Few-shot Learning for Sumerian Named Entity Recognition

Guanghai Wang Yudong Liu and **James Hearne**
Computer Science Department
Western Washington University
Bellingham, Washington 98225
{wangg5,liuy2,hearne}@wwu.edu

Abstract

This paper presents our study in exploring the task of named entity recognition (NER) in a low resource setting, focusing on few-shot learning on the Sumerian NER task. The Sumerian language is deemed as an extremely low-resource language due to that (1) it is a long dead language, (2) highly skilled language experts are extremely scarce. NER on Sumerian text is important in that it helps identify the actors and entities active in a given period of time from the collections of tens of thousands of texts in building socio-economic networks of the archives of interest. As a text classification task, NER tends to become challenging when the amount of annotated data is limited or the model is required to handle new classes. The Sumerian NER is no exception. In this work, we propose to use two few-shot learning systems, ProtoBERT and NNShot, to the Sumerian NER task. Our experiments show that the ProtoBERT NER generally outperforms both the NNShot NER and the fully supervised BERT NER in low resource settings on the predictions of rare classes. In particular, F1-score of ProtoBERT on unseen entity types on our test set has achieved 89.6% that is significantly better than the F1-score of 84.3% of the BERT NER.

1 Introduction

Named Entity Recognition (NER), as a fundamental task in Natural Language Processing, aims to locate and classify named entities such as people, organizations, and locations, etc. The Ur III period (ca. 2112-2004 BC), spanning about 100 years, has a particularly rich source of texts, comprising at least 100,000 documents. These are primarily financial records and potentially support investigations of economic activity in Ancient Mesopotamian society. To give but one example, Liu (2021) aims to do a prosopographical study of individuals delivering animals to the Puzriš-Dagan organization during the Ur III period and identifies the individuals, their family relations and royal status of the historical actors delivering animals, as well as the variety of animals involved. NER applied to this domain can efficiently help Assyriologists recover and analyze the social-economical activities and thus provide a better understanding of the social organization and dynamics of ancient Mesopotamian history.

There is a broad effort in the community of Assyriologists, in collaboration of Computer Scientists, to build reproducible socio-economic networks from the Ur III archives (Journal et al., 2021). This effort shows that the application of NER to these texts is of great use in the quantitative study of Assyriology.

In this work, we conduct experiments to apply models that are based on prototypical networks (Snell et al., 2017) and nearest neighbour classification to the Sumerian NER task. Specifically, we adapt two few-shot learning systems, ProtoBERT (Ding et al., 2021) and NNShot (Yang and Katiyar, 2020), to the Sumerian NER task and have achieved good performance in prediction of rare classes. In summary, our contributions are as follows: (1) We construct two few-shot learning systems, ProtoBERT and NNShot, and apply them on the Sumerian NER task. To the best of our knowledge, this is the first work exploring Sumerian NER task using the few-shot learning approach. (2) We demonstrate that the ProtoBERT approach considerably and consistently outperforms the fully supervised BERT-based model and has shown to be well-suited for prediction of rare class with few labelled examples.

2 Previous Work

Previous studies on Sumerian NER are few, partially due to the lack of language resources and meaningful collaborations between researchers in Computer Science and Assyriology. The few studies include Luo et al. (2015) and Liu et al. (2016). Luo et al. (2015) uses the DL-CoTrain algorithm for personal name identification to minimize the use of the annotated data. The system achieves a high recall (92.5%) and a low precision (56.0%). Liu et al. (2016) chooses to fully utilize the annotated data by applying a wide range of supervised algorithms, including Decision Tree, Gradient Boosting, Logistic Regression, Naive Bayes, SVM and Random Forest, to predict personal names. The supervised approach shows an opposite behavior with a low recall (around 65%) and a high precision (around 86%). A more recent work (Bansal et al., 2021) investigates the Sumerian Machine Translation task in low-resource settings. They also built a variety of algorithms, including HMM, Rules+CRF, Bi-LSTM+CRF, FLAIR and RoBERTa, on the POS and NER tasks on the Sumerian dataset. For the NER task, RoBERTa achieves the best F1-score (95.3%) on a set of 12 entity types and the simple CRF model with well-defined rules significantly outperforms the rest of the models and is the second best with F1-score of 91.3%. We adapt their labelled Sumerian dataset in this work. As they only reported the overall F-scores of those NER systems on the 12 entity types without describing how the dataset is split, their result is not directly comparable with ours. More details about the dataset will be given in Section 3.

