MELM: Data Augmentation with Masked Entity Language Modeling for Cross-lingual NER

Ran Zhou ♣♥ Ruidan He♠♥ Xin Li♠♥ Lidong Bing♠♥ Erik Cambria♥ Luo Si♥ Chunyan Miao♥
♣♥ Nanyang Technological University, Singapore
♠♥ DAMO Academy, Alibaba Group
♥ DAMO Academy, Alibaba Group

{ran.zhou, ruidan.he, xinting.lx, l.bing, luo.si}@alibaba-inc.com
{cambria, ascymiao}@ntu.edu.sg

Abstract

Data augmentation for cross-lingual NER requires fine-grained control over token labels of the augmented text. Existing augmentation approach based on masked language modeling may replace a labeled entity with words of a different class, which makes the augmented sentence incompatible with the original label sequence, and thus hurts the performance. We propose a data augmentation framework with Masked-Entity Language Modeling (MELM) which effectively ensures the replacing entities fit the original labels. Specifically, MELM linearizes NER labels into sentence context, and thus the fine-tuned MELM is able to predict masked tokens by explicitly conditioning on their labels. Our MELM is agnostic to the source of data to be augmented. Specifically, when MELM is applied to augment training data of the source language, it achieves up to 3.5% F1 score improvement for cross-lingual NER. When unlabeled target data is available and MELM can be further applied to augment pseudo-labeled target data, the performance gain reaches 5.7%. Moreover, MELM consistently outperforms multiple baseline methods for data augmentation.

1 Introduction

Named entity recognition (NER) is a fundamental NLP task which aims to locate named entity mentions and classify them into predefined categories. As a subtask of information extraction, it serves as a key building block for information retrieval (Banerjee et al., 2019), question answering (Fabbri et al., 2020) and text summarization systems (Nallapati et al., 2016) etc. However, apart from several high-resource languages, the majority of languages have limited amount of labeled data. Since manually annotating sufficient labeled data for each language is expensive, zero-shot cross-lingual NER (Mayhew et al., 2017; Jain et al., 2019; Bari et al., 2020) has gained increasing attention to transfer the knowledge from a high-resource language to low-resource languages without labeled target language data.

Despite their success, the performance of the existing zero-shot cross-lingual methods heavily relies on considerable amount of source language data. However, for certain domains (e.g. scientific literature, medical and legal documents), even the available training data from source languages is very limited (Liu et al., 2021). Therefore, it is meaningful to develop cross-lingual adaptation methods under low-resource settings.

In this work, we exploit data augmentation for cross-lingual NER adaptation where the training data of the source language is scarce. Data augmentation as an efficient solution to data scarcity, has been applied to many sentence-level tasks. Most prior works keep the sentence label unchanged and generate modification or paraphrase of the original sentence using word replacement approaches (Wei and Zou, 2019; Kobayashi, 2018; Wu et al., 2019; Kumar et al., 2020) or back-translation-based approaches (Sennrich et al., 2016; Fadaee et al., 2017; Dong et al., 2017; Yu et al., 2018). However, for token-level tasks like NER, the alignment between tokens and labels needs to be carefully handled when performing data augmentation, which makes it more challenging. For example, as shown in our experiments, simple word replacement method for NER could replace an entity token with alternatives that mismatch the original label, leading to incompatible token and label sequences.

Li et al. (2020a) fine-tuned a masked language model (MLM) (i.e. MASS (Song et al., 2019)) to conduct word replacement on only non-entity (i.e. context) tokens to avoid the label incompatibility by leaving the entities completely unchanged. However, their performance improvements for a mono-lingual task are less encouraging and our experiments also show this paradigm is less effective for cross-lingual NER. Dai and Adel (2020) created...
an entity dictionary from training set and randomly substituted entity mentions with existing entities of the same class. However there is no new entity created and the substituting entity might not fit into the context. Translate-train is another line of works for cross-lingual NER (Jain et al., 2019; Li et al., 2020b), which first translated the source training examples into the target language and then applied Fast Align (Dyer et al., 2013) to transfer labels. However, these works suffer from the translation errors and word alignment errors.

