A Realistic Study of Auto-regressive Language Models for Named Entity Typing and Recognition

Elena V. Epure, Romain Hennequin
Deezer Research, Paris, France
research@deezer.com

Abstract
Despite impressive results of language models for named entity recognition (NER), their generalization to varied textual genres, a growing entity type set, and new entities remains a challenge. Collecting thousands of annotations in each new case for training or fine-tuning is expensive and time-consuming. In contrast, humans can easily identify named entities given some simple instructions. Inspired by this, we challenge the reliance on large datasets and study pre-trained language models for NER in a meta-learning setup. First, we test named entity typing (NET) in a zero-shot transfer scenario. Then, we perform NER by giving few examples at inference. We propose a method to select seen and rare / unseen names when having access only to the pre-trained model and report results on these groups. The results show: auto-regressive language models as meta-learners can perform NET and NER fairly well especially for regular or seen names; name irregularity when often present for a certain entity type can become an effective exploitable cue; names with words foreign to the model have the most negative impact on results; the model seems to rely more on name than context cues in few-shot NER.

1 Introduction
The standard approach to perform named entity recognition (NER) nowadays relies on training or fine-tuning a deep neural network using a relatively large annotated dataset and most often aims at extracting a few regular named entity types such as person, location and organisation (Yadav and Bethard, 2018; Akbik et al., 2019; Lison et al., 2020). Although recent language models have yielded impressive results, in reality, the task is far from being generally solved.

First, NER generalization to other textual genres, apart from the usual newswire benchmark dataset (Tjong Kim Sang and De Meulder, 2003), is still problematic. In particular, NER in informal text, frequently found in social media or chat-bot interactions, remains a challenging task. This type of text could often lack proper formatting, e.g. word capitalization or complete sentences; and could contain unusual grammatical structures, spellings, or jargon (Aguilar et al., 2018; Yadav and Bethard, 2018; Augenstein et al., 2017; Guerini et al., 2018)

A second challenge is the generalization of NER to a diverse and growing entity type set, belonging to new domains such as movies, music or e-commerce (Ma et al., 2016; Guerini et al., 2018; Lin et al., 2020b). These types are often more heterogeneous (e.g. groups in WNUT includes both sport teams and music bands, Aguilar et al. 2018); could lack name regularity (e.g. creative work titles are not necessarily noun phrases, Lin et al. 2020b); could be composed of common words (e.g. the film title "demolition man" or "Duck" in the character name "Donald Duck", Derczynski et al. 2017), or could be composed of words which are typically from other languages (e.g. Szeged in an English-language text, Augenstein et al. 2017).

A third challenge is the generalization of NER to new entities, unseen during training. This case is the most frequent in real-world where a system should learn from a limited number of examples per type while entity mentions are expected to shift in time (Augenstein et al., 2017; Lin et al., 2020b).

Within the framework of standard NER, these challenges have been mainly addressed by resorting to the annotation of new training datasets with each new case. However, collecting thousands of human annotations for new genres, entity types or entities is expensive and time-consuming, while without sufficient training data, generalization has been shown to be problematic (Augenstein et al., 2017; Lin et al., 2020a; Brown et al., 2020).

In this work, we adopt an alternative study of language models for NER that, instead of relying on large annotated datasets, draws inspiration from human linguistic behavior (Levesque, 2014). As
humans, we can easily recognize entities in text based on our prior domain and common sense linguistic knowledge, or by leveraging contextual cues in text (Lin et al., 2020a). In fact, we are quite good at this task even when identifying new entities in a second language (Kobeleva, 2012) and our capacity for proper names recognition develops in early childhood (Bloom and Markson, 1998).

As emphasized by Janet Pierrehumbert in her EMNLP 2020 keynote (Webber et al., 2020), the human linguistic behavior could lead to infinite examples, many of them new for everyone, including the NLP systems’ designers. Thus, a realistic testing of a NER system which aims at resembling humans should be fast, adaptable and generalizable. Moreover, humans can perform new linguistic tasks quite well even when exposed to a few examples or very simple instructions (Brown et al., 2020).

Hence, along this line of thought, our goal is to study language models for NER in a meta-learning setup, i.e. zero or few-shot learning deemed to be more realistic. As for humans, during pre-training, language models should have accumulated diverse domain and linguistic knowledge, and developed general pattern recognition abilities (Brown et al., 2020). The research question is then: can current language models leverage this knowledge at inference in order to adapt to diverse NER tasks, when exposed to no or a few examples as demonstration?

To answer this, we propose to study named entities of various types individually and in context. First, we probe auto-regressive language models for named entity typing (NET) in a zero-shot transfer scenario. We define a procedure to assess entity exposure and report its impact on results. Second, we adapt the model to perform NER by providing few examples at inference. In this case, NER is no longer modelled as a sequence labeling problem, but as a machine reading comprehension (MRC) task and becomes equivalent to extracting spans of text from the input as answers to simple queries such as “find an organization such as company”. Here, we also test NER for seen versus rare or unseen entities, trying to gain insights into the role of context, i.e. the text surrounding entities.

