Final Project

What we expect

A small research project!

- Submit a project proposal (9/15, optional, not graded)
- Present at a poster session (12/1)
- Submit a final project report to DEN (12/8)

NO LATE SUBMISSIONS ARE ACCEPTED

Report format

ICLR 2023 draft format (https://github.com/ICLR/Master-Template/raw/master/iclr2023.zip)

- You will write in LaTex (submit PDF)
- Overleaf is your good friend.
- Please include a link to your code base.

Three types of projects

1. Application projects
2. Analysis projects
3. Implementation projects

Type 1: Application projects

Goal: solve a real-world problem.

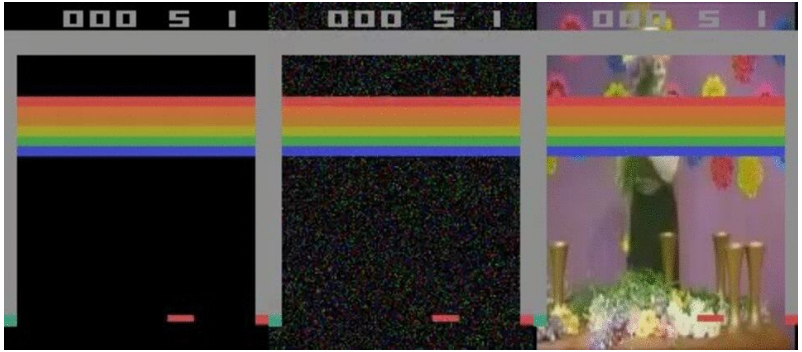
Examples:

- Apply deep reinforcement learning on a specific game
- Accelerate a deep model for a specific task.

Comparing with existing methods is important!

A concrete example

- Maybe you're interested in playing a game with reinforcement learning where agents are robust to noisy environments.
- Find an environment for this and get results for the strongest existing methods you can find.
- Improve on these methods with your own ideas.



https://github.com/facebookresearch/natural_rl_environment

Type 2: Analysis projects

Goal: discover useful/important insights

Examples:

- Analyzing the relationship between some hyperparameters and model performance.
- Analyzing the right way to prompt (communicate) with a large language model.
- Discovering the scenarios where current models usually fail.

You need to be more experienced to find the right thing to analyze.

A concrete example

- Language models do really well on commonsense reasoning tasks. I wonder if that's due to the model taking shortcuts and not really reasoning. 🤔
- They come up with a way to carefully gather data and then adversarially filter it. Models do better on this dataset than other datasets.

WinoGrande: An Adversarial Winograd Schema Challenge at Scale

Keisuke Sakaguchi*, Ronan Le Bras*, Chandra Bhagavatula*, Yejin Choi*†
*Allen Institute for Artificial Intelligence †University of Washington
{keisukes, ronanlb, chandrab, yejinc}@allenai.org

Abstract

The Winograd Schema Challenge (WSC) (Levesque, Davis, and Morgenstern 2011), a benchmark for commonsense reasoning, is a set of 273 expert-crafted pronoun resolution problems originally designed to be unsolvable for statistical models that rely on selectional preferences or word associations. However, recent advances in neural language models have already reached around 90% accuracy on variants of WSC. This raises an important question whether these models have truly acquired robust commonsense capabilities or whether they rely on spurious biases in the datasets that lead to an overestimation of the true capabilities of machine commonsense.

To investigate this question, we introduce **WinoGrande**, a large-scale dataset of 44k problems, inspired by the original WSC design, but adjusted to improve both the scale and the hardness of the dataset. The key steps of the dataset construction consist of (1) a carefully designed crowdsourcing procedure, followed by (2) systematic bias reduction using a novel AfLite algorithm that generalizes human-detectable *word associations* to machine-detectable *embedding associations*. The best state-of-the-art methods on WinoGrande achieve 59.4 – 79.1%, which are ∼15-35% (absolute) below human performance of 94.0%, depending on the amount of the training data allowed (2% – 100% respectively).