To overcome the above challenges, we propose Masked-Entity Language Modeling (MELM) as a data augmentation framework for cross-lingual NER adaptation, but without using translation engines and alignment algorithms. Different from the previous work (Li et al., 2020a), our MELM only masks the entity tokens to fine-tune an off-the-shelf Masked Language Model (MLM), then it is able to generate new training examples with substituted entities. Such strategy is based on two intuitions. First, under the zero-short adaptation, the trained NER model only has access to the source-language training data. Thus attempting to increase the context diversity (Li et al., 2020a) does not help the trained model much when evaluated on testing data from another language with different linguistic characteristics, such as word order. Secondly, under the low-resource setting, the trained NER model is only shown limited number of source-language (e.g. English) entities. Plausibly, showing the model more of such entities can enhance its cross-lingual adaptation capability when we use a pretrained cross-lingual language model (e.g. XLM-R (Conneau et al., 2020)) as the feature extractor. The reason is two-fold: (1) A new English entity or its word piece might exist as it is in the target language, such as ‘Caucasian’ in English versus ‘Caucasien’ in French; (2) Even if the new English entity is translated into the target-language characters, with the help of XLM-R, the NER model can still identify such target entities. Moreover, the entities of the same type tend to cluster together in the embedding space, thus when shown more entities of one type, the model can better recognize that type.

However, simply masking the entity tokens will also bring in the label incompatibility problem. Taking the masked sentence “⟨mask⟩ launches the new iPhone” derived from a training sentence “Cook launches the new iPhone” (see Figure 1) as an example, “Cook” is labeled with PER but the masked token could be predicted as an organization (ORG) “Apple”. To reduce the incompatibility, we impose a label constraint in MELM by injecting the label name into the input sequence of the language model. Concretely, as shown in the right part of Figure 1, we enclose the ⟨MASK⟩ symbol with special tokens denoting the entity label (i.e., ⟨B-PER⟩) and then use the transformed sentences to fine-tune the MLM. By doing so, the prediction of the masked token is conditioned on not only the context but also the entity label. Then the fine-tuned MELM can utilize the rich knowledge from pretrained MLM, such that the generated augmentation sentences present diversified entities that fit into the context with compatible labels.

Source language augmented data from MELM provides enriched entity forms in source language context, which helps cross-lingual NER tagger to
identify related entities in target languages. However, due to different linguistic patterns, the complete absence of target language knowledge is unsatisfactory for achieving optimal cross-lingual performance. Thus, in cases where unlabeled target language data is available, we can introduce the linguistic knowledge of the target language by semi-supervised learning to further bridge the language gap of the cross-lingual adaptation. Specifically, we first generate pseudo-labeled target language data using an NER tagger trained on the source language data. Then such pseudo-labeled data is used for MELM fine-tuning to generate augmented data of the target language. Then both pseudo-labeled and augmented target language data can be combined with the data in the source language for training a more accurate NER model.

To evaluate the effectiveness of our method, we conduct experiments on CoNLL and Wikiann datasets on different low-resource levels and transfer pairs. Analysis on augmented data shows that MELM generates sentences with diverse entities and compatible NER labels. As a result, when MELM is applied on augmenting the source language data, it achieves up to 3.5% F1 score improvement as compared to only using the source gold training data and constantly outperforms multiple baseline augmentation methods. When given the unlabeled target language data, MELM lifts the improvement to 5.7% by using target augmentation data. Moreover, we also apply MELM to augmenting the pseudo-labeled data by Translate-train and also observed consistent improvements.

2 MELM for Data Augmentation

We first introduce our MELM data augmentation framework for cross-lingual NER with only the source language training data. Given a set of labeled NER data $D_{gold}^S$ from the source language, MELM aims to generate a set of new examples denoted as $D_{aug}^S$ from $D_{gold}^S$. In MELM, we finetune a multilingual pretrained language model (PrLM), to generate new training examples via substituting entities while preserving the original label sequences. Specifically, for fine-tuning, we apply the objective of MLM on sentences in $D_{gold}^S$ to reconstruct the masked entity tokens. For data augmentation, given sentences with masked entity tokens, we use the fine-tuned model to make predictions and substitute the entity tokens in the original sentences with the predicted ones. Different from conventional MLM that takes only word tokens as input, we transform the training data to include the label information, such that the prediction of the masked entity token is conditioned on not only the context but also the entity label.

In a high-level overview, our MELM data augmentation process consists of 4 steps. We first perform labeled sequence linearization on $D_{gold}^S$ to inject the label information into the token sequences (Section 2.1). Then, we perform fine-tuning with MLM objective on the linearized training data (Section 2.2). With the fine-tuned model, we then conduct data generation (Section 2.3) and post-processing (Section 2.4) to obtain the final set of augmented data $D_{aug}^S$ in the source language.

