Supply Chain Bot for Retailers

Niharika Krishnan
Sridhar Rathinasabapathy

Before we get started..

NIHARIKA KRISHNAN

- ➢ Currently working on NLP based solutions for retailers at TCS
- ➢ Organiser @ PyLadies Chennai

SRIDHAR RATHINASABAPATHY

- ➢ Sridhar is currently working on NLP and Machine learning solutions for retailers at TCS.
- ➢ Worked on Machine learning and Deep learning based solutions for the last 5 years
- ➢ 15+ years of retail industry experience

niharikakrishnan

sridhar-rathinasabapathy-46320b10

Agenda

- Introduction to chatbots
- AI in Retail
- Supply chain use-case
- Tech stack used
- ML components
  - Training data
  - Data pre-processing
  - Intent Recognition
  - Automated Annotation Pipeline
  - Named Entity Recognition
  - Context awareness
  - Bot as a web service
  - User feedback & Model retraining
- Conclusion

Introduction to Chatbots

- Simulates a human conversation
- Scripted Chatbots → Intent Recognizers → Virtual Agents → Human-like Advisors
- **What drives a BOT Market?**
  - Advances in AI & ML
  - Availability of high compute
  - Bot frameworks available
  - Rising demand for self-service
- **Commercial Bot Frameworks vs Open Source**
  - Amazon Lex
  - Google Dialogflow
  - Azure Bot Service

AI in Retail

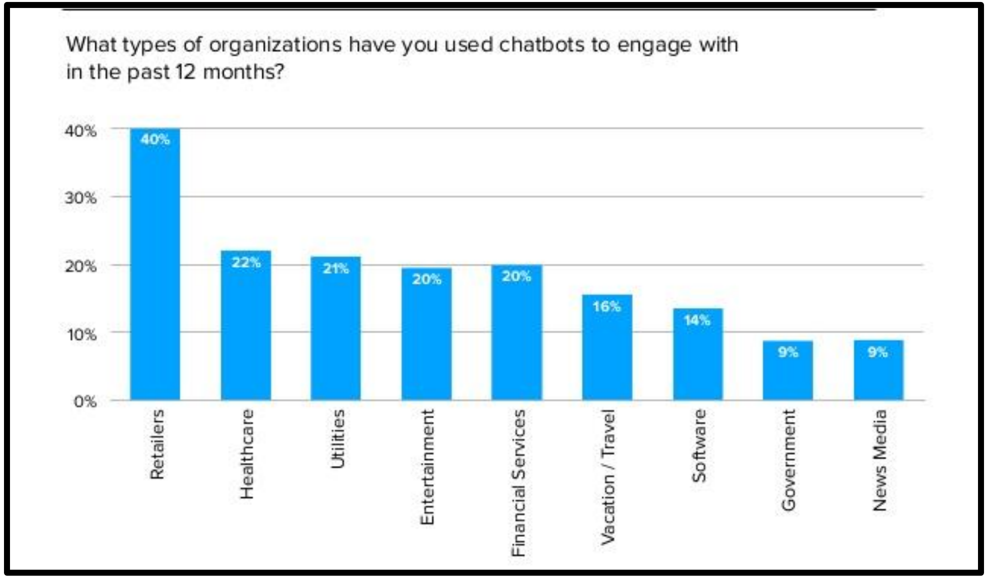
AI Market Size

2018 → USD 9.5 Billion

2025 → USD 118.6 Billion

WHY?

