

CST5: Data Augmentation for Code-Switched Semantic Parsing

Anmol Agarwal*, Jigar Gupta, Rahul Goel, Shyam Upadhyay
Pankaj Joshi, Rengarajan Aravamudhan

Google Assistant

{okanmol, jigargupta, goelrahul, shyamupa}@google.com

Abstract

Extending semantic parsers to code-switched input has been a challenging problem, primarily due to a lack of supervised training data. In this work, we introduce CST5, a new data augmentation technique that fine-tunes a T5 model using a small seed set (≈ 100 utterances) to generate code-switched utterances from English utterances. We show that CST5 generates high quality code-switched data, both intrinsically (per human evaluation) and extrinsically by comparing baseline models which are trained without data augmentation to models which are trained with augmented data. Empirically we observe that using CST5, one can achieve the same semantic parsing performance by using up to 20x less labeled data. To aid further research in this area, we are also releasing (a) Hinglish-TOP, the largest human annotated code-switched semantic parsing dataset to date, containing 10k human annotated Hindi-English (Hinglish) code-switched utterances, and (b) Over 170K CST5 generated code-switched utterances from the TOPv2 dataset. Human evaluation shows that both the human annotated data as well as the CST5 generated data is of good quality.

1 Introduction

Code-switching (CS) occurs when a speaker alternates between two or more languages within a single conversation. With the increasing ubiquity of voice-based assistants (e.g., Google Assistant, Alexa), the ability to parse code-switched utterances has become one of the key challenges towards building multilingual conversational agents.

Unfortunately, majority of semantic parsing datasets are in English, with very few datasets exhibiting code-switching. While data collection efforts (Einolghozati et al., 2021; Mehnaz et al.,

Original English Sentence	CST5 Generated Code Switched Utterance
Create a new alarm for 9AM on monday 18th June	Monday 18th June ko subah 9 bajhe ke liye ek naye alarm ko create karen
Remove Jim from my reminder to party next wednesday	Agle wednesday ko party ke liye Jim ko mere reminder se hata den

Table 1: A few examples of code switched data generated by CST5. Even with 100 seed examples, the model generates good quality data across multiple domains. English and Hindi tokens are color coded.

2021) have been made to bridge this gap, collecting such datasets requires time-consuming and expensive human annotations from raters who are proficient in multiple languages, making it difficult to obtain large datasets. Another line of work generates artificial code-switched data, either using parallel sentences in the two languages as supervision (Winata et al., 2019) or learning a generative language model from large code-switched corpora (Chang et al., 2019). However, these approaches assume the availability of large corpora ($>50k$ sentences), either parallel, monolingual or code-switched in the languages of interest, which may not be available for the domains of interest.

In this paper we propose CST5, a strategy to obtain code-switched labeled data by generating code switched utterances from a English semantic parsing dataset, using only a small (≈ 100 examples) seed set of (English, Code-switched) example pairs to learn the transformation. This allows us to effectively leverage existing semantic parsing datasets in English to derive supervision for code-switched semantic parsing. Using CST5, we achieve an average 25% improvement on the exact match (EM) accuracy, equivalent to having a order of magnitude more training data. A few examples of the CS utterances produced by CST5 are shown in Table 1.

