

Scoping, CLF, SVP

2/3/2020

Objectives

- Learn about the basic considerations involved in scoping AI
- Understand how to train a classifier
- Understand how to train a slot value pairing model

Identifying a Good Problem

- Conversational
- Broad and deep
 - Scope covers the subject area thoroughly
 - Scope allows for more than a single ask and response
- Value adding to the user
 - Why will people use this?
- One that you can measure the success
 - What define VA success for you?
- Solves a real user need
- Provides value to the end user

Discussion Topic - Conversations

Think to the last time you spoke to a subject matter expert the AI will be augmenting...

- What did you ask them?
- Where did the information come from?
- Was it expert knowledge?
- Was it procured from a system or service?
- Did they have follow up questions? What were they?
- Did you have follow up questions? What were they?
- Was there was clarification or context that you didn't provide at the first turn of the conversation? How was this handled?

Scoping Steps

1. Broad Scoping of HitR Experience
2. Competency Design
3. Conversation Design

Broad Scoping

1. Start with a problem to solve
2. Create a list of final actions for the use case
 - a. Identify the key results you're aiming to accomplish
 - b. Focus on the goal of the user
3. Detail out example conversations, both utterances and responses
 - a. What is the ideal human in the room result
 - b. Think through edge cases, follow ups, clarifying questions
 - c. Ask what might a person ask as a part of achieving the actions

Competencies Design

1. Evaluate what can be captured into slots
 - a. What are the key parts of the utterance (typically not the action itself)
 - b. What are the word(s) that can be semantically determined
2. Identify intents based on slots necessary for the action
 - a. What are the high level actions that the user has with each utterance
 - b. Think about how the usage of slots can help aggregate granular intents
3. Define states to support intent structure
 - a. How do the intents allow for traversal over a minimal number of states
 - b. Think about fallback responses and the states support intelligent responses
4. Group states into competencies that accomplish the end goal

Conversation Design

1. Detail out necessary transitions that would enable context retention
 - a. What competencies are related to each other where context should be retained
 - b. What are the slots that can be shared from one competency to another
2. Script response logic to support all scenarios
 - a. Identify a fallback response for each state
 - b. Account for all variations of slots being filled
 - c. Think about how business logic will need to be structured

CLF & SVP Overview

The Query Lifecycle

The **Query Lifecycle** consists of: Classification, Slot Value Pairing, Slot Mapping, and Business Logic.

 **Utterance**

I would like to check the balance in my checking account.

 **Response**

Sure. The balance in your checking account is \$18,879.78.

 **Classification**

 **Slot Value Pairing**

 **Slot Mapping**

 **Business Logic**



Activity: CLF and SVP

Suppose you are designing a conversational AI application to handle drive-through orders at a quick-service restaurant. You have the following intents and slots:

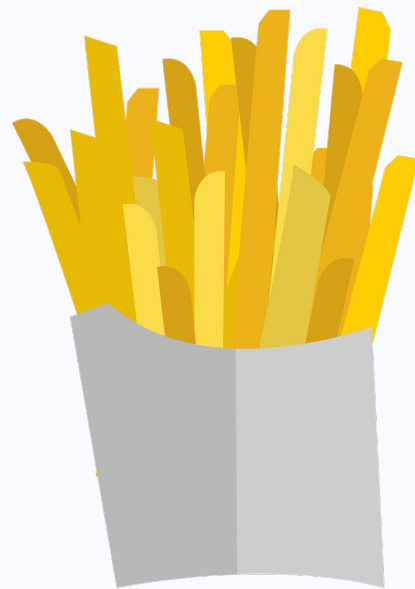
Intents:

- order_food_start: User would like to place an order.
- check_ingredients_start: User would like to ask a question about the menu.

Slots:

- food_item: The item that the user is inquiring about.

For the following utterances, complete classification and slot value pairing.



Activity: CLF and SVP

Utterance 1:

Tell me more about the **truffle fries**. Does it contain real truffles?

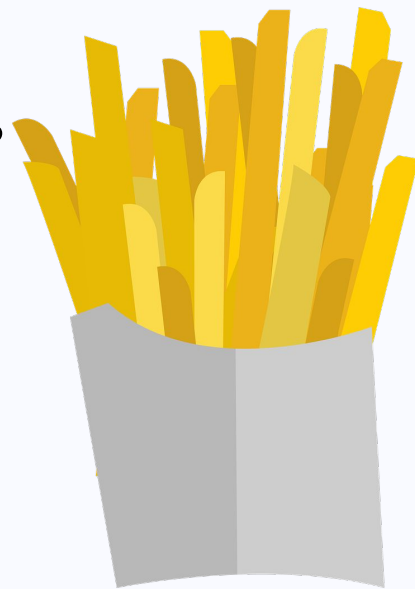
Intent:

order_food_start

check_ingredients_start

Slot:

food_item



Exercise: CLF and SVP

Utterance 2:

I'm vegetarian, so can I get the **veggie lovers**?

