Masters Thesis

Information:

- Working with the Models branch, under professor **Buchs** and three PHD students (Damien, Dimitri, and Aurélien).
- Projected timeline is September to February (*est. 6 months*).
- Tentatively weekly/bi-weekly meetings where I present a short summary of the work that I have done and will do.

First Checkpoint - Scouting

Goal: We want a clear, concise, statement that represents the project that we will pursue. This is expected to be packaged as a research question, and we will have some considerations for producing vs acquiring data in the project.

Note: That the team I'm working under is specialized in Modeling and Verification techniques, and are not experts at process mining, and thus, we need to keep that in mind in terms of the direction we take the project.

Potentials to Explore:

- Content Mapping using Rewrite Rules (essentially universal curriculum creation that maps obtained data into more structured formats with practice problems, etc)
- Translation Department, how do they learn? What is their structure, etc?
- Keeping in mind that we are looking more at User Actions rather than a focus on content (in the sense that we don't necessarily care about the content, just what we DO with the content).

Scouting

Preliminary scouting, looking for keywords: "rewrite rules", "data mapping", "user action", "user process", in context of language learning/e-learning.

"Equivalence of Dataflow Graphs via Rewrite Rules Using a Graph-to-Sequence Neural Model"

- States that "The problem of program equivalence is summarized as determining whether two programs would always produce the same output for all possible outputs." Which if we're thinking about in the context of language learning, might be useful in terms of determining whether or not

two sample problems/curriculums could be considered equivalent, using looser terms?
- The paper then continues and talks about how it uses machine learning model to produce rewrite rules in order to then deterministically check equivalence. We can think about using this type of hybrid approach?
- The idea is that it generates a graph where by rewrite rules can be applied, and then you verify via checking the existence of the path, and check whether the end results are equivalent.
- The contribution is the deep learning model that is able to generate this graph.

"FunMap: Efficient Execution of Functional Mappings for Knowledge Graph Creation"
- Focused on mapping language that reduces the execution times of knowledge graph creation via lossless rewrite rules.
- Aimed at reducing execution in knowledge graphs that contain lots of duplicates.

"Few-Shot Generative Conversational Query Rewriting"
- Shows the importance of query rewriting to the field of info retrieval and other conversational assistance tasks.
- Meaning that there is something to be said about rewriting and verification methods being useful.

"Few-Shot Natural Language Generation by Rewriting Templates"
- Demonstrates a more concrete example of what we mean when we say that we can use ML to do generation of things that could be useful to us.
- Additionally it involves the notion of rewriting, although not the same thing as the rewriting we're necessarily talking about (however, I'm sure from the previous paper on graph to sequence models, that even those lines can be blurred).

"Question Rewriting for Conversational Question Answering"
- Targets the task of conversational assistance via question answering, but tries to improve the performance through some pre-processing of rewrites.
- Claims that "[...] composite architecture allows us to trace errors back to the individual components [...] either due to the incorrect context interpretation or the incorrect question-answer matching." which I presume means that they have a built in way to indicate which one was at fault.

- The setup itself has the original question, the automated rewrites (produced by a ML model), as well as human annotated rewrites. This way they can check whether or not the rewrites are actually of good quality. They go through and test the possible cases where the machine was able to yield a proper answer to the question (ground-truth).
- I'm presuming that if we're working with curriculum generation or some sort of language acquisition tool, that this element/setup might be something useful that we will have to examine.

"Topic Propagation in Conversational Search"

- Another paper that works on rewriting the questions in a conversational assistance context, this time focused on information retrieval.
- It specifically takes note to topic switches, since the conversational context that the paper works with is multi-turn in nature.
- This suggests to me that the techniques used here would show the basis that there's techniques to try in context of multi-turn conversations, or even in the case that there are multiple (interacting) speakers.
- This is also the third (?) paper to mention the Conversational Assistant Track (CAsT) dataset, and might be something that we can consider tapping into as a dataset for our usage (even if it is monolingual and not intended for the purpose of language acquisition).

"Unsupervised Translation of Programming Languages"

- About using ML sequence to sequence model to create translations of programming languages.
- Very straightforward application of seq2seq unsupervised model for a translation task.
- However, given our readings from other papers, this would probably benefit from an element of rewrites that would fix some of the smaller issues that it has.
- From our perspective, we need to really think and address the question of what really is the goal here, and how do we achieve it in a straightforward and intuitive way?

Second Checkpoint - LE Considerations

Goal: We want a clear concise project statement, that outlines the tasks that we want to accomplish, and why this approach is applied.

Need to address:

- What do we need in LE/Language acquisition?
- In a way asking, how do we create more opportunities to use the target-language?
- What are the things (no matter how small or trivial) that hinders/delays the process of learning?

What do we think we need in LE?

- More relevant vocab
- More comfortable pacing
- A way to remove the double task of socializing + language learning, aka, remove the barrier of social anxiety

What do we think we need in language acquisition (self study)?

- More relevant vocab (always)
- Understanding modes of expression rather than translation, aka, remove the barrier of thinking in the source language
- Content**

Vocab:

- We need to be mindful of the context of the vocab (i.e: words that have a literal meaning, but the context/phrase it is used in alters)
- We need to consider the relevance of the vocab (i.e: if i see vocab that are outdated/too formal again i'm gonna lose my mind)
- We need to think about usability of vocab (most of the vocab i lose since i don't really have the need to use it everyday, also consider here the ease of creating a system that allows for constant vocab usage without feeling like a chore)

Pacing:

- In language exchange, which is in real-time, pacing is important because when the pace is too fast, acquisition becomes difficult if not impossible. However, if the pacing is too slow, sometimes acquisition suffers, but it more often creates a case where the social anxiety spikes because the conditions don't feel as natural.
- Even artificial pacing could avoid the spike in social anxiety, think: language classes where forced activities seem less nerve-wrecking. Having a prompt sometimes aids a lot with social anxiety (prompts always gives us a way forward)

- Automation and progression always makes for good and efficient acquisition. We can also think about pacing in the macro sense, i.e: how much learning are we doing each day as a pace. We don't want the learning to proceed too fast (will get dumped) and we don't want the studies to proceed too slow (left behind and not constantly forgotten)

Removing the Double-Task:

- More an efficiency question, since we want the user to be focused on language learning, not worried about the social conventions/interactions with another person.
- This has big connections to pacing, as improving the pacing even artificially can help with removing the double-task.

Understanding modes of expression:

- Different languages express the same concepts sometimes in different ways. Even minute differences will create situations where expressing a certain idea will often times not use the same words, and capturing the same feelings/expressive power is very difficult (something for high level interpreters)
- Finding a way to bridge the gap of expression is very important. We don't have to create an environment where we try to "correct" those other forms of expressions, but rather, create the necessary scaffolds so that those other modes of expressions can be understood/begin to be understood.
- This is where I feel like the ML models of trying to understand intent can probably get to the same level as a human being trying to understand a nuanced expression of thought in a non-native language. I.e: I think a ML model will guess better (knowing my native-language is english) what I mean to express compared to someone who is looking to correct the statement for more native-language accuracy.

Content:

- Varied enough topic-wise to cover different users
- Varied enough topic-wise for a single user to diversify
- Level must match OR have an effective UI to get necessary info
- Varied enough length-wise to cater to different contexts
- Varied enough engagement-wise to support long term studies and also short term sporadic engagements
- Organized enough so finding new content is easy

How do we create more opportunities?

- Can we artificially create relevant opportunities to practice language in a LE setting? I think there is room for that (prompts, generated tasks, etc)
- Can we help push along the naturally occurring opportunities of exchange by removing some of the common barriers? Probably, such things like providing vocab, or even voice recognition of vocab words/tablet to offer search of vocab (i.e: vocab focused?)
- Can we artificially create an entire sequence of engagement/exchange? I.e: one person exchange with a script/curriculum? This one seems harder, but not entirely infeasible depending on level.
- Can we simulate an opportunity after we encounter it once, in order to use it constantly as a source of learning/instruction? (This seems like an interesting idtcea). Has potential for variations based on rewrites and possible model interactions?

Hindrances to learning:

- Lack of constant learning/usage (not enough passive or active)
- Lack of constant engagement by the user (not enough active)
- Lack of structure (not enough feedback)

Tools we have at our disposal:

- Machine Learning models (seq2seq, etc)
- Re-write rules, system verification
- Process-mining libraries, user-actions
- Text-processing libraries

Is it enough? Where could we potentially apply these? Are there other peripherals that we need? (i.e: in the case of requiring voice recognition, etc)

Third Checkpoint - Decision

"Rewrite rules and mapping as a tool"

- Rewrite rules in graph generation, except the graphs are aimed at conversations, with the goal of rewrites optimizing graph generation to create the same conversation, but with slight changes (vocabulary, language?)
- Rewrite rules in graph generation, except the graphs are aimed at large text data (books, textbooks, etc), with the goal of rewrites optimizing graph

generation to create the same curriculum, but with slight changes (patchwork from different resources, etc?)
- This concept is probably doable due to the paper that utilizes rewrite rules to verify that ML generated graphs are equivalent, which improves the process of graph generation.
- I guess the problem here is that with program equivalence, it is on the ground-truth level a yes/no question, whereas with something like conversations and curriculums it's not necessarily a yes/no. Although, I feel like in that case we can add on top another level of verification?

"Process Discovery for Structured Program Synthesis" also provides some support for the feasibility of process discovery and generation of graphs through rewrite rules.

Summary: I want to <u>apply rewrite rules in order to check two graphs</u> for equivalence, <u>where those two graphs are representations of a conversation</u>/curriculum that we can map different content to, <u>with the express purpose of creating a system that effectively generates new scenarios/resources</u> for language learning.
Specifically, those <u>one of those two graphs can be generated via a ML model</u>.

