Data Augmentation through Paraphrase Generation for Real World Few Shot Entity Tagging

Anonymous EACL submission

Abstract

Few shot entity tagging is important because different applications of natural language processing typically have different semantics, necessitating custom models. Here, we study few shot entity tagging in a real world scenario insofar that the training data consists of small number of examples per entity type, every entity type has the same number of examples, and there is not any development set. We perform paraphrase generation for many different domains using a T5 model trained on generic paraphrase data. We find that this method produces gains in tagging accuracy across many different domains, and gains are accentuated with an ensemble voting approach.

1 Introduction

Entity tagging is the task of extracting entity mention spans of specific predefined types from unstructured text. Recent methods for entity tagging are typically fine-tuned on neural language models such as ELMo (Peters et al., 2018), BERT (Kenton and Toutanova, 2019), RoBERTa (Liu et al., 2019), and T5 (Raffel et al., 2020) that are pre-trained on large amounts of raw text.

Because fine-tuning usually requires manually annotated training data, and training data tagged with different entity types is often required when switching to a new domain, it is of interest to discover techniques to reduce the amount of manual annotation required for fine-tuning.

In this paper, we investigate using paraphrase generation for data augmentation for fine-tuning entity tagging models across different domains. We do this specifically for the scenario in which we are training an entity tagging model for a brand new domain. Also in this work, we try to form the training conditions in such a way to be as close to a real world few-shot scenario as possible, which to our knowledge has not been done in such a way in previous work. This involves forming our training data so that each entity type has only a few samples, and the same number of samples per entity type, thereby making no assumption of knowledge of the distribution of entity types. Furthermore, we also assume that there is no separate, labeled development set with which to perform modeling. The paraphrase generation model is trained using PAWS (Zhang et al., 2019), a large general corpus of paraphrase data. We find that up to a certain point, the paraphrases are useful to add to the training data set, but past that point, noisiness in the paraphrases limits their usefulness. In order to better handle the noise, we also experiment with learning an ensemble voting model from the same paraphrase data, which we find to consistently boost model accuracy.

2 Related Work

There have been various methods that have been tried to increase the accuracy of natural language processing models trained with little training data. One avenue of investigation is to rearrange the little training data that there is, for example by swapping words or phrases. Examples of these include (Wei and Zou, 2019) which demonstrates the effectiveness of this approach on various natural language classification tasks, and (Andreas, 2020) which introduces a data augmentation rule that substitutes a phrase with other phrases in every context if they co-occur at least once in some context, and shows its viability for classification and semantic parsing datasets. Another approach is to use back-translation, which has been found to be useful for data augmentation for neural machine translation (Sennrich et al., 2016; Edunov et al., 2018), reading comprehension (Yu et al., 2018), and dialogue summarization (Liu et al., 2022). Back-translation has been found to be helpful for named entity recognition in biomedical domains (Yaseen and Langer, 2021), but so far not for utterances in the dialogue domain (Basu et al., 2022). (Dai and Adel, 2020) experiment with data augmentation by modifying seed utterances using rules many of which involve replacing tokens in utterances with tokens having the same label in other utterances or with tokens from WordNet. Evaluation is performed on biomedical datasets. (Huang et al., 2021), (Ding et al., 2021), and (Das et al., 2021) look at prototype methods for few-shot entity tagging, where each entity type has its own "prototype" representation in embedding space.

Besides these, there have been various other methods for data augmentation that specifically use paraphrase generation. (Jolly et al., 2020) experiment with data augmentation through paraphrase generation for entity tagging, where the paraphrase generation model is trained from data in the target domain. (Okur et al., 2022) experiment with a similar approach, but only apply it for intent classification. For entity tagging, they use a synonym-based approach to automatically label utterances using a generic noun phrase chunker and ConceptNet, which is shown to produce good performance on an in-house dataset.

