Neural Modelling of Gazetteers under Few-Shot and Domain Adaptation Scenarios

Anonymous ACL submission

Abstract

Entity tagging in cross domain or few shot scenarios is challenging because of lack of sufficient data enveloping the entire entity vocabulary. In such cases, gazetteers helps enhancing the NER model performance by presenting additional information about the entity tag dictionary to the neural network models. Performance of gazetteer based neural network models depends on three fundamental factors: vectorization of the gazetteer information, the base encoders used on the gazetteer vector, the fusion methodologies combining gazetteer and textual learnings. The objective of this work is to understand the impact of above and also investigate the performance of an alternate late fusion ensemble technique on 3 virtual assistant related NER tagsets from food services and customer care domain

1 Introduction

Entity recognition (ER) is the task of identifying the locations of entity mentions and their types in utterances. For intelligent virtual agent (IVA) driven enterprise customer care, product entity recognition in customer utterances is important for recognizing and fulfilling customer requests. Entity recognition in these circumstances can be quite challenging because of the requirement of large set of utterances and new product types that are introduced by the enterprises. In this work, we explore various ways gazetteers may be employed in neural network models in order to ameliorate these difficulties in a variety of real world scenarios.

Following are the contributions from this work: (i) Comparison of early and late gazetteer fusion techniques on different datasets and tag sets when they are employed on biLSTM-CRFs and Transformers based named entity taggers. (ii) Development of novel ensemble-based late fusion technique (iii) Exploration of different approaches for vectorizing gazetteer features (greedy approach and sparse vector) (iv) Investigation of gazetteer models in few shot learning, domain adaptation/dependant scenarios.

2 Related work

(Meng et al., 2021) introduces a mixture of experts model for late fusion of gazetteer information to improve entity recognition. (Song et al., 2020) experiments with incorporating very large gazetteers compiled from WikiData (Vrandečić and Krötzsch, 2014) into BiLSTM-CRF models. (Rijhwani et al., 2020) look at using soft gazetteer features, derived from high resource languages, in BiLSTM-CRFs for entity tagging for low resource languages. These features are incorporated as both early fusion and also as late fusion, via use of an autoencoder in the style of (Wu et al., 2018). (Magnolini et al., 2019) experiments with an approach where gazetteer information is supplied by an auxiliary neural network classifier, termed NN_g, that is trained to distinguish words that are part of the gazetteer from words that are not. They find early fusion of NN_g-provided information performs the best. Their preliminary experiment with Transformer-based models did not show encouraging results.

3 Datasets

Datasets used in this work are from food services and customer care domain. The food services data consists of customer utterances from full customer agent dialogs involving order-taking. These dialogs are taken from customer interactions from four different restaurant chains. Dialogs from a particular chain are its own “domain” because each chain has a distinct cuisine and menu. Therefore, the food services data is divided into domains of PIZZA, BURGER, WINGS, HOT DOG. Some of the sample utterances from different domains are "do you have uh lemonade", "number two with an orange juice", "um a bacon egg and cheese biscuit". Audio versions of these dialogs are hand transcribed and the resulting text utterances hand-annotated with two kinds of tags, *Generic* and *grounding* tags. The *Generic* tags consist of tags like ENTITY, ATTRIBUTE, NUM-QTY, SIZE etc where entities correspond to orderable menu items and attributes are customizations of these items. Over 30 generic tags are used and the tag distributions (top five) for the cases S1, S2 and S3 (Table 6) are shown in Fig 1. The *grounding* tag set has specializations of the specific *generic* tags ENTITIES (e.g: Hamburger, Cheeseburger, Ketchup, or Barbecue Sauce) and ATTRIBUTES (e.g: boneless, seasoning). Overall 20 grounding tags specialize ENTITY while another 20 specialize ATTRIBUTE and the corresponding tag distribution for cases S4,S5 (Table 7) is shown in Fig 2