We base our experiments (Section 4) on a medium-sized GPT2 (Radford et al., 2019) and four datasets: CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003), WNUT2017 (Liu, 2014), MIT Movie (Liu, 2014) and extensive lists of named entities collected from DBpedia (Auer et al., 2007). The datasets cover clean and noisy text, and regular entities such as people names and irregular entities such as creative work titles. The results (Section 5) show that auto-regressive language models as meta-learners can perform NET and NER fairly well especially for regular names or seen during pre-training; name irregularity when often present for a certain entity type can become an effective exploitable cue; names with words foreign to the model have the most negative impact on results; the model seems to rely more on cues from entity names than on context in few-shot NER. Lastly, we discuss the implications of this study for future NER research in Section 6 and release the code

2 Related Work

While NER language models have achieved impressive results on standard benchmarks (Akbik et al., 2019), multiple works have emphasized their limitations with regard to their generalization capacity to new textual genres, entity type sets and entity mentions (Derczynski et al., 2017; Lin et al., 2020b). In order to address these, the most recent approaches rely on existing named entity resources, such as gazetteers and dictionaries, to either perform NER in a distant or weak supervision setup (Mengge et al., 2020; Lison et al., 2020; Shang et al., 2018), or to train entity type classifiers adaptable to unseen entities (Guerini et al., 2018).

Separating NER in two different model training objectives, span identification and entity classification, have been shown to also improve performance while generalizing to new entity types (Aguilar et al., 2017; Mengge et al., 2020). In particular, performing span identification with masked entities forces the model to rely on context, thus learning to differentiate entity and non-entity words. The importance of context in NER is also invoked by Lin et al. (2020a) who advocate to complement the manual annotation by entity triggers, i.e. groups of words acting as cues. Then, a neural network learns trigger representations which are used to guide a standard NER model (BiLSTM-CRF) at inference.

In addition to proposing new NER systems, multiple works have also turned towards probing existing models for generalization (Augenstein et al., 2017; Taillé et al., 2020; Fu et al., 2020). Lin et al. (2020b) propose an extensive study using randomization tests to investigate the extent to which a fine-tuned language model relies on: name regularity—

1 Code included in submission will be publicly available.
regular versus irregular names; mention coverage—the ratio of overlapping entities between train and test datasets; and context diversity—the number of unique sentences for each entity type.

In this work, inspired by how humans perform NER, we investigate the extent to which a pre-trained language model could be used for general NER in zero- or few-shot settings. Compared to Lin et al. (2020b), our investigation is not centered on fine-tuning and, implicitly, on the impact of train / test datasets, but directly on the model as-is. To our knowledge, this is the first detailed study of pre-trained models as NER meta-learners. Our analysis covers multiple generalization angles: seen versus unseen/rare entities, regular versus irregular entities, diverse textual genres including noisy text, and contextual versus named entity cues. Our goal is to contribute to a better understanding of language models for NER in a real-world context.

3 NER Meta-learning

Before transformer-based language models (Devlin et al., 2019), the state-of-the-art NER was based on training recurrent neural networks, such as bidirectional LSTMs with a CRF layer, from scratch (Yadav and Bethard, 2018). The common approach with transformers have been to fine-tune them for the desired task, thus specializing their general linguistic knowledge, acquired during pre-training. Recently, transformers have been also explored for modeling NER as a MRC problem with positive results (Mengge et al., 2020; Li et al., 2020).

In the current work, we adopt the latter approach and frame NER as a MRC task. Though, instead of fine-tuning / training a pre-trained language model for MRC, we exploit it as a meta-learner. We divide our study in NET in a zero-shot transfer (in Section 3.1), followed by span extraction and typing in a few-shot settings (in Section 3.2). This separation allows us to acquire knowledge first about entities names and then about entities in context.

3.1 Zero-shot Named Entity Typing

In what follows, we consider auto-regressive generative sequence language models such as GPT2 (Radford et al., 2019) or GPT3 (Brown et al., 2020), leaving this study adaptation to masked language models (Devlin et al., 2019) as future work. The goal of auto-regressive language models is to estimate the empirical distribution from the training data, where each training example \( x \) is a sequence of tokens \( x = (s_1, s_2, ..., s_n) \). Given the sequential nature of the language, it is common to factorize the distribution \( p(x) \) with the Bayes’ rule and express it as a product of conditional probabilities of each sequence’s token \( s_i \) given the previous tokens:

\[
p(x) = \prod_{i=1}^{n} p(s_i|s_1, ..., s_{i-1})
\]

On a new task, the model infers \( p(output|input) \) or more completely written \( p(output|input, task) \). Brown et al. (2020) merge the input and task in a single natural language query and express output as the predicted next sequence of tokens. For instance, the query for NET could be written as "Sentence: is Italy a person, location or organisation? Answer:" and the predicted entity type is considered the token among "person", "location", or "organisation" which the model estimates as most likely to follow the query.