It can be seen that there have been many approaches that have been studied for data augmentation of entity tagging models. When compared with this literature, we believe in the uniqueness of the scenario in which we apply and evaluate data augmentation, one in which there is a brand new domain, in which there is only few-shot labeled data in the domain, there is no development set, but we can evaluate over a large test set in order to verify the ability of the trained models. For example, (Jolly et al., 2020) experiment with data augmentation but do so for the scenario of adding training data for a new intent type to an established large training data set for a particular domain. Because they do have access to a large training data set, they can and do use it to train a paraphrase generation model in the target domain for data augmentation, which we cannot do in our scenario which is starting with only a small seed training data set. (Basu et al., 2022), (Huang et al., 2021), (Ding et al., 2021), and (Das et al., 2021) evaluate their data augmentation in the episodic learning scenario for which models are trained on sampled few-shot data and tested on a small randomly sampled subset of a test set, rather than the whole large test set. We argue that this does not provide a clear picture of how effective those models would be in actual practice. In contrast, (Dai and Adel, 2020) and (Yaseen and Langer, 2021) do evaluate their entity tagging models on a whole large test set. On the other hand, their few-shot training data consists of random samples from a large training data set, which enables the resulting model to gain knowledge of the underlying distribution of entity label. For example, their few shot models would likely model better those entity types that occur more often because there would be more examples of those entity types in their randomly sampled training data. In addition, they both tune their models on a separate development set, which we believe would be hard to obtain in a real world scenario.

3 Datasets

There are five datasets on which we perform experiments, three that are in-house and two that are public. They are all English datasets. The in-house datasets consist of user utterances from three different customer service applications. Two of them, SDA and SDB, are from spoken dialogue systems, where the utterances have been hand transcribed from audio, and the utterances are responding to a "How may I help you" prompt. The third, SM, is from written social media posts where the users are asking for support for their products. Details of the annotation of these datasets can be found in Appendix B. The public datasets are SNIPS (Coucke et al., 2018) and the English ATIS-2 corpus (Hemphill et al., 1990). A summary of the datasets can be found in Table 1.

The training data and validation data are set up as follows. The training data is set up as 10-shot data, meaning in each training dataset there are 10 examples per entity type. SDA and SDB are 10-way 10-shot data. SM is 4-way 10-shot data. We formulate SNIPS and ATIS-2 training as 8-way 10-shot data. The validation data is set up as a subset of the training data, one quarter of its size, in order to better simulate the condition where there is a lack of labeled data.

We extract a few-shot version of the SNIPS dataset as follows. From the original SNIPS dataset which has 39 label types, we select these eight label types: *city*, *country*, *movie_name*, *object_name*, *playlist*, *service*, *year*. For each label type, we extract 10 utterances having that type from the SNIPS training data, ending up with a training set with 80 utterances. For our test set, we select a subset of utterances from the SNIPS test set having at least one mention of one of our eight target label types until there are 50 instances of each label type, ending up with a test set with 347 utterances.

We extract a few-shot version of the ATIS-2 dataset in a similar fashion. We select these eight label types from the original ATIS-2 dataset which has 79 label types: *airline_code*, *airport_code*, *airport_name*, *city_name*, *fare_basis_code*, *fromloc.airport_name*, *fromloc.city_name*, *restriction_code*. From these, we obtain a 10-shot ATIS-2 training set of 80 utterances and a 50-shot ATIS-2 test set of 331 utterances, as subsets of the original ATIS-2 training and test sets, respectively.

4 Models

We train a paraphrase model by fine-tuning a T5 transformer (Raffel et al., 2020) on the PAWS English training set (Zhang et al., 2019). It is a text to text model having a standard encoder decoder architecture. We employed Huggingface's Transformers library (Wolf et al., 2020) for its implementation.

We run the paraphrase model over each utterance in the training set so that 100 paraphrases are generated. Subsequently, we filter out paraphrases that are duplicates of those generated previously. On average there are 36 to 40 unique paraphrases generated per original utterance.

Because the paraphrase model is fine-tuned on text data only, our input to the model is utterance text only, with no tag information. For the same reason, the paraphrases that the model outputs contains text only. In order to use the paraphrases to train an entity tagger, we perform an extra step of gazetteer tagging the paraphrases where the gazetteers are prepared using the seed training data.

Our entity tagging models are based on fine-tuning BERT (Kenton and Toutanova, 2019). The standard architecture consists of the encoder part of BERT followed by a classification layer with no subsequent CRF layer. Its input consists of WordPiece tokenized text.

Hyperparameter settings for these models can be found in Appendix A.

5 Experiments

We train different entity tagging models for different domains. Baselines are established on training models on the few shot training data alone. Other models are trained on a concatenation of this data along with different amounts of paraphrase training data. Training sets are notated as follows. FS represents few-shot hand-annotated training data. FSx1 represents a training data set containing few-shot training data concatenated with paraphrase training data of the same size. This paraphrase data is generated in the following manner. For each few-shot training data example, a paraphrase is chosen at random from the output of the paraphrase generation model when the few-shot example is input. FSx2 represents a training data set containing few-shot training data concatenated with paraphrase training data that is twice its size. It is generated in a similar fashion as the paraphrase data for FSx1, except tthat wo non-duplicate paraphrases are taken from the output of the paraphrase generation model

Results are shown in Table 2, where rows generally represent different domains and columns represent different training data sets. Each cell in the table shows the labeled bracketed F-measure score of a model trained on a particular training data set in a particular domain, where the score is averaged over 50 train/test runs.