2.1 Labeled Sequence Linearization

As illustrated in the right portion of Figure 1, in the linearization step, we insert the label of an entity token (i.e., tokens whose labels are not “O”) before and after it to convert a labeled sentence into a linear sequence. The inserted label tokens are considered as special vocabularies and treated equally as other context tokens during model fine-tuning, such that our fine-tuned MELM model is able to capture the dependencies between the entity labels and their corresponding tokens through distributional hypothesis (Harris, 1954). To make the special label tokens better fit into the context, we initialize their embeddings with those of words that are semantically similar to them (e.g., “person” for ⟨B − PER⟩ and ⟨I − PER⟩). This enables the MLM-based fine-tuning as described in Section 2.2 to exploit semantic information of the seed words learnt from pretraining, to facilitate prediction of the masked entity tokens.

2.2 MELM Fine-tuning

We then fine-tune our MELM with the MLM objective on the linearized training examples from the source language. At the beginning of each epoch, we randomly mask entity tokens in a sentence with a fixed masking ratio $\eta$. Note that here we only mask entity tokens, since our final goal in the generation stage is to produce new entity tokens while leaving other tokens in a sentence unchanged.

Due to data linearization, the model is trained to make predictions conditioned on both the context tokens and the label tokens. In our experiments, we demonstrate that the predictions generated by MELM are significantly more coherent with the
With the fine-tuned model, we proceed to generate augmented data from the gold training set of the source language. Given a linearized sequence with masked entity tokens, we generate the top 5 predictions on each masked token and randomly select one from them as the substitution to the original entity token. Note that we do not simply adopt the top prediction to avoid memorization and unchanged augmented data, as we fine-tune and augment on the same source language gold dataset. In this way, we obtain a new token sequence as an augmented example. We perform $R$ rounds of data augmentation with the above method, and produce $R$ augmented examples for each example in the original training set. $R$ is a hyperparameter.

To increase the diversity of augmented data, we adopt a different masking strategy from the train time. For an entity mention comprising of $n$ tokens, we first randomly sample a total number of $m$ tokens to be masked, where $m$ is a number drawn from Gaussian distribution $m \sim \mathcal{N}(\mu = \xi n, \sigma^2)$ in range $[1, n]$. $\xi$ is a coefficient controlling the Gaussian mean as a proportion of the entity length, and $\sigma^2$ is Gaussian variance. We set $\sigma = 1$ and treat $\xi$ as a hyperparameter. This ensures that single token entities will always be masked for augmentation. Meanwhile, the same sentence will have different augmentation results from different rounds of augmentation, leading to more varied augmented data.

### 2.4 Post-Processing

To remove noisy and less informative samples from the augmented data, we conduct the following post-processing: 1) We discard short sentences with less than 10 tokens. As these sentences contain limited context, the predicted entities are less reliable and could introduce noise for training the NER model; 2) A baseline NER model trained with only the source language gold data is used to assign NER tags to the augmented sentences. We only keep the augmented sentences whose predicted labels are consistent with their original labels.

After post-processing, the remaining source language augmented samples $D_{aug}^S$ are merged with the source language gold training data $D_{gold}^S$ for training the final NER tagger. Note that we also remove sentences with only O ("Other") tag in the label sequence before our MELM fine-tuning since they do not contain any named entities to be augmented.

### 3 Semi-supervised Data Augmentation with MELM

In Section 2, we have described the process of augmenting the labeled source data under our proposed MELM framework which introduces more source entities. Cross-lingual NER tagger also relies on the context patterns of the target language for prediction, which is usually disparate between the source and target languages. Therefore, it is remarkably beneficial to utilize the knowledge of the unlabeled target language data to which we often have access. Our method can be easily extended to augment the target data in this scenario with two steps. In the first step, we assign pseudo labels to the unlabeled target data. We denote the resulting dataset as $D_{pseudo}^T$. Then, we can apply the proposed MELM on $D_{pseudo}^T$ to obtain the augmented dataset $D_{aug}^T$.