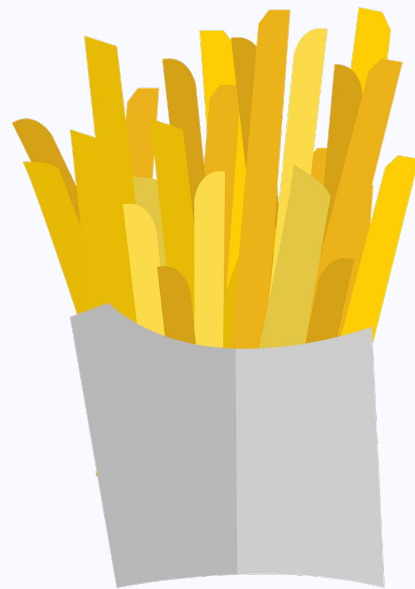
Intent:

order_food_start

check_ingredients_start

Slot:

food_item



Exercise: CLF and SVP

Utterance 3:

Do you have a **gluten-free** menu?

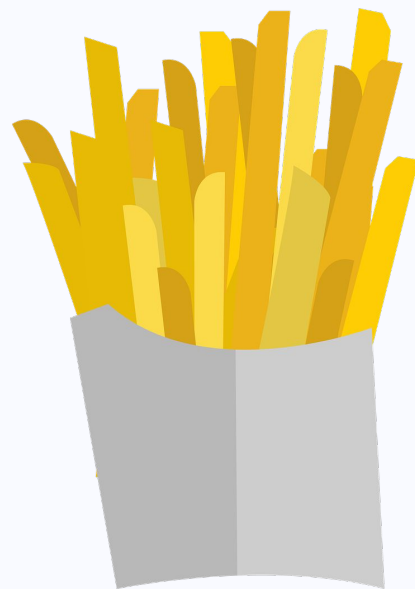
Intent:

order_food_start

check_ingredients_start

Slot:

food_item



Machine Learning

ClinC uses **machine learning** to do classification and slot value pairing.

Similar to how humans can learn from experience, computers can learn from data.



Machine Learning Overview

Machine learning starts with **training data**, which is data representing the intended outcome.

Classification

Training Data: Utterances
Prediction: Intents

account_transfer_start

- Let's transfer some cash!
- I would like to transfer \$100 from checking to savings
- Send \$200 from savings to checking
- Can I complete a transfer?
- Add \$700 from my savings to checking

check_balance_start

- What is the balance in my savings account?
- Give me my checking account balance.
- Balance?
- I would like to check the balance
- Savings account balance?

Slot Value Pairing

Training Data: Utterances -> **Tokens**
Prediction: Slot Labels

amount

source_account

destination_account

- **\$100**
- I would like to transfer **\$100** from **checking** to **savings**
- Send **\$200** from **savings** to **checking**
- Add **\$700** from my **savings** to **checking**
- The destination will be **savings**
- Can I transfer **\$2,000** from **checking** to **savings**?
- Let's transfer **\$1,000**.
- Let's transfer some cash!
- transfer **\$19,999** between my accounts
- let's transfer **\$20,000.00** to my **savings**
- transfer **30 bucks** from my **savings** to the other account
- transfer money from **checking** to **savings**

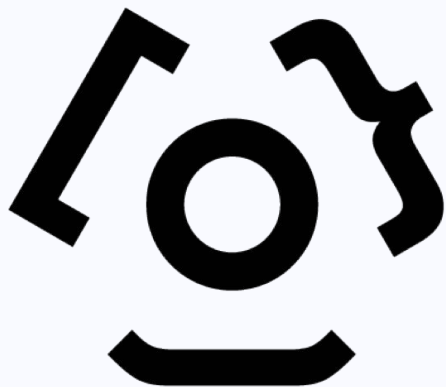
Machine Learning Overview

During **model training**, the ML model learns from patterns (sentence structures and vocabulary) within the training data.



Machine Learning Overview

Once a model is trained, it will use the patterns it learned to make predictions on new data.



Status
Trained

account_transfer_start

Transfer **\$2,500** of that to my **savings**.

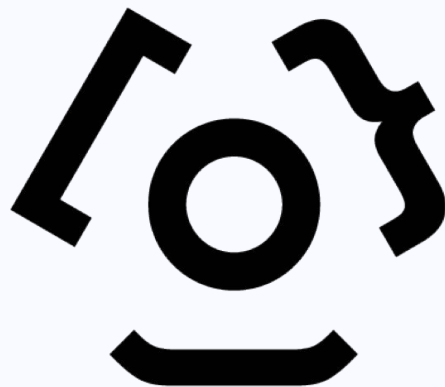
check_balance_start

How much money do I have in my **checking** account?

amount
source_account
destination_account

Machine Learning Overview

The ML model also outputs the level of certainty with each prediction (prediction probability).



Status
Trained

account_transfer_start

Transfer \$2,500 of that to my savings.

0.4775

0.8518

0.6587

check_balance_start

How much money do I have in my checking account?

0.8371

0.9686

amount

source_account

destination_account

Machine Learning Overview

Prediction probabilities must be above a certain threshold value.

