Hang Dai
2300012956

Homework 3

FNLP
May 18, 2025

1 Implement the WordPiece algorithm (55 Credits)

1.1 Explain the principle of the WordPiece algorithm (25 Credits)

Let $\mathcal{D} = \{w^{(1)}, w^{(2)}, \dots\}$ be the training corpus and K the desired vocabulary size. WordPiece learns a subword vocabulary V by greedily maximizing the unigram-model likelihood of $\mathcal{D}$.

Unigram Model and Objective Each word $w \in \mathcal{D}$ is segmented under V into subword tokens $s_1, \dots, s_T$:

$$w = s_1 \oplus s_2 \oplus \cdots \oplus s_T.$$

Define the normalizer

$$Z = \sum_{u \in V} \text{count}(u),$$

and the unigram probability

$$p(s) = \frac{\text{count}(s)}{Z}.$$

The corpus log-likelihood is then

$$L(V) = \sum_{w \in \mathcal{D}} \sum_{t=1}^{T} \log p(s_t).$$

Initialization

- Start with V_0 containing all Unicode characters plus two special tokens: `<unk>` for unknown pieces and the continuation prefix `##`.
- Segment each $w \in \mathcal{D}$ into individual characters, e.g. "playing" $\rightarrow$ `p`, `##l`, `##a`, ..., `##g`.

Greedy Vocabulary Construction Repeat until $|V| = K$:

1. **Count adjacent pairs.** For each pair (a, b), let

$$c(a, b) = \sum_{w \in \mathcal{D}} \#\{\text{occurrences of } (a, b) \text{ in } w\}.$$

2. **Compute merge gain.**

$$\Delta L_{(a,b)} \approx c(a, b) \big[\log c(a, b) - \log c(a) - \log c(b)\big],$$

where $c(a) = \sum_x c(a, x)$ and $c(b) = \sum_x c(x, b)$.

3. **Merge the best pair.** Choose $(a^*, b^*) = \arg\max \Delta L_{(a,b)}$, add the new token a^*b^* to V.

4. **Resegment and recompute.** Using longest-match-first over the updated V, re-tokenize every $w \in \mathcal{D}$ and recompute all counts $c(\cdot)$ and $c(\cdot, \cdot)$.

Tokenization of New Text Given a new word w:

1. From the leftmost character, repeatedly select the longest subword in V matching the current substring (first piece without `##`, later pieces with `##`).

2. If no match exists at a position, emit `<unk>` and advance by one character.

Examples Below are several tokenization cases given a vocabulary

$V = \{$`un`, `##aff`, `##able`, `inter`, `##national`, `##ization`, `re`, `##dis`, `##tribute`, `##ion`, `<unk>`$\}$.

1. **Simple merge**:
$$\texttt{unaffable} \longrightarrow \texttt{un} + \texttt{\#\#aff} + \texttt{\#\#able}.$$

2. **Overlapping subwords**: tokenizing "redistribution":
$$\texttt{redistribution} \longrightarrow \texttt{re} + \texttt{\#\#dis} + \texttt{\#\#tribute} + \texttt{\#\#ion}.$$

3. **Longest-match-first**: tokenizing "internationalization":
$$\texttt{internationalization} \longrightarrow \texttt{inter} + \texttt{\#\#national} + \texttt{\#\#ization}.$$

4. **Fallback to `<unk>`**: tokenizing "xylophone" (assuming no matching subword at any position):
$$\texttt{xylophone} \longrightarrow \texttt{<unk>} + \texttt{<unk>} + \ldots$$

1.2 Implement WordPiece by completing the code in the attachment (file: wpalg.py). (10 Credits)

see in wpalg.py

1.3 Try to train a tokenizer by WordPiece and tokenize a given sentence. (10 Credits)

Original sentence: `nous etudions a l universite de pekin`

'n', '##o', '##u', '##s', 'e', '##tud', '##i', '##o', '##n', '##s', 'a', 'l', 'uni', '##v', '##e', '##rs', '##i', '##t', '##e', 'd', '##e', 'p', '##e', '##ki', '##n'

1.4 Example of an [UNK] Token and Why Llama Doesn't Need It

(a) An example that produces `[UNK]` Consider the short sentence

`Ball is fun`

and our Q3 vocabulary (which has no entry for the single letter "b"). WordPiece will segment it as:

`Ball is fun` $\longrightarrow$ [`[UNK]`, `##a`, `##l`, `##l`, `is`, `f`, `##u`, `##n`].

Because "B" (and hence "b") is not in the vocab, the first character falls back to `[UNK]`. The remaining letters "all" match via the continuation pieces `##a`, `##l`, `##l`.

(b) Why Llama's tokenizer has no `[UNK]` Llama uses a byte-level BPE: every input is first UTF–8–encoded to bytes (0–255), and then those bytes are merged via BPE. Since all 256 byte values are covered in the base alphabet, every character, no matter how rare or in what script, can be represented. There is literally nothing "unknown," so no `[UNK]` token is ever needed.

