NAMED ENTITY RECOGNITION (NER)

Samiya Jabbar

AGENDA

Introduction to NER

Types of Named Entities

System Diagram

Challenges in NER

Approaches to NER

Evaluation and Performance Metrics

Applications of NER

Conclusion

Train NER on Custom Dataset

Practical Implementation – Integration with Chatbot

INTRODUCTION

What is Named Entity Recognition?

- NER is a **subtask** of Natural Language Processing (NLP).
- NER enables the **extraction of meaningful information** from unstructured text.
- NER is the process of **identifying** and **classifying** named entities in text.
- NER has importance in various NLP applications, such as information extraction, question answering, and sentiment analysis.

EXAMPLE OF NAMED ENTITIES

NER recognizes and labels these named entities to extract relevant information from text.

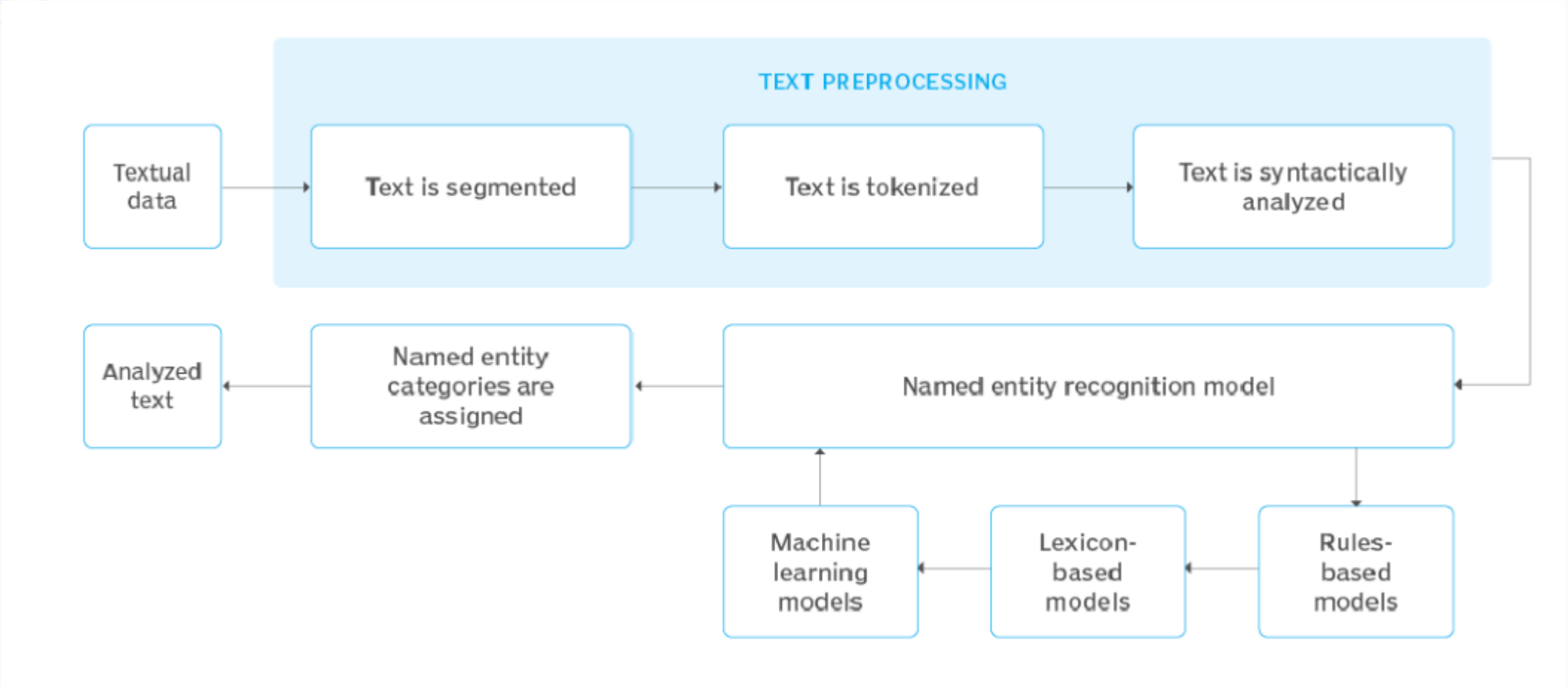
Example sentence: “We are student of NED University, Pakistan. Student of Batch 2021”