Alternatively, NET could be also framed as the most likely statement among multiple competing ones such as "Anne is a person", "Anne is a location". In this case, the sequence with the lowest perplexity is the one that the model is less surprised to see, hence describing the most likely entity type. The perplexity of a sequence \( x \), using a model \( \theta \), is:

\[
PPL_{\theta}(x) = \exp\left\{-\frac{1}{n} \sum_{i=1}^{n} \log p_{\theta}(s_i|s_{<i})\right\}
\]

We adopt this latter approach for NET as it provides a simple framing of the task in zero-shot settings and is fast, i.e. perplexity can be efficiently computed by relying on a single model call. Thus, given an entity mention \( e \) and an entity type set \( T \), the most likely type \( t_e \in T \) for entity \( e \) is:

\[
t_e = \arg \min_{t \in T} PPL_{\theta}(query(e, t))
\]

where query\((e, t)\) is the template "\( e \) is a \( t \)" (e.g. "Cindarella is a city" or "Cindarella is a character").

**Assessing generalisation.** The model’s ability to generalize to new, unseen or rare entities during training, is essential in real-world scenarios. Thus, measuring how the model performs on seen versus unseen entities could provide a more realistic understanding of its real performance. Previous works that led such investigation, trained models from scratch (Lin et al., 2020b; Taillé et al., 2020), so could keep track of which entities were (un)seen
during training. As we focus on pre-trained language models and have no access to the their training data, we need to devise a method for assessing if the model has been exposed to an entity or not.

Carlini et al. (2019) proposed a testing method for unintended memorization of rare sequences centered on perplexity. Given all possible sequences for a matter at hand, or a very large sample ($S$), that are prefixed by the same query (e.g. "the random number is ") and $S = \{281265011, 281265017...\}$, rank them by perplexity and use these ranks to compute exposure. As shown in Eq. 4, for $x \in S$, the exposure metric is negatively correlated with the rank, i.e. the lower the rank the higher the exposure, thus likely memorization.

$$\text{exposure}_\theta(x) = \log_2 |S| - \log_2 \text{rank}_\theta(x)$$ (4)

This testing method is a helpful point of departure, but less applicable to our task. Without a very large set of entities for each type, the estimates could be inaccurate, especially when only few sequences have lower perplexity than a target one (Carlini et al., 2019). Also, we noticed experimentally that the mean perplexity tended to decrease with the number of tokens per entity$^2$, a phenomenon most likely related to the open-vocabulary language modeling over sub-word units (Figure 1). Thus, with the method of Carlini et al. (2019), entities would have a higher chance to be flagged as memorized when they have more tokens.

As originally stated, our goal is to evaluate the model’s behavior with seen and unseen / rare entities too. Thus, we want to be able to assign entities to these two groups when we are confident of their (non-)memorization, while ignoring entities in the gray area. We changed the previously shown memorization method to rely directly on probabilities of entity’s tokens, obtained when calling the model with entities as input prefixed by a fixed string. The test we propose is further summarized: if entity words are known and their sequential transitions are unsurprising, then the model has likely seen the entity during training. Formally, we define two exposure metrics corresponding to these two aspects:

$$\text{exposure}_\theta^{\text{word}}(x) = \prod_{(i,j) \in W_x} \text{test}_\theta^{\text{word}}(x, i, j)$$ (5)

$$\text{test}_\theta^{\text{word}}(x, i, j) = \begin{cases} 1 & \text{if } i = j \\ p_\theta(s_j | s_{<j}) & \text{if } i < j \end{cases}$$

where $W_x$ has tuples marking the start and end indices of each word in $x$ and $T_x$ has indices marking the transitions, i.e. the index of each new word.

In Equation 5, we identify two cases when a word can be considered known by the model. It is directly mapped on a token in the model vocabulary $V$. Or, when the word tokenization results in multiple tokens, the last token becomes an indicator of its memorization. In Equation 6, to test whether the sequential association of words is unsurprising to the model, we take the minimum probability of the tokens marking the start of each new word.

For the final decision, given that the exposure metrics are built on probabilities, entities with values higher or lower than some established thresholds could be assigned to the seen, respectively unseen / rare entity groups. These thresholds could be defined by considering the entity set and the model’s vocabulary size (more details in Section 4). An advantage of our method over (Carlini et al., 2019) is that we do not need access to a very large set for each entity type, the token probabilities being sufficient to establish the degree of exposure.

### 3.2 Few-shot Named Entity Recognition

Few-shot settings have shown competitive results in other NLP tasks such as question answering, translation, and text classification (Brown et al., 2020). Zhao et al. (2021) have also tested information extraction for slot-filling with some slots targeting entities (e.g. the director of a movie). However, they start from the premise that each sentence contains that type of slot, without assessing cases when no named entities exist in text.