Furthermore, we establish new state-of-the-art results on *five* related benchmarks — WSC (→ **90.1%**), DPR (→ **93.1%**), COPA(→ **90.6%**), KnowRef (→ **85.6%**), and Winogender (→ **97.1%**). These results have dual implications: on one hand, they demonstrate the effectiveness of WinoGrande when used as a resource for transfer learning. On the other hand, they raise a concern that we are likely to be overestimating the true capabilities of machine commonsense across all these benchmarks. We emphasize the importance of algorithmic bias reduction in existing and future benchmarks to mitigate such overestimation.

1 Introduction

The Winograd Schema Challenge (WSC) (Levesque, Davis, and Morgenstern 2011), proposed as an alternative to the Turing Test (Turing 1950), has been used as a benchmark for evaluating commonsense reasoning. WSC are designed to be pronoun resolution problems (see examples in Table 1) that are trivial for humans but hard for machines that merely rely on statistical patterns without true capabilities of commonsense reasoning. However, recent advances in neural language models have already reported around 90% accuracy on a variant of WSC dataset.[1] This raises an important question:

Have neural language models successfully acquired commonsense or are we overestimating the true capabilities of machine commonsense?

This question about the potential overestimation leads to another crucial question regarding potential unwanted biases that the large-scale neural language models might be exploiting, essentially solving the problems *right*, but for *wrong* reasons. While WSC questions are expert-crafted, recent studies have shown that they are nevertheless prone to incidental biases. Trichelair et al. (2018) have reported *word-association* (13.5% of the cases, see Table 1 for examples) as well as other types of *dataset-specific* biases. While such biases and annotation artifacts are not apparent for individual instances, they get introduced in the dataset as problem authors subconsciously repeat similar problem-crafting strategies.

To investigate this question about the true estimation of the machine commonsense capabilities, we introduce **WinoGrande**, a new dataset with 44k problems that are inspired by the original design of WSC, but modified to improve both the scale and hardness of the problems. The key steps in WinoGrande construction consist of (1) a carefully designed crowdsourcing procedure, followed by (2) a novel algorithm AfLite that generalizes human-detectable biases based on *word* occurrences to machine-detectable biases based on *embedding* occurrences. The key motivation of our approach is that it is difficult for humans to write problems without accidentally inserting unwanted biases.

While humans find WinoGrande problems trivial with 94% accuracy, best state-of-the-art results, including those from RoBERTa (Liu et al. 2019) are considerably lower, ranging between 59.4% - 79.1% depending on the amount of training data provided (from 800 to 41k instances), which falls 15 - 35% (absolute) below the human-level performance.

[1] https://github.com/pytorch/fairseq/tree/master/examples/roberta. We note that this variant aggregates the original WSC, PDP (Morgenstern, Davis, and Ortiz 2016) and additional PDP-style examples, and recasts them into True/False binary problems.

Type 3: Implementation projects

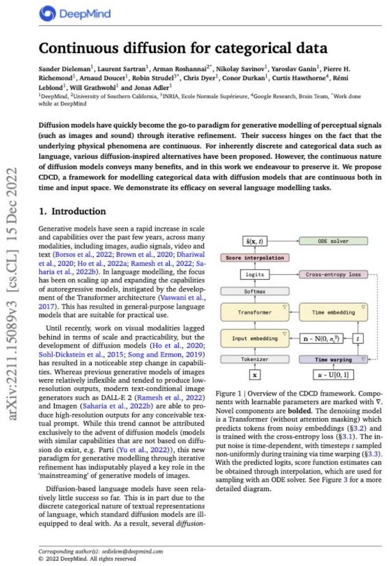
Examples:

- Compile a benchmark for a series of approaches, e.g. federate learning.
- Implement some non-trivial papers that does not provide their implementation, e.g. DeepMind's paper.

Your implementation needs to be non-trivial, clear and novel.