TYPES OF NAMED ENTITIES

Common types of named entities:
PERSON: Individual people, including names
ORGANIZATION: Companies, institutions, organizations
LOCATION: Places such as cities, countries, landmarks
DATE: Specific dates or date ranges
TIME: Specific times or time ranges
MONEY: Monetary values
PERCENT: Percentage values
QUANTITY: Measurements or quantities
ORDINAL: Words representing ranks or positions
MISCELLANEOUS: Other named entities not falling into the above categories

NER can classify and extract information about these specific types of entities from the text

SYSTEM DIAGRAM



CHALLENGES IN NAMED ENTITY RECOGNITION

- **Ambiguity:** Words with multiple meanings
- **Contextual variations:** Different forms or variations of named entities
- **New entities:** Handling new or unseen named entities
- **Domain-specific challenges:** Variations across different domains
- **Language complexity:** Challenges in different languages
- **Co-reference resolution:** Resolving pronouns or referring expressions

APPROACHES TO NAMED ENTITY RECOGNITION

Two main approaches:

- Rule-based approaches: Using handcrafted rules and patterns
- Machine learning approaches: Using statistical or deep learning models

RULE-BASED APPROACHES

- Using predefined rules and patterns based on linguistic knowledge
- Limited flexibility but can be effective in specific cases

Examples: Pattern matching, regular expressions, dictionaries

MACHINE LEARNING APPROACHES

- Utilize statistical or deep learning models to learn from data
- Training on annotated datasets to recognize patterns

Examples: Hidden Markov Models (HMMs), Conditional Random Fields (CRFs), Neural Networks (e.g., LSTM), Transformer-based models (e.g., BERT)

EVALUATION AND PERFORMANCE METRICS

- Precision, recall, and F1-score:

These metrics measure the accuracy and completeness of the named entity recognition system.

- Entity-level evaluation vs. token-level evaluation:

Evaluation can be performed at the entity level or at the token level, depending on the specific requirements of the task

APPLICATIONS OF NAMED ENTITY RECOGNITION

Real-world applications of NER:

Information extraction

Question answering systems

Sentiment analysis

Text summarization

Entity linking

Chatbots and virtual assistants

CONCLUSION

- Significance of NER in NLP and its wide range of applications.
- Ongoing advancements in NER research and its importance in improving various NLP tasks.

TRAIN NER ON CUSTOM DATASET

STEPS:

- Data Collection
- Data Annotation
- Spacy - To Train the NER Model
- Test the model

TRAINING DATASET

Once upon a time, in a beautiful garden called Janna, there lived a little girl named Kamila. Kamila was known for her complete kindness and virtue. She had a special bond with nature and would spend her days among the fragrant flowers, spreading happiness wherever she went.

One day, Kamila came across a group of girls playing under a majestic tree. Their names were Zahra, Fatema, and Areej. They were picking flowers and singing songs of cheerfulness. Kamila joined them, and together they created a beautiful melody that echoed through the garden.

As they continued their joyful activities, a young girl named Jasmin appeared. She had recently arrived from a distant land called Hind. Jasmin was amazed by the peaceful atmosphere and the beauty of the garden. She shared stories of her homeland, bringing a sense of unity among them.

Deep in the garden, they discovered a hidden oasis named Firdaus. Its crystal-clear water reflected the moonlight, creating a mesmerizing sight. The girls sat by the water, captivated by its radiance. They began to share their dreams and aspirations, inspiring one another to reach for the heights.

As they spent more time together, a strong bond formed among the girls. They named their group "Mawhib," symbolizing their unique abilities and virtues. The Mawhib girls became ambassadors of peace, spreading kindness, balance, and trust in the garden and beyond.