We firstly train an NER tagger based on multilingual models like XLM-R on the source language data (i.e. $D_{gold}^S$ and $D_{aug}^S$) and use it to produce the pseudo label sequence for each target sentence. To make the pseudo labels more reliable, we boost the labeling confidence with multi-model agreement inspired by self-training (Xie et al., 2020; Zoph et al., 2020). Specifically, we train three NER taggers with different random seeds and use them to assign pseudo-labels individually. Then, we keep target language samples whose pseudo-label predictions are consistent across all taggers, and thus eliminating low-confidence examples.

After obtaining $D_{pseudo}^T$, we apply MELM as described in Section 2 to generate $D_{aug}^T$. Specifically, we first fine-tune the MELM on the combination of $\{D_{gold}^S, D_{pseudo}^T\}$, and then use it to generate augmented data on $D_{pseudo}^T$. Note that $D_{gold}^S$ is included for fine-tuning since we use the English dev set for model selection.

### 4 Experiments

#### 4.1 Dataset

**ConLL Dataset.** We primarily evaluate our proposed method on the CoNLL2002/2003 NER data (Tjong Kim Sang, 2002; Sang and De Meulder, 2003), which covers four languages: English (En), German (De), Spanish (Es) and Dutch (Nl). We
treat English as the source language and the others as the target languages, resulting in three transfer pairs. As we focus on the low-resource setting of source language, we randomly sample a number of 1k, 2k and 3k English sentences to simulate different low-resource levels. We also scale down the original English development set to only 500 samples as the new development set used for hyperparameter tuning and model selection.

Wikiann Dataset. We further demonstrate the effectiveness of MELM on the Wikiann NER dataset (Pan et al., 2017) from the XTREME benchmark (Hu et al., 2020), which covers a total of 40 languages. Similarly, we sample 1k, 2k training sets respectively for simulating different levels of low-resource settings.

4.2 Experimental Setting

Data Augmentation In MELM, XLM-Roberta-base (Conneau et al., 2020) with a mask-language-modeling head is used as the backbone model for data augmentation. It is fine-tuned on the masked linearized samples for 20 epochs and subsequently used for generating sequences with substituted entity tokens. We use Adam optimizer (Kingma and Ba, 2015) with batch size set to 30 and learning rate set to $1e^{-5}$.

NER Tagger For experiments on CoNLL, we use XLM-Roberta-Large (Conneau et al., 2020) with CRF head (Lample et al., 2016) as the NER tagger. For Wikiann dataset, we adopt the open-source implementation from XTREME 1, which uses XLM-Roberta-Large with a linear classification layer as the NER tagger. We use Adamw optimizer (Loshchilov and Hutter, 2018) with batch size set to 16 and learning rate set to $2e^{-5}$ following previous works. The NER tagger is trained for 10 epochs and the best model is selected using the dev set. We evaluate the trained model on test sets and report test set Micro-F1 scores averaged among 3 runs.

Hyperparameter Tuning Our proposed method introduced 3 hyperparameters: $\eta$ controls the masking ratio in the MELM training phase; $\xi$ controls the masking ratio in the data augmentation phase; and $R$ is the number of rounds for data augmentation. We first tune $\eta$ and $\xi$ with grid search. We generate one round of augmented source data on the CoNLL 1k dataset and train a NER tagger using only the augmented data. The optimal hyperparameter setting is selected based on the F1 score on the dev set. We set $\eta = 0.7$ and $\xi = 0.5$. Then we fix the value of $\eta$ and $\xi$ to tune $R$. We merge the augmented data generated from $R$ rounds on the CoNLL 1k dataset with the English gold data to train a NER tagger and select the best $R$ value according to the F1 score on the dev set. $R$ is set to 3. Details on the hyperparameter tuning procedure can be found in Appendix A.1.

4.3 Baseline Methods

We consider two settings for experiments. In setting 1 (supervised setting), only the source labeled data is available and we only augment the source examples as described in Section 2. We denote our model in this setting as MELM-en. In setting 2 (semi-supervised setting), unlabeled target data is also accessible at the training phase and we further augment the target data as described in Section 3. We denote our model in this setting as MELM-full. We compare to multiple baselines in each setting while we do not compare with translate-train-based models as our proposed method is orthogonal to them – MELM can be applied to the translated sentences to generate augmented data. More analysis is given in Section 5.