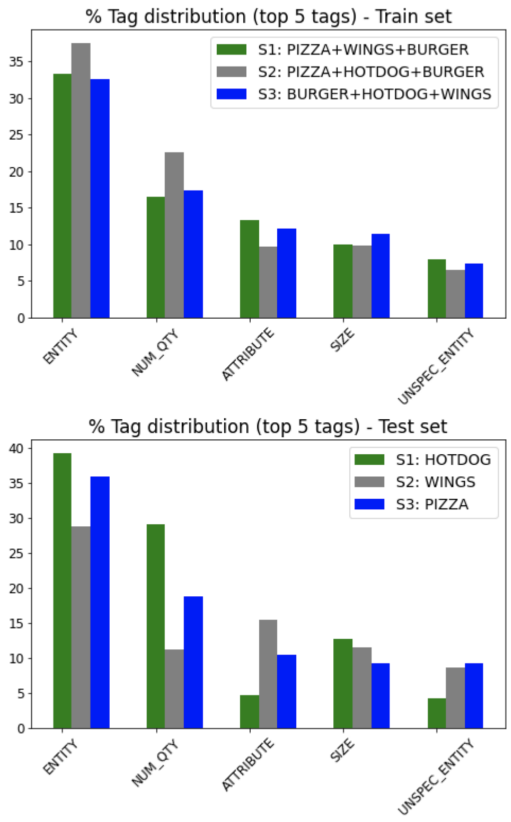

Figure 1: Food Services- Generic Labels Distribution

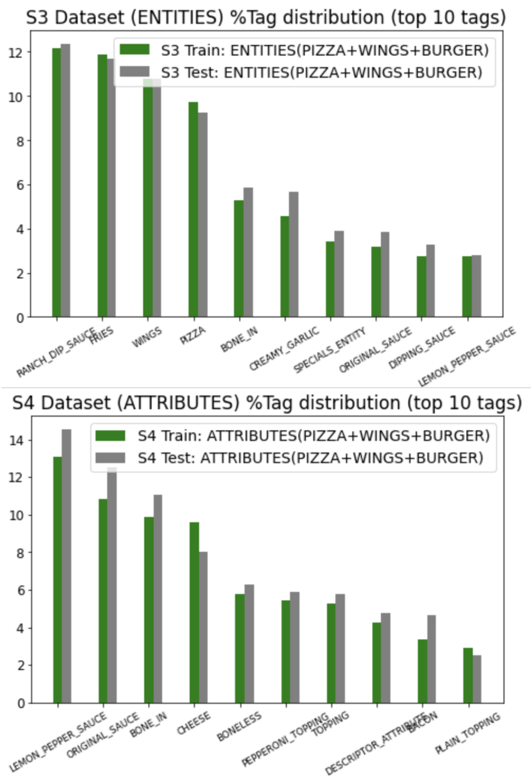

Figure 2: Food Services- Entities and Attributes Grounding Labels Distribution

The IVA customer care data (cases S6-S9, Table 8) consists of customer utterances from phone conversations that are responses to open-ended prompts of human-machine dialogs like “How may I assist you today?” Automatic Speech Recognition (ASR) is used to convert utterance audio into text with a word accuracy of about 85%. Utterances are tagged with product entities they contain. Customer utterances from two different domains are selected: ELECTRONICS (example tag tokens: app-store, phone) and APPLIANCES (example tag tokens: refrigerator, filters, freezer). In each domain, 97 utterances (product only tags) are used for training in a few-shot scenario and another 1,000 utterances are used as a test set. BIO formatting is used for tagging tokens in the case of customer care datasets and IOBES formatting for food services datasets.

4 Models

4.1 Gazetteer Vectorization

Like textual information, the representation of gazetteer inputs has an impact on overall performance. We compare two different ways by which gazetteer information can be vectorized. The first method (**Method 1**) is a simple “gazetteer tag2vec” vectorization. Each word is assigned a unique gazetteer tag based on deterministically tagging the input utterance token with a gazetteer. This approach is often used in the literature (Meng et al., 2021; Song et al., 2020; Chiu and Nichols, 2016). Our implementation greedily assigns tags according to longest word sequence match across all the gazetteers lists and overlapping matches are resolved by preferring earlier matches over the later ones. For example, the gazetteer tag for an utterance “ buy medium ice ginger tea” is “O ATTRIBUTE ENTITY ENTITY ENTITY .” and for “ can i get original ranch” is ”O O O DESCRIPTOR DIP-SAUCE”. Subsequently, the tags are mapped to an embedding layer inside the network whose dimensions are optimally found after hyper-parameter search in the range of 10,50 and 100 units.

