MEM-BERT: Masked Entity Model-Based Transformers for Realtime Personalization

Qingxiaoyang Zhu, Nehal Bengre, Tapas Kanungo

Abstract Transformer models are typically trained offline and then fine-tuned for a specific task. Once fine-tuned, the transformer models are fixed and cannot incorporate new partial information like personal contact, location, or device list. In a real-world scenario like IoT device control via voice assistants like Siri and Bixby, however, personal information recognition accuracy is crucial from a user experience point of view. We propose a masked entity transformer model that can leverage local personal information during inference time without re-training the model. We show that this modeling technique improves the recognition accuracy of entities in a user's personal database while maintaining the recognition accuracy of non-personal entities. We simulate the problem scenario using the CoNNL dataset and provide quantitative results.

1 Introduction

Voice assistants like Apple Siri [1], Amazon Alexa [2] and Google Assistant [3] are becoming ubiquitous in our daily lives. We use voice assistants to find information for us, control IoT devices like lights, thermostats, and TVs, and make phone calls. While there has been great progress in algorithms for understanding task-oriented languages that are typically used in such voice assistants, the identification of

Qingxiaoyang Zhu
University of California Davis / Davis, CA, USA, e-mail: qinzhu@ucdavis.edu

Nehal Bengre
Samsung Research America / Mountain View, CA, USA e-mail: n.bengre@samsung.com

Tapas Kanungo
Samsung Research America / Mountain View, CA, USA e-mail: tapas.k@samsung.com

[1] https://www.apple.com/siri/

[2] https://alexa.amazon.com/

[3] https://assistant.google.com/

personal information can still be challenging. For example, non-Christian names in contact lists and device names defined/provided by users, that are not well covered in the training data, can have lower accuracy due to bias in training data or ambiguity in entity names (e.g. device names that can also be person names). Such recognition issues can lead to lower user satisfaction and the adoption of voice assistants.

In this paper, we formalize and address the issue of real-time personalization of deep-learned transformer-based NLU algorithms. More specifically, if a user has contact names, device names, addresses, etc. in a personal database (PDB), how can one leverage the PDB during inference time to improve the overall entity recognition accuracy? Because such information in the PDB is personalized and may not be covered in the training data, how can the model accompany such diverse information and maintain the understanding accuracy of recognizing both non-personalized and personalized chunks? And since information in the PDB is updated by users very frequently, and users expect the voice assistant to immediately recognize the entities in the PDB with high accuracy, retraining of a deep-learned transformer model is not viable. Furthermore, users typically do not like to share their personal information with the service provider for training purposes due to privacy reasons. Therefore, the PDB is not accessible during the training process of the NLU model and is only available during inference time.

Our solution to this conundrum is to train an entity recognition model using lexicalized as well as de-lexicalized (masked) entities, and de-lexicalize the entities found in the PDB during inference time. This approach improves the entity recognition accuracy of the entities in the personal information while maintaining the accuracy of the other entities. Furthermore, no re-training is required and the model can handle unseen personal information during the inference stage, which has a real-time impact. As a side benefit, since no training is required for personal data, there is no need to share the PDB entries with the service provider, which thereby mitigates privacy concerns. Our contributions are as follows:

- We formalize and empirically prove the real-time personalization problem
- We propose a MEM-BERT with an entity-masking training mechanism to alleviate natural language understanding problems for personal data
- We provided experimental validation using the CoNLL2002 shared task dataset.

We first cover the related literature in Section 2. Next, in Section 3, we describe the personalization problem in the context of voice assistants. We then define our masked entity model and the transformer training process in Section 4. The experimental setup and protocol are described in Section 5. The results of our experiments and error analysis are described in Section 6 and Section 7.