Classification

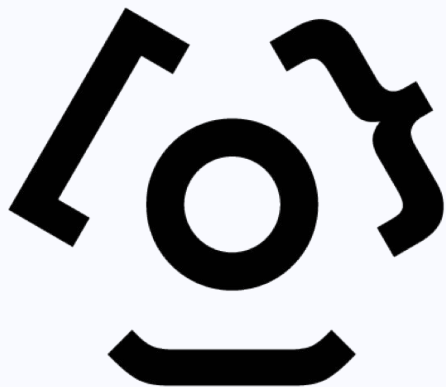
If **intent probability** is below the threshold:

The intent is **out of scope**, and Clinc returns to the root state.

Slot Value Pairing

If **slot probability** is below the threshold:

The slot value is **NULL**.



Status
Trained

account_transfer_start

Transfer **\$2,500** of that to my **savings**.

0.4775

0.8518

0.6587

check_balance_start

How much money do I have in my **checking** account?

0.8371

0.9686

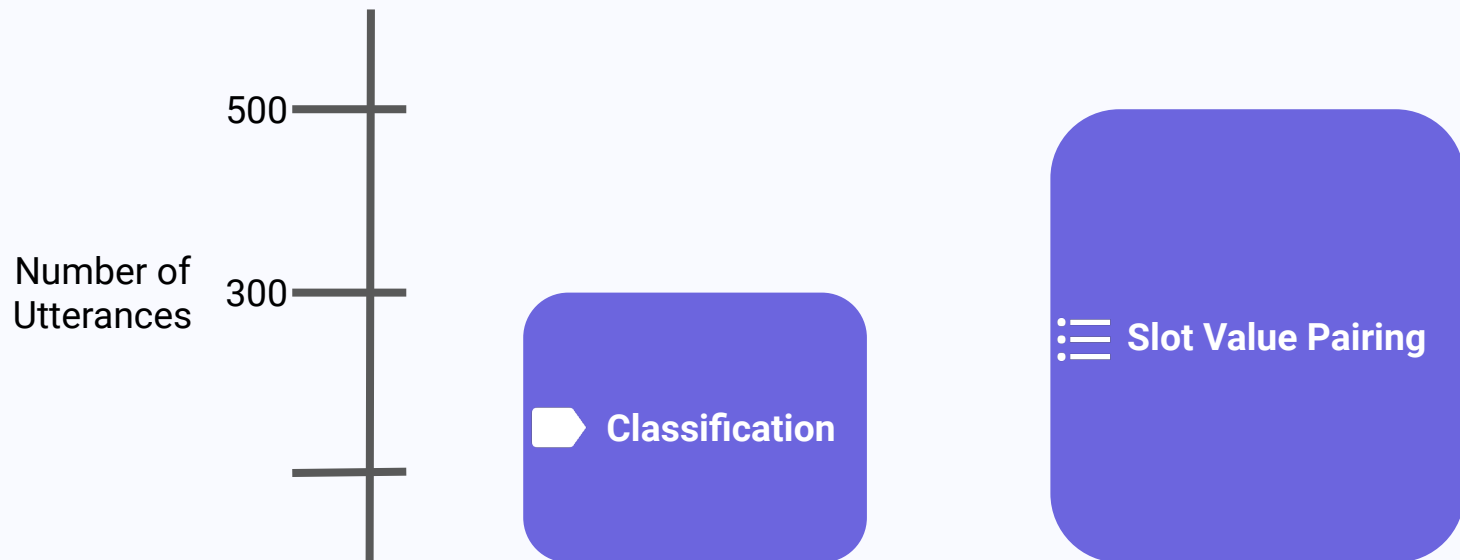
amount

source_account

destination_account

How Much Data is Needed?

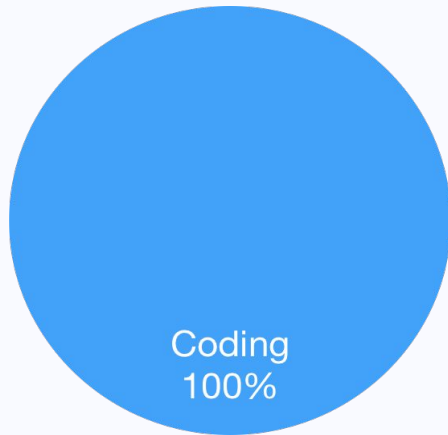
Generally, you need ~300 utterances for each intent, and ~500 utterances for each slot.



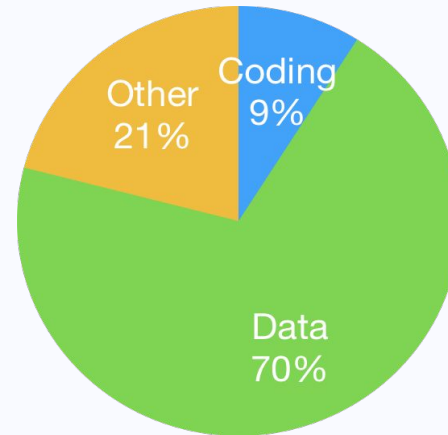
Data Collection

Data Collection

Data collection and curation is one of the most significant aspect of conversational AI development.



