-pixel basis rather then relying on the whole image. As a consequence, many approaches and model architectures proposed to deal with few-shot challenges in text classification are not directly applicable to NER, thus requiring further efforts, extensions, and/or adjustments.

Machine learning models usually need for big datasets with supervised information (a.k.a. ground-truth) to be effectively trained. The sub-area of FSL has the aim at obtaining good performance when only a small set of supervised data is available. Formally, it can be defined as in Definition 3.

Definition 3 (Few-Shot Learning (FSL) [157]). FSL is a type of machine learning problems (specified by E, T, ansd P), where E contains only a limited number of examples with supervised information for the target T.

In concrete terms, FSL aims at learning a classifier able to predict a label y for each input x (e.g., image classification [81], text classification [143]). In the light of the definitions discussed above, we can provide a clear and concise definition for FS-NER as described in Definition 4.

Definition 4 (Few-Shot Named Entity Recognition (FS-NER)). FS-NER is a sub-area of FSL where the machine learning problem specified by E, T, and P is not only constrained by E containing a limited number of examples, but also by T being a NER task.

The substantial difference in FS-NER is that each input sentence $\mathbf{s} = [t_1, t_2, \ldots, t_N]$ has to be associated to multiple labels $\mathbf{y} = [y_1, y_2, \ldots, y_N]$, one for each token $t_i \in \mathbf{s}$, by the few-shot model. In addition to the formal task difference, it is important to note that two tokens $t_i \in \mathbf{s}$ ($i \in [1, N]$) and $t_j \in \mathbf{s}$, ($j \in [1, N]$) are not independent but semantically related. As a consequence, NER works usually leverage their relatedness to build effective machine learning models, but this also means that many FSL works may be unsuitable for FS-NER. However, several efforts have been made during this years to adapt and extend FSL methodologies or to borrow key ideas. In this work, we will provide an in-depth analysis of how FSL works have been adapted to FS-NER.

2.2 The Need for FS-NER

Deep neural networks and Transformer architectures represent the foundation of contemporary NER techniques, which require little to no feature engineering to achieve cutting-edge benchmark performances. However, since they typically require enormous hand-labeled training sets, these gains are frequently challenging to apply in real-world situations. In FS-NER, the number of available training examples is small, thus impacting the reliability of the resulting NER model which, as a consequence, usually overfits data [157].

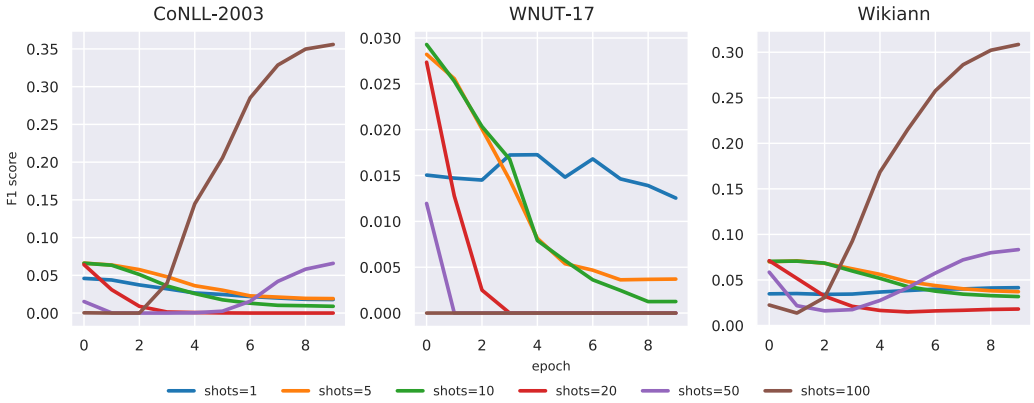

Fig. 2. F1 score trends of few-shot NER models as the training epochs progress.

We empirically show this in Figure 2, where we report trends of F1 scores as the NER training epochs increase on validation sets from three widely used benchmark datasets: CoNLL-2003 [148], WNUT-17 [28], WikiANN (en) [107].[3]

We can observe that performance has a decreasing trend when the number of shots is too small — *shots* <= 20 on CoNLL-2003 and WikiANN, *shots* <= 100 on WNUT-17 data —, meaning that the model is overfitting on its training set. Moreover, even when the F1 score increases as the training progresses, it reaches low values due to the inability of the resulting model to recall all the ground-truth entity mentions contained in the test set. From these two challenges — i.e., reducing overfitting and improving performance of NER models when only few labeled examples are available — arises the need for FS-NER methodologies.

2.3 Applications

In broad terms, FS-NER is needed in every application scenario affected by the rareness of training samples, which is usually due to:

— *Scarcity of resources* — there are many situations where the current state-of-the-art does not provide an available dataset to train our models. For example, most of the current NLP research is focused on 20 out of the 7,000 languages spread all over the world [90], which are thus inevitably disadvantaged with the unavailability of data sources.

— *Difficult data sharing* — depending on the application domain, the possibility to share data and build training corpora may be a challenge. For example, owners of healthcare data, i.e., patients, may often hesitate to share their (sensitive) information for research or commercial purpose, despite the potential impact of such data.

— *Annotation costs* — the manual labelling process of NER data is expensive. It usually requires several annotators and a standard protocol to minimize annotation conflicts. To obtain high-quality training corpora, not only does a significant amount of time have to be devoted to the annotation process, but domain knowledge is also often needed (e.g., healthcare, finance), thus increasing the overall costs.

[3]For the sake of reproducibility, we report here details for this experiment. Models and datasets have been downloaded from the HuggingFace repository [162]. For each dataset, we fine-tuned a BERT base (cased) network [30] for ten epochs with *learning rate* set to 2e-5 and *weight decay* to 0.01. All the other hyper-parameters are set to default. Reported F1 scores have been computed on validation sets. Training sets of few-shot experiments are obtained by randomly sampling from original training corpora.

Hence, many real-world applications involve FS-NER. However, current state-of-the-art does not offer many successful use cases yet — methods are usually tested on benchmark datasets. Wang et al. [159] experiment their few-shot method on a private corpus of 1,600 de-identified EHRs from cardiology, respiratory, neurology and gastroenterology deparments; Ni et al. [102] test their cross-lingual FS-NER approach on a custom multi-lingual dataset with over 50 entity types annotated to build cognitive question answering applications on top of the FS-NER system.

2.4 Taxonomy

In this section, we propose a taxonomy to categorize existing state-of-the-art FS-NER techniques, as shown in Figure 3. The first layer of the taxonomy is based on whether the methodology is focused on model architecture (*model-centric*) or data (*data-centric*), respectively. Input data sources and methodological flows change dependently on the technique, but we summarized high-level details in Figure 4.

Model-centric Methods. In this setting, model architectures are designed to make the most of the few available training samples. *Transfer learning* approaches leverage model weights learned in another domain or language. Differently from transfer learning, which leverage knowledge from the same task but a different domain, *Meta learning* aims at building models able to quickly adapt to new tasks without the need to be re-trained from scratch.

Data-centric Methods. The shift from model-centric to data-centric AI[4] is ongoing and increasingly widespread. This can be justified by the fact that the astonishing improvements brought by deep learning models on the state-of-the-art of several AI tasks has led the research community to find ever more better models, but now that a performance plateau has been reached, efforts are being made to deal with the other important aspect of AI systems: data. In the context of FS-NER, we identified four common methods to deal with the lack of data:

— *Data Augmentation* techniques leverage not only training samples but also external sources and unannotated data (when available) to increase the training corpus size
— *Active Learning* aims at selecting the most informative samples to be annotated from an unlabeled corpus in order to optimize the tradeoff between performance and annotation costs.
— *Distant Supervision* consists in leveraging external data sources and heuristics to provide "weak" labels to data from an unlabeled corpus.
— *Self Learning* approaches use the model itself to provide a label to data from an unlabeled corpus.

In the following Sections, we are describing in details the discussed methodologies.

3 MODEL-CENTRIC METHODS

In this section, we review model-centric FS-NER methods by separating them into two sections, as outlined in Figure 3. Hence, we line up methods as a story and summarize their key characteristics and discuss similarities and differences under each category as well as limitations that have not been addressed yet.

3.1 Transfer Learning

In all the fields of Machine Learning, *Transfer Learning* is the standard approach to deal with the lack of data. The knowledge — i.e., their learned parameters — of models trained on huge datasets

[4] https://datacentricai.org

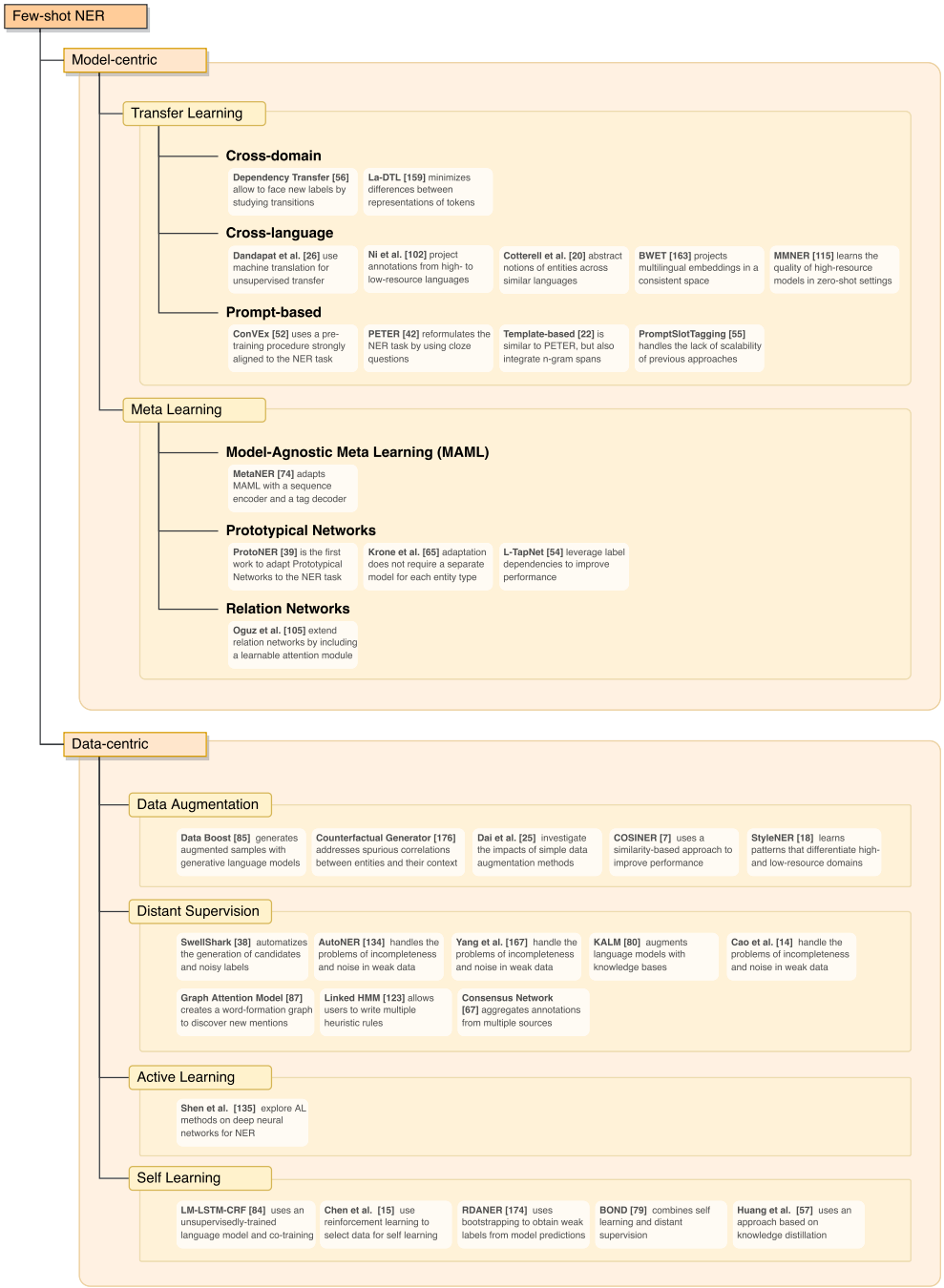

Fig. 3. The proposed taxonomy to summarize FS-NER techniques. We categorize them into two groups, model-centric and data-centric, depending on whether the focus of the methodology is on model architectures or input data sources, respectively. For each group, we further categorize methods into several subgroups with a hierarchical approach.

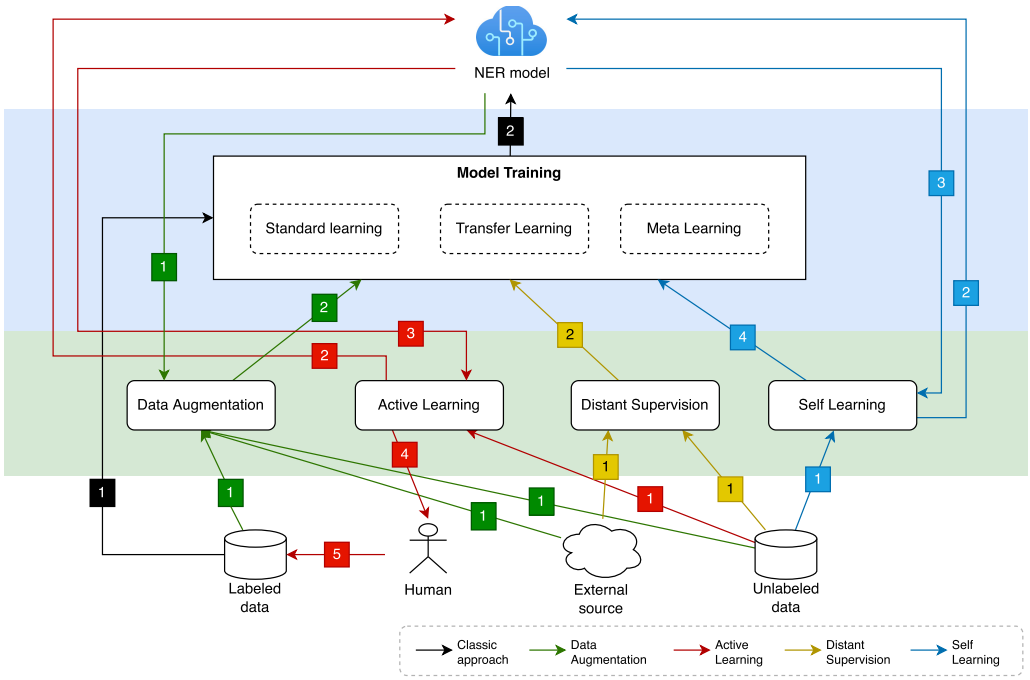

Fig. 4. High-level methodological flows of FS-NER methods. The standard training approach (black arrows) receives labeled data as inputs; data augmentation techniques (green arrows) leverage labeled and unlabeled data, external sources and/or even the model itself to augment the size of the training corpus; active learning (red arrows) selects a subset of unlabeled data to be labeled by a human annotator by leveraging model predictions; distant supervision (yellow arrows) uses external sources and heuristics, while self-learning (blue arrows) uses model predictions to provide annotations to unlabeled data. Models are often trained relying on transfer learning or meta learning approaches. Numerical values are assigned to each data flow to indicate the sequence of operations.

is "adapted" with new training iterations to make it possible for the model to perform well with a target domain where there is a lack of resources. Current Transfer Learning methods for NER are mainly based on deep neural networks and Transformer architectures and usually leverage *feature representation transfer* [109], which makes the model learn to map inputs from different domains in a close feature space, and *parameter transfer* [168], which makes the target model parameters close to those of the source model. We divide transfer learning approaches for FS-NER in three categories: *cross-domain*, *cross-lingual*, *fine-tuning*. In the following, we describe them in detail.

3.1.1 Cross-domain Transfer. In cross-domain Transfer Learning, we aim at transfering the knowledge from a *source* specialty (e.g., Electronic Health Records, a.k.a. EHRs, from the department of cardiology) to a *target* specialty (e.g., EHRs from the department of orthopaedics). Figure 5 presents an example of inputs that can be used to train a cross-domain transfer learning system. The goal of this system is to enhance performance in a distinct target domain, which may feature a dissimilar set of entity types. The corresponding output is also depicted in the figure. In the following, we describe the current state-of-the-art methods.

Dependency Transfer & **Pair-wise Embedding** [56] is a CRF-based method that integrates prior experience of token similarities and label dependencies. In particular, this work handles two challenges when learning emission and transition scores of CRFs: (1) the infeasibility to

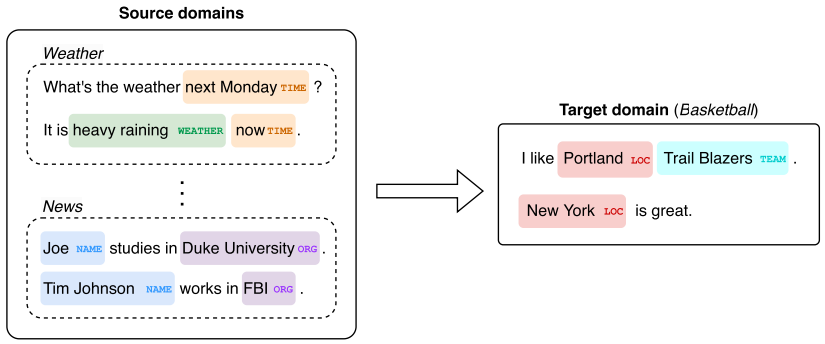

Fig. 5. Example of inputs for cross-domain transfer learning. Knowledge is being transferred to a new domain, which may even have a different set of entity types.

learn transition scores on the few in-domain labeled data or from source domain data due to discrepancies in label sets, and (2) the difficulty in calculating emission scores due to different meanings of words based on their contexts.

To handle the first problem, the *dependency transfer* mechanism is proposed. Formally, label dependencies for the transition scorer can be computed as the transition probability between two labels $f_T(y_{i-1}, y_i) = p(y_i|y_{i-1})$ which are usually stored in a transition matrix $\mathbf{M}^{N\times N}$, where N is the number of labels and $m_{l_1,l_2} \in \mathbf{M}$ corresponds to $p(y_i = l_2|y_{i-1} = l_1)$. Given that in few-shot contexts we could face a new label which is not available in source domains, only three abstract labels are used: O, B, I, and transition scores from B and I labels to another are computed by differentiating transitions to the same B (sB), different B (dB), same I (sI), different I (dI). The transition matrix will thus have 3 rows (O, B, I) and 5 columns (O, sB, sI, dB, dI). The second challenge is then faced by using a *pair-wise embedder*, which pairs the query sentence with all support sentences and uses a Transformer [150] to get paired representations that are different for each support sample.

