End to End NLP Pipeline - NLP LECTURE 2

End to End NLP Pipeline + Assignment

Author: Harshavardhan

INSTRUCTOR: Nitish Sir, CampusX

End to end NLP Pipeline

What is the NLP Pipeline ?

NLP Pipeline is a set of steps followed to build an end to end NLP software. It consists of the following steps:

1. Data Acquisition
2. Text Preprocessing
    a. Text cleanup
    b. Basic Preprocessing (Tokenization etc)
    c. Advanced Preprocessing (POS tagging, chunking, etc)
3. Feature Engineering
    a. Bag of words
    b. Tf-Idf
    c. Word2vec
4. Modeling
    a. Model Building
    b. Evaluation
5. Deployment
    a. Deployment on cloud
    b. Monitoring
    c. Model Update

Note:

- This pipeline is not universal
- Deep learning Pipelines are slightly different
- Pipeline is non-linear

Step 1 : Data Acquisition

- **Case 1: Data is Available in house**
    - **On the table**
        - Data is available so proceed to next step
    - **In the database**
        - Talk to data engineers
    - **Less data is available**
        - Data Augmentation
        - Bigram flip
        - Back translate
        - Synonyms
        - Adding noise

- **Case 2: Data is not available**
    - Conduct survey, gather data and label it, create a heuristic approach

- **Case 3: Others have the data**
    - Public dataset
    - Web Scraping : Scrap your competitor’s website using libraries like BS4
    - API : Search for API’s that meet your requirements
    - PDF : Use pdf reading libraries
    - Image : use OCR libraries
    - Audio : speech to text

Step 2 : Text Preparation

- **Cleaning**
    - HTML tag cleaning
    - Emojis (Unicode normalization)
    - Spelling checking (Text Blob)
- **Basic Preprocessing**
    - Tokenization (Sentence tokenization or word tokenization)
    - Stop words removal
    - Stemming
    - Lemmatization
    - Removing punctuation
    - Lowercasing
    - Language detection

- **Advanced Preprocessing**
    - POS tagging
    - Parsing
    - Coreference resolution

Step 3 : Feature Engineering

The process of converting text into numbers is called Feature engineering. Techniques such as Text vectorization, Bag of words,One hot encoding,Word2vec etc

Feature engineering depends upon the problem statement.

- **ML Based Feature engineering**
    - Create features based on domain knowledge.
    - **Advantages**
        - Interpretability
    - **Disadvantages**
        - Domain knowledge required
- **DL Based Feature engineering**
    - Preprocess and directly pass it to the next step skipping the feature engineering step.
    - **Advantages**
        - No need to engineer features manually
        - Domain knowledge not required

- **Disadvantages**
    - Results are not interpretable

Step 4 : Modeling

- **Modeling**
    - Heuristic approaches
        - When you have less data, you can utilize a heuristic approach. However, as you acquire more data, you can integrate both heuristic and machine learning (ML) approaches. In this combined approach, you can leverage the output of heuristic models as features for your ML model.
    - ML
    - DL (Transfer Learning can be used)
    - Cloud API (If cloud solution exists)
- **Evaluation**
    - Intrinsic evaluation
        - Accuracy
        - Confusion matrix
    - Extrinsic evaluation
        - Evaluate your model in a business setting.

Step 5 : Deployment

- **Deploy on cloud**
  - API (microservice)
  - Chatbot
- **Monitor**
  - Dashboard - KPI
- **Update**
  - Update

Assignment

1. Data Acquisition
    a. From where would you acquire the data?
    Scraping data from Quora website.

2. Text Preparation
    a. What kind of cleaning steps would you perform?
    Removing HTML tags, spelling checking

    b. What text preprocessing step would you apply?
    Lowercasing, removing stop words, Lemmatization

    c. Is advanced text preprocessing required?
    Named Entity Recognition, POS tagging might be useful

3. Feature Engineering
    a. What kind of features would you create?
    Anything that captures the similarity between questions like ,Tf-idf, Word2Vec etc

4. Modeling
    a. What algorithm would you use to solve the problem at hand?
    Any similarity based algorithms can be used to find the most similar questions, like KNN or SVM can be used.

    b. What intrinsic evaluation metrics would you use?
    ROC-AUC curve,F1-score, accuracy, precision, recall

    c. What extrinsic evaluation metrics would you use?
    User engagement

5. Deployment

   a. How would you deploy your solution into the entire product?
      Integrating with Quora's main infrastructure.

   b. How and what things will you monitor?
      Keeping track of accuracy and user feedback etc.

   c. What would be your model update strategy?
      Model needs to be retrained periodically to maintain the accuracy.