Word of their friendship and harmonious ways reached the queen of the garden, Malika. Impressed by their unity, Malika invited the Mawhib girls to her palace, where they met Queen Bilqis from a distant land. Bilqis, a wise and powerful queen, praised their commitment to peace and named them the "Houriya," representing their eternal essence of vitality and purity.

The Houriya continued to bring joy and enlightenment to others, inspiring them to find their own balance and radiance. Their story of unity and peaceful coexistence became a legend, reminding people of the beauty that lies in diversity and the importance of cherishing one another.

And so, the Houriya's tale of friendship, peace, and enlightenment continued to be shared, spreading happiness throughout the garden of Janna and beyond, forever reminding everyone of the power of love and harmony.

Once upon a time, in a land called Aazad, there lived a young boy named Amir. Amir was known for his thriving and prosperous nature. He had a free spirit and a deep love for independence.

One day, as Amir was exploring the mountains of Aban, he came across a group of boys engaged in a great worship ceremony. Their names were Hasan, Abbas, and Ayman. They were known for their noble qualities and their dedication to the right path. Intrigued by their righteousness, Amir decided to join them in their rituals.

In their journey, they encountered a wise sage named Labib. Labib was known for his profound wisdom and his ability to guide others on the right path. He shared stories of brave warriors like Haitham and Jalal, who displayed great courage and glory. The boys listened intently, inspired by the tales of these valiant heroes.

Activate Windows
Go to Settings to activate Windows.

LIST OF MUSLIM GIRL'S & BOYS' NAMES USED IN THE TRAINING DATASET[1]

Boy's Names:

- Amir
- Hasan
- Abbas
- Ayman
- Labib
- Haitham
- Jalal
- Munsif
- Adib
- Isam
- Muslih
- Hamza
- Mansur

Total: 13 boy's names

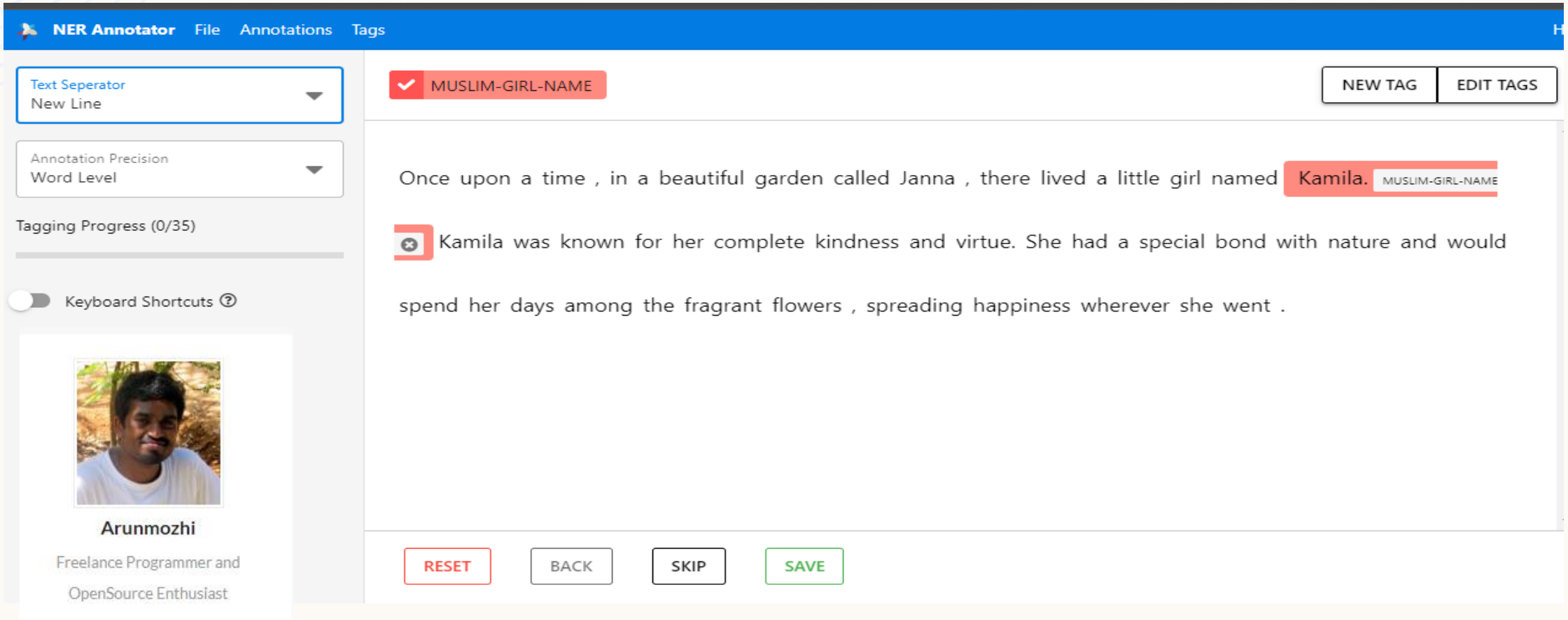
Girl's Names:

