Hyperiondev

TASK

Capstone Project - NLP Applications

Visit our website

Introduction

WELCOME TO THE NLP APPLICATIONS CAPSTONE PROJECT

Great job getting to this Capstone task! In this Capstone Project, you will be categorising and researching NLP applications. Let's get started!

NATURAL LANGUAGE PROCESSING APPLICATIONS

NLP applications can be classified into a number of categories based on what they do. Let's have a brief look at these categories.

Language translation

The volume of information being put up online is growing at an incredible rate, but due to language barriers, not everything is accessible to everyone. Language translation helps us conquer these language barriers by facilitating the translation of technical manuals, support content, or websites quickly and inexpensively. The challenge with language translation technologies is not in translating words, but in understanding the meaning of sentences to provide a true translation. If you've ever watched a movie in one language with subtitles in another language where you have been able to understand both languages, you will realise how prone to error the process of translation, even using human experts can be. Sometimes the subtitles are outright wrong, and sometimes, although the words are translated accurately, the nuance of the meaning is lost or miscommunicated. Really effective NLP language translation is a challenging area in which exciting progress is being made.

Text classification

One of the applications of NLP that we experience on a daily basis is the text classification in our email folders: by using predefined categories, we can organise our spam folders and inbox so that we can access relevant emails or messages more efficiently.

Automatic summarisation

When working with huge amounts of information (like articles, books, and websites), it can be extremely useful to be able to shorten these pieces into condensed forms that only show the pieces of information that are most useful to you. This is what automatic summarisation is about. According to Expert System

(2020): "Automatic summari[s]ation is relevant not only for summari[s]ing the meaning of documents and information, but also for understanding the emotional meanings inside the information, such as in collecting data from social media."

Sentiment analysis

Similarly to how we can infer someone's meaning from their tone, sentiment analysis allows us to detect the emotion behind a piece of text using NLP. This is particularly useful for large companies who want to know what the general sentiment people hold towards their company is. By analysing articles and write-ups about their company using sentiment analysis, they can gain a fairly accurate idea of how people feel about them based on the language used when they are discussed.

Question answering

This is an application of NLP that has come a long way in a short space of time. Simply put, these systems allow a computer to answer a question posed by a human. Siri and Okay Google are well-known voice-based question answering systems, but text-based systems can now be seen on almost every banking or online shopping site in the form of a chatbox. ChatGPT is a relatively advanced implementation of a question-answering model.

Capstone Project

In this task, you will develop a Python program that performs sentiment analysis on a dataset of product reviews.

Follow these steps:

- **Download a dataset of product reviews**: **Consumer Reviews of Amazon Products**. You can save it as a CSV file, naming it: amazon_product_reviews.csv.

- **Create a Python script**, naming it: sentiment_analysis.py. Develop a Python script for sentiment analysis. Within the script, you will perform the following tasks using the spaCy library:

1. **Implement a sentiment analysis model using spaCy**: Load the `en_core_web_sm` spaCy model to enable natural language processing tasks. This model will help you analyse and classify the sentiment of the product reviews.

2. **Preprocess the text data**: Remove stopwords, and perform any necessary text cleaning to prepare the reviews for analysis.

    2.1. To select the 'review.text' column from the dataset and retrieve its data, you can simply use the square brackets notation. Here is the basic syntax:

    ```
    reviews_data = dataframe['review.text']
    ```

    This column, 'review.text,' represents the feature variable containing the product reviews we will use for sentiment analysis.

    2.2. To remove all missing values from this column, you can simply use the `dropna()` function from Pandas using the following code:

    ```
    clean_data = dataframe.dropna(subset=['reviews.text'])
    ```

3. **Create a function for sentiment analysis**: Define a function that takes a product review as input and predicts its sentiment.

4. **Test your model on sample product reviews**: Test the sentiment analysis function on a few sample product reviews to verify its accuracy in predicting sentiment.

5. **Write a brief report or summary in a PDF file**:
sentiment_analysis_report.pdf that must include:

    5.1. A description of the dataset used.
    5.2. Details of the preprocessing steps.
    5.3. Evaluation of results.
    5.4. Insights into the model's strengths and limitations.

Additional Instructions:

- Some helpful guidelines on cleaning text:
    - To remove stopwords, you can utilise the `.is_stop` attribute in spaCy. This attribute helps identify whether a word in a text qualifies as a stop word or not. Stopwords are common words that do not add

much meaning to a sentence, such as "the", "is", and "of". Subsequently, you can then employ the filtered list of tokens or words(words with no stop words) for conducting sentiment analysis.
    - You can also make use of the `lower()`, `strip() and str()` methods to perform some basic text cleaning.
- You can use the spaCy model and the `.sentiment` attribute to analyse the review and determine whether it expresses a positive, negative, or neutral sentiment. To use the `.polarity` attribute, you will need to install the **TextBlob library**. You can do this with the following commands:
        - `# Install spacytextblob`
        - `pip install spacytextblob`
    - Textblob requires additional data before getting started, download the data using the following code:
        - `python -m textblob.download_corpora`
    - Once you have installed TextBlob, you can use the .sentiment and .polarity attribute to analyse the review and determine whether it expresses a positive, negative, or neutral sentiment. You can also incorporate this code to get yourself started:
        - `# Using the polarity attribute`
        - `polarity = doc._.blob.polarity`
        - `# Using the sentiment attribute`
        - `sentiment = doc._.blob.sentiment`

FYI: The underscore in the code just above is a **Python convention** for naming private attributes. Private attributes are not meant to be accessed directly by the user, but can be accessed through public methods.

- You can use the `.polarity` attribute to measure the strength of the sentiment in a product review. A polarity score of 1 indicates a very positive sentiment, while a polarity score of -1 indicates a very negative sentiment. A polarity score of 0 indicates a neutral sentiment.
- You can also use the `similarity()` function to compare the similarity of two product reviews. A similarity score of 1 indicates that the two reviews are more similar, while a similarity score of 0 indicates that the two reviews are not similar.
    - Choose two product reviews from the 'review.text' column and compare their similarity. To select a specific review from this column, simply use indexing, as shown in the code below:

```
my_review_of_choice = data['reviews.text'][0]
```

    - The above code retrieves a review from the 'review.text' column at index 0. You can select two reviews of your choice using indexing.

However, please be cautious not to use an index that is out of bounds, meaning it exceeds the number of data points or rows in our dataset.

- Include informative comments that clarify the rationale behind each line of code.

Rate us

Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved, or think we've done a good job?

Click here to share your thoughts anonymously.

REFERENCES

Expert.ai Team. (2020). Natural language processing applications. Expert.ai. **https://expertsystem.com/natural-language-processing-applications/**