Questions on Chapter 2

1. What are the key stages of a generic pipeline for NLP system development?

Data acquisition
Text cleaning
Pre-processing
Feature engineering
Modeling
Evaluation
Deployment
Monitoring and model updating

2. How can we get data required for training an NLP technique?

Use a public dataset
Scrape data
Product intervention
Data augmentation
Synonym replacement
Bigram flipping
Replacing entities
Adding noise to data

3. List the different data augmentation methods?

Synonym replacement : Randomly choose "k" words in a sentence that are not stop words. Replace these words with their synonyms

Back translation : Say we have a sentence, S1, in English. We use a machine-translation library like Google Translate to translate it into some other language—say, German. Let the corresponding sentence in German be S2. Now, we'll use the machine-translation library again to translate back to English. Let the output sentence be S3.
We'll find that S1 and S3 are very similar in meaning but are slight variations of each other.

Bigram flipping : Divide the sentence into bigrams. Take one bigram at random and flip it.

Replacing entities : Replace entities like person name, location, organization, etc., with other entities in the same category.

Adding noise to data : we can add a bit of noise to data to train robust models.

Advanced techniques There are other advanced techniques and systems that can augment text data. Some of the notable ones are:

Snorkel : Using Snorkel, a large training dataset can be “created”—without manual labeling .
Easy Data Augmentation (EDA) and NLPAug : These two libraries are used to create synthetic samples for NLP.
Active learning : where the learning algorithm can interactively query a data point and get its label.

4. Data can be collected from PDF files, HTML pages, and images, how this data can be cleaned based on their sources?

Text extraction and cleanup refers to the process of extracting raw text from the input data by removing all the other non-textual information, such as markup, metadata, etc., and converting the text to the required encoding format. Typically, this depends on the format of available data in the organization

5. Using dot (.) to segment sentences can cause problems, explain how?

As a simple rule, we can do sentence segmentation by breaking up text into sentences at the appearance of full stops and question marks. However, there may be abbreviations, forms of addresses (Dr., Mr., etc.), or ellipses (...) that may break the simple rule.

6. What are the frequent steps in the data pre-processing phase?

Stop word removal, stemming and lemmatization, removing digits/punctuation, lowercasing.

7. With examples, explain the differences between segmentation and lemmatization.?

Segmentation is the process of divide the text into sentences at appearance of full stops or question marks .

Lemmatization is the process of mapping all the different forms of a word to its base word, or lemma. (was -> be) (better -> good) (meeting -> meeting)

8. What is the difference between code mixing and transliteration?

Code mixing refers to this phenomenon of switching between languages

Transliteration : When people use multiple languages in their write-ups, they often type words from these languages in Roman script, with English spelling. So, the words of another language are written along with English text.

9. Describe the concept coreference resolution.?

Coreference resolution is the task of finding all expressions that refer to the same entity in a text. It is an important step for a lot of higher level NLP tasks that involve natural language understanding such as document summarization, question answering, and information extraction.

10. Explain the feature engineering for classical NLP versus DL-based NLP?

Classical NLP/ML Pipeline Feature engineering is an integral step in any ML pipeline. Feature engineering steps convert the raw data into a format that can be consumed by a machine.

DL Pipeline : the raw data (after preprocessing) is directly fed to a model. The model is capable of “learning” features from the data , so they generally give improved performance.

11. How to combine heuristics directly or indirectly with the ML model?

Create a feature from the heuristic for your ML model : When there are many heuristics where the behavior of a single heuristic is deterministic it's best to use these heuristics as features to train your ML model.

Pre-process your input to the ML model If the heuristic has a really high prediction for a particular kind of class, then it's best to use it before feeding the data in your ML model.

12. What is the difference between models ensembling and stacking?

model stacking : we can feed one model's output as input for another model, thus sequentially going from one model to another and obtaining a final output.

model ensembling : pool predictions from multiple models and make a final prediction.

13. Which modeling technique can be used in the following cases: small data, large data, poor data quality, and good data quality?

Small data volume -> Need to start with rule-based or traditional ML solutions.
Large data volume -> Can use techniques that require more data, like DL.
Data quality is poor -> More data cleaning and pre-processing might be required.
Data quality is good -> Can directly apply off-the-shelf algorithms.

14. What is the difference between intrinsic and extrinsic evaluation?

intrinsic evaluation : The output of the NLP model on a data point is compared against the corresponding label for that data point, and metrics are calculated based on the match (or mismatch) between the output and label.

extrinsic evaluation : focuses on evaluating the model performance on the final objective

15. What are the metrics that can be used in: classification, measuring model quality, information retrieval, predication, machine translation, and summarization tasks.?

Classification -> Accuracy , F1 score
measuring model quality -> AUC
information retrieval -> MRR (mean reciprocal rank) , MAP (mean average precision)
machine translation -> METEOR , BLEU (bilingual evaluation understudy)
summarization tasks -> ROUGE

16. Describe deploying, monitoring, and updating phases of NLP the pipeline.?

Deploying -> model needs to be deployed in a production environment as a part of a larger system, NLP module is typically deployed as a web service.

Monitoring -> model performance is monitored constantly after deploying, as we need to ensure that the outputs produced by our models daily make sense

Updating -> Once the model is deployed and we start gathering new data, we'll iterate the model based on this new data to stay current with predictions

17. Explain how the NLP pipeline is different from a language to another?

The pipeline for some languages may be very similar to English, whereas some languages and scenarios may require us to rethink how we approach the problem.

18. Describe the NLP pipeline for ranking tickets in a ticketing system by Uber.?

The information needed to identify the ticket issue and select a solution
After cleaning up the text by removing HTML tags (not shown in the figure), the pre-processing steps consist of tokenization, lowercasing, stop word removal, and lemmatization.
After pre-processing, the ticket text is represented as a collection of words The next step in this pipeline is feature engineering. The bag of words we obtained earlier is fed to two NLP modules—TF-IDF (term frequency and inverse document frequency) and LSI (latent semantic indexing)—which are used to understand the meaning of a text using this bag of words representation.

Uber collects the historical tickets for each solution from their database, forms a bag-of-words vector representation for each solution, and creates a topic model based on these representations. An incoming ticket is then mapped to this topic space of solutions, creating a vector representation for the ticket. Cosine similarity is a common measure of similarity between any two vectors. It is used to create a vector where each element indicates the ticket text's similarity to one solution. Thus, at the end of this feature engineering step, we end up with a representation indicating the ticket text's similarity to all possible solutions

In the next stage, modeling, this representation is combined with ticket information and trip data to build a ranking system that shows the three best solutions for the ticket. The matches are then ranked based on a scoring function. The next step in our pipeline is evaluation. How does evaluation work in this context? While the evaluation of model performance itself can be done in terms of an intrinsic evaluation measure such as MRR, the overall effectiveness of this approach is evaluated extrinsically.