- Kamila
- Zahra
- Fatema
- Areej
- Jasmin
- Malika
- Bilqis
- Houriya

Total: 8 girl's names

1 "Muslim Islamic Baby Names - Girls & Boys Name." Milligazette. Available at: https://www.milligazette.com/misl/muslim_islam_islamic_baby_names_girl_boy_name.htm. Accessed 8 July 2023.

TOOL USED FOR ANNOTATION[2]



[2] Arunmozhi. "NER Annotator." Tecoholic. Available at: https://tecoholic.github.io/ner-annotator/. Accessed 8 July 2023.

RESULT OF ANNOTATION

annotations.json - Notepad

File Edit Format View Help

{"classes":["MUSLIM-GIRL-NAME","MUSLIM-BOY-NAME"],"annotations":[["Once upon a time, in a beautiful garden called Janna, there lived a little girl named Kamila. Kamila was known for her complete kindness and virtue. She had a special bond with nature and would spend her days among the fragrant flowers, spreading happiness wherever she went.\r",{"entities":[[86,93,"MUSLIM-GIRL-NAME"]]}],["\r",{"entities":[]}],["One day, Kamila came across a group of girls playing under a majestic tree. Their names were Zahra, Fatema, and Areej. They were picking flowers and singing songs of cheerfulness. Kamila joined them, and together they created a beautiful melody that echoed through the garden.\r",{"entities":[[93,98,"MUSLIM-GIRL-NAME"],[100,106,"MUSLIM-GIRL-NAME"],[112,118,"MUSLIM-GIRL-NAME"]]}],["\r",{"entities":[]}],["As they continued their joyful activities, a young girl named Jasmin appeared. She had recently arrived from a distant land called Hind. Jasmin was amazed by the peaceful atmosphere and the beauty of the garden. She shared stories of her homeland, bringing a sense of unity among them.\r",{"entities":[[62,68,"MUSLIM-GIRL-NAME"]]}],["\r",{"entities":[]}],["Deep in the garden, they discovered a hidden oasis named Firdaus. Its

SPACY

A POPULAR NER FRAMEWORK

- SpaCy is a widely used NLP library
- Having built-in NER capabilities and pre-trained models

In [2]:
```
# Import free, open-source library for NLP in Python
#designed to make it easy to build systems for information extraction or general-purpose NLP
import spacy
from spacy import displacy
```

In [3]:
```
# Load model that is pre-tranied on eglish language, sm = small model that is optimized for speed
nlp = spacy.load("en_core_web_sm")
```

In [21]:
```
#Load the text
text =("We are student of NED University, Pakistan. Student of Batch 2021")
#Process the text
doc = nlp(text)
```

