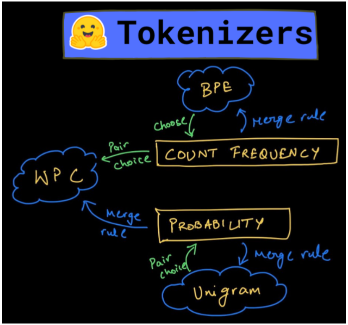
Transfer Learning

Data-driven Tokenization



Learning goals

- Understand the importance of tokenization for Transfer Learning
- Benefits of data driven tokenization over "generic" approaches

WORDPIECE

Voice Search for Japanese and Korean ‣ Schuster & Nakajima (2012)

- *Specific Problems:*
  - Asian languages have larger basic character inventories compared to Western languages
  - Concept of spaces between words does (partly) not exist
  - Many different pronounciations for each character

WORDPIECE

- *WordPieceModel:* Data-dependent + do not produce OOVs
  1. Initialize the the vocabulary with basic Unicode characters (22k for Japanese, 11k for Korean)
     ⚠ Spaces are indicated by an underscore attached before (of after) the respective basic unit or word (increases initial $|V|$ by up to factor 4)
  2. Build a language model using this vocabulary
  3. Merge word units that increase the likelihood on the training data the most, when added to the model
- Two possible stopping criteria:
  Vocabulary size *or* incremental increase of the likelihood

WORDPIECE

Use for neural machine translation ▸ Wu et al. (2016)

- *Adaptions:*
  - Application to Western languages leads to a lower number of basic units ($\sim$ 500)
  - Add space markers (underscores) *only* at the beginning of words
  - Final vocabulary sizes between 8k and 32k yield a good balance between accuracy and fast decoding speed (compared to around 200k from ▸ Schuster & Nakajima (2012))

Independent **vs.** *joint* **encodings for source & target language**

- Sennrich et al. (2016) report better results for joint BPE
- Wu et al. (2016) use shared WordPieceModel to guarantee identical segmentation in source & target language in order to facilitate copying rare entity names or numbers

SENTENCEPIECE ‣ KUDO ET AL. (2018B)

No need for Pre-Tokenization

- BPE & WordPiece require a sequence of words as inputs
  $\rightarrow$ Some sort of (whitespace) tokenization has to be performed before their application
- SentencePiece (as the name already reveals) doesn't need that
  $\rightarrow$ Can be applied to "raw" sentences
  $\rightarrow$ Consists of *Normalizer*, *Trainer*, *Encoder* & *Decoder*
  $\rightarrow$ Under the hood, two different algorithms are implemented
  - byte-pair encoding ‣ Sennrich et al. (2016)
  - unigram language model ‣ Kudo et al. (2018a)
- No language-specific pre-processing

$\Rightarrow$ Basically a nice, end-to-end usable system/pipeline

USAGE OF DIFFERENT TOKENIZERS

Disclaimer I:
You don't know these models yet, this is to give you an impression.

Disclaimer II:
BPE will be introduced in the next chapteron the Transformer.

- **WordPiece:**
  BERT, DistilBERT, ELECTRA,
- **SentencePiece:**
  ALBERT, XLNet, T5
- **BPE:**
  Transformer, GPT-2, RoBERTa

$\Rightarrow$ Additional Resource: ▸ Overview on huggingface