# ConceptNet

ConceptNet is officially described as a semantic network with the aim of helping machines with the meaning of the words that people use.

## Suggested Improvements from the candidacy

* **Fine-grained relations:** Making them more specific, e.g., Above and Below instead of AtLocation.
* **New relations:** Adding new relations, such as NotIsA, CanEat.
* **Context:** Adding context to relations, e.g., finding desk in classroom is more probable than bar.
* **Updated:** Keeping the knowledge graph updated with new concepts, such as social media.

## ConceptNet Domains

### Types of Relations

The types of relations (34 total) in ConceptNet can be summarized as follows. The details can be seen [here](https://github.com/commonsense/conceptnet5/wiki/Relations).

* RelatedTo, FormOf, IsA, PartOf, HasA, UsedFor, CapableOf, AtLocation, Causes, HasSubevent, HasFirstSubevent, HasLastSubevent, HasPrerquisite, HasProperty, MotivatedByGoal, ObstructedBy, Desires, CreatedBy, Synonym, Antonym, DistinctFrom, DerivedFrom, SymbolOf, DefinedAs, MannerOf, LocatedNear, HasContext, SimilarTo, EtymologicallyRelatedTo, EtymologicallyDerivedFrom, CausesDesire, MadeOf, ReceivesAction. ExternalURL.

Among these 34 relationship types, we are more interested in the ones that are distinctly related to commonsense and not the ones, which can be found in a dictionary, such as synonyms.

## Examples of Issues with ConceptNet

Summary of the problems that we encountered with during using ConceptNet:

* One downside of the ConceptNet is its incompleteness and lack of some objects. For example, in the case of [AtLocation relations of Plate](https://conceptnet.io/c/en/plate?rel=/r/AtLocation), we can see Fork mentioned, but not spoon! Another example is the concept *vegetable*, which is missing from the plate relations. This incompleteness makes ConceptNet a poor dataset to compare against.
* Another downside are the weights of these relations. We can expect Knife, Spoon and Fork having similar weight, but the weights are 1.0, 0.0, and 2.83, respectively.
* The other problem is typos in ConceptNet, e.g., edible is dictated as [edable](https://conceptnet.io/c/en/usually_edable).
* Sometimes words are mentioned together in ConceptNet, e.g., [steak and eggs](https://conceptnet.io/c/en/steak_and_eggs) or [breakfast, lunch and dinner](https://conceptnet.io/c/en/breakfast_lunch_and_dinner).
* There are also some entities missing from ConceptNet, such as windshield, racket, surfboard, and logo.
* Some relations are missing, e.g., there is no indication that [the location of a train](https://conceptnet.io/c/en/train?rel=/r/AtLocation&limit=1000) is on train tracks.
* The relationships do not have a context.

### More Info.: Weight Problem in ConceptNet

Once we sorted triples from ConceptNet by their weights, one thing that we noticed was the bizarre appearance of only a few types of connections, e.g., RelatedTo. The incorrect weighting from crowdsourcing creates an issue in ConceptNet that an algorithm cannot find the best candidates to describe an object by simply looking at the top K relevant entities.

To elaborate by an example, given the entity ‘train’, the following two examples showcase the top weights for predicate ‘RelatedTo’ and ‘AtLocation’. As seen if we want to choose the top 10 entities that describe the entity train, they will probably all be entities connected by ‘RelatedTo’ predicate and possibly just one ‘AtLocation’ relationship!

A picture containing application

Description automatically generated Graphical user interface, application

Description automatically generated

### More Info.: Continuous Learning and Dynamic Concepts

To improve ConceptNet, an army of annotators should continuously work to keep up to date with new findings. However, an automatic method, such as WpKG or language models, has an advantage of automation over manual methods. Given upgraded sub-modules and continuous running on data, the information can keep flowing and by power of triple importance weighting, a concept can evolve overtime. For example, a chair in an older scenario could only be related to other furniture, while in a more modern scenario could also be related to computers, office and keyboard.

### More Info.: Faulty Disambiguation in ConceptNet

Sometimes there are entities that have multiple meanings but are not well disambiguated. For example, looking at most relevant entities connected to the concept train (/c/en/train), we can see some relations that are not quite relevant. Here is an example in which train means sports training and not really a train that goes on rails:

* [“/c/en/become\_scuba\_diver:/r/HasPrerequisite:/c/en/train”: 3.464](https://conceptnet.io/c/en/train?rel=/r/HasPrerequisite&limit=1000)

In comparison, a detection-based method such as WpKG, has the entities already disambiguated before aggregating them.

