Low or no resource domain adaptation for task specific semantic parser

Stanford CS224N Custom Project

Sinchan Bhattacharya
Department of Computer Science
Stanford University
sinchanb@stanford.edu

Abstract

Task oriented semantic parsing plays an important role in virtual assistants (like chatbots, voice assistants etc.) to understand the utterances of the speaker by ingesting the text input and applying semantic parser models to decode the meaning of the text. Such semantic parser model needs to be trained on large dataset of individual task, which is impractical since there can be numerous amount of different possible tasks. To overcome this problem there had been several studies which apply high resource source domain and low resource target domain training methodologies to train semantic parsers for specific tasks. In this study we explore what effect does the choice of source domain has on the semantic parser model accuracy for target domain. We show in this study that the choice of source domain for building target domain specific semantic parser is highly critical. We also propose a novel method of building a target domain specific semantic parser model with no target domain data which demonstrates superior prediction accuracy when compared to a zero-shot learner.

1 Key Information to include

- Mentor: Shikhar Murty
- External Collaborators (if you have any): NA
- Sharing project: NA

2 Introduction

Virtual assistants have become a ubiquitously technology, used across industries to improve products and process flows. They are used in cars, cellphones, smart home systems, customer care centers, medical devices, security systems etc. Companies like Amazon[1], Google [2], Apple [3] have developed highly skilled virtual assistants which can provide expected outcomes most of the times. But the technology is still work in progress since, there are a lot of opportunities for improvements. One of the the building blocks for Virtual Assistants are Natural Language Understanding Unit (NLU). Semantic parser is a crucial part of NLU modules and play an important role in understanding the text. A semantic parser captures the semantic information of a given text and translates the text into a formal meaning representation on which a machine can act. Semantic parser for virtual assistants intakes text and then converts it to a form such that the intent of the utterance is understood.
For example, if the utterance is : " Driving directions to the Eagles game" Semantic Parser output : [IN:GET_DIRECTIONS Driving directions to [SL:DESTINATION [IN:GET_EVENT the [SL:NAME_EVENT Eagles] [SL:CAT_EVENT game]]]]
where, IN is Intent and SL is slot. Intent represents an action that fulfills a user's spoken request.Slots are variables that determines certain values for an intent.
Semantic parser needs to be trained on large dataset containing utterances and there corresponding semantically parsed form of text for a given domain. So, to have a robust semantic parser model,

which can be implemented in a virtual assistant, the semantic parser needs to be trained on large dataset from each and every domain where it will have to make predictions. But this is impractical since there could practically be numerous number of domains where the virtual assistant has to accurately operate. to overcome this roadblock domain adaptation techniques are followed where models are trained with large amount of labelled data for specific source domain task and then those models go through a secondary training process but with a much smaller labelled data for a specific target domain task. Although such models get trained on a smaller target domain task, yet they provide fairly acceptable predictions for this target domains.
A fairly unexplored area of domain adaptation methodologies is the effect of source domain selection for training a model for a specific target domain task. The question that we are addressing in this research work is that "What effect does the choice of source domain has on the semantic parser model's prediction accuracy for a specific target domain task?". To answer this question we analyze the domain adaptation technique for semantic parser models for multiple combinations of source-target domain pairs.
While exploring the effect of source domain selection for such domain adaptation techniques, we identified some inherent capabilities of this semantic parser models, and by leveraging those capabilities we propose a novel method of unsupervised domain adaption where with no (or zero) training data for target domain is required and yet the semantic parser could provide improved prediction accuracy.

3 Related Work

A lot of work in the field of domain adaptation had been done both in the field of computer vision [4], [5], [6], [9] and Natural Language Processing (NLP) [7], [8] with a publications supporting this technique more heavier towards computer vision applications. Domain adaptation specifically for semantic parsing has been explored by a group of researchers from Facebook[10] where various low resource domain adaption techniques, like fine-tuning, joint training and meta-learning are explored and demonstrates that meta-learning provides the best prediction accuracy among all techniques. In this paper the BART model [11] (a denoising autoencoder for pre-training sequence-to-sequence model) is used as the baseline model to train on both source domain as well as target domain from the TOPv2 dataset [12] (we will be using the same dataset and baseline model architecture).
There has not been any previous study done to our knowledge where effect of source domain selection on prediction accuracy of target domain is explored for semantic parser models. The research work on low resource domain adaptation[10] trains the target-specific domain model with a minimum of 25 sample per intent-slot (SPIS - samples for a given combination of intent-slot pair) which rounds upto a minimum of 100 samples for each target domain. Whereas, in this paper we are proposing an approach of zero-training of the target domain and yet acheiving acceptable prediction accuracy.