Our work is inspired by the growing popularity

This work was done during an internship at Google

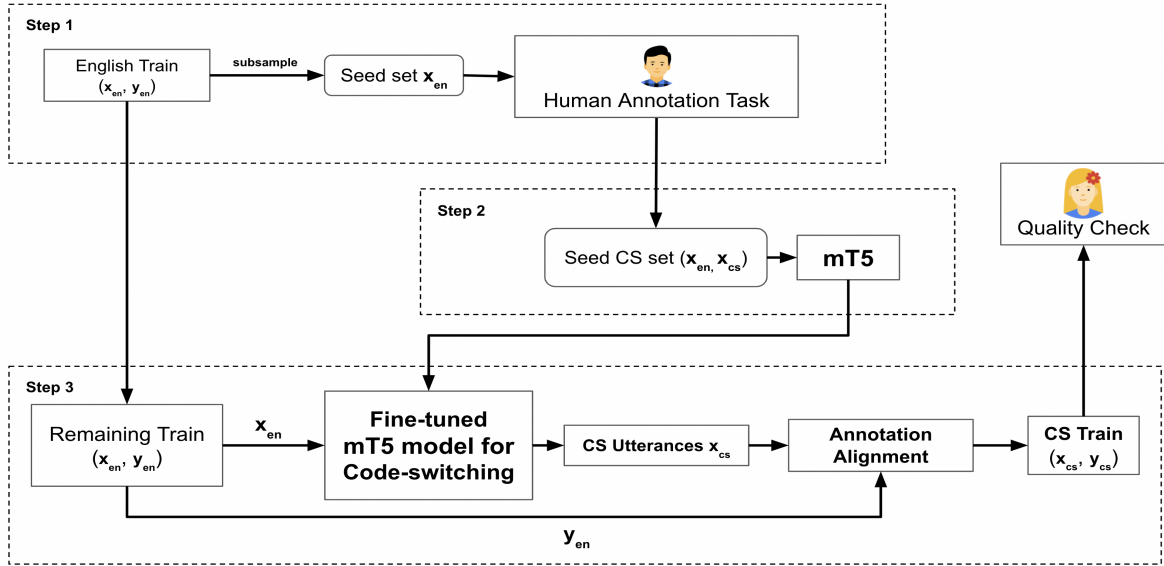


Figure 1: An overview of our approach. We first use human annotators to annotate utterances. We use these to fine tune a mT5 model for code switching and then use the rest of the monolingual corpus to generate more code switched data.

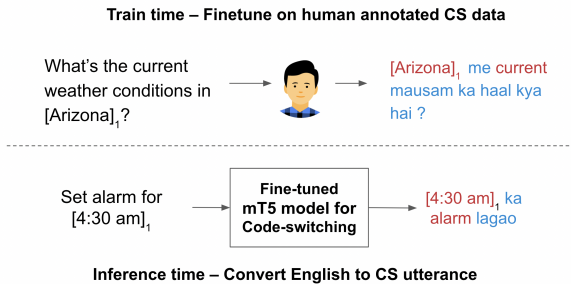


Figure 2: We finetune mT5 on parallel human-annotated data and do inference using the finetuned mT5 model using the monolingual data

of using large pre-trained language models (LM) (such as BERT (Devlin et al., 2018), T5 (Raffel et al., 2019) etc.) as a means of data augmentation (Kumar et al., 2019; Anaby-Tavor et al., 2020; Lee et al., 2021, inter alia). Particularly, CST5 relies on mT5 (Xue et al., 2020), a large pre-trained multilingual LM, which is fine-tuned to perform code-switching using the small seed set, and then applied on a large monolingual semantic parsing dataset in English to generate code switched utterances. By mapping the labeled semantic parse for the English utterance, one can recover the semantic parse to the code-switched utterance, thus obtaining a labeled code-switched semantic parsing dataset. We observe that the mT5 model is well suited to the task of code-switching and the generated code-switched utterances were deemed of high

quality upon manual inspection by human raters, judged to be over 89% semantically equivalent to the original English utterance, and 98% natural.

We use CST5 to generate synthetic code switched data and measure the effect of data augmentation on parser performance with varying seed sizes and across various semantic parsing domains. We observe a 20x data reduction for code switching when using the XXL model. Even with a 100 seed examples, our approach matches the performance of a model trained with 2000 examples for the XXL mT5 parser. To summarize, in this work:

- We introduce a novel data augmentation technique to generate synthetic code switched data for semantic parsing. Using a small seed set of parallel data, the technique converts monolingual data to code switched data.
- We did additional human evaluation of the generated data showing high data quality.
- We conducted experiments on TOPv2 as well as CSTOP(Einolghozati et al., 2021) showing significant gains in EM accuracy.
- We release a code-switched task-oriented semantic parsing dataset, containing over 10k Hindi + English human annotated CS utterances (Hinglish-TOP) along with over 170K synthetically generated CS utterances and their corresponding semantic parse.

2 Related Work

Our work is at the intersection of two areas in NLP: data augmentation for code switching and conversational semantic parsing.

2.1 Data Augmentation for Code Switching

Data augmentation is seen as a flexible and model independent tool for performance improvement by synthetically generating training data for the task of interest (Jia and Liang, 2016; Andreas, 2019; Akyürek et al., 2020, *inter alia*). We discuss some approaches for synthetically generating code switched data below.

Winata et al. (2019) generated code switched utterances using parallel sentences in the two languages as supervision. Chang et al. (2019) proposed an unsupervised method to generate code-switching sentences from monolingual sentences using GANs. Linguistically motivated data augmentation has also been explored in Lee et al. (2019) and Pratapa et al. (2018). Such techniques require expert knowledge or large amounts of parallel data which might be hard to get in a new domain. Instead our work uses small seed sets to generate synthetic data which is more scalable.

Language Model based Data Augmentation

Kumar et al. (2021) used pre-trained transformer models, like BART (Lewis et al., 2020) and GPT-2 (Radford et al., 2019), for general data augmentation. Our work leverages large pretrained language models (LMs) (Raffel et al., 2019; Brown et al., 2020) to generate new synthetic examples. Our approach draws inspiration from recent models like ex2 (Lee et al., 2021), mT5 (Xue et al., 2020) and is most similar to the translate and fill approach suggested by Nicosia et al. (2021). In our work instead of fine tuning the LM to generate translations, we fine tune it to generate CS utterances.

