Masters Thesis

Part 2 of the notes document.

Defining Traces and Gen. Graph

From all of the considerations before (found in the previous document), 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 (where the weights represent the length of the trace).

This definition would reflect the image that we have before.
It also suggests to me that the critical-nodes/landmarks are the thing of interest for us in our model. Finding out how to define them, finding out how to think about them in context, finding out how we could derive/calculate/obtain them, how they fit into the graph, etc.

Critical-nodes/landmarks are natural breaks in the conversation, where from our previous definition, no specific player owns the node. The conversation still proceeds normally, and these nodes are likely to be the ones that are densely connected. I guess from a theoretical perspective, the more landmarks an exchange has, the more difficult it will be (landmarks insinuate more opportunities for creation, and creation is not only the defining aspect of learning, it also is the most challenging task). Analyzing and understanding a graph/model based on the

number of landmarks and/or defining a graph based on a certain number of landmarks (maybe with additional properties?) is definitely useful. The paper that introduced the concept of landmarks also suggests that it would logically give us a metric of distance in the graph. Which for us, I am not quite sure how useful it would be? "Distance" in this context would just be the number of utterances/actions before reaching the landmark. I guess we can say that a longer chain of exchanges is more likely to be prone to error/deviation, and therefore we can probably garner some information from that as well.

The notion here being that traces are highly linear, but we can transform the traces into something less linear, and the resulting graph would theoretically provide us with a framework to do something useful. We just have to formally define it I guess.

It also seems like there may not be an algorithmic way (at least not entirely algorithmic) to determine the connections of the nodes (traces to other traces, traces to landmark, etc). It seems like this graph generation comes from taking lots of traces, and running a trained edge detection model. Which makes sense. I guess my worry is that we don't want to spend a lot of effort obtaining this graph only to find that the information we get isn't so useful? So I guess the formal definitions and clear defining inputs/outputs could help us arrive at a conclusion for whether or not the hypothetically obtained graphs would be useful or not.

I guess it would benefit us to draw up an example of this graph -> specifically cases where we determine short/long cycles in the graphs and what that even may mean in context of a conversation, whether or not the graph still becomes linear, etc etc.

There's something to be said about a problem in a conversation of a foreign language, is the issue of the context switching too fast/the flow of logic is not understood (i.e: "I don't understand how one thought connected to the next", "What I expected them to talk about suddenly changed into something else.") Additionally, this may have connections to different modes of expression between individuals, which further complicates this issue.

Can we alleviate this issue by deriving some understanding from the graph, such that we can discuss guarantees such as, you must pass a certain node before you reach this specific node? (LTL variations?, eventually, until, etc…)
Does the existence of landmarks complicate or alleviate this issue?

Pre Meeting 2 Prep

Some tasks from last time:

- ~~Fix the graphics~~
- ~~Sample conversation graph~~
- ~~Consider how the graph is generated~~
- Research problem to explore

Fixing the Graphics

We'll probably just scrap the graphics from last time and focus on a sample conversation, along with additional drawings for whatever concepts from the papers that we want to expand on.
I think these two graphics would be the most useful right now:

- ~~Sample conversation~~
- ~~Generation of graphs (trace -> graph)~~

Research Problem

Not sure if what we have is enough for a research question. But there is something interesting in taking traces of conversation exchanges, and utilizing them to generate a unique graph that grants us some representation of possible exchanges, as well as a framework for analysis (i.e: reachable states, etc, granting certain factors such as vocabulary, etc).

Some Paper Notes

"Schema-based conversation modeling for agent-oriented manufacturing systems"

- Seems helpful in confirming some notions that we had, such as it stating that "... communication can be better modeled and more easily implemented when a conversation rather than an isolated message is taken as the primary unit of analysis." This confirms the idea that we were starting to see before

regarding "zooming out" and abstracting over the linear conversations, and instead focusing on small exchanges as a unit with "landmarks" where many edges are possible.
- This paper suggests to me that our set up from our sample conversation will reveal a lot about the schema and possible conversation interactions, which can further shed light on interesting points for analysis.

