Adapting Language Models for Domain Shift by Augmenting Domain Description
Domain Description Augmented Language Models for Domain Shift
Domain Description Augmented Training Enables Zero-shot Domain Generalization
(maybe not augmented, here our task must depend on domain description instead of using it as augmented extra info to improve the performance.)

TODOs/Thoughts

1. Pretraining a model that is able to condition domain rules to change the generated code
    a. Prompting Codex to generate a lot of domain-specific DB-question-SQL triples, or even domain rules have given a few in-context examples. Pre-training GPT-Neo/CodexGen to perform well on the 8 datasets, also given domain DB or also questions+domain rules, ask Codex to generate some labeled data for new domains., to fundamentally change the game!
2. GPT3/Codex + Google Search to find real data, modify them and annotate them to create new task datasets.
3. Think it as Memory-assisted prompt editing for GPT/Codex after deployment
    a. allow users to correct errors directly through interaction, without retraining, by giving feedback on the model's output.
4. Ask Mari, Caiming?
5. Think of domain rules as extra database descriptions?

Learning from Human Feedback

Natural Language Knowledge Augmented Domain Generalization
Domain Adaptation by Describing the Knowledge Gap in Natural Language

Motivations

Proposal storying

- NL domain rules are more flexible
- Also can think about this paradigm that abstracts annotation instructions in domain descriptions. Instead of asking human annotators to follow them to label data and train a model to find this pattern, the model can learn to follow them to make predictions by themselves. This saves a lot of time and annotation efforts, and also controls the model in a more direct and explainable way.
- **In-context examples vs domain description**: domain description can disentangle task and domain tasks and make joint cross datasets for the same task easier (more compositional). Otherwise, we can have the same database and questions but with different SQL (>2.5 and > 3). Domain rules focus on real conventions, no need to x NL variances to annotate a lot of data. No need to annotate labels just clarify inputs. Domain rules can be more efficiently collected by domain experts, allowing us to just focus on the real important parts. Maybe the user will change his/her preference over time.

    - The big performance gap between NLP research and real-world applications -> one of the biggest reasons is the “domain” shift
    - There are a lot of domains in real-world that could not be covered by NLP research and PLMs
        - domains: c (Stanford OVAL lab, cross different app apis)
        - domain/metadata: domain/professional knowledge, jargon, preference, conventions, knowledge updates, feedback? Multiturn?…also they change over time…
            - (e.g., even in the same company, different groups using the same or similar databases, have different conventions. For group 1, rating > 2.5 is good; for group 2, rating > 4 is good)
    - Previous ML research focuses on collecting data from single or multi-domain and without involving domain experts; (x, y) is collected and used to train a model y = f(x); thus, the performance drops during testing if x-test distribution is different from x, which we call it distribution shift.
    - Moreover, there are many cases where we could not expect the trained model to adapt to new test domains without any extra domain info (different industries, companies, products, preferences…). In these cases, human task experts also cannot perform well without additional professional domain/convention training. To close this fundamental gap, we argue that researchers should try to collect task data from various domains and annotate domain info for each domain. In this way, instead of collecting (x, y), we have (x, d, y) and train a model y = f(x, d). Once the model is trained on this training data, we can expect it should adapt well to arbitrary domains no matter how big differences they have compared to training domains as long as the domain/convention info d is clear (containing all new domain info you need). Like Domain expert + Task expert (e.g., SQL expert). This is a new task definition and training paradigm that aims to fundamentally (in principle) close the big performance gap between NLP research and real-world applications
- Long-term goal: new domain shift training paradigm, task definition,
- Advantages
    - Can quickly adapt and scale NLP models across different domains by only providing domain info in the new domain without retraining…A more scalable way to solve the domain shift performance drop problem.
    - saving annotation for different domains, not need a task expert to annotate more data.
    - If more people follow this paradigm to define tasks and collect data, we can better leverage many annotated data from different domains for the same task (with GPT3/Codex, we don’t need to annotate a large number of training datasets, but focus more on high-quality real-world test datasets). Good for data sharing across different data annotation efforts (previously very domain-specific labeled data is not useful for other domains), also *good for domain generalization*, making the performance gap smaller.