Note that in this work the model is first trained on a set of source domains and then directly used with a set of unseen target domains without fine-tuning. The transfer of knowledge thus happens with the above-described methodology and not by re-training models.

Label-aware Double Transfer Learning (La-DTL)[5] [159] exploits Maximum Mean Discrepancy (MMD) [46] to reduce the discrepancy between feature representations of tokens with the same label coming from different sources. In particular, the goal is to improve performance on the target domain $\mathcal{D}_t = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N^t}$ by leveraging knowledge from the source domain $\mathcal{D}_s = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N^s}$, with $N^s >> N^t$. Each input sentence is transformed into a sequence of embedding vectors and then fed into a BiLSTM network which encodes contextual information into a fixed-length vector and is shared between source and target models. The reason behind the choice of sharing the BiLSTM layers across domains lies in the poor ability of LSTM networks to generalize well without seeing enough data [173]. Label aware MMD is then computed to reduce the feature representation discrepancy between the two domains. It is a parametric test statistic to measure the distance between the kernel mean embeddings of two distributions, which is computed between the hidden representations from two domains with the same ground truth label y and minimized during training. Hidden representations are then fed to domain-specific CRF layers to predict the label sequence. CFR layers are not shared across domains to enhance target domain performance.

[5]La-DTL [159] code: https://github.com/felixwzh/La-DTL

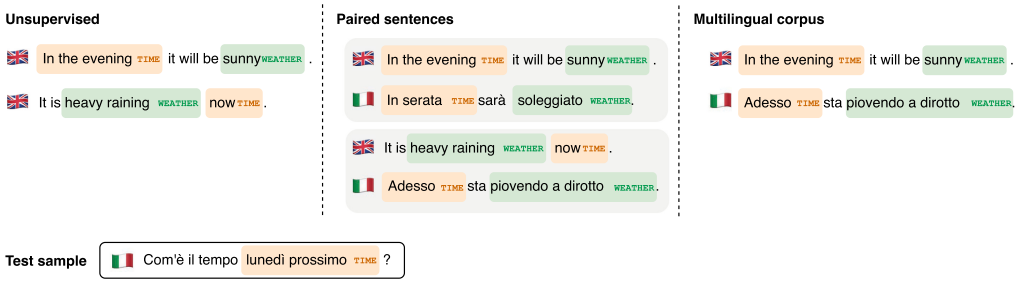

Fig. 6. Training corpora for cross-language transfer. In *unsupervised* scenarios, we only have access to high-resource language data, but need to extract entity mentions for a low-resource language. The ideal scenario is when we have both the high-resource language annotation and the corresponding low-resource language annotation available (*paired sentences*). In some cases, we can use a *multilingual* corpus that contains both high- and low-resource language samples.

3.1.2 Cross-language Transfer. A high number of FS-NER works focus on leveraging cross-lingual information to improve the model performance. Based on the availability of data, many transfer learning scenarios are possible, as shown in Figure 6. Most of the approaches rely on a projection-based transfer scheme [63, 102, 155, 169]: one side of bitext is annotated with a tagger for a high-resource language and then the annotation is projected over the bilingual alignments obtained through unsupervised learning [104]. Projected annotations are then used as weak supervision to train the tagger in the target language. However, paired sentences are not always readily available. In some cases, all that is available are individual sentences in a high-resource language, or a multilingual corpus that includes samples from multiple languages. In the following, we explore methods for transferring knowledge across languages in FS-NER tasks.

Dandapat et al. [26]. The extreme scenario in cross-lingual transfer learning is when no labeled data is available in the target language (*unsupervised transfer*) [115, 163]. Dandapat et al [26] address the problem of unavailability of parallel data between the low-resource language (Hindi) and English with machine translation systems (i.e., Google Translate[6]). In particular, cross-lingual features are extracted from a resource-rich language by (1) aligning the low-resource language sentence and its translation, (2) applying a good performing model to the English sentence and (3) for each word in the low-resource sentence, (3.1) English word(s) that map to the source word are found thanks to the alignment function and (3.2) the feature vector of the mapped word, initialized to all zeros, is updated by adding 1 if its label matches with any of the labels in the feature vector. The quality of this approach strongly depends on the quality of the Machine Translation system.

Ni et al. [102] propose an *annotation projection* approach which receives parallel sentence pairs, where the English sentence serves as the source and the corresponding sentence in the low-resource language serves as its translation. Given a sentence pair $(\mathbf{x}, \mathbf{y})$, where $\mathbf{x} = (x_1, x_2, \ldots, x_s)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_t)$, the projection procedure requires two steps: (1) first, an English NER model is applied to the English sentence $\mathbf{x}$ to obtain a set of NER tags $\mathbf{l} = (l_1, l_2, \ldots, l_s)$; (2) then, NER tags are projected to the target sequence $\mathbf{y}$ by leveraging alignment information. Note that this work considers "parallel" sentences and generates a weak set of projected data, which is then filtered to select good-quality samples to improve performance. While this annotation projection approach is *data-centric* (it may be reported under the *Distant Supervision* category), authors also

[6] https://translate.google.com

propose a *representation projection* approach which allows to use the same NER model for both English and the target language by providing "universal" word embeddings as inputs. In particular, similarly to Mikolov et al. [92], given a target language f, a dictionary containing English and target-language word pairs $\{(x_i, y_i, w_i)\}_{i=1,\ldots,n}$ is generated, where x_i and y_i are the English and target-language word, respectively, and $w_i = P(x_i|y_i)$ is a weight representing the relative frequency of x_i given y_i. The training objective is to find a linear mapping $\mathbf{M}_{f\rightarrow e}$ which projects target-language words to english:

$$\mathbf{M}_{f\rightarrow e} = \arg\min_{\mathbf{M}} \sum_{i=1}^{n} w_i ||\mathbf{u}_i - \mathbf{M}\mathbf{v}_i||^2, \tag{1}$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are the embedded representations for the English word x_i and the target-language word y_i, respectively; weights w_i allow to provide a higher importance to more frequent pairs.

The problem of this approach is that the bitext assumption is often not present in the case of low-resource languages [20, 26].

Cotterell et al. [20] allow the notion of named entities to be shared across "genetically" similar languages by using a character-level neural CRF to abstract the notion of a named entity across *similar* languages. Differently from projection-based approaches, this strategy does not require a bi-text assumption. First, given a language label l, a language-specific CRF $p_\theta(\mathbf{y}|\mathbf{x}, l)$ is created. Transition between tags and the character-level neural network are shared between languages to allow knowledge transfer. Given a low-resource target language τ and a high-resource language σ, the training objective is:

$$\mathcal{L}(\theta) = \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}_\tau} logp_\theta(\mathbf{y}|\mathbf{x}, \tau) + \mu \cdot \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}_\sigma} logp_\theta(\mathbf{y}|\mathbf{x}, \sigma), \tag{2}$$

where μ is a tradeoff parameter, $\mathcal{D}_\tau$ is the target-language dataset, $\mathcal{D}_\sigma$ is the source-language dataset. If we have a set of m high-resource languages $\{\sigma_i\}_{i=1}^m$ available, we can add a summand to the set of high-resource language datasets $\mathcal{D}_{\sigma_i}$ used. Experiments prove that while this transfer approach has little or no effect on big datasets, it is extremely useful in few-shot scenarios (e.g., from $F1 = 0.49$ to $F1 = 0.76$ on a 100-shot Galician dataset with knowledge transfer from Spanish).

Bilingual Word Embedding Translation (BWET)[7] [163] uses **Bilingual Word Embeddings (BWEs)** to project two sets of embeddings into a consistent space with a small dictionary [92, 139] or in an entirely unsupervised manner (i.e., no labeled data is available for the low-resource language) by using adversarial training [179]. Specifically, four methodological steps are performed:

(1) Separate word embeddings are trained for each monolingual corpus.
(2) Word embeddings are then projected into a shared latent space with BWEs. Assuming we have a dictionary $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^D$, where x_i and y_i are embeddings of a word pair, a mapping matrix W is computed by minimizing the following:

$$\arg\min_{\mathbf{W}} \sum_{i=1}^{D} ||W\mathbf{x}_i - \mathbf{y}_i||^2, \tag{3}$$

which is equivalent to Equation (1) but without the notion of weights.
(3) Each word in the low-resource language is translated by finding its nearest neighbor in the shared latent space. **Cross-domain similarity local scaling** (**CSLS**) metric [66] is employed to address the *hubness* problem of the shared latent space [31] (i.e., mapped words in

[7]BWET [163] source code: https://github.com/thespectrewithin/cross-lingual_NER

the shared space may be near to many items that are "universal" neighbors of a large number of different mapped samples):

$$CSLS(\mathbf{x}_i, \mathbf{y}_j) = 2cos(\mathbf{x}_i, \mathbf{y}_j) - r_T(\mathbf{x}_i) - r_S(\mathbf{y_j}), \tag{4}$$

where $r_T(\mathbf{x_i}) = \frac{1}{|\mathcal{N}_T(\mathbf{x_i})|} \sum_{\mathbf{y_t} \in \mathcal{N}_T(\mathbf{x_i})} cos(\mathbf{x_i}, \mathbf{y_t})$ is the mean cosine similarity between $\mathbf{x_i}$ and its neighborhood $\mathcal{N}_T(\mathbf{x_i})$. The target word $\mathbf{y}_j$ maximizing the CSLS value is then selected as the proper translation.

(4) Translated words are used along with tags from the high-resource language corpus to train a NER model.

Massively Multilingual Transfer for NER (MMNER)[8] source code: https://github.com/afshinrahimi/mmner. [115] assumes to have a collection of H models trained in a high-resource setting from a different language from our target task. Showing that simply choosing one of the models or performing majority voting are inaccurate choices, the authors develop a generative model to learn the quality of models in the setting of zero annotations available in the target language, and use few annotations (when available) to find the posterior for parameters of a Bayesian inference model. Specifically, given transfer models predictions y_{ij}, where $i \in \{1, 2, \ldots, N\}$ is an instance and $j \in \{1, 2, \ldots, H\}$ denotes one of the H available transfer models, the generative process assumes the true label $z_i \in \{1, 2, \ldots, K\}$ being corrupted by each transfer model when producing predictions. Models assigning a high probability to the correct label are considered more reliable with respect to that label. For inference, mean-field variational Bayes [62] is employed.

3.1.3 Prompt-based Transfer. Recent work in NLP has demonstrated the impressive gains obtainable with the pre-training and fine-tuning approach, especially when applied to transformer language models [150]. During the pre-training phase, a large unsupervised dataset is used to train the language model to find informative representations of inputs which can then be used to solve downstream tasks. Hofer et al. [53] show that pre-training on domain-specific corpora and reducing out-of-vocabulary words can significantly improve performance of NER models in few-shot settings. The number of publicly available pre-trained models in a variety of domains and languages is high and constantly increasing.[9] Just to mention one example, *BioBERT* [70] pretrains a BERT-based language model on PubMed abstracts and PMC full-text articles to apply the advancements of transformers for biomedical text understanding. However, how to replicate these contributions in low-resource languages, where also unsupervised text data is difficult to obtain, is an open challenge. Bondarenko et al. [10] fine-tune a BERT language model pre-trained on russian data (RuBERT) to adapt it to NER and Relation Extraction. Schneider et al. [129] transfer learned the information encoded in a multilingual BERT model to a corpora of clinical narratives and biomedical scientific articles in Brazilian Portuguese. Reimers et al. [119] propose a knowledge distillation based approach to extend existing sentence embeddings models to new languages.

Recent work leverages *prompts* to exploit the knowledge acquired by such architectures during the pre-training phase by re-phrasing the task to a masked language modeling task which is closer to the target NER task. Figure 7 shows the workflow of PromptSlotTagging [55] as an example. In the following, we describe methods that propose the use of prompts to improve FS-NER performance.

Conversational Value Extractor (ConVEx) [52] is an efficient pre-training and fine-tuning approach which has its foundation in the fact that a stronger alignent between a pretraining task

[8] MMNER [115].
[9] https://huggingface.co/models

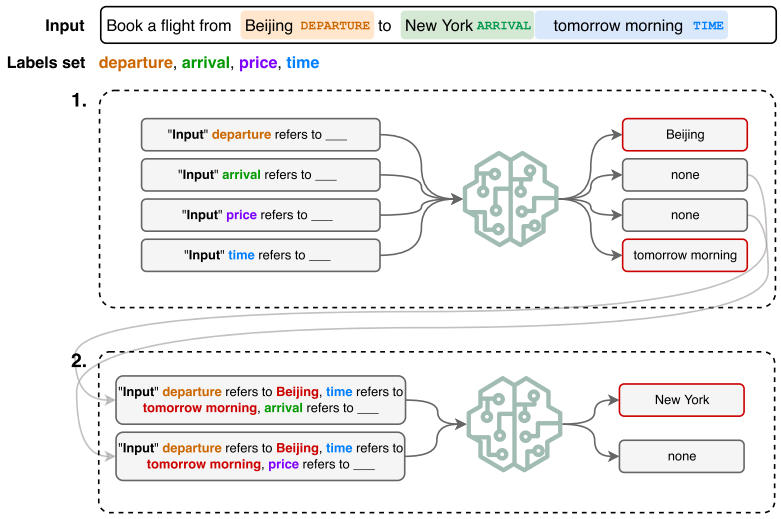

Fig. 7. PromptSlotTagging model. In the first phase, the input sentence is embedded with inverse prompts and decoded by the language model. In the second phase, predictions are iteratively refined by reinforcing prompts with previously-predicted values.

and an end task can yield performance gains [44, 73]. A *pairwise cloze* pre-training procedure is proposed, which is more closely related to the target slot-labeling task and facilitates training all the necessary layers for slot-labeling, so these can be fine-tuned rather than learned from scratch. Specifically, the model receives a *template* sentence along with the input sentence: a *keyphrase* (which is in common between the two sentences) in the template sentence is masked out, and the model has to predict which tokens in the input sentence constitute the keyphrase.

Pattern-Exploiting Training for NER (PETER) [42] extends ***Pattern-Exploiting Training (PET)*** [127], a prompt-based method originally designed for sentence classification. PETER reformulates the fine-tuning task by using cloze-questions to provide task descriptions which enable the model to leverage the knowledge it acquired during the pre-training phase. In particular, given a sequence of tokens $\mathbf{x}$, PETER follows the procedure described below:

- $|\mathbf{x}|$ new input examples are generated, one per token in the input sentence $t \in \mathbf{x}$. The generation of new samples is based on a *pattern* $P(\mathbf{x})$ which manipulates them with the structure: "$\mathbf{x}$. In the sentence above, the word t refers to a *[entityType]* entity".
- A language model is trained to assign a binary label to transformed samples indicating its truthfulness.
- Unlabeled data may be employed to generate a soft-labeled dataset from predictions of the previously-trained model(s). This can be thus considered as an hybrid approach which also uses self-learning (see Section 4.4 to handle the FS-NER problem.

Template-based NER[10] [22] approach is similar to PETER, but it also handles entity n-gram word spans. The language model used is fixed to BART and several patterns (a.k.a. *templates*) are experimented: (1) "$\mathbf{x}$. *candidate span* is a *entity-type* entity."; (2) "$\mathbf{x}$. The entity type of *candidate span* is *entity type*."; (3) "$\mathbf{x}$. *candidate span* belongs to *entity type* category."; (4) "$\mathbf{x}$. *candidate span* should be tagged as *entity type*.". For each template, also a "none-entity" version is generated (e.g.,

[10]Template-based NER [22] source code: https://github.com/Nealcly/templateNER

Table 1. Overview of Evaluations Conducted in the Reviewed Articles Included in the Transfer Learning Category

Category	Method	Schema	Dev	SNIPS		CoNLL			GUM		WNUT		OntoNotes		CM-NER	ICON 2013	Wiki ANN	BC2GM	NCBI	MIT-MR				MIT-RR				ATIS			
				1	5	1	5	sub	1	5	1	5	1	5	10%	100%	100	10	10	10	20	50	100	10	20	50	100	10	20	50	100
Cross domain	Hou et al. [56]	IOB	✔	67.27	73.27	47.94	46.56	–	3.48	12.46	23.01	33.59	26.10	22.81	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	La-DTL [159]	BIOES	✔	–	–	–	–	–	–	–	–	–	–	–	71.15	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cross language	Dandapat et al. [26]	IOB	✖	–	–	–	–	–	–	–	–	–	–	–	–	78.33	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	Ni et al. [102]	IOB	✔	–	–	–	–	63.00	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	Cottarell et al. [20]	IOB	✔	–	–	–	–	–	–	–	–	–	–	–	–	–	68.02	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	BWET [163]	IOB	✔	–	–	–	–	66.67	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	MMNER [115]	IOB	✔	–	–	–	–	69.67	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Prompt-based	ConVEx [52]	IOB	✖	–	76.8	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	PETER [42]	IOB	✖	–	–	–	–	–	–	–	–	–	–	–	–	–	–	23.0	0.0	–	–	–	–	–	–	–	–	–	–	–	–
	Template-based NER [22]	IOB	✖	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	42.4	54.2	59.6	65.3	53.1	60.3	64.1	67.3	77.3	88.9	93.5	–
	PromptSlotTagging [55]	IOB	✖	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	53.31	60.19	66.13	69.63	52.10	61.49	66.83	70.98	–	–	–	–

[1]In Hou et al. [56] and ConVEx [52] k-shot scenarios are simulated by generating support sets S so that all the labels in the domain appear at least k times in S; in Cui et al. [22] and PromptSlotTagging [55] k samples per entity type are randomly sampled; Cottarell et al. [20] generate k samples for each language; in PETER [42] k samples are randomly sampled from the original training set, but it is not guaranteed that they contain entity mentions.
When datasets have multiple entity types, we report averaged scores. Details of the datasets are reported in Appendix. The column *Schema* reports the annotation schema used, while *Dev* indicates whether a development set to choose hyperparameters has been used or not. For each dataset, we report the number of shots or the percentage of training data and the results obtained when available. *sub* indicates a subset of Spanish, Dutch and German data extracted from CoNLL.

the first template becomes "**x**. *candidate span* is not a named entity."). In their experiments, authors declare that the best template is the first one, but an ensemble approach leveraging all the templates achieves even higher performance.