One of the papers that we stashed introduced the concept of disfluency, which is the notion of making mistakes (repeating words, uncertainty pauses, etc) in normal conversation (that I presume would be otherwise considered correct). This perhaps may be something that could be a side-effect of effectively using a modeled graph? Perhaps something to think about in context of using our generated graph for an analytical purpose. (Considering intent, flow of conversation, reachability of states, can we use all of these notions to minimize disfluency or provide support?)

I guess the theme is that we're trying to apply notions of formal modeling to provide some sort of framework for analysis in order to solve issues that the field traditionally applies language models and machine learning in order to "numerically" solve

Post Meeting 2 Notes

We have some new directions:

- Move starting from the conversation, to derive something like a sub-conversation within the conversation, away from a graph.
- If we start from a set of labels that have high abstraction (open.question, closed.question, response, check.reception, clarification, etc). But Dimitri said having a higher abstraction is better.
- Create a graphic -> two full conversations, where every node is labeled, we have some way to define a sub-conversation from the conversations, and then we can see if we can relate the two conversations based on sub-conversations.

Next meeting is Monday, 05 October.

Sample Conversation - Subconversation

With the goal of seeing if we can formulate a sub-conversation from a trace of a conversation, and then relate the two sub-conversations resulting from two different traces, we start with defining a set of labels.
It was suggested that we keep the labels broad first.

open.question
closed.question
respond.agree
respond.deny
display.reflection
give.opinion
deflection **
use.social.convention
relax.atmosphere

**We define deflection as a followup that is neither respond.agree nor respond.deny, but is categorically attached as a response to the original question.

This first set of labels seems reasonable. There are some slight flexibility, such as *give.opinion* labels for sentences that are not opinions but simple declarations, or *respond.agree* to mean both "responded to the question with an actual answer" and "responded to the question in the affirmative" and the same but in the negative for *respond.deny*. (Note that responding by explicitly saying you wish not to answer a question is not deflection, and is often rare in non-hostile environments)
With regards to deflection, unsure how often it comes up in conversation, but it may be interesting to see (because it is a unique category), although I guess we could also group it with respond.deny, but I feel they are different enough (it's value neutral rather than a response in the negative or denial to answer, whereas I feel answering a question at all is a response in the affirmative).

Trace 1 - International Law

U100 欸，你也是從台灣來的, closed.question
U101 也是？你的flag是中國的, deflection
U102 我不知道他們有台灣的國旗, deflection
U103 當然有，如果你是台灣就因該貼台灣的, give.opinion
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？, open.question
U125 臺北 , respond.agree
U126 欸，我也是 , respond.agree
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 , respond.deny
U131 對，可是去了話可能回不來 , respond.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，不知道他們在幹什麼。。。, respond.agree
U137 好，這裏念完書回台灣 , give.opinion
U138 哈哈哈 , relax.atmosphere
U139 哈哈哈 , relax.atmosphere
U107 你是台灣來念書的嗎？closed.question
U108 來念書，可是其實從美國來的, respond.agree
U109 我台灣出生的，可是在美國長大的, respond.agree
U110 你是在這裏工作嘛？, closed.question
U111 對，我是個律師, respond.agree
U112 哦~ 什麼律師？, open.question
U113 international law , respond.agree
U114 哇，那日內瓦一定很好, give.opinion
U115 還好啦，respond.agree