One of the key problems for low-resource NER is the lack of annotated data. As one of the common strategies, cross-lingual NER attempts to address this challenge by transferring knowledge from one or more high-resource source languages with abundant annotated data to a low-resource target language with few or no labels. The knowledge transfer is either through annotation projection from the source language to the target language (Bharadwaj et al., 2016; Xie et al., 2018; Feng et al., 2018; Rahimi et al., 2019) or through using a shared encoder in a multi-task architecture (Lin et al., 2018; Kruengkrai et al., 2020). Along the research line of enabling parameter reuse across a variety of tasks, Pfeiffer et al. (2020) proposes MAD-X, an adapter-based framework which includes language adapters, task adapters and invertible adapters, in a multilingual context. MAD-X outperforms the state of the art in cross-lingual transfer on NER across diverse languages. However, the highest F1-score is achieved on Arabic which is 59.41%. For other low-resource languages, such as Icelandic, Quechua and so on, the F1-scores are mostly around 30-50%. Cross-lingual methods have achieved notable success, but in certain circumstances, such as insufficient pre-training corpora or when the target language is far from the source language, their performance suffers. Sumerian language, as a long dead language, suffers both which makes the cross-lingual methods not readily apply.

Few-shot classification (Vinyals et al., 2016; Bao et al., 2019) can effectively recognize new classes from very few labelled examples and thus has recently drawn a lot of attention. Snell et al. (2017) proposed Prototypical Networks based on the idea that there exists an embedding space in which images of the same class cluster around a single prototype representation for each class. In other words, two images of the same class should be close to each other, and two images of the different class will be far away. Adapting this idea from image classification, Fritzler et al. (2019); Hou et al. (2020); Ding et al. (2021) address the few-shot NER problem and have achieved considerable success. Yang and Katiyar (2020) proposed token-level nearest neighbor classification based methods for the few-shot NER problem to address some potential issues of prototypical NER in learning class prototypes, such as learning a noisy prototype of the 'O' class.

Usually the overall F1-scores are high if BERT is chosen as backbone encoder in deep learning NER systems (Devlin et al., 2019). However, Tänzer et al. (2022) demonstrates that BERT fails to predict minority classes when the number of examples is limited. They observe that BERT needs at least 25 examples of a minority label to start learning on the CoNNL-03 dataset (Sang and De Meulder, 2003). If the examples are fewer than 25, the F1-score will be 0. When the examples exceed 100, the performance improves rapidly. They also observe similar phenomena on other datasets. For example, learning on the JNLPBA dataset (Collier and Kim, 2004) requires at least 50 examples. They also construct a prototypical few-shot learning model to overcome BERT's limitation. The results show the few-shot learning model consistently surpasses the performance of BERT on minority classes. For instance, it is outperforming BERT by 40 F1 points on LOC class when the dataset has 15 sentences containing that class. All this has shown that few-shot learning is well suited for the setting when the number of labelled examples is very constrained, which further justifies our choice of exploring this approach in the Sumerian NER task. And our experimental results have also echoed their findings.

The paper is organized as follows. In Section 3 we give a more detailed description of the Sumerian NER task and the dataset. In Section 4 we describe our models. Section 5 contains the description of our experimental setup and the report and analysis of the results. We conclude our work with some discussion of the future work in Section 6.

3 Dataset and task description

The dataset we used in this work is originally from the CDLI database (Cuneiform Digital Library Initiative `http://cdli.ucla.edu/`). CDLI is a project that curates an electronic documentation of ancient cuneiform texts, comprised of cuneiform texts, images, transliterations and sundry information concerning the Ur III period and its immediate aftermath. It is a joint project of the University of California, Los Angeles, the University of Oxford, and the Max Planck Institute for the History of Science, Berlin.

Although the texts we are investigating were originally written in cuneiform script, CDLI provides them in transliterated form, using the English alphabet. Fig. 1 shows a tablet from CDLI repository, (id P407107). An image of the original tablet with its cuneiform inscription is on the left; the transliteration is in the middle and the modern English translation appears on the right.