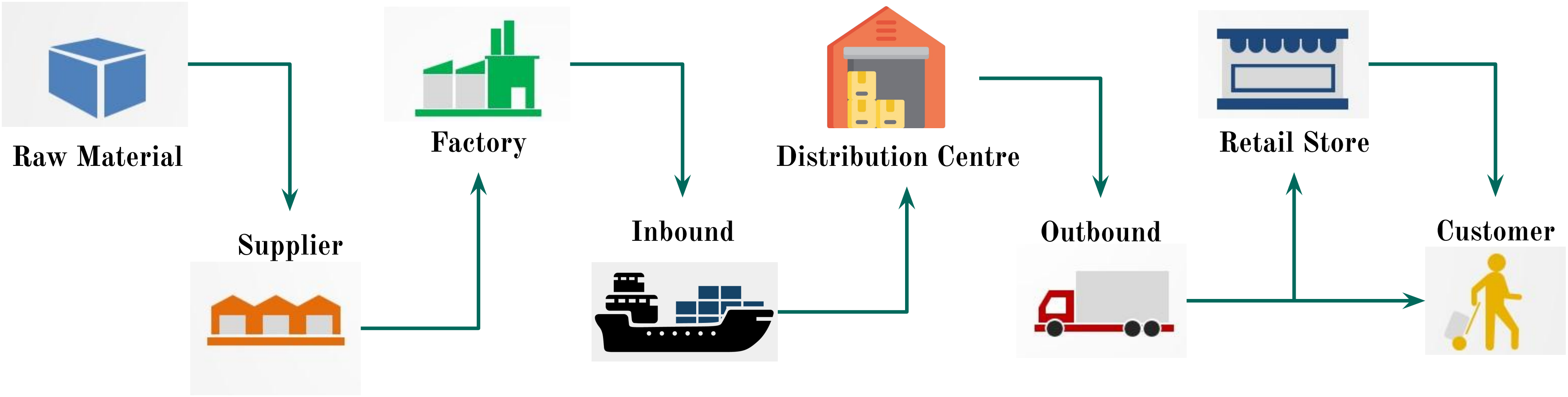
- ➢ Driving Customer Engagement
- ➢ Enhance Shopping Experience
- ➢ Product & Inventory Tracking & Updates
- ➢ Order Tracking & Updates



What types of organizations have you used chatbots to engage with in the past 12 months?

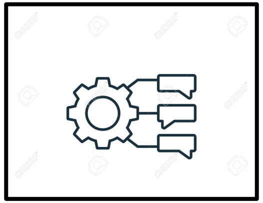
Source : Chatbots magazine

Supply Chain

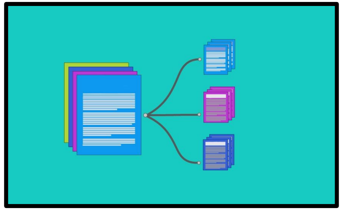


- ➢ **Customer Engagement -** Omnichannel touchpoints.
- ➢ **Customer Order processing -** Order management, Track orders
- ➢ **Warehouse Management -** Inventory management, Vendor orders, logistics
- ➢ **Store replenishment -** Meet customer demand

Building blocks - Supply Chain Bot

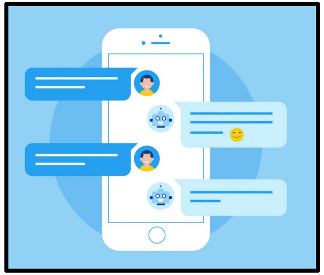


Natural Language Processing

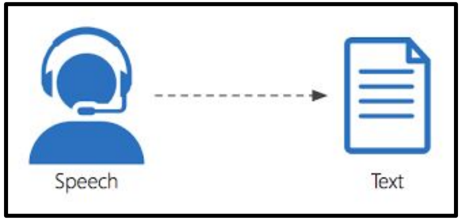


Intent Recognition - Routing across subject areas

Named Entity Extraction



Guided Conversation
Conversational UI



Voice Recognition

Multi-Lingual Support

Tech Stack

- Python 3.x
- Flask
  - Bot as a web service
- SpaCy
  - Named Entity recognition
- FastText
  - Intent Recognition
- Symspell, regex
  - Spell Checking Functionality