A concrete example

- “Continuous diffusion for categorical data” is an example of a conference-level paper of sufficient <u>complexity</u> and no code available that could be implemented. There is no existing code-base (like MinGPT) to build off of really.
- You would implement this and try to reproduce the results obtained by the authors (remember constrained compute)
- Your code should be well-documented and allow us to very easily run experiments to reproduce your results
- See high-quality paper repos for what to aim for



arXiv:2211.15089v3 [cs.CL] 15 Dec 2022

DeepMind

Continuous diffusion for categorical data

Sander Dieleman[1], Laurent Sartran[1], Arman Roshannai[2*], Nikolay Savinov[1], Yaroslav Ganin[1], Pierre H. Richemond[1], Arnaud Doucet[1], Robin Strudel[3*], Chris Dyer[1], Conor Durkan[1], Curtis Hawthorne[4], Rémi Leblond[1], Will Grathwohl[1] and Jonas Adler[1]
[1]DeepMind, [2]University of Southern California, [3]INRIA, Ecole Normale Supérieure, [4]Google Research, Brain Team, [*]Work done while at DeepMind

Diffusion models have quickly become the go-to paradigm for generative modelling of perceptual signals (such as images and sound) through iterative refinement. Their success hinges on the fact that the underlying physical phenomena are continuous. For inherently discrete and categorical data such as language, various diffusion-inspired alternatives have been proposed. However, the continuous nature of diffusion models conveys many benefits, and in this work we endeavour to preserve it. We propose CDCD, a framework for modelling categorical data with diffusion models that are continuous both in time and input space. We demonstrate its efficacy on several language modelling tasks.

1. Introduction

Generative models have seen a rapid increase in scale and capabilities over the past few years, across many modalities, including images, audio signals, video and text (Borsos et al., 2022; Brown et al., 2020; Dhariwal et al., 2020; Ho et al., 2022a; Ramesh et al., 2022; Saharia et al., 2022b). In language modelling, the focus has been on scaling up and expanding the capabilities of autoregressive models, instigated by the development of the Transformer architecture (Vaswani et al., 2017). This has resulted in general-purpose language models that are suitable for practical use.

Until recently, work on visual modalities lagged behind in terms of scale and practicability, but the development of diffusion models (Ho et al., 2020; Sohl-Dickstein et al., 2015; Song and Ermon, 2019) has resulted in a noticeable step change in capabilities. Whereas previous generative models of images were relatively inflexible and tended to produce low-resolution outputs, modern text-conditional image generators such as DALL-E 2 (Ramesh et al., 2022) and Imagen (Saharia et al., 2022b) are able to produce high-resolution outputs for any conceivable textual prompt. While this trend cannot be attributed exclusively to the advent of diffusion models (models with similar capabilities that are not based on diffusion do exist, e.g. Parti (Yu et al., 2022)), this new paradigm for generative modelling through iterative refinement has indisputably played a key role in the 'mainstreaming' of generative models of images.

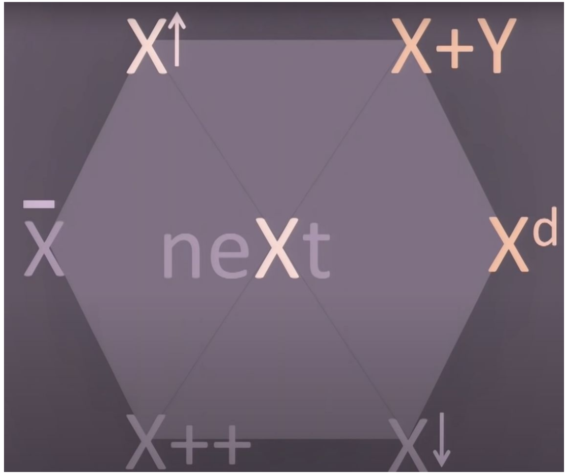
Diffusion-based language models have seen relatively little success so far. This is in part due to the discrete categorical nature of textual representations of language, which standard diffusion models are ill-equipped to deal with. As a result, several *diffusion-*

Figure 1 | Overview of the CDCD framework. Components with learnable parameters are marked with ∇. Novel components are **bolded**. The denoising model is a Transformer (without attention masking) which predicts tokens from noisy embeddings (§3.2) and is trained with the cross-entropy loss (§3.1). The input noise is time-dependent, with timesteps t sampled non-uniformly during training via time warping (§3.3). With the predicted logits, score function estimates can be obtained through interpolation, which are used for sampling with an ODE solver. See Figure 3 for a more detailed diagram.