As aforementioned, we adapt the labelled Sumerian dataset and the tagset directly from Bansal et al. (2021). The dataset has 22,728 sentences, 61,478 tokens, and 12 entity types (not including the O type that indicates a word is not a named entity of interest). All these entity types, their meaning and counts in the dataset are shown in Table 1. The tablet in Fig. 1 demonstrates an example with multiple named entity types in it. According to a domain expert, *den-lil2-la2* in line 2 on the obverse is labelled as the named entity tag 'DN' (Divine Name), *ki-maszki* in line 3 labelled as 'GN' (Geographical Name), *a-mur-dsuen* and *ur-ku3-nun-na* in line 2 and 3 on the reverse are labelled as 'PN' (Personal Name), and *ses-da-gu7* in line 4 labelled as 'MN' (Month Name). The task of a few-shot Sumerian NER tagger is to identify these named entity types from the transliterations of tablets based on a few labelled examples.

The counts in Table 1 show that the dataset is quite unbalanced. Some entity types have many

Tag	Meaning	Count
DN	`Divine Name`	`900`
FN	`Field Name`	`1,463`
GN	`Geographical Name`	`1,351`
PN	`Personal Name`	`17,729`
RN	`Royal Name`	`150`
SN	`Settlement Name`	`521`
WN	`Watercourse Name`	`304`
EN	`Ethnos Name`	`60`
MN	`Month Name`	`79`
ON	`Object Name`	`18`
TN	`Temple Name`	`60`
O	`Others`	`38,822`
AN	`Agricultural Name`	`1`

Table 1: Twelve NER tags and O-tag, their meanings and counts in the dataset

more labelled examples than others. For example, entity types 'EN', 'MN', 'ON' and 'TN' only have tens of labelled examples. The least entity type is 'AN' which only has 1 example. On the contrary, 'PN' has a dominant number of examples in the dataset, over half of that of the non-named entity type 'O'. We decide to discard 'AN', 'ON' and 'EN' entity types from our training and test process. For tag 'AN' and 'ON', the number of their labelled examples is too low to enable an effective episodic-based few-shot learning process. Even though 'EN' tag has the same number of examples as the 'TN' tag, because of the data splitting and relabelling issues described in Sec. 5.1 and Sec. 5.3, we choose to drop it from our tag set as well. However, we still report the experimental results both without and with the 'EN' tag in Table 6 and Table 7, respectively. More details about data splitting and relabeling can be found in Section 5.1.

4 Methods

In the following, we will describe three models applied to the low-resource Sumerian NER task. They are BERT+LC, ProtoBERT and NNShot. As Transformer-based pre-trained language models (Devlin et al., 2019) have shown significant impact on the NER task, a pre-trained language model (PLM) on Sumerian will also be used in our NER models. In our experiment, we adapt a PLM on Sumerian explored in Bansal et al. (2021). The PLM is pre-trained using RoBERTa (Liu et al., 2019) on their Sumerian monolingual dataset.

Cuneiform Tablet	Transliteration	English Translation
	&P407107 = HSS 68, 209	
	@tablet	
	@obverse	
	1. 4(u) la2 1(disz@t) udu	39 sheep
	2. kasz-de2-a {d}en-lil2-la2	for banquet of Enlil
	3. u4 ki-masz{ki} ba-hul	when Kimash was destroyed
	4. u4 2(u) 4(disz)-kam	on the 24th day
	@reverse	
	1. zi-ga	were withdrawn
	2. bala a-mur-{d}suen	which were bala-tax of Amur-Suen
	3. ki ur-ku3-nun-na	from Ur-kununa
	blank space	
	4. iti ses-da-gu7	Month: eating sheshda
	5. mu us2-sa ur-bi2-lum{ki} ba-hul	Year after: Urbilum was destroyed

Figure 1: Tablet no. P407107 inscribed with the original Sumerian cuneiform script, the digitized transliteration, and human-translated English text line by line.

4.1 BERT+LC

Although a supervised setting is not the main goal of this work, it is nevertheless interesting to explore the standard setup of using the BERT model with a linear classifier built on top of its encoding representations, and compare it with the few-shot learning models. The model is trained to minimize the cross-entropy loss on the given training data. More details on the hyper-parameters and data setup can be found in Sec. 4.4 and Sec. 5.1, respectively.