What people think about AI



The reality*

**Figure eight (Crowdfunder) 2017 Data Scientist Report*

Data Collection Methods

There are three main ways to collect data:

- Manual data entry
- Data import
- Crowdsourcing

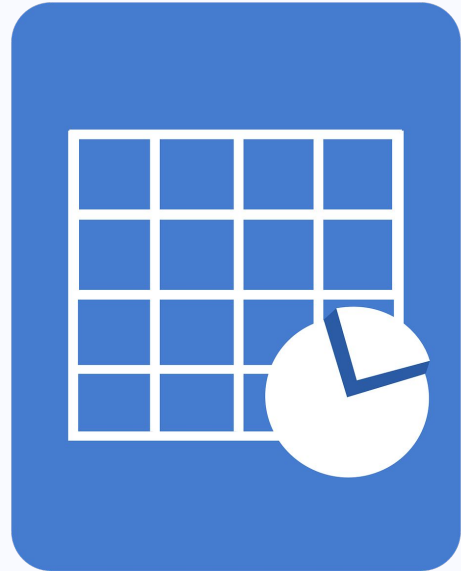
Data Collection Methods

Manual data entry is the most time consuming to gather data for production, and it won't produce robust conversational AI applications, but can be used to quickly validate a flow



Data Collection Methods

Importing data is a good option when you have existing data (i.e. call logs)



Data Collection Methods

Crowdsourcing is generally the most practical way to acquire quality data.



Crowdsourcing Best Practices

Keep instructions concise

Imagine that you are seated inside of a restaurant on a Sunday afternoon with your wife and two kids. After 10 minutes of waiting, a waitress shows up to take your order. After your wife orders a salad, your 2 kids order off of the kids menu. Finally, it is your turn to order. What do you say?

What are ways you would provide your food order to a waitress in a restaurant?

Crowdsourcing Best Practices

Provide diverse examples

- I would like to order a grilled cheese.
- How about the burger?
- Some fries, please?

- I would like to order a grilled cheese.
- I would like to order a burger.
- I would like to order fries.

Crowdsourcing Best Practices

Avoid domain-specific terminology

Provide ways to order food at a restaurant.

Provide utterances related to food-ordering in a restaurant.

Crowdsourcing Best Practices

- Keep instructions concise
- Provide diverse examples
- Avoid domain-specific terminology

More crowdsourcing jobs
with fewer utterances



Fewer crowdsourcing jobs
with many utterances

Activity: Data Curation

Suppose you are training a robot babysitter to identify healthy foods for your toddler to eat.

Examine the following training data sets, and identify the issue.



Activity: Data Curation

Healthy

Carrots

Unhealthy

Carrots



No differentiation between healthy and unhealthy

Activity: Data Curation

Healthy

Carrots
Carrots
Carrots

Unhealthy

Chocolate Chip Cookies



Not enough data diversity

Activity: Data Curation

Healthy

Carrots
Cucumber
Lettuce

Unhealthy

Chocolate Chip Cookies
Soda Pop
French Fries



Not enough data

Activity: Data Curation

Healthy

Carrots
Cucumber
Lettuce
Apples
Watermelon
Vanilla Ice Cream
Exercise
Plain Yogurt

...

Oatmeal

Unhealthy

Chocolate Chip Cookies
Soda Pop
French Fries
Sport Drinks
Eggplant
Apple Juice
Watch Cartoons
Cheesecake

...

Bacon

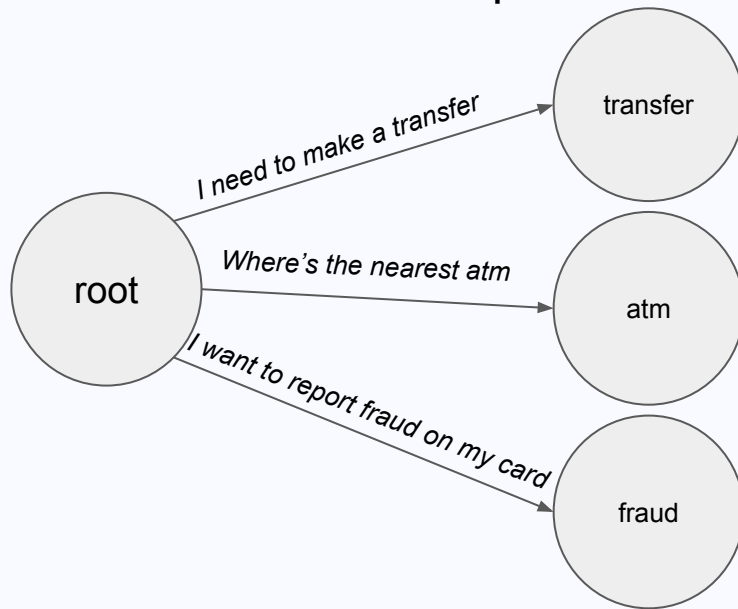


Incorrect Data

CLF Data

Classification

- Intent classification tags a user's query into one of N distinct labels
- Data for classification is a set of representative utterances of what users may say and labeled as how it should be interpreted