4 Approach

Our main goal for this project is to train a semantic parser model with different combinations of source-target domain pairs with various dataset size for the target domain. The multiple training policies would result into multiple endpoint evaluation (discussed in Section 5.2) for each model training policy.
To give an example, lets assume there are a total of 6 different domains, where 2 domains has sufficient training data. These 2 domain will be used as source domain, since it has sufficient data to train a semantic parser model. Lets call this two domain as S1 and S2.
The remaining 4 domain has smaller training data and can be considered as the target domain for the semantic parser. We can name this target domains as T1, T2, T3 and T4.
Now we can form pairs of source-target domain data in the following manner:
1. (S1, T1), (S1, T2), (S1, T3), and (S1, T4)
2. (S2, T1), (S2, T2), (S2, T3), and (S2, T4)
Hence, we can see that each source domain has been paired with a target domain once.
Now the semantic parser model will train on the Source domain first with the larger source domain dataset and then the trained model would re-train on the target domain dataset, which is comparatively smaller. Since, training and re-training could possibly cause confusion, for the sake of better naming we will call the re-training phase as fine-tuning. After the model is fine-tuned on the target dataset, the model is used for making prediction on utterances on a test dataset comprising of the target domain

utterances.
As there are a total of 8 different combinations of source-target dataset, hence there would be 8 different fine-tuned models, each trained on a specific source domain and fine-tuned on a specific target domain. Thus, there would be 8 different evaluation metric, coming from each source-target domain pair models, which would be analysed on Section 6.
As mentioned above, the semantic parser model should be able to intake a natural language utterance and convert it into semantically parsed format. So, we can consider semantic parsing as a Machine Translation task. The source domain data that we have, although large compared to the target domain data, yet it is not quite large to train a language model from scratch. Hence, we chose the baseline model to a pre-trained language model, as we know that pre-trained language models have shown to quickly generalize and provide good predictions in low resource setting [13]. We do not use models like BERT[13] or RoBERTa[14] as they have pre-trained encoder only and is not suitable for a compositional parser task. We choose to use the BART model, which is a seq2seq architecture with both pre-trained encoder and decoder. The pre-trained encoder and decoder of the BART model is used to initialize our semantic parser model [10]. Details of the model used and its parameters are discussed in Section 5.3.
We discuss the novel method of domain adaptation with zero target domain training data in details in Section 5.3 as well.

5 Experiments

In the following section the experiments and the data on which the experiment are done are explained in details

5.1 Data

The dataset used for this research work is TOPv2 [12]. This dataset contains utterances from 6 different domains : *alarm, navigation, event, messaging, music, and timer*. Each domain data is already split into train, validation and test dataset (details of the dataset can be found in Appendix Table1).
The two domains considered as source (S1 and S2) from this dataset are *alarm and navigation*, since this two domains contain the largest amount of training data. The domains considered as target (T1, T2, T3, and T4) are *event, messaging, music and timer*. The dataset are paired together to form 8 different pairs of source-target domain data:
1. (alarm, event), (alarm, messaging), (alarm, music), (alarm, timer)
2. (navigation, event), (navigation, messaging), (navigation, music), (navigation, timer)
These 8 different domain pairs is trained and fine-tuned and then evaluated with certain metrics (explained in Section 5.2) and then the metrics are compared in the Analysis section.
As mentioned above, each of the domain specific dataset was already split into train, validation and test dataset. But for the purpose of our problem we modified the train, validation and test dataset for the target domains. The dataset for source domains were kept as is :
alarm : Train samples = 20430, validation sample = 2935.
navigation : Train samples = 20998, validation sample = 2971.
Test dataset for the source domains were not required since we are not interested in the prediction accuracy of the semantic parser for the source domains.
For the target domain, since we are interested to know how the semantic parser model fine-tunes when provided with different amount of low-resourced target domain data, hence we create several snippets of training dataset for each of the target domain. We wanted to investigate what happens when we provide the fine-tuning model with a very small dataset and then keep on increasing the data sample to a finite number for every fine-tuning mode. The different sample sizes for each snippets of data used were 25, 50 and 100 where the samples are chosen through purely random manner and stored as our new training dataset. For each of the fine-tuning mode we need a validation dataset which needs to have a lower number of samples than the training data. We generated one validation dataset with 20 samples, again generated randomly, for every target domain dataset.
For testing the fine-tuned models on there domain adaptability we need a test dataset for each of the target domain. Since, BART is quite a large model and training and fine-tuning of the model takes quite a good amount of time, we considered reducing the test dataset to a size of 500 samples each, so that we could save some time during the model testing process. The 500 samples for the test dataset

