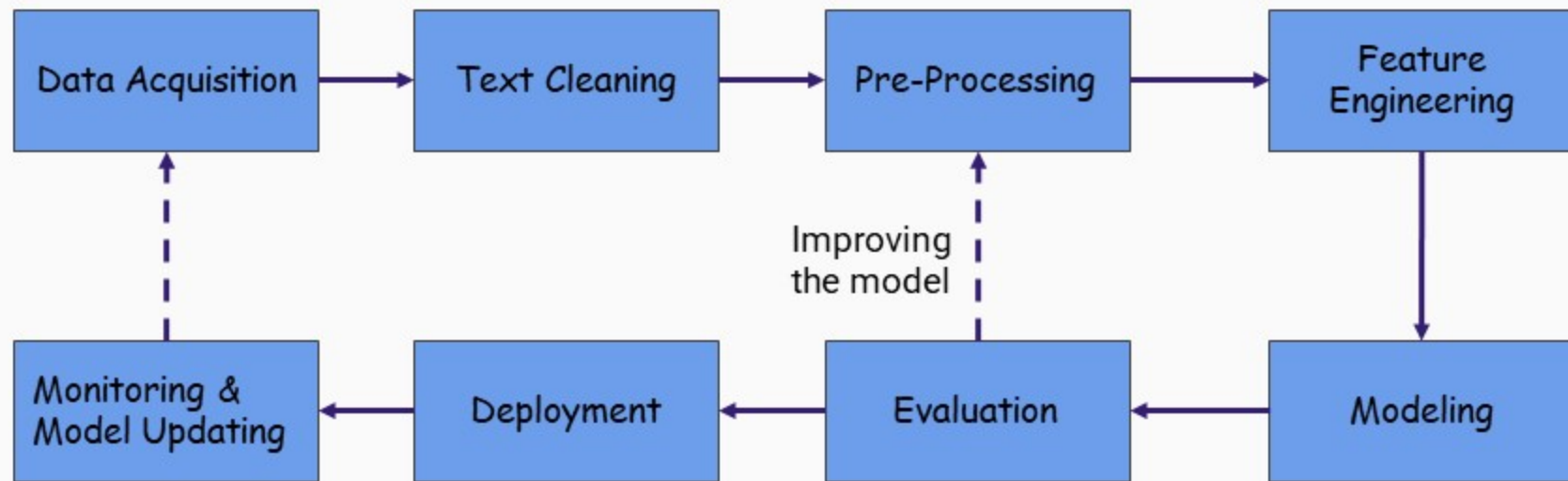


Chap : NLP Pipeline

By Waad ALMASRI

Generic NLP Pipeline



Materials

- Data Acquisition
- Text Extraction & Cleanup
- Pre-Processing
- Feature Engineering
- Modeling
- Evaluation
- Post-Modeling Phases

Data Acquisition



Data Acquisition

When do we need to do Data Acquisition?

- Class: ...

Data Acquisition

When do we need to do Data Acquisition?

- When we have little or no data

Do clients really come with Machine Learning problematic without having the data corresponding to it?

- Unfortunately, they do. And, that's why data is so expensive and why data scientist are paid a fortune 😊

Data Acquisition:

How can we collect data and from where?

- Public datasets: use google search engine to find relevant datasets
- Scrape data: web scraping internet websites (very much used in the industry)
- Product Intervention: Method applied in the industry (Google, Meta, Microsoft, Netflix, etc.). It is when the AI teams works with the product team to collect richer data to serve the model they are deploying.
- Data Augmentation:
 - Synonym replacement
 - Back Translation
 - TF-IDF-based word replacement
 - Bigram flipping
 - Replacing entities
 - Adding noise to data
 - Advanced techniques

Data Acquisition: Data Augmentation

Synonym replacement:

Choose “k” random words that are not stop words and replace them by their synonyms.

Refer to Synsets in Wordnet*

**Miller, George A. "WordNet: a lexical database for English." Communications of the ACM 38.11 (1995): 39-41.*

Data Acquisition: Data Augmentation

Back Translation:

Take every sentence S1, translate it into some other language S2(for example German, Hungarian, etc.), then translate it back to english S3.

S3 and S1 are very similar in the meaning but are slightly different in the words.

Example: English -> Hungarian -> English

I do not think that
attending classes
should be mandatory
in universities.