Common CLF Issues

- Common problems:
 - Noisy / Messy Utterances
 - Consistency across intents
 - Overfitting on the wrong words

Curating Slot Data

- **Noisy / Messy Utterances**

- CLF models train on the tokens and patterns of tokens represented in training data
- Keep your training utterances on topic and don't include multi-sentence unless needed
- Good sanity check: Try utterances that should be out of scope or with multiple sentences
 - I've been short on cash lately, ~~how much can I withdraw at once?~~
 - I'm going out to dinner tonight. ~~I need to transfer money to my checkings account.~~

- **Solution:**

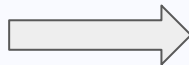
- Consistency is important. If noise is not equally represented across intents, you can easily over fit noise to intents
- Use the CLF Data Insights to see if unexpected tokens are highly impacting a specific intent

CLF Consistency

- **Consistency across intents**

- There is a CLF model at each state. These models are influenced by all outgoing intents from that state
- When collecting and curating data, the impact of all the intents must be incorporated
- Good Sanity Check: Ask yourself what new information you're teaching the classifier with each new utterance

<i>IntentA</i> Training Data
“How do I...”
“How do I...”
<i>IntentB</i> Training Data
“How can I...”
“How can I...”



Potentially overfit on the early pattern instead of the actual intent

CLF Overfitting

- **Token Overfitting**
 - Since CLF data is typically collected prior to slot data, it can be easy to overfit to slot values
 - Solution:
 - Make sure to diversify what tokens are used in slot data as well.
 - This also matters across intents.
 - There's 2 ways to correct overfitting
 - Add tokens across all relevant intents
 - Reduce occurrences of highly weighted tokens

Classifier Weights

Classifier Weights help you diagnose issues in classification data

- Utterances with high influence index terms will more likely classify into that intent.
- Terms that can appear in multiple intent (such as stopwords) should not have high influence index.

	Influence Index
account_transfer_start	
give	0.48
withdrawal	0.42
left	0.37
after	0.34
in	0.26
Show More	
check_balance_start	
's	0.43
my	0.43
i	0.41
account	0.39
balance	0.39
Show More	
clean_goodbye_start	
later	0.68
goodbye	0.57
bye	0.57
long	0.38
so	0.37
Show More	
clean_hello_start	
hello	0.68
hi	0.56
suh	0.56
good	0.37
day	0.37
Show More	

Example: Classifier Weight

Data Insights ×

Within the state root, you have 10 total utterances. The influence index of *specific words* is displayed below:

show_account_number_start	Influence Index
number	0.4
my	0.34
checking	0.3
savings	0.25
account	0.25

[Show More](#) ▾

show_balance_start	Influence Index
balance	0.37
balances	0.24
bank	0.22
say	0.2
does	0.2

[Show More](#) ▾

? Words With A High Influence Index Hold The Most Weight In Your Data Set Close

Arbitrarily high importance relative to the intent's purpose

May need utterances with "savings" and "checking" to reduce overfitting with show_account_number_start

SVP Data

Gathering SVP Data

- SVP takes the form of labeled utterances
- Labeling can be done either manually or through crowdsourcing
- If crowdsourcing, it's important to provide good examples of proper labels

Ticker

Q: What is the P/E Ratio of AAPL?

Give me P/B Ratio for GOOGL

Show me MSFT's PEG Ratio

Metric

Tell us the Revenue today for AMZN

Display current Beta at MZLA

How much Free Cash Flow does YHOO have?

What's FB's last reported EBITDA?

Common SVP Issues

- Common problems:
 - Keyword vs. Context
 - Diversity
 - Consistency

Curating Slot Data

- **Keywords vs. Context**

- SVP models rely on both the tokens as well as the surrounding sentence context
- Be careful not to overfit SVP models on specific keywords
- Good sanity check: Try queries with slot values that shouldn't be labelled!
 - “Do you mind *checking* my *checking* account for me”
 - “I’m not sure if I’ve been *saving* enough these days, what’s my balance on my *saving* account?”
 - “I have like *12* different accounts, what’s the balance for the one ending in *12*?”

- **Solution:**

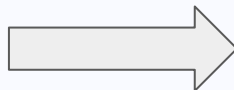
- Unlabeled data matters!
- Make sure your unlabeled data set contains examples of common words that should not be labeled

Slot Diversity

- Diversity

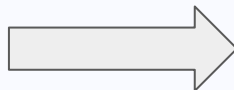
- SVP models rely on both the tokens as well as the surrounding sentence context.
- Make sure tokens seen during training are well distributed

Training Data
“ <i>checking</i> account balance”
“how much \$ do I have in my <i>checking</i> account?”
“ <i>checking</i> account money amount”



“What’s my balance for my *saving* account?”

Training Data
“ <i>roth ira</i> account balance”
“how much \$ do I have in my <i>vacation</i> account?”
“ <i>emergency</i> account money amount”



“What’s my balance for my *saving* account?”