We see that adding some paraphrase data (FSx1) always leads to an increase in performance. Adding even more paraphrase data (FSx2) generally leads to increases or decreases in performance. By examining the paraphrase data, one reason why adding more paraphrase data does not always increase the accuracy of the model appears to be because the paraphrase data is not always of the highest quality. For instance, there are some examples of duplication that are not uncommon in text to text models, such as the output being “John Smith John Smith John Smith” when the input is ”John Smith.”

Comparing results of different domains, we see that certain domains such as SNIPS and ATIS-2 in general achieve lower accuracies than other domains such as SDA and SDB. This may be attributed to differences in breadth of entity types in different domains, with broader types being harder to entity tag. Here, by breadth we mean the number of surface phrases that may receive a particular entity tag. For example, the entity type *movie_name* in SNIPS is broad because there are hundreds of new movie names introduced every year while the entity type *phone* in SDA is not as broad because there are comparatively fewer surface phrases corresponding to different commercially available phone products.

One way to mitigate the effect of noise is to perform ensemble voting across different models. We perform a simple voting procedure where in each

Name of Dataset	Number of Entity Types	n-shot Training	Number of Utterances in Test Set
In-house Spoken Dialogue (SDA)	10	10	1000
In-house Spoken Dialogue (SDB)	10	10	1000
In-house Social Media (SM)	4	10	1036
SNIPS	8	10	347
ATIS-2	8	10	331

Table 1: Characteristics of few-shot datasets.

Domain	FS	FSx1	FSx2
SDA	0.8176	0.8270	0.8467
SDB	0.8390	0.8470	0.8295
SM	0.1760	0.1980	0.1802
SNIPS	0.1780	0.2046	0.2071
ATIS-2	0.3060	0.3340	0.3533
Average	0.4633	0.4821	0.4834

Table 2: Doubling the few-shot training data (FS) with paraphrases (FSx1) leads to an increase in performance (labeled bracketed F measure score of the model on the test set) across domains. Tripling it with even more paraphrases (FSx2) leads to more uneven results in terms of performance.

Domain Name	FSx1	FSx1 +Ens	FSx2	FSx2 +Ens
SDA	0.8270	0.8496	0.8467	0.8588
SDB	0.8470	0.8620	0.8295	0.8346
SM	0.1980	0.2315	0.1802	0.1986
SNIPS	0.2046	0.2262	0.2071	0.2444
ATIS-2	0.3340	0.3532	0.3533	0.3767
Average	0.4821	0.5045	0.4834	0.5026

Table 3: Ensemble voting over 50 models trained with few shot and paraphrase data of certain sizes (FSx1+Ens and FSx2+Ens) consistently improves the accuracy over single models trained on few shot and paraphrase data of the same size (FSx1 and FSx2, respectively).

training data condition, FSx1 or FSx2, we train 50 models and test them on the same test set examples. Each word in each example is then tagged with the label that most of the models assigned to that word. The results are shown in Table 3. They show that ensemble voting always increases the accuracy of the corresponding non-ensembled model.

6 Conclusions

We performed experiments in order to evaluate the effectiveness of using a general purpose paraphrase generation model for data augmentation in a few-shot scenario for entity tagging. We have attempted to fashion these experiments to mirror a real world situation, where there are few examples in the few-shot data, the examples that do exist probably do not reflect accurately the distribution of target entity types, and where there is no development set data. We have performed these experiments in different domains to evaluate the generality of our findings.

We have found that a general purpose paraphrase generation model is generally useful for data augmentation in a few-shot scenario for entity tagging. However, because of noise in paraphrase generation, if more and more paraphrases are being added to the training set, it appears that the performance of the resulting model can level off and eventually decrease. In order to help reduce the effect of this noise, we have experimented with the idea of ensemble voting across models trained on different paraphrases. In our experiments, this strategy always had a positive effect on model performance.

In future work, we would like to experiment with ways to increase the quality of the paraphrase generation model, perhaps by employing few-shot learning methods to it. We are also interested in ways to make the ensemble voting approach more lightweight, such as through the use of distillation.