- Get feedback on whether or not that project goal is reasonable

Next steps are:
- Research on a codebase to do the project in (some OpenNMT-py library, model checking recommendations? Some way to port in between or do everything in one language? We could look at C++ again…)
- Familiarize with the tools (graph generation, rewrite rules, model checking)
- Demonstrate familiarity with tools -> Proof of concept with tasks that will benefit the project (i.e: graph generation, verify a generated graph, etc)

Estimate: 10 days

Fourth Checkpoint - Precision
~~9 Days until thesis report #1 due~~
~~8 Days until thesis report #1 due *(Sunday 06 Sep)~~
- ~~Scout more papers in the field~~

7 Days until thesis report #1 due
- Make a commitment

    - Choose a thing to map (text -> resource (curriculums), convo topics/exchanges).
    - Choose a method (rewrites, graph generations, etc) and also have presentation notes on these methods.

Paper Notes:

1. “Extensions to Justification Theory”
   - Justification Theory is a framework
   - Sort of like a knowledge-graph, except explains truth value of facts
   - *Branch Evaluation* -> term meaning mapping many facts to a single fact (true, false, unknown).
   - Paper applies it strictly for the legal context (mapping conclusion with proper justifications)
   - For us, we can think about fuzzy justification-theory, which may sound like it breaks the whole point of justification theory, but rather, we are really just expanding the strictness a bit to encompass useful flexibility for language learning contexts. For instance, there are many ways to express a certain idea. We can map the many different ways to express an idea/concept to justifications (base vocab, grammar points, etc, that really make up the phrases in general, making it possible, ergo “justifying” it.)

2. “LogicalFactChecker: Leveraging Logical Operations for Fact Checking with Graph Module Network”
   - Uses table and statement input, in order to generate a graph that relates the inputs, and derives a conclusion (fact checking)
   - We can take this concept and apply it in a different direction, can we generate the table as the output? Given the proper input of a statement and a graph that we want? This would mean that we would be able to generate the proper context vocab needed to achieve the results that we want.
   - ML based, and not sure where our models tech would fit in, but good to keep this info in the pocket

3. “Model-Checking Quantitative Alternating-time Temporal Logic on One-counter Game Models”
   - ATL is used to specify temporal properties of systems with several interacting entities
   - (Above is really good for us, since that’s exactly the system that we’re looking to replicate)

- The difficulty is verifying satisfiability given the uncertainty in having several interacting entities (can have halting problems)
- Energy and Energy-Parity games -> game where you must maintain a certain level of energy, and maintain energy while satisfying a parity condition
- (Above, can we have a situation where parity is something ambiguous like "understandable/comprehensible"? Or do we not even have to worry about in the models perspective, we just have something external to determine the state of that parity?)
- Seems that the rest of the paper has very detailed definitions of games and graphs, meaning that we can use this to better understand how we would define our own terms (mapping the concept of a language exchange for instance). This paper thus deserves a flag.

4. "DAS3H: Modeling Student Learning and Forgetting for Optimally Scheduling Distributed Practice of Skills"
   - Useful for looking at the dataset that they discuss and test on.
   - Paper directly talks about and discusses learning and learning concepts, which are useful and relevant to us.
   - However, this is a purely ML-based approach, and is about giving student feedback, we're not working on modeling at any level.
   - Despite that, consider this paper still relevant, since it is topical.

5. "Abusive Language Detection in Online Conversations by Combining Content- and Graph-based Features"
   - Mentions the concept of the "conversation graph" as well as characteristics derived from those structures.
   - Context is in the form of preventing abusive language, but the overall content can be changed -> we can find any type of language as long as we can define it, meaning that this methodology is still super important to us.
   - The paper also outlines (form wise, not content) why a ML-centric approach will not solve the issue (depends on training data, which in the given form, will not be robust enough for the content)
   - The paper still focuses lots on ML methodology, but the part about conversational graphs is still useful form for us, even if it is just partial to their paper

6. "Language-Conditioned Graph Networks for Relational Reasoning"
   - Still in the form of a graph, but also presumably ML-centric
   - Still relevant, but stashed most likely

7. “Language-Constraint Reachability Learning in Probabilistic Graphs”
   - Discusses a ML-approach to reachability in probabilistic graphs
   - Useful for us when we’re discussing how our structure will turn out

8. “Fuzzy Petri Nets for Human Behavior Verification and Validation”
   - Useful for us when we’re discussing how our structure will turn out

Considerations for Mapping:
“Are we doing text-> resources for curriculum creation, or convo topics/exchanges -> graphs for verification?” How do we formally represent the information and the structure of our inputs and outputs, so that everything is PRECISE and clear (input and output wise), and thus also goal wise.

Ideas:
- ~~Represent a conversational exchange as a graph:~~
    - ~~Probabilistic~~
    - ~~Nodes are a “speech act” (utterance, sentence, phrase, etc)~~
    - ~~Edges are a “relationship”/”links” (these speech acts are related: followed by, preceded by, equivalent to, etc)~~
    - ~~Applying logics of ATL/QATL through the links, nodes can be “~~
- Represent a conversational exchange as a graph:
    - Start and End node, probabilistic
    - Nodes are states in the conversation (arbitrary breaks between exchange?)
    - Links are a “speech act” (utterance, sentence, phrase, etc)
    - Constraints are vocabulary known at each state
    - Checks for liveness -> checks for reachability -> checks for completion of the graph (start to end node reached)
    - Allows for us to tack on the task of “create a model that generates a user that is able to satisfy completion, or keep liveliness, or reachability to a certain state in the graph”

Do we use a Petri-Net here? Or is another graph structure easier for what we want to achieve (since we may want to be able to use the graph as an input to a ML-model later on?) How do we create constraints for link if we don’t use the petri-net model?

- I guess the paper "Model-checking QATL on one-counter game models" shows us a way to represent a graph as a one-counter game via links having weights.
- I guess this opens us up for our own contribution. What is the most logical way for us to represent this in order to achieve our goal? This is kind of the papers we need to explore: graph games, temporal logics, model-checking

Graph-Games:

- Parity-game (one or two counter)
- Energy-game (infinite or finite state)

Papers are already out with proofs on whether or not certain games and problems are decidable, and currently this information is above our heads. This is definitely an area of scouting required to set up a state of the art to understand the problem we want to solve is solvable or not (to begin with) before we even attempt to model and solve it.

Considerations for Parity Games:
Which means that our current scouting should be directed towards the different forms of games that can be run via graphs, in order to find out the best way to represent our "language exchange game" and begin defining the nuances.

Parity Game Paper Notes:

- Parity game is described as a graph game (nodes and edges), between two players, whereby each node is designated a P1 or P2 node, and at those nodes, the appropriate player chooses the next node. Additional conditions may apply.
- Formal definitions of these games and graphs are like what we've seen before through sets and mappings.
- Keep in mind the notions of perfect/partial observation (meaning access to the graph)
- "Games with a Weak Adversary" paper states that games that are partial observation, with multiple (two) players, is not decidable for ensuring reachability or safety. However, if the players form a chain of observation, i.e: P1 > P2 in terms of information on the graph, then the problem becomes decidable.
- The same paper also discusses how new decidability results are obtained for partial-observation stochastic games where P2 > P1 (knowledge).

- For these graph games so far, we've only seen cases where the transitions are not probabilistic, and decisions for the successors are made by the corresponding player on each node. If we map this notion onto a conversation, this obviously wouldn't match since in a given conversation, we're not 100% sure if what we want to say is received the way we want to say it (i.e: probabilistically speaking, successors are not deterministic). Keep an eye out for probabilistic graphs, and discussions on whether or not this is even possible (could be what we look into for our thesis?)
- Set-based symbolic algorithms paper states that "all w-regular winning conditions (such as safety, reachability, liveness, fairness) as well as LTL winning conditions can be translated to parity conditions. Which is an interesting fact to keep in mind since it means that we have some way to describe and formalize the abstract things we're thinking about in terms of "keeping a conversation going", i.e: maintaining parity
- Furthermore the same paper discusses explicit vs symbolic algorithms. Where the definition is that explicit algorithms act on the explicit representation of the graph. Versus the symbolic, where we only work on a predefined set of operations and not on the graph itself. This is important because for big graphs, or even infinite graphs, the symbolic algorithms are the only viable option, meaning that symbolic algorithms are scalable (to any system). However, for our case, that doesn't mean that we can't find some use in explicit algorithms, perhaps there's another angle, we'd have to consider.
- Now, set-based symbolic algorithms are symbolic algorithms that are based on set-operations, such as union, intersect, complement, inclusion, as well as predecessor (pre). The "space" that they're talking about in terms of complexity, of course, are in reference to the sets that are stored by the algorithm.
- Note that parity games are history-free, meaning that a winning strategy is based on the current board position only. (Will this affect us, is this a variation that we might be interested in researching more/learning more about in order to apply to our context?)
- "Infinite state energy games" paper suggests that there are different games, and this tells us that probably parity game setup might not be the thing that we'll end up using, but of course, that information is all still very important context for us to understand our final approach to setting up language exchange in a graph-based game, in order to apply some sort of variation to it for modeling/verification.