2.2 Conversational Semantic Parsing

Today most commercial (Lialin et al., 2020) conversational semantic parsing systems utilize hierarchical representations, such as the TOP representation (Gupta et al., 2018), which is typically modelled using sequence to sequence task (Rongali et al., 2020; Shaw et al., 2020). Our semantic parsing model is most similar to Cole et al. (2021) where we use a T5 model as the base semantic parser.

Code-Switched Semantic Parsing Traditionally, CS has been explored in word-level language

identification (Molina et al., 2019) and named entity recognition (Aguilar et al., 2019) and shallow parsing (Sharma et al., 2016; Bhat et al., 2018). Recent works like Duong et al. (2017) introduced a CS test set for semantic parsing, Samanta et al. (2019) introduce a variational auto-encoder based generation technique, some other works take inspiration from machine translation (Tarunesh et al., 2021) and adversarial networks (Chandu and Black, 2020). These works rely on specialized architectures and only produce code switched sentences, whereas our work simply relies on a LM finetuning and outputs both sentences and semantic parses.

More recently, Einolghozati et al. (2021) released a Spanglish semantic parsing dataset named CSTOP and Mehnaz et al. (2021) released a Hinglish conversation summarization dataset named GupShup. Similar to the previous works, we release a TOPv2 derived code-switched dataset TOP-Hinglish. We also release the synthetic augmented dataset which is an order of magnitude larger than any such previous dataset.

3 The Code Switched Dataset

3.1 Collecting Hinglish-TOP

For the Hinglish-TOP data collection, we randomly sampled a set of 10,896 utterances from the TOPv2 dataset (Chen et al., 2020). These utterances are distributed across 8 domains (as present in the original TOPv2 dataset). We used a skewed train, validation and test set distribution to focus on getting a bigger test set. Following this we sampled around 2993, 1390 and 6513 utterances from the train, validation and test splits of the original TOPv2 dataset. We observed that there were overlaps across the three splits in the original TOPv2 dataset which we also preserved as is. A domain-wise breakdown of the annotated CS data and a few examples can be found in Table 2.

Our data collection process can be broken down into the steps described below. For these steps we preserved the domain and split information for the sampled utterances. Our approach is also detailed in Figure 1.

3.1.1 Pre-Processing

As a pre-processing step, the slots and intents present in the semantic parse of the English utterance were marked within the utterance using special span identifiers (span-ID), which aligns their position in the semantic tree. By marking these

Domain	Pre-processed English Utterance (Input)	Code-Switched Utterance (Output)
Navigation (14.5%)	What’s the traffic like on [Long Island] ₁ going to [the Hamptons] ₂ [tonight] ₃	[Aaj raat] ₃ [Hamptons] ₂ jaate hue [Long Island] ₁ par traffic kaisa hoga
Weather (16.6%)	Whats the current weather conditions in [Arizona] ₂ ?	[Arizona] ₂ me current mausam ka haal kya hai ?
Reminder (14.1%)	Remind [me] ₁ to [send out the company meeting notes] ₂ [at 5 pm today] ₃ .	[Muje] ₁ aaj [shaam 5 baje] ₃ [company meeting note ko send] ₂ karne ka remind karaye
Alarm (18.2%)	Set me an alarm [every Thursday at 5AM] ₁ [until the 1st July] ₂	Muje [1 july tak] ₂ ke liye [har thursday ko subah 5 baje] ₁ ka alarm set kare
Music (10.8%)	Can I listen to the 5th [song] ₁ on the [album] ₂ one more time ?	kya mai [album] ₂ par 5th [song] ₁ ek bar fir sun sakta hoon ?
Timer (10.9%)	I want you to change the [timer] ₁ to a [30 minutes] ₂ instead of [15 minutes] ₃ .	Mai chahta hu ki aap [timer] ₁ ko [15 minutes] ₃ ke bajaye [30 minutes] ₂ me badal den.
Event (6.9%)	When does [cider tasting] ₁ season start in [Woodinville, Washington] ₂	[Woodinville , Washington] ₁ me [cider tasting] ₁ season kab shuru hoga
Messaging (8.0%)	Send a message to [Diana] ₁ and [Rich] ₂ that [I will be earlier than expected to the library] ₃ .	[Diana] ₁ aur [Rich] ₂ ko message bhejo ki [mai library time se pehle pahunch jaunga] ₃

Table 2: Examples of CS utterances obtained from annotators for the 8 domains in the TOPv2 dataset. The relative sizes of each domain is in parenthesis. The CS utterance is aligned with the English utterance’s parse to identify the CS utterance’s semantic parse (Section 3.1.2). Note: The brackets also demonstrate the semantic parse for the utterance.

spans we know what words in the sentence correspond to the leaf arguments and ask the raters to mark the same in the code-switched query. Some examples of pre-processed English utterances can be seen in Table 2, 2nd column. The pre-processing step allows us to reconstruct the semantic parse for the code-switched utterance, described next.

