به نام خدا

درس مبانی پردازش زبان و گفتار

کارگاه Hazm

سینا علی‌نژاد

۹۹۵۲۱۴۶۹

سوال ۱-

1. Define the product requirements and use cases: The first step is to define the product requirements and use cases. This involves understanding the target audience, the environments in which the product will be used, the desired performance metrics, and the constraints and limitations of the system.
2. Collect and prepare data: The next step is to collect and prepare data for training and testing the ASR system. This involves gathering speech data from a diverse set of speakers, transcribing the data, and splitting it into training, validation, and test sets.
3. Choose a modeling approach: There are several modeling approaches for ASR, including hidden Markov models (HMMs), deep neural networks (DNNs), and hybrid HMM-DNN systems. The choice of modeling approach will depend on the product requirements, the available data, and the computational resources.
4. Train the model: Once the modeling approach is chosen, the next step is to train the ASR model. This involves optimizing the model parameters to minimize the error between the predicted and true transcriptions.
5. Evaluate the model: After the model is trained, it is important to evaluate its performance on a held-out test set. This involves measuring metrics such as word error rate (WER), sentence accuracy, and latency.
6. Fine-tune and optimize the model: Based on the evaluation results, the model may need to be fine-tuned and optimized to improve its performance. This involves adjusting the model architecture, hyperparameters, and training procedures.
7. Deploy the model: Once the ASR model is trained and optimized, it can be deployed in the target environment. This involves integrating the model with the product software, testing it in real-world conditions, and monitoring its performance over time.
8. Maintain and update the model: ASR models may need to be updated and maintained over time to adapt to changes in the environment, the users, and the product requirements. This involves collecting and annotating new data, retraining the model, and deploying updates.

سوال ۲-

When we have several ASR models available and each of them produces 16 outputs with the order of probability of correctness, we need to select and output the final 16 outputs from this large number of outputs. There are at least two methods that are commonly used in practice:

1. Fusion/Combination: In this method, we combine the outputs of multiple ASR models to produce a single set of outputs. The idea is that different models may be better at recognizing different parts of the speech, and by combining their outputs, we can obtain a more accurate transcription. There are different ways to combine the outputs, such as:

- Voting/Majority: We select the output that is most frequently produced by the models.
- Weighted Voting: We assign a weight to each model based on its performance and use the weights to calculate a weighted sum of the outputs.
- Confidence-based Combination: We use the confidence scores of the models to combine the outputs. For example, we can select the output with the highest confidence score or use a weighted sum of the confidence scores.

1. Reranking: In this method, we first select a subset of the outputs produced by the ASR models and then rerank them based on a scoring function. The idea is that the initial set of outputs may contain errors and by reranking them, we can obtain a more accurate transcription. There are different ways to rerank the outputs, such as:

- Language Model (LM) Rescoring: We use a language model to rescore the outputs based on their likelihood of occurring in the language. The language model can be trained on a large corpus of text data.
- Neural Network (NN) Rescoring: We use a neural network to rescore the outputs based on their features, such as the acoustic features, the language features, and the context features. The neural network can be trained on a large corpus of speech and text data.
- Hybrid LM-NN Rescoring: We use a combination of a language model and a neural network to rescore the outputs.

Both fusion/combination and reranking methods have their advantages and disadvantages, and the choice of the method depends on the specific application, the available resources, and the desired performance.

سوال ۳-

1. POS Tagger (postagger): A POS tagger is a function that assigns a part of speech (POS) tag to each word in a sentence. POS tags indicate the grammatical role of a word in a sentence, such as noun, verb, adjective, and adverb.
2. Tagger: A tagger is a more general term that can refer to any function that assigns a label or a tag to a linguistic unit, such as a word, a phrase, or a sentence. POS tagger is a type of tagger.
3. Chunker: A chunker is a function that groups words or phrases into syntactic or semantic units, such as noun phrases, verb phrases, and clauses. Chunking is a shallow parsing technique that does not produce a full parse tree.

4. Stemmer: A stemmer is a function that reduces a word to its stem or root form by removing the prefixes and suffixes. Stemming is a text normalization technique that is used to conflate the inflected forms of a word to a single representation.
5. Lemmatizer: A lemmatizer is a function that reduces a word to its lemma or dictionary form by using a morphological analysis. Lemmatization is a more sophisticated text normalization technique that can handle irregular and complex words.
6. Formalizer: A formalizer is a function that converts an informal or colloquial text into a formal or standard form. Formalization is a text enhancement technique that is used to improve the readability, clarity, and style of a text.
7. Normalizer: A normalizer is a function that converts a text into a standard or canonical form by removing the variations and inconsistencies. Normalization is a text preprocessing technique that is used to facilitate the subsequent NLP tasks, such as tokenization, POS tagging, and parsing.
8. Parser: A parser is a function that analyzes the syntactic or semantic structure of a sentence and produces a parse tree or a dependency graph. Parsing is a fundamental NLP task that is used to extract the meaning and the relations of the words in a sentence.
9. Embedder: An embedder is a function that maps a linguistic unit, such as a word, a phrase, or a sentence, into a dense and continuous vector space. Embedding is a text representation technique that is used to capture the semantic and syntactic properties of the linguistic units.
10. Word Embedder: A word embedder is a type of embedder that maps words into vectors. Word embedding is a widely used text representation technique that is used in many NLP applications, such as machine translation, sentiment analysis, and text classification.