4.2 ProtoBERT

ProtoBERT (Ding et al., 2021) is a few-shot learning model that combines the few-shot capabilities of prototypical networks (Snell et al., 2017) with the BERT's pre-trained knowledge. The model aims to build an embedding space through the training process so all the inputs can be clustered around its own "prototype" that represents the centroid of the class each input is associated with. Classifying a new input can then be done by finding its closest centroid and being assigned with the corresponding entity type. The training process is organized into a series of "episodes". Each episode consists of a support set and a query set that are randomly sampled from the training set. As a support set contains a limited number of "training" examples and a query set "test" examples, each episode essentially mimics the test-time scenario in a few-shot learning setting. In an N-way K-shot learning setup, each support set has N classes and K samples per class and the query set has N classes as well.

In our implementation, we follow the algorithm in (Ding et al., 2021) and run 500 episodes for training. In this model, for each class c, its prototype p_c is calculated by averaging the embeddings of examples that belong to class c in the support set S:

$$p_c = \frac{1}{|S_c|} \sum_{x \in S_c} f(x) \qquad (1)$$

where S_c denotes the set of all elements in S that belong to class c and function f denotes the BERT architecture augmented with a linear classifier. The model parameter of f is updated after each episode in the training process by minimizing the cross-entropy loss between the probability calculated through softmax and the one-hot ground-truth label of x.

After computing all prototypes in a support set S, we compute the distance from each input x in the query set Q to each prototype. As used in Ding et al. (2021), we also use the squared Euclidean distance as the metric function $d(f(x), p) = \|f(x) - p\|_2^2$. Once we get the distances between x and all the prototypes, a softmax function is used to compute the prediction distribution of x over all prototypes. The entity type of the nearest prototype is the prediction of x.

4.3 NNShot

NNShot (Yang and Katiyar, 2020) is a few-shot learning method based on token-level nearest neighbor classification. Unlike ProtoBERT where the training classes are clustered based on the token representations, NNShot does the inference on a query example directly based on the nearest neighbor metric. That is, NNShot simply computes the distance score between example x in the query set and all examples in the support set. It then assigns x the label of the example in the support set that is closest to x. In this work, we use the same distance metric as we use in ProtoBERT.

4.4 Hyper-parameters

We experimentally set a fixed collection of hyper-parameters across all the datasets. The Adam optimizer (Kingma and Ba, 2015) is used in the training stage. The learning rate is $1e^{-3}$. For BERT+LC, we set the batch size as 16. For few-shot learning systems, we use 2-way 5 $\sim$ 10 shot setting as suggested in Ding et al. (2021) for building a support set. This strategy allows each class in a support set to have a variable number of examples between 5 to 10, which effectively alleviates the sampling constraint between the two classes. All the models are implemented using the Hugging Face library[1].

5 Experiments

5.1 Data and tag processing

In the following, we describe how we split the tag set and generate our training, dev and test datasets accordingly for each model.

5.1.1 Tagset and dataset splitting

We largely follow the work of (Ding et al., 2021) for our tag set and data set splitting. We first divide the 12 entity types into three mutually disjoint subsets so the entity types in the test set are “new” classes or “unseen” in the training set, and vice versa. In practice, new entity types may appear in an existing or new data set or a new domain where no insufficient number of annotated examples has become available. To be aligned with a realistic setting, we reserve the entity types that have fewer examples for the test set, and those that have more examples for the training set. Based on the characteristics of our dataset, and common practice, for each specific entity type, we placed 5 to 10 examples. However, it turns out that the numbers of examples of entity types ‘AN’ and ‘ON’ are too low (1 and 18, respectively) for our few-shot learning models to get stable results. Thus we drop these two entity types from our tag set and filter out those sentences that have ‘AN’ or ‘ON’ entity type when we construct training, dev and test sets.