PromptSlotTagging[11] [55] handles a major issue of the previously-described prompt-based NER approaches [22, 42]: while they have shown a consistent improvement in FS-NER performance, their downside lies in a lack of scalability, i.e., the number of transformed samples explodes as the quantity of available data increases, thus implying high training and prediction times. To speed up the overall process, *inverse prompt* are proposed to reversely predict slot values given entity types. For example, given the entity type *"arrival"* and a sentence **x** = "book a flight from Beijing to New York tomorrow morning", it may be transformed as: "**x**. Arrival refers to ___.", and then the language model learns to decode multi-word spans ("New York" in this example). Specifically, at training time a pre-trained language model is fine-tuned with answered prompts, and cross-entropy loss is computed only on answer tokens, not on the whole sentence.

3.1.4 Performance Evaluation. Experimental results reported in the reviewed articles are summarized in Table 1. The quality of FS-NER systems is heavily dependent on the domain application: results on the SNIPS dataset [21] covering a variety of areas (i.e., weather, music, playlist, book, search screen, restaurants, creative work), in particular, were noticeably better than those on the GUM dataset [175] extracted from Wikipedia. Moreover, the results suggest that the selection of training data has a significant impact on the performance of FS-NER models. Techniques that employ more sophisticated criteria for selecting data for training simulations outperform those using a simple random selection, such as PETER [42]. This finding emphasizes the importance of careful data selection and highlights the potential of advanced data selection techniques to improve few-shot learning performance.

The results also show how FS-NER systems can facilitate the acquisition of generalizable knowledge. The ConVEx technique [52], which exploits knowledge from pre-trained language models and significantly enhances the performance on SNIPS data [21], provides an example of how leveraging pre-existing knowledge can enhance FS-NER performance. This finding supports the notion that pre-training on large amounts of data can provide a foundation for acquiring transferable knowledge that can be applied to new tasks. Furthermore, the superior performance

[11]PromptSlotTagging [55] source code: https://github.com/AtmaHou/PromptSlotTagging

of PromptSlotTagging [55] compared to Template-based NER [22] highlights the potential of advanced prompt engineering techniques to improve FS-NER performance.

3.2 Meta-learning

Inspired by human intelligence, meta-learning (a.k.a. learning to learn) [128, 147] aims at quickly adapting models to new tasks without the need to re-train them from scratch and with only few data points. As humans, we are able to easily learn new skills after a few minutes or zero experience: for example, if we can ride a bike, we will easily learn to ride a motorcycle. This can be accomplished by ML models with a meta-learning phase during which the model learns to adapt to a large variety of tasks.

3.2.1 Background. The basic meta-learning process can be described as follows: (1) during the meta-training phase, the meta-learner is trained in multiple *episodes*, each constituted by a support and a query set (equivalent to the training and test sets when considering a standard classification task); (2) in the meta-test phase, the knowledge acquired during meta-train is transferred to the target task by using its training data to train the classifier for each episode. The support/query set split of the meta-training phase is designed to simulate the training/test splits which will be encountered during the meta-test phase.

Applications of meta-learning to few-shot scenarios can be categorized in metric-based methods [140, 152], memory-based methods [118, 126] and optimization-based methods [35, 64]. In metric-based methods, models consists of two consecutive parts: (1) an embedding function, which focuses on learning transferable embeddings, and (2) a classifier, which identifies the correct label given the defined metric (e.g., distance, relation). In memory-based techniques, a neural network is usually trained to learn how to store and retrieve "memories" to use for downstream tasks. Finally, optimization-based algorithms focus on the optimization algorithm for the meta-training phase.

Model Agnostic Meta-Learning (**MAML**) [35], Prototypical Networks [140] and Relation Networks [145] are the basis of most of the current applications of meta-learning to FS-NER. The objective of MAML is to train a model so that it can easily learn a new task from a small amount of data. Differently, Prototypical Networks assume that points of the same class cluster around a single *prototype* representation in an embedded space, and thus few-shot classification can be done by finding the nearest prototype. Relation Networks propose a learnable non-linear module which outputs relation scores over element-wise sum of support and query features. In the following, we briefly describe the above-mentioned standard approaches and then we will dive into NER extensions.

Model-Agnostic Meta Learning (MAML). MAML first forms a set of training tasks $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2 \dots \mathcal{T}_n\}$, each task $\mathcal{T}_i$ being represented by a training and a validation set. For each task, model parameters updates are computed with gradient descent:

$$\theta_i' = \theta - \alpha \nabla \mathcal{L}_{T_i}^{train}(f_\theta), \tag{5}$$

where α is a universal learning rate (i.e., shared across tasks) and $\mathcal{L}_{T_i}$ is the training-loss related to the ith task. Meta-optimization objective is to optimize model performance across all the tasks:

$$\min_\theta \sum_{\mathcal{T}_i} \mathcal{L}_{T_i}^{val}(f_{\theta_i'}) = \sum_{\mathcal{T}_i} \mathcal{L}_{T_i}^{val}(f_{\theta_i'}) \tag{6}$$

Hence, the validation error on sampled tasks $\mathcal{T}_i$ serves as the training error of the meta-learning process. Model weights are updated as follows:

$$\theta_i' = \theta - \beta \nabla \sum_{\mathcal{T}_i} \mathcal{L}_{T_i}^{val}(f_{\theta_i'}), \tag{7}$$

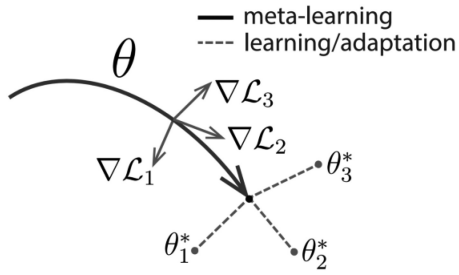

Fig. 8. How MAML works at meta-test time. Every task τ_i has an optimal parameter θ_i^* and the adaptation along the gradient $\nabla\mathcal{L}_i$ provides a parameter θ_i' that should be close to θ_i^*. Illustration from Finn et al. [35].

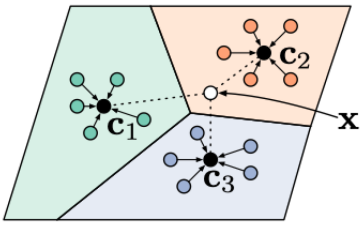

Fig. 9. Prototypical networks for few-shot learning. Prototypes are computed as the average of embedded support examples for each class. Illustration from Snell et al. [140].

where β is the learning rate of meta-optimization. In Figure 8 we show MAML working at meta-test time: each task τ_i has an optimal parameter θ_i^*, and adjusting the parameter along the gradient $\nabla\mathcal{L}_i$ yields a parameter θ_i' that is expected to be close to θ_i.

Prototypical Networks. Differently, Prototypical Networks are trained so that the outputs of the last but one layer allow an easy separation of instances into meaningful clusters, i.e., outputs are similar for objects belonging to the same class and diverse for objects of different classes. In this way, it is possible to assign a class to unseen instances even with few samples. More specifically, they consider the available support set for class k to compute an M-dimensional representation $c_k \in \mathbb{R}^M$, named *prototype*, as the mean vector of the embedded support points (as shown in Figure 9):

$$\frac{1}{|S_k|}\sum_{\mathbf{x}_i \in S_k} f_\phi(\mathbf{x}_i), \tag{8}$$

where $S_k = \{\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots\}$ is the support set of class k and $f_\phi : \mathbb{R}^D \rightarrow \mathbb{R}^M$ is the embedding function with learnable parameters ϕ.

The class assignment for the query set is then based on a softmax over distances to the prototypes in the embedding space:

$$p_\phi(y = k|\mathbf{x}) = \frac{exp(-d(f_\phi(\mathbf{x}), c_k)}{\sum_{k'} exp(-d(f_\phi(\mathbf{x}), c_{k'})} \tag{9}$$

Parameters ϕ are trained by episodes, each generated by randomly sampling a subset of classes from the training set, then choosing a subset of samples for each chosen class to act as the support set. Then, negative log-likelihood of the true class k $J(\phi) = -logp_\phi(y = k|\mathbf{x})$ is minimized with stochastic gradient descent.

Relation Networks. Similarly to Prototypical Networks, this approach maps data in an embedded space, but it further defines a relation classifier instead of relying on a fixed metric. More

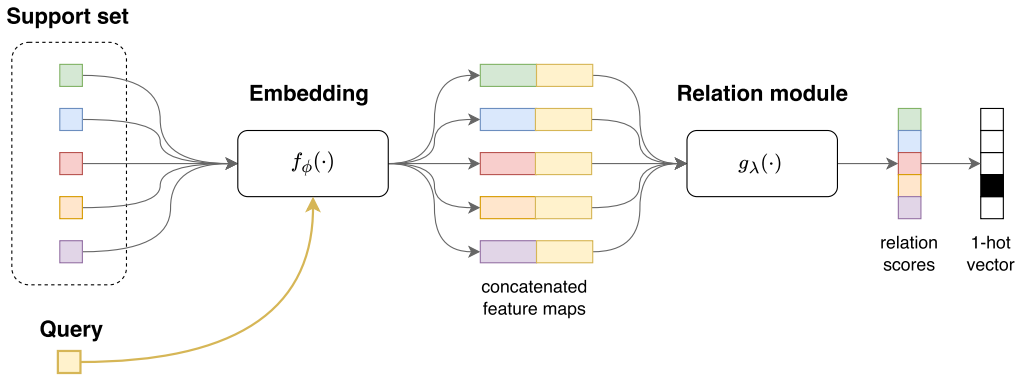

Fig. 10. Architecture of a Relation Network for a 5-class classification problem and one shot for each class.

specifically, given the embedding function $f_\phi : \mathbb{R}^D \rightarrow \mathbb{R}^M$, a feature map for each class c_k (analogous to the prototype in Prototypical Networks) is generated by performing an element-wise sum over the feature maps $f_\phi(\mathbf{x_i})$ of all samples $\mathbf{x_i}$ from each training class. The class feature map c_k and the embedding $f_\phi(\mathbf{x_j})$ of a query sample $\mathbf{x_j}$ are then combined with an operator $C(c_k, f_\phi(\mathbf{x_j}))$ (e.g., concatenation), and the combined feature map is fed into the relation module g_λ which produces the *relation score* r_{kj}, i.e., a scalar value representing the similarity between the two embeddings, $r_{kj} = g_\lambda(C(c_k, f_\phi(\mathbf{x_j}))$. Parameters (i.e., ϕ, λ) are learned by regressing relation scores to the ground truth with **mean squared error** (**MSE**) loss:

$$\sum_{k=1}^{K}\sum_{j=1}^{n}(r_{kj} - \mathbf{1}(y_j == k))^2, \tag{10}$$

where K is the number of classes, $\mathbf{1}(y_j == k)$ equals to 1 when the label y_j of the query sample $\mathbf{x}_j$ corresponds to class k, 0 otherwise. An example of Relation Network for a 5-way 1-shot classification problem is shown in Figure 10.

3.2.2 Applications to Few-shot NER. While having being widely applied for few-shot image classification [36, 78, 116, 158], only recently meta-learning attempts have been made in NLP applications. Gu et al. [47] used meta-learning in neural machine translation, adapting the model to low-resource languages. Huang et al. [58] applied MAML to the query generation task. Qian and Zhou [112] proposed DAML, which learns general and transfereable information by combining multiple dialog tasks during training. Lin et al. [103] use meta-learning to generate personalized responses by leveraging just a few dialog samples.

In the following, we describe some peculiarities of works adopting meta-learning or its variants to solve the FS-NER problem.

ProtoNER[12] [39] is the first work to adapt Prototypical Networks to NER tasks. The standard approach is modified as follows:

— Situations where words from the same sentence fall in different support and query sets is prevented, so as to not break the sentence structure, given that words in sentence influence each other.

[12]ProtoNER [39] source code: https://github.com/Fritz449/ProtoNER

— Since words associated to class O (outside of an entity mention) should not be necessarily neighbors in the embedded space, the similarity score for the O class is replaced with a scalar value (treated as an hyper-parameter).
— Two training sets are used: *out-of-domain* and *in-domain*. The former is quite large and refers to a different class w.r.t. NER target, the latter is a small dataset of labeled examples of the target class. The base model is first trained on the out-of-domain training set and its weights are then used to initialize the Prototypical Network.

MetaNER [74] is the first work to extend a MAML-based approach to sequence labeling. It decompose the NER process into two modules: a *sequence encoder* and a *tag decoder*. Data from available tasks $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2 \dots \mathcal{T}_n\}$, where $\mathcal{T}_i$ contains annotated data $(\mathcal{X}_i, \mathcal{Y}_i)$ ($\mathcal{X}_i$ being the sequence of tokens and $\mathcal{Y}_i$ the corresponding sequence of NER tags) are used to train a meta-knowledge learner for the sequence encoder, while new unseen domain data $\mathcal{T}_{new}$ is used to fine-tune the learned sequence encoder and a new tag decoder. An adversarial network is used to improve the generalization of the model.

Krone et al. [65] propose a less-restrictive extension of Prototypical Networks w.r.t. ProtoNER, since it does not require a separate network for each named entity type. Prototypical Networks are here applied to jointly perform intent classification and slot filling. Let $S_l = \{(x_l^i, t_l^i, y_l)\}$ be the support set instances with intent class y_l and $S_a = \{(x_{[1:j]}^i, t_{[1:j]}^i, y^i | t_j^i = a\}$ be the the support set of sub-sequences with slot-label $t_j^i = a$. Intent class prototypes c_l and slot-label prototypes c_l are separately computed as follows:

$$c_l = \frac{1}{|S_l|} \sum_{(x_l^i, t_l^i, y_l) \in S_l} f_\phi(x^i), \tag{11}$$

$$c_a = \frac{1}{|S_a|} \sum_{(x_{[1:j]}^i, t_{[1:j]}^i, y^i) \in S_a} f_\phi(x_{[1:j]}^i). \tag{12}$$

As in standard prototypical networks, a softmax over distances to prototypes in the embedding space is leveraged to assign classes. Parameters are learned by minimizing a composed loss as the sum of intent-classification and slot-filling negative log-likelihoods. Slot-filling loss is computed as the sum of negative log-likelihoods for each token in the sequence.

Label-enhanced Task-Adaptive Projection Networks (L-TapNet)[13] [54] combines the advantages of taking into account the similarity of query samples to class representative samples, which is a fundamental characteristic of Prototypical and Relation networks, with the performance improvement brought by taking label dependencies [60, 89] and the semantic relations between label names and slot words (for example, the word *rain* and the label name *weather* are highly related) into account. Specifically, given a support set S and a query sentence $\mathbf{x}$, a linear-CRF is applied and thus label sequence class assignment is done as follows:

$$p_\phi(\mathbf{y}|\mathbf{x}, S) = \frac{exp(\mathcal{T}(\mathbf{y}) + \lambda \mathcal{E}(\mathbf{y}, \mathbf{x}, S))}{\sum_{\mathbf{y}' \in \mathbf{Y}} exp(\mathcal{T}(\mathbf{y}') + \lambda \mathcal{E}(\mathbf{y}', \mathbf{x}, S)} \tag{13}$$

In the above-reported equation, $\mathcal{T}(\mathbf{y}) = \sum_{i=1}^{n} f_\tau(y_{i-1}, y_i) = \sum_{i=1}^{n} p(y_i|y_{i-1})$ is the output of a *Transition scorer* which captures the dependencies between labels in a matrix $\tau^{N \times N}$, where N is the number of labels. Since source domain data used in the meta-training phase may have different label sets w.r.t. target domain data, a *collapsed dependency transfer* mechanism [56] is employed by

[13] L-TapNet [54] source code: https://github.com/AtmaHou/FewShotTagging

Table 2. Overview of Evaluations Conducted in the Reviewed Articles Included in the Meta Learning Category

			OntoNotes	ACE2005	ATIS		TOP		SNIPS					
Method	**Schema**	**Dev**	20	100%	20	100	20	100	1	5	10	15	20	100
ProtoNER [39]	IOB	✔	65.63	–	–	–	–	–	–	–	–	–	–	–
MetaNER [74]	BIOES	✔	–	76.21	–	–	–	–	–	–	–	–	–	–
Krone et al. [65]	IO	✖	–	–	40.90	41.28	28.27	28.33	–	–	–	–	59.73	62.14
L-TapNet [54]	IOB	✔	–	–	–	–	–	–	70.41	75.01	–	–	–	–
Oguz et al. [105]	IO	✖	–	–	–	–	–	–	–	82.7	86.3	87.5	–	–

When datasets have multiple entity types, we report averaged scores. Details of the datasets are reported in Appendix. The column *Schema* reports the annotation schema used, while *Dev* indicates whether a development set to choose hyperparameters has been used or not. For each dataset, we report the number of shots or the percentage of training data and the results obtained when available.

modeling labels with a higher level of abstraction: B (beginning), I (inside) and O (outside) labels are used to represent all slot words, and transitions are modeled by differentiating transition from the same or a different entity type (e.g., $p(B_{person}|I_{location})$ is the transition probability from a I label to a B label with different entity types).

In Equation (13), $\mathcal{E}(\mathbf{y})$ is the output of an *Emission scorer*. To compute the emission score of a word $f_{\mathcal{E}}(y_i, \mathbf{x}, S) = p(y_i|\mathbf{x}, S)$, an extension of TapNet [172] is proposed. Differently from Prototypical Networks, TapNet projects data in an embedded space where words with different labels are well-separated to reduce misclassifications thanks to three key elements: an embedding network f_θ, a set of class reference vectors $\Phi = [\phi_1, \ldots, \phi_N$, and a mapping $\mathbf{M}$ of embedded features to a new classification space, which is constructed such that the embedded feature vectors and the class reference vectors with matching labels align closely. Both the embedding network parameters θ and the reference set Φ are update episode after episode with a softmax based on Euclidean distance between the mapped query sample and reference vectors. L-TapNet enhances this framework by finding a projector $\mathbf{M}$ that aligns query samples not only to class reference vectors but also label semantic representations $\mathbf{s} = [s_1, \ldots, s_N]$ computed with a BERT transformer network. The label-enhanced reference τ_j is thus computed as $\tau_j = (1 - \alpha) \cdot \phi_j + \alpha s_j$, where α is a balancing parameter.