The second vectorization method (**Method 2**) represents each word in the input utterance as a sparse vector of non-zero elements corresponding to the set of all of the gazetteer tag types occurring with the same word in the training data. Size of the vector corresponding to each word is equal to total number of tags. For example, Tables 1 and 2 show the sparse vectors corresponding to the previous example utterances. It is not the first kind of soft gazetteer representation that has been proposed (Rijhwani et al., 2020; Ding et al., 2019; Liu et al., 2019), but it is quite a bit simpler because the vector elements are strictly binary without considering the span information, and to our knowledge not previously explored.

Tags	buy	medium	ice	ginger	tea
O	1	0	0	0	0
ENTITY	0	0	1	1	1
ATTRIBUTE	0	1	1	1	0

Table 1: Method2 Gazetteer Vectorization (Food Services-Generic tag example)

Tags	can	i	get	original	ranch
O	1	1	1	0	0
BUFFALO SAUCE	0	0	0	1	0
DIP SAUCE	0	0	0	1	1
DRESSING	0	0	0	0	1
DESCRIPTOR	0	0	0	1	0

Table 2: Method2 Gazetteer Vectorization (Food Services- Entity Grounding tag example)

Both the gazetteer vectorization methodologies (**Method1+Method2**) are used for datasets with multiple labels like food services datasets (generic, grounding tags). That is, as shown in Table 3, columns 2 and 3 form the gazetteer inputs for the utterance "buy medium ice ginger tea". But only **Method1** vectorization is used in case of single label datasets like customer care datasets (*product* alone tags).

Sample Utterance- Food Services data, Generic tags			
Text	Method 1	Method 2	Gold Tags
buy	O	[1,0,0]	O
medium	ATTRIBUTE	[0,0,1]	B-ATTRIBUTE
ice	ENTITY	[0,1,1]	E-ATTRIBUTE
ginger	ENTITY	[0,1,1]	B-ENTITY
tea	ENTITY	[0,1,0]	E-ENTITY
Sample Utterance- Food Services data, Entity Grounding tags			
Text	Method 1	Method 2	Gold Tags
can	O	[1,0,0,0,0]	O
i	O	[1,0,0,0,0]	O
get	O	[1,0,0,0,0]	O
original	DESCRIPTOR	[0,1,1,0,1]	B-DIP-SAUCE
ranch	DIP SAUCE	[0,0,1,1,0]	E-DIP-SAUCE

Table 3: Text and Gazetteer sample inputs

The gazetteer vocabulary is built based on the respective domain corpus for the food services datasets corresponding to both the generic and grounding tags. For the Customer care datasets, apart from using the training corpus, product items from wikidata corresponding to home appliances (Q212920) and electric appliances (Q581105) are also added after some data cleaning.

4.2 BiLSTM versus Transformer

Entity tagging is modeled using either Bidirectional LSTM CRF or Transformer.

Model-A: Bidirectional LSTM CRF. This is a hierarchical bi-LSTM tagger for global coherence, in the style of (Collobert et al., 2011). We first construct character-compositional word embeddings following (Dos Santos and Zadrozny, 2014) and concatenate those with lookup table embedding weights obtained from glove6b-100 pre-trained word embeddings (Pennington et al., 2014). The outputs are passed to a bi-LSTM transduction layer, and finally to a CRF layer for global coherence (Ma and Hovy, 2016). For biLSTM-CRF models, based on previous studies, the learning rate (lr) is set as 0.015 with ADAM optimizer, number of layers as 2, early stopping criteria on the validation score (patience of 5 epochs) is used

Model-B: Transformer. This is an 8-layered relative positional attention based encoder-only Transformer model (Shaw et al., 2018) initialized from a pre-trained Masked Language Model (MLM) trained on a large corpus of online data including 3 years of Reddit (Henderson et al., 2019), all of Wikipedia, online forums and reviews along with task-oriented dialog data. Byte-Pair Encoding is applied on the input utterances with fixed embedding dimension of 512, maximum token length of 128 and ADAM optimizer with learning (lr) of 0.00001 is used.