- The paper "Parity Games, Imperfect Information and Structural Complexity" presents the notion that given the classic case of reducing imperfect information of one Player by tracking the uncertainty of that one Player (causing an exponential lower bounds for complexity), is to look at special cases where we can bound the uncertainty of that Player, and the example it gives is that the faulty sensor which grants uncertainty/or the private information of the system does not grow when the size of the system grows (however, in the basic case of language exchange, this won't be the case since increasing the number of exchanges intuitively requires more knowledge, increasing uncertainty? Yet, this still suggests to us that when we model this through the lens of LE, that there is special cases that are worth exploring, and perhaps this mapping could give us some insight on interesting things we can do with the system.)
- We know that we can check a graph to see properties/conditions, aka they are decidable. So the question becomes, "Can we generate a random parity-game (meeting some parameters) and know that it will likely be decidable?" Then, we can provide a mapping of a LE to the parity game, and knowing that our generated graph will be decidable, apply all the notions we know about validation and verification in order to provide benefits to the process of language learning!
- We should look into papers involving "reactive synthesis" or "LTL synthesis"
- When we're looking at these papers, notice how we handle cases of non-determinism (whether they are explored as a separate problem, whether they are reduced or transformed to deterministic aspects, etc)
- Also important to keep in mind that although for the super strict world of models and verification that some of these things might not work out perfectly, we're only looking to apply it in the case of LE, and any type of flexibility we have in terms of defining our system or specification to accommodate that goal is a shortcut that we're afforded to take.
- Strategy Improvement as a concept should also be something that could be useful, since the idea maps very well to the notion of "sticking to talking about what you know"
- The paper "solving random parity games in polynomial time" mentions that they look at the cases where the game is fair (relatively equal nodes and priorities) but suggest that biased cases can be further work. Since our LE context necessarily creates situations where there will be bias (most often, people are not of the same level in a LE), this will probably be something worth looking at for us.

- The same paper contributes that there is a threshold such that underneath this threshold, polynomial solutions are available. Additionally, show that non-sparse parity graphs are easier to solve (even though they have more strategies)
- Lossy Channel paper says that channel systems, communicating finite-state machines, make all verification questions undecidable. So the next question is why we would want to study it or how we can study it to get something out of it even though we know that verification is undecidable?
- Interestingly enough, the lossiness assumption (that messages sent may not be received or received correctly) makes safety/reachability and termination decidable (ah and there's the rub).
- This suggests to me that when we change the restrictions and context of the game, it can make things that are undecidable decidable and vice versa. Meaning that defining our context and being hyper-specific (re: LE model) will be very important to studying the resulting game.

Notes from Dimitri's Thesis:

The structure of his thesis is as follows:

- Introduction to the thesis
- State of the Art (which is a compilation of things that have been done in the field in relation to the problem the thesis wishes to resolve)
- The thesis method proper (introduction -> specifics -> limitations, broken up into different parts of the goal)
- Conclusion (contributions of the work, short future works section)
- References

*Dimitri's goal is outlined as: "to construct the reachability graph of a gamma-program, given this graph is finite and computable, to analyze its state space. Then we show how we can take advantage of this alternative representation to perform model checking and verify properties that would be hard to verify otherwise, such as invariants."

What we need to take note on is the fact that when we're being asked for precision (our current task for the next meeting), we're really being asked to develop the GOAL section of the thesis method proper, as well as putting it into context given the larger picture of the state of the art.

Organization of Next Meeting:
We'll set up the next meeting for hybrid flex between online/offline. We should aim to provide the following:
- Folder of relevant papers as well as outline information (maybe calendar?)
- Presentation Slides in PDF format

That way, we can either prep for an in-person meeting, as well as an online substitute.

Which means we'll need to consider which papers are relevant for our discussion, as well as how we want to construct the process via PPT slides. We'll need to consider our topic in a precise manner, re: graph games and applying rewrite rules? Mappings? We can first start by revisiting the ideas as well as the papers to get a better sense of how we can draw connections.

Basic Concepts:
- Graph Games (Parity, Energy, Automatas)
- Representing a conversation (Language Exchange) as a Graph
- Verification and their decidability: liveness, reachability, termination, etc
- Synthesis (of Graphs via rule-based approaches or ML approaches)
- Mapping (rule-based or ML)
- Process-centric approaches

Papers on Mapping/Process-centric Approaches
- Graph to sequence neural-network (proving equivalence of graphs)
- Process discovery via graph rewrite rules

Papers on Graph Games:
- Games with a weak adversary, forming chains of certainty -> decidable
- Set-based algorithm for parity games, outlining the concept of sets and how they function when thinking about translating LTL and other graph verification to be represented as parity games
- Parity games imperfect information, discussing how parity games complexities can work and how different setups are
- Solving stochastic parity games, gives us some info on working on random games (which we intuitively get from generated graphs)
- Lossy Channel, giving us a new model to think about, in the setup of the world of graph games

What we need to provide:

- Representation of Language Exchange
- Connecting the Graph game theory with the implementation of mapping/process-centric approach
- Outlining goals and give a timeline

Note: Let's start by really putting into words what kind of product I would want in an LE context that *personally* would be really nice to my learning. Then branch from there! Trying to go as specific as possible.

- I want to be provided *vocabulary* for words that are expected to pop up in the conversation so that I can reference them in real time and apply my own knowledge, effectively acting as practicing creating structures, rather than testing vocab. (separating the test of vocab and grammar in creation)
- I want to be provided some *expectation* of what the other person could say, so that I can *anticipate* the intent of the other, solving the issue of, "i knew all the words they said, but realized too late, or wasn't expecting it" as well as, even if i didn't know x word, I can guess from the context (i.e: in the situation of me saying: "the machine isn't working, there was no ticket -> La machine n'est marche pas, il n'y a pas de ticket", and the really quick response, I can expect the intent of the question to be, "what did you order?" as she checks the screen.)
- I want something to guide the conversation so that down time and awkward pauses are more minimal and have less impact, since creation (coming up with sentences) is already hard enough. But not something scripted, as I do want to practice proper creation.
- Or even, input some information/answer some questions, to then semi-accurately provide me content around my level (even easier in implementation, bring out the documents that are around my level).
- The question that is important to ask is also why I would even want any of this in the first place, can't they all be replaced by a classroom simulation? Or is the point that this alternative should serve to be able to replace the simulation, or augment the simulation to begin with?
- We would also want something that can structure/map genuine conversation in a language such that it can be studied/used as example conversation data/used as a scaffold for LE. (Role play?)
- In terms of role play, deviating from the script creates opportunities for creation and genuine conversation, so can we create systems of analysis to display/generate possible deviations/paths such that we can create opportunities for genuine exchange of conversation from a scaffolded situation? (role play with ad-libbing)

How to get relevant vocab:

- ML model -> word generation
- Graph links -> graph generation
- Pre-loaded -> form the connections after

Anticipation of conversation/Guided conversation:

- Process Discovery -> generation
- Map a conversation -> graph generation
- Topic change prediction -> ML model
- LTL Synthesis -> automata theory

Questions:

- Can having old conversations with new people cause the conversations to feel new?
    - What makes a conversation feel old/new?
    - What makes a person feel old/new?
- A conversation feels new when it contains something we have not heard before.
- A conversation feels new when responses deviate from your anticipation.
- A person feels new when you don't feel like you know what they are thinking, or why they think the way they do.
- A person feels new when you don't know some information about them.
- A person feels old when they say the same things constantly.
- A person feels old when you don't have to anticipate what they say.

- Does a LE actually need to be a new conversation for you to learn to acquire the target language?
    - Do we need conversations at all to acquire a new language?
    - Can conversations help you acquire the target language?
    - Can we make conversations help you in acquiring target languages?
- **Conversations give us a chance to create**, and creation helps acquire new languages.
- Conversations are just constantly creating sentences.
- Those **creations don't have to be 100% new**.
- Conversation doesn't have to be 100% new to be helpful in language acquisition.
- Creation can be partial, as long as the user feels they contributed something, they **had a part in the creation**, they will learn something.

- What defines a Language Exchange?
    - Does conversation have to be at the root of language exchange?
    - Is conversation the only defining trait of a language exchange?
    - Are there other forms of language exchange beyond communication between two people?
- Language Exchange is when two (or more) people share a conversation in a foreign language (foreign for one or both parties).
- Yes, language exchange is unique because it is based on the exchange between two (or more) parties, which is unattainable by other means because **we cannot replace the existence of another party**.

- What makes a Language Exchange successful?
- LE is successful if one party has **learned something** about the target language.
- LE is successful if one party has **gained practice** in conversing in the target language.
- LE is successful if the result of the LE **contributes as a resource** for learning (even if the learning is not done by either party).
- LE is successful if the exchange leaves either party in a **good spirit**.

- Can a LE offer more than just conversation?
- **Confidence** is important
- **More practice** in creation is important
- Additional experience facing situations of pressure/**"in the moment" feelings** are important
- Motivation by creating stakes is important
- **Interacting socially** with others is important
- Giving feedback/teaching is important
- Watching another conversation is important

- Can a language exchange be scripted?
    - How much structure can we script before it feels unnatural?
        - What makes something feel unnatural?
- A LE can be scripted as long as it leaves room for creation for both parties.
- A LE will start to feel unnatural if there is not enough room for creation from the parties.
- Lack of creation will cause feelings of unnaturalness, because **expression is personal and unique**.

- **There are commonalities in expression**, despite expression being personal and unique.
- A LE could start to feel unnatural if expression doesn't conform to the expectations of either party.
- **Too much variation and deviation could collapse structure**, which provides some **sense of comfort** in a social exchange.
- What is **unnatural for someone may be less unnatural for others**. A situation where both parties are comfortable with themselves yet make each other uncomfortable is possible.
- Some uncomfortability is important to the process of learning. **Scripted parts can assist in making sure uncomfortability doesn't become damaging**.