We compare with the following baselines in setting 1. **EnGold only**: The NER tagger is only trained on the source language gold data. **Label-wise mention replacement** (Dai and Adel, 2020): This method creates a dictionary of entity mentions for each entity type from the training data. Then named-entity mentions in training samples are randomly replaced with other mentions of the same entity type. **Entity replacement**: We directly utilize an MLM for data augmentation without labeled sequence linearization as used in MELM. The prediction of a masked token does not consider label information and is solely dependent on the context words. The augmented examples are used in the same way as MELM for training the NER tagger. **Context replacement** (Li et al., 2020a): This method uses an MLM to predict the masked context tokens in the training samples. The entities and their labels remain unchanged.

For setting 2, we compare the following methods to show the effectiveness of augmenting the target data. **EnGold**: Same as above. **EG+TP**: This denotes the setting where the NER tagger is trained on the English gold (EG) data $D_{gold}^S$.
and the target language pseudo-labeled data (TP) $D_{aug}^S$. **EG+TP+EA:** In addition to EG+TP, this method further adds the English augmented data $D_{aug}^S$ from MELM for training. **MELM-full (EG+TP+EA+TA):** This is our proposed method described in Section 3, which combines $D_{gold}$, $D_{pseudo}^T$, $D_{aug}^S$, and $D_{aug}^T$ as the training set.

### 4.4 Experimental Results on CoNLL

**Setting 1:** We show experiment results on CoNLL 1k, 2k, and 3k datasets in Table 1. **EnGold only** already achieves promising performance due to the strong capability of XLMR in learning cross-lingual representations. **Label-wise** brings more than 1% average performance gain on the 1k dataset. However, its entity diversity is limited to the original training set. Also, as randomly substituted entities might not fit into the context, noise in augmented data may become dominant for 2k & 3k dataset, leading to marginal gain or reduced performance. Similarly, the generated entity from **Entity Replace** might belong to a different class. In consequence, this method does not guarantee consistent improvement across different low-level settings. For **Context Replace**, despite that it guarantees the validity of entity labels, results show that minor changes to context do not bring significant performance gains on CoNLL 2k, 3k datasets.

As opposed to the baseline methods above, **MELM-en** captures both context words and label information when predicting masked entities. Thus, by taking advantage of pretrained knowledge, generated sentences possess diversified entities which correspond to the original labels. From Table 1, we observe that our method consistently outperforms all other baselines, yielding +3.5, +1.0, +1.1 average performance gains in F1 on CoNLL 1k, 2k, 3k datasets respectively compared to EnGold only. We note that MELM also further boosts NER performance on the English test set, which is desirable in mono-lingual settings as well.

**Setting 2:** Table 2 shows the experimental results under setting 2. Compared with **EnGold only** baseline, the injection of target language linguistic pattern from pseudo-labeled data in EG+TP leads to substantial improvement. It yields +3.8, +2.8, +2.5 averaged improvements in the averaged F1 scores. This is because the effect of more source language data is not as significant as the target language linguistic pattern. However when MELM is also applied to the target language data, i.e., **EG+TP+EA**, only marginally improves over **EG+TP**. This is because the effect of more source language data is not as significant as the target language linguistic pattern. We note that MELM also further boosts NER performance on the English test set, which is desirable in mono-lingual settings as well.

In comparison with **EG+TP**, augmented data **EA & TA** accounts for +1.9, +1.1, and +1.3 gain for 1k, 2k, and 3k, respectively. The contribution of augmented data in the semi-supervised setting is smaller compared with the supervised setting (i.e., setting 1), because (1) there is still a moderate level of noise in pseudo-labels even after filtering by agreement; (2) As GoldEn-only performance on target languages is significantly lower than English, labels assigned to target language augmented data contain a considerable amount of noise, which propagates into augmented data after filtering.

### 4.5 Experimental Results on Wikiann

To demonstrate the effectiveness of MELM on a wider range of languages, we present experiment-
Table 3: Data Augmentation Results on WikianN. Languages from the same linguistic family as English (i.e. Indo-European) are highlighted in green. Several distant languages are highlighted in red.

| Method | 1k | 2k | 3k |
|--------|----|----|----|
| De     | 73.63 | 69.81 | 75.83 |
| ES     | 57.06 | 53.91 | 64.65 |
| NL     | 50.54 | 47.45 | 56.85 |
| Avg    | 61.85 | 59.61 | 66.87 |

Table 4: Case study of top-5 predictions by MLM and MELM. Predictions that do not belong to the original class are highlighted in red.