3.1.2 Code-switching and Alignment

Native Hindi-English speakers were chosen as human annotators for code-switching the English utterances. We asked the the annotators to generate a hinglish code switched utterance for the given sentence while preserving the naturalness and semantic equivalence of the code-switched utterance. This task was done by 3 annotators. A primary annotator whose task is to code-switch the English utterance. Additionally, we had one more annotator who checked the utterances for naturalness and semantic equivalence. We keep the utterance when both the checker agreed on the naturalness and semantic equivalence. The definition of naturalness and semantic equivalence is same as in Section 3.3. Some examples of code-switched utterances after the data annotation task are shown in Table 2. This step is shown as Step 1 in Figure 1.

Annotation Alignment After obtaining human annotated code-switched utterances with the slot spans (3rd column Table 2), we transfer the slot names (e.g., *date_time*, *alarm_name* etc.) from the English parse to the code-switched utterance to obtain human labeled data for code-switched semantic parsing. As we preserved the slot spans

(Table 2) as the part of the parsing task there is a 1:1 mapping between the english parse and the code switched utterance. The example below denotes this step, we map each date_time slot to a unique span id to get a 1:1 alignment.

```
[ Set alarm [ for 4:30 am on Tuesday ]date_time and [
  Thursday ]date_time of next week ]create_alarm
Set alarm [ for 4:30 am on Tuesday ]1 and [ Thursday ]2
of next week
```

At the end of this step, we had collected 10k CS utterances, along with the semantic parses derived using the alignment above. Details on our human annotated Hinglish-TOP dataset are in Table 4.

3.2 CS Data Generation using mT5

In order to generate synthetic code-switched utterances, we fine-tuned a mT5 (Xue et al., 2020) model for the task of code-switching the English utterances using the training data obtained in the previous step. The input to the T5 model is pre-processed English utterance (second column Table 2) and the output is the code-switched utterance containing slot spans (3rd column Table 2). The TOPv2 dataset originally had over 180K utterances, of which we manually annotated only 10.8K (3K train, 1.3K validation and 6.5K test split) utterances. The annotated train set was used to fine-tune the model. The remaining utterances from TOPv2 was used as monolingual input to the mT5 model to generate synthetic code-switched utterances.

Since mT5 is a text to text model, the generated synthetic code-switched utterances were not always structurally correct. This resulted in errors when

English Utterance	Code-switched Utterance	Rating
What is the weather in Canada ?	Canada me mausam kaisa hai ?	Natural & Semantically similar
How long is left before my alarm goes off ?	Mera alarm kab tak band rahega ?	Natural but Not Semantically similar: English utterance is asking for time remaining, while CS utterance is asking for trigger time
Remind me the first Tuesday of every month that bulk trash pick up is the next morning .	Muje every month first Tuesday ko remind karaye ki bulk trash pick up is agle subha hai .	Not Natural but Semantically similar: Phrase "bulk trash pick up is agle subha hai" is un-natural
How long of a delay is traffic causing	traffic kaisa delay causing hai	Not Natural & Not Semantically similar

Table 3: Examples of CS utterances generated by the XXL mT5 model, along with their human ratings.

English Vocabulary size	4857
Romanized Hindi Vocabulary size	1931
Total utterances	10,896
Avg. # of Hindi tokens per utterance	4.36
Avg. # of English tokens per utterance	3.82
Avg. # of CS points per utterance	3.56

Table 4: Dataset statistics for the human annotated Hinglish-TOP dataset.

trying to align the output with the English semantic parse as mentioned in Section 3.1.2.

3.2.1 Data Filtering

Sometimes the code-switched utterances generated from the fine-tuned mt5 model have structural errors which prevent aligning the semantic parse for these code-switched utterances to their English counterparts. We filter out such utterances using a syntactic rule based filter. In particular we removed examples which contain:

- Unequal number of argument spans in English and code-switched utterances.
- Improper span-ID formatting (when the span-ID could not be extracted) for code-switched utterances.
- Unequal number of opening and closing spans for code-switched utterances.
- Non-matching span-ID in English and code-switched utterances.

Examples of these errors can be seen in Table 5.