2 Expand BERT's tokenizer with WordPiece

Question 1 (10 Credits): Training the WordPiece tokenizer

We used Hugging Face's `tokenizers` library to train a WordPiece tokenizer on `pubmed_sampled_corpus.jsonline`.

- **Trainer & model:** `WordPieceTrainer` + `WordPiece`
- **Parameters:**
  - `vocab_size=50000`
  - `min_frequency=5`
  - `special_tokens=[''[PAD]'',''[UNK]'',''[CLS]'',''[SEP]'',''[MASK]']`
  - `continuing_subword_prefix="##"`
- **Resulting vocabulary size:** 50 000 subword tokens.

Question 2 (5 Credits): Selecting and sampling new tokens

To pick 5,000 domain-specific tokens, we first computed each token's frequency in the corpus and then removed any token either already present in `bert-base-uncased` or that can be a subword of words in `bert-base-uncased`. From the remainder, we selected the top 5,000 by frequency.

Sample of 50 new tokens

- `apoptosis`, `cytokines`, `microbiome`, `macrophages`, `neutrophils`
- `lymphocytes`, `fibroblasts`, `microglia`, `transcriptional`, `oxidative`
- `epigenetic`, `inflammation`, `endothelial`, `gene expression`, `proliferation`
- `mitochondria`, `cytoplasmic`, `nucleotides`, `mesenchymal`, `immunotherapy`
- `macrophage`, `neuroinflammation`, `epithelial`, `autophagy`, `phosphorylation`
- `histone`, `synaptic`, `astrocytes`, `genotype`, `microbial`
- `biomarkers`, `antioxidant`, `fibrosis`, `neurodegenerative`, `transgenic`
- `hematopoietic`, `nucleic`, `ligands`, `plasmid`, `lipid`
- `oncology`, `protease`, `dendritic`, `interleukin`, `cytotoxic`
- `bioinformatics`, `myocardial`, `immunoglobulin`, `pathogenesis`, `astrocytes`

Observations Many new tokens are full biomedical terms (e.g. `immunoglobulin`, `pathogenesis`, `neutrophils`). This should reduce subword fragmentation of disease names and technical processes.

Question 3 (10 Credits): Tokenization comparison & average length

(a) Three sample sentences from HoC

1. Flux measurements of pyruvate through lactate dehydrogenase and alanine aminotransferase in the bioreactor system were 12.18 +/- 0.49 nmols/sec/10(8) cells and 2.39 +/- 0.30 nmols/sec/10(8) cells , respectively , were reproducible in the same bioreactor , and were not significantly different over the course of 2 days .
2. In this study , we investigated the effect of As(2)O(3) on proliferation and apoptosis in human myeloma cell line and primary myeloma cells , and then we studied that As(2)O(3) exerts antimyeloma effects by inhibiting activity in the α-tubulin and Hsp90 through western blot analysis and immunoprecipitation .

3. An excess of the soluble TEM5 extracellular domain or an inhibitory monoclonal TEM5 antibody blocked contact inhibition of endothelial cell proliferation resulting in multilayered islands within the endothelial monolayer and increased vessel density during capillary formation .

(b) Tokenized results

Original BERT tokenizer

1. 'flux', 'measurements', 'of', 'p', '##yr', '##u', '##vate', 'through', 'lac', '##tate', 'de', '##hy', '##dro', '##genase', 'and', 'alan', '##ine', 'amino', '##tra', '##ns', '##fera', '##se', 'in', 'the', 'bio', '##rea', '##ctor', 'system', 'were', '12', '.', '18', '+', '/', '-', '0', '.', '49', 'nm', '##ols', '/', 'sec', '/', '10', '(', '8', ')', 'cells', 'and', '2', '.', '39', '+', '/', '-', '0', '.', '30', 'nm', '##ols', '/', 'sec', '/', '10', '(', '8', ')', 'cells', ',', 'respectively', ',', 'were', 'rep', '##rod', '##ucible', 'in', 'the', 'same', 'bio', '##rea', '##ctor', ',', 'and', 'were', 'not', 'significantly', 'different', 'over', 'the', 'course', 'of', '2', 'days', '.'

2. 'in', 'this', 'study', ',', 'we', 'investigated', 'the', 'effect', 'of', 'as', '(', '2', ')', 'o', '(', '3', ')', 'on', 'proliferation', 'and', 'ap', '##op', '##tosis', 'in', 'human', 'my', '##elo', '##ma', 'cell', 'line', 'and', 'primary', 'my', '##elo', '##ma', 'cells', ',', 'and', 'then', 'we', 'studied', 'that', 'as', '(', '2', ')', 'o', '(', '3', ')', 'ex', '##ert', '##s', 'anti', '##my', '##elo', '##ma', 'effects', 'by', 'inhibit', '##ing', 'activity', 'in', 'the', 'α', '-', 'tub', '##ulin', 'and', 'hs', '##p', '##90', 'through', 'western', 'b', '##lot', 'analysis', 'and', 'im', '##mun', '##op', '##re', '##ci', '##pit', '##ation', '.'