- What makes a conversation easy to have?
    - What makes a conversation uncomfortable?
- Conversation is easy to have when **both parties feel like they can contribute**.
- Conversation is easy to have when both parties feel like they **don't have to overly change their own pace** to accommodate the other.
- Conversation is easy to have when both parties **have the means to express** themselves fully.
- Conversation is easy to have when both parties feel like their **expression is received correctly** by the other.
- Conversation is easy to have when both parties feel like they are **not being judged**.
- Conversation can become uncomfortable if either party feels like they **cannot express themselves**.
- Conversation can become uncomfortable if either party **feels like they cannot be understood**.
- Conversation can become uncomfortable if either party is **unsure how to continue** the conversation.
- Conversation can become uncomfortable if either party is unresponsive.

- What makes a conversation interesting?
- Conversation is interesting when the topics are of interest to either party.
- Conversation is interesting when the goals of either party are being achieved.
- Conversation is interesting when it is new to both parties.

Common Themes:

- Too much variation = Uncomfortable.
- Too little variation = Uninteresting.
- Creation allows for expression, which helps with learning.
- Expression is a two-way street. It must be received correctly.
- Conversation allows for creation through opportunities to branch (deviate).
- Creation with the most personal expression deviates from the core structure.
- Commonality exists in the system.
- Commonality can be used as a structure.
- Structures can provide some comfort.
- Confidence is directly impacted by comfort.
- Treating the exchange of conversation as a system.

<u>CORE 1</u>

Too much variation = Uncomfortable

+

Too little variation = Uninteresting

+

Treating the LE as a system

=

Represent scaling uncomfortability/variation/etc as progressive states ending in a bad state, meaning liveness/termination can be checked formally.

<u>Imagine the Product:</u>

- Input name, and a bunch of vocabulary you can think of in the target language. System creates matches based on the number of good graphs it can produce. Those graphs/structures are provided along with additional vocabulary. This process only gets easier and easier because vocabulary can be accumulated and topics can be accumulated. Presumably confidence/comfort also increases (if at a marginal rate).
- Good-faith assumptions (players wish to obtain the goal, players are working collaboratively to accomplish the goal, players have similar goals of learning, etc)
- Treating the LE system as a graph-game that is actively being played by actual players (although formally the graph has already "been played" theoretically), players travel through the states (appropriate options appear on the screen) and "tips" appear on each individual screen that can grant vocab, feedback (approaching winning/losing regions).

- Essentially, users would have individual screens which provide information! (yea yea, conversation and all that, but people don't have to stare at their screen, and since the system has perfect information, no sync is required. Assistance tools are always helpful, especially when learning. This creates the use case for in-classroom augmentation.)

<u>The Internal Description:</u>

**This is the list of functions we need*

- Input: Name, List of Vocab
- Formalism to describe LE as a graph (system of vertices and edges)
- Graph-generation (ML model)
- Graph-verification properties via graph-game theory
- Vocab recommendation (ML model)
- UI interaction on screens session (WebApp easiest, manage an event via user signup, database management, etc)

<u>Getting Started:</u>

- Formalism to describe LE as a system of vertices and edges, and all the extended properties that we want to have.
  - Break down a conversation between two people.
  - Break down a conversation.
  - Note the parts and properties.
    - Think about how they can be organized/expressed.

Formalism to describe LE as a system of vertices and edges:

- Players
- The entire system of graphs is a LE system
- The graph itself is a singular exchange
- The graph is composed of nodes and edges
- Nodes represent arbitrary states in the conversation
- Edges represent an utterance (speech act) transitioning the state

LE = (V, U, T)

Where V is a set of nodes, representing arbitrary states in the conversation, partitioned into V0 and V1, nodes owned by the respective players, indicating their "turn" to advance the state. Note that in our cases, V0 and V1 partitions do not have to be fair.

Where U is a set of actions, representing speech acts in the conversation.

Where T is a set of transitions, the edges, representing the speech acts having an effect on the state of the conversation. Formally for a transition from state ***s*** to state ***s'*** through speech act ***u*** is (s, u, s'). T is a subset of V x U x V.

This is one of the simplest ways to represent our language exchange.
Now the next step is to add on top of the base definition to get functionality that we need. To consider a conversation:

A conversation is an exchange of speech/utterances between two people, where each speech/utterance is a combination of words, with the intent to convey a concept/an idea.

- Utterance -> words + intent

Where we can categorize intent using a set of understood speech acts (agreement, check.reception, accept.coordination, give.recall, etc)
Words are simply a collection of tokens (from the comp ling sense)

So what happens when we think about each speech act u in U as a tuple of ({set of tokens}, intent)? Does this ruin the system (make things undecidable, etc) or does it not really matter, as the content doesn't matter for deciding whether or not we can reach a certain state?

- In a way, we can imagine that it does matter, since we cannot randomly assign a given tuple to any transition/edge. Although we do say that the states in the graph are arbitrary states, what we mean by this is that they are not strictly defined, but there is some level of understanding that the states that are connected and closer have more of a relationship than states that are further away.
- Although, we also do have to maybe consider, is that true? Are any two points in the conversation really as far away as we think, or are we just missing a transition/edge that logically connects the two? (i.e: sometimes things can feel like they're complete non-sequiturs, or sometimes things that we imagine are unrelated can actually be bridged very well by an appropriate segue.)

Well, it's certainly not *random*, but it's certainly not as clear as we might think it is. The bottom line is still that we have to consider whether or not this type of relationship complicates the graph, making decidable results undecidable.

- What does the paper mean when they say "dimensions of the energy"? And what do they mean when they translate it to a pushdown automata? This is the exploration we need to do to determine whether or not we can actually map this the way we want to.

We'll start with Pushdown Automatas, which are a type of finite state automata that is non-deterministic and utilizes a stack. These types of automatas are useful when designing systems that can reject based on the stack. I.e: this transition requires you to pop an X from the top of the stack, if no X to pop, then reject. Or, this end state requires the stack to be empty. If not empty, reject. And then we can have the notion of accepting states.
One-Counter Pushdown Automata is just a pushdown automata, but there is only one stack symbol (not counting indicator for empty stack).
Here, trying to figure out if dimension of energy, which is described in the paper as, $\{-1,0,1\}^n$ means that transitions can only ever increment, decrement, or give no change, but that they cannot be +2, -2, etc… or rather, from the definition E = (e1, e2, … , e_n) just simply means that each player only has one value for energy. (This definition seems to make more sense, and in this case, gives us some confidence that the problem we're looking at could be decidable, depending on how we define the energy in the system of a conversation).

Small pre-meeting 1 research:
We wanna be looking up some of the additional research that's more related to graph-games and verification processes.
But the general idea is that we kinda would like some feedback from the team in order to figure out what the next direction could be.

Efficient synthesis of a parity/energy game from vocab resources?
Words like "create" or "generate" parity games don't really yield any results that I'm thinking of. Perhaps this is the direction that we can go in? The idea that the parity game states are technically arbitrary, but learning a way to map them/generate them properly such that we know that x,y,z properties are guaranteed. Is that even possible? (likely? probability?)

Fifth Checkpoint - Post Meeting #1

Notes

- Equivalence needs to be precisely defined, it involves a notion of properties that are preserved (i.e: reflexivity, transitivity, symmetry)
- Fairness needs to be properly defined. It is not about the partitioned number of nodes so much as it should be about there is a guarantee that "the longest single player path should be X".

- We need to take into consideration whether our modeling problem is finite or infinite in reference to the graphs generated. Additionally, if we are bounded but infinite, that's also acceptable.
- We need an explicit sample/example of a conversation graph model using the definition we have created.
- We need to be mindful to consider the state space of the problem.
- There are similarities here to petri-nets as a generalization, so there is room for us to look there in terms of papers.
- Very important: a graph that has more edges than vertices will be a huge problem, since then we cannot very well conclude or enforce properties. We want to say, "avoid this region of nodes" rather than have a case where a single node leads to any and all other nodes (extreme case). Meaning we need some level of restriction.
- We need to redraw/change the graphics we have to be more defined and specific, according to the considerations and criticisms listed above (with consideration to input outputs).
- There is an important question to discuss about HOW the graph is being generated. What we can do (that makes a lot of sense) is to have a system observe conversations, create traces, and then take that set of traces and apply a blackbox (ML alg?) to generate a graph that we know adheres to some value, and then we can analyze this graph in order to ensure properties, and then that graph can be utilized by players who then get good feedback.

Next meeting is on 28 Sep (Mon).

Tasks to do:

- Draw up a sample conversation to get a better idea of how the interaction will work (this will give us more answers or questions that we can look into).
- Consider how the graph is generated. It makes sense to do some research here (don't get too bogged down, just look for ideas) and then formulate the basic premise to make sure that we have a reasonable idea (like Dimitri did).
- Draw the graphic for the graph generation.
- Consider how we can restrict a conversation graph to make sure that we have a good amount of nodes balanced with transitions. This warrants research, and we can definitely spend more time here.
- Fix the graphics according to the listed issues (taking into account proper definitions of equivalence, fairness, how the entire lifecycle will work, etc).
- After those parts are finished, we should have a clearer idea of the research problem that we can start to explore.

Drawing up a sample conversation:

We define *SC*, a sample conversation.
SC = (V, U, T, W, C) as previously discussed.

Where V is partitioned into V_0 and V_1, $V_0, V_1 \in V$, denoted $b_i \in V_0$, $r_i \in V_1$ which indicates that our partitioned nodes will be denoted via colors Blue and Red.

Where U is the set of actions, which are speech, defined as a tuple (w_i, c_i) where $w_i \in$ W, indicating the words in the language, and $c_i \in C$, the set of collaborative acts labels indicating intent.