U123 要怎麼說，"*international law is the diminishing point of law?*", open.question
U116 國際法是法律的遞減點？, give.opinion (**elicit.task.information ??)
U117 不確定，可是聽起來是對的，respond.agree
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, respond.agree
U121 不同國家的法律永遠，很少，會同意。, give.opinion
U122 對！就是這個意思, respond.agree
U123 你現在念碩士吧？，closed.question
U124 對，現在在寫thesis, respond.agree
U125 你thesis題目是什麼？, open.question
U126 我喜歡語言交換，所以在看用電腦模範(mo2fan3)語言交換有關的，respond.agree
U127 哇，聽起來很複雜, give.opinion
U128 哈哈，有一點，respond.agree
U129 但是我覺得是應爲電腦係裏現在做跟語言有關係的都是在做機器翻譯，很少研究用機器幫學習語言的方法, give.opinion
U130 欸，那你這種研究是哪裏不一樣？, open.question
U131 有點難解釋，因爲我也還沒有完全想通細節，可是現在電腦係裏很火的是機器學習，然後機器學習係裏跟語言有關的都大大是跟自動翻譯有關的，像把語言變成數字去解決這個問題，我比較喜歡學語言的理論(li3lun4)，所以想研究一些比較把語言留著的分析，respond.agree
U132 啊，那這種研究的數學會比較複雜嗎？跟機器學習比, open.question
U133 mmm, 對我來說複雜一點，可是它們兩個用的數學不太一樣，機器學習用optimization跟derivatives，這種語言交換模範的數學比較像theoretical mathmatics，很多跟graph theory有關的, respond.agree
U134 但是其實我覺得我的Theoretical math(純粹chun2cui4數學)比我的applied math(應用ying4yong4數學)差一點, give.opinion

Trace 2 - Meeting Kevin

U200 欸，hello, use.social.convention
U201 hey, use.social.convention

U202 用中文沒問題吧？, closed.question
U203 沒問題，respond.agree
U204 你跟你的前一個partner都是用中文嗎？, closed.question
U205 差不多，有時候會換英文, respond.agree
U206 那我們也盡量用中文, give.opinion
U207 yea, 挺好的 , respond.agree
U208 那你現在都在做些什麼？, open.question
U209 想在在日内瓦念masters, 寫thesis, respond.agree
U210 你在加州念書吧？, closed.question
U211 第幾年了？, closed.question
U212 對，我在USC第四年了, respond.agree
U213 啊，那快畢業了, 現在一定很亂，give.opinion
U214 yea, 很累，respond.agree
U215 其實畢業以後想去紐約找工作, give.opinion
U216 哦，紐約不錯，你是做finance, right?, closed.question
U217 yea, respond.agree
U218 紐約的finance很好。但是現在哪裏找工作都很不容易, give.opinion
U219 你可以再念一年書嗎？還是沒辦法？, closed.question
U220 可以，我也是想說如果找工作不容易可以再念個CS minor, respond.agree
U221 哦，finance念CS minor很好，一定會用到。respond.agree ** (or is it give.opinion here?)
U222 嗯，所以還有很多選擇, give.opinion
U223 啊，挺好的, respond.agree

Draw and Correlate

So the next step here is to provide a graphic for both traces of conversations including their proper labeling, and then trying to correlate between the two traces to see if we can define a "sub-conversation."
Additionally, determine what direction we can take these sub-directions in and use them to what end.

I guess from the perspective of doing the correlating and sub-conversation definition, we should start by outlining instead of jumping straight to just doing

preliminary algorithms. Might also be worth first looking into the "inductive miners" algorithm etc that's in the literature to see what we've got as groundwork. Additionally, I do think we need to substantially increase the length of the conversation trace to get a good sense of the normal shifts (since our topics that we immediately remember are all introduction based, and that will add a bias).

We should also keep in mind that there might be other metrics that we want to keep in mind when we're trying to create a sub-conversation (i.e: the length of an utterance based on word count, the length of an utterance based on sentence count, etc) I think these will assist us in defining sub-conversations since they provide some indirect semantic information about an utterance (i.e: sometimes a long utterance with long word count and sentence count with a respond.agree after an open.question tag can imply an "explanation" while a respond.agree with a short word count after an open.question could be closer to a deflection than an explanation. Or something like, long respond.agree with a long give.opinion from the same speaker afterwards means preempting by giving a clarification or expanding, and something like different speaker give.opinion might be displaying active listening via confirming information.