are selected randomly from the large test dataset available. To give an example of the sample sizes selected for any given target domain:
Three sets of training data selected randomly from the over-all training dataset of target data: 100 samples, 50 samples and 25 samples. One validation dataset with 20 samples, generated by random sample selection from the over-all validation dataset. One test dataset with 500 samples, where the samples are selected randomly from the over-all test dataset.
A question may arise as to why the dataset are generated randomly which may cause an imbalance of intent-slot pair in the dataset which can ultimately add bias to the model training and evaluation. But for this research we are not trying to improve the accuracy of predictions for domain adaptable semantic parser model, rather this is a exploratory study to compare how the model training and evaluation differs when the source domains are swapped and small dataset for target domains are used. Even if the model training and evaluation contains bias because of random sample selection for training, validation and test data, yet it does not affect the analysis for this study as we are comparing prediction accuracy and other endpoint metrics for each fine-tuning mode. As the results are compared for each source-target domain pair and different fine-tuning modes hence the effect of bias in model training and evaluation (which, if present, will exist in all fine-tuning modes or during evaluation) gets cancelled off.
Each of the dataset contains three columns : domain, utterance and semantic parse.
The domain column signifies to which domain the particular utterance belongs to. Utterance is the natural language text that a user would speak when providing any command or request. Semantic parse is the converted form of the uttered text into a semantically parsed format which a computer program would understand. Each of the semantic parsing contains intents, represented as IN and slots, represented as SL. To visualize a semantic parsed representation see Appendix (Fig. 1)
The semantic parser model is fed with utterance data and it needs to predict the semantically parsed format of the text. Hence, the models will have the *utterance* column as input and *semantic parse* column as target output. All the dataset, namely train, validation and test dataset, is pre-processed. the pre-processed data is used for training, validation and evaluation of the models. The pre-processing steps followed is as below:
The *utterance* column (input data to the models) is kept as is. The *semantic parse* column (target output data) is pre-processed by first removing any white spaces at the beginning and end of the text. Then the intents of the utterance is placed at the beginning, followed by all the slots present in the utterance placed side by side, spaced with white spaces and they are sorted in the same manner as they appear in the text utterance. Then any utterance text following the intent is removed as it is not useful for the machine to perform any action, then the third brackets enclosing the over-all semantic parser are removed. For example
utterance: Alarm for 8:30 am tomorrow, one time
semantic parse: [IN:CREATE_ALARM Alarm [SL:DATE_TIME for 8 : 30 am tomorrow] , [SL:PERIOD one time]]
processed semantic parser: IN:CREATE_ALARM[SL:DATE_TIME for 8 : 30 am tomorrow] [SL:PERIOD one time]
The processed semantic parser and utterance is then tokenized using the 'facebook/BART-base' tokenizer, since we are using the 'facebook/BART-base' BART model for training and fine-tuning (details about BART model is discussed in Section 5.3). The tokenized processed semantic parser is used as target output and the tokenized utterance is used as input to the model training and evaluation process.

5.2 Evaluation method

The paper for low-resource domain adaptation [10] using TOPv2 data uses exact match as the metric for evaluating the model outcome. Exact match is the string comparison between the target and predicted string and it true only if the all the characters and its positions of the predicted string exactly matches with the target. We incorporated the same evaluation metric to evaluate our model.
But we see discrepancies in using the exact match metric only for model evaluation. There are cases where the semantic parser provides a good prediction of semantically parsed text which almost matches with the target text but with extra white spaces in between words or unnecessary words. For example, if take the above sample:
target semantic parser: IN:CREATE_ALARM[SL:DATE_TIME for 8 : 30 am tomorrow][SL:PERIOD one time] *predicted semantic parser*: IN:CREATE_ALARM[SL:DATE_TIME for 8 : 30am tomorrow][SL:PERIOD one time]