# Proposed Method

We propose using few-shot large language models to expand ConceptNet, as initially explained in our Symmetry paper and performed on a specific knowledge graph, called WpKG.

Priority of the updates (v0.1):

1. Adding new concepts, using the current relationship types.
2. Adding new types of relations (fine-grained or completely new).
3. Vetting the added triples and improving their accuracy as explained below.
4. Adding weights, either through large language model (LLM) or another specifically trained model.
5. Adding context to the triples, e.g., bottle on table is more probable in a bar than in a classroom.
6. Disambiguation of the existing and added concepts.
7. Fixing typos.

IMPORTANT: We need to show that the expanded ConceptNet is better than the original ConceptNet. This could be done in multiple ways:

* Upgrading the Numberbatch embedding, but it is old already – 2017.
* Finding a recent paper that uses ConceptNet for a task or benchmark and see if the expanded ConceptNet results in improvements in those tasks. The assumption is that the expanded ConceptNet is very similar to the original ConceptNet and makes it easy to integrate with a work that already uses ConceptNet.

## Vetting the added triples

As demonstrated in the Symmetry paper, we do not expect all the added triples to be true. In fact, around 30% could be incorrect. To alleviate this problem to some degree, we can follow the steps of the [West-2021](https://arxiv.org/abs/2110.07178) paper:

1. Annotate many triples using mTurk. Classified into correct and incorrect.
2. Train a smaller critic model, such as RoBERTa, on this annotated dataset.
3. Use the trained critic model, to filter out possibly low-quality triples.

* Repetition. If we see a fact multiple times, it more probable to be correct.
* If LLM know the meaning of words over time?

# Literature

## [Penguins Don’t Fly](https://aps.arxiv.org/abs/2205.11658)

### Summary

The paper introduces a new framework to construct and generate exemplars for generic rules, which can be instances or exceptions.

Based on previous language and philosophical research, the paper categorizes generic relations into three sub-categories. Based on these categories, the paper introduces different logical templates to generate instances and exceptions. The templates are then converted into prompts for language models. The output is filtered out and the viable results are selected.

Datasets: Datasets used for extracting generics are: GenGen-2022 (not released yet), [GenericsKB-2020](https://arxiv.org/abs/2005.00660), and ConceptNet-2017.

Models used: RoBERTa model used for all discriminators (filters). Train/dev/test is 80/10/10.

Baseline: GPT-3 with few-shot learning as in Appendix E.

A hair dryer is used to dry hair. But a hair dryer can also be used to dry clothes.

Annotations: Three annotators from mTurk. [MACE](https://aclanthology.org/N13-1132/) method used to filter annotators. Annotation needed to partition generics into three subcategories (653 generics). Fleiss kappa is also used to check amount of agreement among annotators. Viability filter also needs annotations (a set of 7,665 generations were annotated). Output selection also needs annotation.

Generation: For generation, a new method, called NeuroLogic, is used (from the same research group). NeuroLogic is an unsupervised decoding algorithm, which takes two inputs: a prompt and a set of lexical constraints. The output is a completion of the prompts, which has high likelihood give the prompt and has high satisfaction of the constraints.

Top-k chosen: The outputs are ranked by perplexity (fluency) and by the probability of a specific NLI label (for relevance). The two ranks are averaged. For NLI labels, the paper hypothesizes that a good exception aligns with NLI’s contradiction, and a good instantiation aligns well with the entailment. The top k generations are chosen.

Filtering: A discriminator model is trained to assess the viability of each generation.

Selection: Two other models are trained, for instantiations and exceptions. Their goal is to assess the validity (following exemplar template).

### More Details

“Commonsense knowledge bases, that are used extensively in many NLP tasks as a source of world-knowledge, can often encode generic knowledge but, by-design, cannot encode such exceptions.”

“In particular, edges in a CKB are assumed to represent generalities (i.e., they do not always need to be true), not universal statements. While this allows the representation of salient commonsense knowledge without exhaustive annotation (i.e., an open-world assumption (Re- iter, 1978b)), it also results in resources that are less informative for more specialized knowledge (e.g., “bird” has the CapableOf relation while “penguin” does not in Figure 1).”

“In this work, we present a novel computational framework for constructing and generating EXEMPLARS for a generic that incorporates various theories from semantics.”

