Report: Entity Extraction and Sentiment Analysis Using NLP Techniques

Introduction

In this report, I summarize the approach taken to develop three distinct Natural Language Processing (NLP) scripts: one for named entity extraction using spaCy, one for sentiment analysis using DistilBERT, and one for web scraping and text extraction using BeautifulSoup. Each script leverages different NLP techniques and models to process and analyze textual data. This report outlines the methodology, challenges encountered, and reflections on the accuracy of the implemented solutions.

Approach:

1. Entity Extraction (using spaCy)

Objective: To identify and categorize named entities within a given text. The primary focus is on extracting entities of types 'PERSON' and 'ORG'.

Methodology: Utilized the pre-trained `en_core_web_sm` model from spaCy for Named Entity Recognition (NER). The model was used to process the input text, and entities were categorized into predefined types.

Implementation: The script defines a function `extract_entities` that processes the text and extracts entities based on their labels. The entities are stored in a dictionary with sets for 'PERSON' and 'ORG'.

2. Sentiment Analysis (using DistilBERT)

Objective: To determine the sentiment of the provided text, classifying it as either 'positive' or 'negative'.

Methodology: Leveraged the pre-trained `distilbert-base-uncased-finetuned-sst-2-english` model, which is fine-tuned for sentiment analysis. This model was used to analyze the sentiment of the tokenized input text.

Implementation: The `analyze_sentiment` function tokenizes the input text and runs it through the model to get sentiment predictions. The sentiment is then mapped to a label based on the model's output.

3. Web Scraping and Text Extraction (using BeautifulSoup)

Objective: To fetch HTML content from a URL and extract the text from paragraph elements.

Methodology: The script performs an HTTP GET request to retrieve the HTML content, which is then parsed using BeautifulSoup to extract text from all `<p>` tags.

Implementation: Two main functions were implemented—`fetch_html` to retrieve the HTML content and `parse_html` to extract and concatenate the text from paragraph tags.

Challenges Faced

1. Entity Extraction:

Accuracy: The spaCy model `en_core_web_sm` is a small model, which may not capture all named entities with high accuracy. There could be limitations in recognizing less common entities or those in contextually complex sentences.

Improvement: For higher accuracy, using larger models such as `en_core_web_md` or `en_core_web_lg` could be considered.

2. Sentiment Analysis:

Context Sensitivity: The DistilBERT model, while effective, may not fully understand nuanced or complex sentiment expressions, especially when sarcasm or mixed emotions are present.

Performance: The model's performance is highly dependent on the quality and relevance of the input text. Fine-tuning the model further on domain-specific data could enhance results.

3. Web Scraping:

HTML Structure Variability: Web pages vary in their HTML structure, which can affect the effectiveness of text extraction. Some pages may have non-standard tags or dynamic content that complicates parsing.

Error Handling: Ensuring robust error handling for network issues and HTML parsing errors was crucial to maintaining the script's reliability.

Reflections on Accuracy

1. Entity Extraction: The spaCy-based approach provides a good starting point for entity recognition but may miss certain entities due to the model's size and the inherent limitations of pre-trained models. The categorization of entities into 'PERSON' and 'ORG' was effective, but broader entity categories could enhance the utility of the extraction.

Accuracy Evaluation ¶

TOKEN_ACC	Tokenization	1.00
TOKEN_P		1.00
TOKEN_R		1.00
TOKEN_F		1.00
TAG_ACC	Part-of-speech tags (fine grained tags, Token.tag)	0.97
SENTS_P	Sentence segmentation (precision)	0.92
SENTS_R	Sentence segmentation (recall)	0.89
SENTS_F	Sentence segmentation (F-score)	0.91
DEP_UAS	Unlabeled dependencies	0.92
DEP_LAS	Labeled dependencies	0.90
ENTS_P	Named entities (precision)	0.85
ENTS_R	Named entities (recall)	0.85
ENTS_F	Named entities (F-score)	0.85

Reference: [English · spaCy Models Documentation]

2. Sentiment Analysis: The DistilBERT model offers a reliable sentiment classification for straightforward texts. However, its performance might degrade with complex or ambiguous sentiment expressions. The binary sentiment classification (positive/negative) is useful but limited in capturing the full spectrum of sentiments. As per the official HuggingFace site an accuracy of 91.3 is seen on their dev data set.

Reference: [distilbert/distilbert-base-uncased-finetuned-sst-2-english · Hugging Face]

3. Web Scraping: The accuracy of text extraction is heavily dependent on the HTML structure of the target web pages. While the BeautifulSoup library is powerful, its effectiveness can be compromised by non-standard HTML practices or dynamically loaded content.

Conclusion

The implemented scripts are sufficient for a simple web page parsing, performing NER and then doing sentiment analysis, but depending on the use case and nature of the work we need to do certain modifications to them. For better accuracy with sentiment analysis, we should use untrained BERT model and train it on our own sentiment analysis dataset or we can go with a neural network approach to build our own model for sentiment analysis. As for NER, to improve accuracy we can go with “en_core_web_md” or “en_core_web_lg” models that have been trained on significantly larger training dataset.