NATURAL LANGUAGE PROCESSING (PRACTICE)

NLP 251 - Lab 1: Lab Introduction

Department of Computer Science and Engineering
Ho Chi Minh University of Technology, VNU-HCM

General Introduction

- The lab sessions take place over 15 weeks (2 periods per week).
- Each week usually consists of two types of exercises: **In-class Exercise** (Deadline: 1 day) and **Home-class Exercise** (Deadline: 1–2 weeks).
- The exercises will include two formats: coding exercises and mathematical exercises.
- Each exercise allows up to 7 late submission days. For every late day, 1 point will be deducted from the total score.

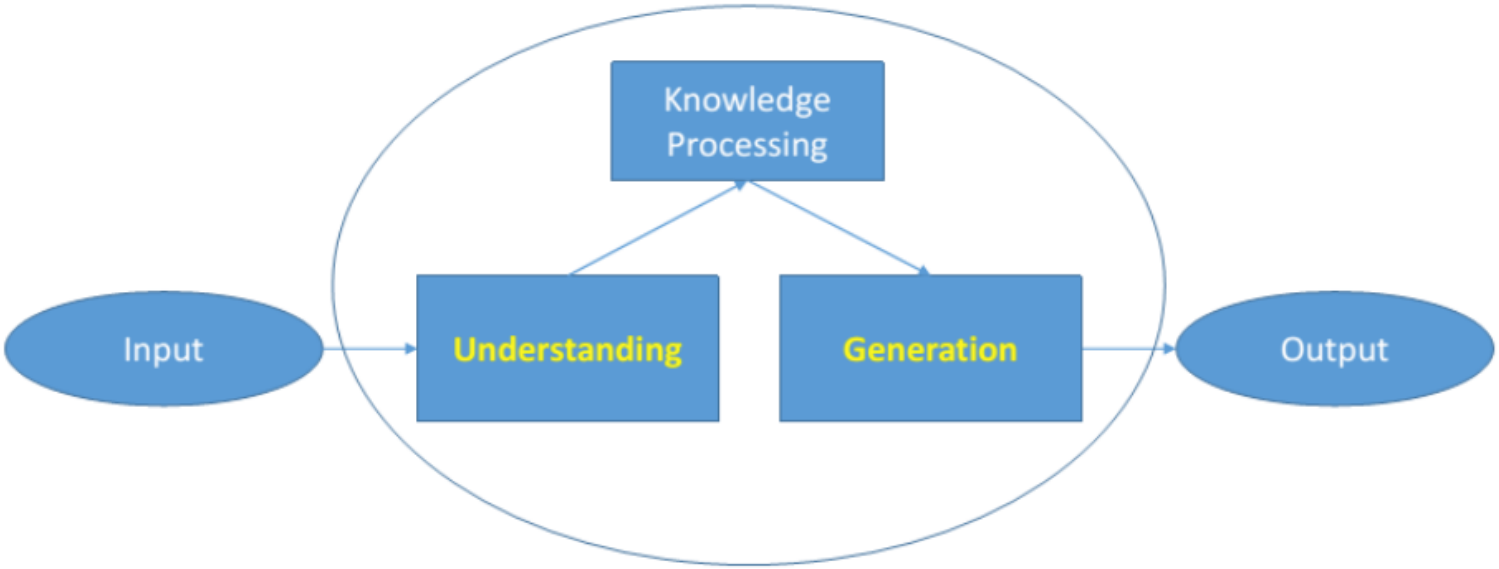
Content Overview

- Preprocessing
- Language Models
- Vector Semantics and Embeddings
- Linear Models
- Optimizers
- Neural Networks
- Recurrent Neural Networks
- Sequence-to-Sequence Models
- Transformer Architectures
- Applications

Natural Language Processing

Introduction

NLP (Natural Language Processing) is the field of study and application aimed at enabling computers to understand and generate natural language.