Similar to previous works, we use the query to formulate the task and insert examples, which are used only at inference, without triggering updates of the pre-trained model weights. We show in Figure 2 a query generated from WNUT2017 dataset for the entity type **product**. The query has two parts: a prefix and a test sentence. The prefix is appended to each sentence and introduces examples. Below, we provide 4 examples, two with entities and two without. We use the token "none" to mark the absence of an entity of the target type. The second part with the last two lines introduces the test sentence, for which NER should be performed.

\[\text{exposure}_\theta^{\text{trans}}(x) = \min_{i \in T_x} p_\theta(s_i | s_1)\] (6)
Figure 1: Mean and standard deviation of log perplexity computed with GPT2 for large lists of persons (left), locations (middle) and organisations (right) from DBpedia. Values are grouped by the number of tokens per entity.

Sentence: I don’t like to be stuck at home
product: none
Sentence: Where is Gelato Gilberto?
product: none
Sentence: Well, I was gonna buy a Zune HD
product: Zune HD
Sentence: BEAUTY TIPS: SK-II UV Cream
product: SK-II UV Cream
Sentence: CVS sells their own epipen
Correct prediction: epipen

Figure 2: NER query and expected generated answer. The first 8 query lines are examples (two negatives, two positives), the last two lines the test sentence.

Previous works have shown that the choice of the query has a significant impact on the task’s accuracy (Li et al., 2020). Moreover, in the few-shot learning case, the set of examples and their order can lead to different results too (Zhao et al., 2021). For instance, the model can have the tendency to predict the majority token or the one nearest to the end of the query. In order to overcome this, Zhao et al. (2021) propose a procedure to calibrate the model’s output probabilities by taking into account the model’s bias towards certain outputs.

We use the same calibration procedure in our experiments. Also, we run each experiment multiple times, with varied examples as demonstration, in order to track the variance. As for the query format, we stick to the one shown in the example and leave for future work the exploration of other formats.

Assessing generalization. As emphasized for NET, the generalization of the model to rare or unseen entities is a more realistic setup for evaluation. Thus, we could select two groups of sentences, with seen respectively unseen entities, from the test set with our memorization testing, and report performance on each separately. However, splitting smaller datasets, such as WNUT2017, is not reliable. Also, this would not allow us to learn about the role of contextual cues, i.e. the other parts of the sentence, in named entity extraction. As Lin et al. (2020b) highlighted, we should aim at NER models that rely more on context for generalization, rather than memorizing entities, in particular for irregular entity types such as creative work titles.

Therefore, the experiment design we propose for assessing generalization is to fix the prefix of the query containing the training examples and compute performance on three variations of the test set: test as-is, test seen and test unseen. To obtain the last two test sets, we replace each entity with a named entity randomly selected from a list of seen, respectively unseen names for each entity type. Thus, context stays fixed, but named entities are changed. For test seen, we sample named entities to replace existing ones from a list created with our named entity memorization method. For test unseen, we choose among random lowercase strings, which do not exist in the English language. As the query examples contain proper entity names, much more different in form than these randomly generated strings, the model should be constrained to rely more on context for prediction.

4 Experiments

We use a medium-sized GPT2 in our experiments (Radford et al., 2019). A larger one like GPT3 yielded better results as a meta-learner in past experiments (Zhao et al., 2021). However, for this study, we decided upon a model that was easily accessible and had lower memory requirements.

The used datasets are summarized in Table 1. CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) and WNUT2017 (Derczynski et al., 2017), commonly found in NER benchmarks, are kept as they are. The MIT Movie dataset (Liu, 2014), origi-
| Dataset        | Named entity type set                                      |
|---------------|----------------------------------------------------------|
| CoNLL-2003    | person, location, organisation                           |
| WNUT2017      | person, location, corporation, group, product, creative work |
| MIT Movie     | person, creative work                                    |
| DBpedia       | person, location, organisation, creative work            |

Table 1: Entity types in each dataset used in the study.

normally created for slot filling, is modified by ignoring some slot types (e.g. genre, rating) and merging others (e.g. director and actor in person, and song and movie title in title). MIT movie dataset contains only lowercase text, sometimes with typos, thus falling under the noisy genre as WNUT2017.

In NET, we consider the entity mentions from each dataset in its entirety, (train, test, and dev if available). We also collected very large lists of different entity types from DBpedia (Auer et al., 2007). These are particularly interesting because Wikipedia has not been included in the GPT2 training corpora (Radford et al., 2019). The NER experiments are run only on test sets while the train sets are used for sampling examples for the query. We report F1-score for NET evaluation, and accuracy computed separately for positive (AccP) and negative (AccN) examples in NER. Although we could report an aggregated metric such as F1-score, we are interested in checking these scores separately in order to assess the model performance for both tasks, the prediction of "none" vs. a name.