Oguz et al. [105] extend Relation Networks for NER by including a learnable attention module. Differently from L-TapNet [54], authors suggest to not use slot descriptions or slot names but small amounts of annotated samples from different domains as training inputs due to two main reasons: (1) slot descriptions require qualified linguistic expertise and (2) there is not always a relation between slot names and the corresponding tokens. To handle the scenario where a NER task has to be solved with unseen slot labels, the same meta-learning procedure as Matching Networks [152] is applied to guarantee a robust episodic training: each episode is constructed by randomly selecting C unique classes and K labeled examples for each class for support $S = \{(x_i, y_i)\}_{i=1}^m$ and query $Q = \{(x_i, y_i)\}_{i=1}^m$ sets, with $K > 1$ and $m = K * C$. Each episode is divided in an *embedding stage* producing feature maps from S and Q with contextual embeddings such as ELMO and BERT, and a *relation module* which calculates the relation scores between support and query, thus assigning the label of the most relational value as a label of query. A learnable *attention module* inspired from Jetley et al. [61] is included to highlight the relevant and suppress the misleading samples between support and query samples, thus learning how to attend local and global features.

3.2.3 Performance Evaluation. We summarize the experimental results of FS-NER techniques based on meta-learning in Table 2. The table shows that there is a significant discordance in the choice of annotation schemes, use of validation sets, and datasets to be utilized. This shows that

there does not exist unambiguous agreement among researchers regarding the most effective methods to develop FS-NER models. The only comparison that can be made is between L-TapNet [54] and the methods used by Oguz et al. [105] on the SNIPS dataset [21] in a 5-shot scenario. The results show that, despite not using slot descriptions or slot names, Oguz et al. [105] technique surpasses L-TapNet [54] in terms of performance. However, we also note that the two techniques appear to use different annotation schemes, which makes it difficult to draw fair conclusions.

3.3 Summary and Discussion

Since when deep learning has started to proliferate, model architectures have been the focus of research. In the context of FS-NER, works are based on transfer or meta learning. The two approaches, while similar in their objective (i.e., improving the performance in the presence of scarse data resources), are slightly different in the mode in which they operate: while transfer learning uses a model trained with data from a similar domain or language to shift its knowledge to another model, meta learning focuses on training procedures allowing models to quickly adapt to new tasks.

When a model trained on a similar task is available, transfer learning can be easily applied without too many adaptation, and guarantees high-quality results, especially with transformer architectures. However, this ideal scenario is quite uncommon, since real-world task may be similar but deal with another language, or other entity types, thus requiring many adaptations which may also limit the resulting performance — e.g., multi-lingual transfer usually requires a machine-translation step whose error inevitably propagate across the transfer-learning framework. To overcome this, meta-learning models easily adapt when new tasks emerge (e.g., a new entity type is required to be recognized).

4 DATA-CENTRIC METHODS

The ever-increasing attention toward Data-Centric AI [108] is reflected in a high number of FS-NER works focusing on data to improve performance. In this section, we review data-centric FS-NER approaches by separately focusing on *data augmentation*, *distant supervision*, *active learning* and *self learning*, as outlined in Figure 3. Hence, we line up methods as a story and summarize their key characteristics and discuss similarities and differences under each category as well as limitations that have not been addressed yet.

4.1 Data Augmentation

One common way to deal with the lack of data is *data augmentation*, which consists in increasing the size of the available dataset with new samples generated by means of heuristics or external data sources. Augmentation methods explored in current literature for NLP tasks usually manipulate words in the original sentence by word replacement [13], random deletion [160], word position swap [93], and generative models [171]. Applying these transformations to NER input samples is not possible due to the token-level classification implied by this task (each manipulation impacts labels). Thus, data augmentation techniques for NER are comparatively less studied [25]. In the following, we describe current methods applying data augmentation in few-shot scenarios. We provide an example of an augmented sample for each method in Figure 11.

Data Boost [85] explores the text generation ability of Language Models to generate augmented samples. In particular, GPT-2 [113] is used as a conditional generator and guided toward specific class labels by means of a **Reinforcement Learning** (**RL**) approach. A RL stage is added in between the softmax and argmax function of the conditional generator. The *state* at step t is the generated sentence before t $s_t = \mathbf{x}_{<t}$, where $\mathbf{x}_{<t} = \{x_0, x_1, \ldots, x_{t-1}\}$; the *policy* π_θ is the probability that token x_t is chosen (*action* a_t), i.e., the softmax output of the hidden states

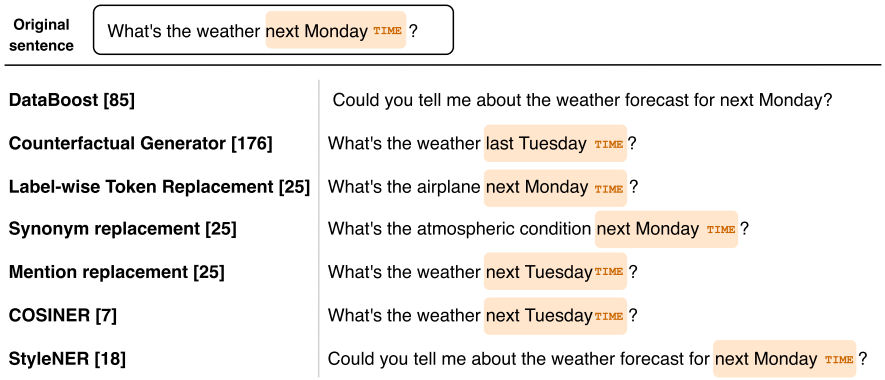

Fig. 11. Examples of augmented sentences with different data augmentation techniques.

$\pi_\theta(a_t|s_t) = softmax(h^\theta_{<t})$. *Reward*, following **Proximal Policy Optimization** (**PPO**) [130] for a given conditional token x^c_t is computed as follows:

$$R(x^c_t) = \mathbb{E}\left[\frac{\pi_{\theta_c}(a_t|s_t)}{\pi_\theta(a_t|s_t)} \cdot G(x^c_t)\right] - \beta \cdot KL(\theta||\theta_c), \tag{14}$$

where $G(x^c_t)$ is the *salience gain* which quantifies the generated token resemblance to target label lexicon by a logarithm summation of cosine similarity with each word in the salient lexicon; KL is the **Kullback-Leibler** (**KL**) divergence between conditional θ_c and unconditional θ distributions; β is a weighting parameter. However, this method has been conceived for text data augmentation, and does not directly apply to NER, as it does not provide the mapping of the entity mention from the original to the augmented sentence.

Counterfactual Generator[14] [176] addresses the poor generalization ability of few-shot systems to spurious correlations between the entities and their contexts. For example, in the sentence "John lives in New York", the entity "New York" and its context ("John lives in") are highly correlated, but it is not true that one *causes* the other. Thus, based on the idea that entities and contexts are not in causal relationships, Zeng et al. [176] provide a framework to generate weakly-labeled counterfactual examples in few steps: (1) a vocabulary with the desired entity type is prepared, (2) an entity is randomly selected from the input sentence and (3) replaced with a different entity from the entity set to form a counterfactual example; finally, (4) a discriminator model is trained to distinguish good examples from counterfactuals: if the discriminator, i.e., a NER model, can correctly recognize the replaced entity, the counterfactual example is considered in the new training set.

Dai et al. [25] investigate the adaptation of many simple data augmentation methods for NER problems. They are listed as follows:

— *Label-wise Token Replacement (LwTR)*: we randomly decide if a token from the input sentence has to be replaced. If yes, it is randomly replaced with another available token with the same label.
— *Synonym Replacement (SR)*: similar to LwTR, but tokens are replaced with synonyms retrieved from WordNet.
— *Mention Replacement (MR)*: similar to LwTR, but only applied to mentions, which are randomly replaced with other mentions from the original training set with the same type.

[14]Counterfactual Generator [176] code: https://github.com/xijiz/cfgen

— *Shuffle within Segments (SiS)*: sentences are split into segments with the same label, and tokens are randomly shuffled within these segments.

Experimental results show that all the investigated data augmentation techniques are effective in few-shot settings (e.g., 50- and 100-shot scenarios), but they often fail on full datasets due to the degrading effects of noise.

COSINER [7] is a more sophisticated variation of **Mention Replacement** (**MR**) [25]. It employs the cosine similarity to replace mentions with those that are the most similar, as opposed to randomly replacing them. Concept embeddings $V_{concept}$ extracted from each mention in the training set are used to calculate similarity and computed as follows:

$$V_{concept} = V_{concept} + lr \cdot (1 - sim) \cdot V_{context}, \tag{15}$$

where

$$lr = \frac{1}{C_{concept}}$$

and

$$sim = \max\left(0, \frac{V_{concept}}{||V_{concept}||} \cdot \frac{V_{context}}{||V_{context}||}\right),$$

$V_{context}$ being the embedding extracted from a transformer network used as a feature extractor. The similarity-based nature of this method shows improvements in performance w.r.t. techniques based on random replacements.

StyleNER[15] [18] learns patterns (e.g., style, noise, abbreviations) that differentiate data from high-resource and low-resource domains, and a shared space where they are aligned. The key idea is that text data may differ in textual patterns (e.g., long and complex sentences from research abstracts versus social media data), but their semantics are transferable. Given a source domain D_{src} and a target domain D_{tgt} dataset, sentences are linearized by inserting entity labels before their corresponding text span, and a random pair of sentences is then extracted from the two datasets and provided as input to the model. The model works in two steps:

— *Denoising reconstruction*: the model learns input embeddings from their corresponding domain. Inputs are perturbed in several ways (e.g., shuffling, dropping, masking) to inject noise, and the model has to capture semantics and learn patterns that differentiate sentences across domains. Then, a decoder reconstructs noisy sentences to their corresponding domain.
— *Detransforming reconstruction*: based on their semantics, sentences are transformed from one domain to the other. Then, the model generates embeddings for transformed sentences and learns to reconstruct each of them in their corresponding domain.

A discriminator is also trained to distinguish the domain of an embedded sentence, which allow to determine if the encoder can generate meaningful representations or the model has to bypass the intermediate mapping step between domains.

4.1.1 Performance Evaluation. The experimental results summarized in Table 3 offer empirical proof to back up the claim that using data augmentation methods consistently results in performance gains. Moreover, the data suggest that limited contexts, such as those encountered in few-shot scenarios, can benefit even more from data augmentation. It is worth noting, however, that comparing the effectiveness of various data augmentation techniques across different datasets is difficult. Nevertheless, the results obtained on OntoNotes[16] by ProtoNER [39] presented in Table 2

[15]StyleNER [18] code: https://github.com/RiTUAL-UH/style_NER
[16]https://catalog.ldc.upenn.edu/LDC2013T19

Table 3. Overview of Evaluations Conducted in the Reviewed Articles Included in the Data Augmentation Category

Method	Schema	Dev	Shots	Dataset	F1	+%
Counterfactual Generator [176]	IOB	✔	100	CNER	47.6	0.4
				IDiag	67.9	9.6
				CLUENER	34.8	4.6
Dai et al. [25]	IOB	✔	50	MaSciP	71.2	3.1
				i2b2-2010	41.5	6
COSINER [7]	IOB	✖	108 (2%)	NCBI-Disease	69.2	4.1
			91 (2%)	BC5CDR	83.2	4
			251 (2%)	BC2GM	66.5	2.1
StyleNER [18]	IOB	✔	1000	OntoNotes + T-Twitter	59.21	0.78

When datasets have multiple entity types, we report averaged scores. The last column (+%) reports the improvement of performance obtained by augmenting the original dataset.

suggest that meta-learning outperforms data augmentation techniques, achieving better performance with only 20 training samples. This finding suggests that while data augmentation alone is probably worse than model-centric approaches, combining data augmentation with specially designed methods for few-shot learning may lead to even better performance.

4.2 Distant Supervision

In few-shot scenarios, labeled data could be retrieved from heuristics, different domains or laguages, external knowledge bases, or ontologies. *Distant supervision* [94] aims at leveraging such resources to heuristically annotate training data. For example, in biomedicine there are a lot of curated resources available: *NCBO Bioportal* [161] houses 541 biomedical ontologies, ***Medical Subject Headings (MeSH)***[17] is a controlled vocabulary with 347,692 classes of medical items, and so on. Combining ontologies is a difficult task due to their heterogeneous structures, concept granularities and overlaps or conflicts between definitions of entities. Generally, the main steps of distant supervision are (1) *candidate generation*, i.e., the identification of potential entities, and (2) *labeling heuristics* to generate noisy labels, as shown in the example of Figure 12. The use of distant supervision for FS-NER in low-resource languages has yet to be deeply explored. The amount of external information available in low-resource settings might be very limited: for example, the Wikipedia knowledge graph contains 4 million person names in English while only 32 thousand in Yorùbá [1]. Furthermore, without further tuning under better supervision, distantly supervised models have low recall [14]. In the following, we describe methods that apply distant supervision to NER tasks.

SwellShark[18] [38] aims at automatizing the generation of candidates and noisy labels without hand-labeled data. Its inputs are (1) a collection of unlabeled documents and (2) some form of weak supervision, typically ontologies and heuristic rules. The candidate generator Γ_Θ is defined as the mapping function from a document collection D into a candidate set $\Gamma_\Theta : D \rightarrow \{x_1, x_2, \ldots, x_N\}$, where each candidate x_i is a character-level span within the document. Candidates in SwellShark are determined by heuristics, e.g., dictionary of noun phrases matching with regular expression. To filter actual entities from candidates, a *labeling function generator* Γ_λ is defined as a function which receives a resource R for weak supervision (e.g., ontology, term frequencies) and generates labeling functions $\Gamma_\lambda : R \rightarrow \{\lambda_1, \lambda_2, \ldots, \lambda_N\}$ (e.g., lexicon matching, frequency-based thresholds).

[17] https://www.nlm.nih.gov/mesh/meshhome.html
[18] SwellShark [38] code (Snorkel project): https://github.com/snorkel-team/snorkel

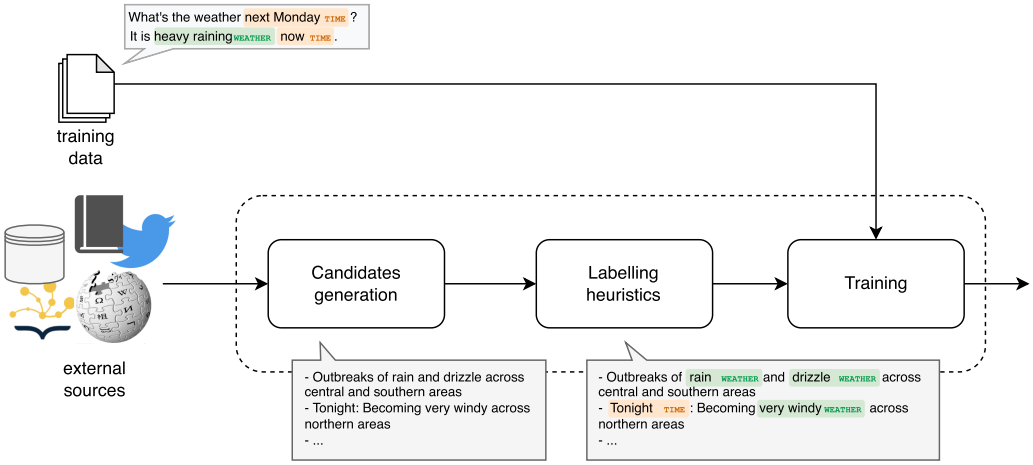

Fig. 12. General flowchart of a distant supervision approach.

Choosing heuristics intrinsically involves a tradeoff between development time and performance: greedy heuristics such as dictionary matching often imply low recall, while noun phrases candidates generate large sets which usually consist in an increase of noise, and hand-tuned heuristics may indeed result in high performance, but require more efforts.

AutoNER[19] [134] handles the problems of *incompleteness* and *noise* in automatically-labeled NER data, which characterize methods based on the detection of entity spans with heuristic rules, such as regular expressions [38, 120] and exact string matching [43, 51]. Nevertheless, unmatched tokens in are simply ignored when entities are not covered by the used ontology, introducing many false-negative labels (incompleteness). Furthermore, these methods often require expensive expert effort to cover many special cases. To handle these problems, AutoNER marks some high-quality out-of-dictionary phrases as "potential entities" without requiring human effort. In particular, to leverage the information embedded in dictionaries, AutoNER proposes the *tie or break* tagging scheme, which tags two adjacent tokens as (1) *tie* if they belong to the same entity type, (2) *unknown* if at least one of them belongs to an unknown-type phrase, and (3) *break* otherwise. This scheme is used for entity span detection, while entity types are then identified based on feature vectors of candidates. Predicting whether two adjacent tokens refer to the same entity or not, AutoNER builds more robust distant supervised models (since it is often noisy on boundaries of entity mentions but not in their inner ties).

One of the main advantages of AutoNER is that entity mentions may be marked as *unknown*, allowing us to include token which we are unable to identify their types based on distant supervision. For example, "prostaglandin synthesis" may be present in both disease and chemical lexicons, or lexicon may not cover all the possible entity types. AutoPhrase [133] is used to automatically mine new phrases from unlabeled text and a dictionary of high-quality phrases. All the out-of-dictionary phrases are then labeled as *unknown* and added to the dictionary.

Yang et al.[20] [167] also addresses the *incomplete* and *noisy* annotation problems. Weak sentences are obtained by using a dictionary D built from named entities available in the training dataset H to weakly annotate a large unlabeled pool of sentences. Note that in this work "distant" resources

[19] AutoNER [134] code: https://github.com/shangjingbo1226/AutoNER
[20] Yang et al. [167] code: https://github.com/rainarch/DSNER

are used to augment the training set, but the dictionary is built by relying on the available data only, which may be a limit in highly-constrained few-shot settings, where training data do not cover all the possible named entities. To handle the *incomplete* annotation problem, sentences are allowed to have partial annotations: some token spans are annotated with definite labels, while all the others are associated to all the possible labels. The *noisy* annotations problem is handled with a reinforcement learning approach which follows Feng et al. [34] to obtain clean instances from distantly supervised NER data. In particular, the state $\mathbf{S_t}$ is a vector containing: (1) representation of the current sentence obtained with a BiLSTM layer, (2) label scores computed with a MLP layer from the shared encoder, and (3) distant annotation of the instance. The action $a_t \in \{0, 1\}$ indicates whether to select the tth distantly supervised sentence, and the policy is learned by optimizing NER performance (reward).