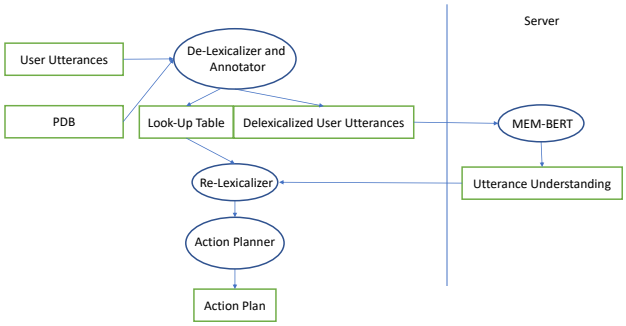

Fig. 1: In this figure, we see that during inference time the delexicalizer masks the entities in the utterance, which appear in PDB, and maintains a Look-Up table for the delexicalized phrases. Then, the utterance with delexicalized entities is sent to MEM-BERT on the server. The marked-up utterance (the utterance with NLU result) is returned to the user's device, which then joins the marked-up utterance with the Look-Up table to relexicalize the masked entities and sends it for action execution. Blue blocks indicate operation processes, while green blocks indicate intermediate data that is input to or generated by the operation processes.

2 Related Work

Transformers [24, 2, 11, 17, 18, 15] are deep-learned architectures that have led to a paradigm shift in the way we do natural language processing research. These models can be pre-trained on a large text corpus and subsequently "fine-tuned" on domain-specific problems. While these models have dramatically increased the recognition accuracy, the training infrastructure and the time required to update the model for a small update in the training data are not practical for real-time updates like additions to the contact information of a mobile device. Even on GPUs, the transformer training algorithms can take several minutes for just one epoch. Thus in the context of real-time personalization where we expect that after adding a name to the contact list the NLU algorithm can immediately leverage the contact information during the process of inference is not practical.

To demonstrate our approach to the problem of using partial information (part of the utterance appears in the personal information), we apply our technique to the named-entity recognition problem and in particular use the CoNLL dataset [21, 20]. While there are other datasets like ATIS [7], SNIPS [1] and MASSIVE [5], the CoNLL dataset has a significant number of name entities compared to ATIS and SNIPS and thus is a better approximation to the personalization task we are addressing in this paper.

Bias in transformer models is well-known in the literature [6]. Researchers have also attempted to mitigate gender bias [13]. However, these algorithms are offline

and cannot be used in situations where we get new information about names in real-time.

Gazetteers-based algorithms can be used for personalization and improving entity recognition accuracy [12, 3, 4, 14]. These techniques require retraining with the updated gazetteers which again cannot enable real-time updates. Similarly, personalized transformers [26] improve recognition accuracy by using temporal information. However, the technique again requires the neural model to be retrained for every update.

Entities convey indispensable information in utterances and need to be paid attention to. Researchers have investigated the way to let the language model make sense of entities in the medical domain by introducing an XML tag to quote entity terms [10]. Similarly, previous work focuses on entities during data augmentation to generate high-quality and variant data. However, these methods are not real-time oriented when new entities come into the scope.

Our approach, in contrast, does not require the retraining of the transformer model after each update of the PDB. Our approach could handle general templates that include personal information types and unify various personal information into the information type, which is similar to the concept conveyed by previous work [23, 22]. Our approach doesn't send the PDB information to the inference algorithm, which helps to preserve privacy. Furthermore, the approach is PDB language-agnostic and thus improves the recognition accuracy of the PDB entities irrespective of their source language. This is an important practical benefit since incorrect PDB entity recognition can lead to bad user experience.

3 The Realtime Personalization Problem

Voice assistants are invoked using a task-oriented language [19]. The voice commands include slots containing named entities like people names, device names, location, time, etc. Although entities can be categorized according to the entity type, such entities in the surface form, if used for natural language understanding tasks (e.g. named entity recognition, intent detection, slot filling and so on) can be impacted by the huge variation of entity values. For instance, a *people* entity type could be realized by various names of people or a *device* entity type could be realized by numerous names of devices.

Sample voice assistant command utterances including personalization data are shown in Figure 2. While deep-learned transformer models can recognize these entities, when the entities are not from a demographic that it was trained on, the accuracy can drop.

Users typically refer to a person, a device, a place, or a subject in general by using a preferred personalized identification phrase most of the time. The contact lists, device names roster, data type list, etc. are such personalized data that is typically available to the NLU system via a personal database (PDB), which maintains the mapping between the user's personalized identification of the entity with that

entity type. These PDBs are updated intermittently by the users when they add a new device or a contact into their PDB. Furthermore, the names in the PDB can be from a different demographic than what the original transformer was trained on.