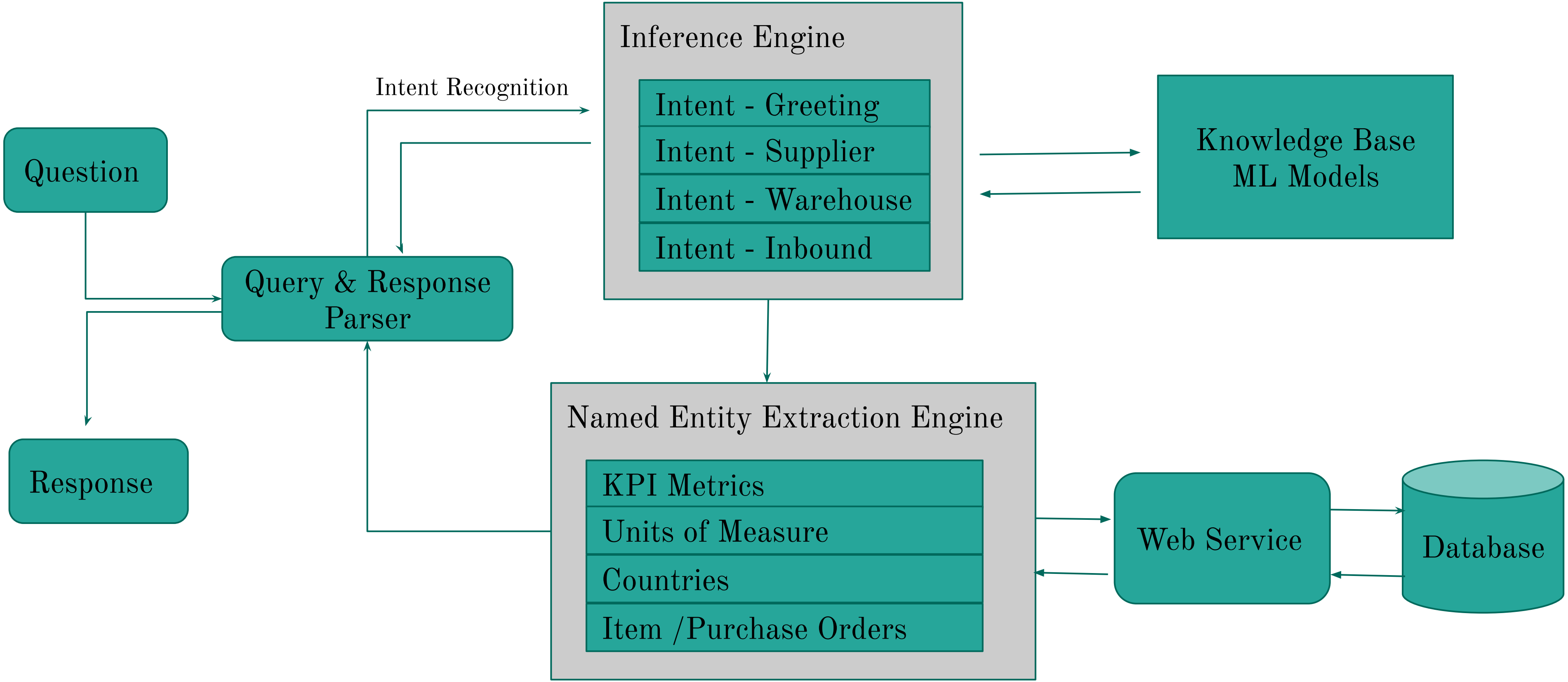
- Smtplib, email
- Cryptography
- Json

Open Source Technologies

vs

Commercial software

Chatbot Architecture



Training Data

- Domain-Specific
- Built using Python Script
- Covers all kinds of user-inputs
    - Supports symbols, acronyms
- Normalized Text

"Recommended shipments in units for XXX supplier?"

"Number of items available to ship from XXX supplier?"

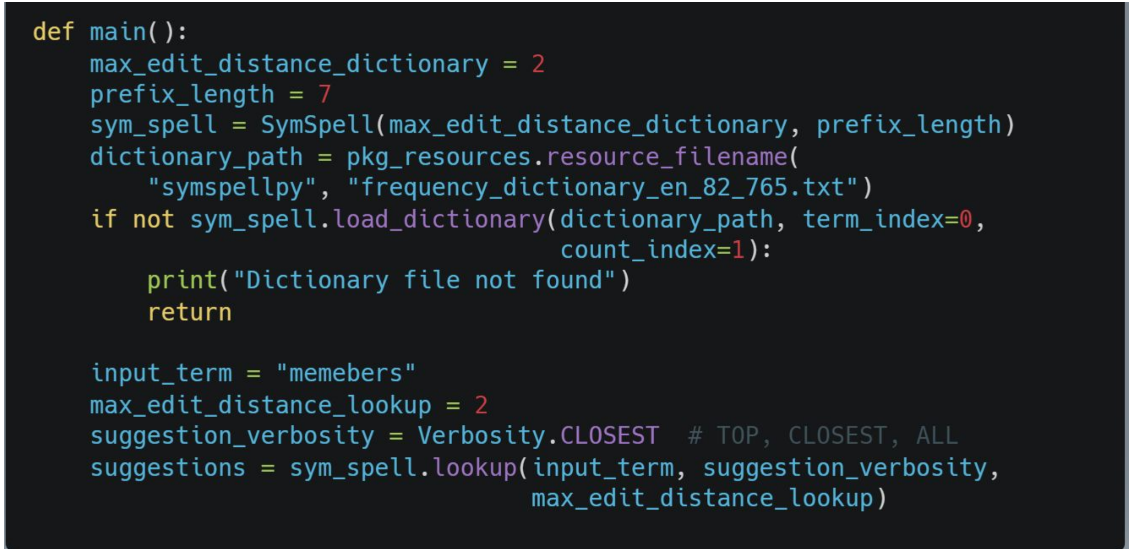
"Please give me the available to ship in dollars and percentage?"