We have some samples of possible sub-conversations from trace 1, they're based on simple rules of the labels

CLARIFICATION: respond.agree -> long utterance with give.opinion from the same speaker

(('意思明白就好了，國際法真的有一點是個diminishing point', 'respond.agree', '1'), ('不同國家的法律永遠，很少，會同意。', 'give.opinion', '1'))

(('哈哈，有一點', 'respond.agree', '1'), ('但是我覺得是應爲電腦係裏現在做跟語言有關係的都是在做機器翻譯，很少研究用機器幫學習語言的方法', 'give.opinion', '1'))

(('mmm, 對我來説複雜一點，可是它們兩個用的數學不太一樣，機器學習用optimization跟derivatives，這種語言交換模範的數學比較像theoretical mathmatics，很多跟graph theory有關的', 'respond.agree', '1'), ('但是其實我覺得我的Theoretical math(純粹chun2cui4數學)比我的applied math(應用ying4yong4數學)差一點,', 'give.opinion', '1'))

ACTIVE.LISTENING: respond.agree -> long utterance with give.opinon from different speaker

(('美國真的是，不知道他們在幹什麼。。。', 'respond.agree', '2'), ('好，這裏念完書回台灣', 'give.opinion', '1'))

(('international law', 'respond.agree', '2'), ('哇，那日內瓦一定很好', 'give.opinion', '1'))

(('我喜歡語言交換，所以在看用電腦模範語言交換有關的', 'respond.agree', '1'), ('哇，聽起來很複雜', 'give.opinion', '2'))

Post Meeting 3 Notes

Went over some of the new information regarding finding sub-conversations within the bigger conversation traces. We're concerned now with understanding how in the field we would go about defining a metric of similarity/distance between traces. As well as understanding if we can define and create archetypes for sub-conversations, which would allow us to do a bunch of other things.

Next meeting is Monday, 12 October

We're looking at finishing these tasks:

- ~~Research into the field regarding similarity/distance metrics for traces~~
- ~~LTL-variations or variations for dealing with levels of uncertainty in our patterns for trace labeling~~
- Probability Models and how we can integrate them for our use, i.e: if we can classify archetypes of sub-conversations, what is the probability that we will reach a certain sub-conversation state? (i.e: argument, etc)
- Find sample data we can steal

First let's solidify some of our properties (labels):

open.question - a question that expects more than an exact singular response.
closed.question - a question that expects an exact singular response.
respond.agree - response to posed question that is in the affirmative or satisfies the question parameters.
respond.deny - response to the posed question that is in the negative, or denies answering the question.
display.reflection - an utterance that is meant to be part of an inner monologue

give.opinion - an opinion given to the other speaker on the topic, or a statement spoken by the speaker.
deflection - a response that is neither an affirmative/negative, and satisfies the parameters of the question, but the other speaker cannot confirm whether this satisfaction is valid or has occurred.
use.social.convention - speaker engages in a social convention (i.e: polite nod, greeting, gestures)
relax.atmosphere - speaker engages in a social convention with the intent to affect the mood (i.e: laughter, inside joke/statement based on rapport)

Then we can also consider some properties based on these labels:

- Clarification (respond.agree.1 -> give.opinion.1)
- Active.Listening (resoond.agree.1 -> give.opinion.2)

Next, we have to define what we mean by "distance, or similarity metric for comparing conversation traces" since there are tons of different definitions of understanding distance or similarity.
The traditional CS sense of distance seems to be regarding edit distance or distance based on vocabulary (word count). Our notion of distance is moreso to distinguish between two arbitrary conversation traces based on the labels that we have on the intent of each of the utterances in the trace, in the proper order in which they appear.
So I think terms like distance or similarity might get us more bogged in the machine learning/info retrieval world, whereas we want to be more in the models equivalence world, so maybe the terms "equivalence" or "stutter equivalence" might be a better starting point.
Though it seems like for some of these papers, there is a sort of relevance for both of these notions of distance, and perhaps they are used together to achieve our goals? We'll start from our end and work towards the middle.
Terms like "clustering" also appear, but again, draws out a lot of ML literature.
"Trace analysis" is also a good term.