A natural language understanding model trained on the existing task-oriented language may not perform well on the new utterances with user-personalized entities that are given in real-time (directly conduct inference using the trained model). The question to ask is: Can the NLU system leverage the PDB in a way so that there is no need to re-train the transformer (after initial training)? In addition, the model should be able to leverage existing data without collecting new personalized data to train the model in order to protect users' privacy. While leveraging PDB, the model should maintain the original performance.

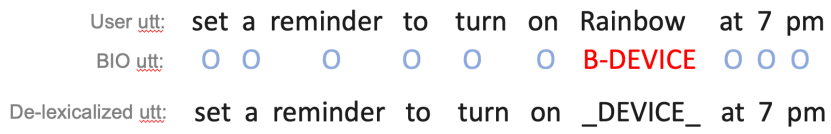

Fig. 2: An example of utterances in voice assistants

4 Method

4.1 Personalized Contact List Synthesis

Due to privacy policies, we are not able to publish users' data. To simulate the name personalization scenario, where users maintain a personalized contact list of person names and use those personalized names in commands to voice assistant, we leverage a public multilingual person name dataset as our PDB[4] to extract name lists across nationalities. The PDB includes 195,313 unique names covering 107 languages. We selected Chinese names as a personalized contact name list because it's the second-largest usage group and the Chinese names in this collection are not well-represented in the original corpora (BookCorpus and English Wikipedia) that BERT was trained on.

To replicate the scenario where users use personalized person names, we synthesize the personalized data by replacing the original *person* named entity in English utterances with a random Chinese name from the Chinese name candidates.

[4] The dataset is available here:
https://www.kaggle.com/datasets/wchowdhu/multilingual-person-name-dataset

4.2 Masked Entity Model

For the large pre-trained language model BERT [2]-based NLU system, which leverages attention mechanism to perform semantic understanding of the overall sentence and has the ability to extract contextualized token-level representations, we propose a Masked Entity Training Task to enable the model to process personalized utterances in real-time without model retraining. We use base BERT (model structure unchanged) as our test bed and named entity recognition (NER) task as our down-streaming task. Inspired by the mask language model (MLM), we randomly mask a sequence of named entities with a special token (e.g. mask entities in a person type as special token _PERSON_ and entities in a location type as special token _LOCATION_), which represents the named entity type (slot-type) of the delexicalized sequence, with a given probability. The probability of swapping the named entity sequence is tunable as a hyperparameter and the special token is added to the original BERT tokenizer. During the learning stage, the model learns how to encode contextual information of personalized information and personal variation into an embedding of the special token through the attention mechanism. The special token is therefore trained to represent a category of name entities, instead of a specific personalized named entity. Meanwhile, the model keeps the knowledge of a non-personalized named entity or non-masked personalized named entity from the original lexical phrase. Thus, the masked entity training strategy is as follows: (1) In the training set, find the location or person entities in each utterance. (2) With a probability p, randomly swap the location or person entity with its type symbol, rather than keep it lexicalized. (3) Instead of using the original set for training, use the randomly delexicalized utterance for training the MEM-BERT model.

By incorporating slot-type tokens through a random masking mechanism, an original utterance including personalized information may be converted into a partially masked utterance, which is one of the possible templates used for personalization. Therefore, when accessing the full training set, the model learns partial knowledge about personalization templates simultaneously and accumulates the full knowledge of all possible templates while maintaining the ability to process original natural language utterances. The randomization of masking is designed for the purpose of avoiding model collapse on a limited templated set and keeps the generalizability to compatible with non-masked natural language utterances.

5 Experimental Protocol

5.1 Dataset

We use CoNLL2003 [20] named entity tagging dataset as a base dataset for our experiments. The named entity tagging task is a sequence tagging task where each

token corresponds to one of four tags. The dataset consists of four types of entity tags - Person, Location, Organization, and Miscellaneous. Splits are as follows: Training set 14041, Validation set 3250, and Test set 3453. Table 1a indicates the train set characterization based on the 4 NER-tags. Table 1b indicates the test set characterization based on the 4 NER-tags. Table 2 shows the co-occurrence of the various NER tags.