Finally, post data-filtering, we align the semantic parse annotations of the English utterance to the code-switched utterances using the same approach as in Section 3.1.2. The amount of syntactically correct synthetically generated data different seed set sizes is shown in Table 7. Post data-filtering using 100 fine-tuning examples, the small mT5 model had

a yield of 47.3% on the remaining TOPv2 dataset, while the XXL model successfully converted 82.0% of the English utterances.

3.3 Data Quality Assessment

We conduct both intrinsic and extrinsic evaluation of the generated synthetic data. The results of extrinsic evaluation are presented in Section 5. In this section we focus on 2 main questions:

What is the data quality of the human annotated data? To measure the quality of the CS data we created a second annotation task in which speakers proficient in both English and Hindi rated the CS utterances for their *naturalness* and *semantic equivalence* to the original English utterance. For naturalness, raters were just shown the CS utterance and were asked to answer the question: *Would a native speaker utter this utterance naturally in a conversation?* For semantic equivalence, raters were shown both the English and the CS utterance and were asked: *Do these sentences convey exactly the same meaning?* This serves as an upper bound on naturalness and semantic equivalence. Example ratings for naturalness and semantic equivalence are shown in Table 3.

We did this study on a sample of 2021 utterances. Each CS utterance was rated by two human raters, and conflicts in opinion were resolved by using a third rater and taking the majority opinion. For the human annotated data (Table 6, the raters found over 99% of the sampled utterances to be natural as well as semantically equivalent to the source English utterance.

How good are the synthetic CS utterances when judged by a human? We also performed the same quality measurements on the mT5 generated data to estimate naturalness and semantic equivalence. We randomly sampled 500 generated queries from the small and XXL model trained using the

English Utterance	Generated Code-Switched Utterance	Error Type
[9 pm] ₁ [appointment for photos] ₂ and remind [me] ₃ [an hour before] ₄	[mujhe] ₃ [9 pm] ₁ ko [photos ke liye appointment] ₂ hai aur [mujhe] ₃ [ek ghante pehle] ₄ yaad dilaayen	Unequal number of argument spans in English (one [me] ₃) and code-switched query (two [mujhe] ₃)
play [song] ₁ [Heart is on fire] ₂ on [spotify] ₃	[spotify] ₃ par [song] ₁ [Heart is on fire] ₂ ko bajao	Improper span ID formatting for Code-Switched utterances
Change [banking] ₁ reminders [from] ₂ [once a week] ₃ [to] ₄ [twice a week] ₅	[banking] ₁ reminders ko [[ek bar har week] ₃ [dohrayen] ₄	Unequal number of opening and closing spans for Code-Switched utterances
Remind [me] ₁ to [email] ₂ [Michelle] ₃ [on Tuesday] ₄ [about] ₅ [the recital] ₆	[Mujhe] ₁ [Tuesday ko] ₇ [Michelle] ₃ ko [email] ₂ karne ke liye yaad dilaayen	Non-matching span IDs in english and code-switched utterances

Table 5: Examples of errors that were automatically filtered from the mT5 generated code switched utterances.

Data source	Naturalness	Semantic Equiv.
Human	99.2% (0.72)	97.7% (0.69)
mT5 Small	86.1% (0.8)	43.7% (0.88)
mT5 XXL	98.1% (0.95)	89.5% (0.73)

Table 6: Naturalness and Semantic Equivalence Statistics from Data Quality Task. The κ values for rater agreement is listed in brackets.

Seed Set Size	Small mT5 Throughput	XXL mT5 Throughput
100	47.3%	82.0%
500	64.4%	93.7%
1k	72.8%	96.1%
2k	89.7%	97.5%
3k	92.7%	98.3%

Table 7: Percentage of input English queries that get code switched post data-filtering with varying number of seed examples for both XXL and small model. We use the remaining TOPv2 dataset (173,042 utterances) for generating the CS utterances.

largest seed set. We observed that the data generated by the XXL model was of much higher quality compared to the small model. For the XXL model, the raters found over 98% of the sampled generated utterances to be natural and over 89% were semantically equivalent to the source English utterance. Additionally, we observed high κ (>0.65) values for inter-annotator agreement; details can be found in Table 6.

4 Experimental Setup

Additionally for extrinsic evaluation, we answer the following research questions through our experiments: **(a)** What effect does adding augmented data to the training set have on the semantic parsing task? **(b)** What is the effect of varying the size of

initial seed set for the data augmentation step on the overall performance?; **(c)** How does the performance vary across domains?

To determine the quality of the augmented data, we trained semantic parsing models for code-switched utterances. We fine tuned a second mT5 model for this task. We compared the performance of our semantic parsing models with and without augmented data while keeping everything else the same. To study the efficacy of our technique with regards to the seed set, we trained models by varying the seed set size 100, 500, 1000, 2000 and 3000 utterances. Further, using an mT5 model fine-tuned on this training data, we create augmentation data using the rest of un-annotated utterances from the TOPv2 dataset as described in Section 3.2. Although we only release the augmented (model-generated) data from the model trained on full 3000 training examples, we created an independent augmented data set for each batch of training data from the training split.

