1 The autumn semester & catchup

The spring semester journal ended quite abruptly, as a global pandemic washed over our lands. I was forced to spend my time keeping up with my work, and unfortunately had to downprioritize the journal for the last few months. I hope to keep it up for the remainder of my time here at the university, but probably on a monthly basis. This will be the august entry.

The topic at hand has become quite a lot clearer for me after learning the different methods and architectures of NLP-focused deep learning, and writing and submitting an essay on my understanding of the topic to my supervisor in may. I have also started to use (and appreciate) LaTeX, and will be utilizing that going forward.

2 Data analysis

2.1 DSTC 2

The first thing that I deemed necessary was to get to know the dataset better. For me to utilize weak supervision, I would have to gain some knowledge of both the structure and the content of the utterances by the users. My first method was to generate *n-gram*-models of the utterances themselves. I was interested in seeing how the typical utterances were formed, which words and phrases were uttered the most often, and if I could use some inductive thought to create some regular expressions to catch the least complex parts of the data. I am not planning on using these models as voters as I think they are too volatile to noise, but they did provide me with some valuable insight.

The first thing I noticed is that for the three most common slots, *area*, *food*, and *pricerange* were all relatively easy to distinguish. The following factors were very easy to spot:

- The cardinal directions are used almost exclusively as a mean to describe the location of the restaurant they want suggested. This is not too strange, as the user is given a specific goal at the beginning of the dialogue, where the cardinal direction is given. Naming locations does not occur. This makes identifying both the occurance of the slot and the value given trivial.
- The pricerange is very often explicitly given, in a very specific way. It is very rare that if the pricerange is given, it is given through an expression including *cheap*, *expensive*, or *moderately*.
- The type of food is also very often specified by using a type from the

provided ontology, followed by the word *food*. This is slightly less accurate than the two other slots, but still fairly simple.

Utilizing these observations, and some other instinctive knowledge of the structure of the dialogue gained from the *n-gram*-models, like the use of *dontcare* for the *this*-slot, or the fact that almost everybody asked for the phone number, address, or post code by mentioning the mean of contact by using the terms explicitly, I was quickly able to achieve 80% accuracy on the development data. I generated an ordered collection of the examples that were not correctly labeled, or missed by the expressions, and with some minor tweaks was able to achieve 90% accuracy. The main complexity of the remaining examples were due to information that I had not used in my expressions, like previous utterances in the dialogue. Seeing that models on the dataset achieved scores of around 97% accuracy, it was clear that this is not where the complexity lies in the data. The same models mentioned earlier achieved 74% accuracy on the *goal-labels* measured by a binary match, which is obviously a much more difficult task.

This did however raise a question - is the core information in the data too trivial for the application of expert knowledge? The goal of the thesis is to see if some heuristics can be applied to the data, and if most of the utterance-wise coverage can be found by regular expressions created over a few hours, most of the complexity will rely on the structural information in the dialogue. This structural information *is* an important part of weak supervision, but the lack of application of domain knowledge is somewhat troubling. Due to this discovery, I have looked at utilizing another dataset for the thesis - the *Maluuba Frames* dataset by Microsoft.

2.2 Maluuba Frames

The *Maluuba Frames* - or just *Frames* - dataset is a relatively new slot- based dialogue corpus. It has a lot of simlarities to the DSTC dataset, it contains utterances from a user that is looking for some service, with restrictions to price range and area. It is different in that the user is often asking for specific locations, and prices are defined continuously. The responses by the program may also suggest other places, unlike *DSTC* which only responds. There is a lot of information that I cannot find a use for right now, but that might be interesting at a latter point in time, such as reasoning over how to process the request by the user. Structurally the dataset is also very different - both the mean and median length of the dialogues is 14, or twice the amount of turns for the *DSTC* set.

One of the reasons that the *Frames* dataset is interesting is that it opens for the use of external knowledge bases. By having real life places mentioned in the utterances by the users, one could easily imagine the opportunities of

using that information with for example *Wikidata*. One possibility is generating *SparQL*-queries from information found in user utterances, to gain some greater understanding of the dialogue as a whole. All in all, there are many interesting opportunities that I would like to explore.

I have not properly explored the *Frames*-dataset yet, but the first thing that I have done is to separate a test set, that will remain untouched untill the final experiments. This test set consists of 15% of the original dataset.

3 Slot detection

I spent some time reading up on and experimenting with measures for slot detection. I have been looking at utilizing *Frequency Distribution*-models to use as voters. The implementation I have created only detects the presence of a slot, but does not consider any value. That will be further discussed in Section 3.1. A dictionary is generated for each slot, that contain all the words that occur in the utterances that have been labeled with that slot. For a Corpus $C = \{U_1, \ldots, U_n\}$, where each utterance U is defined as $U_i = \{w_1, \ldots, w_m\}$ where w_i is a word that occurs in the utterance, the set of slots $S = \{s_1, \ldots, s_k\}$, where each s_i is the name of a slot, and the function $\tau : U \rightarrow S$ maps some utterance to some slot, the *Frequency Distributed*-value for a word and a slot is given by equation 1.

$$FD(w, s) = \sum_{U_i \in U} \sum_{w_j \in U_i} \forall_{s'} \in \tau(U_i)(P(s|s')P(w|w_j)) \tag{1}$$

What we want to achieve is a function that will tell us if a given word is significant for a given slot. We find the proportionality of a word given a slot with equation 2. Laplacian smoothing is used in the case that the word has not been observed with a slot.

$$Prop_{FD}(w, s) = \frac{FD(w, s) + 1}{(\sum_{U_i \in U} \sum_{w_j \in U_i} FD(w_j, s)) + 1} \tag{2}$$

We can then use equation 3 with some given threshold ϵ to see if the word is representative for the slot. For a given slot s, word w, and threshold ϵ, if $Repr_{FD}(w, s, \epsilon) = k - 1$, it is significant. While the results are acceptable when using all the words of an utterance, with an accuracy of 74% when using $\epsilon = 0.1$, much can be explained by the distributions over the slots. There are many more occurances of the three main categories *food*, *area*, and *pricerange*,

than the others. When only comparing them to each other, the accuracy drops to some low 20%.

$$Repr_{FD}(w, s, \epsilon) = \sum_{s' \in S} \{s \neq s'\} \begin{cases} 1, & \text{if } Prop_{FD}(w, s) \geq Prop_{FD}(w, s') + \epsilon \\ 0, & \text{else} \end{cases} \tag{3}$$

What can instead be done is to generate a mapping from the distribution $\lambda : W \rightarrow S$, where any word $w \in W$ that is not significant for any slot is discarded. Then for an utterance, we can use the parameter γ to determine how much of the utterance needs to be significant for one slot over any others. Training a model on 5% of the labeled data using equation 4 with $\gamma = 0.2$ I achieved 70% accuracy on the three main categories, albeit much higher for area than the others. Some future research will go into finding ways of improving the two other categories.

$$UtteranceRate_{FD}(U_i, s, \gamma) = \frac{1}{|U_i|\gamma} \sum_{w \in U_i} P(s|\lambda(w)) \tag{4}$$

3.1 Further implementation

- I want to find a way to use the hypotheses included in the *DSTC* dataset for slot detection. There is probably some interesting information to be used there.
- Explore the *Frames* dataset further, create some regular expressions and test some Frequency distributions on it.
- Find other models that can be used for slot detection, for example models from sklearn.
- Experiment with small subsets of the different sets for these models and the frequency distributions to see how much data is required for some decent results
- It might be interesting to see if equation 3 can be useful for values other than $k - 1$.