Corresponding author(s): sedielem@deepmind.com

How do I come up with an idea?

- Take existing idea to new dimension (like Flickr to YouTube)
- Combine two existing ideas
- Given a method, apply it to new problems
- Given a solved problem, find other approaches to solving it
- Make existing idea faster, cheaper, more personalized, etc.
- Do exactly the opposite of existing idea



https://www.youtube.com/watch?v=fYnJPtEJj4s

Sections in a paper/your report (in general)

0. ~~Abstract~~ (not required)
1. Introduction (~1 page)
2. Related work (~0.75 page)
3. Problem formulation (for application or analysis project)
   Scope and structure of the implementation (for implementation project)
   (~1 page)
4. Methodology (1~2 page)
5. Results and discussion (~1 to 2 page)
6. Conclusion and future work (~0.25 page)

mandatory

Introduction

What you should consider when starting a project.

- What is the goal of the project?
- Why is the project important?
- Briefly introduce your idea or analysis
    - What is the problem you observed in existing methods?
    - What is your solution for the problem?
- Summarize the contribution

You can try to include these points in your proposal.

Related work

We require you to include 10 relevant papers.
(You should read at least 25 papers for a serious research.)

For each paper, you need to summarize

- its contribution
- how it compares with your idea

can be as brief as 2 sentences, depending on your space and the relevance of the paper.

Lewis et al. (2019) propose a question answering dataset that is similar in spirit to ours but covers fewer languages and is not parallel across all of them

Problem formulation (application and analysis projects)

- Describe the task
  - e.g. the input and output of the task
- The datasets and metric you use
  - describe why and what
  - don’t include implementation details, e.g. it’s a json file.

For implementation projects

Scope of your code

- Describe and explain the papers you implemented

Structure of your code, e.g.

- Abstraction of the problem
- Components in your implementation

Methodology (application and analysis projects)

- The ideas you have tried for this project.
- If you tried many, you can list the most important ones.
- Put the less important ones in the appendix.

Baseline (application project)

- Briefly describe your baseline.
- It needs to be reasonably strong.
  - i.e. the best method from relevant literature
- Sometimes you may have to make your own baseline.

Results

- Application project:
  - Compare your proposed idea(s) with the baseline.
- Analysis project:
  - Show the experimental results
  - Discuss the insights
- Implementation project:
  - Show the experimental results
  - Reproduce the results in the paper.
  - (or prove that the paper is wrong)

Conclusion

- Briefly summarize your contribution.
- Limitations of your methods.
- Discuss what could possibly improve your project.

Appendix

- Yes, you can have a long appendix.
- But we may or may not read it.
- You should make your main text self-contained.

But please remember to include the work distribution among your group members.

What if it does not work…

- It’s ok.
- But you need to show that you have tried reasonably hard.
  (You have 4 or 5 people!!!)
- Give us a possible reason for why it does not work.
- Consider based on what assumptions you thought you idea would work.

Reasonable high-level project ideas

- Take an existing conference-publication-level paper and try to extend it in a notable way
- Take an existing conference-publication-level paper and implement it for a totally new task
- Take an existing conference-publication-level paper with no available implementation and reproduce the paper results
- Rigorously explore a particular subject with a more theory-based approach (proofs, synthetic dataset experiments, etc.)

Unacceptable projects

- A project that is close to an existing tutorial on the internet
- A project that leverages an existing code base with minimal changes
- Opinion paper without experiments/results