We see that the prediction quite good for the purpose of language understanding but it is still not an exact match because there is no space between '30' and 'am' for the predicted text whereas the observed text has a space in between. This discrepancy can be over come by removing all white spaces from target and predicted when calculating the exact match metric, but in cases where there are unnecessary words (like articles) present in the prediction or inflected forms of words present in the predicted text, the exact match metric fails. Again this can be over-come by stemming and lemmatization and further processing of the target and predicted text before metric calculation but this makes the processing step complicated. Hence, we came up with simpler metrics that can capture the predictability of the model from the essence of language understanding through semantic parsing. The new metric compares the intents present in the target semantic sentence with the intents present in the predicted semantically parsed sentence, we call this metric as intent_matching,
intent_matching = *if(*All intents in predicted == All intents in target*)* then 1, else 0
intent_matching is calculated as percentage over all the test sample (500 for our research). The predicted semantic parsed text will be ultimately fed to the next module which can take actions according to the intent and slots present in the utterance, hence, getting a predicted intent for a given domain comparable to the human annotated intents is essential for a good semantic parser. The same process can be done for matching slots, which we did not measure for this study for the sake of time. We also diagnosed different BLEU score of the model but concluded that only lower n-gram BLEU score makes sense, as we are using quite a small dataset for fine-tuning the models. So, we are recording 2 different BLEU scores: BLEU 2-gram with weight distribution of (0.95,0.05) and BLEU 2-gram with weight distribution of (0.5,0.5) and we will refer them as BLEU1 and BLEU2 henceforth.

5.3 Experimental details

As mentioned in the Approach section, we chose to use a pre-trained seq2seq model as our baseline model. The model we chose was BART, specifically **facebook/bart-base**. Hence, the choice of tokenizer was also facebook/bart-base tokenizer. The facebook/bart-base has a total 139 million trainable parameters. We froze the encoder part of the BART model and only trained the weights of the decoder part. During training process both the utterance (input) and semantic parsed text (target) is truncated or padded to a length of 32 characters (average sentence length, refer Table 1 in Appendix). The padding is is done with token from facebook/bart-base and a prefix is also added to both the input and target sentence.
The optimizer selected for finding optimal weights for the decoder is Adam with a learning rate=2e-5. There are two parts in the training process, firstly the pre-trained model is trained on the larger source domain dataset and we will call this as training step, then we fine-tune the trained model from the previous step by training it on a smaller target domain dataset and we call this step as fine-tuning step. During the training step the number of epochs = 5 and during the fine-tuning step the number of epochs = 1, the epoch are selected such that the over-all training time is lowered. The batch_size for both the above steps is 16. Over-training of the model addressed by applying Early Stopping method with the validation data. The model training can be represented by Figure 2:

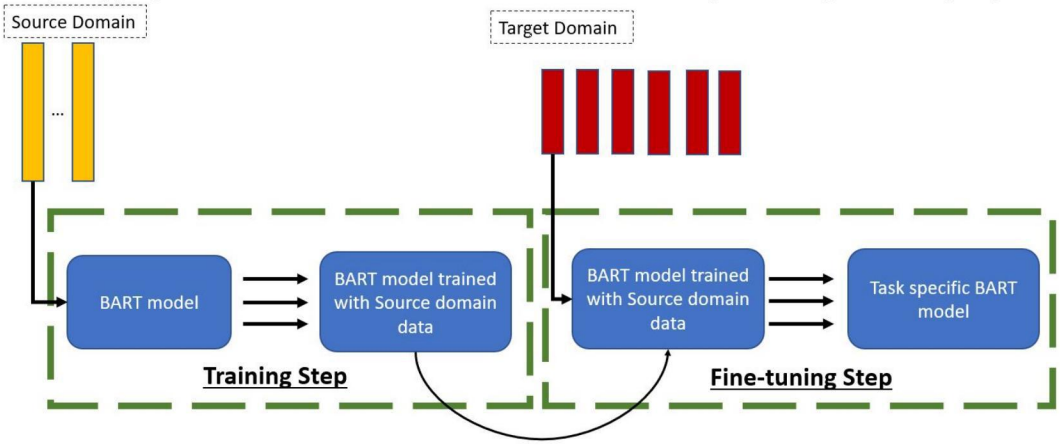

Fig 2. Domain adaptive low-resource fine-tuning model training architecture

The above process of domain adaptation training is repeated for a total of 24 times. There are a total of 8 pairs of source-target pair data and there are a total of 3 different sizes of training data, hence the