Table 1: Train/Test set characterization showing counts and the percentage of the train cases for each of the tags over the training/test cases for the baseline. Since one utterance may contain multiple types of entity, the sum of the percentage is not equal to one.

(a) Training Set

Tags	**Count**	**Percentage**
PER	4373	31.14%
ORG	4587	32.67%
LOC	5127	36.51%
MISC	2698	19.21%

(b) Test Set

Tags	**Count**	**Percentage**
PER	1025	29.68%
ORG	1229	35.58%
LOC	1266	36.65%
MISC	563	16.30%

Table 2: Characterization of co-occurrence tags showing counts and percentage of the test cases for each of the co-occurrence tags.

Co-occurrence tags	**Count**	**Percentage**
PER_LOC	371	10.74%
PER_LOC_MISC	48	1.39%
PER_MISC	54	1.56%
PER_ORG_LOC	65	1.88%
PER_ORG_MISC	65	1.30%
PER_ORG	65	3.97%
PER_ORG_LOC_MISC	25	0.72%

To replace the English names with foreign names (in our case Chinese names), we use the Multilingual person name dataset which has name entities in 107 languages. The dataset has the binary label of identifying if the given word is a name or not. After processing the data, we were able to get 1629 unique Chinese names. These Chinese names consisted of a mixture of both Chinese characters and English characters.

While there are other types of sequence tagging tasks like slot filling, where corresponding popular datasets are ATIS [7] and SNIPS [1], the datasets used for the task do not have tags that can be personalized or replaced by foreign names/entities. Thus we did not include them in our experiments and leave them for future work.

5.2 Model Configurations used in Experimentation

Our baseline model is $BERT\_BASE$ ([2]), with 12 layers, 12 self-attention heads, and 768 hidden size. The total amount of model parameters is 85M. We added a token-level classification head for the NER task. The model is trained with $2e-5$ learning rate and 0.1 dropout rate. The number of epochs is set to 20 with an early stopping strategy, the batch size is 32 and the optimizer is Adam[8].

To test if the model trained commonly on NER task has the ability to handle real-time personalized data on the personal named entity, we first create test data (*personalized_test(PER)*) by swapping all person named entities into names in another language (we use Chinese here to prove the concept).

To handle the real-time personalization problem, data augmentation could be a common and promising strategy. To systematically investigate the effect of the delexicalization mechanism and data augmentation mechanism, we created two variations of the training set. One training set (*baseline_train_delex_PER*) is constructed by replacing all PER named entities into slot types. Another training set (*baseline_train+delex_PER*) is created by first filtering out all utterances including PER from the original training set, then duplicating the selected samples and delexicalizing them, and finally adding delexicalized samples back to the original training split. In the test split, all person named entities are delexicalized with the generic entity type tag to simulate the personalization setting by leveraging the user-provided PDB information (*baseline_test(PER)+PDB_delex*). This is only to simulate the actual real-world scenario.

To further investigate the promise of our proposed method, we created another training split (*baseline_train_delex_PER_LOC (randomly replace 40%)*) with a 40% probability to swap a PERSON named entity into a person slot type, the same strategy is applied to LOC named entity. The corresponding test split delexicalizes all PER named entities and LOC named entities to simulate the usage of PDB (*baseline_test(PERLOC)+PDB_delex*). The dataset is aligned with our proposed masked entity training task, the model accepts training data including randomly delexicalized entities with the corresponding entity type as a special token, which is not separated by the tokenizer.

6 Results

Our evaluation of NER systems follows the CoNLL2002 [21] shared task approach. The standard data format is the BIO (Begin, Inside and Outside) format. We use the seqeval [16] python package for our NER evaluation. We compute precision, recall, and F1-score for each of our experiments in Table 3. We also compute these metrics for our fine-grained tags (PER, LOC, ORG, MISC) and present the analysis in Table 4.