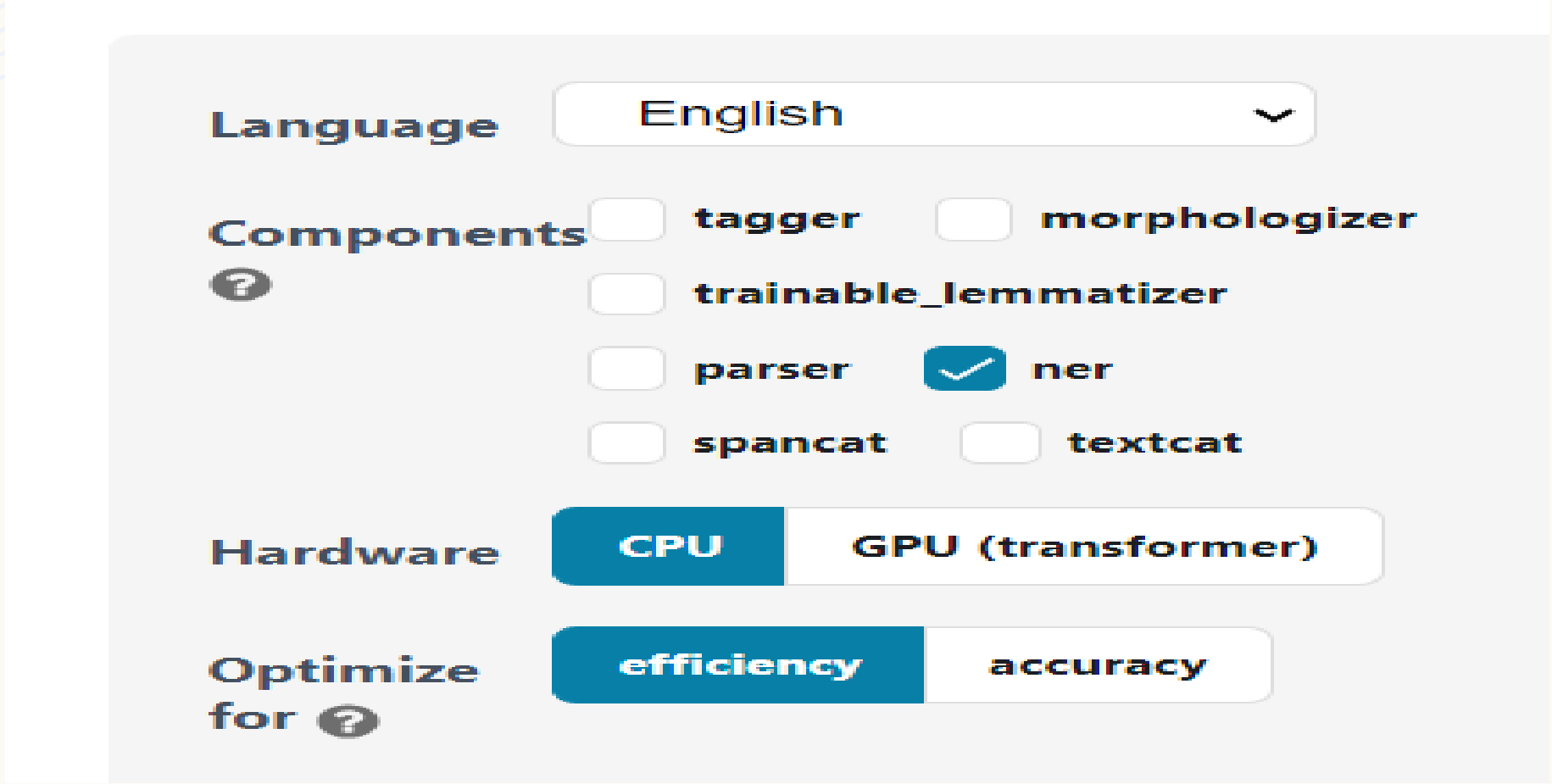
In [22]:
```
# Iterate over the entities in the document and print entities
ents = [(e.text, e.start_char, e.end_char, e.label_) for e in doc.ents]
print(ents)
```

```
[('NED University', 18, 32, 'ORG'), ('Pakistan', 34, 42, 'GPE'), ('2021', 61, 65, 'DATE')]
```

In [23]:
```
# using displaycy function for visualization purpose
displacy.render(doc, style='ent', jupyter=True)
```

We are student of NED University ORG , Pakistan GPE . Student of Batch 2021 DATE

SPACY CONFIG WIDGET[3]



[3] "Training." spaCy. Available at: https://spacy.io/usage/training. Accessed 8 July 2023.