## [NeuroLogic](https://arxiv.org/abs/2112.08726) Decoding

This paper introduces a method to do constrained text decoding. Asymptotic runtime is the same as beam search.

Datasets:

* CommonGen (Lin et al., 2020): Short description from a set of concepts.
* Evaluate Gender Bias in Machine Translation (Stanovsky et al., 2019)
* Recipe Generation (Kiddon et al., 2016)

## [I’m Not Mad](https://www.semanticscholar.org/paper/%E2%80%9CI%E2%80%99m-Not-Mad%E2%80%9D%3A-Commonsense-Implications-of-Negation-Jiang-Bosselut/ba40d5fa06a0b14aa50c681c0a38746e348a4491): Commonsense Implications of negation and contradiction

The paper claims to be the first comprehensive study, which focuses on commonsense implications of negated statements and contradictions. A new dataset is introduced, called ANION, which has 624K if-then rules, focusing on negated and contradictory events.

The paper claims that negated observations are rarely mentioned in commonsense knowledge datasets. For example, only 3% of ConceptNet covers negated examples.

Claim: “Training on examples of negated events leads to large improvements in the quality of generated inferences with minimal drop-off in the quality of inferences for affirmative events.”

## [It’s Not Rocket Science](https://www.semanticscholar.org/reader/9c3fde3474bfe58793f0a80811c4d20ec779ab4d)

Two types of figurative speech are analyzed: idioms and similes (comparisons). The paper has collected narratives containing figurative speech. Each narrative has plausible and implausible continuations relying on the correct interpretation of the expression. Then, models are trained to choose or generate the plausible continuations based on the narratives. Datasets have been built as well.

Questions It's Not Rocket Science:

* How is data collected?
  + Collected idioms/similes; Collected narratives for idioms/similes; Used MTurk to have plausible and implausible completions. Idioms: 3-4 workers. Similes: 10 annotators per narrative.
* Tasks?
  + One is discriminative. Given a narrative and two continuations, the goal is to choose the more plausible one.
  + The other is generative. Given narrative, generate plausible next sentence. A reference plausible continuation exists.
* How are models designed?
  + Generative:
    - Zeo-shot: GPT-2 XL and GPT-3.
    - Few-shot: GPT-3 with 4 training examples.
    - Supervised: GPT-2 XL, T5, and BART large. 5 epochs for idioms and 20 epochs for similes.
    - Knowledge-enhanced
* How is the evaluation done?
  + Automatic:
    - Generative:
      * Rouge-L (n-gram overlap) and BERT-Score (similarity based).
    - Discriminative:
      * ­Choosing the plausible continuation among two candidates. The plausible and implausible continuations were sourced using MTurk.
  + Human Evaluation:
    - Generative:
      * Amazon MTurk: Each judged by 3 workers and aggregated using majority voting. Krippendorff’s alpha was calculated, too.
        + Absolute evaluations: 50 sampled narratives for each task (simile and idiom). Asked if the generated continuations are plausible or not. Alongside human generation.
        + Comparative evaluations: sampled 100 narratives for idioms and 75 for similes. Shuffling narratives and asking which one is plausible, all or none.

Possible Contribution: Does upgrading ConceptNet with GPT-3 help in case of knowledge-enhanced models? How do knowledge-enhanced models work? Look at the [code](https://github.com/tuhinjubcse/FigurativeNarrativeBenchmark). The analysis part of the paper says:

“The knowledge-enhanced models provide various types of inferences corresponding to different relations in ConceptNet and ATOMIC. We are interested in understanding the source of improvements from the knowledge-enhanced models over the supervised baseline, by identifying the relations that were more helpful than others.”

## [CommonsenseQA 2.0](https://www.semanticscholar.org/paper/CommonsenseQA-2.0%3A-Exposing-the-Limits-of-AI-Talmor-Yoran/d65a064eb837f838faf6ff67781b62450b92b159)

Gamification method to generate new dataset to better challenge models. The players’ goal is to compose questions that mislead a rival AI. 14,343 yes/no questions. Few-shot GPT-3: 52.9%. T-5-based UNICORN (11B parameters): 70.2%; Human: 94.1%.

Problem: Models that perform well on datasets are brittle with out-of-domain and adversarial examples.

Solution: Gamification as a framework for data creation.

New Dataset: <https://allenai.github.io/csqa2/>

ConceptNet: ConceptNet is used for topic prompts. 1,875 high-ranking concepts in ConceptNet are used.