6.1 The Problem with Personalization

To check whether the baseline BERT model trained on the original NER training set has the ability to handle real-time personalized data, we compare the performance of the baseline model's inference on the original CoNLL2002 test split with the performance of the baseline model's inference on the personalized test split (test split with person names swapped). As shown in the first two rows in Table 3, F1-score on overall NER dropped with a corresponding 35.15% increase in error rate, which indicates that the original model has a limited ability to handle personalization data without retraining. The potential reasons are that swapped Chinese names are from an out-of-training vocabulary distribution and the model could not represent the Chinese names due lack of training samples. Even though the model might understand the subwords of the Chinese name after the tokenization, the full term is unknown within the context. As the result, the base BERT model has a weak ability to process language code switch to Chinese and any novel personalized data.

We further investigate how personalized person names affect entities in categories other than *PERSON*. The result in Table 4 shows prediction performance on fine-grained categories, indicating that the F1-score on person named entity decreased by absolute -12.5%, with a corresponding relative error rate increase of 431%. F1-score on both ORG and MISC categories also decreased slightly. Since the side effect of personalized PER entities on other entities would appear only in utterances that have at least two types of entities and one of which is PER type, the results in Table 4 are diluted by utterances that contain only one type of entity. To remove the dilution and understand the effect of personalized person names on recognizing other entity types, we filter out such utterances where non-PER entities co-occur with the PER type and calculate the NER performance. Table 5 shows the fine-grained comparison between BERT-NER on original test data and BERT-NER on personalized test data, where the F1-Score of LOC increased by 0.2% (with a relative 3.28% error decrease) while the F1-Score of MISC, ORG and PER decreased by 1.1% (relative 6.43% error rate increase), 0.1%(relative 0.67% error rate increase), 11%(relative 440% error rate increase), respectively. The observation cross-validated that BERT-NER model trained on the original data set suffers from the personalization issue on personalized entity type during the inference stage. Other entity types that co-occur with the personalized entity type can also experience degradation in accuracy.

6.2 Impact of Data Augmentation and Delexicalization

Augmenting the baseline training set with the additional dataset including additional knowledge about personalization could be a common strategy to handle performance degradation. As shown in Table 4, using additional split with person entities delexicalized and leveraging the PDB during the inference indeed

Table 3: Overall NER metrics and relative NER F1 error with respect to the baseline.

Experimental Settings-Train	Experimental Settings-Test	NER F1 (Relative error)	NER Recall	NER Precision
baseline_train	baseline_test	90.18	89.05	91.34
baseline_train	personalized_test(PER)	86.72 (35.15)	87.04	86.41
baseline_train + delex_PER	personalized_test(PER)+PDB delex	90.52 (-3.52)	89.15	91.94
baseline_train_delex_PER	personalized_test(PER)+PDB delex	90.20 (-0.21)	89.06	91.39
baseline_train_delex_PER_LOC (randomly replace 40%)	personalized_test(PER&LOC)+PDB delex	93.01 (-28.93)	92.15	93.90

Table 4: Fine-grained NER metrics per tag and relative error with respect to the baseline.

Experimental Settings-Train	Experimental Settings-Test	LOC F1 (Relative Error)	ORG F1 (Relative Error)	PER F1 (Relative Error)	MISC F1 (Relative Error)
baseline_train	baseline_test	92.5	87.3	97.1	77.1
baseline_train	personalized_test(PER)	92.7 (-2.67)	87.1 (1.57)	84.6 (431)	76.7 (1.75)
baseline_train + delex_PER	personalized_test(PER) +PDB delex	91.6 (12.00)	87.3 (0.00)	99.0 (-65.5)	76.8 (1.31)
baseline_train_delex_PER	personalized_test(PER) +PDB delex	91.3 (16.00)	86.1 (9.45)	99.9 (-96.55)	76.1 (4.37)
baseline_train_delex_PER_LOC (randomly replace 40%)	personalized_test(PER&LOC) +PDB delex	96.3 (-50.67)	89.7 (-18.9)	99.1 (-68.97)	79.0 (-8.3)

Table 5: Testing data that includes *PERSON*(PER) entity type and at least one of non-PER entity types (co-occurrence), fine-grained NER metrics per tag and relative error with respect to the baseline.