TRAINING ON CUSTOM DATASET

Train_Custom_NER.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

```
[5]  db = DocBin() # create a DocBin object
```

```
[16] import json
     f = open('annotations.json')
     TRAIN_DATA = json.load(f)
```

```
for text, annot in tqdm(TRAIN_DATA['annotations']):
    doc = nlp.make_doc(text)
    ents = []
    for start, end, label in annot["entities"]:
        span = doc.char_span(start, end, label=label, alignment_mode="contract")
        if span is None:
            print("Skipping entity")
        else:
            ents.append(span)
    doc.ents = ents
    db.add(doc)

db.to_disk("./training_data.spacy") # save the docbin object
```

```
100%|██████████| 35/35 [00:00<00:00, 1240.37it/s]
```

```
[19] ! python -m spacy init config config.cfg --lang en --pipeline ner --optimize efficiency
```

TRAINING RESULTS

```
[20] ! python -m spacy train config.cfg --output ./ --paths.train ./training_data.spacy --paths.dev ./training_data.spacy
```

```
2023-07-09 19:43:43.153310: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find Te
ℹ Saving to output directory: .
ℹ Using CPU

=========================== Initializing pipeline ===========================
[2023-07-09 19:43:46,499] [INFO] Set up nlp object from config
[2023-07-09 19:43:46,518] [INFO] Pipeline: ['tok2vec', 'ner']
[2023-07-09 19:43:46,522] [INFO] Created vocabulary
[2023-07-09 19:43:46,524] [INFO] Finished initializing nlp object
[2023-07-09 19:43:46,674] [INFO] Initialized pipeline components: ['tok2vec', 'ner']
✓ Initialized pipeline

============================= Training pipeline =============================
ℹ Pipeline: ['tok2vec', 'ner']
ℹ Initial learn rate: 0.001
```

E	#	LOSS TOK2VEC	LOSS NER	ENTS_F	ENTS_P	ENTS_R	SCORE
0	0	0.00	27.50	0.00	0.00	0.00	0.00
17	200	128.80	847.35	100.00	100.00	100.00	1.00
39	400	0.00	0.01	100.00	100.00	100.00	1.00
66	600	0.00	0.00	100.00	100.00	100.00	1.00
101	800	0.00	0.00	100.00	100.00	100.00	1.00
146	1000	0.00	0.00	100.00	100.00	100.00	1.00
198	1200	0.00	0.00	100.00	100.00	100.00	1.00
264	1400	0.00	0.00	100.00	100.00	100.00	1.00
343	1600	0.00	0.00	100.00	100.00	100.00	1.00
443	1800	0.00	0.00	100.00	100.00	100.00	1.00

```
✓ Saved pipeline to output directory
model-last
```

TESTING RESULTS

```
[21] nlp_ner = spacy.load("/content/model-best")
```

```
[22] # input test data
     doc = nlp_ner('''In the bustling city of Azaan, a young girl named Safiya wandered through the vibrant marketplaces.
     As she strolled along, she crossed paths with a curious boy named Zayn, known for his adventurous spirit.
     Together, Safiya and Zayn embarked on a thrilling quest to uncover the secrets of the ancient temple of Aisha.''')
```

```
[23] spacy.displacy.render(doc, style="ent", jupyter=True) # display in Jupyter
```