For the mT5 models, we use the public checkpoints provided by Xue et al. (2020). We only roughly tuned the hparams. For the XXL (13 billion parameters) mT5 model training, we use a learning rate of 0.001, batch size of 512 and fine tune for 20k steps. For the small mT5 model (300 million parameters), we fine tune for 200k steps keeping rest of the parameters exactly the same. We checked for overfitting by choosing earlier (10k, 50k steps) checkpoints but obtained best results with the parameters mentioned.

Models used for comparison For varying the size of the seed set, we trained three mT5 models.

1. A CS utterance generation model, used to generate the augmented data for each seed set.
2. A semantic parser trained solely on the seed

Human Annotated Utterances	
EN: Will it be [above] [59] in [celsius] at [Menlo Park] [tomorrow morning]	CS: Kya [kal subah] [Menlo Park] me [59] [celsius] se [upar] hoga?
EN: Remind [me] [on September 1] that [its my best friend's birthday] !	CS: [September 1 ko] [mujhe] remind karaye ki [mere best friend ka birthday he]
Model Generated Utterances	
EN: I need to change [my] [car maintenance] reminder to [Tuesday 9 AM]	CS: Mujhe [mere] [car maintenance] reminder ko [Tuesday subah 9 bajhe ke liye] badalne ki zaroorat hai
EN: Text [my] [mom] that [I will um reach home in ten minutes]	CS: [Meri] [ma] ko text karo ki [mai um ten minutes me pahuch jaunga]

Table 8: Example of Human annotated CS utterances and Model Generated CS utterances, along with the original English utterance. Same slots have been marked with the same colour across utterances.

set.

3. A semantic parser trained on the seed set and the augmented data.

The models were tested over the test data set from the data annotation task of 6.5k utterances.¹ We used the exact-match accuracy to judge the performance of these models.

5 Results

As seen from Figure 3, we observe that increasing the size of the training data, improves the performance of both models(with and without augmented data set) significantly. While there is a consistent improvement on adding more training data, the improvement itself seems to diminish with more seed set examples. Empirically, we observe that the models which make use of the augmented data reaches saturation, i.e. the value after which additional data points don't add to the overall performance of the model, quicker than the models which do not make use of the augmented data. This shows that the synthetic data generated by our technique is useful for the end task of semantic parsing.

As seen from Figure 3, the performance of the mT5 model trained using the augmented data was consistently better than that trained without it. The XXL model trained using our technique with a seed

¹Note that we report all results on this code switched human annotated testset.

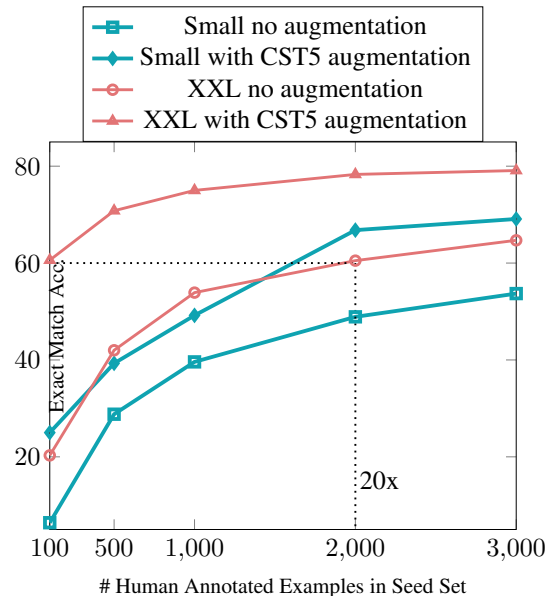


Figure 3: Comparison of Exact Match Accuracy for semantic parsing models trained with and without CST5 augmentation, for varying seed set size. The dotted line shows the XXL model reaches EM of 60% using CST5 augmentation with 100 examples, equivalent to fine tuning a XXL model with 2k examples.