We've found a healthy amount of papers to start, we'll sort through them and take relevant notes here:

Paper Notes

Since we're concerned with figuring out the notion of *comparison metrics*, i.e: distance between two partial traces (defined sub-conversations or larger conversation archetypes) and/or similarity between those two partial traces, we are focusing on this notion of *how do we compare traces*?

"What to Do When You Can't Do It All: Temporal Logic Planning with Soft Temporal Logic Constraints" (Aug 2020)

- Paper aims to find inside a graph, an infinite path that satisfies some set of soft LTL constraints, while also satisfying a hard LTL constraint.
- Off the bat the direction is different, but this notion of satisfying LTL constraints might be useful enough for us to think about how we can define our comparison metric.
- They use a cost function to ensure a level of priority between their soft restrictions that they wish to enforce. In this same method, we can consider our sub-conversation labels or sub-conversation archetypes as some sort of "soft-constraint" that each trace of our data could possibly satisfy or match. Meaning that We can consider the resulting "priorities" of the list of labels, and a comparison metric can be the similarity of the priorities of the labels (especially in classes we deem more connected. I.e: argumentation or interrogation might be more close as priorities than say, polite conversation or idle conversation)

"TraceSim: A Method for Calculating Stack Trace Similarity" (Sep 2020)

- Paper works in the context of grouping bug-report triages. They are concerned with the automation of grouping similar reports and problems in order to improve efficiency in triage. This is of interest to us since their metric of grouping might prove insightful.

- Paper suggests that previous work in the field looks at grouping via information retrieval techniques, or string matching methods (both of which are probably not helpful to our case).
- The paper's approach is based on combining TF-IDF, levenshtein distance, and machine learning. This suggests to us that the method might not be directly useful to us, but at the very least, shows one of the popular ways that the field deals with notions of comparing traces (of information that are more complex/abstract than simple strings of data).
- A good takeaway though, is how they think about the problem and isolate the categories imo. It is split into two primary tasks: for a given report, find in the database similar reports and rank them based on likelihood of belonging to the same group (ranked retrieval), and distribute a given set of reports into buckets (clustering). That is to say, this slight nuance probably matters in the sense of how we think about approaching the issue, and we should be careful to take this into account when thinking about similarity metrics.
- Overall, the paper is still concerned about the problem of creating a good similarity metric, and thus, I feel the paper still useful.
- They suggest that TF-IDF and string matching has not been used together, and thus they aimed to do so in this paper, while also adding on top a ML element. They acknowledge that ML techniques are less consistent/stable, which is why they have not been really applied in this context, but they say that ML is more flexible, and can still provide good results, so they also add it to the mix. Hence their proposal.
- The algorithm is as follows: first they determine whether the stack trace is a stack overflow exception, since they say those often have repeated lines and can be better calculated using just TF-IDF. Then, if not, they go ahead and computes the weights for each line in the trace, since different lines have different impact to the similarity. Then the levenshtein edit distance is calculated.
- The takeaway from this algorithm application is this imo: first, we can have multiple ways to approach the similarity calculation, just like how they separated the case of SOE, where they know this specific type of problem will require a different nuance. Second, is that simple metrics of difference

such as levenshtein edit distance may already provide a lot of information in relation to similarity. However, there is also the big question on the weights being brought up. We did have this notion that the content of the trace may be important to the overall similarity/effect on our analysis, however, our preliminary starting point omits content in favor of form for now. However, I would say that this shows that eventually we will need to also move towards content-aware similarity metrics. But I agree for now we should focus on form.