    - Domain info d is also flexible. We can explicitly update it to control the model’s prediction and do it anytime. E.g., new fancy terms/words occur (temporal data), you can just add it to new domain info *without retraining and re-collecting data*.
    - A user can quickly update the rule to correct the parser’s mistakes without retraining.
    -
- Ours vs prior work
    - Ours vs WILDS
        - WILDS: input distribution shift to test model’s generalization ability. They still are possible to answer without extra information. Their domain differences are hard to write in text. f(x_hat)
        - Ours: semantic shift to test if the model can incorporate new domain info. You must refer to new domain info to answer. f(x, d)
        - Compared to WILDS, we focus on task settings where domain experts/info are a must to adapt trained models on new domains (not just x distribution shift), which is also very common occurred in the real world.
    - Ours vs In-context learning/tuning - task instructions
        - They focus on zero-shot task generation. Here we focus on zero-shot domain/control generalization in the same task. Our descriptions are more fine-grained.
        - Methodology is the similar/same? But usage is different. Training models to more fine-grained (instead of task-level instructions - maybe can with same input in the same task but different) adapt to new explanations. *Ask Yizhong* (vs InstructGPT/T0)?
        - Explanations as task instructions, f(x, t) -> f(x, d), d as a domain explanation
        - see domain description as the instruction before the question. maybe no need to annotate training data? Maybe still need to, that is why T0 needs to annotate a lot of task instructions.
        - Can show how GPT3/Codex/InstructGPT/T0/Yizhong’s model fails on our new tasks.
        - It is also similar to task instructions. They focus on task-level (their goal is to enable zero-shot task generalization) but we focus on domain level (with task, to enable zero-shot “domain” generalization). We try to solve one of the biggest bottlenecks in ML current research.
        - Similar to incorporating task instructions in model training to enable zero-shot task generalization, we aim to incorporate natural language explanation/metadata of target domains (domain knowledge) in model training to enable zero-shot domain generalization.
    - Ours vs Retrieval augmented methods
        - They retrieve relevant information to augment models to improve predictions. Don’t directly/fine-grained control the prediction.
- More ….
-

Research questions

- How does the domain/semantic-specific rules/info look like?
    - Natural language sentences describing rules, easy for retrieving some rules given a question
    - Key-value pairs?
    - Follow by a few labeled examples? Negative examples, unfavorable clues, such as Things to avoid, from the instruction
- How to collect these domain descriptions?
    - During testing, can directly run the model with x and an empty d and check its predictions to find errors -> then iteratively add rules to domain descriptions to adjust the domain description.
    - How scalable would the domain info collection be?
- how to make this metadata/explanations (by human/domain experts or auto-generating from related domain corpus)?
    - A more challenging setting is to only include domain docs instead of domain knowledge (database schema detailed description)
- What metadata/explanations could be really helpful? In what formats?
- Do we need real data? Maybe not very necessary for the training data (even for domain info), but must have real test data.
- Noah - it is hard to collect this domain info? Answer: we only need to make sure test datasets are real and have good domain info. We needn't do it for training. And domains should be defined and in these cases defining domain info is hard…
- Why not retrieve d from the wiki?
    - Most people who want to adapt NLP models are domain experts. It might be more efficient for them to implicitly collect them to sort out this domain info.
    - Some
    - Many domain-specific terms, /knowledge, conventions, or preferences could not be googled or searched
    -

- Explanation of different types
    - Explanation as user feedback
        - Training the model to interactively incorporate humans' feedback to change its predictions
        - Task expert + user personal feedback (controllable)
        - Can include tasks: Spider + feedback and more! To show another usage…
    - Explanation as dialog (combined with feedback to *have another project*)

- See user follow-up questions as explanations to adapt the model to dialogs
- Explanation as reasoning info/steps
  - See prediction reasoning info/steps as the explanation to improve explainability and performance.
  - *Prompt GPT3 to generate extra explanations given the task inputs and then feed them back to the original inputs as explanations and see if this improves GPT3's performance?* chain of thoughts*, can use GPT3 auto-generated explanations to train smaller models* (we should exclude these tasks in this project since PLMs are able to generate this info. Or maybe yes because they are still not perfect?)