4.3 Fusion Approaches

For each tagging architecture, we experiment with various early and late fusion techniques for combining word inputs and gazetteer inputs.

Model-C: Early fusion. For early fusion, the gazetteer features after vectorization explained in Section 4.1 are concatenated with the respective embeddings of the baseline models (A or B). So the neural network is jointly learned on the concatenated word and gazetteer embedding representations.

Model-D: Mixture of experts late fusion. The gazetteer embeddings after vectorization are passed through one of the encoders (Model A or B) while the textual embeddings are passed through another encoder in the parallel network. Finally, as shown in Fig 3, sigmoid gate that is applied on concatenated word (W) and gazetteer (G) encodings, is learned by the model altogether, following (Meng et al., 2021; Arnaud et al., 2019). During the learning process, the back-propagation of gradients is controlled by the sigmoid gate and hence it controls the flow of information to both the word and gazetteer in the computational graph during both training and testing.

Model-E: Ensemble-based late fusion. In this model, similar to Model-D, a parallel encoder is built for the gazetteer encoder. However, as shown in Fig 3, instead of using a gate to control the backpropogation, a random binomial variable is used for alternate switching on/off the word based encoder (A or B) and gazetteer encoder. During testing, the information/neurons from both the encoders are combined together giving equal weightage. The hypothesis of trying out the above model is to allow the encoders to learn independently without influencing each other.

As explained in Table 4, eight combinations of model architecture and fusion approach are analyzed in this paper: (A),(B),(A+C),(A+D), (A+E), (B+C),(B+D) and (B+E). For the early fusion models like (A+C) and (B+C), the same encoder(either biLSTM1 or Transformer1) is used for encoding the word and gazetteer information after concatenating all the embeddings at the early stage. For the late fusion, two different encoders are used and fusion happens late as shown in Fig 3.

4.4 Hyperparameter search

The hyperparameters and corresponding ranges are as follows: Batch sizes:- 4,8,12,24,32; biLSTM hidden units:- 256,384,512; Gazetteer layers :- 1,2; Gazetteer Method1 vectorization embedding (DSZ) sizes:- 10,50,100; Transformer Embedding dimension (512). A default dropout value of 0.2 is applied on the Gazetteer embeddings during training to address the issue of relying too much on gazetteer information. For modelling results alone that is section 5.2, hyperparameter search is done and the best hyperparameters are selected based on a validation set. Test results are obtained by averaging the results from three training runs.

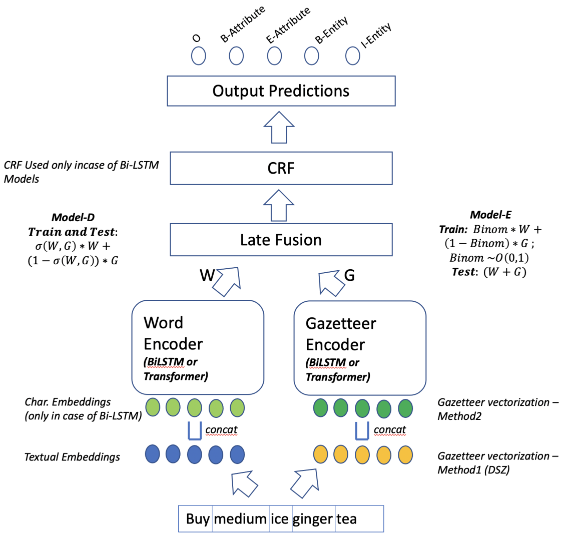
O
B-Attribute
E-Attribute
B-Entity
I-Entity
Output Predictions
CRF Used only incase of Bi-LSTM Models
CRF
Model-D
Train and Test:
σ(W,G) ∗ W +
(1 − σ(W,G)) ∗ G
Late Fusion
Model-E
Train: Binom ∗ W +
(1 − Binom) ∗ G ;
Binom ~O(0,1)
Test: (W + G)
W
G
Word Encoder (BiLSTM or Transformer)
Gazetteer Encoder (BiLSTM or Transformer)
Char. Embeddings (only in case of Bi-LSTM)
concat
concat
Gazetteer vectorization – Method2
Textual Embeddings
Gazetteer vectorization – Method1 (DSZ)
Buy medium ice ginger tea