## [Analyzing Commonsense Emergence in Few-shot Knowledge Models](https://www.semanticscholar.org/paper/Analyzing-Commonsense-Emergence-in-Few-shot-Models-Da-Bras/8bdba45e46471ce23ac2dde24c849623997daaa7)

“Our results show that commonsense knowledge models can rapidly adapt from limited examples, indicating that KG fine-tuning serves to learn an interface to encoded knowledge learned during pretraining.”

Code: <https://github.com/allenai/few-shot-comet/>

ConceptNet: “Indeed, commonsense knowledge models [Bosselut et al., 2019] — which finetune pretrained LMs on examples from commonsense knowledge graphs [Speer et al., 2017, Sap et al., 2019a, Jiang et al., 2021] — learn to express declarative commonsense relationships much more effectively.”

“Furthermore, we observe that larger knowledge models exhibit most of their parameter change across a more concentrated set of parameters, implying they learn knowledge during pretraining in a less entangled manner due to their capacity.”

“Finally, we find that using natural language prompts to represent relation inputs accelerates commonsense emergence.”

**Training:** Triples from ATOMIC-2020 are translated into natural language using templates. n, such as n=3, examples per relation are used per relation. Loss method is negative log-likelihood of the tokens of the tail entity for each triple. AdaFactor optimizer with constant. Minibatch size of 4 and trained for 3 epochs. T5-11B model used for the experiments. For hyperparameter tuning, five sets of hyperparameters are used. Validation size is equivalent to the training size (n examples per relation).

**Evaluation:**

* **Human:** “We ask annotators to label the plausibility generated tuples using a [4-point Likert scale](https://chartexpo.com/blog/4-point-likert-scale): {always/often true (+2), sometimes/likely true (+1), false/untrue (−1), nonsensical (−2)}.” Three annotations per relation. +1 and +2 are acceptable. Majority vote used as acceptability. Fleiss’s kappa used for evaluation’s agreement measure.
* **Automatic:** BLEU-1, METEOR, ROUGE-L, CIDEr.

For few-shot experiments, average performance across 5 runs with different training sets is reported.

**Question asked:** Do few-shot knowledge models learn?

“**Few-shot augmentation baselines** are not fine-tuned, but receive examples using the same prompt formats as the finetuning baselines”

“We find that both the few-shot augmentation and few-shot learning settings are able to produce high quality commonsense knowledge tuples, indicating that large LMs are also efficient few-shot learners (in addition to showcasing impressive few-shot prompting abilities).”

“Using only n = 3 tuples per relation, both GPT-3 and COMeT (T5) produce high-quality tuples that are accepted as plausible by human evaluators more than 70% of the time — 75.7% for COMeT (T5).”

**How do models learn?**

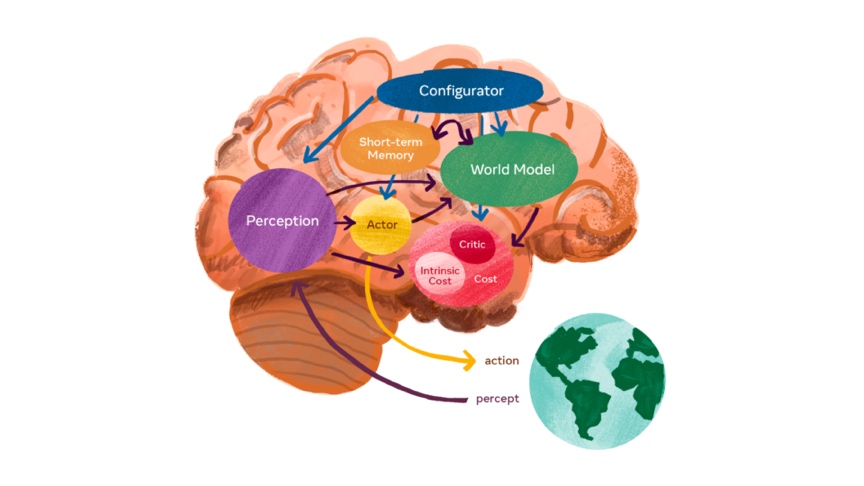
“We define three measures to investigate parameter change: **absolute parameter change**, **angular parameter change**, and **distribution of parameter change**.”