Task definition

- Tasks (to find some tasks requiring task expert + domain expert, also the number of domains as groups should not be too many)
  - y-d: NER (Xuan), intent detection (Ruiqi)
  - Compositional generalization?
  - SKG benchmarks incorporating domain knowledge
    - new text-to-SQL Spider/QA: before tradition -> Spider: f(x) -> f(x, s); now Spider 1.0 -> 2.0: f(x, s) -> f(x, s, d)
      -
    - Split by databases, apps, tables…
  - Extending real Spider to SParC~ real conversational text-to-SQL: ATIS
  - Generalize to new Python libraries with API docs -> scale-up PL impact
    - Maybe no need to split data by library
    - Check if PL people do very good SQL visualization -> explaining SQL
  - Check all possible **cross-domain** NLP/cv tasks
  -
  - Dialog state tracking
    - Think about the AllState case
    - Can think of it as NER (maybe not exactly the same), text-to-SQL in single/dialogs. Collect many different existing datasets… think about how to **abstract more domain/semantic shift info besides ontology**
    - Prompt GPT3/InstrcutGPT to generate some dialogs, and then fake some dialog ontology schema as the domain description, to annotate single turn DST labels. Also, think about what are the other
  - Professional domain intensive tasks - Data Science/Robotics/Healthcare/Law/Bio…

    - All different sentiment analysis datasets combined (need to have a big performance drop from in-dis to out-dis)? - classification (**Amazon-wilds dataset**), MultiNLI? sentiment analysis on different products
    - Culture/geo shift
        - Good is not that positive in the US but is good in East Asia - knowing this helps humans adapt to different cultures.
    - Others
        -
- Annotation
    - Leveraging existing datasets
        - Questions and databases are naturally from domain experts and annotate domain descriptions for all databases.
        - metadata/explanation, refer to tableau AskData preprocess (find docs)) - make sure a non-domain SQL expert can write correct SQL queries given the extra domain info.
        - For example, rewrite Spider questions and add domain metadata/explanations (with multiple valid gold queries), metadata for columns and tables… and more… different question styles mapping to the same query.
        - Rewrite inputs in existing datasets to exclude domain/feedback/follow-up info for the same domain in a description and questions that don't have this info.
    - Leveraging GPT3/Codex
        - Generating domain-specific questions with domain knowledge
    - Asking Turkers/Upwork to upload their own databases and questions, pay them by their quality.
    - Asking friends who are working in different industries
        - Siwei -finance
        - Rahul - Bio?
    - Resources
        - No need to find domain-specific questions directly for included tasks. can rewrite domain questions in other tasks to reflect this domain shift.
        - Professional forums/Kaggle/StackOverflow-ben

Methods

- T0
    - See task instructions as explanations, f(x, t) -> f(x, d), d as a domain explanation
- In-context learning/tuning
    - InstructGPT
        - Use it to benchmark
        -

Datasets

- Text-to-SQL
    - Medical Info: MIMIC, more, code
    - ***Text to SQL in the wilds***
    - Finance: shared alphavantage data,example-1(not very good)
    - SciSever SDSS database + SQL, more (ask Yiru)
    - Kaggle DB, SQLite
    - Zillow db, airbnb
    - Many data at the bottom
- QA
    - FinNLP

Ask people (**not just asking for DB data but also textual data - general QA/Summarization/NER…**)

- Different domains: Mari, Peng Xu, Spider followers-SJTU
- BioNLP/MedNLP
    - Next-Generation Analytics for Omics Data, GDC data, demo
    - BioNLP workshops
    - People: Sheng Wang (Drug Discovery), Lu Yang, Sitong, Xuan Wang (Chemistry), Chang Ma
    -
- Law/Legal:
    - Legal NLP
    - People: Ben
- Econ/Finance/Business
    - MS common data model, Wharton Data, FinNLP, InsuranceQA, FinQA
    - People: Peng Chen (Econ),tan tian, Xuyan Xiao, Ruiting (Accounting), Zhao Jin (Fin)...
- Kelvin Lu: temporal shift
- CV/Robotics
    - Jesse Thomason
- Sports and entertainment
    - Michale Lee, zhoujun CHENG.
- Public service
    - Data mining datasets
    - Free gov data