S1

nem hiszem, hogy
kötelező lenne az
órákon való részvétel
az egyetemeken.

S2

I don't think it
would be mandatory
to attend classes at
universities.

S3

Data Acquisition: Data Augmentation

- **TF-IDF-based word replacement:**

**Xie, Qizhe, et al. "Unsupervised data augmentation for consistency training." arXiv preprint arXiv:1904.12848 (2019)*

- **Bigram flipping:**

Divide the sentence into bigrams, then take one bigram at random and flip it.

Example: The engine of my car is making noises → The engine of my car making is noises.

- **Replacing Entities:**

Replace entities (personname, location, organisation, etc.) with others in the same category.

Example: I live in Paris → I live in California.

- **Adding Noise to data:**

Randomly choose a word in a sentence and replace it with another closer in spelling (add a spelling mistake) Or use the “fat finger” problem on mobile keyboards.

Data Acquisition: Data Augmentation


Advanced Techniques:

- Snorkel
- Easy Data Augmentation EDA & NLPAug
- Active Learning: used in scenarios where there is abundant unlabeled data and manual labeling is expensive.

Text Extraction & Cleanup

This step depends on the format the textual data available are (i.e. normal textual sentences, PDFs, HTML pages, etc.).

It is the most time-consuming part of an NLP Project.

- HTML Parsing & Cleanup: Python Libraries like beautiful soup & Scrapy
- Unicode Normalization: to parse non-textual symbols & special characters (text encoding) Example: text = I love  → text.encode('utf-8') = I love Pizza
\\xf0\\x9f\\x8d\\x95
- Spelling Correction: with fast typing & "Fat Finger" typing, wrong spelling is very common. Thus, one can use refer to a dictionary or use an API. One example of an API that can be used in python to correct spelling is REST API released by Microsoft.

Text Extraction & Cleanup

- System Specific Error Correction: or scenarios where data is
 - In the form of PDF documents.
 - PyPDF, PDFMiner, etc.: libraries to extract text from PDF documents
 - Not all PDF documents can be processed by these libraries. So, Good Luck!
 - In the form of scanned documents:
 - Text is extracted from scanned documents using Optical Character Recognition (OCR) using libraries like Tesseract
 - The percentage of errors in an OCR output depends on the quality of the scanned document
 - To correct the misspelling:
 - Use a spell checker like pyenchant
 - Use Neural Network architecture to train word/character-based language models, which are used for correcting OCR text based on the context
 - In the form of speech: use Automatic Speech Recognition (ASR)