Figure 3: Late Fusion

Mod	Word Enc	Gaz Enc	Fusion	CRF
A	biLSTM1	No	No	Yes
A+C	biLSTM1	biLSTM1	Early	Yes
A+D	biLSTM1	biLSTM2	MoE	Yes
A+E	biLSTM1	biLSTM2	Ensemble	Yes
B	Transformer1	No	No	No
B+C	Transformer1	Transformer1	Early	No
B+D	Transformer1	Transformer2	MoE	No
B+E	Transformer1	Transformer2	Ensemble	No

Table 4: Model Summary

5 Results and Discussion

We first analyze the the impact of gazetteer vectorization on food services generic tag and grounding tag dataset. Then we examine the impact of different combinations of models (Table 4) in domain adaptation scenario for food services generic tags datasets (S1, S2, S3), domain dependent scenario for food services grounding tags datasets (S4,S5), few shot indomain (S6,S8) and out-domain testing scenarios (S7,S9) for customer care datasets.

5.1 Impact of gazetteer vectorization

To understand the gazetteer vectorization impact, we experiment with different gazetteer vectorization approaches while fixing the model as Model A+C (biLSTM-CRF and early fusion). 3 cases are tested: First case is domain adaptation scenario C1 (train- PIZZA, test- HOT DOG generic tags) and two other domain dependant scenario C2 (train and test - PIZZA+WINGS+BURGER Entities), C3 (train and test - PIZZA+WINGS+BURGER Attributes). Further, the number of layers is fixed as 2, hidden units size as 256, Method1 gazetteer vectorization embedding units size (DSZ) as 100 for direct comparison. The gazetteer vocabulary for the training are based on their corresponding domain's gazetteers. Results in Table 5 show that the combination of Methods 1 and 2 (M: 1+2) performs even better. When provided to the neural network model, the different points of view conveyed by Method 1 (M:1) and Method 2 (M:2) helps the model generalize better.

Case	noGaz	M:1	M:2	M:1+2
C1	0.762	0.811	0.783	0.828
C2	0.826	0.883	0.839	0.891
C3	0.831	0.889	0.840	0.917

C1: PIZZA(train), HOT DOG (test)
C2: PIZZA+HOTDOG+BURGER - Entities (train and test)
C3: PIZZA+HOTDOG+BURGER - Attributes (train and test)

Table 5: Impact of gazetteer vectorization on F1 scores

5.2 BiLSTM versus Transformer and Fusion Approaches

Food services datasets: The models are trained on generic tags corresponding to combined multiple food services domains datasets like HOT DOGS+WINGS+PIZZA and tested on datasets corresponding to target domains like BURGER. This results in three cross domain cases: S1, S2 and S3 with the average train/test split across cases as 35k/2k utterances and tag distribution as shown in Fig 1. Modeling results over these datasets are shown in Table 6 with micro F1 scores (mean and std deviation). As can be seen, the impact of gazetteer information on model performance seems to be more pronounced with the usage of biLSTM-CRF base encoders. However, Transformer Model with Ensemble based late (B+E) fusion performs best in S1 and S3 datasets while biLSTM with Ensemble based late (A+E) fusion perform best in S2 dataset. To summarize, in all 3 cases, Ensemble late fusion model performs best.