References

- Pang Wei Koh’s talk

Questions to ask domain experts

1. What is your professional domain?
2. When and why do you need to analyze data? For what? For research/publishing? Also, corresponding industrial needs? What roles?
    a. Accounting terms (EPS…), SEC..alphasense startup?
    b. Role: data scientist in the research department of a hedge fund company
3. In what kinds of formats? Like databases, tables, graphs, lists? All numbers, text, mixed, or others?
    a. Textual data, tables. mixed…
4. What does the data look like? About what?
    a. SEC, credit card transactions from data vendors, website clicking
5. Where are these data? Is it publicly available? Online forums? Bloomberg?
    a.
6. What kinds of questions do you ask about the data?
    a. More analytical? Retrieval? Preprocessing? Plotting? Modeling?
    b. Examples? Including much jargon and ignoring domain knowledge? How complex?
        i. later…
    c. Where can I find these questions? If not, how can we collect, modify, or create them?
        i. Alternative data research, company earning's call…. NYT/Bloomberg's coverage report… later…
7. What tools and programming languages do you use?
    a. Python, SQL, R, Stata…
8. Are possible data-question-program triples available?
9. Do you know possible data in other domains?

Slides:

Introduction Slides Semantic Shift: Introduction

Synthesis Data Generation for Text to SQL Semantic Shift: Synthesis Data for Spider

Tasks:

Text to code snippet

1. NL2Bash

Sentiment Analysis

1. Amazon Review
2. GoEmotions: A Dataset of Fine-Grained Emotions
3. TweetEval:

Summarization

1. BigPatent: summarize patent in 8 fields.
2. BillSum: US Legislation
3. arXiv: research fields
4. PubMed: bio
5. MeQSum

Text Classification/NLI

1. e-SNLI: premise + hypothesis + **explanation**
2. SentEval: 10 tasks for sentence representation evaluation
3. WOS (Web of Science Dataset): documents in 134 categories which include 7 parents categories (domain).
4. MIMIC
5. BIOSSES: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5870675/
6. MedNLI
7. PHM2017: Detection of Personal Health Mentions in Social Media
8. EUROVOC
9. Evidence Inference

QA

1. MedQuAD
2. CliCR: Clinical Case Reports for Machine Reading Comprehension
3. BIOMRC

IE

1. GENIA: NER, RE, dependency parsing
2. MedMentions

Medical:

Asma Ben Abacha: built many medical data.
https://paperswithcode.com/datasets?mod=medical
https://github.com/orgs/bigscience-workshop/projects/6/views/1

Data collection TODOs

1. How to find databases in different professional domains?
    a. Should not be that hard - IBM domains, research datasets in different communities…
    b. Jiacheng Codex
2. Question, SQL, Question-SQL
    a. Maybe sometimes first finding SQL/questions is better.
    b. Codex - google web research
    c. Github (.sql, sqlite…), stackoverflow… but be aware of codex already has seen them…
3. Think about the possibility of using Upwork/AmazonTurk/fiverr to find professional workers and ask them to provide the database and q-sql pairs.

Summarization over different tasks in specific domains

1. Clinical
   https://arxiv.org/abs/2112.05780

Text-to-SQL

1. Healthcare
   MIMICSQL
   Text-to-SQL data based on MIMIC-III
   Paper: https://arxiv.org/abs/1908.01839
   Data: mimicsql_data
   Author: Ping Wang

   Text to SQL in the wilds

   OMOP SQL query https://data.ohdsi.org/QueryLibrary/

   UMLS
   https://www.nlm.nih.gov/research/umls/implementation_resources/query_diagrams/index.html

   RxNorm
   https://www.nlm.nih.gov/research/umls/rxnorm/docs/techdoc.html

2. Molecules
   Integrated Database of Small Molecules
   https://idsm.elixir-czech.cz/
   SPARQL GUI

   DrugBank
   Chembl
   https://github.com/datagrok-ai/public/blob/master/packages/Chembl/queries/queries.sql