"For item 123456, give me recommended shipments units for XXX DC?"

"How many purchase orders from XXX supplier are scheduled to arrive on time?"

Data Pre-processing

- Expanding acronyms - Pre-defined List of Words
- Removal of Punctuations & Special Characters - **regex()**
- Conversion to lowercase - **str.tolower()**
- Spell Checkers - **Symspell**

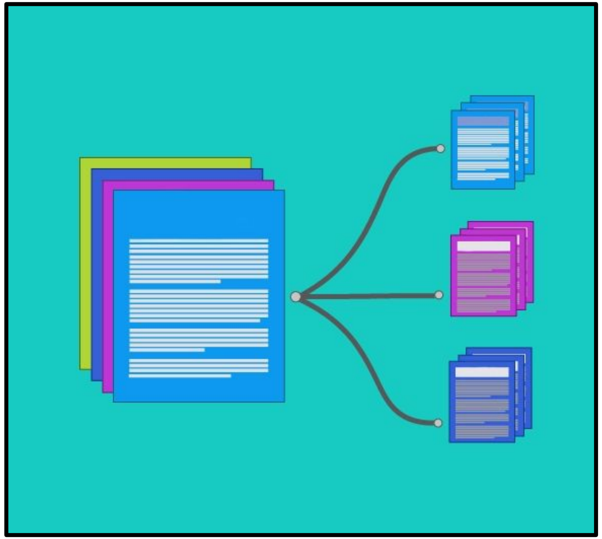


```
def main():
    max_edit_distance_dictionary = 2
    prefix_length = 7
    sym_spell = SymSpell(max_edit_distance_dictionary, prefix_length)
    dictionary_path = pkg_resources.resource_filename(
        "symspellpy", "frequency_dictionary_en_82_765.txt")
    if not sym_spell.load_dictionary(dictionary_path, term_index=0,
                                     count_index=1):
        print("Dictionary file not found")
        return

    input_term = "memebers"
    max_edit_distance_lookup = 2
    suggestion_verbosity = Verbosity.CLOSEST  # TOP, CLOSEST, ALL
    suggestions = sym_spell.lookup(input_term, suggestion_verbosity,
                                   max_edit_distance_lookup)
```

Source: https://symspellpy.readthedocs.io/en/latest/examples/lookup.html#basic-usage

Intent Recognition

- Intent: represents the purpose of a user's input
- How to train the chatbot to understand various types of queries?
- Each query → Mapped to an intent → Each intent → Specific chatbot action