The next set of experiments are done only on PIZZA+WINGS+BURGER datasets (both train and test) with grounding tags (as explained in Section 3) corresponding to both entities (S4) and attributes (S5) in domain dependant scenario with train/test split as 35k/5k utterances. As entities or attributes are specific to a particular domain (Fig 2), domain adaptation testing scenario is not carried out for this case. The results are shown in Table 7, wherein for both versions of Grounding tags (S4,S5), improvements using the gazetteer information are equally pronounced on models with both biLSTM-CRF and Transformers base encoders. But in both the cases S4 and S5, Transformer based late fusion models performs best with Ensemble fusion again performing on par with MoE fusion.

Customer Care Few shot datasets: In Table 8, results corresponding to models trained on in-domain ELECTRONICS few shot data, tested on in-domain ELECTRONICS domain data (S6) and out-domain APPLIANCES domain data (S7) are shown. Similar experiments considering APPLIANCES few shot data are tabulated as S8 and S9. Overall, for both in-domain testing and out-domain testing, biLSTM-CRF based gazetteer models irrespective of the fusion methodologies, show better generalization compared to that of Transformers.

Key Inferences:

- Ensemble model performs well on most of the food services generic tags datasets (domain adaptation scenarios), with best micro F1 score improvement of 1% in case S1, (model B+E) over the baseline (model B). Even in food services grounding tag datasets (domain dependant scenarios), best F1 score improvement of 8% improvement can be observed in case S5 (model B+E) with respect to the baseline (B).

- Even though Ensemble models on many occasions perform well, it still does not always provide best results in terms of consistent performance across all the datasets particularly few shot scenarios. Probable reasons could be that, in the ensemble model, since both the word/gaz learnings are combined with equal weightage during the validation phase for saving the checkpoint with best combined optimum, there could be scenarios where the optimum of the word encoder is not combined with the optimum of the gazetteer encoder.

- With regards to impact of base encoder on gazetteer model performance, biLSTM-CRF as base encoder consistently outperforms the Transformer. On an average, across all the cases (S1-S9), gazetteer models based on biLSTM-CRF performs around 6% better in micro F1 score with respect to the baseline (no gazetteer) and their best performance is observed in the few shot datasets. Transformers on other hand, on average performs around 1% better and performs less in few shot datasets. One probable reason that transformers are not able to replicate the performance, particularly in few shot scenarios could be the inductive bias of the transformer and with more number of layers/ parameters together with less data, it may not be able to transduce the gazetteer information well in either of the fusion techniques.

6 Future Work

We plan to extend this research work by developing an alternate Ensemble technique using bi-LSTM models for the gazetteer encoding and Transformer Model for textual encoding. Another parallel area of research that we plan to explore is to automate generation of template utterances for NER data augmentation using both gazetteer and textual embeddings.

7 Conclusions

- We compared gazetteer models across different datasets and scenarios, vectorizations, models, fusion methodologies.
- Providing the model with both ambiguous sparse vector and greedy tag representation of gazetteer information improves performance over either one or the other.
- We proposed a novel ensemble-based late fusion approach which performed on par or better than the other fusion approaches in food services datasets.

Case	A	(A+C)	(A+D)	(A+E)
S1	.802±.02	.81±.03	.812±.04	.832±.01
S2	.573±.01	.594±.01	.597±.00	**.603**±.01
S3	.670±.00	.698±.01	.663±.00	.662±.00
Case	B	(B+C)	(B+D)	(B+E)
S1	.840±.01	.838±.01	.838±.01	**.845**±.01
S2	.595±.00	.599±.00	.600±.01	.588±.00
S3	.735±.01	.732±.01	.726±.01	**.745**±.01

S1: PIZZA+WINGS+BURGER(train), HOT DOG (test)
S2: PIZZA+HOT DOG+BURGER(train), WINGS (test)
S3: BURGER+HOT DOG+WINGS(train), PIZZA (test)

Table 6: Food Services: Generic Labels results.

Case	A	(A+C)	(A+D)	(A+E)
S4	.836±.01	.892±.01	.893±.01	.888±.00
S5	.832±.00	.894±.01	.889±.01	.910±.01
Case	B	(B+C)	(B+D)	(B+E)
S4	.840±.00	.867±.01	**.905**±.00	.901±.01
S5	.837±.01	.847±.00	.918±.00	**.919**±.00

PIZZA+WINGS+BURGER(train/test-35k/5k utterances)
S4: Grounding Labels corresponding to Entities
S5: Grounding Labels corresponding to Attributes

Table 7: Food Services : Grounding Labels results.