| Text | Label | German | Arabic | English | French | Dutch | Indian | Spanish | Russian | Italian | Portuguese | Turkish |
|------|-------|--------|--------|---------|--------|-------|--------|---------|---------|---------|-----------|---------|
| MELM | Steve, Robert | Jobs, Apple, Microsoft | B-PER, B-ORG | O | O | O | O | O | O | O | O | O |
| MLM  | Steve, Robert | Jobs, Apple, Google | B-PER, B-ORG | B-MISC | B-LOC | O | O | O | O | O | O | O |

Table 5: Percentage of augmented data without entity label change. Augmentation with MELM incurs less label changes as compared to directly using MLM for augmentation.

| Method | 1k | 2k | 3k |
|--------|----|----|----|
| Entity Replace | 50.86% | 45.45% | 47.35% |
| MELM | 81.95% | 85.06% | 87.26% |

Table 6: Comparison between translate-train method and semi-supervised MELM. Translate-train trains separate models for each target language, using a combination of English gold data and target translated data.

| Method | De | En | Nl | Avg |
|--------|----|----|----|-----|
| 1k EG+TT | 74.45 | 75.88 | 78.40 | 76.24 |
| EG+TT+EA+TA | 77.10 | 78.86 | 79.65 | 78.54 |
| 2k EG+TT | 74.45 | 75.88 | 78.40 | 76.24 |
| EG+TT+EA+TA | 77.10 | 78.86 | 79.65 | 78.54 |
| 3k EG+TT | 74.45 | 75.88 | 78.40 | 76.24 |
| EG+TT+EA+TA | 77.10 | 78.86 | 79.65 | 78.54 |

5 Further Analysis

5.1 Entity-Label Alignment

One of our main motivations in this work is to constrain the prediction of masked entity tokens to be conditioned on their labels, such that there are less unmatched token-label pairs due to entity token changes. Apart from quantitative results shown in Section 4.4 and 4.5, we further conduct several analysis on the augmented data to verify the effectiveness of our method in achieving this objective.

As MELM randomly selects from the top 5 predictions, we take a closer look at the top 5 predictions from MELM and compare them with top 5 predictions when directly augmenting using an MLM. Examples in Table 4. qualitatively show that. As original MLM predictions are only conditioned on the context words, a considerable amount of predicted words do not belong to the original entity class. However when the label information is explicitly injected using MELM, the fine-tuned model learns to predict an entity that matches the original entity label and fits in the context as well. It is noteworthy that MELM leverages pretrained knowledge to generate real-world entities that do not exist in the original NER dataset (such as ‘Steve Martin’ (American actor) and ‘Steve Reich’ (American composer)). As a result, MELM successfully resolves the challenge of token-label mismatch after token replacement and produces high-quality and diversified augmented data.

We also quantitatively analyze the frequency of label changes. In our post-processing steps, an EnGold-only model is used to tag and filter augmented samples. As this baseline model already achieves exceptional results on the source language (F1≈ 90), the assigned label can be approximately
Table 7: We generate augmented data using MELM on translated CoNLL 1k dataset in all three target languages. For fair comparison, the translate-train method is also trained on the combination of English gold data and translated data in all target languages concurrently.

| Method             | De  | Es  | Nl  | Avg |
|--------------------|-----|-----|-----|-----|
| EG+mTT             | 73.42 | 72.71 | 76.74 | 74.29 |
| MELM on EG & mTT   | 73.76 | 75.02 | 77.74 | 75.51 |

considered as ground-truth label of the augmented token. Therefore, we take the percentage of augmented data whose assigned labels match the original label sequence, as a measure of the augmentation method’s capability to maintain token-label compatibility. As shown in Table 5, when an MLM is directly used for predicting masked entity tokens, nearly half of the sentences are subject to label changes because an out-of-class token is predicted. In contrast, in MELM, the majority of augmented data is compatible with the original label sequence. Therefore, we conclude that MELM is indeed capable of conditioning on the label information and the augmented sentences match the original label sequences in most cases.

5.2 MELM on Translated Data

On token-level cross-lingual tasks, translate-train methods utilize well-developed translation systems and carefully-designed word-alignment techniques to generate labeled translated sentences. MELM under semi-supervised settings already outperforms an in-house SOTA translate-train method on CoNLL 1k, 2k and 3k datasets, as shown in Table 6 where EG+TT refers to the combination of EnGold and translated target data. Although MELM does not rely on translation systems, it can be integrated with translate-train methods to generate augmented data on translated sentences as well. Therefore, we implement MELM on the combination of English 1k gold data and its translated sentences in all target languages to generate multi-lingual augmented data, which allows us to train a “one model for all” NER tagger. As shown in Table 7, using our MELM for augmenting further improves the performance of the multi-lingual translate-train method by +1.2 on average.