**Zero-shot NET.** To classify named entities, we create prompts starting from entity types and choose as predicted value the entity type which leads to the lowest perplexity. In practice, we use multiple keywords for each entity type starting from their definition. We also include character for person; company, group, institution, club, and corporation for organization; place, city, and country for location. As the perplexity tends to decrease with the number of tokens, we choose all keywords such that they are part of the model vocabulary. Thus, creative work is replaced by work, title, movie, song, and book. The only entity types for which we do not include other keywords are: product, corporation and group in WNUT2017.

For the exposure computation, we prefix each entity with the default unknown token when retrieving probabilities. The thresholds for word and transition exposures are established per dataset. For the lower limit, we consider the size of the GPT2 vocabulary ($\approx 50K$); thus, assuming a uniform distribution\(^5\), each token would have a $2e^{-05}$ probability to be generated next. CoNLL-2003 has many one-word named entities with rare transitions. For this reason, we focus only on exposure\(e^{\text{word}}\) to establish if a named entity is seen ($\geq .8$) or rare/unseen ($\leq 1e^{-04}$). The rest of named entities are not classified. In contrast, in MIT Movies, named entities are often composed of multiple English-language known words. Thus, exposure\(e^{\text{trans}}\) is more informative for selecting seen entities ($\geq .001$) and rare/unseen entities ($\leq 1e^{-05}$).

We sample the two groups from the DBpedia lists using either one of the two exposure metrics or both (thresholds in Appendix A). In this way, we investigate the impact of knowing words vs. recognizing word transitions on a larger sample. We ignore one-word named entities from MIT Movies and DBpedia because they are rare or often spurious. Finally, we only run the NET experiment on the complete WNUT2017 dataset because the number of named entities of each type is too small to allow reliable seen vs. rare/unseen split.

**Few-shot NER.** We opt for a maximum of training examples in the query that can be kept in memory, in our case 16. Out of these, 9 contain named entities of the targeted type and 7 are randomly chosen from the rest of the dataset. We run each experiment three times with different random seeds to capture variance. The test set is slightly modified too: for each entity type, we keep all positive sentences and sample negative sentences such that the ratio positive-negative is about 2:1. The maximum number of tokens asked when querying the model is set to 15. The calibration we apply follows the steps described in (Zhao et al., 2021).

We design the NER meta-learner to extract one named entity of the prompted type at a time, leaving the extraction of multiple named entities per text for future. Because a test sentence can mention multiple named entities of the same type, we consider a generated answer to be correct if it matches one of the existing named entities. In computing accuracy, we rely mostly on exact named entity matching with some exceptions. The evaluation is insensitive to letter case (e.g. ‘none’ and ‘None’ are considered equivalent). Also, we noticed that the model tends to add spaces for named entities written

\(^5\)This assumption is strong, but used only to establish an order of magnitude for the unseen exposure\(e^{\text{trans}}\) threshold.
Table 2: Zero-shot NET results for CoNLL-2003 and MIT Movie named entity lists.

| Dataset      | Type      | All   |          |          |          | Rare/Unseen |          |          |
|--------------|-----------|-------|----------|----------|----------|-------------|----------|----------|
|              |           | F1-score | Support | F1-score | Support | F1-score   | Support |
| CoNLL-2003   | person    | 0.90   | 3613     | 0.93     | 695      | 0.86       | 619      |
|              | location  | 0.66   | 1331     | 0.74     | 546      | 0.37       | 80       |
|              | organisation | 0.70   | 2401     | 0.74     | 770      | 0.63       | 289      |
|              | macro-average | 0.75   | 7345     | 0.81     | 2011     | 0.62       | 988      |
| MIT Movie    | person    | 0.80   | 2866     | 0.82     | 605      | 0.81       | 369      |
|              | creative work | 0.60   | 2122     | 0.64     | 402      | 0.58       | 256      |
|              | macro-average | 0.70   | 4988     | 0.73     | 1007     | 0.69       | 625      |

Table 3: Zero-shot NET results for DBpedia.

| Pruning      | Type      | Seen | Rare | Support |
|--------------|-----------|------|------|---------|
|              | person    | 0.88 | 0.64 | 10000   |
|              | location  | 0.80 | 0.62 | 10000   |
|              | organisation | 0.76 | 0.67 | 10000   |
|              | creative work | 0.69 | 0.37 | 10000   |
|              | macro-average | 0.78 | 0.58 | 40000   |

Table 4: Zero-shot NET results for WNUT2017.

| Type      | Supporting | F1-score | Support |
|-----------|------------|----------|---------|
| person    |            | 0.79     | 1317    |
| location  |            | 0.63     | 616     |
| corporation |          | 0.16     | 231     |
| group     |            | 0.44     | 412     |
| product   |            | 0.46     | 353     |
| creative work |         | 0.46     | 361     |
| macro-average |        | 0.49     | 3290    |

Table 5: NER results (accuracy ± standard deviation) for CoNLL-2003 and MIT Movie.