- Interesting caveat to note, when they talk about weights, they talk about whether the line is closer to the top or bottom of the stack, which is something that isn't necessarily content based. In this sense, we can still hold onto our form-based similarity metric. For us, our context, we know that what is said in the middle of the conversation/trace should be more impactful than the beginning, and the end (beginning is usually greetings, ending is usually greetings, etc). This type of understanding is really important.
- They define a modified version of levenshtein edit distance, since they say that swapping order of lines is irrelevant. This will probably prove true for conversation traces in the case that we use the same edit distance concept. Good to keep in mind.

"Trace Clustering in Process Mining" (May 2014)

- Standard process mining (with van der aalst!), this time, talking about a trace clustering technique, which uses a number of distance metrics.
- This of course, will be up to the papers we've currently looked at, the most relevant in terms of what we'll be looking for I think. However, do note the paper is from 2014 so I'm sure there has been new techniques built on top of this. Regardless, this is still a good base to build off of.
- The method used is super straight forward: Every item in a trace has a specific numeric value. A profile is a set of related items, describing the trace from a perspective. Meaning that traces with similar behavior should have a similar profile resulting from the specific numeric values of the items

in the trace. This builds the definition of their trace profiles for comparison.
- The distance metrics used are euclidean, hamming, and jaccard.
- Clustering is done via k-means, quality threshold, agglomerative hierarchical clustering, and self organizing maps. We'll take notes on the ones we're not as familiar with.
- Quality Threshold is originally used in bioinformatic for gene expressions, and is like k-means, but you don't have to pre-specify number of clusters and is predictable (produces same clusters over multiple runs). The threshold is the maximum diameter of the cluster.
- AHC merges nearest traces to form clusters, and gives us a hierarchy (as the name suggests) of clusters.
- SOM is described as a neural network technique. Similar values are mapped onto the same node or neighboring nodes. Essentially a dimension reduction technique where you map high dimensional data onto a graph.
- I guess for this paper, it is important to note that they do clustering, meaning that they don't have specific archetypes that they're already looking for (Which might be the case for our context). Nevertheless, the description of the profile is definitely a great base for us in terms of looking at how we can define our traces such that we can apply similarity metrics.

"Trace Clustering on Very Large Event Data in Healthcare using Frequent Sequence Patterns" (Jan 2020)

- From the abstract, I picked this paper just because of the notion of frequent sequence patterns in healthcare data, since they mentioned that usual approaches had difficulty with healthcare data. I wanted to see what the difficulties were, and how they tried to resolve it, since I'm guessing it'll give me good insight on solving difficulties we'll inevitably come across for conversation data.
- One issue highlighted is that patients receiving the same diagnosis can be treated for different purpose (i.e: reconstructive breast surgery for breast

cancer patient versus gender change), the result is that the process models derived don't match with that of the clinician's intended direction.

- They outline the 3 challenges to trace similarity for healthcare data: 1. Scaling to large dataset is difficult, 2. The issue we mentioned above, expectations that traces in a group show homogenous behavior, whereas in healthcare, treatment for the same group can have disjoint paths, etc. 3. Feature metrics are usually used to represent similarity measures, and those are often unintelligible to the human. They also mention lastly that even in the same cluster, there are too many variations that creating a process model is difficult.
- Their solution? Frequent Sequential Pattern Matching. They compute this sequence pattern to learn the behavioral criteria of a group, and then determine based on ranking whether patients belong in a group or not.
- They do so based on getting a small sample set *P* provided by medical experts, for which they know definitely belongs to a group *C* of interest.
- More specifically, they use the group *C* as a baseline to sus out all the other traces in the group to determine whether or not that trace should be in the group. They define Frequent Sequence Patterns as a sequence that occurs within a trace with a frequency of no less than a certain threshold (arbitrarily defined). I guess for our context a sequence pattern is just a defined sub-conversation? Or once we get more descriptive we can have sub-sub-conversations? (i.e: argumentative conversation -> hostile argumentation, argumentative conversation -> supportive argumentation)
- Seems like they use this *C* group and train to find frequent sequence patterns via ranking. They describe this as finding the behavioral element of the *C* group. Which I guess makes sense. They're honing in on specific chains of actions, which is what a behavior is. In many ways, this is similar to the way we can think about how we define a sub-conversation. They are really just to a sequence pattern (i.e: respond.agree.1 -> give.opinion.1) that we look for.