3. Bio
   BioSQL https://biosql.org/wiki/Main_Page
   https://biopython.org/wiki/BioSQL
   https://biosql.org/wiki/Schema_Overview
   https://research.cs.cornell.edu/btr/bioinformaticsschool/slides/sun-DBs.pdf
   https://www.ncbi.nlm.nih.gov/sra/docs/sra-bigquery-examples/
   https://hal.archives-ouvertes.fr/hal-00964169/document

4. Finance
   SEC
   https://www.sec.gov/dera/data/financial-statement-data-sets.html

   IRC
   https://www.irs.gov/irm/part2/irm_02-005-013

5. Legal

   USPTO
   https://www.kaggle.com/code/jessicali9530/how-to-query-uspto-oce-patent-claims-research-data/notebook

   https://arxiv.org/abs/1404.7447

   US Supreme Court Database
   https://ericnystrom.org/posts/Supreme_Court_Database_for_the_SQL-minded/
   Real questions about SQL https://forums.epo.org/patstat-product-line-28/

6. Sports

NBA
https://github.com/mpope9/nba-sql

NFL
https://www.kaggle.com/competitions/nfl-big-data-bowl-2021/
https://github.com/BurntSushi/nfldb

QA

1. ClinicalKBQA
   Paper: https://arxiv.org/abs/2108.00513
2. emrQA
   Paper: https://aclanthology.org/D18-1258/
3.

Summarization

1. MEDIQA
   Paper: https://aclanthology.org/2021.bionlp-1.8/
2.

Database:

- ICU database:
  - AmsterdamUMCdb
  - eICU
  - HiRID
  - https://mimic.mit.edu/docs/iv/modules/icu/
- Drug
  - DrugBank SQL Examples
  -

Other:
N2c2: Unstructured notes from the Research Patient Data Registry at Partners Healthcare
https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/
Synthetic Healthcare Data Generator: https://github.com/synthetichealth/synthea
Synthetic Health Data Challenge Winning Solutions Webinar
EMS SQL database
Openfda Data

Healthcare data sources doctors use

- UpToDate
- Pediatric: American association of pediatric guidelines
- Radiology: https://radiopaedia.org/
- AAFP
- ACS surgery
- ACR radiology
- American college of gastroenterology

A real question example pediatric doctors ask:

- 6-month-old comes in, throwing up for 3 days, and doesn't have a fever. What's the cause?

Action List

Based on different domains, know what is the best datasets? Know different NLP tasks. Go over the data.

[Sync with Yiheng]
Hard Points:

1. **How do we define rules and domain knowledge?**
    a. **When should we provide domain knowledge?**
    b. **What kind of domain knowledge should we provide?**
    c. **What domain rules could actually help the model? (QA, …)**
    d. **Could separate the domain knowledge and data, much less annotation effort!**
2. **Where to find the SQL data?**
    a. **A lot of DB, but without SQLs and questions.**
3. **How to select useful data?**
    a. **First inference by codex without any domain knowledge provided, and focus on the false part?**

- **Paper reading**
    - **Look into paper: BIER**
        - **To see how they construct the dataset based on other existed datasets – "DIVERSE"**
            - Selection methodology:
                - 1) Diverse tasks;
                    - Queries and documents: long/short
                - 2) Diverse domains;
                    - Broad/specialized domains
                - 3) Task difficulties;
                    - Several tasks are based on existing literature and selected popular tasks which we believe are

recently developed, challenging, and are not yet fully solved with existing approaches.

        - 4) Diverse annotation strategies
          - Crowd-workers, experts, and feedback from large online communities.
      - How to construct based on the selected datasets?
        - *Include all info in each dataset? CHECK*
    - **Could we leverage the experience from it?**
      - **Define a selling point of the domain shift benchmark:**
        - “Natural/Real”?
        - “Coverage” in tasks and domains?
        - …
      - **When finding tasks, we have to think about:**
        - *Knowledge-intensive tasks* – Semantic parsing, QA, Fact verification, …? [knowledge intensive? Is it easy to clearly separate domain knowledge?]
        - The “Gap” is supposed to be large
        - The diversity between tasks – 3?
      - **When finding datasets, we have to think about:**
        - Difficulty
        - Quality – All high quality?
        - How to select the data from the original datasets?
  - **Looking for more papers in:**
    - **Domain shift in Semantic Parsing, QA, Verification, NER, etc.**