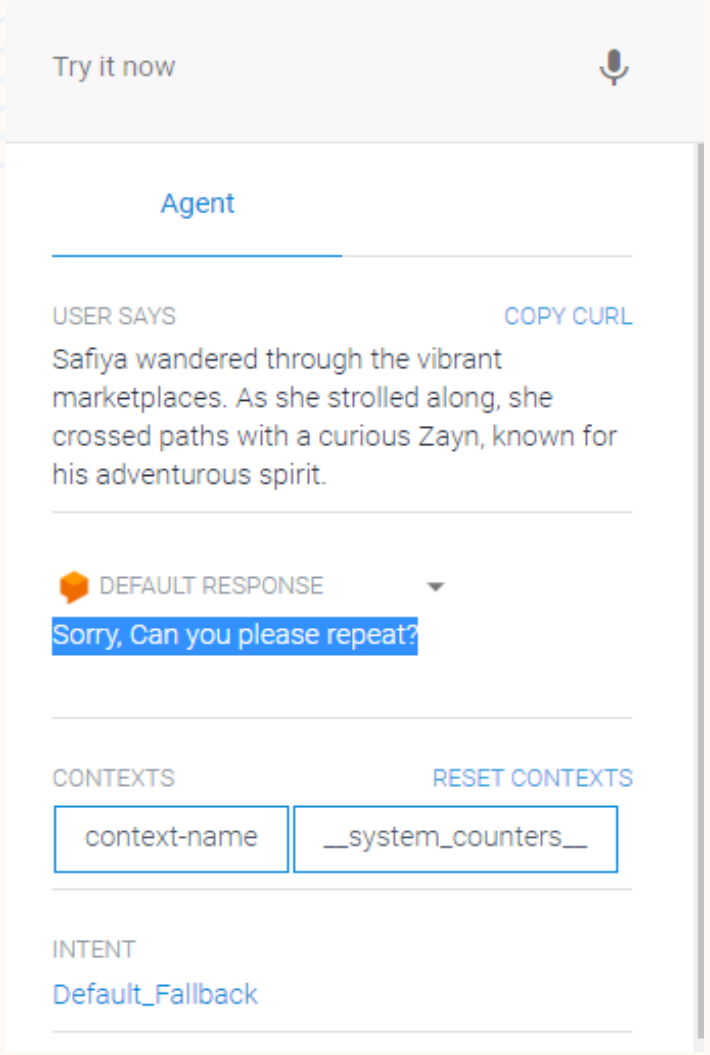
In the bustling city of Azaan MUSLIM-BOY-NAME , a young girl named Safiya MUSLIM-GIRL-NAME wandered through the vibrant marketplaces.

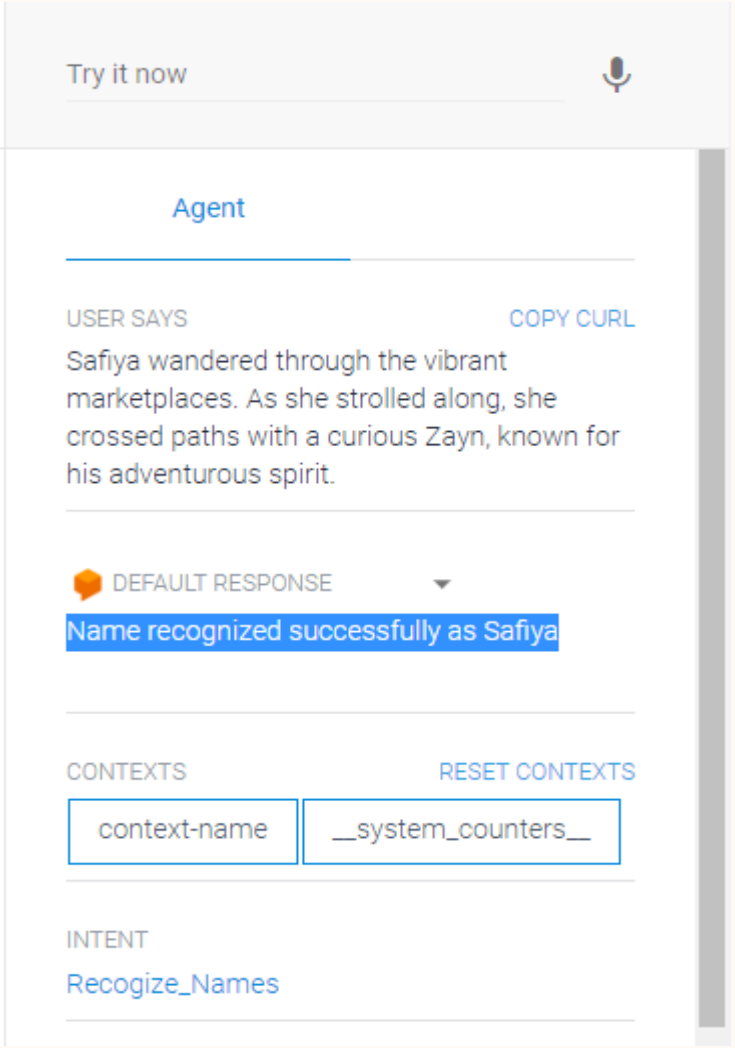
As she strolled along, she crossed paths with a curious boy named Zayn MUSLIM-GIRL-NAME , known for his adventurous spirit.