We will give the set C here for completeness.
{display.solidarity, display.hostility, relax.atmosphere, use.social.convention, check.reception, check.comprehension, display.active.listening, display.reflection, coordinate.teamwork, accept.coordination, refuse.coordination, give.task.information, give.explanation, elicit.task.information, give.self.information, elicit.partner.information, give.recall, elicit.recall, give.proposition, give.positive.opinion, give.negative.opinion, elicit.proposition, elicit.opinion, agree, incorporate, manage.task, manage.tool, other*, outside.activity*}
Here for the most part things make sense, we just need to define other and outside.activity a bit differently since they were originally used in context of working on task, but not general conversation (where other was essentially general conversation).
We would like to consider *outside.activity* as "non-sequiturs" in order to limit or restrict, but the problem is that every single speech has the *potential* to be a non-sequitur. **We have to think about whether defining every variant of a conversation** (i.e: each action will have two copies at least, itself, and the non-sequitur version) will impact our graph by increasing the nodes or the transitions. So i think actually theoretically speaking, since each node will have at least one transition outwards (minus the special case of end states), we are only looking to restrict the transitions, meaning we are looking to place restrictions on how many different actions you are allowed at each given state. This suggests that if we define the states (arbitrarily) more specifically, then we will be in better shape. Additionally, we can always merge a lot of states together into one region?

(we can do this later if we need to restrict) because technically there's not really a difference between what their job is if we can generate things about their job. However, this specificity may be a sweet spot we're looking for, so we will start out specific and merge if necessary.

For now we can move on.

Here is a sample graph in the context of a single possible exchange between two players. For the sake of getting a good perspective, I will detail the conversation first using Chinese (first but not native language) and provide the English translation below.

你好 - 你好，我是X
可不可以再所一遍？
我是Y
您貴姓？- 你的名字是什麼？
誒，很少聽到人家這樣問，哈哈哈。
你可以叫我Z
是【地方】的名字嗎？
你是哪裏人？
你看起來像【地方】人
爲什麼來會想來LE？
你的工作是什麼？
【工作】是做些什麼？...
你在哪裏學的【語言】？
去過【地方】，覺得怎麼樣？

So far from writing this, (thinking from the perspective of what it would feel like to be a generator) I get the impression that the more information we have on the player, the better sort of expansion of vocabulary we can make (i.e: if we know the occupation, then we can more easily expand-generate that portion of the conversation, granted that the vocabulary is sufficient)
Additionally, in preparing for the data-collection portion, i think it may be possible to collect my own data (traces over conversation i have with other people, for this intent) and perhaps this could go well/not go so well, but i'd like to give it a shot at least if we at some point consider the type of data i could collect to the be relevant.）
Additionally, from this small generation exercise, I can see that it would make the most sense to do what Dimitri suggests, in terms of running traces over lots of

different conversations, have some black-box/algorithm/model to generate a graph from there (where we input areas we think are similar in order for the graph to train) and then see if we can generate some new graphs given different parameters.
From this, we can also tell that the restriction is very important. Since the states are arbitrary, perhaps we need to come up with some rules of engagement, that specifies the parameters for a response since the introduction sections clearly branch the same way

Sample Conversation Graphs
Matching to the graph: (*note, replace V with 'u', we're working with transition actions here rather than states, since states are arbitrary)
V0 你好, use.social.convention
V23 你好, use.social.convention

V2 你好，我是X, use.social.convention
V1 可不可以再所一遍？, elicit.recall

V3 您貴姓？, elicit.partner.information
V9 誒，很少聽到人家這樣問，give.self.information
V25 哈哈哈。, relax.atmosphere

V6 你的名字是什麽？, elicit.partner.information
V4 我是X, give.self.information
V5 我是Y, give.self.information

V10 是【地方】的名字嗎？, elicit.partner.information
V7 不是, give.self.information (or do we alter here to give.yes, give.no?)
V36 是, give.self.information

V8 你是哪裏人？

V12 爲什麽來會想來LE？, elicit.partner.information
V24 喜歡學語言，家裏都講中文，因該偶爾練習, give.self.information
V26 哈哈哈。, relax.atmosphere

V13 你的工作是什麽？, elicit.partner.information
V27 我是個學生 , give.self.information
V31 在【學校】學【科目】, give.self.information

V32 在學什麼？, elicit.partner.information
V33 學【科目】, give.self.information
V34 哦，你覺得怎麽樣？, elicit.opinion
V35 哦，我其實很喜歡【科目】, give.opinion

V13 你的工作是什麽？, elicit.partner.information
U38 電腦軟體，在一個公司叫【公司名字】, give.self.information
U39 欸，我也做電腦軟體，你們都用什麽寫code?, elicit.partner.information
U40 我大大都用【codebase】, give.self.information
U41 我在一個不動產公司工作, give.self.information
U42 哦，你覺得現在房子價錢經濟怎麽樣？, elicit.opinion
U43 那是幫人家買房子吧，還是也幫租房子？, check.comprehension
U44 我們的公司兩個都有, give.explanation

V14 你看起來像【地方】人 , elicit.partner.information
V28 怎得嗎？我不是 , give.self.information (here followed by give.explanation? Or would it be an opinion? Does it matter?)
V29 是的，我在【地方】出生的, give.self.information

V15 你在哪裏學的【語言】？, elicit.partner.information
V30 小時候在家裏跟爸媽都所中文, give.self.information
V16 在大學的時候學的, give.self.information
V36 我去過【所中文的國家】在那裏學的, give.self.information
V37 可是大學的時候也有學一些, give.explanation

V17 去過【地方】， 覺得怎麽樣？, elicit.opinion
V18 我很喜歡, give.opinion
V19 我覺得還好, give.opinion
V20 你在那裏都做些什麽？, elicit.partner.information
V21 我在那裏工作, give.self.information
V22 我在那裏讀書, give.self.information

What i'm noticing here is that there is a lot of parallel ways to say the same thing. Meaning that there might be some merit to restricting/having traces, and having some tests for equivalence (i.e: some properties are preserved) in order to create a parallel model describing the potential paths, which ultimately becomes very useful for giving potential anticipations!
So here we note that we have different sentences, but they contain the same intention. And we can make a determination based on intention and based on the

words chosen that they may very well lead to the same state in the conversation. But this decision is made via the generator of the final graph (after running all the traces) so perhaps we should make the traces FIRST, and then connect afterwards.

Running a break here for a bit, I feel we're currently a little too close to the generation of this example without some of the important considerations/fundamentals in the subconscious, so we'll go back to looking at papers for now to gather some more relevant information.
Perhaps "graph reduction" would be a useful topic to search, perhaps could give us ideas on how to better organize our graph or how to eventually better formulate a graph. That and information on "graph generation" or anything along those lines.
There's a keyword here, "model compression", or "knowledge distillation" that seems like it would be information important to know since we wish to restrict?
So far we have

- Paper on knowledge tracing, which could be useful re: the tracing part
- Process discovery for program synthesis, which sounds like what we're trying to do
- Distilling knowledge graphs, which might be relevant to us, unsure yet

Are we reading this field right, that most of the work regarding updates to edges (remove, add) is a problem that is popular to solve via ML models?
It would seem that edges and connections between nodes are learned (which is in line with what we discussed re: running traces to create a graph)

I think the paper "About Graph Degeneracy, Representation Learning and Scalability" addresses the concerns we are having about the usability of the graph we are imagining if the number of edges becomes too big, since we won't be able to leverage any useful information. This will probably be the first paper I read, and using the keywords, maybe I can find additional papers that discuss this issue. That way, I can have these in mind when I go about creating the example graph.
Notes from the paper

- Represent graphs as we expected, and the new term we learned to talk about the number of edges is the "degree" of the node. This gives us a way to discuss the issue we talked about before. We do not want the degree of the node to become too big, since that will give us not great results (in the ML world, they would probably say that the embeddings won't be of good quality).
- They represent the degrees via a degree matrix (makes sense)

- They represent the graph using an adjacency matrix (also makes sense)
- They call the process of ML to find a good way to represent a graph to be: Graph Representation Learning (intuitive name) and I believe this would be the result of the generator black-box, since we would run traces of already established conversation graphs, learn a representation, produce a representation, and have some sort of mapping to turn this representation back into a graph for us visually, or to be checked in some way. We also then always have this representation to do further work on.
- They determine the quality of the embedding by training a logistic-regression model to classify the node/missing edge detection (aka, how close the model gets to representing the graph properly). Which then they use some F1 score. This makes sense, ensuring that the representation of the vector space is as close as possible to the original graph.
- The term Graph Degeneracy is also now available to us. They define a graph to be *k-degenerate* if all the subgraphs have a vertex of degree at-most *k*. This makes sense, and is the exact definition of the issue we want restricted. We now have a clear defined term for it.
- When we think about graph degeneracy in terms of "levels" upper levels being the higher k-degen values, representing nodes that are very connected to every other node, where we have the lower levels populated with lots of nodes that are more sparsely connected, the concept that they paint in the paper becomes a lot more clear. When we move higher up in the k-degen values, the more the core becomes connected, making it harder to derive embeddings (conclude information), and the more we go down the levels, the more we expect to be able to pick up information. For this, they used link prediction as a metric, meaning that they tried to remove edges and trained to create embeddings and see how effective they were.
- The paper states that "embeddings obtained…" did not lead to "...outstanding accuracy scores" which they attribute to the method not accounting for "...node labels or attributes." Which means that this could be something that we look into, see if this interestingly enough could be something that works for us (reduce k-degen, but have some metric of intent via collaborative acts and attempting an almost process mining like approach to the problem)

Next, we'll read the paper regarding graph node embeddings, as I feel like any insight on this topic will be somewhat useful for the generation part of what we're doing. Having a better sense of how the current field intends to generate graphs (via trained models based on nodes -> embeddings) will give us more insight and perhaps correct our approach.