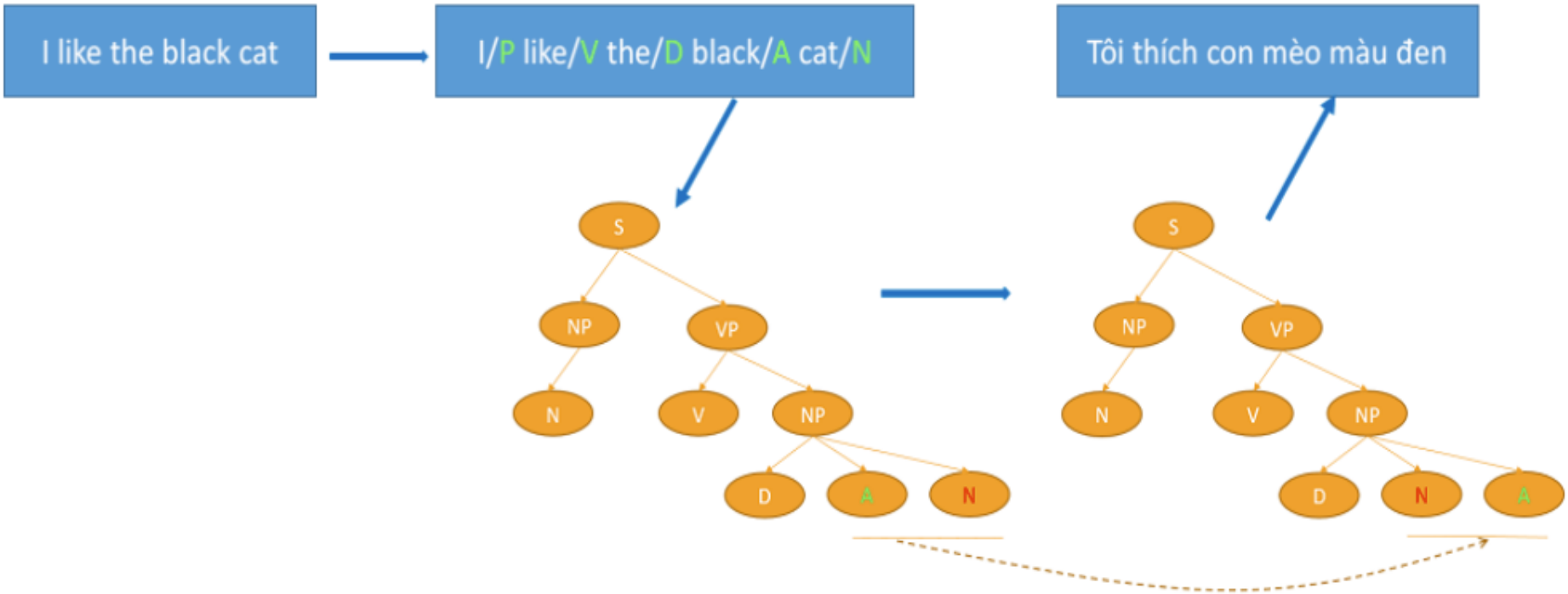


What makes language hard

1. Ambiguity
2. Context Dependency
3. Cultural and Regional Differences
4. Background Knowledge
5. Massive amounts of unstructured and semi-structured data with new challenges

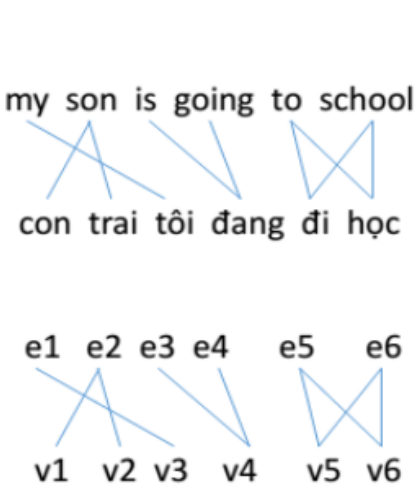
History of NLP

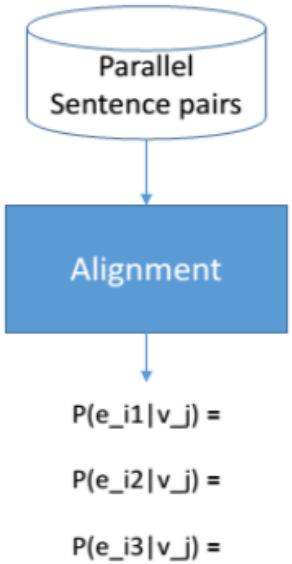
1 From 1950s -> 1980s: Rule-based Apprroach + experts