With this setup, we generate the train, dev and test sets where each set only contains instances of its own pre-assigned entity types. As the average sentence length in our dataset is quite small which is around three, all the entity types except for ‘EN’ do not co-occur with other entity types in a same sentence. However, almost all of the sentences containing ‘EN’ also contain other entity types, including ‘PN’, ‘GN’, etc. Relabelling (Yang and Katiyar, 2020) as a common strategy in few-shot learning systems to get mutually disjoint subsets is when a sentence has more than one entity type, any entity type that does not belong to the pre-assigned set is relabelled as ‘O’ type. We follow this process for the ‘EN’ tag and conduct the experiment. As the number of ‘EN’ tokens is only 60 out of 61,478 in the entire dataset, and ‘EN’ is the only tag that involves relabelling, we also experimentally exclude ‘EN’ and conduct the experiment. Experiments show a significant improvement in system performance (24 point increase in F-1) without ‘EN’ and relabelling. In the setup without relabelling, we drop ‘EN’ along with ‘AN’ and ‘ON’ from the 12 entity types which leaves us 9 types among which the top-4 are ‘DN’, ‘FN’, ‘GN’ and ‘PN’ and are assigned to the training set, ‘MN’ and ‘TN’ to the test set, and ‘RN’, ‘SN’ and ‘WN’ to the dev set. Type ‘O’ is present across all the three subsets. In total, 73 sentences are removed from the dataset owing to this process, accounting for around 0.3% of the total number of the sentences in the dataset. We first report the experimental results of all the models in the setting of not including ‘EN’ in Sec. 5.2. The results with ‘EN’ are presented and discussed in Sec 5.3 where the influence of relabelling is discussed in detail.

5.1.2 Data and tags for BERT+LC

The remaining data is split into training and test sets based on their pre-assigned entity types. Since BERT+LC is fully supervised, it cannot handle unseen classes in the test phase. For that reason, a few examples of ‘MN’ and ‘TN’ need to be reinserted into the training set. Because our dataset is very small compared to other widely studied languages and BERT+LC requires more data than the few-shot learning setting, we omit the dev set at this step. That means BERT+LC’s training set contains all 9 entity types. However, we only include 8 examples of ‘MN’ and ‘TN’ in this training set to make it comparable with the few-shot learning models that only use 5 $\sim$ 10 examples in each

[1]https://huggingface.co

Model	Dev	Test
BERT+LC	——	0.843
NNShot	0.714	0.857
ProtoBERT	**0.823**	**0.896**

Table 2: F1-scores for all models on test set and few-shot models on dev set.

Model	Entity	P	R	F1
BERT+LC	MN	**1.0**	0.914	**0.955**
	TN	**1.0**	0.378	0.549
NNShot	MN	0.869	0.883	0.876
	TN	0.735	0.926	**0.820**
ProtoBERT	MN	0.949	**0.933**	0.941
	TN	0.703	**0.963**	0.813

Table 3: Precision (P), Recall (R) and F1-score of individual entity types on test set.

Model	Entity	P	R	F1
NNShot	SN	0.595	0.581	0.588
	WN	**0.789**	0.818	0.803
	RN	0.636	0.840	0.724
ProtoBERT	SN	**0.714**	**0.789**	**0.750**
	WN	0.765	**0.912**	**0.832**
	RN	**0.857**	**0.960**	**0.906**

Table 4: Precision (P), Recall (R) and F1-score of individual entity types on dev set.

support set for each entity type.

5.1.3 Data and tags for few-shot learning systems

Section 5.1.2 describes the augmentation of the training set for the BERT+LC model but the test set remains the same across the 3 models. To construct the training and dev sets for the few-shot learning models, we select sentences containing ‘DN’, ‘FN’, ‘GN’ and ‘PN’ for the training set and sentences having ‘RN’, ‘SN’ and ‘WN’ for the dev set.

Fig. 2 summarizes our process of splitting the set of entity tags and data for the three models. Without including ‘EN’ and relabelling, the training sets for the BERT+LC model and the two few-shot learning systems have 22,530 sentences and 21,642 sentences, respectively. The dev set has 498 sentences, and test set 107 sentences.

5.2 Results

Table 2 summarizes the overall results of the three models on the test dataset, and the results of the two few-shot learning models on the dev set, when not including ‘EN’. As shown in Fig. 2 and described in Sec. 5.1, the training dataset that used by BERT+LC model is the combination of training set used by ProtoBERT and NNShot, dev set, 8 instances of entity type ‘MN’ and 8 instances of entity type ‘TN’ but the test data remains the same across the three models. In our few-shot systems, we use a 2-way 5 $\sim$ 10 shot setting. All the F1-scores we report are micro averaged F1-score.

As shown, both ProtoBERT and NNShot perform better than BERT+LC on the test set. NNShot outperforms BERT+LC by 1.4% F1-score and ProtoBERT outperforms BERT+LC by 5.3% F1-score. The gap between NNShot and ProtoBERT becomes more evident on the dev set with ProtoBERT outperforming NNShot by over 10% F1-score. This suggests that ProtoBERT can outperform NNShot by a larger margin when they run on a larger dataset.