total number of times the experiment was run is, 8 x 3 = 24 times. At each model training process the pre-trained BART model is trained on the source data and then on the target domain data. As there are three different sizes (100, 50 and 25 sample) of the training data hence, for each source-target pair domain data the over-all model building is done three times dis-jointly, i.e. the source domain training is done once and then the trained model is fine-tuned with three different data sizes of the same target domain data in parallel. This whole process is repeated 8 times.
After each complete model building process the final fine-tuned model is tested on the corresponding target domain test dataset for which the model was fine-tuned. The results are recorded and analyzed in the later section.

5.4 Novel approach - Zero training on target domain

While performing domain adaptation techniques, we wanted to explore an idea - What would happen if we do not fine-tune the model on the target domain yet use the source domain trained model to do inferences on a target domain? To give an example, we trained the BART model on *alarm* source domain dataset. Now, without doing any fine-tuning of the model, we apply the model on the *messaging* test dataset for inferencing. We see the model fails to predict a single exact match or even the intent for any of the test samples and this was expected as the model has never encountered these intents.But what was surprising was that the source trained BART model has inherent property of performing semantic parsing, irrespective of the target domain. So although the intents and slots predicted by this model are wrong yet the semantic parsing of the sentence seems correct. To give an example:
utterance: don't read that text
target semantic parse: IN:IGNORE_MESSAGE don't read that text
model predicted semantic parse: [N:DELETE_ALARM don't read that text
or utterance: Can you pass this pic along?
target semantic parse: [IN:SEND_MESSAGE Can you pass this [SL:TYPE_CONTENT pic] along ?]
model predicted semantic parse: IN:UPDATE_ALARM Can you pass this [SL:ALARM_NAME pic] along ?
So, we see that the zero target domain trained model can still semantically parse the text due to its inherent quality of semantic parsing that it learned during source domain training.
Next what we did is to logically replace the source domain intents on the predictions with expected target domain intents. For example GET_ALARM is replaced with GET_MESSAGE, SNOOZE_ALARM is replaced with IGNORE_MESSAGE intent and then when we test our source trained model on the *message* target domain data, we see the intent matching percentage increased from 0% to 67.6% which opened up the opportunity for zero learning target domain adaptation method. But we were not satisfied with this method, since in this method we are applying supervised manual translation of source intent to domain intent and in the realm of machine learning we explore ways of automating processes intelligently. Hence, we further explored methods of translating source intent to target intent without any manual intervention.
For that purpose we use the glove-wiki-gigaword-200 vocabulary to convert the source and target intent words to vector. The word before the underscore represent the intent for all the intent token present in the data. So, we split the intents with underscore and only retain the word before underscore, i.e. we save CREATE from CREATE_ALARM and DELETE from DELETE_ALARM. We make a list of intent words from the predicted semantic parsed sentences (which contains source domain intents) and call it as source intent word list. We create another list of possible target domain intent word and call it as target intent word list. Then we convert all the words in both the list to vectors using the glove-wiki-gigaword-200.Next we perform a nearest neighbor (calculating the nearest vector through Euclidean distance measure) to find the closest matching word between the two list such that the source intent words can be mapped to target intent words. For e.g.:
BART trained on *alarm* data and inferred on *messaging* data:
The nearest neighbor of source intent -> target intent
SNOOZE -> IGNORE, SILENCE -> IGNORE, UNSUPPORTED -> IGNORE
UPDATE -> SEND, DELETE -> IGNORE, GET -> GET
CREATE -> GET
Once the intent is translated is through nearest neighbour technique we get the following result:

Trained on source (alarm) inferred on target (messaging)			
	Source Trained model as is	Manual intent translation	Novel Nearest neighbor source-target intent translation
Intent match%	0%	67%	20.4%
Trained on source (alarm) inferred on target (music)			
	Source Trained model as is	Manual intent translation	Novel Nearest neighbor source-target intent translation
Intent match%	0%	50.8%	37.2%

Table 2- Source trained model with no target training and evaluated on target test (500 sample) dataset

The above table shows promising result for the novel approach of zero training on target domain and yet able to predict a percentage of utterances. The assumption to apply such a model in real-world scenario would be to have a model that can determine which domain an utterance belongs to, which can be easily achieved through a text classifier (see Appendix).