Label Consistency

- Consistency
 - Make sure tokens are labeled consistently!
 - Model may get confused otherwise

Training Data

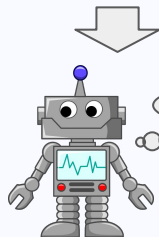
"*checking account* balance"

"how much \$ do I have in my *saving* account?"

"*retirement account* money amount"

"what's the balance for my *secondary* account?"

Model unintentionally tries to learn unexpected + unnecessary patterns/rules.



I'm confused, so if the sentence starts with an account type, then I should be labeling the word "account" but otherwise don't?

Slot Data Distribution

- Slot data distribution provides the number of times a token appears as a slot in training data.
- The tool is helpful for
 - Prevent biased SVP model training data
 - Analyze or debug SVP models behavior

Dest Source

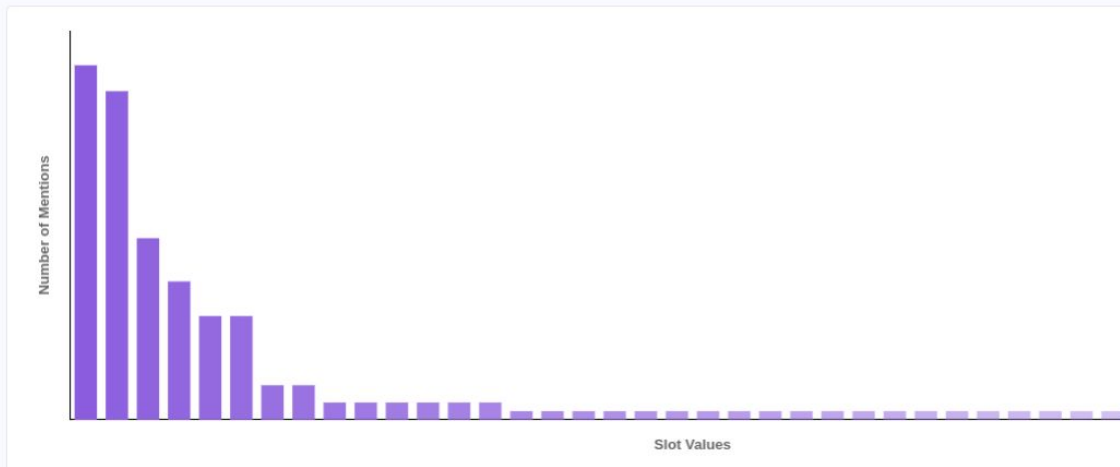


Slot Data Distribution - Preventing Poor SVP

Models

- A robust SVP model requires:
 - Uniform distribution
 - Diversity in slot values

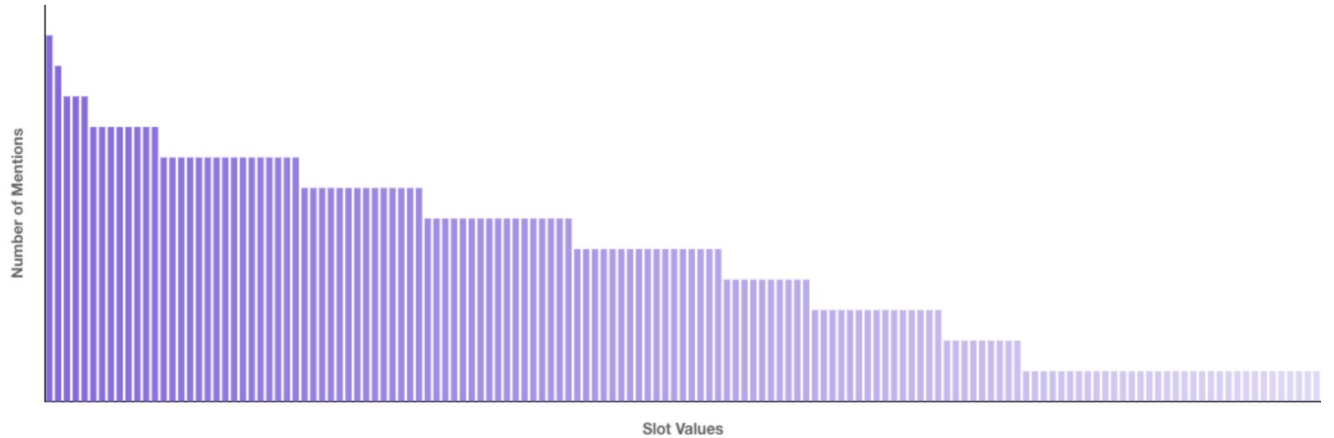
Company



Company Name	%
google	22.04
apple	20.43
ford	11.29
facebook	8.602
twitter	6.45
microsoft	6.45
walmart	2.15

Slot Data Distribution

This is a close to ideal distribution graph.