Together, Safiya MUSLIM-GIRL-NAME and Zayn embarked on a thrilling quest to uncover the secrets of the ancient temple of Aisha. MUSLIM-GIRL-NAME

PRACTICAL IMPLEMENTATION[4]

USING DIALOGFLOW





[4] "Webhook service." Cloud Google. Available at: https://cloud.google.com/dialogflow/es/docs/fulfillment-webhook. Accessed 9 July 2023.

PRACTICAL IMPLEMENTATION

CODE - USING FLASK

NER-Dialogflow.py

```python
from flask import Flask, request, jsonify
import spacy

app = Flask(__name__)

@app.route('/webhook', methods=['POST'])
def webhook():
    data = request.get_json()
    query = data['queryResult']['queryText']
    boy_name = None
    girl_name = None
    print(query)

    my_spacy_model = spacy.load("model-best")

    # Extract Muslim boy and girl names using the SpaCy model
    doc = my_spacy_model(query)
    print('these are doc ets', doc.ents)
    for ent in doc.ents:
        print("Entered in loop")
        if ent.label_ == 'MUSLIM-BOY-NAME':
            boy_name = ent.text
            print('the text contains boy name:', boy_name)
            result = boy_name
        elif ent.label_ == 'MUSLIM-GIRL-NAME':
            girl_name = ent.text
            print('the text contains girl name:', girl_name)
            result = girl_name
```

NER-Dialogflow.py

```python
            result = boy_name
        elif ent.label_ == 'MUSLIM-GIRL-NAME':
            girl_name = ent.text
            print('the text contains girl name:', girl_name)
            result = girl_name

    # Prepare the response
    response = {
        'fulfillmentText': f'Name recognized successfully as {result}',
        'outputContexts': [
            {
                'name': 'projects/<PROJECT_ID>/agent/sessions/<SESSION_ID>/contexts/context-name',
                'lifespanCount': 5,
                'parameters': {
                    'boy_name': boy_name,
                    'girl_name': girl_name
                }
            }
        ]
    }

    return jsonify(response)

if __name__ == '__main__':
    app.run(debug=True)
```

PRACTICAL IMPLEMENTATION

RESULTS (VS-CODE TERMINAL)

```
127.0.0.1 - - [10/Jul/2023 02:43:10] "POST /webhook HTTP/1.1" 200 -
Safiya wandered through the vibrant marketplaces.  As she strolled along, she crossed paths with a curious Zayn, known for his adventurous spirit.
these are doc ets (Safiya,)
Entered in loop
the text contains girl name: Safiya
127.0.0.1 - - [10/Jul/2023 02:43:53] "POST /webhook HTTP/1.1" 200 -
```

REFERENCES

[1] "Muslim Islamic Baby Names - Girls & Boys Name." Milligazette. Available at: https://www.milligazette.com/misl/muslim_islam_islamic_baby_names_girl_boy_name.htm. Accessed 8 July 2023.

[2] Arunmozhi. "NER Annotator." Tecoholic. Available at: https://tecoholic.github.io/ner-annotator/. Accessed 8 July 2023.

[3] "Training." spaCy. Available at: https://spacy.io/usage/training. Accessed 8 July 2023.

[4] "Webhook service." Cloud Google. Available at: https://cloud.google.com/dialogflow/es/docs/fulfillment-webhook. Accessed 9 July 2023.

THANKYOU!

- **Chatgpt** - for providing guidance and making our work easy
- **Arunmozhi** – for providing a free open-source annotation tool
- **Sir Lubaid** – For providing us an opportunity to work on it