6 Related Works

Word-replacement-based Data Augmentation

Wei and Zou (2019) applied synonym replacement based on WordNet (Miller, 1998) to sentence classification tasks, and achieved comparable results with only 50% full dataset. However, the diversity of synonym replacement is limited by the knowledge base. Kobayashi (2018) generated augmented data for sentence classification tasks, by replacing words with paradigmatically-related words using a bi-directional LSTM language model conditioned on the sentence label. Wu et al. (2019) extended language-model-based word replacement from LSTM to Transformer models (BERT) and added sentence label embedding to all tokens, to condition word replacement on the sentence label. Kumar et al. (2020) further generalized language-model-based word replacement to both auto-encoding and auto-regressive transformer models. They prepended the sentence with sentence label during training and augmentation, to incorporate sentence class information.

Translation-based data augmentation

Back-translation (Sennrich et al., 2016; Fadaee et al., 2017; Dong et al., 2017; Yu et al., 2018) translates source language sentences into a target language, and subsequently back to the source language. Translated sentences preserve the overall semantics of the original sentences and it has been proven successful on translation, QA and sentence classification.

7 Conclusions

We have proposed a data augmentation framework for low-resource cross-lingual NER. Our method fine-tunes an MLM to efficiently capture label information when predicting masked entity tokens. As a result, augmented data presents diversified entities which are coherent with the original label sequence. The proposed framework has demonstrated encouraging performance improvement in various low-resource settings and across a wide range of target languages. It can also be extended to semi-supervised settings to further improve the cross-lingual adaptability.

Acknowledgements

This research is partly supported by the Alibaba-NTU Singapore Joint Research Institute, Nanyang Technological University.

---

3Details of our in-house translate-train method can be found in Appendix A.2.
References

Partha Sarathy Banerjee, Baisakhi Chakraborty, Deepak Tripathi, Hardik Gupta, and Sourabh S Kumar. 2019. A retrieval-based approach on question and answering and ner for unstructured information without using sql. *Wireless Personal Communications*, 108(3):1909–1931.

M Saiful Bari, Muhammad Tasnim Mohiuddin, and Shafiq Joty. 2020. Multimix: A robust data augmentation strategy for cross-lingual nlp. *arXiv preprint arXiv:2004.13240*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Gaëlle Gauza, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *ACL*.

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*, pages 3861–3867.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *EMNLP*.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *NAACL*, pages 644–648.

Alexander Richard Fabbri, Patrick Ng, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. Template-based question generation from retrieved sentences for improved unsupervised question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4508–4513.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *ICML*.

Alankar Jain, Bhargavi Paranjape, and Zachary C. Lipton. 2019. Entity projection via machine translation for cross-lingual NER. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1083–1092, Hong Kong, China. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *HLT-NAACL*.

Kun Li, Chengbo Chen, Xiaojun Quan, Qing Ling, and Yan Song. 2020a. Conditional augmentation for aspect term extraction via masked sequence-to-sequence generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7056–7066.

Xin Li, Lidong Bing, Wenhuan Zhang, Zheng Li, and Wai Lam. 2020b. Unsupervised cross-lingual adaptation for sequence tagging and beyond. *arXiv preprint arXiv:2010.12405*.

Linlin Liu, Bosheng Ding, Lidong Bing, Shafiq Joty, Luo Si, and Chunyan Miao. 2021. Mulda: A multilingual data augmentation framework for low-resource cross-lingual ner. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics, ACL*, volume 21.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Stephen Mayhew, Chen-Tse Tsai, and Dan Roth. 2017. Cheap translation for cross-lingual named entity recognition. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2536–2545, Copenhagen, Denmark. Association for Computational Linguistics.

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.00623*.

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*
Erik 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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In 54th Annual Meeting of the Association for Computational Linguistics, pages 86–96. Association for Computational Linguistics (ACL).

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In International Conference on Machine Learning, pages 5926–5936. PMLR.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002).

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 (EMNLP-IJCNLP), pages 6382–6388.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In International Conference on Computational Science, pages 84–95. Springer.