Greetings

Metric Definitions

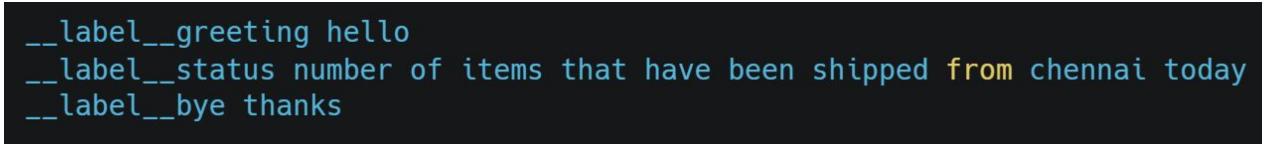
Domain-Specific Questions

Level 1 problem solving

Raise a helpdesk ticket

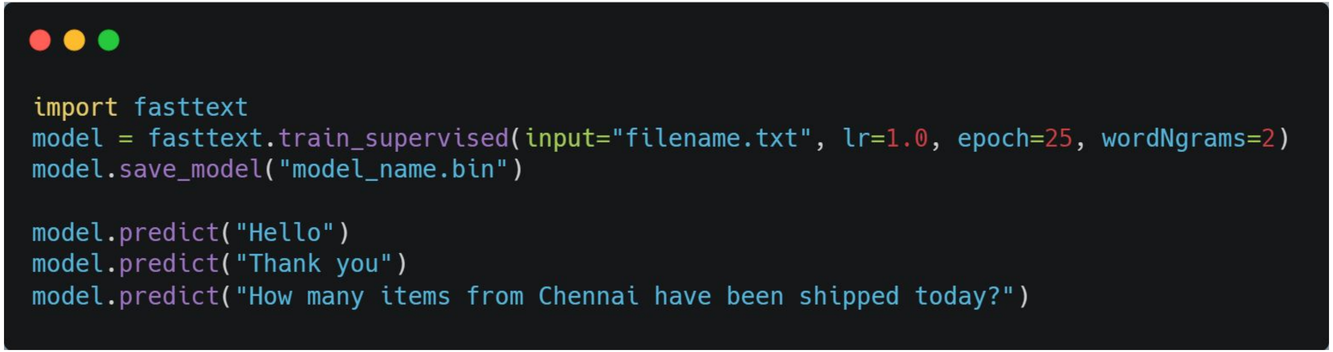
Intent Recognition - Fasttext

- ➢ Library for efficient text classification and representation learning
- ➢ pip install fasttext
- ➢ Training Data Format - __label__XYZ sentence



```
__label__greeting hello
__label__status number of items that have been shipped from chennai today
__label__bye thanks
```

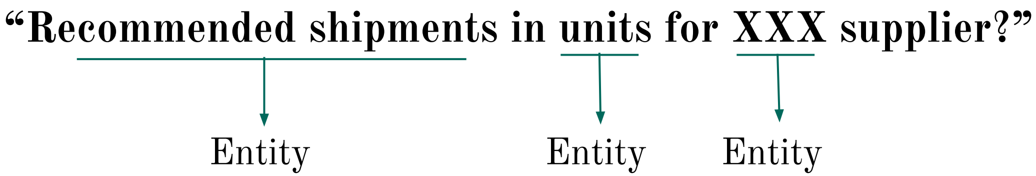
- ➢ Supervised Learning & Prediction



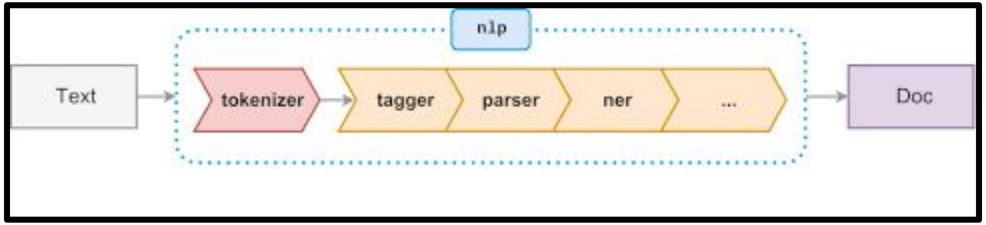
```
import fasttext
model = fasttext.train_supervised(input="filename.txt", lr=1.0, epoch=25, wordNgrams=2)
model.save_model("model_name.bin")

model.predict("Hello")
model.predict("Thank you")
model.predict("How many items from Chennai have been shipped today?")
```

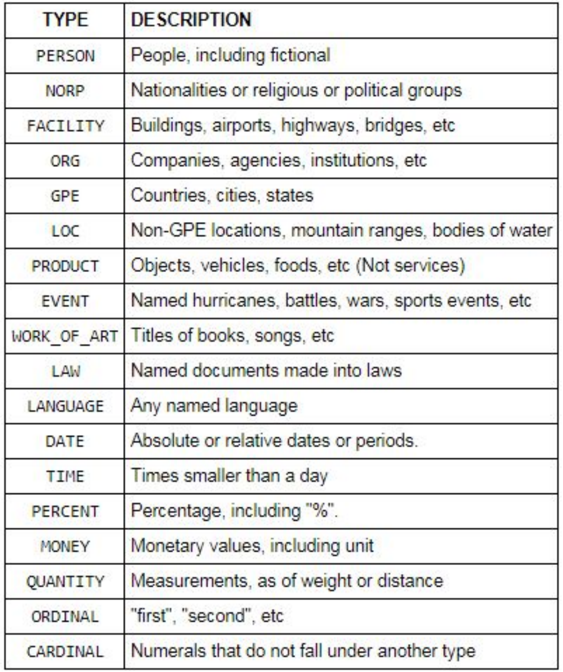
Source: https://pypi.org/project/fasttext/

Named Entity Recognition


“Recommended shipments in units for XXX supplier?”