Pre-Processing

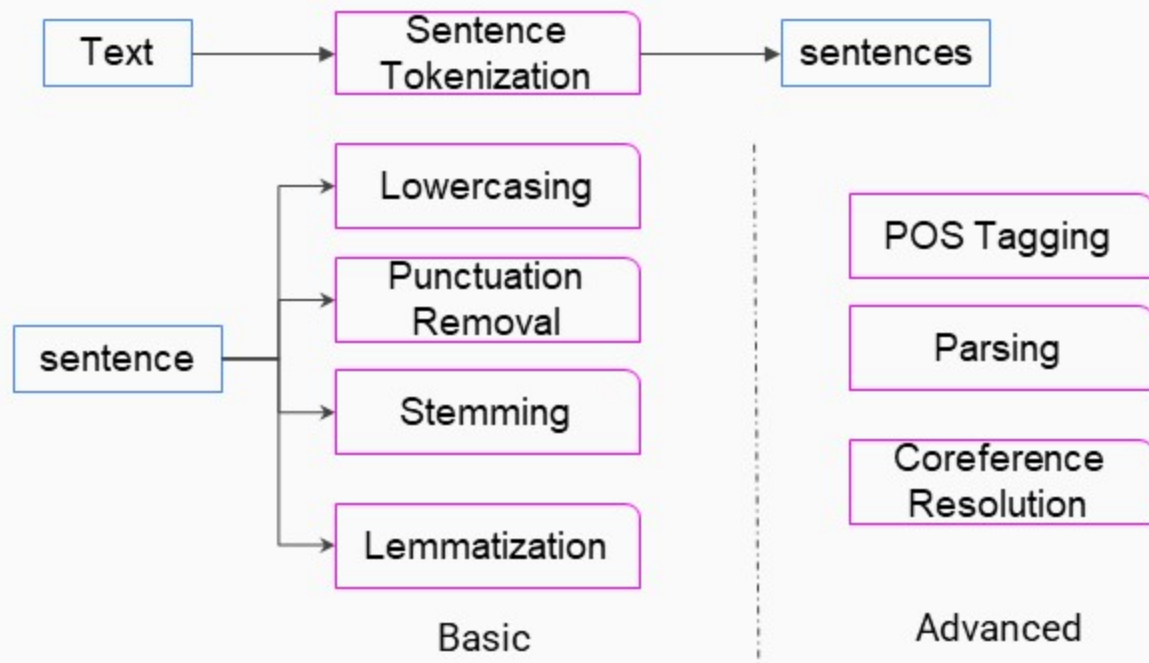


Pre-Processing

Some common pre-processing steps:

- Preliminaries:
 - Sentence segmentation & word tokenization
 - `from nltk.tokenize import sent_tokenize, word_tokenize`
- Frequent steps: stop word removal, stemming & lemmatization, removing digits/punctuation, lowercasing, etc.
- Advanced processing: POS Tagging, parsing, coreference resolution, etc.

Pre-Processing



Feature Engineering



Feature Engineering/Extraction: Text Representation

- Feature engineering refers to the task of transforming text into a numeric vector that can be understood by ML algorithms
- 2 different approaches for feature engineering:
 - Classical NLP and traditional ML pipeline
 - Usually handcrafted, inspired by the task at hand and the domain of knowledge: e.g. for a sentiment classification on product reviews, we can count the number of positive & negative words in each review and hence transform the textual data to numerical to be fed to ML models.
 - Advantages:
 - Interpretable models
 - A feature influence is quantifiable
 - Disadvantages:
 - The fact that they are handcrafted can become a bottleneck for both model performance.
 - Classical NLP and DL pipeline
 - Raw data is directly fed to the model; the model directly learns the features from the data
 - Advantages:
 - Learnt features are more in line with the task and achieve an improved performance
 - Disadvantages:
 - Model loses interpretability (very important in a business driven use case)

Modeling



Building a Model

- Start with simple heuristics
 - Handcrafted rules via regex, dictionaries, etc.
 - Tools for defining advanced regular expressions: Stanford NLP's TokenRegex and Spacy's rule-based matching
- Combine heuristics directly or indirectly with the ML model
 - 1) Create a feature from the heuristic for your ML model
 - 2) Pre-process your input to the ML model
 - 3) If possible start by using NLP Service Providers such as Google Cloud Natural Language, amazon comprehend, Microsoft Azure Cognitive Services, IBM Watson Natural Language Understanding, which provide off-the-shelf APIs to solve various NLP tasks.
Thus, you can have an estimate of the feasibility of the task and of the data quality

Building THE model

- Ensemble and stacking
- Better feature engineering
- Transfer Learning
- Reapplying heuristics
- Active learning

Tips on what decision path to make given your data volume and quality

Data attribute	Decision path	Examples
Large data volume	<ul style="list-style-type: none">• Use techniques that requires more data, like DL.• Use a richer set of features• Apply unsupervised techniques if the data is sufficiently large but unlabelled	Having lots of reviews and metadata associated with them, a sentiment analysis can be built from scratch
Small data volume	<ul style="list-style-type: none">• Need to start with rule-based or traditional ML solutions that are less data hungry• Adapt cloud-based APIs and generate more data with weak supervision• Use transfer learning if there is a similar task that has a large data	The situation at the start of a new project
Data quality is poor and the data is heterogeneous in nature	More data cleaning and data preprocessing might be required	This involves issues like code switching, unconventional languages, transliteration*, noise (like social media text)
Data quality is good	Apply off-the-shelf algorithms or cloud APIs	Legal text or newspapers
Data consists of full-length documents	Choose the right strategy for breaking the document into lower levels, like paragraphs, sentences, or phrases, depending on the problem	Document classification, review analysis, etc.

*example of a transliteration: وعد → waad, example of a translation: وعد → promesse

evaluation



Evaluation

Model evaluation consists of the measure of the model's performance on **unseen** data.