Side note: we stashed a paper on using stutter equivalence to compress parity games! Interesting intersection! But not relevant to what we're considering now.

"Match2: A Matching over Matching Model for Similar Question Identification" (June 2020)

- This paper looks at similarity measures for natural language, which is of course related, even if at the content-level and not at the form-level in which we are working. Will provide us some key insight on how they go about the process.
- They measure the similarity of two questions based over the same answer. This conceptual explanation makes intuitive sense, and in a way, for us, grants some insight on how we could do pattern matching, since we deal with chains of behaviors, that could be responses to another chain of behavior, and thus the responses could give similarity measures to what they are responding to.
- The important distinction to note is that they state: "similar questions can be addressed by similar *parts* of the answer." Meaning that they're looking at a section of the answer that corresponds to the questions, in order to tell us the similarity of the questions. I.e: three points of comparison, in order to compare two objects (within relation to another. Sounds like the same concept as a landmark)
- Since that is the end goal, the idea is that there is some understanding of similarity between the Question stored and the corresponding answer. Then, the similarity metric is compared between the question in question and the stored answer. If the same parts are related, we know the question is probably the same. Intuitive.
- Seems like the paper entirely focuses on ML models to determine similarity between two objects, which isn't particularly helpful to us, but is a good reference point for the actions taken in the field.
- However, I still think the paper relevant in terms of applying the next step after we have a basic similarity metric for traces (in order to find similar traces, group, etc)

"Distributed Many-to-Many Protein Sequence Alignment using Sparse Matrices" (Sep 2020)

- I guess this is more abstract in concept, but our traces are not really that much different compared to protein sequences. This alignment using sparse matrices may be relevant to us in terms of evaluation of trace similarity.
- This paper seems a bit over my head atm, it references some methods by name that I am not familiar with. I'll have to check back on this paper.

"Development of distance measures for process mining, discovery and integration" (Jan 2001)

- We focus on the comparison metrics outlined by the paper
- One aspect they use is to transform the dependency graph into a numerical value to do euclidean distance.
- They represent this via a process matrix, which is just a matrix representation of the dependency graph (nodes and edges represented by a nxn matrix with 1s and 0s).
- They normalize the matrices for the two traces, and then do matrix subtraction, to find non-zero values, which denmark the points of difference.
- Then they have some more advanced calculations to denote degree of difference (using sum of the squares of the elements.
- But the basic notion makes sense.

"Context Aware Trace Clustering: Towards Improving Process Mining Results" (2009) **this paper has the most relevance for showing similarity metrics

- This paper specifically outlines some methods to clustering and the challenges that exist for clustering these traces, and seems to be from the basis of the literature (van der aalst strikes again)
- Bag of activities -> takes traces and turns it into a vector and the set of all activities defines the number of dimensions in the vector. The values of the vector correspond to the frequency of the activity. They suggest this method lacks context information, and lacks information about the order. Both of which may be important.

- K-gram model, which is like the bag of activities, but works on k-grams instead of singular activities. Again, turns it into a vector. The issue with this one is the increase in size.
- We have hamming distance, but the issue is that process traces rarely ever have the same length, and two of the same traces from a process might manifest differently, meaning it is too strict.
- Edit distance (levenshtein), similar to hamming distance, but the issue is again, traces are not really ever the same length, as well as the manifestation issue, meaning it is again too strict.
- Thus to make the edit distance concept more robust, they add the notion of a cost function and a context. Meaning to penalize edits that are unnecessary, in order to determine a better distance.
- They put limits on substitution and insert delete. Based on context. (In this case, context means 3-gram)
- The example, they have a list of traces, the three grams, and the context (those that appear surrounding) and then create a context set for each activity. Then the context set for two activities (where they share common context)
- Then they can calculate co-occurrence. The calculation is done relatively formulaically, based on whether we're looking at co-occurrence of like or unlike symbols, which yields different calculations for finding co-occurrence values based on frequency of the k-gram.
- The algorithm continues to do calculations for other properties, such as probability of occurrence of symbols, normalization, normalized co-occurrence frequencies, etc.
- The paper states this allows for the algorithm to find traces that are highly similar.