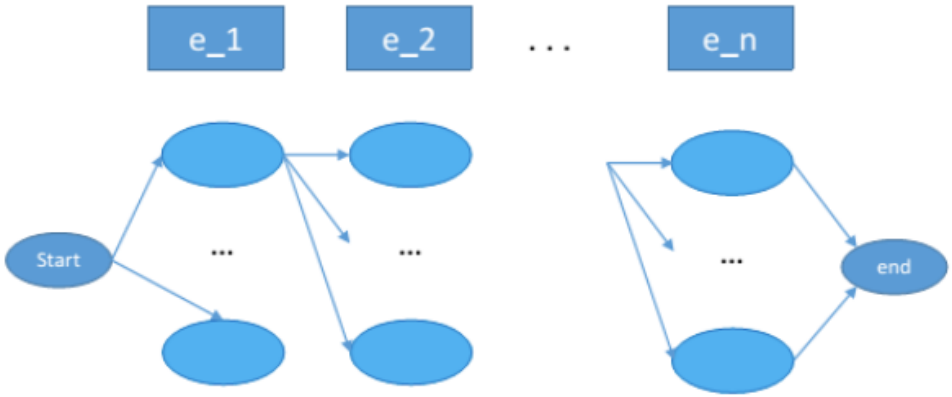


History of NLP

2 Late 1980 -> 2000s: Applying Statiscal Machine Learning

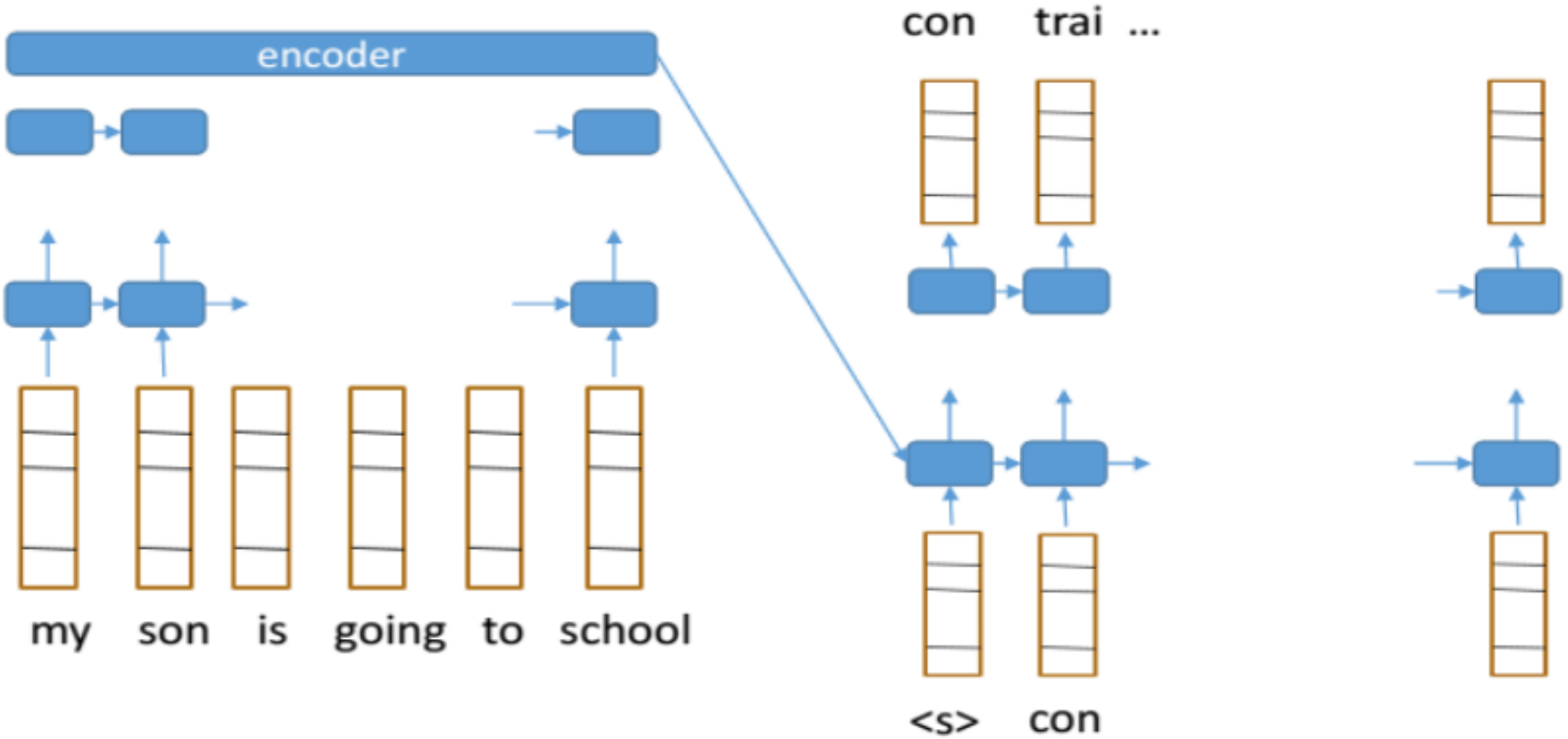






History of NLP

3 Late 2010s: Appling Deep Learning: Representation learning



Natural Language Processing Task

- **Text Classification:** Sentiment analysis, Spam detection, ...
- **Machine Translation:** Translating text from one language to another
- **Text Summarization:** Generating a concise summary of a longer text
- **Question Answering:** Answering questions based on a given context or document

Natural Language Processing Task

- **Language Modeling:** Predicting the next word or sequence in a sentence
- **Text Generation:** Generating coherent and contextually relevant text
- **Topic Modeling:** Discovering the abstract topics that occur in a collection of documents.
- **Word Embedding:** Learning continuous vector representations of words
- ....

Traditional Methods

- **Rule-Based Systems**:
  - Grammar rules
  - Handcrafted features
- **Statistical Models**:
  - N-grams
  - Hidden Markov Models (HMMs)
  - Conditional Random Fields (CRFs)
- **Bag of Words (BoW)**:
  - Simple representation of text
  - Ignores word order
- **TF-IDF (Term Frequency-Inverse Document Frequency)**:
  - Weighs the importance of words
  - Widely used in information retrieval

Traditional Methods

- **Word Embeddings:**
  - Word2Vec, GloVe, FastText
  - Capture semantic relationships between words
- **Sequence Models:**
  - Recurrent Neural Networks (RNNs)