7 Limitations

We have not tried to compare these methods to other methods, such as back translation or editing of texts. Also, have not tried to combine this method with the other methods to produce more accuracy. Another limitation is that it is hard to use an ensemble model in a production environment because of its high overhead. The approach in the paper generally increases the accuracies of models over baseline, but if the baseline score is very low, the approach in the paper is not powerful enough to increase the accuracy of the model high enough that it would be of use in applications.

References

Jacob Andreas. 2020. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566, Online. Association for Computational Linguistics.

Samyadeep Basu, Amr Sharaf, Karine Ip Kiun Chong, Alex Fischer, Vishal Rohra, Michael Amoake, Hazem El-Hammamy, Ehi Nosakhare, Vijay Ramani, and Benjamin Han. 2022. Strategies to improve few-shot learning for intent classification and slot-filling. In *Proceedings of the Workshop on Structured and Unstructured Knowledge Integration (SUKI)*, pages 17–25.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3861–3867.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca J Passonneau, and Rui Zhang. 2021. Container: Few-shot named entity recognition via contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 6338–6353.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A few-shot named entity recognition dataset. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3198–3213, Online. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. 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*.

Jiaxin Huang, Chunyuan Li, Krishnan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2021. Few-shot named entity recognition: An empirical baseline study. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10408–10423.

Shailza Jolly, Tobias Falke, Caglar Tirkaz, and Daniil Sorokin. 2020. Data-efficient paraphrase generation to bootstrap intent classification and slot labeling for new features in task-oriented dialog systems. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 10–20.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

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 preprint arXiv:1907.11692*.

Yongtai Liu, Joshua Maynez, Gonçalo Simões, and Shashi Narayan. 2022. Data augmentation for low-resource dialogue summarization. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 703–710, Seattle, United States. Association for Computational Linguistics.

Eda Okur, Saurav Sahay, and Lama Nachman. 2022. Data augmentation with paraphrase generation and entity extraction for multimodal dialogue system. *arXiv preprint arXiv:2205.04006*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Usama Yaseen and Stefan Langer. 2021. Data augmentation for low-resource named entity recognition using backtranslation. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 352–358, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLPAI).

Adams Wei Yu, David Dohan, Thang Luong, Rui Zhao, Kai Chen, and Quoc Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *Proceedings of the Sixth International Conference on Learning Representations (ICLR 2018)*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling. 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 1298–1308.

A Hyperparameter Settings

The hyperparameter settings for training the T5 paraphrase generation model are as follows: *model_name*: t5-base, *max_seq_length*: 512, *learning_rate*: 3e-4, *weight_decay*: 0.1, *adam_epsilon*: 1e-8, *warmup_steps*: 0, *train_batch_size*: 6, *eval_batch_size*: 2, *num_train_epochs*: 2, *gradient_accumulation_steps*: 16, *seed*: 42.

The hyperparameter settings for training the entity tagging models are as follows: *learning_rate*: 5e-5 *num_train_epochs*: 50, *train_batch_size*: 32.

B Human Annotation Details

The annotation for SPA and SPB proceeded as follows. For each of SPA and SPB, a large list of entity types was prepared by an application designer. From that list, one of the authors prepared a subset of 10 entity types corresponding to the entity types that most frequently occurred in that application. That person also prepared a list of raw utterances for that domain, and an initial annotation guide. One subset of 40 utterances was sent to a human annotator A along with the annotation guide for annotation. That subset was subsequently doubly annotated by human annotatior A and the author. Based on evaluation of similarities and differences in the doubly annotated data, the annotation guide was revised and a final annotation version of that subset was prepared. This process was repeated another time, after which human annotator A and human annotators B and C annotated the rest of the data for that domain. The author would occasionally spot check the annotations and direct them to be corrected if necessary.

For the annotation for SM, a complete list of entity types was prepared by another application designer. This entire list, four entity types, was chosen for annotation. One of the authors prepared utterances for the SM domain from social media extracted by a running production system. That author also prepared an annotation guide, after which that author alone annotated all of the utterances. While annotating, occasionally the annotation guide would be need to be modified if certain examples were found that did not fit situations handled by the guide. The author annotated twice certain groups of the same utterances, at different times, and compared results which were harmonized if necessary, as one form of quality control. For other groups of utterances, the author visually checked the annotation, but the author did not do this checking for all of the utterances in the corpus.