This paper, “Fast and Accurate Network Embeddings via Very Sparse Random Projection” claims that proper construction of similarity matrix is more important than the dimensionality reduction method. More notes as follows:

- We should note that this paper and the one before it mentions DeepWalk, which seems like the benchmark for this type of node embedding work, so we should find that original paper and read it next.
- They note that real world graphs are usually sparse, making direct techniques on similarity matrices moot, since most values are 0. However, also important to note that they state just because two nodes are not adjacent does not imply they are not strongly related, as they could have connection via a large number of paths, implying their relationship is still tight. This leads them to consider higher order graphs. Can we imagine here that when they work with higher order representation, that something like considering the different levels of k-degeneracy could serve that purpose?
- Their method to construction of a better similarity matrix is through random projection. This set up defined by the graph reminds me of the same fundamentals behind the digital fingerprinting, where we are multiplying with some gaussian matrix as a projection to get images to become a much smaller representation via projection. The paper states that this action preserves the pairwise distances, which would make sense because it explains why fingerprints are reasonable representations of images, since the information from the pixels re: the pairwise distance would be maintained. Ah, everything makes so much more sense now…
- I guess the take away from the paper is that these are some optimizations to the embeddings, and that development is being done in that field. As long as we make sure we’re reasonable with the expectations of the creation of the graph, there shouldn’t theoretically be a problem with dealing with generation of a graph via these trained models (in terms of complexity or quality of the embeddings).

The next paper entitled, “GIKT: A Graph-based Interaction Model for Knowledge Tracing” aims to solve the problem of student knowledge dependencies in the long term dependencies by using graph-based models to represent those connections. This also involves propagating embeddings. From the other papers, we can expect this type of paper to sort of be an “applying the concepts” from before into a “specific context” (being students’ knowledge representations). This paper may be useful to us in the sense that we’re also trying to represent something with long term dependencies and relationships. Notes are as follows:

- Defines knowledge tracing as whether or not a student can answer a question correctly given previous history. We can parallel this concept with our context, can people involved in a conversation carry the conversation to a satisfiable endpoint?
- However, as the paper suggests, models look at answering this question of whether or not a question can be answered, not necessarily providing information to assist in answering these questions. Which is perhaps something that should be explored, since our restrictions are a lot less strict in the sense that there are multiple ways to reach satisfiable states in a conversation, versus there may only be a couple of correct answers to a question!!
- It's interesting to think about, from the perspective of answering a question, the paper states that a single skill can correspond to multiple questions, and a question will require multiple skills. From the perspective of carrying a conversation, we may find that the relationships are too broad (k-degen too high) and that we may need to restrict down to this model at the very least.
- The paper tries to use Graph convolutional networks to find higher-order relations, to mitigate this issue of skills and questions, where questions answered are sparse, or different difficulty of questions based on the same skill, etc. etc.
- The paper additionally worries about capturing long term student states in context of answering questions.
- Beyond that, this paper is more on ML model specifics, we'll stash it in the etc section for now.

This next paper, "GTEA: Representation Learning for Temporal Interaction Graphs via Edge Aggregation" seems to be wanting to deal with multidimensional temporal edges in the context of GNN. Which I'm not exactly sure what that means yet, but I guess reading the paper to find out will be helpful. I presume that our edges are single-dimension temporal, since they are responses to each other, but perhaps learning about a higher dimensional representation of temporal edges might give us additional ideas/context.

- The paper denotes the two challenges as being the single temporal series edges, as well as the issue we have discussed, now referred to as k-degeneracy.
- Ah, okay, the paper is referencing interactions between multiple nodes acting as different players, and temporal interactions between each one. Our context is one of those node connections, so it seems this paper won't be particularly relevant for us. Stashed.

The next paper, “Self-Enhanced GNN: Improving Graph Neural Networks Using Model Outputs” Just from the title, this seems like it matches the concept that Dimitri mentions about how the process works, we take the outputs of a black-box that produced an output of a graph from many traces. This seems like the paper is suggesting that there is a way to improve the output from that black-box (so to speak).

- So the paper addresses the concept that we can use the already accurate GNN to create updates to the input data to make the input data better, which improves the output. Seems like the same idea of stacking or ensemble learning in a way.
- For us, our context is between two players, where they discuss a context where it is many players (network). However, they mention the concept of removing inter-class edges and adding intra-class edges. This gives me the idea of, since all the states in our conversation are arbitrarily defined, and we noticed from our example graph that different subgraphs can be categorized as similar/different, maybe there is some merit in stealing this part of the technique to address our context, since we can create a situation where our “black-box” would be able to better draw connections between different subgraphs, and thus be able to produce a high-quality model that connects some concepts but not others (essentially helping us with the issue of k-degeneracy).
- The intuition for their edge addition/removal is based on some labels based on whether or not they are inter/intra class. They use GNN produced labels (which are relatively accurate already) to determine which edges to add/remove. Put into context, I guess this would mean, we have a GNN produce labels for whether or not it thinks a group of sub-graphs are of one class (serves the same purpose in the conversation, etc) and if so, then we can remove some of the edges from the node leading to these because they belong to the same class (*we can draw a picture describing this, with the concept of different sheets of sub-graphs, and all the edges from one node to those sheets, and then showing removal by greying out the other sheets). I think that concept makes a lot of sense, and it should be noted that this would probably mitigate a lot of the concerns of k-degeneracy in our graph?
- The question then becomes, well, for our context, how do we know which edges to add or which edges to remove? This I guess could be a result of fine tuning, such that we can in the perfect world, have few enough edges that we can conclude information, but many enough edges to capture the variations in the way people express thoughts.

Last paper is one that we've pulled from Round 1 of paper scouting, that we briefly went over, but will now go over in more detail since I feel we have a bigger picture allowing us to find parts of the paper more topical. The paper is titled, "Process Discovery for Structured Program Synthesis"

- So from the preliminary reading, I am a bit reminded of why we stashed this paper to begin with. It works in the context of structured-programs, for automated robots and deriving processes from watching a small number of demos in order to create instructions for those automated robots. So it is essentially the opposite context compared to what we need.
- However, there may still be some understanding that can be derived if we read (briefly) about something that is exactly the opposite direction of what we need.
- The input to their algorithm is an event log, the output is a program (that a robot can follow). This is where maybe I can imagine that instead of a singular event log, we can maybe input something along the lines of an optimized event log (maybe one produced from a black-box)? In order to use this same method to produce a "program" which put into words for our context, is a semi-script, or a guideline to follow for the conversation?
- Getting a bit better of an understanding of how they borrow the regular expressions to define a structured program as a syntax tree. Figure 1 provided gives a nice example.
- The miner algorithm they develop inputs traces and outputs a syntax tree. If we think about abstracting our graph, can we ever get to a situation where an abstracted graph (like a syntax tree) could be useful for achieving our goal? Or if not really, can we think about how we can take a syntax-tree and un-abstract it enough for our purposes?? I guess this could potentially be an area of discussion.
- The idea here for the miner is that in concept, when we have traces, we can notice that for instance, if a node 1 is followed by node 2 and not vice versa, true in all traces, then we can condense node 1 and 2 into a sequence. These rewrite rules can be applied, etc etc, and we end up with our final result presumably. This is perhaps where the original interest in the paper came from, is learning more about the idea of using traces to create rewrite rules that we can then use to abstract details to create general patterns in order to produce some guideline to an interaction (in their context, create a program).
- Note that the paper also states that when they condense using rewrite rules, they also try to simplify the result for the program produced.

- The algorithm proposed they states differs in that it iterates rather than recurses, and accommodates duplicate activities in the final process model, which actually is a function that maybe is more useful to us, since duplication is something that might be valuable in a conversation, since restrictions are looser in terms of how a conversation goes that allowing duplicates could allow us to capture more of what we want to capture.

Finished up the first run of example conversation graph. Noticed that we organized nodes via what we believe their intent to be. Effectively the graph doesn't change, but visually it is a lot neater. Maybe it is still worth it to have a graph that is fully connected, we would need to change the canvas size of the image to fit.
Note that we should add more in the days leading up, that way we can get a really nice portion of the net that is relatively thorough in its expression of what we are visualizing in our head.

From adding more conversation, Dimitri's comment that the graph seems like it would just end up being linear seems like a strong possibility, since a two person exchange is usually just that (an exchange) where it feels like it is turn based. So I guess the interesting thing could be looking at (and effectively thinking about) how the format can maybe change if we add another or more players.
Additionally, we should try and think back on the different conversations we have actually had and try to implement them into the graphic, because right now, generating some sample ones seems to fall into this trap of question/answer sequences, and it's not very interesting conceptually to analyze.

Although, after reading through some of the other papers, perhaps there's something to be said about understanding regions of the graphs (from multiple traces), and recognizing their similarities, as well as vocabulary that might provide useful for these regions, and then providing them on new traces (recognizing the pattern the conversation is going, and then providing information). However, how is this different from just a markov process that tries to guess what happens next? (autocomplete?)
There's also the potential that the arbitrary states can be merged in some logical/reasonable way. For instance, although we have different conversations, perhaps we notice that certain regions tend to converge to the same arbitrary state, or an arbitrary state within a region. Perhaps that convergence is important for us, and maybe we can try and reason with that and display that in the graphic?

Perhaps this can be better understood/dictated with energy? Or some sort of condition on the nodes themselves? The concept being whether or not we can label certain states comfortable or less comfortable. We did comment on this same concept above, check out the notes from the paper that suggested that the embeddings produced sometimes don't perform very well since they don't really take into account labeling etc. So perhaps we can use some sort of labeling to ensure that we can still produce useful graphs.
Maybe the nodes that serve as ones that can lead to different branches of conversations are of importance as states. They may be arbitrary, but those "key" states where a conversation now has the potential to go down a different region (due to many selection of possible next branches) may be categorically interesting in analysis. Is there something we call this in the literature? (Is it attractors?) Certain conversations beget strangely similar conversations? Do conversations become fluid flow? Do conversations spawn eventual fixed points?