5.5 Results

The evaluation metrics- exact match%, BLEU1, BLEU2 and intent match% from all 24 different model training scheme is demonstrated below.

Target Domain	event		messaging	
Source Domain	alarm	navigation	alarm	navigation
Exact match %	3%	8.8%	0%	0
Intent match %	84.21%	88%	34%	4.6%
BLEU1	2.18e-18	3.21e-18	~0	~0
BLEU2	9.43e-156	11.54e-156	~0	~0
Target Domain	**music**		**timer**	
Source Domain	alarm	navigation	alarm	navigation
Exact match %	0%	0%	0.2%	0%
Intent match %	6.8%	4.8%	40.2%	0.4%
BLEU1	~0	~0	1.13e-18	~0
BLEU2	~0	~0	6.67e-156	~0

100 target samples

Table 3-Evaluation of source-target domain adaptation through fine-tuning – 100 sample test data

Target Domain	event		messaging	
Source Domain	alarm	navigation	alarm	navigation
Exact match %	0.8%	7.8%	0%	0%
Intent match %	20.8%	53%	0%	0%
BLEU1	~0	~0	~0	~0
BLEU2	~0	~0	~0	~0
Target Domain	**music**		**timer**	
Source Domain	alarm	navigation	alarm	navigation
Exact match %	0%	0%	0%	0%
Intent match %	0%	0%	2.8%	0%
BLEU1	~0	~0	~0	~0
BLEU2	~0	~0	~0	~0

50 target samples

Table 4-Evaluation of source-target domain adaptation through fine-tuning – 50 sample test data

The results table with 25 target training sample is not included as we see that the over-all performance is quite low for 50 sample target domain training, further reduction of the sample size provides results that are not comparable.The analysis of the results are discussed in the next Section

6 Analysis

From Table 3 and 4 we see how the predictability of the models changes depending on what the source domain and target domain were. On Table 4 we see most of the metrics to be nearly zero and that is because the sample size for target domain training is quite low. If we keep our focus on Table 3, for the first example of target domain, event, we see there is a significant increase in prediction of semantic parsed sentence and also the intent prediction for the utterances when the model is trained on navigation source domain instead of alarm source domain. Similarly, for the case of timer target domain training the exact match prediction and intent match prediction is quite high when the model is trained on navigation source domain as compared to alarm source. The analysis results are as expected as we can intuitively think that utterances related to timer is expected to be somewhat similar in nature as utterances for alarm but quite dissimilar to utterances for navigation. Likewise utterances for event would be more similar to navigation utterances, like "take me to an event", than alarm utterances. We went a step further by comparing cosine similarities between the utterances of source domain with utterances of target domain. We combine all the utterances in the over-all training dataset to form a large string and then perform cosine similarity test between these large strings.
The cosine similarity we observe between alarm utterances and timer utterances is 0.4287.
The cosine similarity we observe between navigation utterances and timer utterances is 0.2227.
This shows that domains with similar utterance should be paired for domain adaptation training to provide improved prediction success.
Similarly testing the cosine similarity between alarm and event utterances we get the cosine similarity as 0.2058. The cosine similarity between navigation utterance and event utterance is 0.3701.
This further proves that similarity between utterances between domains play an important role in the outcome of the low resource domain outcome technique. To get better prediction accuracy for a specific target domain the source domain for training needs to be selected carefully. The source domain utterances which has the highest cosine similarity with the target domain utterances must be selected to achieve maximum prediction accuracy.

7 Conclusion

We performed an exploratory study of understanding the effect of selection of source domain has on the prediction accuracy of a low resource target domain trained semantic parser. We observe that selecting the correct source domain for a given target domain which needs to be predicted is very critical as the final exact match and intent accuracy of the model is highly dependant on what source domain was used during the training phase. We also propose a method of choosing the correct source domain for a given target domain to achieve optimal prediction accuracy in the target domain. The method proposed is calculating the cosine similarity between the utterances of source and target domain and choosing the the source domain for training which has the maximum utterance cosine similarity with the target.
We also proposed a novel approach of domain adaptation with no target domain training which showed significantly high prediction accuracy.
In this study we considered the training-fine-tuning process of domain adaptation. Meta-learning is another process of domain adaptation which has shown promising results. It will be interesting to perform the same study on meta-learning technique and analyze if the above observations still holds true.
Since time was limited for data processing, model building and training, we chose smaller epochs, smaller fine-tuning data, random selection of samples for testing and fine-tuning instead of a supervised selection of samples. Moreover, we evaluated our hypothesis of effect of source domain selection using the BART seq2seq model, but this needs to be tested on other models as well. In this research we considered semantic parsing as the task to base all our studies, but this research can be scaled to much wider application of Natural Language Processing
The observations from this research work demonstrates the potential of significantly improving domain adaptation techniques where there is low or no resources available for the target domain.