"Conformance Checking over Uncertain Event Data" (Sep 2020)

- Looks at implications for process mining if the traces contain uncertain data, specifically in their case, data that is a range of values rather than an absolute (I guess projected to our case, could be something like, % chance for each possible label).

- They outline possible causes of uncertainty: incorrectness, coarseness, ambiguity.
- Their goal is to establish some baselines in regards to conformance checking/etc on uncertain traces where the uncertainty is not an outlier but rather systemically part of the process.
- I think this notion might be important to think about since conversation data and natural language is one of the pillars of systemic uncertainty. Perhaps this will help us later down the road after we establish how we should think about distance/similarity.
- They separate uncertainty into two types: strong and weak uncertainty. Strong uncertainty is that the possible values it can take is known, but probability of distribution is unknown. Weak uncertainty is that both possible values and probability of distribution are known.
- They add another type of uncertainty: where we are uncertain whether the event recorded occurred at all, questioning the validity of the record.
- They solve the question: "given an uncertain trace and a petri net model, can you get an upper and lower bound for conformance?" They want optimal alignments, essentially, to determine whether or not the uncertain trace can fit certain realizations but not others.
- There's a brute force method, list all the possible uncertain traces and then calculate all alignments.
- However, their approach is obviously more efficient. They define a behavioral net, which is a petri-net that can replay all realizations of uncertain traces. That is done via creating a dependency graph first.

Draw some preliminary conclusions

Based on the papers we've collected, it seems that we have a basis for establishing notions of similarity and distance between two traces, based on modified edit distances, context, etc. These starting points should be enough to give us an idea of whether we want the set up to be such that we define archetypes and search for them (and maybe their probability of occurring) or we identify common patterns in given traces and determine the closest match to what archetypes appear the most as sub-conversations. It seems both are relevant and useful, and

may pose useful for language learning (in a way they are "predicting" or giving a heads up on the context of the conversation. I.e: if i know that the archetype of the conversation is argumentative but not confrontational, I can assume that it is similar to a debate, where a deeper knowledge of vocab might be necessary. Versus if we're engaged in a nonconfrontational shallow conversation about an arbitrary topic, i can better succeed if i determine i know the vocab for the topic required, or something to that effect).

From our search, seems like there are two papers to get deeper on, first being the 2009 paper on context aware clustering, not for the reason that the method is particularly new, but rather that it serves as a good basis, we should use some of the notions and find more recent papers to see if the notions of distance/similarity has changed or seen new updates!
Second being the uncertainty paper. It seems like it will be the most relevant to the nuance and variations that we could deal with.

Next step should be finding some sample data we can steal.
(Meaning that we have access to the text, and relatively simple access to the text for manipulation, so we can add our own labels).
We found some big datasets, we should rummage around to see if we can easily use it for our purposes.

For the data from Taskmaster 1, seems like we might want to change the labels to include things like requests? Or I dunno, seems like an assistant-user request oriented conversation is very different from a casual conversation at a LE… We might want to think about getting data from elsewhere or….

Additional Notes on Conformance Checking Uncertain Events and Context Aware Traces

- Conformance via alignments is done by finding deviation from the trace executed and the firing of transitions in the petri-net.
- Literally based on comparing whether or not a trace can be replayed on the petri-net move by move. If not possible either way, a "no move" symbol is

added. Then the final phase, all “no move” symbols are removed, and the alignment is successful if top row corresponds to a trace in the log, and the bottom row corresponds to a firing sequence in the petri-net.

- Since there are many alignments, they use cost function to determine the best one.