Experimental Settings-Train	Experimental Settings -Test	LOC F1 (Relative error)	ORG F1 (Relative error)	PER F1 (Relative error)	MISC F1 (Relative error)
baseline_train	baseline_test	93.9	85.0	97.5	82.9
baseline_train	personalized_test(PER)	94.1 (-3.28)	84.9 (0.67)	86.5 (440)	81.8 (6.43)

recovers from the personalization issues to some degree – the F-1 score of PER improved by absolute 1.9% (with a relative 65.5% error decrease). However, other non-personalized entities, such as LOC and MISC are recognized with slightly worse accuracy. Similar behavior is seen in the table for the direct delexicalization of person entities and leveraging the PDB during the inference, which does not increase the training time because no additional data is added to the training process.

To better understand the effect of delexicalizing PER on other types of entities, we further analyze the performance of other entity types when co-occurring with delexicalized PER in the same sentence. Table 6 with fine-grained metrics shows that delexicalizing PER tags not only improves the recognition performance of *PERSON* entities (+1.9% F1-score), it also improves the performance of LOC(+3.3% F1-score), ORG(+5.9% F1-score) and MISC(+3.7% F1-score) tags in co-occurring test cases. A possible reason could be that the model is able to conduct a better

inference due to a better understanding of personalized phrases and the whole sentence. On the other hand, delexicalization may introduce side effects to the training process which leads to a worse recognition performance on the non-personalized entities which are not co-occurring with delexicalized entities, because delexicalizing all phrases from the single entity type may reduce the diversity of the data and cause a collapse of model learning.

From the systematic experiment, we observed that leveraging delexicalization (either in a data augmentation or replacement way) and PDB information outperforms the baseline model on personalization setting (even outperforming the model on original test data), because it potentially simplified the name entity recognition task by reducing all lexicalized entities' length to be one and unified the language utterance level feature for all lexicalized entities as a special token. The model only needs to learn the meaning of the introduced special token to understand a named entity type. Therefore, the model learned the template knowledge of the sentence.

Table 6: Performance on co-occurrence tags after delexicalizing PER and comparing this with the baseline performance.

Experimental Settings -Train	**Experimental Settings -Test**	**LOC F1**	**ORG F1**	**PER F1**	**MISC F1**
baseline_train	baseline_test	93.8	83.5	97.6	81.8
baseline_train_delex_PER	personalized_test(PER) +PDB delex	97.1	89.4	99.5	85.5

6.3 Impact of Masked Entity Training (Randomly Delexicalizing PER and LOC Tags)

The proposed masked entity training task, that is training the BERT-NER model with randomly masked entities in the personalized categories (PER and LOC entity types here), and utilizing PDB information during the inference outperforms the common delexicalization strategy on recognizing personalized entities, as well as other non-personalized entity types which either co-occur with the personalization entity types in the same sentence or not. Table 3 shows a significant improvement of resolving the personalization problem on overall NER metrics (2.83% increase on overall F1-score) and Table 4 shows fine-grained improvement on other non-personalized entity types, such as ORG (2.4% increase on F1-score) and MISC entity(1.9% increase on F1-score) types. Table 4 also shows that the proposed method boosted the recognition performance of personalized entity types, namely LOC(3.8% increase on F1-score and relative error decrease by 50.67%) and PER(2% increase on F1-score and relative error decrease by 68.97%).

To understand the effect of randomly masked LOC and PER on other entity types, we analyze fine-grained metrics. Table 7 shows that delexicalizing the PER and LOC tags not only improves the performance of PER and LOC tags, but also ORG and MISC tags for co-occurring cases in the test set. The possible reason could be that model better learned to infer on the single special tokens locally (delexecalized PER and LOC here) and therefore better understands the global semantic meaning for the whole sentence, which is helpful to infer other types of NER in the same sentence (ORG and MISC here). As such, ORG and MISC slots that appear at the same time with PER and/or LOC slots in the same utterance would be benefited from delexicalized PER and LOC slots and be better recognized. Besides, Table 4 indicates that the model trained with the proposed method also performed well on other non-personalized entities that may not co-occur with personalized entity types. The possible reason is that the randomization mechanism on entity masking avoids the model degradation because of the access to diverse data and also supports the model to acquire template knowledge for handling upcoming personalization information.