Knowledge-Augmented Language Model (KALM) [80] augments a traditional language model with a knowledge base without requiring any additional component; in addition, it learns to recognize entities in an entirely unsupervised way by using entity type information which is latent in the model. In particular, KALM has the ability to predict masked words from a vocabulary V_g like any other language model, but it has a separate vocabulary V_i for each entity type and is able to predict whether to expect an entity from the context. Formally, given a latent variable τ_i denoting the entity type i and previously observed words $c_t = [y_1, y_2, \ldots, y_{t-1}, y_t]$, the probability of the next word is computed as follows:

$$P(y_{t+1}|c_t) = \sum_{j=0}^{K} P(y_{t+1}, \tau_{t+1} = j|c_t) = \sum_{j=0}^{K} P(y_{t+1}|\tau_{t+1} = j, c_t) \cdot P(\tau_{t+1} = j|c_t), \tag{16}$$

where $P(y_{t+1}|\tau_{t+1}, c_t)$ is the distribution of entity words of type τ_{t+1} and $P(\tau_{t+1} = j|c_t)$ is the probability that the next word has a given type j. Both are computed as in standard language models, i.e., by projecting the hidden state of a LSTM model and normalizing with a softmax. However, the base model is enhanced by using as input not only the embedding vector of an input word, but also the embedding of the type of the previous word: in this way, KALM is able to model context by taking count of entity types, and it allows to learn latent types more accurately during training.

Cao et al.[21] [14] try to maximize the potential of automatically weakly labeled data (e.g., anchors from Wikipedia) by dealing with the *incomplete* and *noisy* annotations problems. To obtain a high recall, the framework generates as many weakly-labeled data as possible with a *label induction* approach assigning labels to words based on Wikipedia anchors and taxonomy, and a *data selection* scheme which computes a scoring function to distinguish high-quality data from weak sentences and a neural model is then trained on such data. For data selection, the same approach as Ni et al. [102], which is based on annotation confidence and coverage, is applied. For sequence labeling from high-quality weakly-labeled data, Partial CRFs [146] are employed, while a *classification module* regards name tagging from noisy weakly-labeled data as a multi-label classification problem predicting each word label separately. The first layers of the two network are shared so as to allow knowledge distillation.

Graph Attention Model[22] [87] uses a domain-specific 2dictionary to create a *word formation graph* which captures variants of entities and thus discovers as new entity mentions as possible. In particular, the vertex set contains all words in the dictionary, which are connected by undirected

[21] Cao et al. [14] code: https://github.com/zig-kwin-hu/Low-Resource-Name-Tagging
[22] Graph Attention Model [87] code: https://github.com/yx100/GAT-BiLSTM-CRF

edges if they appear in the same entity type. Mention candidates are then extracted with a graph-matching algorithm. After extracting all the candidates, a *word-mention graph* integrates word formation of candidate entities into their sentences: the vertex set contains words in the input sentence and mention candidates extracted. In this way, links between words allows us to capture contextual information, while word-mention links capture the semantic of mentions. Graph information is then leveraged by a learning model including (1) a word embedding layer to represent words, (2) a BiLSTM layer to capture contextual information, and (3) a **graph attention network** (**GAT**) [151] to incorporate the information of mention candidates.

Linked HMM[23] [123] claims that the standard approach of generating candidate spans and then independently labeling each candidate, as in SwellShark [38], limits the applicability to tasks where candidate generators exist, and increases human efforts. Furthermore, candidate generators should identify all the possible entity spans, since errors propagate in the pipeline. Hence, Linked HMM allows users to write multiple rules which provide partial tags to sequences, whose accuracy is estimated by using an identifiable probabilistic generative model *without* labeled training data. The estimated posterior distribution is used over the true tags to train a sequence tagger. The rules which can be provided are categorized into two types: (1) *tagging rules*, which vote on the correct tags of sequence elements (they take a training input sequence and output a sequence of the same length indicating their votes on the true tags); (2) *linking rules*, which vote on whether an adjacent element should have the same or different tag (they allow distant supervision to propagate along sequences in an user-controlled way). These two types of rule are then used by a *linked Hidden Markov Model* to estimate the true tags for training by reconciling incomplete and conflicting information from multiple rules provided by users.

Consensus Network[24] [67] can be trained on imperfect annotations from multiple sources (e.g., crowd annotations, cross-domain data) by learning representations for each source and dynamically aggregating them by a context-aware attention mechanism. This is based on the intuition that different sources of supervision may have different strengths based on scenarios where they are applied. The framework first uses a multi-task learning schema based on a BiLSTM-CRF network to *decouple* model parameters into a set of annotator-invariant model parameters and a set of annotator-specific representations, then trains a context-aware attention module for a consensus-based representation by combining predictions on the target data. Specifically, scores from the annotator k are obtained by combining emission and transformation score matrices from the BiLSTM-CRF network with annotator-dependent matrix $\mathbf{A}^{(k)}$ which represents the pattern of annotation bias, i.e., the entry $\mathbf{A}_{ij}^{(k)}$ is the probability of assigning the wrong label j instead of the correct label i. Scores from different annotators are then combined with weighted voting, where the weight is given by F1 scores on the training set, and an attention module is added to improve generalization, since it provides more weight to sources which are more related to the input sentence.

4.2.1 Performance Evaluation. The experimental findings summarized in Table 4 demonstrate that distant supervision techniques can yield results that are comparable in quality to gold labeling at a significantly lower cost. The development of annotation functions, heuristics, dictionaries, or rules can serve as a cost-effective alternative to time-consuming and expensive labeling processes, even though some human effort may still be required. Moreover, combining weak labels with dictionaries has been found to be more effective than relying on partial training data or techniques such as data augmentation, as can be seen by comparing results obtained on biomedical datasets

[23]Linked HMM [123] code: 3https://github.com/BatsResearch/safranchik-aaai2020-code
[24]Consensus Network [67] code: https://github.com/INK-USC/ConNet

Table 4. Overview of Evaluations Conducted in the Reviewed Articles Included in the Distant Supervision Category

Method	Schema	Dev	Annotation effort	*NCBI-Disease*	*BC5CDR*	*LaptopReview*	*NEWS*	*CoNLL*	*W2019-language*	*W2019-food*	*AMT*
SwellShark [38]	IOB	✖	Labelling functions	67.1	83.7	–	–	–	–	–	–
			+ Specialized candidate generator	80.8	83.85	–	–	–	–	–	–
AutoNER [134]	Tie or Break	✖	Dictionaries	75.52	84.8	65.44	–	–	–	–	–
Yang et al. [167]	IOB	✔	Partial annotations	–	–	–	79.22	–	–	–	–
KALM [80]	IOB	✖	Dictionaries	–	–	–	–	89.0	–	–	–
Cao et al. [14]	IOB	✔	–	–	–	–	–	–	86.14	70.1	–
Graph Attention [87]	IOB	✔	Dictionaries	89.41	91.72	–	–	–	–	–	–
Linked HMM [123]	IOB	✖	Rules	79.03	82.96	69.04	–	–	–	–	–
Consensus Network [67]	IOB	✔	Crowdsourcing	–	–	–	–	–	–	–	79.99

When datasets have multiple entity types, we report averaged scores.

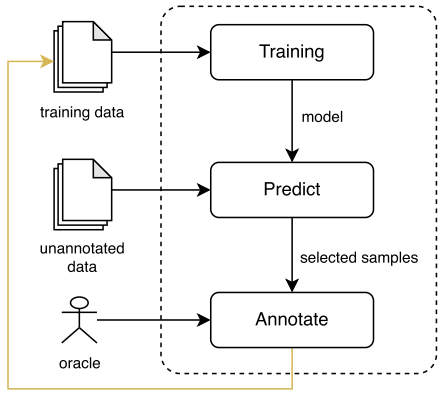

Fig. 13. Active Learning workflow.

(i.e., NCBI-Disease [32] and BC5CDR [77]) with those presented in Tables 3, 5. Furthermore, the results obtained using the KALM method [80] for CoNNL [125] demonstrate its superiority over transfer-learning methods (refer to Table 1) that rely on a limited number of labeled samples. In fact, KALM [80] yields outcomes that are almost as accurate as those generated using gold labels as shown in Table 5, hence highlighting the efficiency of weak labeling methods in minimizing labeling costs while retaining optimal quality.

4.3 Active Learning

Active Learning aims at selecting *informative* sets of examples for training, actively querying the user for labels, as shown in the cycle depicted in Figure 13. The most common approach is *uncertainty sampling* [72], in which the model selects examples based on the uncertainty of its predictions (a general approach uses entropy as an uncertainty measure [136]). While its theoretical properties have been extensively studied in past works [4, 6, 9, 27], Active Learning approaches are recently spreading in natural language processing tasks. Zhang et al. [180] are the firsts to investigate active learning for sentence classification: they use the ***Expected Gradient Length (EGL)*** [132] as an active learning strategy aiming to select the instances which would result in the maximum change in the current model parameter estimates if their labels were

provided. In the following, we will describe current applications of Active Learning to NER tasks.

Shen et al. [135] are the firsts to explore AL methods on **Deep Neural Networks** (**DNNs**) for the NER task. Due to the complexity of DNNs, the model is not retrained in each active learning round, but it is incrementally trained with each batch of new labels. The work uses uncertainty-based sampling strategies by considering three ranking methods: **Least Confidence** (**LC**) [23], **Maximum Normalized Log-Probability** (**MNLP**), and **Bayesian Active Learning by Disagreement** (**BALD**) [41]. Specifically, LC uses the probability of not predicting the most confident sequence from the model to sort samples in descending order:

$$1 - \max_{y_1,\ldots,y_n} P(y_1,\ldots,y_n|\mathbf{x}), \tag{17}$$

where $y_1,\ldots,y_n$ is the sequence of slot labels and $\mathbf{x} = \{x_i\}$ is the sentence, i.e., a sequence of tokens x_i, $i \in \{1,\ldots,n\}$. Since this approach naturally favours long sentences (which is a downside since they require more effort to be annotated), MNLP is proposed:

$$\max_{y_1,\ldots,y_n} \frac{1}{n} \sum_{i=1}^{n} logP(y_i|y_1,\ldots,y_{n-1},\mathbf{x}) \tag{18}$$

Finally, BALD uses a set of M models $P^1, P^2, \ldots, P^M$ sampled from the posterior. Uncertainty is then computed as the fraction of models which disagreed on the most popular choice:

$$\frac{1}{n} \sum_{j=1}^{n} 1 - \frac{\max_y |\{m : \arg\max_{y'} P^m(y_i = y') = y|}{M}, \tag{19}$$

where $|\cdot|$ is the cardinality of a set.

Monte Carlo dropout [40] is used to sample from the posterior. Results show that the performance of the best model trained in a standard supervised fashion is almost reached with approximately 30% of training data.

4.4 Self-training

Self-training, also referred to as *self-learning*, is an approach similar to distant supervision, where the model is trained on examples labeled by the model itself. As shown in Figure 14, the difference is that the labeling heuristics is replaced by the model itself. Originally proposed by Scudder et al. [131], it is one of the earliest semi-supervised methods. In the NLP field, it has been successfully applied to neural machine translation [50] and sentence classification [99]. Zoph et al. [183] show that self-training guarantees improvements in performance in both high- and low-data scenarios, while data augmentation results sometimes in decreases of performance of pretraining. In the following, we explore self-training approaches for FS-NER.

LM-LSTM-CRF[25] [84] leverages the knowledge obtained with a language model trained in an unsupervised fashion to improve sequence-labeling performance. It leverages word-level and character-level information of input-samples in a co-training fashion, i.e., each input is represented by different sets of features, each of them providing complementary information. Both a language model and a sequence-labeling model share the same character-level layer, which consists in two LSTM units learning character-level information in a completely unsupervised way to capture style and structure of input texts. However, since the two task handled are not strongly related,

[25]LM-LSTM-CRF [84] code: https://github.com/LiyuanLucasLiu/LM-LSTM-CRF

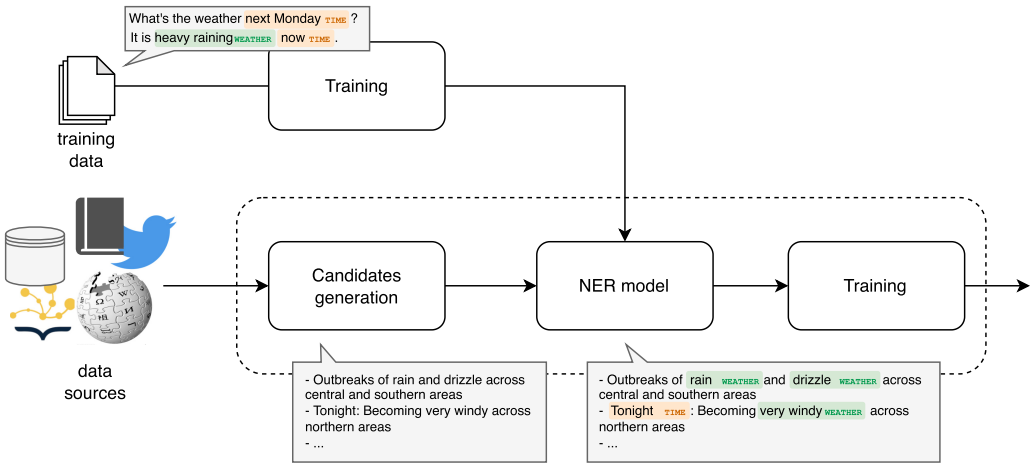

Fig. 14. General flowchart of a self-training approach.

empirical results show that this can hurt the overall performance. Hence, outputs of character-level layers are transformed into task-specific spaces, so that the language model can indirectly provide its knowledge to the NER model thanks to the shared layer, without sharing its feature space.

Chen et al. [15], considering that performance of self-training highly depends on how new data are selected, use a reinforcement learning to learn how to select instances from an unlabeled pool to be added to the training set in self-training scenarios. The NER model, which had been previously trained with few samples, will classify the new selected samples and be retrained. The self-training approach is considered as a decision process, described as a function that receives the self-labeled instances as inputs and outputs the acceptance or rejection of such instances to the training set. A **Deep Q-Network** (**DQN**) [96] then learns the selection strategy based on performance improvements on a development set. The Q-function is implemented using a neural network with three layers which receives the state $s = (h_s, h_c, h_p, h_t)$ of the learning framework as input, where h_s is a representation of the input sample, h_c is the confidence of tagging the instance using the model, h_p is the marginals of the prediction, h_t is the hidden representation from the model. The Q-function $Q^{\pi}(s, a) \rightarrow R$ receives the state s and an action a as inputs and returns a reward as a result of the execution of a. The policy π has the aim at maximizing the reward of actions. In this work, the reward is defined as the difference in NER performance when adding a new sample (i.e., the action) to the training dataset. This is a similar setting to Fang et al. [33], who apply DQNs to find the best Active Learning policy.

RDANER[26] [174] uses a bootstrapping approach to obtain model predictions on easily-obtainable unlabeled data and retrain the model with the augmented weak dataset. Firstly, a general domain pre-trained language model $\mathcal{M}$ is fine-tuned on the few-shot labeled corpus available $\mathcal{D}^L$. Then, $\mathcal{M}$ is used to annotate an unsupervised dataset $\mathcal{D}^U$ to obtain a weakly annotated dataset $\mathcal{D}^U_{weak}$ which is then combined with $\mathcal{D}^L$ to get the augmented corpus. A threshold θ is used to filter out tags with low probabilities assigned by $\mathcal{M}$. This process is iterated until the achievement of an acceptable level of accuracy or the maximum number of iterations. Experiments show that this simple approach has worse F1-scores than domain-specific pre-trained language models (e.g.,

[26]RDANER [174] code: https://github.com/houking-can/RDANER

Table 5. Overview of Evaluations Conducted in the Reviewed Articles Included in the Self Learning Category

Method	Schema	Dev	CoNNL					WSJ	Sci-ERC		BC5CDR		NCBI-Disease		Twitter	OntoNotes			Webpage	WikiGold			WNUT17		MIT-M		MIT-R		SNIPS		ATIS		Multiwoz		I2B2	
			0	5	10%	100%	sub	100%	10%	100%	10%	100%	10%	100%	0	0	5	10%	0	0	5	10%	5	10%	5	10%	5	10%	5	10%	5	10%	5	10%	5	10%
LM-LSTM-CRF [84]	BIOES	✔	–	–	–	91.71	–	97.53	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Chen et al. [15]	N/A	✔	–	–	–	–	84.95	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
RDANER [174]	N/A	✔	–	–	–	–	–	–	58.83	68.96	78.25	87.38	78.14	87.89	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
BOND [79]	IOB	✖	81.48	–	–	–	–	–	–	–	–	–	–	–	48.01	68.35	–	–	65.74	60.07	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Huang et al. [57]	IOB	✔	–	65.4	89.5	–	–	–	–	–	–	–	–	–	–	–	71.1	86.7	–	–	68.4	75.9	37.6	50.5	55.9	66.6	51.3	74.1	83.0	94.2	90.5	90.3	22.5	84.1	39.3	87.1

When datasets have multiple entity types, we report averaged scores. Details of the datasets are reported in Appendix. The column *Schema* reports the annotation schema used, while *Dev* indicates whether a development set to choose hyperparameters has been used or not. For each dataset, we report the number of shots or the percentage of training data and the results obtained when available.

BioBERT [69], SciBERT [8]), but it could be a good alternative in low-resource languages and domains where huge domain-specific language model variants are not available.

BOND[27] [79] combines a self-training approach with distant supervision. During the first stage, distant labels are generated with external knowledge bases and a pre-trained BERT model is adapted to the distantly supervised NER task with early stopping. In the second stage, a teacher-student framework is employed, where the student model θ^{stu} is trained with pseudo-labels generated by the teacher model θ^{tea}. The teacher is initialized with weights $\hat{\theta}$ learned during the first stage, while the student may be initialized with $\hat{\theta}$ or pre-trained BERT layers θ^{BERT}. At the t-th iteration, the teacher generates weak labels which the student learns to fit. Then, teacher and student are updated $\theta^{tea}_{(t+1)} = \theta^{stu}_{(t+1)} = \hat{\theta}^{tea}_{(t)}$. In this way, pseudo-labels are progressively refined so that the student can exploit their knowledge and avoid overfitting.