Situation Assessment

In terms of the goals that we had outlined:

- ~~Sample conversation~~
- ~~Understand graph generation better.~~
- Draw the graphic for the graph generation.
- ~~Research restricting the number of edges in a graph to ensure we can get good information.~~
- Fix presentation graphics.
- Define a specific research problem.

So far, we have started the sample conversation, and we're looking to diversify the example so that we can try and get a better picture of our situation. We have looked into graph generation re: node embeddings (generation via ML models that learn edge connections given a set of nodes). Additionally, we have been introduced to a concept called k-degeneracy, a method attempting to improve the embeddings, which is good for us because it results in a restriction to the graph.
We still need to finish the sample conversation, but only after applying some of the other things that we've decided to look into.

Here are some things that we decided to look into:

- The k-degeneracy paper suggested that the embeddings they got did not always lead to very good accuracy scores, since they don't account for node labels or attributes. So maybe there is some sort of labeling/attribute re: our conversation properties that we can use to improve the resulting quality.

- We need to answer the question of the difference between our goal and something that can just be solved via a simple markov-process (autocomplete).
- Perhaps to answer the previous question, we need to revisit concepts like applying energy to the graph, think about concepts like attractors the graph (fixed points?) do arbitrary states in the graph generate similar outcomes/conversation paths, and can we determine those through some of our graph properties, etc?

Answering these questions will give us a better direction towards committing to a specific research question!

The Paper, "Extremal Results for Graphs of Bounded Metric Dimension"

- is interesting in that it presents the concept of a "landmark" inside graphs, and the problem of determining a landmark ℓ that distinguishes two arbitrary nodes u and v
- We can probably think about this problem in the context of our graph, since there might be something worth thinking about in context of certain nodes being valuable (serving as landmarks).
- Introduces the concept of a resolving set, which is the set of landmarks that distinguishes any two nodes in the set. The resolving set will not have two vertices have the same distance vector.
- Thinking about it like this, will this property be possible inside our conversation graph? Are there landmarks that distinguish parts of the conversation? I imagine there could be if we're looking at traces of conversations. Then there might be something to be said about these landmark nodes that could provide information for us when we generate a graph? (We can think about this while creating the example graph)

To begin our day, we'll read through the rest of the papers we have in backlog, draw another portion of the sample graph, and start making some decisions in the direction we're headed!

The Paper, "A Parity Game Tale of Two Counters" discusses some concepts that are of interest to us:

- Tangles, which they describe as a strongly-connected subgame within the parity game for which one player has a strategy to win all the cycles in that subgame.
- They draw some distinctions from other similar concepts (snares, dominions, etc)

- This concept is perhaps interesting to us, since we're interesting in finding out properties that we can utilize.
- Concept of priority promotion, discussing the basis of regions, and how these regions of the graph have the property that all plays in the region grant a win to a certain player. There are some algorithms to calculate this.
- Then there's the definition of tangles, which is where the player wins all cycles in the subgame. There's a relationship between tangles and dominions.
- From the paper it seems like a lot of algorithms compute regions by removing certain discovered subgraphs from the game, in order to guarantee that what remains fits certain properties that are easier/further calculated on. Perhaps this works to our advantage since our graphs won't contain things such as two nodes for which an infinite cycles emerges. (perhaps this also implies that our graph needs a temporal element?)

The next paper we look at today, "Edge Degeneracy Algorithmic and Structural Results" looks at the notion of edge degeneracy through their version of a cops and robbers game played out on a graph. Their parameters for the graphs are a bit different from the ones we imagine, but the way they go about describing and setting up their problem could be useful to us nonetheless.

- Interesting because the way they have the graph set up, they explicitly set up the graph with the intent that there are parallel edges. Which makes sense since the game is a cops and robbers game, intuitively meaning that movement from one node to the next is not a simple one edge case. The idea that those parallel edges are arbitrary in the sense that it doesn't matter which one they take, but not arbitrary in existence, since it affects the calculation. I guess there are alternative setups where we can get the same effect using singular edges if we factored in weights?
- Which then begs the question, perhaps there is another way to represent our graph in a way that resolves our issue of a singular path/fear of highly connected degenerate notes?
- For now we'll skip the formal proofs until we revisit

Gotta think about how the example graph interacts with some of these concepts.

Do our graphs work in the sense that we are always a progression down the "layers" of the graph, never returning to the same state? The situation should be that there almost should never be a self linked node, but cycles can occur, presumably, not to the same state, but to the same "level" of the state (in terms of

progression). Since there are logical derivations or references to previous mentions in the conversation.
Perhaps there's something to be analyzed in terms of an arbitrary distinction of whether or not a conversation felt "successful", and elements/properties that can be explored on whether or not that property exists in a certain (generated) graph, etc. etc.

Sample Conversation (continued)
Try some conversations that we've had before?
U100 欸，你也是從台灣來的, check.reception
U101 也是？你的flag是中國的, check.comprehension
U102 我不知道他們有台灣的國旗, give.explanation
U103 當然有，如果你是台灣就因該貼台灣的, display.solidarity
U104 不然現在人家會以爲你有corona, relax.atmosphere
U105 哈哈哈, relax.atmosphere
U106 trueeeee, use.social.convention

U124 你是台灣的哪裏人？, elicit.partner.information
U125 臺北 , give.self.information
U126 欸，我也是 , display.solidarity
U127 好巧哦，小世界 , use.social.convention

U128 好久沒有回去台灣了 , display.reflection
U129 現在想回去也沒辦法 , display.reflection
U130 欸，可是他們抵抗corona還很好哦 , give.negative.opinion
U131 對，可是去了話可能回不來 , agree
U132 哈哈哈 , relax.atmosphere
U133 也有一點不想回來，就回台灣好了, display.reflection
U134 書也不想念了，美國政府也沒好事 , display.reflection
U135 哈哈 , relax.atmosphere
U136 美國真的是，不知道他們在幹什麽。。。, agree
U137 好，這裏念完書回台灣 , give.recall
U138 哈哈哈 , relax.atmosphere
U139 哈哈哈 , relax.atmosphere

U107 你是台灣來念書的嗎？elicit.partner.information
U108 來念書，可是其實從美國來的, give.self.information
U109 我台灣出生的，可是在美國長大的, give.explanation
U110 你是在這裏工作嘛？, elicit.partner.information

U111 對，我是個律師, *give.self.information*
U112 哦~ 什麼律師？, elicit.partner.information
U113 international law , *give.self.information*

U114 哇，那日内瓦一定很好, display.solidarity
U115 還好啦，use.social.convention
U123 要怎麼説，"*international law is the diminishing point of law?*",
elicit.task.information
U116 國際法是法律的遞減點？, *give.task.information* (**elicit.task.information ??)
U117 不確定，可是聽起來是對的，*give.positive.opinion*
U118 哈哈哈, relax.atmosphere
U119 哈哈哈, relax.atmosphere
U120 意思明白就好了，國際法真的有一點是個diminishing point, agree
U121 不同國家的法律永遠，很少，會同意。, display.active.listening
U122 對！就是這個意思, agree

This branch is highly linear, even though it arguably has three layers, where those layers could have branched into a different direction?
What made us layer the exchange into three parts? How did we know where to separate? This is I guess from a human perspective, intuition? From an algorithm or computational perspective this seems less clear. Perhaps this is what is known in the literature as a conversational turn? These turns are oftentimes researched in user-intent literature, and almost always done via ML.
Here I guess from our sample, we can see that from these separating points, we could have branched off to talk about a litany of other things (i.e: from 1 -> 2, not asked about profession, and instead ask about some other personal info 1-> X. i.e: from 2 -> 3 not mentioned Geneva as a location, and instead asked about a different opinion of their profession 2-> Y, the result being we don't get the specific piece of dialogue, but I guess in theory we get SOME form of dialogue regardless.
I guess there is something to be said about being able to recognize these points/nodes in a sample graph through aggregation of lots of sample graphs (traces), and to be able to generate a bigger wholistic graph provides a more robust view of the conversation, which gives us what exactly??
I guess there is an intuitive question to be asked here is that do we think that certain states (although arbitrary) have more logical branches (edges) than some other states in the graph? I think the answer is definitely yes. If we're thinking about some abstract "rules" of a conversation, then we can reasonably expect that it follows a certain flow. Then there are definitely nodes that (in the literature

would be called) “attractors” that offer more branches, for example, in our sample conversation those would be nodes after u106, u113.
The next question is what we attribute to those being points that we think are attractors (theoretically speaking, since we only created a sample trace). From a human intuition, perhaps it is because it is topic/intent based, but that in itself feels flimsy still, especially from our previous project on collaborative acts, modeling the labels of the acts didn’t always yield as conclusive information as we would have hoped. This perhaps implying that it isn’t as simple as a question of “look at the intent/topic switches and we should be able to determine attractors”. Additionally, we notice that these “attractor nodes” don’t necessarily “belong” to either player. Since these natural breaks result in either player having the potential to start a new conversation path. Perhaps this is something of a variance to our originally thought graph (game?).

Well, I guess from a theoretical standpoint, having many traces, that we then use to generate a graph, in order to determine nodes with high k-degeneracy/attractor/critical node would be interesting. Perhaps there is some sort of algorithm/process we could undergo to identify these in a real-time trace (after some training/previous calculations?) As these would be useful to identify?

I think the next step would be to generate a bunch of interesting/useful questions that we could approach, and determine from that list which are even possible to research (and then set up a timeline for steps we would need to research that question in terms of resources, data, etc)!