The advantage of using this training setup is that it has the same number of examples as the original baseline training setup, but it performs 28.93% better than the baseline with respect to all the NER tags Table 3. In addition to this, it also performs 50.67% better than the baseline for LOC tag for the overall test set (which includes both the co-occurring and the co-occurring tags) in Table 4.

Table 7: Performance on co-occurrence tags after delexicalizing 40% PER, LOC tags and comparing this with the baseline performance.

Experimental Settings -Train	**Experimental Settings -Test**	**LOC F1**	**ORG F1**	**PER F1**	**MISC F1**
baseline_train	baseline_test	95.9	80.8	98.2	80.2
baseline_train_delex_PER_LOC (randomly replace 40%)	personalized_test(PER&LOC) +PDB delex	99.5	91.5	99.5	87.3

7 Discussion

Anyone who has used voice assistants has experienced the frustration that happens when the voice assistant does not recognize names in your contact list. This can be due to the name being from a less common demographic and the transformer not being trained on that demographic. One of the ways of having instant gratification is to have a mechanism (for example, regular expression match) by which at least all the entities in the PDB are recognized correctly as soon as an entity is added to the PDB by the user. However, this approach will not help in the recognition of the rest of the entities. By training MEM-BERT using masked entities, the model learns not only the lexicalized language model, but it also learns

the de-lexicalized language model. While we demonstrated the approach using BERT, the same approach should be possible for other BERT and GPT extensions.

We showed how a BERT-based NER system can perform poorly when provided with Chinese names. This is due to the inherent bias in the training datasets used in training BERT. Such biases will always exist in machine-learned models that are data-driven. One extension that can be tried is to use mBERT [2], which is trained using a multilingual dataset. However, it still will not have an easy way to leverage partial information available in the PDB directly without any re-training.

A possible issue with the current approach can happen if the same entity name can appear as two different entity types. For example, a name that can be either a person type or a location type. To handle such cases the PDB would have to store both types associated with the lexical form of the entity. During inference time, the context would have to be leveraged to disambiguate the type. A Viterbi- or CRF-like algorithm [25, 9] would help. A simpler solution in practice would be to allow only unique names to be added to the PDB.

8 Conclusion

We presented a transformer training technique that addresses the issues of real-time personal database updates and privacy. The MEM-BERT transformer is trained to identify masked/delexicalized entities like names and locations. This training process teaches the model to learn entity occurrence patterns and not just the surface form patterns. Thus during inference time, the entities in the personal database can be leveraged by substituting the corresponding de-lexicalized tokens. We demonstrated our approach using the BERT architecture and the CoNLL dataset. We showed that MEM-BERT can use partial information (de-lexicalized entities) and improve the recognition accuracy of entities in the personal database to close to 100% while maintaining the accuracy of other entities. An additional benefit of this approach is that since a de-lexicalized entity is sent to the inference system, and since no retraining is involved, the entities in the personal database can be secure on the users' phone or device and do not have to be shared with a service provider, reducing privacy concerns. Finally, since transformers are biased due to the bias existing in the training corpus, in the case that the model has difficulty recognizing an entity of specific demography, the user can add the entity to their personal database.

Acknowledgement

We would like to sincerely thank the reviewers for their comments and constructive feedback.