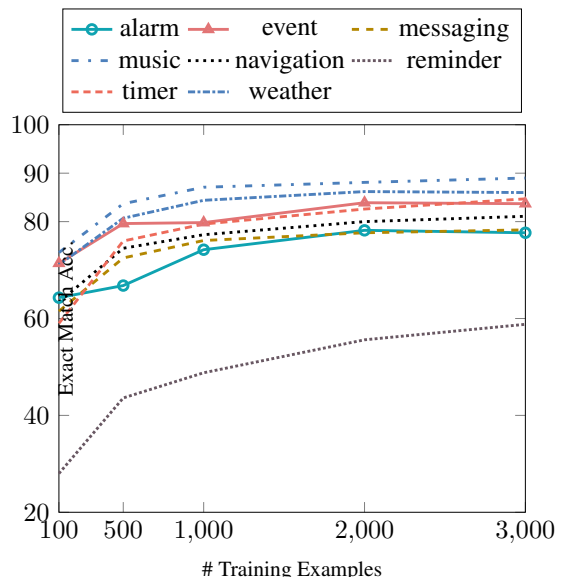


Figure 4: Exact Match Accuracy for different domains in the Hinglish-TOP dataset, with varying amount of seed training examples used for data augmentation.

set of only 100 examples performs similar to the model trained with 2000 human annotated examples. This is over a 40% absolute improvement with 100 examples. Even with 3000 human annotated examples we see over 15% improvement in EM accuracy. This goes to show the effectiveness of our data augmentation technique, especially in low-resource settings.

The XXL mT5 model was able to produce a significantly higher number of code-switched utterances compared to the small mT5 model, which ultimately led to the significant gain in its performance. The number of generated code-switched utterances post data-filtering can be seen in Table 7. As seen from the results (Figure 3) the XXL mT5 model is significantly better than the small mT5 model, both as a parser and a data generator. We also observed that adding the augmented data, allows the model to handle more complex queries. A few examples from the models trained on full 3000 training examples have been included in Table 8.

Moreover, after an in-depth domain-wise analysis of the models performance (Figure 4), we observed that the model struggled with the *reminder* domain. It is interesting to note that the *reminder* domain had the highest number of unique intents, 16. Also, reminder domain had the highest number of average intents per utterance of 2.66. This collectively led to the low performance of the model on this domain. Some of the domains saturated quicker and did not see much improvement upon adding more data points. We can empirically observe that for domains with more inherent difficulty (reminder for example) the value of added data is more than simpler domains (weather for example).

Results on CSTOP Additionally, we repeated a similar experiment for CSTOP (Einolghozati et al., 2021), which contains Spanish-English code-switched utterances annotated in the TOP schema. Unlike our approach CSTOP doesn't have parallel data with TOPv2. Since only the weather domain was common between CSTOP and TOPv2, we work with the weather domain subset. As we need parallel English data to finetune the mT5 model, we used Google Translate to generate parallel English data for 100 Spanish-English queries. We used this to finetune a mT5 model and generated new synthetic data using TOPv2 weather domain as the monolingual dataset. For semantic parsing using 100 queries from the training data we observe

similar trends as Hinglish. We observe an improvement in the parsing performance from 69.2% EM to 77.8% EM on the testset using the synthetic code-switched data.

6 Conclusion

We proposed CST5, an approach to overcome the scarcity of labeled data for code-switched semantic parsing. CST5 generates code-switched utterances from English utterances using a large pretrained LM mT5, and a small number of seed examples. We showed that the generated utterances were of high quality, as determined by human raters in a quality annotation task. By aligning the generated utterances with the semantic parse of the original English utterance, we derived a large supervised dataset for training a code-switched semantic parser using a labeled dataset in English.

We did both intrinsic and extrinsic evaluation of the human-annotated as well as the synthetic data. Human raters found 98% of the synthetic data to be natural and 89% to be semantically equivalent. Externally, our experiments also demonstrated that data augmentation using CST5 effectively reduces the data requirements by 20x for the code-switched semantic parsing task. To further research in this area, we will release the dataset of over 10k manually annotated Hinglish utterances, along with over 170k examples generated using CST5². While we applied CST5 to Hindi-English code-switching, our approach is general and can be applied to any other language pair that exhibits code-switching. In the spirit of CST5, we believe that using large pre-trained language models to perform data augmentation for other code-switched NLP tasks is an attractive future direction to explore.

References

- Gustavo Aguilar, Fahad AlGhamdi, Victor Soto, Mona Diab, Julia Hirschberg, and Tamar Solorio. 2019. Named entity recognition on code-switched data: Overview of the calcs 2018 shared task. *arXiv preprint arXiv:1906.04138*.
- Ekin Akyürek, Afra Feyza Akyürek, and Jacob Andreas. 2020. Learning to recombine and resample data for compositional generalization. *arXiv preprint arXiv:2010.03706*.
- Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, N. Trep-

²This data is available at <link hidden for anonymity>.