We further analyze the performance of the three models on the individual entity types on the dev and test sets. The results are summarized in Table 3 and Table 4, respectively. Table 3 shows that predictions on ‘MN’ are overwhelmingly better than that on ‘TN’. We believe this is mainly because ‘MN’ as month name is a much easier entity type to identify than ‘TN’ (a temple name). Among the three models, BERT+LC system produces the best F1-score on ‘MN’ that is 1.4% higher than that of ProtoBERT. However, BERT+LC produces the worst performance on ‘TN’ with an F1-score of 54.9%, around 27% lower than that of NNShot and ProtoBERT. This is mainly due to its low recall on ‘TN’ even though its precision is 100%. A further post-processing step often takes place when conducting NER using Sumerian data: we allow a domain expert to go over the automatically identified name list (or a sample of the list) for further verification. We believe a system that has a higher recall is more useful in practice than a system that has a 100% precision but low recall. That said, we think ProtoBERT has its own advantages in practice than the other two systems in low-shot settings. This is consistently suggested by Table 3 and Table 4 with the high recall scores of ProtoBERT in all the individual entity types across dev and test set. Table 4 shows that ProtoBERT dominantly outperforms NNShot on all the individual entity types on dev set in F1-score and recall. The only place where ProtoBERT falls behind NNShot is on ‘WN’ by 2.4% in precision.

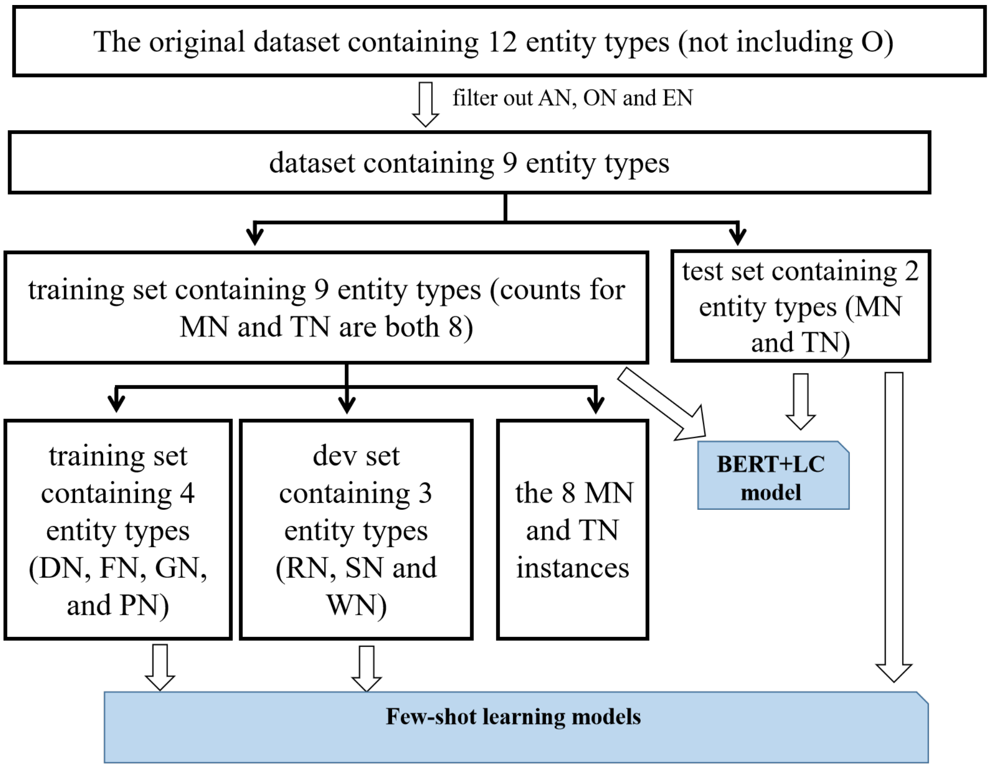
The original dataset containing 12 entity types (not including O)
filter out AN, ON and EN
dataset containing 9 entity types
training set containing 9 entity types (counts for MN and TN are both 8)
test set containing 2 entity types (MN and TN)
training set containing 4 entity types (DN, FN, GN, and PN)
dev set containing 3 entity types (RN, SN and WN)
the 8 MN and TN instances
BERT+LC model
Few-shot learning models

Figure 2: The process of tag and data splitting for three models

Model	Dev	Test
BERT+LC	——	**0.822**
NNShot	0.714	0.821
ProtoBERT	**0.823**	0.659

Table 5: F1-score of few-shot learning models on test set with EN.