It depends on two factors:

1. Using the right metric
2. Following the right evaluation process

There are two types of evaluation:

1. Intrinsic: evaluation of the intermediate objective
2. Extrinsic: evaluation of the final objective

Intrinsic evaluation

- Measured on a test set consisting of the ground truth labels (human annotated, correct answers, etc.)
- Labels could be binary (for classification), words (for named entity recognition) or large texts (translated text)
- Intrinsic evaluation can be automated most of the time. For some cases, like machine translation or summarization, it can be hardly automated, because it is rather subjective

Popular Metrics and their NLP application

Metric	Description	Applications
Accuracy	For categorical or discrete output. It is the % of correct predictions compared to the total predictions.	Classification tasks, such as sentiment classification (multiclass), paraphrase detection (binary), etc.
Precision	The model's exactness i.e., the % of correct predictions given all the positive cases (the class of interest).	Classification tasks where mistakes in the positive class are more costly than in the negative one (i.e., healthcare)
Recall	The model's sensitivity, the % of correct positives given all positive predicted cases	Classification tasks where retrieving positive results is more important (e-commerce search, information-retrieval like search engines)
F1 score	Trade-off between precision and recall	Used with accuracy in most classification results, entity extraction, retrieval-based questions answering, etc.
AUC	Captures the count of correct positive predictions versus the count of incorrect positives as we vary the threshold for prediction.	Measures the model's quality independent of the prediction threshold. It finds the optimal prediction threshold for a classification task.

Popular Metrics and their NLP application

Metric	Description	Applications
Mean Reciprocal rank (MRR)	Evaluates the responses, of an NLP system, ordered by the probability of correctness. It is the sum of inverse ranks.	Information retrieval tasks like article search, e-commerce search, etc.
Mean Average Precision (MAP)	Like MRR, with one difference it computes the mean precision of retrievals.	Information retrieval
Root Mean Squared Error (RMSE)	For real-value predictions.	Regression tasks (temperature, stock price, etc. prediction)
Mean Absolute Percentage error (MAPE)	For continuous predictions.	Regression tasks, used with RMSE
Bilingual evaluation understudy (BLEU)	Captures the amount of n-gram overlap between the output sentence and the reference ground truth sentence	Machine-Translation tasks, text summarization, etc.

Popular Metrics and their NLP application

Metric	Description	Applications
Metric for Evaluation of Translation with Explicit Ordering (METEOR) https://en.wikipedia.org/wiki/METEOR	An improved version of BLEU. It compares generated words not only to exact target words but also their synonymns and stemmed versions.	Machine-Translation tasks.
Recall-Oriented Understudy for Gisting Evaluation (ROUGE)	A metric to compare generated text versus ground truth (reference text), based on recall.	Text Summarization
Perplexity	Computes the confusion of an NLP model.	Language models. Language-generation tasks.

MRR

Query	Proposed Results	Correct response	Rank	Reciprocal rank
cat	catten, cati, cats	cats	3	1/3
torus	torii, tori , toruses	tori	2	1/2
virus	viruses , virii, viri	viruses	1	1

Given those three samples, we could calculate the mean reciprocal rank as $(1/3 + 1/2 + 1)/3 = 11/18$ or about 0.61.

Considering n-grams

Let's try computing the BLEU** score on two other candidate translations *C3* and *C4* to check if everything looks fine.

- *R1*: The cat is on the mat.
- *R2*: There is a cat on the mat.
- *C3*: There is a cat on the mat.
- *C4*: Mat the cat is on a there.

$$\text{BLEU}^{**}_1(\text{C3}) = 7/7 = 1.0$$

$$\text{BLEU}^{**}_2(\text{C3}) = 6/6 = 1.0$$

$$\text{BLEU}^{**}_1(\text{C4}) = 7/7 = 1.0$$

$$\text{BLEU}^{**}_2(\text{C4}) = 0/6 = 0.0$$

- *R1*: The cat is on the mat.
- *R2*: There is a cat on the mat.
- *C5*: There is a cat.

The scores are:

$$\text{BLEU}^{**}_1(\text{C5}) = 4/4 = 1.0$$

$$\text{BLEU}^{**}_2(\text{C5}) = 3/3 = 1.0$$

Intrinsic evaluation metrics per application

- Classification: confusion matrix, precision, recall, f1 score
- Information search and retrieval: ranking-based metrics such Mean Reciprocal Rank (MRR), (Mean Average Precision) MAP, etc.
- Text-generation:
 - For machine translation: Bilingual evaluation Understudy (BLEU) and (Metric for Evaluation of Translation with Explicit ORdering) METEOR
 - For dialog generation: perplexity

Extrinsic Evaluation

- Good intrinsic evaluation does not mean a good extrinsic evaluation.
- In industrial projects, it is the extrinsic evaluation that matters.
- Thus, why do we do intrinsic evaluation at all?
- Extrinsic evaluation often includes project stakeholders outside the AI team, sometimes even end users. Intrinsic evaluation can be mostly done by the AI team itself.
- Only when we get consistently good results in intrinsic evaluation should we go for extrinsic evaluation.