Jiateng Xie, Zhitlin Yang, Graham Neubig, Noah A Smith, and Jaime G Carbonell. 2018. Neural cross-lingual named entity recognition with minimal resources. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 369–379.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020. Self-training with noisy student improves imagenet classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10687–10698.

Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. 2018. Fast and accurate reading comprehension by combining self-attention and convolution. In International Conference on Learning Representations.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. 2020. Rethinking pre-training and self-training. Advances in Neural Information Processing Systems, 33.
A Appendix

A.1 Hyperparameter Tuning

Masking hyperparameters. To determine the optimal setting for fine-tune mask rate $\eta$ and augmentation masking coefficient $\xi$, we conduct a grid search on both hyperparameters in range $[0.3, 0.5, 0.7]$. We finetune MLM and generate English augmented data on CoNLL 1k following our method in Section 2. The augmented data is used to train a NER tagger and its performance on English dev set is recorded. As shown in Table 8, we achieve the best dev set F1 when $\eta = 0.7$ and $\xi = 0.5$, which is adopted for the rest of this work.

| $\eta$ | 0.3 | 0.5 | 0.7 |
|--------|-----|-----|-----|
| 0.3    | 76.90 | 75.64 | 78.08 |
| 0.5    | 76.16 | 78.06 | 78.56 |
| 0.7    | 75.94 | 78.09 | 78.37 |

Table 8: Dev set F1 for masking hyperparameter tuning.

Number of augmentation rounds. Merging augmented data from multiple rounds increase entity diversity until it saturates at certain point. Continuing adding in more augmented data begins to amplify the noise in augmented data and leads to decreasing performance. To determine the optimum number of augmentation rounds $R$, we merge different amount of augmented data with English gold data to train a NER tagger, with $R$ ranging from 1 to 6. As shown in Table 9, dev set F1 increases with increasing amount of augmented data until $R=3$, and starts to drop further beyond. Therefore, we choose $R = 3$ for all of our experiments.

| $R$ | 1 | 2 | 3 | 4 | 5 | 6 |
|-----|---|---|---|---|---|---|
| Dev F1 | 92.35 | 92.36 | 92.84 | 92.72 | 92.59 | 92.39 |

Table 9: Dev set F1 for number of augmentation rounds.

A.2 Translate-train Method

We describe the in-house SOTA translate-train method used in Section 5.2 below. The method first replace entities in the sentence with class-specific placeholders. The modified sentence is then translated with Google translate and the placeholders indicates the entities’ position in the translated sentence. Secondly, entities in the original sentence is enclosed with brackets and the entire sentence is translated. The span of words enclosed by brackets in the translated sentence is treated as translation of the entity. Lastly, the translated entity in step 2 is inserted to the placeholder’s location in the translated sentence from step 1. The entity class represented by the placeholder is inherited as label for the inserted entity translation.

As shown in Table 10, the method above outperforms multiple previous SOTA methods when implemented on CoNLL full dataset. The paper for this method is currently under review.

A.3 Statistics for Reproducibility

In Table 11, we show validation F1 for EnGold-only and MELM-en on CoNLL 1k, 2k, 3k datasets. We also summarize the estimated time for fine-tuning MELM and the number of parameters used.

| Method | De | Es | Nl | Avg |
|--------|----|----|----|-----|
| (Mayhew et al., 2017) | 60.1 | 65.0 | 67.6 | 64.23 |
| (Xie et al., 2018) | 57.8 | 72.4 | 70.4 | 66.87 |
| (Jain et al., 2019) | 61.5 | 73.5 | 69.9 | 68.3 |
| (Bari et al., 2020) | 65.24 | 75.93 | 74.61 | 71.93 |
| (Jain et al., 2019) | 66.90 | 70.49 | 73.46 | 70.28 |
| (Li et al., 2020b) | 70.99 | 74.64 | 76.63 | 74.09 |
| in-house method | **73.89** | 73.58 | **79.75** | **75.74** |

Table 10: Comparison of the in-house SOTA translate-train method used with previous SOTA methods.

| Dev F1 | MELM Train Time | #Parameter |
|--------|-----------------|------------|
| EnGold-only | MELM-en |
| 1k | 91.80 | 93.69 | ~ 20min | 270M |
| 2k | 94.34 | 94.60 | ~ 35min | 270M |
| 3k | 95.26 | 95.12 | ~ 45min | 270M |

Table 11: Statistics for reproducibility.

A.4 Computing Infrastructure

Our experiments are conducted on NVIDIA V100 GPU.