5.3 Discussion on the influence of relabeling

NER is a sequence labeling problem and it is very common that a sentence contains several different entity types. For few-shot learning systems, we keep entity types in training, dev and test sets mutually disjoint because we want to test the systems on unseen entity types. A common strategy for avoiding the same entity types from occurring in different subsets is relabeling. Tokens in the training set whose labels belong to the test set are relabelled to 'O' type. Same operation is performed on test set and dev set to keep these subsets mutually disjoint. Fortunately, in our Sumerian dataset, the sentences are normally short and the majority of them only contain one of the twelve types. For most of the entity types, we can easily select sentences that only contain one entity type and no need to relabel any tokens in those sentences. However, 'EN' is an exception. Almost every sentence that has 'EN' entity type has the existence of multiple other entity types, which means all those entity types should be relabelled to 'O' when we include 'EN' in our target tag set. As 'EN' was initially assigned as an entity type for the test set, we did our first experiment with 'EN' included in the test set and ran all the three models on this set. Table 5 shows that the performance of ProtoBERT drops dramatically. F1-score drops to 65.6% (with 'EN') from 89.6% (without 'EN'). In this setting, the result of BERT+LC is produced by adding 8 randomly sampled 'EN' examples to the model's training set, and the rest of 'EN' examples goes to the test set.

Table 6 shows the confusion matrix calculated on the test set both without and with the 'EN' type to see how the results of ProtoBERT are allocated. The first column of the table with entity types is the gold labels in test set. The first row shows what label each gold label was predicted to be by the system. As shown in the table, with 'EN' included in the test set, 54 'O' are labeled to 'TN' and 52 'O' are labeled to 'EN'. That means with the inclusion of 'EN' many more false positive for 'TN' and 'EN' are produced. We believe this is mainly caused by the fact that almost all the sentences that have 'EN' also have many other entity types and these entity types are relabelled to 'O'. In ProtoBERT, when

Test set without EN			
	O	MN	TN
O (118)	108	3	7
MN (60)	0	56	4
TN (27)	1	0	26

Test set with EN				
	O	MN	TN	EN
O (225)	157	2	**54**	**52**
MN (60)	1	58	1	0
TN (27)	2	0	25	0
EN (50)	**12**	0	0	38

Table 6: Prediction of ProtoBERT on test set without and with EN.

Test set without EN			
	O	MN	TN
O (118)	109	7	2
MN (60)	0	53	7
TN (27)	1	1	25

Test set with EN				
	O	MN	TN	EN
O (225)	251	1	6	7
MN (60)	6	50	0	4
TN (27)	7	0	20	0
EN (50)	12	0	0	38

Table 7: Prediction of NNShot on test set without and with EN.

we calculate prototypes for each class, we average all the tokens in support set with the same entity type. After we relabel some tokens to 'O', the prototype of 'O' becomes noisy. That's why the model often gets confused between 'O' type and other types such as 'EN' or 'TN', which leads to poor performance of a model. Again, 'MN' as an easy entity type shows to be stable and is not affected by this relabelling as much.

Table 7 shows that the influence of relabeling for NNShot is not as obvious as that for ProtoBERT. The main reason is that the model is based on token-level nearest neighbor classification. When we query a token, it goes to find its closest example and uses its type which can counteract to a certain extent the effect of 'O' type relabeling issue.

Previous work of few-shot learning systems on English also shows the performance is not as good as expected (Fritzler et al., 2019; Huang et al., 2020; Ding et al., 2021). As the prototypes in their work are also learnt from a similar relabelling process, it could be one of the reasons that affects the system performance. Yang and Katiyar (2020) proposes STRUCTSHOT for few-shot NER to better model the label dependencies in a sentence. Although the label dependency issue in Sumerian NER is not as outstanding, it still exists. We are planning to leave it as future work to further investigate effective ways to deal with the relabelling issues caused by the 'O' type.