*side note, when we draw our sample conversation, we do have runs, but couldn’t we just compress that to be a self-directed node? Or is it better to have the unfolded version? My intuition says that unfolded is probably better?

Further considerations
Potentially Interesting Properties/Questions:

- Use traces to generate a graph, in order to determine nodes with high k-degeneracy/are attractors/critical branching nodes. Is there an algorithm to do this on a real-time process, or can we produce a graph that has these properties (because they would be useful to us)? But why would they even be useful???
- I guess the concept is that certain states could be considered attractors/hubs/well-connected, and we could determine why they’re useful

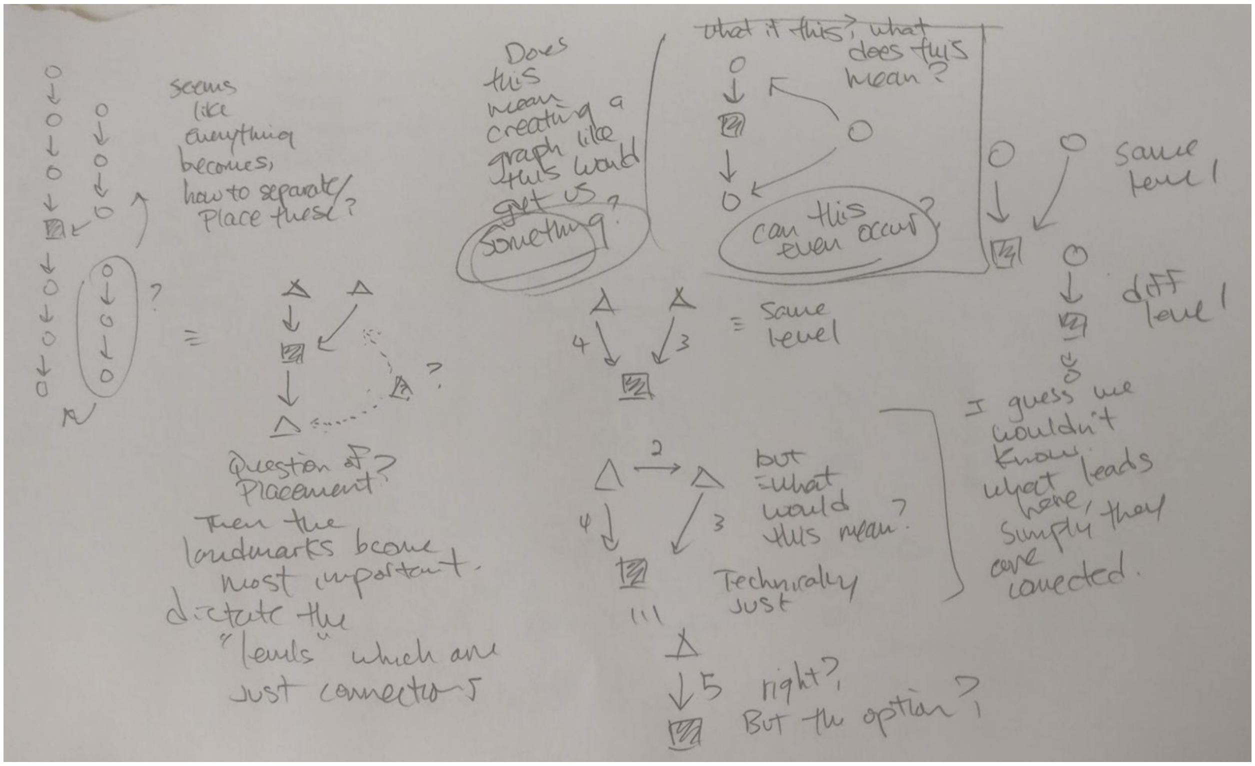
through analysis. If we can also guarantee a number of them in the conversation, we can have graphs that function well as guidelines?

- Labeling nodes to get better generated models? This is sort of an extension of the previous project where we noticed that a model on the labels didn't produce conclusions that we hoped to get.
- Need to answer how our process differs from just a markov process (autocomplete)
- Consider depths of conversation? We know from our sample conversation that usually we don't have edges that link back to previous nodes (temporally speaking) except maybe in the case of a generated graph, we can have nodes that are connected parallel to a certain "depth" in the conversation (i.e: if we believe that conversations have a beginning such as an introduction, etc etc, natural progression) then we can say that there are qualities of the depth of conversation that we could find interesting. Here "depth" in the sense of parallel edges. I guess visually it looks like there's a tree-like structure since we do know that conversation has a general "form" so to speak.
- This piggybacks on the concept from before regarding attractors/hubs/well-connected nodes, as each of these can be described as being the core to that "depth" of the graph, and that can have interesting analysis implications.
- The problem being that we could never collect enough conversations to build an actual graph that is interesting enough (compared to just a scripted conversation).
- **Can we model a conversation? Yes. But why would we want to? What do we get?**
- We get properties we can check? What properties are these, why does getting them give us anything meaningful in analysis? I guess this relates a bit to the notion of process mining, where the question is what do we get from process mining analysis? Bottlenecks, points of failures, etc… These concepts are ones that could be applied in the same manner to conversations? Are we arriving somewhere then where formal modeling has less expressiveness to tackle the problem we want to tackle, compared to something like process mining?
- Can we use models to do matchmaking?? Or would that be a byproduct of a model we generate

There's now the concept of the attractor node/branching node that doesn't belong to either player, and where it connects, how we determine it, etc etc. This could

change the structure/rules/setup of the graph, and may have implications on the information we're looking for. For now this is our next step.
There can be the concept: "through a graph of traces, where do we put the critical nodes such that it can allow for multiple traces to stack properly onto each other?", and is this closest to a question of edge prediction (i guess, node prediction)? Or topic turn prediction? Or is it something that we can determine as a result from the description of a game? (i.e: these critical nodes are states that have the property such that they lead to xyz game result, such as traces or dominions and their "distractions" as mentioned in one of the other papers).
The intuition should tell us that factors such as "words used" and "intent label" have some influence over whether any given node can be converted to a critical node. Additionally their positioning in the entire trace of the conversation, does that matter?

Landmark Nodes? More Abstract graphs from Traces?
Does any of it matter if the graphs just become linear traces?
Is there a way we can expand the set up such that the graphs aren't just branching trees? Yes. If we think about the way we set up the "levels" for the previous conversation traces, we can group the nodes that are part of one "topic" (which I guess is also an open question on how we do so, i think in relation to the "landmark" or critical node makes a lot of sense) the resulting graph is a linkage of conversations (which can still be partitioned via who initiated that conversation) separated by landmark nodes (anyone could initiate next) and those conversations acting as nodes, have edges which are now the arbitrary ones. (one conversation links to another arbitrarily, the nodes now contain all the relevant info). In this case, we can say that certain nodes connect to each other as conversations can relate to each other without a landmark to transition them (sometimes conversation happen to "bleed", this often occurs, right?). This should yield us more of a graph rather than a tree, no? And in this case, it might be interesting for us to use this type of graph to better understand not what is being said in the conversation rather, but this feels more like an abstraction to how a conversation progresses?
So, seems like there's something useful here? I'll attach the image from my paper notes:



Important to note here is thinking about what happens if we have a node that links to nodes of different levels (separated by a landmark) or, what happens if there is a cycle (can this even occur?) I guess this entirely depends on how these graphs are made, and what types of conversations we find. Perhaps there is something to be said about modeling conversations in such a way, maybe this type of analysis will give us some insight on progression of conversation, and maybe this modeling can give us something useful for language processing? (maybe?)
But this reminds us of CTL, so perhaps it is wise to look into that field of literature, to see what is going on in that world.

More Paper Notes

Now with this in mind, there are some additional papers that we find might be useful to give us some more context/things to think about.

The paper "Just Add Functions: A Neural-Symbolic Language Model" provides an interesting look at modifying a neural model to better perform on symbolic language, involving numbers, and geographic locations.

- One of the interesting concepts is that they do this by attaching some sort of rules called micro-models, which are specific to the target-domain. (i.e: they notice that four-digit years generally don't stray very far apart. This relationship they can code as an inductive bias over a probability distribution, which they can use to augment the model.

- The core concept being that there is a symbolic element that is fired only when a prediction is made for a class in a hierarchy. This is why this gives good domain-specific output.
- Additionally, the benefit is that symbolic forms are much easier to manage in terms of space, and do not need converting (like embeddings do).
- From our perspective, this gives us the ideas for maybe not being so afraid to augment, and have smaller modules that augment, as they could still provide useful results.

The paper “Learning Natural-Language to LTL Executable Semantic Parser for Grounded Robotics” is applying a model to better learn natural language commands for use in robotic agents. Although not directly related, actually works well within the realm of modeling natural language.

- This paper also mentions the case for an “inductive bias” when it comes to natural language, which they also express using symbolics (in this case LTL).
- This tells me that this would be a reasonable avenue to explore when looking at labeling/rules/augmentations to modeling in the context of language.
- This paper more so turns the problem into translation of natural language to LTL, and tries to run resulting performances through these learned LTL formulas, with some additional checks/information assurances.
- For us, might not be so useful, but the notion of utilizing LTL, as well as some of the limitations they mention that led them to their methodology is important enough to note.

From all of these considerations, it suggests that we should define the traces we get and the generated graphs resulting from analyzing the traces separately.

Trace = (V, U, T) as defined before, where V is a set of nodes, U are sets of actions, and T are the transition mappings.

We then define what our generated graph would look like:

Gen. Graph = (N, L, M, W)
Where N is a set of traces, represented by nodes.
L is a set of landmarks, represented by nodes.
M is the transition mappings.
W is the mapping function mapping weights to each transition.

This definition would reflect the image that we have before.