- ➢ Locate and classify named entities
- ➢ Unstructured text → Predefined categories (Entities)
- ➢ Domain-specific Jargon Words?
- ➢ Pre-trained vs Custom Models

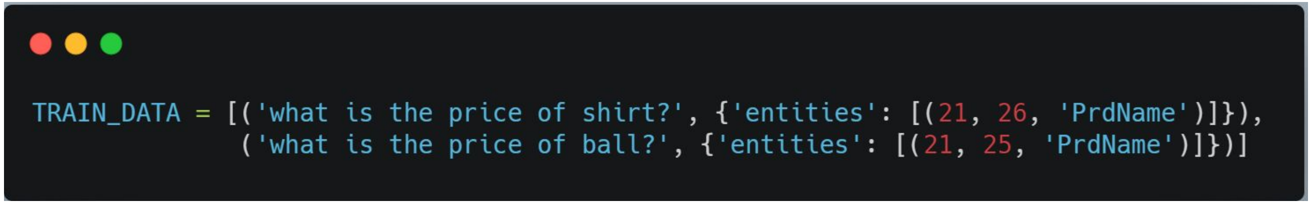




TYPE	DESCRIPTION
PERSON	People, including fictional
NORP	Nationalities or religious or political groups
FACILITY	Buildings, airports, highways, bridges, etc
ORG	Companies, agencies, institutions, etc
GPE	Countries, cities, states
LOC	Non-GPE locations, mountain ranges, bodies of water
PRODUCT	Objects, vehicles, foods, etc (Not services)
EVENT	Named hurricanes, battles, wars, sports events, etc
WORK_OF_ART	Titles of books, songs, etc
LAW	Named documents made into laws
LANGUAGE	Any named language
DATE	Absolute or relative dates or periods.
TIME	Times smaller than a day
PERCENT	Percentage, including "%".
MONEY	Monetary values, including unit
QUANTITY	Measurements, as of weight or distance
ORDINAL	"first", "second", etc
CARDINAL	Numerals that do not fall under another type

Source: https://spacy.io/

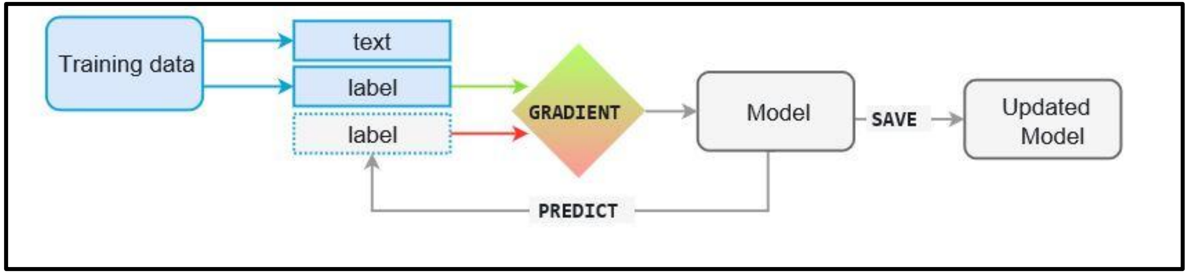
Automated Annotation Pipeline



```
TRAIN_DATA = [('what is the price of shirt?', {'entities': [(21, 26, 'PrdName')]}),
              ('what is the price of ball?', {'entities': [(21, 25, 'PrdName')]})]
```

- Create training data
- **Checkpoint:** Save training data as CSV
- Load CSV, capture entities & it’s values
- Map entities with corresponding positional values
- Append multiple user queries & entities - SpaCy training Data format
- Split dataset into training and test
- **Checkpoint:** Save annotated training dataset as CSV & Validate
- Initiate Model Building