TOP 5 VALUES	# OF MENTIONS	BOTTOM 5 VALUES	# OF MENTIONS
1. Chesapeake Energy	12	140. Microsoft	1
2. Microsoft	11	141. McDonald's	1
3. Exxon Mobil	10	142. Best buy	1
4. AMD	10	143. Purdue Pharma	1
5. Nintendo	10	144. pharma	1

Slot Data Distribution - Example

- “What’s the market cap of Berkshire Hathaway? I checked on **Google** earlier but couldn’t find it”.
Why is google being extracted?
- Analyze SVP Models Behavior
 - False-negative: Debugging either it’s unbalanced slot **distribution** or lack of slot **diversity**
 - False-positive: Overfitting to words can lead to always extracting them as slot values.

Scoping Considerations

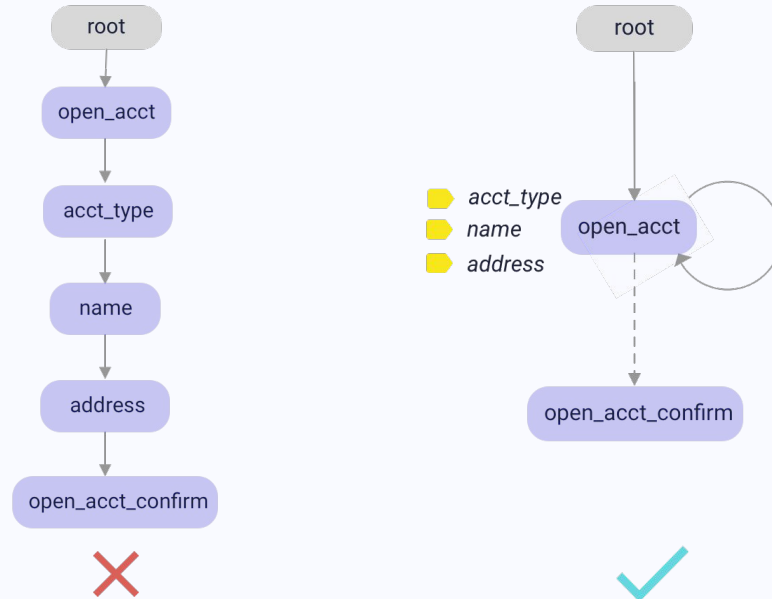
Competencies - Slot Value Pairing - Scaling

- Slots and states (when it's single state competency) scale inversely
- More states means less slots

	Many slots	"Middle"	Fewer Slots
Competency	order_vehicle	order_uber, order_lyft	order_uberXL, order_uberX, order_uberS
Slots	Vehicle type	Vehicle type	Start location
	Company	Start location	Destination
	Start location	Destination	
	Destination		

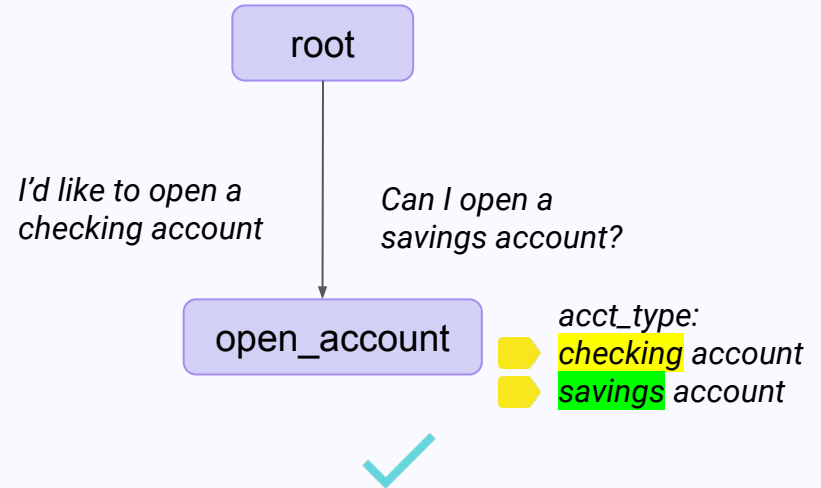
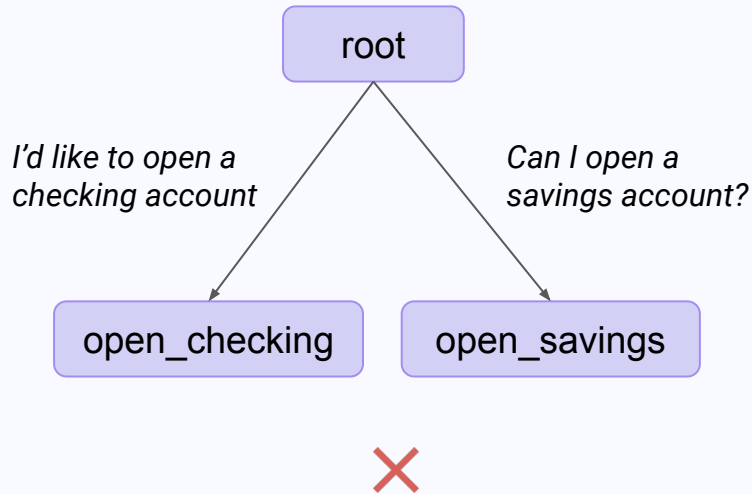
Common scoping problems to look out for