3. 'an', 'excess', 'of', 'the', 'soluble', 'te', '##m', '##5', 'extra', '##cellular', 'domain', 'or', 'an', 'inhibitor', '##y', 'mono', '##cl', '##onal', 'te', '##m', '##5', 'antibody', 'blocked', 'contact', 'inhibition', 'of', 'end', '##oth', '##elial', 'cell', 'proliferation', 'resulting', 'in', 'multi', '##layer', '##ed', 'islands', 'within', 'the', 'end', '##oth', '##elial', 'mono', '##layer', 'and', 'increased', 'vessel', 'density', 'during', 'cap', '##illa', '##ry', 'formation', '.'

Expanded tokenizer

1. 'flux', 'measurements', 'of', 'pyruvate', 'through', 'lactate', 'dehydrogenase', 'and', 'alanine', 'amino', 'transferase', 'in', 'the', 'bior', 'ea', '##ctor', 'system', 'were', '12', '.', '18', '+/-', '0', '.', '49', 'nmol', 's', '/', 'sec', '/', '10', '(', '8', ')', 'cells', 'and', '2', '.', '39', '+/-', '0', '.', '30', 'nmol', 's', '/', 'sec', '/', '10', '(', '8', ')', 'cells', ',', 'respectively', ',', 'were', 'reproducible', 'in', 'the', 'same', 'bior', 'ea', '##ctor', ',', 'and', 'were', 'not', 'significantly', 'different', 'over', 'the', 'course', 'of', '2', 'days', '.'
   Reduced by 17 words

2. 'in', 'this', 'study', ',', 'we', 'investigated', 'the', 'effect', 'of', 'as', '(', '2', ')', 'o', '(', '3', ')', 'on', 'proliferation', 'and', 'apoptosis', 'in', 'human', 'myeloma', 'cell', 'line', 'and', 'primary', 'myeloma', 'cells', ',', 'and', 'then', 'we', 'studied', 'that', 'as', '(', '2', ')', 'o', '(', '3', ')', 'exerts', 'anti', 'myeloma', 'effects', 'by', 'inhibiting', 'activity', 'in', 'the', 'α', '-', 'tubulin', 'and', 'hsp90', 'through', 'western', 'blot', 'analysis', 'and', 'immunoprecipitation', '.'
   Reduced by 21 words

3. 'an', 'excess', 'of', 'the', 'soluble', 'te', '##m', '##5', 'extracellular', 'domain', 'or', 'an', 'inhibitory', 'monoclonal', 'te', '##m', '##5', 'antibody', 'blocked', 'contact', 'inhibition', 'of', 'endothelial', 'cell', 'proliferation', 'resulting', 'in', 'multil', 'aye', '##red', 'islands', 'within', 'the', 'endothelial', 'monolayer', 'and', 'increased', 'vessel', 'density', 'during', 'capillary', 'formation', '.'
   Reduced by 10 words

Tokenizer	**Avg. tokens per example**
Original BERT	67.54
Expanded vocabulary	59.56
Reduced by	7.98

Table 1: Average sequence length on HoC training set

(c) Average token length on HoC training set The expanded vocabulary reduces average length by ∼8 tokens (more than 10% shorter), which can speed up training.

Question 4 (5 Credits): New parameters & initialization

By adding 5 000 new tokens, the embedding matrix grows from $|V_0| \times d$ to $(|V_0| + 5000) \times d$, where $d = 768$. Thus we introduce

$$5000 \times 768 \;=\; 3{,}840{,}000$$

new parameters. We initialized them with the same Gaussian scheme as BERT's original embeddings.

Question 5 (15 Credits): Downstream performance

We fine-tuned both the original and the expanded-vocabulary BERT on the HoC classification task for 20 epochs (batch size 8, lr=1e-5).

As we can see from 2, the original BERT tokenizer can outperform the expanded one

Analysis:

Model	Accuracy	Macro F1	Micro $F1_1$
Original BERT	0.793	0.792	0.793
Expanded vocabulary	0.690	0.627	0.690

Table 2: HoC test performance

- **High initialization noise for new embeddings.** Adding 5,000 new tokens injects 5,000 randomly initialized embedding vectors. At the start, these vectors behave like noise, slowing down convergence as the model must spend many steps to "calibrate" them.
- **Insufficient corpus size to support a larger vocabulary.** A modestly sized downstream training set may only contain rare biomedical terms a handful of times. Such sparsely observed tokens cannot learn stable representations, increasing the risk of overfitting rather than improving modeling.
- **Tokenizer–pretraining mismatch.** BERT's original embeddings were finely tuned on massive general-domain corpora (Wikipedia & BookCorpus). Inserting new tokens disrupts the pretraining distribution, especially when new and old subwords overlap, potentially breaking the established subword segmentation logic.
- **Model capacity and parameter bloat.** Expanding the embedding matrix by $5{,}000 \times 768 \approx 3.8$ million parameters without adjusting the overall optimizer settings, learning rate schedule, or regularization makes it harder for the network to reach a global optimum on limited data.
- **Lack of domain-adaptive pretraining.** To fully leverage new biomedical tokens, one typically needs to continue masked-language-model pretraining on large PubMed/PMC corpora. Without this domain-adaptive step, the added tokens remain "empty shells" with no strong contextual grounding.