Custom SpaCy Statistical Model

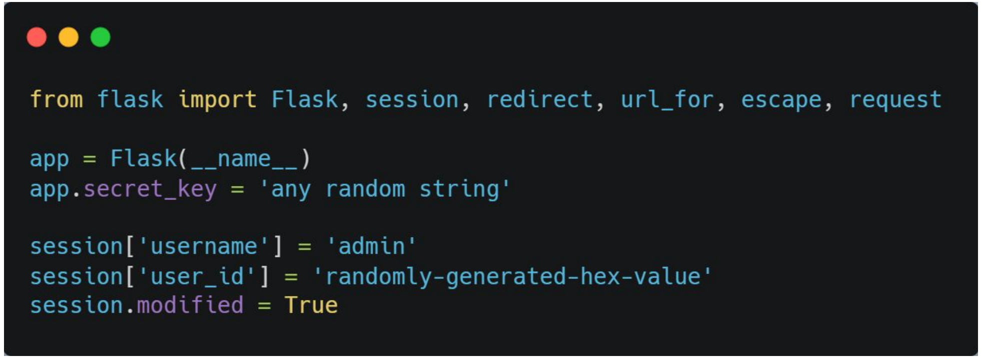


- **Training data:** Examples and their annotations.
- **Text:** The input text the model should predict a label for.
- **Label:** The label the model should predict.
- **Gradient:** Gradient of the loss function calculating the difference between input and expected output.

Source: https://spacy.io/

Context Awareness

- ➢ Chatbot remembers & retains previous entities
- ➢ **Flask Sessions** - consistent across different functionalities
- ➢ Dictionary object - key value pairs: session variables and values
- ➢ Unique session ID for each user session→ session data mapped to each user ID



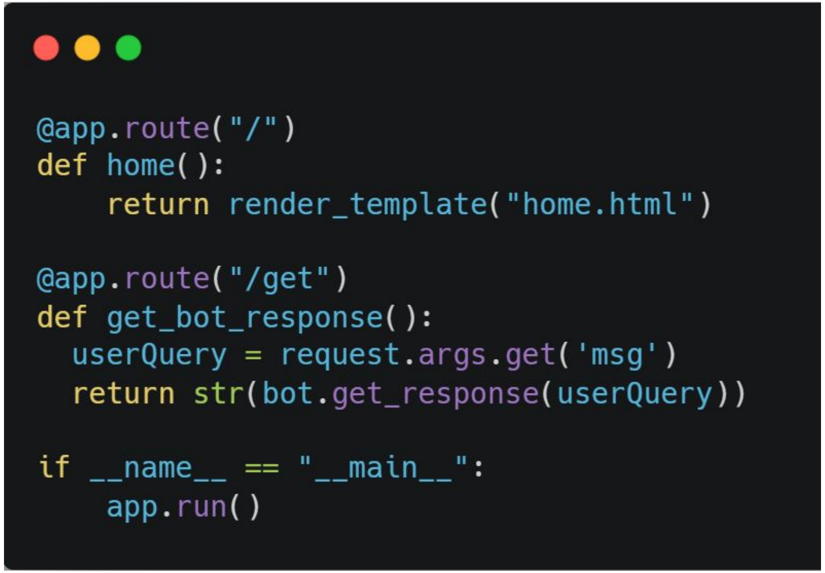
```
from flask import Flask, session, redirect, url_for, escape, request

app = Flask(__name__)
app.secret_key = 'any random string'

session['username'] = 'admin'
session['user_id'] = 'randomly-generated-hex-value'
session.modified = True
```

- ➢ Slot-based logic
- ➢ Check mandatory entities through session keys
- ➢ Last queried context sustained until user logs out the application

CHATBOT DEPLOYMENT

Bot as a web service

- ➢ Flask based web application
- ➢ Can interact with Angular/React UI
- ➢ Web-service Driven Data Fetch
- ➢ JSON Based Responses to UI



```python
@app.route("/")
def home():
    return render_template("home.html")

@app.route("/get")
def get_bot_response():
  userQuery = request.args.get('msg')
  return str(bot.get_response(userQuery))

if __name__ == "__main__":
    app.run()
```

User Feedback

- Real-time feedback captured in database
- Thumbs Up vs Thumbs Down
- Session level, User level, Solution level feedback
- Analytics and visualization of responses
- Understand user requirements
- Adapt to evolving business requirements
- Model Improvement

Model Performance

- Model prediction accuracy
  - Intent Recognition
  - NER
- Model drift
- New Intents + New entities

Model Retraining

- Increase in Training Data
- Hyperparameter Tuning

CONCLUSION

Build your own supply chain chatbot? Tips ..

- Leverage NLP
- Integrate with enterprise supply chain ecosystem applications
- Consume relevant web services
- Leverage ML text classifiers
- Support pointed questions and pointed answers
- Enable capability to re-learn from mistakes/errors
- Exhibit Context Awareness
- Reduce number of clicks for user to get to the information of interest

QUESTIONS

niharikakrishnan

sridhar-rathinasabapathy-46320b10