Huang et al.[28] [57] leverage the labeled training set $\mathcal{D}^L$ and all the available in-domain unlabeled samples $\mathcal{D}^U$ by resorting to the knowledge-distillation approach proposed by Xie et al. [164] in image classification. The algorithm operates in three steps: (1) a teacher model θ^{tea} is learned with labeled tokens $\mathcal{D}^L$; (2) the teacher model is used to generate soft labels $\hat{\mathbf{y}}_\mathbf{i}$ on unlabeled tokens $\mathbf{x}_i \in \mathcal{D}^U$, $\hat{\mathbf{y}}_\mathbf{i} = f_{\theta^{tea}}(\mathbf{x}_i)$; (3) a student model θ^{stu} is trained on labeled and unlabeled tokens with a composed Kullback-Leibler loss $\mathcal{L}$:

$$\frac{1}{|\mathcal{D}^L|} \sum_{\mathbf{x}_i \in \mathcal{D}^L} \mathcal{L}(f_{\theta^{stu}}(\mathbf{x}_i), \mathbf{y}_i) + \frac{\lambda^U}{|\mathcal{D}^L|} \sum_{\hat{\mathbf{x}}_i \in \mathcal{D}^U} \mathcal{L}(f_{\theta^{stu}}(\hat{\mathbf{x}}_i), \hat{\mathbf{y}}_i), \tag{20}$$

where λ^U is a weighting hyper-parameter. Experimental results show that self-training usually results in significant performance improvements also when combined with noisy supervision.

4.4.1 Performance Evaluation. The experimental results shown in Table 5 highlight that the self-learning techniques discussed are the most commonly approaches used in zero-shot contexts among the ones reviewed in this work. However, due to the heterogeneity of the datasets, comparing techniques belonging to this category against each other is not always feasible. However, we can examine the results of BOND [79] on the CoNNL dataset [125] in a zero-shot context, which can be compared with those achieved by KALM [80], which attains superior performance thanks to the usage of dictionaries. In addition, the results obtained by BOND [79] in a zero-shot context are superior to those achieved by Hou et al. [56] in the Transfer Learning category as shown in Table 1. We also observe that BOND [79] achieves better performance in a zero-shot context as compared to ProtoNER [39], despite the latter method utilizing 20 instances for training as shown in Table 2. Similarly, RDANER's results on BC5CDR can be compared with COSINER in Table 3.

[27] BOND [79] code: https://github.com/cliang1453/BOND

[28] Huang et al. [57] code: https://github.com/few-shot-NER-benchmark/BaselineCode

Despite COSINER [7] utilizing fewer training data (i.e., 2% of the original dataset), it presents better performance than RDANER [174]. However, when dealing with 10% of NCBI-Disease [32], RDANER [174] performs better than COSINER [7].

Huang et al. [57] results for MIT-M [83] and MIT-R [82] can only be compared with those of Template-based NER [22] and PromptSlotTagging [55] in a 10-shot context. Huang et al [57] model obtains superior performance on MIT-M [83] with half the number of training samples and inferior results compared to Template-based NER [22] on MIT-R [82]. Furthermore, Huang et al. [57] model displays a superior performance on ATIS [49] than Template-based NER [22]. On SNIPS [21], the results outperform both Hou et al. [56] (Table 1) and L-TapNet [54] and Oguz et al. [105] (Table 2). On i2b2-2010 data [149], superior results as compared to Dai et al. [24] confirming that data augmentation alone has a lower effect on the performance of the model w.r.t. to other few-shot learning approaches.

The results show that the size and diversity of the training data, as well as the specific techniques utilized, have significant effects on the model's performance in zero-shot or few-shot contexts. These findings suggest a need for further research to investigate how to optimize the combination of training techniques, data variety, and model architecture to achieve improved results in these circumstances.

4.5 Summary and Discussion

Data-centric approaches for FS-NER try to make the most of the few available training samples or to leverage in the cleanest way possible an available but unannotated corpus, thanks to the intervention of humans, external resources, and/or the model itself.

When the available few-shot training corpus is the one and only source of information available, data augmentation can be usually applied since it increases the size of the dataset by transforming the available samples. However, the majority of research work assumes the presence of external resources such as a vocabulary of entities [176] or synonyms [25].

In general, an unannotated corpus is the key to achieve better results in few-shot contexts — methods basically differ on how they handle it to find greedy annotations. Active learning usually requires one (or many) human (s) to optimize the tradeoff between the annotation efforts required and the resulting performance. On the other hand, distant supervision leverages external resources, such as heuristic rules and ontologies, to obtain weak labels. Similarly, self-learning gets weak labels by the model itself, which could be particularly useful when using language models to leverage the knowledge they acquired during the pre-training stage.

5 CHALLENGES AND RESEARCH DIRECTIONS

In this section, we provide a discussion on the key aspects and research directions for the further development of the FS-NER state-of-the-art. Specifically, we critically evaluate the limitations of current FS-NER systems, identifying partially solved problems, as well as challenges that have yet to be addressed. Furthermore, we draw on a comprehensive examination of recent advancements in FS-NER and highlight the emerging trends and technologies that are gaining prominence.

5.1 Practical Applicability: How Many Shots do I Need?

As we can see in Figure 2, the number of training examples needed for few-shot learning to attain good performance might vary across different datasets and tasks. The complexity of the task, the diversity of the instances in the training set, and the similarity between the training and test sets are only a few of the factors that may contribute to this variability.

To learn effectively, tasks that are more complicated or call for a higher degree of abstraction would need more training instances. Several examples could be required to learn the pertinent

aspects, for instance, if the task entails identifying complicated patterns or mastering sophisticated decision rules.

The model may be able to acquire more generalizable features and need fewer examples to perform well if the training set includes a broad group of examples that span a wide range of variations. The model could need more examples if the training set is wider or more homogeneous so that it can learn enough variation.

Additionally, the similarity between the training and test sets can also play a role. If the test set is very different from the training set, the model may need more examples to generalize to new examples in the test set. On the other hand, if the test set is similar to the training set, the model may be able to achieve good performance with fewer examples.

Understanding the factors that affect the number of training examples required for FS-NER learning can help researchers develop new approaches to improve model performance in low-data regimes. While one potential area of research could focus on developing methods that can learn more complex and abstract representations from few examples, we argue that another possible direction could involve exploring ways to generate more diverse training sets, such as using generative models to augment existing examples or incorporating external knowledge sources to provide additional training data. While generative models and distant supervision have been already investigated in FS-NER, their ability in generating heterogeneous training sets has yet to be explored.

5.2 Performance Evaluation

Despite the growing number of research studies on FS-NER, comparing their performance is a challenge due to the lack of a unified evaluation protocol. With regards to the datasets used to experiment methods, we have identified a set of benchmark corpora used in most of the works: CoNLL-2003 [148], SNIPS [21], Ontonotes [111], WikiANN [107].

However, research in this field has not reached a consensus yet on the three aspects described below:

- *Hyper-parameters choice.* When building FS-NER systems, it should be kept in mind that the lack of data should also affect the development set on which hyper-parameters could be optimized. Hence, as in previous works [42, 127], we suggest to choose hyper-parameters based on literature and practical considerations, rather than on experiments made with a development set.
- *Annotation scheme.* There are many annotation schemes to build NER datasets, each with its advantages and disadvantages: the IO scheme is the simplest one and allows to distinguish tokens that are at the *inside* and *outside* of entity mentions, but it fails to detect consecutive but different entities; to solve this, the IOB scheme adds a B label identifying the *beginning* of entity mentions, while the IOBES scheme enriches the information about boundaries by adding E and S labels to distinguish *ending* tokens and *single*-token entities, respectively, which could be useful to detect nested entities. However, the choice of the annotation scheme has a strong impact on the resulting performance of NER systems. In the FS-NER area, despite the lack of an in-depth study on their differences, the most adopted annotation schemes are IO and IOB, probably due to their simplicity. Nested entities are indeed rare and quite difficult to have available in a few-shot training corpus, but this means that the identification of nested entities is not possible with current FS-NER methods, which could be an interesting focus of future research.
- *Few-shot scenarios.* FS-NER experiments are usually run by simulating scenarios with scarce resources. The most two common approaches consist in taking into consideration a pre-defined number of training samples (e.g., 1-shot, 10-shots, 50-shots experiments), or a

pre-defined percentage of the original training set (e.g., 1%, 10%). Since the second approach may be confusing due to the fact that original training sets usually have a different size, we suggest future research to rely on the first one. Related to this, we suggest future works also to study the impact of different shots choices: since performance is strongly dependant on the training examples used to train the model, it may heavily vary based on which examples we select, thus hindering the robustness of experimental conclusions.

5.3 Applications

Although there is a growing body of research on FS-NER, the applicability of the proposed techniques in real-world scenarios where the rareness of training samples is involved has never been studied in-depth. Current works experimenting their techniques with real use cases usually focus on healthcare: Wang et al. [159] use Chinese EHRs data from an affiliated hospital with 1600 de-identified clinical notes from departments of cardiology, respiratory, neurology and gastroenterology; La Gatta et al. [42] experiment with an Italian dataset of cardiological clinical notes. However, we argue that the need for FS-NER methodology is more widespread and future works should also focus to other domain (e.g., finance). To the best of our knowledge, the only other work using FS-NER approaches to solve a real-world problem is from Ni et al. [102] who create a custom multi-lingual dataset (i.e., Japanese, Korean, German, Portuguese) with over 50 entity types to build cognitive question-answering applications.

5.4 The Ever-increasing Performance of Generative Models

The progress of **Natural Language Processing** (**NLP**) has been propelled by the creation of generative **Large Language Models** (**LLMs**), which can be seen in studies such as Devlin et al. [30], Brown et al. [11], and Chowdhery et al. [19]. LLMs have already become a crucial component of many commonly used products, such as the coding assistant Copilot [17], the Google search engine,[29] and the more recent addition of ChatGPT.[30] However, it is not immediately clear how these models may impact future research works in few-shot **Named Entity Recognition** (**NER**).

One potential way that generative models could impact FS-NER is by generating synthetic training data. Synthetic data can be used to augment limited training datasets, which is particularly useful in few-shot learning scenarios. By using generative models to generate synthetic examples, it may be possible to improve the ability of FS-NER models to generalize to new, unseen entities. StyleNER [18] (described in Section 4.1), is a first attempt to use generative models for FS-NER, but it relies on external data to learn their patterns and generate new samples that imitate their style. This poses many challenges on the choice of the external data to be used and their effectiveness on the few-shot problem at hand.

Another way that generative models could impact FS-NER is through their ability to generate context-aware representations. Named entities are often highly dependent on the context in which they appear, and the ability to generate context-aware representations may be useful for FS-NER models. By leveraging the power of generative models, it may be possible to learn more nuanced representations of named entities and their associated contexts.

Overall, there is still much work to be done to fully investigate the possible effects of generative models on FS-NER. Yet, it is evident that these models have a lot of potential for future advancements in FS-NER and other natural language processing tasks. The optimal techniques to utilize generative models in the context of few-shot learning will need to be determined through thorough experimentation and review, as with any new technology.

[29] https://blog.google/products/search/search-language-understanding-bert/
[30] https://openai.com/blog/chatgpt/

5.5 A Focus on Developmental Approaches

While standard procedures to train deep-learning models have shown robust performance improvements over time, there is an increasing research interest toward *developmental approaches* [64, 106, 165, 182, 184]. Smith et al. [137] show that two-year old children can infer a new category from only one instance. This is presumably due to the fact that the human brain, during early learning, is trained to develop foundational structures to learn the learning procedure [68], which is similar to the key-idea behind meta-learning. Instead of learning from scratch, neural networks should be able to recognize regularities from past tasks to help them solving new tasks [64]. Due to these parallelisms, current research in developmental psychology could guide future works in FS-NER. For example, Orhan et al. [106] and Xing et al. [165] leverage experiments showing that learning object names can change the visual features children use for word learning, thus recognizing that language helps recognizing new visual objects.

6 CONCLUSION

This survey provides a comprehensive review of state-of-the-art algorithms for few-shot Named Entity Recognition. We analyze and discuss this application field in detail and propose a taxonomy to summarize existing techniques into two macro-categories: model-centric and data-centric. According to the way models are defined and data are manipulated to handle the few-shot learning task, we further categorize different methods in each macro-category into subgroups. Algorithms in each subgroup are analyzed and lined-up as a story to highlight advantages, disadvantages and to help readers understand how research is moving toward future directions. Not only does our categorization and analysis offer a clear and comprehensive understanding of existing methods in the field, but it also provides useful resources for practitioners to select the most suitable technique suiting their needs, and for researchers to advance the state-of-the-art on few-shot NER.

APPENDIX

A BENCHMARK DATASETS

In Table 6 we report details about the datasets used to assess the performance of methods reviewed in this work.

Table 6. Datasets used in Evaluations of FS-NER Approaches (Listed in Alphabetical Order)

Dataset	Ref.	Year	Description
ACE2005	link[1]	2006	Text documents in multiple languages, including English, Arabic, and Chinese, along with annotations for named entities, their attributes, and relations between them.
AMT	[122]	2014	Crowd-annotation dataset based on the 2003 CoNNL shared NER task collected using Amazon's Mechanical Turk.
ATIS	[49]	2016	Audio recordings and corresponding manual transcripts about airline travel information dialogues with 17 unique intent categories.
BC2GM	[138]	2008	Abstracts from the MEDLINE database annotated with gene entity mentions.
BC5CDR	[77]	2016	PubMed articles annotated with chemical-disease relations.
CLUENER	[166]	2020	Text samples from social media, news articles and other sources annotated with person names, locations, organizations, as well as entity types such as time and quantity.
CM-NER	[159]	2018	De-identified EHRs from four different specialties (i.e., cardiology, respiratory, neurology and gastroenterology).
CNER	link[2]	2019	Chinese clinical NER dataset including anatomy, disease, imaging examination, laboratory examination, drug and operation entities.
CoNLL	[125]	2003	News articles annotated with person names, organization and locations.
GUM	[175]	2017	Conversations, news articles, fiction, academic papers linguistically annotated at multiple levels (part-of-speech tags, syntactic parse trees, named entities and coreference chains).
I2B2-2010	[149]	2011	Clinical notes annotated with various medical entities including diseases, symptoms, treatments and tests.
ICON 2013	link[3]	2013	Documents in Indian languages (Bengali, Hindi, Marathi, Tamil and Telugu) annotated with some predefined categories of interest such as Person, Location, Organization, Miscellaneous (e.g., date, time).
IDiag	[177]	2020	Health record images converted into text paragraph by OCR and then annotated with diagnoses.
LaptopReview	[110]	2014	Focuses on laptop aspect term (e.g.,, "disk drive") recognition.
MaSciP	[100]	2019	Synthesis procedures annotated with synthesis operations and their typed arguments (e.g., Material, Synthesis-Apparatus).
MIT-M	[83]	2013	Samples annotated with named entity types related to movies (Actor, Award, CharacterName, Director, Genre, Opinion, Origin, Plot, Quote, Relationship, Soundtrack, Year).
MIT-R	[82]	2013	Samples annotated with named entity types related to restaurant reviews (i.e., Rating, Amenity, Location, RestaurantName, Price, Hours, Dish, Cuisine).
Multiwoz	[12]	2018	Dialogues between a user and a virtual assistant covering multiple domains (e.g., restaurants, hotels, transportation).
NCBI-Disease	[32]	2016	PubMed abstracts annotated with disease mentions.
NEWS	[71]	2006	Data from news sources provided by Microsoft Research Asia.
OntoNotes	link[4]	2013	News, telephone speech, weblogs, usenet newsgroups, broadcast and talk shows in three languages (English, Chinese, Arabic)
Sci-ERC	[88]	2018	Scientific papers and abstracts annotated with information related to biomedical events such as experiments, observations and findings.
SNIPS	[21]	2018	Spoken queries covering multiple domains including music, weather and navigation.
TOP	[48]	2018	Navigation and event search samples, where the 35% of the utterances contain multiple, nested intent labels.
Twitter	[45]	2015	Dataset from the WNUT 2016 NER shared task consisting of 2400 tweets with 10 entity types.
T-Twitter	[121]	2020	Tweets from 2014 to 2019 annotated with persons, locations and organizations.
W2019-food	[14]	2019	Samples from the 20190120 Wikipedia dump where "Food and drink" category entities are separated in Drinks, Meat, Vegetables, Condiments and Breads entity types.
W2019-language	[14]	2019	Samples from the 20190120 Wikipedia dump in low-resource languages (i.e., Welsh, Bengali, Yoruba, Mongolian and Egyptian Arabic).
Webpage	[117]	2009	20 personal, academic, and computer science conference webpages covering 783 entities belonging to the same four types as in CoNLL03
WikiANN	[107]	2017	cross-lingual name tagging and linking based on Wikipedia articles in 295 languages.
WikiGold	[5]	2009	Wikipedia articles randomly selected from a 2008 English dump and manually annotated with the four CoNLL03 entity types.
WNUT17	[29]	2017	Dataset focused on the identification of unusual, previously unseen entities in the context of emerging discussions.
WSJ	[91]	1993	Wall Street Journal portion of Penn Treebank dataset containing 25 sections and categorizing each word into 45 POS tags.

[1] https://catalog.ldc.upenn.edu/LDC2006T06
[2] http://www.ccks2019.cn/?pageid=62
[3] http://ltrc.iiit.ac.in/icon/2013/nlptools/index.html
[4] https://catalog.ldc.upenn.edu/LDC2013T19

REFERENCES

[1] David Ifeoluwa Adelani, Michael A. Hedderich, Dawei Zhu, Esther van den Berg, and Dietrich Klakow. 2020. Distant supervision and noisy label learning for low resource named entity recognition: A study on Hausa and Yorùbá. ICLR Workshops (AfricaNLP & PML4DC 2020), Apr 2020, Addis Ababa, Ethiopia. hal-03359111.

