Natural Language Processing

Dr. Chetana Gavankar, Ph.D,

IIT Bombay-Monash University Australia

Chetana.gavankar@pilani.bits-pilani.ac.in

BITS Pilani

Pilani Campus

innovate achieve lead

BITS Pilani
Pilani Campus

Session 7

These slides are prepared by the instructor, with grateful acknowledgement of Jurafsky and Martin d many others who made their course materials freely available online.

Session Content

- POS Tagging with Bi-LSTM-CRF
- Transformer model
- POS Tagging with Pre-trained Models (BERT, RoBERTa)
- Prompting techniques for POS Tagging
- LLM agents and POS Tagging

Bi-LSTM-CRF Architecture

The Go-To Neural Baseline for Sequence Tagging.

Concept: Combines a powerful feature extractor (Bi-LSTM) with a structured prediction layer (CRF) to ensure valid tag sequences.

Input Layer: Word Embeddings (e.g., Word2Vec, GloVe) + **Character Embeddings** (handles unknown words).

Bi-LSTM Layer (Feature Extractor):

- Processes the sentence in **forward and backward** directions.
- Captures **long-range dependencies** and **bidirectional context**.

CRF Layer (Structured Decoder):

- **Crucial for POS:** It learns constraints (e.g., a **Verb** is unlikely to follow a **Determiner**).
- Finds the sequence of tags that has the **highest overall probability** across the entire sentence.

Bi-LSTM-CRF

- LSTM: Solves the Vanishing Gradient problem of simple RNNs using Gates (Input, Forget, Output).
- Bidirectional Advantage: The tag for 'bank' in "I walked to the river bank." depends on both 'river' (left context) and '.' (right context).
- CRF Output: Instead of just picking the highest probability tag at each word (which can lead to impossible sequences),
- CRF layer calculates

$$P(\mathbf{y}|\mathbf{x}) = \frac{\prod_{i=1}^{N} \Psi_i(y_{i-1}, y_i, \mathbf{x})}{\sum_{\mathbf{y}' \in \mathcal{Y}(\mathbf{x})} \prod_{i=1}^{N} \Psi_i(y'_{i-1}, y'_i, \mathbf{x})}$$

- Goal: Maximize the numerator (score of the true path) relative to all possible paths.

Performance & Applications

- Typical Accuracy: approx. 97% - 98% on Penn Treebank.
- Pros: Highly accurate, automatically learns features, good for low-resource languages (with transfer learning).
- Cons: Slower training than non-recurrent models, replaced as SOTA by Transformers.

Transformer Model

- Key Idea: Replaces recurrence (RNN/LSTM) with the Self-Attention Mechanism.
- Self-Attention: Allows the model to weigh the relevance of every other word in the sentence simultaneously, regardless of distance.
- Example: When tagging "He," the model instantly links it to "ran" and "ball" equally well.
- Model Architecture: Utilizes multi-layer Encoder blocks consisting of Multi-Head Attention and Feed-Forward Networks.

POS Tagging with Pre-trained Models (BERT, RoBERTa)

Method: Fine-tuning a Pre-trained Language Model (PLM).

Pre-training: The model learns grammar and world knowledge on trillions of words (e.g., predicting masked words).

Fine-Tuning: A simple classification layer is added on top of the PLM's output vectors.

The entire model is then trained for a few epochs on the specific POS tagging corpus.

Output: The model produces a highly contextualized vector for each token, which the final layer maps to the appropriate tag (e.g., CLS --> NN).

Transformer Advantages & Impact

- SOTA Performance: approx. 98.5% - 99.5% accuracy.
- Efficiency: Highly parallelizable (faster training than RNNs).
- Context: Captures the deepest contextual meaning, solving nearly all lexical ambiguity issues.
- Impact: This approach is the standard for POS tagging in modern libraries like spaCy and Hugging Face Transformers.

Prompting

- Concept: Leverage the vast linguistic knowledge and zero-shot capabilities of Large Language Models (LLMs) (e.g., GPT-4, Llama).
- Traditional Fine-Tuning is Replaced by Prompting:
- Prompting: The POS task is framed as a natural language instruction.
- Example Prompt: "Tag the following sentence using the Penn Treebank tagset. Output only the word/tag pairs. Sentence: The concert was great."
- Zero-Shot/Few-Shot Learning: LLM can tag a sentence accurately without being explicitly fine-tuned on a POS corpus, based solely on its general pre-training.

LLM Agents & System Integration

LLM Agent: An LLM that is given the ability to use external tools to complete a task.

The POS Agent Flow:

Input: User gives a complex request (e.g., Analyze the sentiment of all nouns in the user reviews.)

LLM Agent: Breaks the task down.

Tool Use (POS Tagging): The LLM might decide to use a specialized, fast POS tagger (like a fine-tuned BERT model) as a tool for the labeling sub-task.

Integration: The agent feeds the tags back into its context to perform the final task (e.g., sentiment analysis on the words tagged as NN).

Benefit: Combines the LLM's reasoning/planning ability with the speed/accuracy of a specialized POS model.

Conclusion

- Modern POS Tagging is Solved?
- The task has achieved near-human performance thanks to Deep Learning.
- Current SOTA: Fine-tuned Transformer models (BERT/RoBERTa).
  Future Trend (LLMs): POS tagging is becoming an internal utility within larger LLM agent systems.
- The focus on how to use tags to power complex downstream AI systems.

References

- https://www.nltk.org/
- https://likegeeks.com/nlp-tutorial-using-python-nltk/
- https://www.guru99.com/pos-tagging-chunking-nltk.html
- https://medium.com/greyatom/learning-pos-tagging-chunking-in-nlp-85f7f811a8cb
- https://nlp.stanford.edu/software/tagger.shtml
- https://www.forbes.com/sites/mariyayao/2020/01/22/what-are--important-ai--machine-learning-trends-for-2020/#601ce9623239
- https://medium.com/fintechexplained/nlp-text-processing-in-data-science-projects-f083009d78fc
- https://www.nltk.org/book/ch02.html
- https://towardsdatascience.com/pos-tagging-using-crfs-ea430c5fb78b

References

- https://github.com/nadavo/MEMM-POS-Tagger
- https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.352.9257&rep=rep1&type=pdf
- https://www.alibabacloud.com/blog/hmm-memm-and-crf-a-comparative-analysis-of-statistical-modeling-methods_592049
- https://towardsdatascience.com/pos-tagging-using-rnn-7f08a522f849

POS tagging in Indian Languages

- https://www.nltk.org/book/ch05.html