References

[1] https://www.amazon.com/gp/help/customer/display.html?nodeId=GVP69FUJ48X9DK8V. Retrieved 27 February 2020.
[2] "Doing more to protect your privacy with the Assistant". Google. 23 September 2019. Retrieved 27 February 2020.
[3] "Improving Siri's privacy protections". Apple Newsroom. Retrieved 27 February 2020.
[4] Sebastian Schrom, Stephan Hasler, & Jürgen Adamy. (2020). Improved Multi-Source Domain Adaptation by Preservation of Factors.
[5] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In Proceedings of the IEEE International Conference on Computer Vision, pages 4068–4076, 2015.
[6] Mei Wang, & Weihong Deng (2018). Deep visual domain adaptation: A survey. Neurocomputing, 312, 135-153.
[7] Zhao, H., Zhang, S., Wu, G., Moura, J., Costeira, J., & Gordon, G. (2018). Adversarial Multiple Source Domain Adaptation. In Advances in Neural Information Processing Systems. Curran Associates, Inc..
[8] Bhatt, H., Sinha, M., & Roy, S. (2016). Cross-domain Text Classification with Multiple Domains and Disparate Label Sets. (pp. 1641-1650).
[9] Sicheng Zhao, Guangzhi Wang, Shanghang Zhang, Yang Gu, Yaxian Li, Zhichao Song, Pengfei Xu, Runbo Hu, Hua Chai, & Kurt Keutzer. (2020). Multi-source Distilling Domain Adaptation.
[10] Xilun Chen, Asish Ghoshal, Yashar Mehdad, Luke Zettlemoyer, & Sonal Gupta. (2020). Low-Resource Domain Adaptation for Compositional Task-Oriented Semantic Parsing.
[11] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, & Luke Zettlemoyer. (2019). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.
[12] Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2787–2792, Brussels, Belgium. Association for Computational Linguistics.
[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
[14] 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. CoRR, abs/1907.11692.

A Appendix (optional)

Domain	#Train	#Validation	#Test	Max # words in an utterance
alarm	20430	2935	7123	23
event	9170	1336	2654	51
messaging	10018	1536	3048	35
music	11563	1573	4184	22
navigation	20998	2971	6075	41
reminder	17840	2526	5767	90
timer	11524	1616	4252	26
weather	23054	2667	5682	21
Total	**125k**	**17k**	**39k**	

Table 1:Each observation in the cell contains intent, utterance, and semantically parsed utterance.

utterance : " Driving directions to the Eagles game"
Semantic Parser output : [IN:GET_DIRECTIONS Driving directions to [SL:DESTINATION [IN:GET_EVENT the [SL:NAME_EVENT Eagles] [SL:CAT_EVENT game]]]]

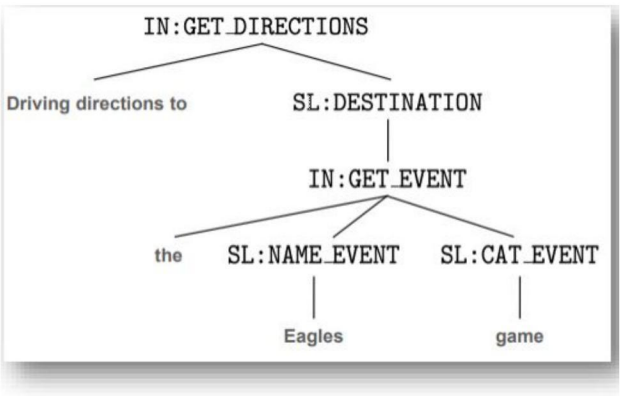

Figure 1: Semantic Parse architecture of the utterance

Target Domain	event	
Source Domain	**alarm**	**navigation**
Exact match %	0%	3.6%
Intent match %	0%	33.6%
BLEU1	~0	~0
BLEU2	~0	~0

25 target samples

Table 4-Evaluation of source-target domain adaptation through fine-tuning – 25 sample test data