[2] Tareq Al-Moslmi, Marc Gallofré Ocaña, Andreas L. Opdahl, and Csaba Veres. 2020. Named entity extraction for knowledge graphs: A literature overview. *IEEE Access* 8 (2020), 32862–32881. DOI: https://doi.org/10.1109/ACCESS.2020.2973928

[3] Diego Mollá Aliod, M. Zaanen, and Daniel Smith. 2006. Named entity recognition for question answering. In *Proceedings of the Australasian Language Technology Workshop 2006*.

[4] P. Awasthi, Maria-Florina Balcan, and Philip M. Long. 2017. The power of localization for efficiently learning linear separators with noise. *Journal of the ACM* 63, 6 (2017), 1–27. https://doi.org/10.1145/3006384

[5] Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R. Curran. 2009. Named entity recognition in wikipedia. In *Proceedings of the 1st 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources@IJCNLP 2009*. Iryna Gurevych and Torsten Zesch (Eds.), Association for Computational Linguistics, 10–18. Retrieved from https://aclanthology.org/W09-3302/

[6] Maria-Florina Balcan, A. Beygelzimer, and J. Langford. 2009. Agnostic active learning. *Journal of Computer and System Sciences* 75, 1 (2009), 78–89.

[7] Ilaria Bartolini, Vincenzo Moscato, Marco Postiglione, Giancarlo Sperlì, and Andrea Vignali. 2022. COSINER: COntext SImilarity data augmentation for named entity recognition. In *Proceedings of the International Conference on Similarity Search and Applications*. Tomás Skopal, Fabrizio Falchi, Jakub Lokoc, Maria Luisa Sapino, Ilaria Bartolini, and Marco Patella (Eds.), Lecture Notes in Computer Science, Vol. 13590, Springer, 11–24. DOI: https://doi.org/10.1007/978-3-031-17849-8_2

[8] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.), Association for Computational Linguistics, 3613–3618. DOI: https://doi.org/10.18653/v1/D19-1371

[9] A. Beygelzimer, S. Dasgupta, and J. Langford. 2009. Importance weighted active learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*.

[10] I. Bondarenko, S. Berezin, A. Pauls, T. Batura, Y. Rubtsova, and B. Tuchinov. 2020. Using few-shot learning techniques for named entity recognition and relation extraction. In *Proceedings of the 2020 Science and Artificial Intelligence Conference*. 58–65. DOI: https://doi.org/10.1109/S.A.I.ence50533.2020.9303192

[11] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever,and Dario Amodei. 2020. Language models are few-shot learners. arXiv:2005.14165. Retrieved from https://arxiv.org/abs/2005.14165

[12] Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. MultiWOZ - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.), Association for Computational Linguistics, 5016–5026. Retrieved from https://aclanthology.org/D18-1547/

[13] Hengyi Cai, Hongshen Chen, Yonghao Song, Cheng Zhang, Xiaofang Zhao, and Dawei Yin. 2020. Data manipulation: Towards effective instance learning for neural dialogue generation via learning to augment and reweight. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 6334–6343. DOI: https://doi.org/10.18653/v1/2020.acl-main.564

[14] Yixin Cao, Zikun Hu, Tat-seng Chua, Zhiyuan Liu, and Heng Ji. 2019. Low-resource name tagging learned with weakly labeled data. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, Hong Kong, China, 261–270. DOI: https://doi.org/10.18653/v1/D19-1025

[15] Chenhua Chen and Yue Zhang. 2018. Learning how to self-learn: Enhancing self-training using neural reinforcement learning. *2018 International Conference on Asian Language Processing* (2018), 25–30.

[16] Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *SIGKDD Explor.* 19, 2 (2017), 25–35. DOI: https://doi.org/10.1145/3166054.3166058

[17] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz

Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *CoRR* abs/2107.03374 (2021). arXiv:2107.03374. Retrieved from https://arxiv.org/abs/2107.03374

[18] Shuguang Chen, Gustavo Aguilar, Leonardo Neves, and Thamar Solorio. 2021. Data augmentation for cross-domain named entity recognition. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.), Association for Computational Linguistics, 5346–5356. DOI: https://doi.org/10.18653/v1/2021.emnlp-main.434

[19] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling language modeling with pathways. arXiv:2204.02311. Retrieved from https://arxiv.org/abs/2204.02311

[20] Ryan Cotterell and Kevin Duh. 2017. Low-resource named entity recognition with cross-lingual, character-level neural conditional random fields. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Asian Federation of Natural Language Processing, Taipei, Taiwan, 91–96. Retrieved from https://aclanthology.org/I17-2016

[21] Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces. arXiv:1805.10190. Retrieved from https://arxiv.org/abs/1805.10190

[22] Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Online, 1835–1845. DOI: https://doi.org/10.18653/v1/2021.findings-acl.161

[23] Aron Culotta and Andrew McCallum. 2005. Reducing labeling effort for structured prediction tasks. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*. AAAI Press, 746–751.

[24] Hongliang Dai, Donghong Du, Xin Li, and Yangqiu Song. 2019. Improving fine-grained entity typing with entity linking. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, Hong Kong, China, 6210–6215. DOI: https://doi.org/10.18653/v1/D19-1643

[25] Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*. Donia Scott, Núria Bel, and Chengqing Zong (Eds.), International Committee on Computational Linguistics, 3861–3867. DOI: https://doi.org/10.18653/v1/2020.coling-main.343

[26] Sandipan Dandapat and Andy Way. 2016. Improved named entity recognition using machine translation-based cross-lingual information. *Computación y Sistemas* 20, 3 (2016), 495–504. DOI: https://doi.org/10.13053/cys-20-3-2468

[27] S. Dasgupta, A. Kalai, and C. Monteleoni. 2005. Analysis of perceptron-based active learning. In *Proceedings of the International Conference on Computational Learning Theory*.

[28] Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*. Association for Computational Linguistics, Copenhagen, Denmark, 140–147. DOI: https://doi.org/10.18653/v1/W17-4418

[29] Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*. Leon Derczynski, Wei Xu, Alan Ritter, and Tim Baldwin (Eds.), Association for Computational Linguistics, 140–147. DOI: https://doi.org/10.18653/v1/w17-4418

[30] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies. Jill Burstein, Christy Doran, and Thamar Solorio (Eds.), Association for Computational Linguistics, 4171–4186. DOI: https://doi.org/10.18653/v1/n19-1423

[31] Georgiana Dinu and Marco Baroni. 2015. Improving zero-shot learning by mitigating the hubness problem. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). arXiv:1412.6568. Retrieved from http://arxiv.org/abs/1412.6568

[32] Rezarta Islamaj Dogan, Robert Leaman, and Zhiyong Lu. 2014. NCBI disease corpus: A resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics* 47 (2014), 1–10. DOI: https://doi.org/10.1016/j.jbi.2013.12.006

[33] Meng Fang, Yuan Li, and Trevor Cohn. 2017. Learning how to active learn: A deep reinforcement learning approach. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 595–605. DOI: https://doi.org/10.18653/v1/D17-1063

[34] Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement learning for relation classification from noisy data. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence, the 30th innovative Applications of Artificial Intelligence, and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence*. Sheila A. McIlraith and Kilian Q. Weinberger (Eds.), AAAI Press, 5779–5786. Retrieved from https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17151

[35] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning* Doina Precup and Yee Whye Teh (Eds.), PMLR, 1126–1135. Retrieved from http://proceedings.mlr.press/v70/finn17a.html

[36] Chelsea Finn, A. Rajeswaran, Sham M. Kakade, and S. Levine. 2019. Online meta-learning. In *Proceedings of the International Conference on Machine Learning*.

[37] Joseph Fisher and Andreas Vlachos. 2019. Merge and label: A novel neural network architecture for nested NER. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*. Anna Korhonen, David R. Traum, and Lluís Màrquez (Eds.), Association for Computational Linguistics, 5840–5850. DOI: https://doi.org/10.18653/v1/p19-1585

[38] Jason Fries, Sen Wu, Alex Ratner, and Christopher Ré. 2017. SwellShark: A generative model for biomedical named entity recognition without labeled data. arXiv:1704.06360. Retrieved from https://arxiv.org/abs/1704.06360

[39] Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. 2019. Few-shot classification in named entity recognition task. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. ACM, Limassol Cyprus, 993–1000. DOI: https://doi.org/10.1145/3297280.3297378

[40] Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Proceedings of the Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016*. Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (Eds.), 1019–1027. Retrieved from https://proceedings.neurips.cc/paper/2016/hash/076a0c97d09cf1a0ec3e19c7f2529f2b-Abstract.html

[41] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning*. Doina Precup and Yee Whye Teh (Eds.), PMLR, 1183–1192. Retrieved from http://proceedings.mlr.press/v70/gal17a.html

[42] Valerio La Gatta, Vincenzo Moscato, Marco Postiglione, and Giancarlo Sperlì. 2021. Few-shot Named Entity Recognition with Cloze Questions. arXiv:2111.12421. Retrieved from https://arxiv.org/abs/2111.12421

[43] Athanasios Giannakopoulos, C. Musat, Andreea Hossmann, and Michael Baeriswyl. 2017. Unsupervised aspect term extraction with B-LSTM & CRF using automatically labelled datasets. In *Proceedings of the WASSA@EMNLP*.

[44] Michael R. Glass, Alfio Gliozzo, Rishav Chakravarti, Anthony Ferritto, Lin Pan, G. P. Shrivatsa Bhargav, Dinesh Garg, and Avirup Sil. 2020. Span selection pre-training for question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.), Association for Computational Linguistics, 2773–2782. DOI: https://doi.org/10.18653/v1/2020.acl-main.247

[45] Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia lab @ ACL WNUT NER shared task: Named entity recognition for twitter microposts using distributed word representations. In *Proceedings of the Workshop on Noisy User-generated Text*. Association for Computational Linguistics, Beijing, China, 146–153. DOI: https://doi.org/10.18653/v1/W15-4322

[46] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alex Smola. 2012. A kernel two-sample test. *Journal of Machine Learning Research* 13, 25 (2012), 723–773.

[47] Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.), Association for Computational Linguistics, 3622–3631. DOI: https://doi.org/10.18653/v1/d18-1398

[48] Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.), Association for Computational Linguistics, 2787–2792. DOI: https://doi.org/10.18653/v1/d18-1300

[49] Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM. In *Proceedings of the Interspeech 2016, 17th Annual Conference of the International Speech Communication Association*. Nelson Morgan (Ed.), ISCA, 715–719. DOI: https://doi.org/10.21437/Interspeech.2016-402

[50] Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2020. Revisiting self-training for neural sequence generation. In *Proceedings of the 8th International Conference on Learning Representations*. OpenReview.net. Retrieved from https://openreview.net/forum?id=SJgdnAVKDH

[51] W. He. 2017. Autoentity: Automated entity detection from massive text corpora. https://hdl.handle.net/2142/97395

[52] Matthew Henderson and Ivan Vulić. 2021. ConVEx: Data-efficient and few-shot slot labeling. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 3375–3389. DOI: https://doi.org/10.18653/v1/2021.naacl-main.264

[53] Maximilian Hofer, Andrey Kormilitzin, Paul Goldberg, and Alejo J. Nevado-Holgado. 2018. Few-shot learning for named entity recognition in medical text. arXiv:1811.05468. Retrieved from http://arxiv.org/abs/1811.05468

[54] Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 1381–1393. DOI: https://doi.org/10.18653/v1/2020.acl-main.128

[55] Yutai Hou, Cheng Chen, Xianzhen Luo, Bohan Li, and Wanxiang Che. 2022. Inverse is better! fast and accurate prompt for few-shot slot tagging. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics, Dublin, Ireland, 637–647. DOI: https://doi.org/10.18653/v1/2022.findings-acl.53

[56] Yutai Hou, Zhihan Zhou, Yijia Liu, Ning Wang, Wanxiang Che, Han Liu, and Ting Liu. 2019. Few-shot sequence labeling with label dependency transfer and pair-wise embedding. arXiv:1906.08711. Retrieved from https://arxiv.org/abs/1906.08711

[57] Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2021. Few-shot named entity recognition: An empirical baseline study. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.), Association for Computational Linguistics, 10408–10423. DOI: https://doi.org/10.18653/v1/2021.emnlp-main.813

[58] Po-Sen Huang, Chenglong Wang, Rishabh Singh, Wen-tau Yih, and Xiaodong He. 2018. Natural language to structured query generation via meta-learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Marilyn A. Walker, Heng Ji, and Amanda Stent (Eds.), Association for Computational Linguistics, 732–738. DOI: https://doi.org/10.18653/v1/n18-2115

[59] Weizhi Huang, Ming He, and Yongle Wang. 2021. A survey on meta-learning based few-shot classification. In *International Conference on Machine Learning and Intelligent Communications*. Xiaolin Jiang (Ed.), Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Vol. 438, Springer, 243–253. DOI: https://doi.org/10.1007/978-3-031-04409-0_23

[60] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. DOI: https://doi.org/10.48550/ARXIV.1508.01991

[61] Saumya Jetley, Nicholas A. Lord, Namhoon Lee, and Philip H. S. Torr. 2018. Learn to pay attention. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. Retrieved from https://openreview.net/forum?id=HyzbhfWRW

[62] Michael I. Jordan (Ed.). 1998. *Learning in Graphical Models*. NATO ASI Series, Vol. 89. Springer Netherlands. DOI: https://doi.org/10.1007/978-94-011-5014-9

[63] Sungchul Kim, Kristina Toutanova, and H. Yu. 2012. Multilingual named entity recognition using parallel data and metadata from wikipedia. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

[64] Taesup Kim, Jaesik Yoon, Ousmane Amadou Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. 2018. Bayesian model-agnostic meta-learning. In *Proceedings of the Advances in Neural Information Processing Systems*.

[65] Jason Krone, Yi Zhang, and Mona Diab. 2020. Learning to classify intents and slot labels given a handful of examples. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*. Association for Computational Linguistics, Online, 96–108. DOI: https://doi.org/10.18653/v1/2020.nlp4convai-1.12

[66] Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *Proceedings of the 6th International Conference on Learning Representations*. OpenReview.net. Retrieved from https://openreview.net/forum?id=H196sainb

[67] Ouyu Lan, Xiao Huang, Bill Yuchen Lin, He Jiang, Liyuan Liu, and Xiang Ren. 2020. Learning to contextually aggregate multi-source supervision for sequence labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.), Association for Computational Linguistics, 2134–2146. DOI: https://doi.org/10.18653/v1/2020.acl-main.193

[68] Barbara Landau, Linda B. Smith, and Susan S. Jones. 1988. The importance of shape in early lexical learning. *Cognitive Development* 36, 4 (1988), 299–321.

[69] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* (2019), btz682. DOI: https://doi.org/10.1093/bioinformatics/btz682

[70] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 4 (2020), 1234–1240.

[71] Gina-Anne Levow. 2006. The third international chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the 5th Workshop on Chinese Language Processing*. Hwee Tou Ng and Olivia O. Y. Kwong (Eds.), Association for Computational Linguistics, 108–117. Retrieved from https://aclanthology.org/W06-0115/

[72] David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the SIGIR '94*.

[73] Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020. Pre-training via paraphrasing. In *Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*. Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.), Retrieved from https://proceedings.neurips.cc/paper/2020/hash/d6f1dd034aabde7657e6680444ceff62-Abstract.html

[74] Jing Li, Shuo Shang, and Ling Shao. 2020. MetaNER: Named entity recognition with meta-learning. In *Proceedings of The Web Conference 2020*. ACM, Taipei Taiwan, 429–440. DOI: https://doi.org/10.1145/3366423.3380127

[75] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering* 34, 1 (2020), 1–1. DOI: https://doi.org/10.1109/TKDE.2020.2981314

[76] J. Li, A. Sun, J. Han, and C. Li. 2022. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering* 34, 1 (2022), 50–70. DOI: 10.1109/TKDE.2020.2981314

[77] Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. BioCreative V CDR task corpus: A resource for chemical disease relation extraction. *Database J. Biol. Databases Curation* 2016 (2016). https://academic.oup.com/database/article/doi/10.1093/database/baw068/2630414

[78] Y. Li, Yongxin Yang, W. Zhou, and Timothy M. Hospedales. 2019. Feature-critic networks for heterogeneous domain generalization. In *Proceedings of the International Conference on Machine Learning*.

[79] Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. BOND: BERT-assisted open-domain named entity recognition with distant supervision. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (Eds.), ACM, 1054–1064. DOI: https://doi.org/10.1145/3394486.3403149

[80] Angli Liu, Jingfei Du, and Veselin Stoyanov. 2019. Knowledge-augmented language model and its application to unsupervised named-entity recognition. 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 1142–1150. DOI: https://doi.org/10.18653/v1/N19-1117

[81] Bing Liu, Xuchu Yu, Anzhu Yu, Pengqiang Zhang, Gang Wan, and Ruirui Wang. 2019. Deep few-shot learning for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing* 57, 4 (2019), 2290–2304. DOI: https://doi.org/10.1109/TGRS.2018.2872830

[82] Jingjing Liu, Panupong Pasupat, Scott Cyphers, and James R. Glass. 2013. Asgard: A portable architecture for multilingual dialogue systems. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 8386–8390. DOI: https://doi.org/10.1109/ICASSP.2013.6639301

[83] Jingjing Liu, Panupong Pasupat, Yining Wang, Scott Cyphers, and James R. Glass. 2013. Query understanding enhanced by hierarchical parsing structures. In *Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 72–77. DOI: https://doi.org/10.1109/ASRU.2013.6707708

[84] Liyuan Liu, Jingbo Shang, Xiang Ren, Frank Fangzheng Xu, Huan Gui, Jian Peng, and Jiawei Han. 2018. Empower sequence labeling with task-aware neural language model. In *Proceedings of the 32nd AAAI Conference on Artificial*

Intelligence, the 30th innovative Applications of Artificial Intelligence, and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence. Sheila A. McIlraith and Kilian Q. Weinberger (Eds.), AAAI Press, 5253–5260. Retrieved from https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17123

[85] Ruibo Liu, Guangxuan Xu, Chenyan Jia, Weicheng Ma, Lili Wang, and Soroush Vosoughi. 2020. Data boost: Text data augmentation through reinforcement learning guided conditional generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online, 9031–9041. DOI: https://doi.org/10.18653/v1/2020.emnlp-main.726

[86] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering* 35, 1 (2021), 1–1. DOI: https://doi.org/10.1109/TKDE.2021.3090866

[87] Y. Lou, T. Qian, F. Li, and D. Ji. 2020. A graph attention model for dictionary-guided named entity recognition. *IEEE Access* 8 (2020), 71584–71592. DOI: 10.1109/ACCESS.2020.2987399

[88] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.), Association for Computational Linguistics, 3219–3232. DOI: https://doi.org/10.18653/v1/d18-1360

[89] Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 1064–1074. DOI: https://doi.org/10.18653/v1/P16-1101

[90] Alexandre Magueresse, Vincent Carles, and Evan Heetderks. 2020. Low-resource languages: A review of past work and future challenges. arXiv:2006.07264. Retrieved from https://arxiv.org/abs/2006.07264

[91] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguistics* 19, 2 (1993), 313–330.