| Type      | Support | AccP | AccN  |
|-----------|---------|------|-------|
| person    | 1537    | 0.71±0.11 | 0.58±0.13 |
| location  | 1899    | 0.83±0.02 | 0.45±0.04 |
| organisation | 1843 | 0.76±0.02 | 0.32±0.11 |
| creative work | 1908 | 0.73±0.05 | 0.80±0.01 |
| person    | 906     | 0.28±0.07 | 0.98±0.01 |

5 Results and Discussion

Zero-shot NET. The results presented in the tables 2 and 3 show that the model can often associate categories to entity mentions with high accuracy, in particular for regular types such as person. We see a lower performance for creative work in MIT Movies and location in CoNLL-2003, these being often confused with person, respectively organisation. The first confusion is not surprising given that movie titles could often contain character names while character is included in the person type. Also, this type includes irregular names, known to be more challenging (Aguilar et al., 2017; Lin et al., 2020b). The second confusion, location-organisation, is already mentioned as a common issue (Derczynski et al., 2017). Without context, disambiguation is problematic, if not impossible.

The performance of NET for seen samples is larger than for rare/unseen samples. However, we can see in Table 2 that the drop is much smaller for MIT Movie than for CoNLL-2003. The difference between the two entity lists lies in the criterion we used for identifying seen vs. rare/unseen entities, either focused on knowing the words or recognizing the transitions between words. The obtained results suggest that for correctly classifying a named entity, the model’s exposure to its words weighs much more than its exposure to the exact entity, i.e. its word transitions. This is further confirmed on the larger DBpedia named entity lists (Table 3).

Eventually, when pruning both word and transition exposure metrics, the results are only marginally improved (see Appendix A).

For WNUT2017 (Table 4) that contains the noisiest entity names coming from social media, the results are significantly lower, apart from person and location. These could be linked to the fact that the model has not seen the words used to denote many names during pre-training. Thus, the classification may be challenging without context. We noticed similar confusion patterns as before: corporation or group (associated with organisation) with location, and creative work with person.

In practice, the model generates sometimes “null”, “.” etc.
Few-shot NER. The results for 16-shot NER are presented in Tables 5 and 6. The pre-trained language model could perform NER surprisingly well in meta-learning settings, even on noisy data. AccN tends to be lower for CoNLL-2003, which suggests that more negative examples should be given at inference. Similarly, for MIT Movie, the model tends to predict often "none" for creative work, issue that might be overcome by including more effective positive examples in the query.

In Table 6, we show the language model performance on WNUT2017, the noisiest dataset from those included. We cannot directly compare these scores with the baseline (Aguilar et al., 2018) because of the way we created the test sets, which had an influence on the number of false positives. However, we could still outline some trends. Similar to the past work, location and person are among the easiest to extract types, while product is the hardest. In contrast, we can extract corporation and creative work types quite well.

The last columns in Table 6 show AccP when all named entities are replaced by various random strings while fixing the context and the training examples. These random strings are selected from a seen named entity list per type or one containing randomly generated names (lists available in Appendix A). As for NET, the impact of named entity (in)exposure during pre-training is significant. The scores are much larger on Test seen, respectively smaller on Test unseen, than on Test as-is, with two exceptions for group and product.

Table 6: NER results (accuracy ± standard deviation) for the WNUT2017 dataset.

| Named entity type  | Support  | Test as-is | Test seen | Test unseen |
|--------------------|----------|------------|-----------|-------------|
|                    |          | AccP       | AccN      | AccP        | AccN        |
| person             | 490      | 0.63±0.08  | 0.56±0.17 | 0.79±0.12   | 0.57±0.24   |
| location           | 187      | 0.61±0.03  | 0.67±0.11 | 0.79±0.00   | 0.50±0.12   |
| corporation        | 93       | 0.64±0.05  | 0.49±0.15 | 0.87±0.01   | 0.42±0.09   |
| group              | 180      | 0.56±0.01  | 0.25±0.06 | 0.70±0.06   | 0.64±0.06   |
| product            | 144      | 0.40±0.16  | 0.41±0.29 | 0.50±0.12   | 0.38±0.18   |
| creative-work      | 184      | 0.57±0.01  | 0.50±0.08 | 0.76±0.04   | 0.49±0.04   |

Overall, the score differences between Test seen and Test as is are larger than the ones between Test unseen and Test as is. The model appears to prioritize named entities cues more than context cues in few-shot settings. When choosing query or training examples, we may need to focus more on providing effective and diverse named entities patterns for an entity type than diverse context patterns. Further experiments are required to explore this hypothesis, but results from previous work seem to point in the same direction (Lin et al., 2020b).

6 Conclusion

We presented a study of pre-trained auto-regressive language models as NET and NER meta-learners with the goal to uncover the generalisation to noisy text, diverse named entity types, and rare or new named entities. We deem this setup more realistic, comparable to human NER linguistic behavior and easily adaptable to new domains. The results are promising despite the fact that we did not use the best model available (e.g. GPT3) or engineer the best possible queries for the model.