## [COMET-ATOMIC-2020](https://ojs.aaai.org/index.php/AAAI/article/view/16792): On Symbolic and Neural Commonsense Knowledge Graphs

**Idea:** KGs are incomplete and limited. SoTA language models are said to be struggling with implicit commonsense (Mostly pre-GPT-3). The paper’s solution: “To overcome this limitation, Bosselut et al. (2019) take the best of both worlds between commonsense knowledge graphs and pretrained language models. The commonsense transformer, or COMET, adapts pretrained neural language models by training on example tuples from commonsense knowledge graphs. It takes a head/source phrase and a relation (e.g., take a nap Causes) and generates the tail/target phrase (e.g., have energy). Bosselut et al. (2019) show that COMET trained on the CONCEPTNET and ATOMIC knowledge graphs is able to adapt to generate novel (and valid) commonsense knowledge tuples.”

# Other Commonsense Literature

## [Yann LeCun’s position paper](https://openreview.net/forum?id=BZ5a1r-kVsf)



* “The world model module constitutes the most complex piece of the architecture. Its role is twofold: (1) to estimate missing information about the state of the world not provided by perception, and (2) to predict plausible future states of the world. The world model may predict natural evolutions of the world or predict future world states resulting from a sequence of actions proposed by the actor module. The world model is a kind of simulator of the part of the world relevant to the task at hand. Since the world is full of uncertainty, the model must be able to represent multiple possible predictions. A driver approaching an intersection may slow down in case another car approaching the intersection doesn’t stop at the stop sign.”
* “The cost module computes a single scalar output that predicts the level of discomfort of the agent. It is composed of two submodules: the intrinsic cost, which is hard-wired and immutable (not trainable), and computes the immediate discomfort (such as damage to the agent, violation of hard-coded behavioral constraints, etc.), and the critic, which is a trainable module that predicts future values of the intrinsic cost. The ultimate goal of the agent is to minimize the intrinsic cost over the long run. “This is where basic behavioral drives and intrinsic motivations reside,” LeCun says. So it will factor in intrinsic costs, such as not wasting energy, as well as costs specific to the task at hand. “Because the cost module is differentiable, the gradient of the cost can be back-propagated through the other modules for planning, reasoning, or learning.””

### World Model Architecture

* “The centerpiece of the architecture is the predictive world model. A critical challenge with constructing it is how to enable it to represent multiple plausible predictions. The real world is not entirely predictable: There are many possible ways a particular situation can evolve, and there are many details of a situation that are irrelevant to the task at hand. I may need to anticipate what cars around me are going to do while I drive, but I don’t need to predict the detailed position of individual leaves in the trees that are near the road. How can a world model learn abstract representations of the world so that important details are preserved, irrelevant details are ignored, and predictions can be performed in the space of abstract representations?”
* “Human and non-human animals seem able to learn enormous amounts of background knowledge about how the world works through observation and through an incomprehensibly small amount of interactions in a task-independent, unsupervised way. It can be hypothesized that this accumulated knowledge may constitute the basis for what is often called common sense.”
* “Common sense can be seen as a collection of models of the world that can tell an agent what is likely, what is plausible, and what is impossible. Using such world models, animals can learn new skills with very few trials. They can predict the consequences of their actions, they can reason, plan, explore, and imagine new solutions to problems. Importantly, they can also avoid making dangerous mistakes when facing an unknown situation.”
* “The idea that humans, animals, and intelligent systems use world models goes back a long time in psychology (Craik, 1943). The use of forward models that predict the next state of the world as a function of the current state and the action being considered has been standard procedure in optimal control since the 1950s (Bryson and Ho, 1969) and bears the name model-predictive control. The use of differentiable world models in reinforcement learning has long been neglected but is making a comeback (see for example (Levine, 2021))”
* “Common sense knowledge does not just allow animals to predict future outcomes, but also to fill in missing information, whether temporally or spatially. It allows them to produce interpretations of percepts that are consistent with common sense. When faced with an ambiguous percept, common sense allows animals to dismiss interpretations that are not consistent with their internal world model, and to pay special attention as it may indicate a dangerous situation and an opportunity for learning a refined world model.”
* “I submit that devising learning paradigms and architectures that would allow machines to learn world models in an unsupervised (or self-supervised) fashion, and to use those models to predict, to reason, and to plan is one of the main challenges of AI and ML today. One major technical hurdle is how to devise trainable world models that can deal with complex uncertainty in the predictions.”