  - Long Short-Term Memory (LSTM)
  - Gated Recurrent Units (GRUs)
- **Transformers:**
  - BERT (Bidirectional Encoder Representations from Transformers)
  - GPT (Generative Pre-trained Transformer)
  - T5 (Text-to-Text Transfer Transformer)
- **Pre-trained Language Models:**
  - Fine-tuning for specific tasks
  - Transfer learning

Important Libraries

- **nltk:** For preprocessing
- **word_tokenize:** For Vietnamese preprocessing
- **Stanford NLPCore:** For parsing
- **Sklearn:** model_selection, metric and train_test_split
- **Word2Vec:** Word Embedding
- **Pytorch, TensorFlow:** Deeplearning Framework
- **Transformer, Huggingface:** Pretrained models

Example Math Exercise

Problem

Suppose we have a collection of 3 documents document 1 says "lions eat fat cats"; document 2 says "cats eat fat mice"; document 3 says "mice eat fat cheese". Compute the cosine similarity of document 1 and document 2 with equal TF weighting:

1. 0.25
2. 0.0
3. 0.5
4. 0.75
5. 1.0

Problem

Consider the following corpus of 3 sentences:

- *I am here*
- *Who am I*
- *I would like to go*

Caculate P(here Iam) assuming a bi-gram language model.

1. 1/2
2. 2/3
3. 1/3
4. 1

Example Code Exercise

- Get text from web data, pdf, word, txt.
- Check for plagiarism by word embedding and similarity scores.
- Building a spell-checking model based on n-grams
- Labeling parts of speech with a Hidden Markov Model (HMM)
- Training Sequence Model.
- Finetune Pretrained Model.

Q&A

ANY QUESTIONS?

THANKS FOR LISTENING!