Case	A	(A+C)	(A+D)	(A+E)
S6	.701 ±.02	.701±.04	.727±.00	.723±.00
S7	.157±.05	.622±.01	**.625**±.00	.606±.01
S8	.537±.05	.608±.02	.643±.01	**.647**±.00
S9	.041±.00	.472±.04	**.523**±.02	.413±.09
Case	(B)	(B+C)	(B+D)	(B+E)
S6	.735±.00	**.752**±.03	.743±.01	.737±.01
S7	.294±.05	.291±.01	.271±.03	.240±.04
S8	.597±.01	.579±.01	.614±.00	.598±.00
S9	.062±.05	.025±.02	.043±.00	.007±.00

S6: ELECTRONICS (train/test)
S7: ELECTRONICS (train) APPLIANCES (test)
S8: APPLIANCES (train/test)
S9: APPLIANCES (train) ELECTRONICS (test)

Table 8: Customer Care: Few shot results

- For both the domain adaptation and dependant scenarios, addition of gazetteer information to the model helped substantially with late fusion techniques.
- For the few-shot scenarios both in-domain and out-domain testing, adding gazetteer information consistently improved the model's performance with biLSTM-CRF's more than that of Transformer's.

7.1 References

References

Estèphe Arnaud, Arnaud Dapogny, and Kévin Bailly. 2019. Tree-gated deep mixture-of-experts for poserobust face alignment. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2(2):122–132.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537.

Ruixue Ding, Pengjun Xie, Xiaoyan Zhang, Wei Lu, Linlin Li, and Luo Si. 2019. A neural multi-digraph model for Chinese NER with gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1462–1467, Florence, Italy. Association for Computational Linguistics.

Cicero Dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *International Conference on Machine Learning*, pages 1818–1826. PMLR.

Matthew Henderson, Paweł Budzianowski, Inigo Casanueva, Sam Coope, Daniela Gerz, Girish Kumar, Nikola Mrkšić, Georgios Spithourakis, Pei-Hao Su, Ivan Vulić, et al. 2019. A repository of conversational datasets. *arXiv preprint arXiv:1904.06472*.

Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. 2019. Towards improving neural named entity recognition with gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5301–5307, Florence, Italy. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Simone Magnolini, Valerio Piccioni, Vevake Balaraman, Marco Guerini, and Bernardo Magnini. 2019. 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.

Tao Meng, Anjie Fang, Oleg Rokhlenko, and Shervin Malmasi. 2021. 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.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Shruti Rijhwani, Shuyan Zhou, Graham Neubig, and Jaime Carbonell. 2020. Soft gazetteers for low-resource named entity recognition. *arXiv preprint arXiv:2005.01866*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*.

Chan Hee Song, Dawn Lawrie, Tim Finin, and James Mayfield. 2020. Improving neural named entity recognition with gazetteers. *arXiv preprint arXiv:2003.03072*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Minghao Wu, Fei Liu, and Trevor Cohn. 2018. Evaluating the utility of hand-crafted features in sequence labelling. *arXiv preprint arXiv:1808.09075*.

A Dataset

For the food services dataset, the annotator population consisted of English-speaking workers in India, Colombia, and the United States working for a subcontracting firm. For the customer care dataset, the annotator population consisted of English-speaking workers from Colombia.

For both the food services dataset and the customer care dataset, the annotator populations were aware of how the annotations that they generate would be used. Because we indirectly employed the annotator populations for the annotation tasks, we do not have concrete information about how much they were paid, though we believe that the companies for whom they work do abide by the labor laws of the countries in which they do business.

B Compute Resources

GeForce GTX 1080 Ti GPU Machines are used with memory of 11.2 GB. Each of the food services cases took $\tilde{3}$ GPU hrs and customer care cases took 0.5 GPU hrs.