- per, and Naama Zwerdling. 2020. Not enough data? deep learning to the rescue! In *AAAI*.
- Jacob Andreas. 2019. Good-enough compositional data augmentation. *arXiv preprint arXiv:1904.09545*.
- Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Manish Shrivastava, and Dipti Misra Sharma. 2018. Universal dependency parsing for hindi-english code-switching. *arXiv preprint arXiv:1804.05868*.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
- Khyathi Raghavi Chandu and Alan W. Black. 2020. Style variation as a vantage point for code-switching. In *Interspeech*.
- Ching-Ting Chang, Shun-Po Chuang, and Hung yi Lee. 2019. Code-switching sentence generation by generative adversarial networks and its application to data augmentation. In *INTERSPEECH*.
- Xilun Chen, Asish Ghoshal, Yashar Mehdad, Luke Zettlemoyer, and Sonal Gupta. 2020. Low-resource domain adaptation for compositional task-oriented semantic parsing. *arXiv preprint arXiv:2010.03546*.
- Jeremy R Cole, Nanjiang Jiang, Panupong Pasupat, Luheng He, and Peter Shaw. 2021. Graph-based decoding for task oriented semantic parsing. *arXiv preprint arXiv:2109.04587*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Long Duong, Hadi Afshar, Dominique Estival, Glen Pink, Philip R Cohen, and Mark Johnson. 2017. Multilingual semantic parsing and code-switching. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 379–389.
- Arash Einolghozati, Abhinav Arora, Lorena Sainz-Maza Lecanda, Anuj Kumar, and Sonal Gupta. 2021. El volumen louder por favor: Code-switching in task-oriented semantic parsing. *arXiv preprint arXiv:2101.10524*.
- Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. *arXiv preprint arXiv:1810.07942*.
- Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. *arXiv preprint arXiv:1606.03622*.
- Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2021. [Data augmentation using pre-trained transformer models](#).
- Varun Kumar, Hadrien Glaude, Cyprien de Lichy, and William Campbell. 2019. [A closer look at feature space data augmentation for few-shot intent classification](#). In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 1–10, Hong Kong, China. Association for Computational Linguistics.
- Grandee Lee, Xianghu Yue, and Haizhou Li. 2019. Linguistically motivated parallel data augmentation for code-switch language modeling. In *INTERSPEECH*, pages 3730–3734.
- Kenton Lee, Kelvin Guu, Luheng He, Tim Dozat, and Hyung Won Chung. 2021. Neural data augmentation via example extrapolation. *arXiv preprint arXiv:2102.01335*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Vladislav Lialin, Rahul Goel, Andrey Simanovsky, Anna Rumshisky, and Rushin Shah. 2020. Update frequently, update fast: Retraining semantic parsing systems in a fraction of time. *arXiv preprint arXiv:2010.07865*.
- Laiba Mehnaz, Debanjan Mahata, Rakesh Gosangi, Uma Sushmitha Gunturi, Riya Jain, Gauri Gupta, Amardeep Kumar, Isabelle Lee, Anish Acharya, and Rajiv Ratn Shah. 2021. Gupshup: An annotated corpus for abstractive summarization of open-domain code-switched conversations. *arXiv preprint arXiv:2104.08578*.
- Giovanni Molina, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, Nicolas Rey-Villamizar, Mona Diab, and Tamar Solorio. 2019. Overview for the second shared task on language identification in code-switched data. *arXiv preprint arXiv:1909.13016*.
- Massimo Nicosia, Zhongdi Qu, and Yasemin Altun. 2021. Translate & fill: Improving zero-shot multilingual semantic parsing with synthetic data. *arXiv preprint arXiv:2109.04319*.
- Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.
- Subendhu Rongali, Luca Soldaini, Emilio Monti, and Wael Hamza. 2020. Don't parse, generate! a sequence to sequence architecture for task-oriented semantic parsing. In *Proceedings of The Web Conference 2020*, pages 2962–2968.
- Bidisha Samanta, Sharmila Reddy Nangi, Hussain Jagirdar, Niloy Ganguly, and Soumen Chakrabarti. 2019. A deep generative model for code-switched text. In *IJCAI*.
- Arnav Sharma, Sakshi Gupta, Raveesh Motlani, Piyush Bansal, Manish Srivastava, Radhika Mamidi, and Dipti M Sharma. 2016. Shallow parsing pipeline for hindi-english code-mixed social media text. *arXiv preprint arXiv:1604.03136*.
- Peter Shaw, Ming-Wei Chang, Panupong Pasupat, and Kristina Toutanova. 2020. Compositional generalization and natural language variation: Can a semantic parsing approach handle both? *arXiv preprint arXiv:2010.12725*.
- Ishan Tarunesh, Syamantak Kumar, and Preethi Jyothi. 2021. [From machine translation to code-switching: Generating high-quality code-switched text](#). 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 3154–3169, Online. Association for Computational Linguistics.
- Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2019. [Code-switched language models using neural based synthetic data from parallel sentences](#). In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 271–280, Hong Kong, China. Association for Computational Linguistics.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.