6 Conclusions and Future Work

We have applied three models, BERT+LC, NNShot and ProtoBERT, to explore the Sumerian NER in low resource settings, and have presented our preliminary results. This is the first work of exploring few-shot NER on the Sumerian language dataset. Our experiments show that ProtoBERT as a few-shot learning model has consistently outperformed the fully supervised model BERT+LC model in few-shot settings and has generally achieved better performance than NNShot. Though as a token-level nearest neighbour classification method, NNShot is less sensitive to the noisy 'O' type that is introduced by the relabeling step, it may not be as stable as ProtoBERT owing to the nearest neighbor mechanism in the training stage. We show that BERT-LC fails to do a good job in learning more examples in few-shot settings. While we investigate the efficacy of prototypical networks-based ProtoBERT and nearest neighbour metric-based NNShot learning models in the few-shot Sumerian NER task, it will be particularly interesting to 1) extend our work to a larger test set; 2) explore new methods such as STRUCTSHOT (Yang and Katiyar, 2020) to solve the noisy 'O' type issue introduced by relabelling; 3) experiment on using more sophisticated cross-lingual approaches including adapter-based models on Sumerian NER.

Acknowledgements

We thank Dr. ChangYu Liu, Professor at Nanjing Normal University, for his assistance with the tablet example in this paper.

References

Rachit Bansal, Himanshu Choudhary, Ravneet Punia, Niko Schenk, Jacob L Dahl, and Émilie Pagé-Perron. 2021. How low is too low? a computational perspective on extremely low-resource languages. arXiv:2105.14515.

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2019. Few-shot text classification with distributional signatures. *arXiv preprint arXiv:1908.06039*.

Akash Bharadwaj, David R Mortensen, Chris Dyer, and Jaime G Carbonell. 2016. Phonologically aware neural model for named entity recognition in low resource transfer settings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1462–1472.

Nigel Collier and Jin-Dong Kim. 2004. Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pages 73–78, Geneva, Switzerland. COLING.

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.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Hai-Tao Zheng, and Zhiyuan Liu. 2021. Few-nerd: A few-shot named entity recognition dataset. arXiv:2105.07464.

Xiaocheng Feng, Xiachong Feng, Bing Qin, Zhangyin Feng, and Ting Liu. 2018. Improving low resource named entity recognition using cross-lingual knowledge transfer. In *IJCAI*, volume 1, pages 4071–4077.

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*, SAC '19, page 993–1000, New York, NY, USA. Association for Computing Machinery.

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. *arXiv preprint arXiv:2006.05702*.

Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2020. Few-shot named entity recognition: A comprehensive study. arXiv:2012.14978.

IDEAH Journal, Anya Kulikov, Adam Anderson, and Niek Veldhuis. 2021. Sumerian networks: Classifying text groups in the drehem archives. *IDEAH*. Https://ideah.pubpub.org/pub/q22859lx.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. arXiv:1412.6980.

Canasai Kruengkrai, Thien Hai Nguyen, Sharifah Mahani Aljunied, and Lidong Bing. 2020. Improving low-resource named entity recognition using joint sentence and token labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5898–5905, Online. Association for Computational Linguistics.

Ying Lin, Shengqi Yang, Veselin Stoyanov, and Heng Ji. 2018. A multi-lingual multi-task architecture for low-resource sequence labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 799–809.

Changyu Liu. 2021. Prosopography of individuals delivering animals to puzriš-dagan in ur iii mesopotamia. *Akkadica*, 142:113–142.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv:1907.11692.

Yudong Liu, James Hearne, and Bryan Conrad. 2016. Recognizing proper names in ur iii texts through supervised learning. In *Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference*, page 535–540, Florida, USA.

Liang Luo, Yudong Liu, James Hearne, and Clinton Burkhart. 2015. Unsupervised sumerian personal name recognition. In *Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference*, page 193–198, Florida, USA.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for ner. *arXiv preprint arXiv:1902.00193*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. arXiv:1703.05175.

Michael Tänzer, Sebastian Ruder, and Marek Rei. 2022. Memorisation versus generalisation in pre-trained language modelsl perspective on extremely low-resource languages. arXiv:2105.00828.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29.

Jiateng Xie, Zhilin Yang, Graham Neubig, Noah A Smith, and Jaime Carbonell. 2018. Neural cross-lingual named entity recognition with minimal resources. *arXiv preprint arXiv:1808.09861*.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. arXiv:2010.02405.