- **Domain shift definition over different tasks**
  - **[decided to include] Semantic Parsing**
    - **Sync with Yiheng**
      - **to see what has been done in semantic parsing on 8 traditional datasets**
        - Domain knowledge annotations in Dialogue (3 dev sets)
        - Domain knowledge annotations as Lists (7 dev sets without ATIS)
      - **Get to know how he defines the domain rules**
        - **The format: is this format the best? What other formats could we design?**
          - For Dialogue form, the annotation process is difficult and the model is unlikely to return the questions for the ambiguities in the questions.
          - ** But for the Dialogue form, if it works, the system could detect the ambiguity of the question and automatically narrows down the range of domain knowledge, which would be helpful for real use.*
          - For now, the better form is Lists. Other forms?

- **The content: when do we have to use domain rules?**
  - We as normal people, not domain experts, cannot understand the questions or cannot answer the questions correctly.
  - The questions have some ambiguities or are under specification.
  - The domain knowledge is quite helpful and important to answer the question correctly.
  - [Aim] With domain knowledge provided, we normal people could answer the domain-specific question correctly.
- **Introducing 3-4 new datasets/domains**
  - **Go over and summarize the datasets in this doc to think and see:**
    - **Whether the data must need domain knowledge to answer**
      - ** here the “must” means that the questions are likely to have incorrect predictions without the domain knowledge provided.*
        - *For we humans, we cannot understand the questions or answer the questions correctly without domain knowledge, not as domain experts.*
      - **What kind of domain rules does it have to provide in order to answer such questions, convention (definable in this single database) or jargon (Domain-specific), or even other cases?**
        - **Is it easy to separate specific and general domain rules from the data? (Annotation cost)**
        - **Is it helpful?**
    - **Whether the data is natural and of high quality written by a domain expert**
      - **Where does the DB come from?**
      - **Where do the question and SQL come from?**
      - **Are the questions similar to academic questions?**
    - **Could we use the data?**
      - **The necessity of domain knowledge?**
      - **Contributions?**
      - **How to select from the dataset? How to collect domain knowledge?**
    - **[Datasets]**

                - [MIMICSQL(HealthCare)tutorial, WangSheng, DrugBanktutorials, BioSQL, NBA]
                - []
            - [Methods to do]
                - Inference with CodeX without domain knowledge – Focus on the wrong part – see if there requires any domain knowledge – think of the points listed above
                - Provide specific examples.
        - Look for other domains' data from the web
- [decided to include] QA
    - Go over existing domain specific QA datasets to think and to see:
        - For QA, would domain rules be a must to answer the questions? Or would it be easier to answer such questions with domain rules?
            - Are there some similar rules among different questions? Do the questions share domain rules?
            - What kind of domain rules does it have to provide in order to answer such questions, convention or jargon, or even other cases?
                - Is it easy to separate specific and general domain rules from the data? (Annotation cost)
                - Is it helpful?
        - Whether the data is natural and of high quality written by domain expert
            - Where does the data come from? How to collect?
            - Are the questions similar to academic questions?
        - Could we use the data?
            - The necessity of domain knowledge?
            - Contributions?
            - How to select from the dataset? How to collect domain knowledge?
        - [Datasets]
            - [FinQA, ClinicalKBQA,]
        - [Methods to do]
            - Inference with CodeX without domain knowledge – Focus on the wrong part – see if there requires any domain knowledge – think of the points listed above
            - Provide specific examples.
- Others
    - Find other tasks that require domain knowledge
        - Verification?
        - Document classification: eg. Review analysis(音响和洗衣机声音)

            - **NER?**
            - **other knowledge intensive tasks**
- **Illustration of the gap between performances**
    - **Gap between “domain knowledge provided” and “domain knowledge not provided” on general model**
        - **To prove that the domain knowledge is helpful**
    - **Gap between “general model+domain knowledge” and “fine tuned domain expert model”**
        - **To prove that it still needs further study to use domain knowledge**

[Previous works]
Annotated datasets
This document includes domain knowledge of restaurants, geoquery, and advising.