Lastly, we emphasise the key conclusions with a focus on future NER research. With pre-trained encoders, entity memorization should not be studied in fine-tuning only while neglecting the memorization during pre-training. We have proposed an effective method to help with this. Also, a language model already pre-trained on a general task and a large corpus could effectively bootstrap NER for new applications, especially when entities are common constructs in a language. Frequent name irregularity for a type in context can become a regularity effectively exploited by the model. Context is important but with limited impact. Choosing good query examples of named entity patterns in context for few-shot NER is still a matter of investigation.
References

Gustavo Aguilar, Adrian Pastor López-Monroy, Fabio González, and Thamar Solorio. 2018. Modeling noisiness to recognize named entities using multi-task neural networks on social media. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1401–1412, New Orleans, Louisiana. Association for Computational Linguistics.

Gustavo Aguilar, Suraj Maharjan, Adrian Pastor López-Monroy, and Thamar Solorio. 2017. A multi-task approach for named entity recognition in social media data. In Proceedings of the 3rd Workshop on Noisy User-generated Text, pages 148–153, Copenhagen, Denmark. Association for Computational Linguistics.

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. Pooled contextualized embeddings for 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), pages 724–728, Minneapolis, Minnesota. Association for Computational Linguistics.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In The Semantic Web, pages 722–735, Berlin, Heidelberg. Springer.

Isabelle Augenstein, Leon Derczynski, and Kalina Bontcheva. 2017. Generalisation in named entity recognition. Comput. Speech Lang., 44(C):61–83.

Paul Bloom and Lori Markson. 1998. Capacities underlying word learning. Trends in Cognitive Sciences, 2(2):67–73.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In Advances in Neural Information Processing Systems, volume 33, pages 1877–1901. Curran Associates, Inc.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In Proceedings of the 28th USENIX Conference on Security Symposium, SEC’19, page 267–284, USA. USENIX Association.

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, pages 140–147, Copenhagen, Denmark. Association for Computational Linguistics.

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, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jinlan Fu, Pengfei Liu, and Qi Zhang. 2020. Rethinking generalization of neural models: A named entity recognition case study. Proceedings of the AAAI Conference on Artificial Intelligence, 34(05):7732–7739.

Marco Guerini, Simone Magnolini, Vevake Balaraman, and Bernardo Magnini. 2018. Toward zero-shot entity recognition in task-oriented conversational agents. In Proceedings of the 19th Annual SIGDial Meeting on Discourse and Dialogue, pages 317–326, Melbourne, Australia. Association for Computational Linguistics.

Polina P Kobeleva. 2012. Second language listening and unfamiliar proper names: Comprehension barrier? RELC journal, 43(1):83–98.

Hector J Levesque. 2014. On our best behaviour. Artificial Intelligence, 212:27–35.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5849–5859, Online. Association for Computational Linguistics.

Bill Yucheng Lin, Dong-Ho Lee, Ming Shen, Ryan Moreno, Xiao Huang, Prashant Shiralkar, and Xiang Ren. 2020a. TriggerNER: Learning with entity triggers as explanations for named entity recognition. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8503–8511, Online. Association for Computational Linguistics.

Hongyu Lin, Yaojie Lu, Jialong Tang, Xianpei Han, Le Sun, Zhicheng Wei, and Nicholas Jing Yuan. 2020b. A rigorous study on named entity recognition: Can fine-tuning pretrained model lead to the promised land? In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 7291–7300, Online. Association for Computational Linguistics.


Pierre Lison, Jeremy Barnes, Aliaksandr Hubin, and Samia Touileb. 2020. Named entity recognition without labelled data: A weak supervision approach. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1518–1533, Online. Association for Computational Linguistics.

JJ (Jingjing) Liu. 2014. A conversational movie search system based on conditional random fields. In Interspeech 2012.

Yukun Ma, Erik Cambria, and Sa Gao. 2016. Label embedding for zero-shot fine-grained named entity typing. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 171–180, Osaka, Japan. The COLING 2016 Organizing Committee.

Xue Mengge, Bowen Yu, Zhenyu Zhang, Tingwen Liu, Yue Zhang, and Bin Wang. 2020. Coarse-to-Fine Pre-training for Named Entity Recognition. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6345–6354, Online. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. OpenAI blog.

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, pages 2054–2064, Brussels, Belgium. Association for Computational Linguistics.

Bruno Taillé, Vincent Guigue, and Patrick Gallinari. 2020. Contextualized embeddings in named-entity recognition: An empirical study on generalization. In Advances in Information Retrieval, pages 383–391, Cham. Springer International Publishing.

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 Seventh Conference on Natural Language Learning at HLT-NAACL 2003, pages 142–147.

Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors. 2020. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Online.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In Proceedings of the 27th International Conference on Computational Linguistics, pages 2145–2158, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. arXiv preprint arXiv:2102.09690.
A Additional experiment details and results

In experiments, we used has 1x GTX 1080 with 11GB RAM. We show the running time for each experiment in Table 7.

| Dataset        | Experiment     | Time (s) |
|----------------|----------------|----------|
| CoNLL-2003     | NET person     | 210      |
|                | NET location   | 85       |
|                | NET organisation| 144     |
|                | NER person     | 6371     |
|                | NER location   | 6906     |
|                | NER organisation| 6202   |
| MIT Movie      | NET person     | 131      |
|                | NET creative work| 100    |
|                | NER person     | 4850     |
|                | NET creative work| 2713   |
| WNUT2017       | NET person     | 86       |
|                | NET location   | 49       |
|                | NET corporation| 29       |
|                | NET group      | 40       |
|                | NET product    | 38       |
|                | NET creative work| 39     |
|                | NER person     | 2759     |
|                | NER location   | 1081     |
|                | NER corporation| 730      |
|                | NER group      | 1372     |
|                | NER product    | 1127     |
|                | NER creative work| 1502  |

Table 7: Running time in seconds for each experiment.

Lists of seen and unseen entity names used to create the test seen and test unseen datasets for assessing NER generalization:

- **Seen, person**: 'Mary', 'Steve', 'Davis', 'Danny', 'Rose', 'Edward', 'Rob', 'Harry', 'Tom', 'Paul', 'Sam', 'Robert', 'Alex', 'Michelle', 'James'
- **Seen, location**: 'Florida', 'Toronto', 'Germany', 'India', 'Scotland', 'Washington', 'Syria', 'Ukraine', 'Houston', 'America', 'France', 'Australia', 'Turkey', 'NEW YORK', 'Chicago'
- **Seen, corporation**: 'Reuters', 'CNN', 'NBA', 'Uber', 'YouTube', 'CBC', 'Netflix', 'Microsoft', 'Twitter', 'Facebook', 'Apple', 'MAC', 'Tesla', 'Disney', 'Reddit'
- **Seen, group**: 'Army', 'Chicago Blackhawks', 'Real Madrid', 'CIA', 'Senate', 'ART', 'NBA', 'The Black Keys', 'Crystal Palace', 'European Union', 'green day', 'Labor', 'Chelsea', 'the warriors', 'Democrats'
- **Seen, product**: 'Air Music Jump', 'Android', 'Linux OS', 'iOS', 'Windows 7', 'Tesla', 'Google Music', 'SQL', 'Amazon Prime', 'Nintendo plus', 'google pixel', 'iPhone', 'Xbox 360', 'Legendary Skin', 'Bio Spot'
- **Unseen**: 'xgwqicng', 'kiooaqil', 'wpvqymid', 'rrmihdgc', 'owb1mgbx', 'tiybjelq', 'ytbllh', 'ybwifxx', 'svlsskx', 'jdtqyoov', 'tzrtffbu', 'jwvywjbh', 'hzhwhahw', 'gjmquke', 'gmen-qwpb'
Table 8: Thresholds used for pruning the exposure metrics in order to select seen and rare/unseen entities.

| Dataset    | Pruning | Seen     | Unseen / rare |
|------------|---------|----------|---------------|
|            |         | exposure\(_w\) | exposure\(_t\) | exposure\(_w\) | exposure\(_t\) |
| DBpedia    | word    | 1        | -             | 1e-06         | -              |
|            | transition | -     | 0.1           | -             | 1e-06         |
|            | both    | 0.2      | 0.05          | 1e-06         | 1e-06         |
| CoNLL-2003 | word    | 0.8      | -             | 1e-04         | -              |
| MIT Movie  | transition | -     | 0.001         | -             | 1e-05         |

Table 9: Zero-shot NET results for DBpedia seen and rare entities. The first part corresponds to word pruning, the second part to transition pruning, and the last part to both word and transition pruning.

| Pruning | Type      | Seen | Rare | Support |
|---------|-----------|------|------|---------|
| Word    | person    | 0.88 | 0.64 | 10000   |
|         | location  | 0.80 | 0.62 | 10000   |
|         | organisation | 0.76 | 0.67 | 10000   |
|         | creative work | 0.69 | 0.37 | 10000   |
|         | macro-average | 0.78 | 0.58 | 40000   |
| Transition | person    | 0.84 | 0.79 | 10000   |
|          | location  | 0.83 | 0.75 | 10000   |
|          | organisation | 0.78 | 0.70 | 7014    |
|          | creative work | 0.63 | 0.55 | 6012    |
|          | macro-average | 0.77 | 0.70 | 33026   |
| Complete | person    | 0.90 | 0.81 | 2368    |
|          | location  | 0.82 | 0.64 | 1112    |
|          | organisation | 0.76 | 0.56 | 780     |
|          | creative work | 0.60 | 0.40 | 589     |
|          | macro-average | 0.77 | 0.60 | 4849    |