## [Is Reinforcement Learning (Not) for Natural Language Processing?](https://arxiv.org/abs/2210.01241)

[Project Link](https://rl4lms.apps.allenai.org/)

“The RL4LMs project attempts to alleviate these pitfalls by:

(1) providing guidelines for when RL should be used and what kinds of current NLP tasks and metrics are best suited for it in the form of a new ever-evolving benchmark dubbed GRUE (General Reinforced-language Understanding Evaluation).

(2) demonstrating how to use RL for language via a novel RL algorithm NLPO (Natural Language Policy Optimization) created to be more stable and less susceptible to both large language action spaces and high variance in rewards.

(3) A practical day-to-day guide including high-quality implementations and hyperparameters of NLPO along with multiple existing online RL algorithms such as PPO and A2C to train any causal or seq2seq transformer in the popular HuggingFace library.”

## [Do Androids Laugh at Electric Sheep? Humor "Understanding" Benchmarks](https://arxiv.org/abs/2209.06293)

“We challenge AI models to “demonstrate understanding” of the sophisticated multimodal

humor of The New Yorker Caption Contest. Concretely, we develop three carefully circumscribed tasks.”

“We identify performance gaps between high-quality machine learning models (e.g., a fine-tuned, 175B parameter language model) and humans. We publicly release our corpora including annotations describing the image’s locations/entities, what’s unusual about the scene, and an explanation of the joke.”

Text

Description automatically generated

# Ideas

## Knowing what you do not know

The language models are fantastic at different tasks, but do they know what they do not know? If they do not, how can we augment them to understand what they do not know? Then, empower them with extra knowledge, e.g., a complementary knowledge graph of things that can later be used to upgrade a language model, using fine-tuning or other methods. This also improves the explainability of a language model as it makes it clear what things the model knows or does not! Basically, adding a level of self-awareness regarding mistakes to language models. This is because the common theme between different papers is that the model does not know this, so let’s create a training dataset for that reason, then fine-tune the model to get better results. If we could automate this process, we will have a self-evolving language model that can find its own shortcomings and improve on them.

Based on the literature, there are different things that language models lack in commonsense. Can we devise a generic framework for language models to improve themselves and find this missing information in multiple steps: (1) finding that it does not know.; (2) try to find the answer using web, etc., similar to [Binder](https://lm-code-binder.github.io/) and [iPython generation](https://twitter.com/goodside/status/1581337959856422912?s=20&t=EyYFPinD_68Y3kUouJo5Cw); (3) store in a knowledge graph; (4) use the KG to improve the model.

There are two types of self-knowledge about missing knowledge that we can think of:

* Based on analyzing a knowledge graph or dataset, and saying some parts are missing. For example, birthplace of a person.
* Based on a benchmark, evaluating how many of the incorrect answers the model knows are incorrect or how many it cannot know to answer beforehand.
  + Incorrect results: What percentage can the model self-assess that answered wrong?
    - Do we really need human annotations on what the correct result is?
  + Incorrect results: What percentage the model knew it could not answer beforehand? Before even trying to answer.
  + Correct answers: Same two assessments as in incorrect answers.
  + Does the LM know how it can obtain the correct answer? Which external knowledge to use to augment itself?
  + Another traditional way of knowing what the model does not know is through benchmarks.

Here are some benchmarks and tasks that we can analyze on, or the language model has wrong answers in:

* **CommonsenseQA** (Talmor et al., 2019) asks commonsense questions about the world involving complex semantics that often require prior knowledge.
* **StrategyQA** (Geva et al., 2021) requires models to infer a multi-hop strategy to answer questions.
* We choose two specialized evaluation sets from the BIG-bench effort (BIG-bench collaboration, 2021): **Date** Understanding, which involves inferring a date from a given context, and **Sports** Understanding, which involves determining whether a sentence relating to sports is plausible or implausible.
* **SayCan** dataset (Ahn et al.,2022) involves mapping a natural language instruction to a sequence of robot actions from a discrete set.
* **AI2 Reasoning Challenge (ARC)** (Clark et al., 2018)
* [Penguins Don’t Fly](https://aps.arxiv.org/abs/2205.11658): exemplars for generic relations, that could be instances or exceptions.
* [I’m Not Mad](https://www.semanticscholar.org/paper/%E2%80%9CI%E2%80%99m-Not-Mad%E2%80%9D%3A-Commonsense-Implications-of-Negation-Jiang-Bosselut/ba40d5fa06a0b14aa50c681c0a38746e348a4491): curated negated and contradiction statements and training on them.
* COMET-ATOMIC: Curated commonsense statements and training on them.