**special\_tokens\_map.json**

Defines the main special tokens for the tokenizer:

* [CLS] (classification token)
* [MASK] (mask token)
* [PAD] (padding token)
* [SEP] (separator token)
* [UNK] (unknown token)[[1]](#fn1)

**tokenizer.json**

Describes the full tokenizer configuration and vocabulary:

* **Version:** 1.0
* **Truncation:** Right, max length 128, "LongestFirst" strategy
* **Padding:** Fixed to length 128, pad token [PAD] (ID 0)
* **Special Tokens:** [PAD] (0), [UNK] (100), [CLS] (101), [SEP] (102), [MASK] (103)
* **Normalizer:** BertNormalizer (lowercase, clean text, handle Chinese chars)
* **Pre-tokenizer:** BertPreTokenizer
* **Post-processor:** Adds [CLS] at start and [SEP] at end (for single and pair sequences)
* **Decoder:** WordPiece with prefix ##
* **Vocabulary:** Extensive, includes [unused0] to [unused992], standard punctuation, Latin, Cyrillic, Arabic, Indic, and other scripts, plus many English words and subwords[[2]](#fn2)

**tokenizer\_config.json**

Gives further tokenizer configuration:

* **Special tokens** and their properties (IDs 0, 100, 101, 102, 103)
* **Lowercasing:** true
* **Model max length:** 512
* **Tokenizer class:** DistilBertTokenizer
* **Other options:** No cleanup of tokenization spaces, supports Chinese character tokenization[[3]](#fn3)

**trainer\_state.json**

Tracks model training state and history:

* **Epoch:** 2.0 (out of 3 planned)
* **Global step:** 426 (out of 639)
* **Logging:** Every 10 steps, with loss, grad norm, learning rate, etc.
* **Eval loss:** 0.127 (after epoch 1), 0.155 (after epoch 2)
* **Batch size:** 16
* **Save steps:** 500
* **Total FLOPs:** 225,724,447,309,824
* **Callbacks:** TrainerControl flags for stopping, saving, logging, etc.[[4]](#fn4)

**Summary Table**

|  |  |  |
| --- | --- | --- |
| File | Status | Key Contents/Notes |
| special\_tokens\_map.json | Readable | Defines [CLS], [MASK], [PAD], [SEP], [UNK] tokens |
| tokenizer.json | Readable | Full tokenizer config, vocab, normalization, post-processing, special tokens |
| tokenizer\_config.json | Readable | Tokenizer settings, max length, lowercasing, special tokens, class |
| trainer\_state.json | Readable | Training progress, eval loss, step/loss history, batch size, callbacks |
| rng\_state.pth | Not readable | No content found (likely binary PyTorch RNG state) |
| [scheduler.pt](http://scheduler.pt) | Not readable | No content found (likely binary PyTorch scheduler state) |
| training\_args.bin | Not readable | No content found (likely binary with training arguments) |

**Conclusion:**  
All text-based configuration and training state files were successfully read and contain standard BERT/DistilBERT tokenizer and training information. The binary files could not be read, which is expected for PyTorch state and argument files[[5]](#fn5)[[6]](#fn6)[[7]](#fn7). If you need details from those, you would have to open them in a Python environment using the appropriate PyTorch or Hugging Face tools.