References

1. A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, et al. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*, 2018.
2. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
3. B. Fetahu, A. Fang, O. Rokhlenko, and S. Malmasi. Gazetteer enhanced named entity recognition for code-mixed web queries. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1677–1681, 2021.
4. B. Fetahu, A. Fang, O. Rokhlenko, and S. Malmasi. Dynamic gazetteer integration in multilingual models for cross-lingual and cross-domain named entity recognition. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2777–2790, 2022.
5. J. FitzGerald, C. Hench, C. Peris, S. Mackie, K. Rottmann, A. Sanchez, A. Nash, L. Urbach, V. Kakarala, R. Singh, et al. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. *arXiv preprint arXiv:2204.08582*, 2022.
6. I. Garrido-Muñoz, A. Montejo-Ráez, F. Martínez-Santiago, and L. A. Ureña-López. A survey on bias in deep nlp. *Applied Sciences*, 11(7):3184, 2021.
7. C. T. Hemphill, J. J. Godfrey, and G. R. Doddington. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*, 1990.
8. D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
9. J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.
10. C. Lin, T. Miller, D. Dligach, S. Bethard, and G. Savova. Entitybert: Entity-centric masking strategy for model pretraining for the clinical domain. Association for Computational Linguistics (ACL), 2021.
11. Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
12. S. Magnolini, V. Piccioni, V. Balaraman, M. Guerini, and B. Magnini. How to use gazetteers for entity recognition with neural models. In *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*, pages 40–49, 2019.
13. R. K. Mahabadi, Y. Belinkov, and J. Henderson. End-to-end bias mitigation by modelling biases in corpora. *arXiv preprint arXiv:1909.06321*, 2019.
14. T. Meng, A. Fang, O. Rokhlenko, and S. Malmasi. Gemnet: Effective gated gazetteer representations for recognizing complex entities in low-context input. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1499–1512, 2021.
15. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
16. H. Nakayama. seqeval: A python framework for sequence labeling evaluation, 2018. Software available from https://github.com/chakki-works/seqeval.
17. M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. arxiv 2018. *arXiv preprint arXiv:1802.05365*, 12, 1802.
18. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

19. E. Razumovskaia, G. Glavaš, O. Majewska, A. Korhonen, and I. Vulic. Crossing the conversational chasm: A primer on multilingual task-oriented dialogue systems. *arXiv preprint arXiv:2104.08570*, 2021.
20. E. F. Sang and F. De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003.
21. E. F. Tjong Kim Sang. 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)*, 2002.
22. W. Van Melle. Mycin: a knowledge-based consultation program for infectious disease diagnosis. *International journal of man-machine studies*, 10(3):313–322, 1978.
23. W. Van Melle, E. H. Shortliffe, and B. G. Buchanan. Emycin: A knowledge engineer's tool for constructing rule-based expert systems. *Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project*, pages 302–313, 1984.
24. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
25. A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269, 1967.
26. L. Wu, S. Li, C.-J. Hsieh, and J. Sharpnack. Sse-pt: Sequential recommendation via personalized transformer. In *Fourteenth ACM Conference on Recommender Systems*, pages 328–337, 2020.

9 Appendix – Experimental Setup Description

This is a section we provide more detail on the experimental setups mentioned in Tables 3, Table 4 and Table 7

- baseline_train + baseline_test: This setup consists of the train set and test set unmodified. This is our baseline.
- baseline_train + real_name_swaped_test: This setup consists of the train set unmodified. The test set has all the PER values replaced by Chinese names.
- baseline_train + plus_allname_delex + delex_test: This setup consists of a split to swap all PER values with a PER slot type in addition to the original baseline train split. The corresponding test split delexicalizes all PER named entities
- replacePERToDelex_in_baseline_train + delex_test: This setup consists of a split to swap all PER values with a PER slot type. The corresponding test split delexicalizes all PER named entities
- 0.4replacePERLOCToDelex_in_orig_train + 1delexPER_LOC_test: This setup consists of a split with 40% probability to swap a PER value with a PER slot type and LOC value with a LOC slot type. The corresponding test split delexicalizes all PER named entities and LOC named entities

Each model takes 13 minutes to finish training and takes 0.32 minutes to finish inference.

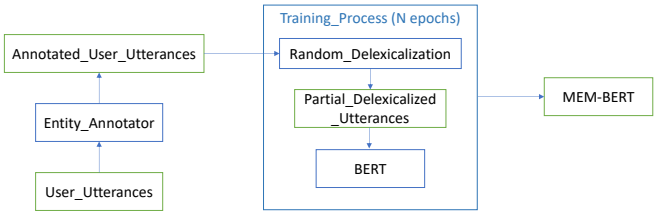

Fig. 3: In this figure, we see that during training time the delexicalizer partially randomly masks the entities in the PDB in each epoch and sends the marked-up utterances with partially delexicalized entities to BERT on the server. The training process is conducted on the server. After N epochs training, the MEM-BERT is converged for later usage on the user device.