Category	Problem
States	Using too many states (rigid conversations)



Common scoping problems to look out for

Category	Problem
Classification	Designing with overly similar classes

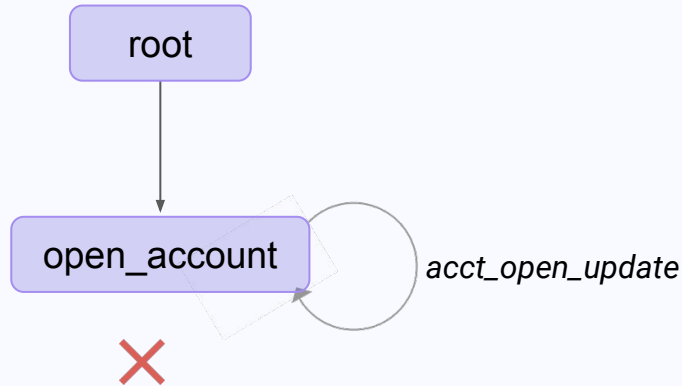


Common scoping problems to look out for

Category	Problem
Classification	Not considering the AI's prompt, and subsequent response

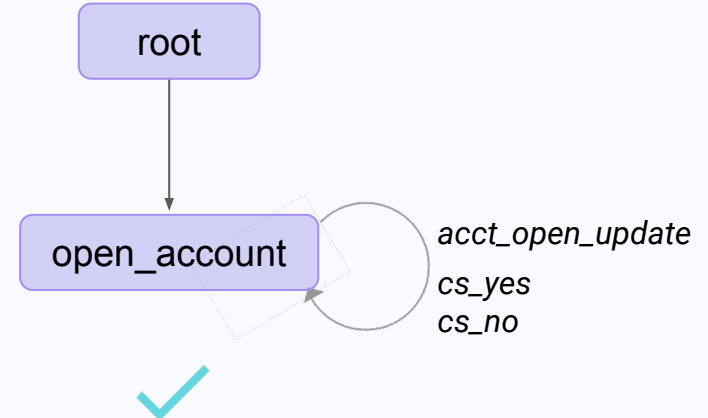
Q: I want to open an account

R: OK, what account do you want to open?



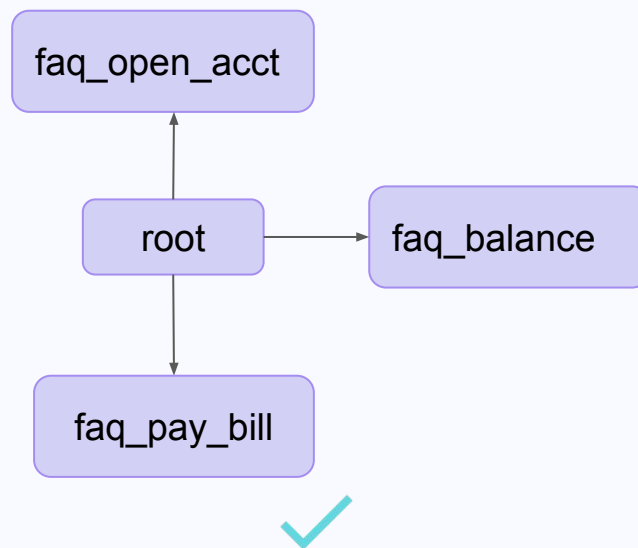
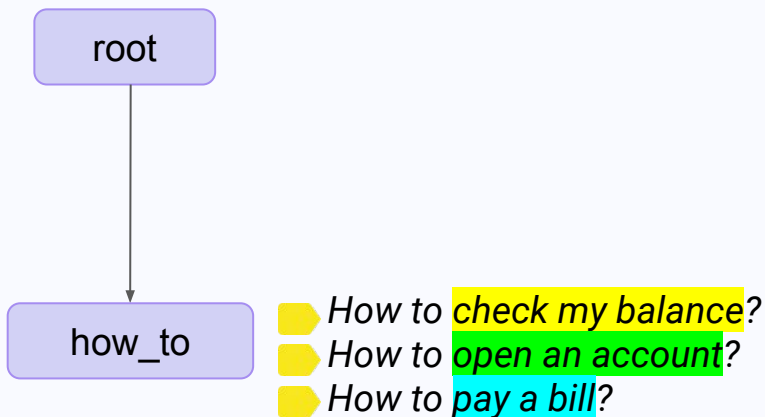
Q: I want to open a checking account

R: You want to open a checking account. Is that correct?



Common scoping problems to look out for

Category	Problem
SVP	Requiring too many words to be labeled



Common scoping problems to look out for

Category	Problem
SVP	Designing semantically ambiguous slots

“Can I get a tall vanilla coffee”
size flavor drink_type

“Can I get a vanilla tall coffee”
flavor size drink_type



“Can I get a tall vanilla coffee”
tall ← *small, medium, grande, large, venti*
vanilla ← *hazelnut, caramel, chai, skinny vanilla*
coffee ← *latte, cappuccino, tea*