[92] Tomás Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. arXiv:1309.4168. Retrieved from https://arxiv.org/abs/1309.4168

[93] Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 2339–2352. DOI: https://doi.org/10.18653/v1/2020.acl-main.212

[94] Mike D. Mintz, Steven Bills, R. Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*.

[95] Tom M. Mitchell. 1997. *Machine Learning, International Edition*. McGraw-Hill. Retrieved from https://www.worldcat.org/oclc/61321007

[96] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charlie Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518 (2015), 529–533.

[97] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2012. *Foundations of Machine Learning*. MIT Press. Retrieved from http://mitpress.mit.edu/books/foundations-machine-learning-0

[98] Aldrian Obaja Muis and Wei Lu. 2017. Labeling gaps between words: Recognizing overlapping mentions with mention separators. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Martha Palmer, Rebecca Hwa, and Sebastian Riedel (Eds.), Association for Computational Linguistics, 2608–2618. DOI: https://doi.org/10.18653/v1/d17-1276

[99] Subhabrata Mukherjee and Ahmed Hassan Awadallah. 2020. Uncertainty-aware self-training for text classification with few labels. arXiv:2006.15315. Retrieved from https://arxiv.org/abs/2006.15315

[100] Sheshera Mysore, Zach Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey Flanigan, Andrew McCallum, and Elsa Olivetti. 2019. The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic structures. In *Proceedings of the 13th Linguistic Annotation Workshop*. Annemarie Friedrich, Deniz Zeyrek, and Jet Hoek (Eds.), Association for Computational Linguistics, 56–64. DOI: https://doi.org/10.18653/v1/w19-4007

[101] Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. 2021. Named entity recognition and relation extraction: State-of-the-art. *ACM Computing Surveys* 54, 1 (2021), 39 pages. DOI: https://doi.org/10.1145/3445965

[102] Jian Ni, Georgiana Dinu, and Radu Florian. 2017. Weakly supervised cross-lingual named entity recognition via effective annotation and representation projection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Regina Barzilay and Min-Yen Kan (Eds.), Association for Computational Linguistics, 1470–1480. DOI: https://doi.org/10.18653/v1/P17-1135

[103] A. Obamuyide and A. Vlachos. 2019. Model-agnostic meta-learning for relation classification with limited supervision. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

[104] F. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics* 29, 1 (2003), 19–51.

[105] Cennet Oguz and Ngoc Thang Vu. 2021. Few-shot learning for slot tagging with attentive relational network. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, Online, 1566–1572. DOI: https://doi.org/10.18653/v1/2021.eacl-main.134

[106] A. Emin Orhan, Vaibhav Gupta, and Brenden M. Lake. 2020. Self-supervised learning through the eyes of a child. *Advances in Neural Information Processing Systems* 33 (2020), 9960–9971

[107] Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 1946–1958. DOI: https://doi.org/10.18653/v1/P17-1178

[108] Hima Patel, Shanmukha C. Guttula, Ruhi Sharma Mittal, Naresh Manwani, Laure Berti-Équille, and Abhijit Manatkar. 2022. Advances in exploratory data analysis, visualisation and quality for data centric AI systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Aidong Zhang and Huzefa Rangwala (Eds.), ACM, 4814–4815. DOI: https://doi.org/10.1145/3534678.3542604

[109] Nanyun Peng and Mark Dredze. 2017. Multi-task domain adaptation for sequence tagging. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*. Association for Computational Linguistics, Vancouver, Canada, 91–100. DOI: https://doi.org/10.18653/v1/W17-2612

[110] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation*. Preslav Nakov and Torsten Zesch (Eds.), The Association for Computer Linguistics, 27–35. DOI: https://doi.org/10.3115/v1/s14-2004

[111] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the 17th Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Sofia, Bulgaria, 143–152. Retrieved from https://aclanthology.org/W13-3516

[112] Kun Qian and Z. Yu. 2019. Domain adaptive dialog generation via meta learning. In *Proceedings of the ACL*.

[113] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[114] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* 21, 1 (2020), 67 pages.

[115] Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 151–164. DOI: https://doi.org/10.18653/v1/P19-1015

[116] A. Rajeswaran, Chelsea Finn, Sham M. Kakade, and S. Levine. 2019. Meta-learning with implicit gradients. In *Proceedings of the Advances in Neural Information Processing Systems*.

[117] Lev-Arie Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the 13th Conference on Computational Natural Language Learning*. Suzanne Stevenson and Xavier Carreras (Eds.), ACL, 147–155. Retrieved from https://aclanthology.org/W09-1119/

[118] S. Ravi and H. Larochelle. 2017. Optimization as a model for few-shot learning. In *Proceedings of the International Conference on Learning Representations*.

[119] Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics, 4512–4525.

[120] Xiang Ren, Ahmed El-Kishky, C. Wang, Fangbo Tao, Clare R. Voss, and Jiawei Han. 2015. ClusType: Effective entity recognition and typing by relation phrase-based clustering. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[121] Shruti Rijhwani and Daniel Preotiuc-Pietro. 2020. Temporally-informed analysis of named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.), Association for Computational Linguistics, 7605–7617. DOI: https://doi.org/10.18653/v1/2020.acl-main.680

[122] Filipe Rodrigues, Francisco C. Pereira, and Bernardete Ribeiro. 2014. Sequence labeling with multiple annotators. *Machine Learning* 95, 2 (2014), 165–181. DOI: https://doi.org/10.1007/s10994-013-5411-2

[123] Esteban Safranchik, Shiying Luo, and Stephen H. Bach. 2020. Weakly supervised sequence tagging from noisy rules. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[124] Miguel G. San-Emeterio. 2022. A survey on few-shot techniques in the context of computer vision applications based on deep learning. In *Proceedings of the International Conference on Image Analysis and Processing*. Pier Luigi Mazzeo, Emanuele Frontoni, Stan Sclaroff, and Cosimo Distante (Eds.), Lecture Notes in Computer Science, Vol. 13374, Springer, 14–25. DOI: https://doi.org/10.1007/978-3-031-13324-4_2

[125] Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the 7th Conference on Natural Language Learning*. Walter Daelemans and Miles Osborne (Eds.), ACL, 142–147. Retrieved from https://aclanthology.org/W03-0419/

[126] A. Santoro, Sergey Bartunov, M. Botvinick, Daan Wierstra, and T. Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *Proceedings of the International Conference on Machine Learning*.

[127] Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty (Eds.), Association for Computational Linguistics, 255–269. DOI: https://doi.org/10.18653/v1/2021.eacl-main.20

[128] J. Schmidhuber. 1994. On learning how to learn learning strategies. https://people.idsia.ch/~juergen/FKI-198-94ocr.pdf

[129] Elisa Terumi Rubel Schneider, João Vitor Andrioli de Souza, Julien Knafou, L. E. S. Oliveira, Jenny Copara, Yohan Bonescki Gumiel, Lucas Ferro Antunes de Oliveira, E. Paraiso, D. Teodoro, and Claudia Maria Cabral Moro Barra. 2020. BioBERTpt - A portuguese neural language model for clinical named entity recognition. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*.

[130] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. arXiv:1707.06347. Retrieved from https://arxiv.org/abs/1707.06347

[131] H. J. Scudder. 1965. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory* 11, 3 (1965), 363–371.

[132] Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1070–1079.

[133] Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R. Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering* 30, 10 (2018), 1825–1837.

[134] Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. 2018. Learning named entity tagger using domain-specific dictionary. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.), Association for Computational Linguistics, 2054–2064. DOI: https://doi.org/10.18653/v1/d18-1230

[135] Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kronrod, and Animashree Anandkumar. 2017. Deep active learning for named entity recognition. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*. Association for Computational Linguistics, Vancouver, Canada, 252–256. DOI: https://doi.org/10.18653/v1/W17-2630

[136] C. E. Shannon. 2006. A mathematical theory of communication. *The Bell System Technical Journal* 27, 3 (2006), 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x

[137] Linda B. Smith and Lauren K. Slone. 2017. A developmental approach to machine learning? *Frontiers in Psychology* 8 (2017). https://www.frontiersin.org/articles/10.3389/fpsyg.2017.02124/full

[138] Larry L. Smith, Lorraine K. Tanabe, Rie Johnson nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, C. Friedrich, Kuzman Ganchev, Manabu Torii, Hongfang Liu, Barry Haddow, Craig A. Struble, Richard J. Povinelli, Andreas Vlachos, William A. Baumgartner, Lawrence E. Hunter, Bob Carpenter, Richard Tzong-Han Tsai, Hong-Jie Dai, Feng Liu, Yifei Chen, Chengjie Sun, Sophia Katrenko, Pieter W. Adriaans, Christian Blaschke, Rafael Torres, Mariana L. Neves, Preslav Nakov, Anna Divoli, Manuel Maña-López, Jacinto Mata, and W. John Wilbur. 2008. Overview of BioCreative II gene mention recognition. *Genome Biology* 9 (2008), S2–S2.

[139] Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *Proceedings of the 5th International Conference on Learning Representations*. OpenReview.net. Retrieved from https://openreview.net/forum?id=r1Aab85gg

[140] Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*. Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.), 4077–4087. Retrieved from https://proceedings.neurips.cc/paper/2017/hash/cb8da6767461f2812ae4290eac7cbc42-Abstract.html

[141] Yisheng Song, Ting Wang, Puyu Cai, Subrota K. Mondal, and Jyoti Prakash Sahoo. 2023. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Comput. Surv.* 55, 13s (2023), 40 pages. https://doi.org/10.1145/3582688

[142] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. 2019. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 403–412.

[143] Shengli Sun, Qingfeng Sun, Kevin Zhou, and Tengchao Lv. 2019. Hierarchical attention prototypical networks for few-shot text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, Hong Kong, China, 476–485. DOI: https://doi.org/10.18653/v1/D19-1045

[144] Flood Sung, Yongxin Yang, L. Zhang, T. Xiang, P. Torr, and Timothy M. Hospedales. 2018. Learning to compare: Relation network for few-shot learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), 1199–1208.

[145] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1199–1208. DOI: https://doi.org/10.1109/CVPR.2018.00131

[146] Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan T. McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics* 1 (2013), 1–12. DOI: https://doi.org/10.1162/tacl_a_00205

[147] S. Thrun and L. Y. Pratt. 1998. *Learning to Learn*, Sebastian Thrun and Lorien Pratt (Eds.). Springer, New York, NY, Number of pages: VIII, 354, Edition number: 1. DOI: https://doi.org/10.1007/978-1-4615-5529-2

[148] Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003*. 142–147. Retrieved from https://aclanthology.org/W03-0419

[149] Özlem Uzuner, Brett R. South, Shuying Shen, and Scott L. DuVall. 2011. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association* 18, 5 (2011), 552–556. DOI: https://doi.org/10.1136/amiajnl-2011-000203

[150] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*. Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.), 5998–6008. Retrieved from https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[151] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations*. OpenReview.net. Retrieved from https://openreview.net/forum?id=rJXMpikCZ

[152] Oriol Vinyals, Charles Blundell, T. Lillicrap, K. Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *Proceedings of the Advances in Neural Information Processing Systems*.

[153] Laura von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Michal Walczak, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, Jochen Garcke, Christian Bauckhage, and Jannis Schuecker. 2021. Informed machine learning - a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering* 35, 1 (2021), 1–1. DOI: https://doi.org/10.1109/TKDE.2021.3079836

[154] Bailin Wang, Wei Lu, Yu Wang, and Hongxia Jin. 2018. A neural transition-based model for nested mention recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.), Association for Computational Linguistics, 1011–1017. DOI: https://doi.org/10.18653/v1/d18-1124

[155] Mengqiu Wang and Christopher D. Manning. 2014. Cross-lingual projected expectation regularization for weakly supervised learning. *Transactions of the Association for Computational Linguistics* 2 (2014), 55–66.

[156] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.* 53, 3 (2021), 34 pages. https://doi.org/10.1145/3386252

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

[158] Yu-Xiong Wang, Ross B. Girshick, M. Hebert, and Bharath Hariharan. 2018. Low-shot learning from imaginary data. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), 7278–7286.

[159] Zhenghui Wang, Yanru Qu, Liheng Chen, Jian Shen, Weinan Zhang, Shaodian Zhang, Yimei Gao, Gen Gu, Ken Chen, and Yong Yu. 2018. Label-aware double transfer learning for cross-specialty medical named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 1–15. DOI: https://doi.org/10.18653/v1/N18-1001

[160] Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the*

9th International Joint Conference on Natural Language Processing. Association for Computational Linguistics, Hong Kong, China, 6382–6388. DOI: https://doi.org/10.18653/v1/D19-1670

[161] Patricia L. Whetzel, Natasha Noy, N. Shah, P. Alexander, Csongor Nyulas, T. Tudorache, and M. Musen. 2011. BioPortal: Enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. *Nucleic Acids Research* 39, W541-5 (2011), W541–W545.

[162] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:1910.03771. Retrieved from https://arxiv.org/abs/1910.03771

[163] Jiateng Xie, Zhilin Yang, Graham Neubig, Noah A. Smith, and Jaime Carbonell. 2018. Neural cross-lingual named entity recognition with minimal resources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 369–379. DOI: https://doi.org/10.18653/v1/D18-1034

[164] Qizhe Xie, E. Hovy, Minh-Thang Luong, and Quoc V. Le. 2020. Self-training with noisy student improves imagenet classification. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), 10684–10695.

[165] Chen Xing, Negar Rostamzadeh, Boris N. Oreshkin, and Pedro O. Pinheiro. 2019. *Adaptive Cross-Modal Few-Shot Learning*. Curran Associates Inc., Red Hook, NY.

[166] Liang Xu, Yu Tong, Qianqian Dong, Yixuan Liao, Cong Yu, Yin Tian, Weitang Liu, Lu Li, and Xuanwei Zhang. 2020. CLUENER2020: Fine-grained named entity recognition dataset and benchmark for chinese. arXiv:2001.04351. Retrieved from https://arxiv.org/abs/2001.04351

[167] Yaosheng Yang, Wenliang Chen, Zhenghua Li, Zhengqiu He, and Min Zhang. 2018. Distantly supervised NER with partial annotation learning and reinforcement learning. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, 2159–2169. Retrieved from https://aclanthology.org/C18-1183

[168] Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. In *Proceedings of the 5th International Conference on Learning Representations*. OpenReview.net. Retrieved from https://openreview.net/forum?id=ByxpMd9lx

[169] David Yarowsky and G. Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics*.

[170] Wenpeng Yin. 2020. Meta-learning for few-shot natural language processing: A survey. arXiv:2007.09604. Retrieved from https://arxiv.org/abs/2007.09604

[171] Kang Min Yoo, Youhyun Shin, and Sang-goo Lee. 2019. Data augmentation for spoken language understanding via joint variational generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 7402–7409.

[172] Sung Whan Yoon, Jun Seo, and Jaekyun Moon. 2019. TapNet: Neural network augmented with task-adaptive projection for few-shot learning. In *Proceedings of the 36th International Conference on Machine Learning*. Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.), PMLR, 7115–7123. Retrieved from http://proceedings.mlr.press/v97/yoon19a.html

[173] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks?. In *Proceedings of the Advances in Neural Information Processing Systems*. Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (Eds.), Vol. 27, Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2014/file/375c71349b295fbe2dcdca9206f20a06-Paper.pdf

[174] Houjin Yu, Xian-Ling Mao, Zewen Chi, Wei Wei, and Heyan Huang. 2020. A robust and domain-adaptive approach for low-resource named entity recognition. In *Proceedings of the 2020 IEEE International Conference on Knowledge Graph*. Enhong Chen and Grigoris Antoniou (Eds.), IEEE, 297–304. DOI: https://doi.org/10.1109/ICBK50248.2020.00050

[175] Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Lang. Resour. Evaluation* 51, 3 (2017), 581–612. DOI: https://doi.org/10.1007/s10579-016-9343-x

[176] Xiangji Zeng, Yunliang Li, Yuchen Zhai, and Yin Zhang. 2020. Counterfactual generator: A weakly-supervised method for named entity recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online, 7270–7280. DOI: https://doi.org/10.18653/v1/2020.emnlp-main.590

[177] Xiangji Zeng, Yunliang Li, Yuchen Zhai, and Yin Zhang. 2020. Counterfactual generator: A weakly-supervised method for named entity recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online, 7270–7280. Retrieved from https://www.aclweb.org/anthology/2020.emnlp-main.590

[178] Chuxu Zhang, Kaize Ding, Jundong Li, Xiangliang Zhang, Yanfang Ye, Nitesh V. Chawla, and Huan Liu. 2022. Few-shot learning on graphs: A survey. arXiv:2203.09308. Retrieved from https://arxiv.org/abs/2203.09308

[179] Meng Zhang, Yang Liu, Huanbo Luan, and M. Sun. 2017. Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

[180] Ye Zhang, Matthew Lease, and Byron C. Wallace. 2017. Active discriminative text representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[181] Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering* 34, 12 (2021), 1–1. DOI: https://doi.org/10.1109/TKDE.2021.3070203

[182] Yaohui Zhu, Chenlong Liu, and Shuqiang Jiang. 2020. Multi-attention meta learning for few-shot fine-grained image recognition. In *Proceedings of the IJCAI*.

[183] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. 2020. Rethinking pre-training and self-training. In *Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*. Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.), Retrieved from https://proceedings.neurips.cc/paper/2020/hash/27e9661e033a73a6ad8cefcde965c54d-Abstract.html

[184] Haosheng Zou, Tongzheng Ren, Dong Yan, Hang Su, and Jun Zhu. 2021. Learning task-distribution reward shaping with meta-learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 12 (May 2021), 11210–11218. DOI: https://doi.org/10.1609/aaai.v35i12.17337

Received 28 January 2022; revised 16 April 2023; accepted 22 June 2023