Post-Modeling Phases



Post-Modeling Phases

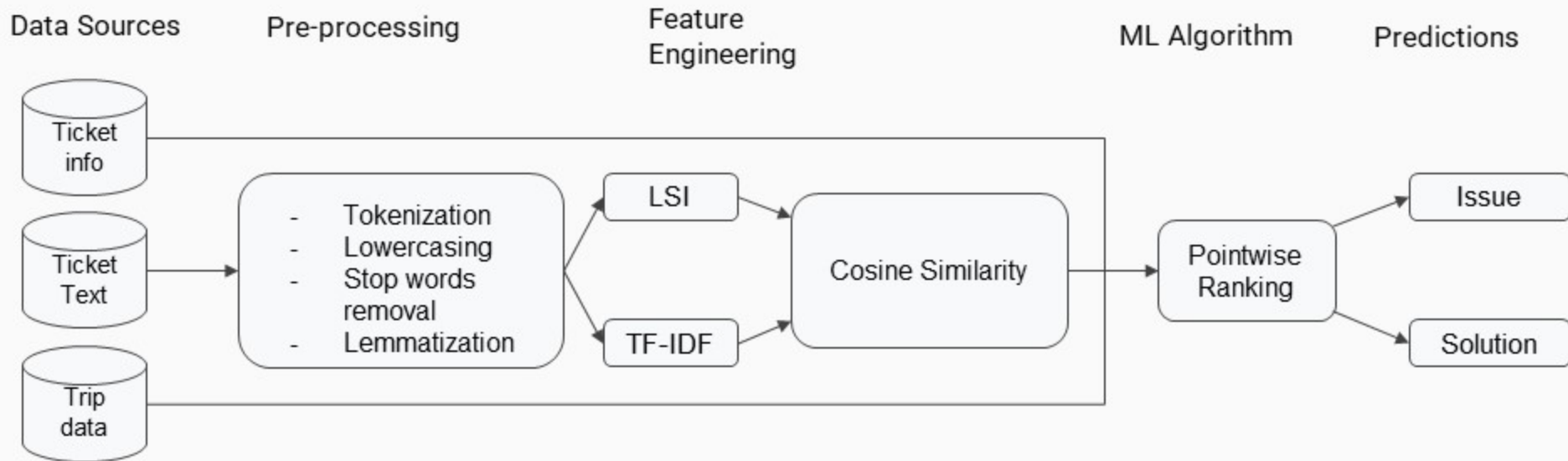
- **Deployment:**
 - The NLP model needs to be deployed in a production environment; it entails plugging the NLP model into a broader system
 - This might involve ensuring that the input/output data pipelines are in order and that the NLP module is scalable under heavy load
- **Monitoring:**
 - NLP model's monitoring consists of ensuring that the outputs produced by the model on a daily basis make sense. This is usually done through a performance dashboard showing the model params and key performance indicators (KPI)
- **Model updating:**
 - It is necessary when data or concept drift is detected in our model. In other terms, the outputs of the NLP model are no longer persistent
 - It is necessary when the NLP model deployed was not built on a large amount of data. When deployed, we start gathering new data and therefore iterate the model on these new data

Model Updating

Situation	Decision	Application
More data is generated/ is available post-deployment	New data used to re-train model. Online learning to train the model automatically on a daily basis.	Abuse-detection systems
No data available post-deployment	Manual labeling to improve the model's performance.	Use cases with no direct feedback
Low latency model required or real-time response model (online)	Use models that can be inferred quickly, or use a memorization strategy (caching, bigger computing power)	Chatbots, emergency tracking system, etc.
Offline model	Use advanced slower models=> optimizing costs	Systems allowing batch-training: retail product, etc.

Use Case Study

Uber ticketing assistant



"The greatest challenges humans face throw-out their lives are two:

1. the challenge of where to start
2. the challenge of when to stop."

Sameh Elsayed