Natural Language Processing

Introduction to tokenisation

Marco Kuhlmann
Department of Computer and Information Science

What is tokenisation?

- **Tokenisation** is the task of breaking running text into smaller segments such as words or characters.
- Tokenisation simplifies natural language processing by reducing unstructured text to more useful units.
- Tokenisation is the first step in mapping text to a numerical representation that computers can process.

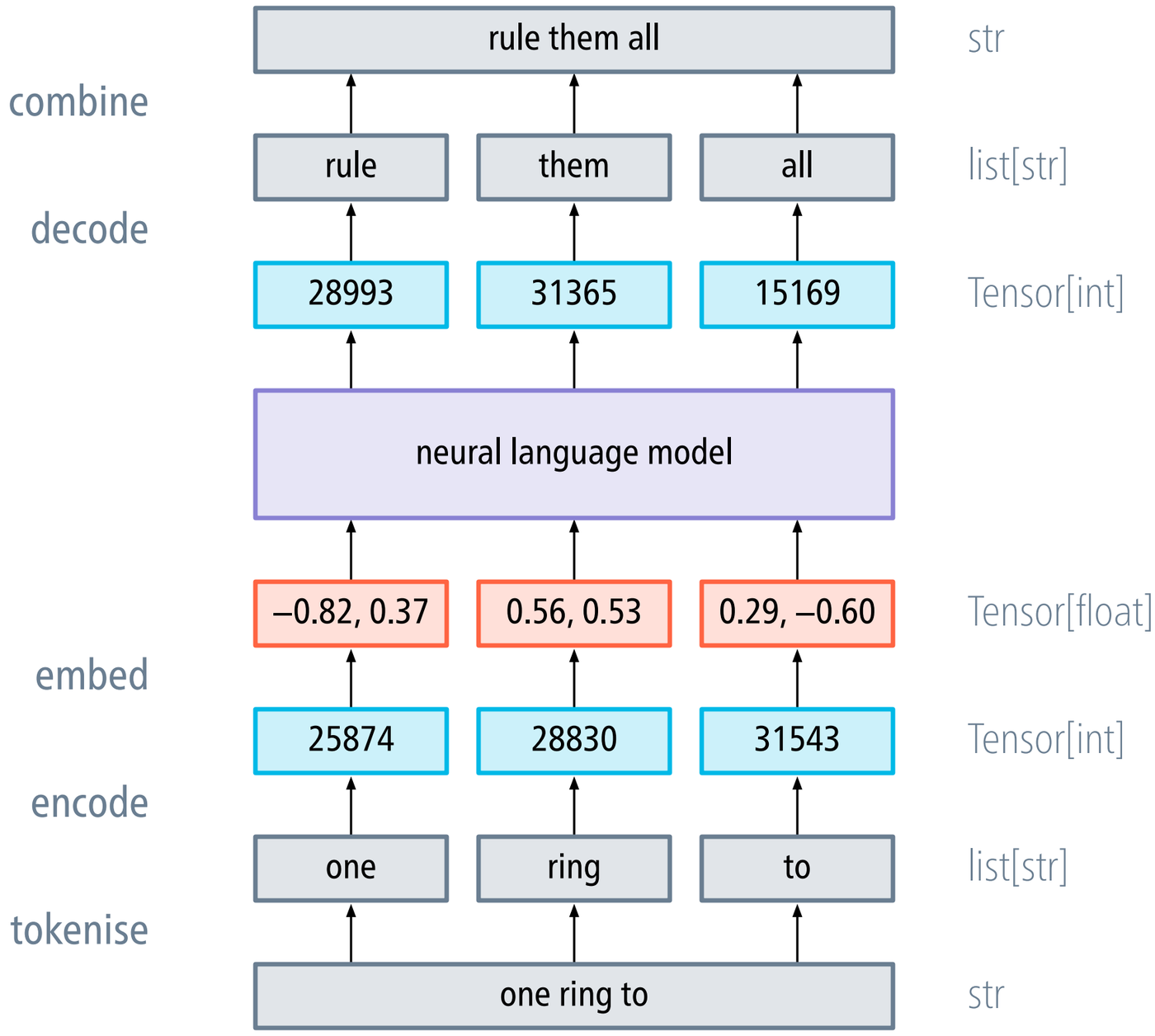
Words provide important signals

The gorgeously elaborate continuation of "The Lord of the Rings" trilogy is so huge that a column of words cannot adequately describe co-writer/director Peter Jackson's expanded vision of J.R.R. Tolkien's Middle-earth.

positive

… is a sour little movie at its core; an exploration of the emptiness that underlay the relentless gaiety of the 1920's, as if to stop would hasten the economic and global political turmoil that was to come.

negative


rule them all
str
combine
rule
them
all
list[str]
decode
28993
31365
15169
Tensor[int]
neural language model
–0.82, 0.37
0.56, 0.53
0.29, –0.60
Tensor[float]
embed
25874
28830
31543
Tensor[int]
encode
one
ring
to
list[str]
tokenise
one ring to
str

Whitespace tokenisation

```
# Tokenise text by splitting at whitespace
def tokenize(text: str) -> list[str]:
    return text.split()

# Create a vocabulary
vocab: set[str] = set(tokenize(text))
# {'cannot', 'huge', 'column', 'that', 'is', …}

# Create a string-to-ID mapping
stoid: dict[str, int] = {s: i for i, s in enumerate(vocab)}
# {'cannot': 0, 'huge': 1, 'column': 2, 'that': 3, 'is': 4, …}
```

Whitespace tokenisation

The gorgeously elaborate continuation of “The Lord of the Rings” trilogy is so huge that a column of words cannot adequately describe co-writer/director Peter Jackson’s expanded vision of J.R.R. Tolkien’s Middle-earth.

Regex-based tokenisation

The gorgeously elaborate continuation of “ The Lord of the Rings ” trilogy is so huge that a column of words cannot adequately describe co-writer / director Peter Jackson ’s expanded vision of J. R. R. Tolkien ’s Middle-earth .

```
re.findall(r"[A-Za-z]\.|\w+(?:-\w+)*|'\w+|[^\w\s]+", text)
```

single letters followed by a period	whole words, incl. hyphenated words	genitives ('s) and contractions ('ve)	punctuation, other non-word characters

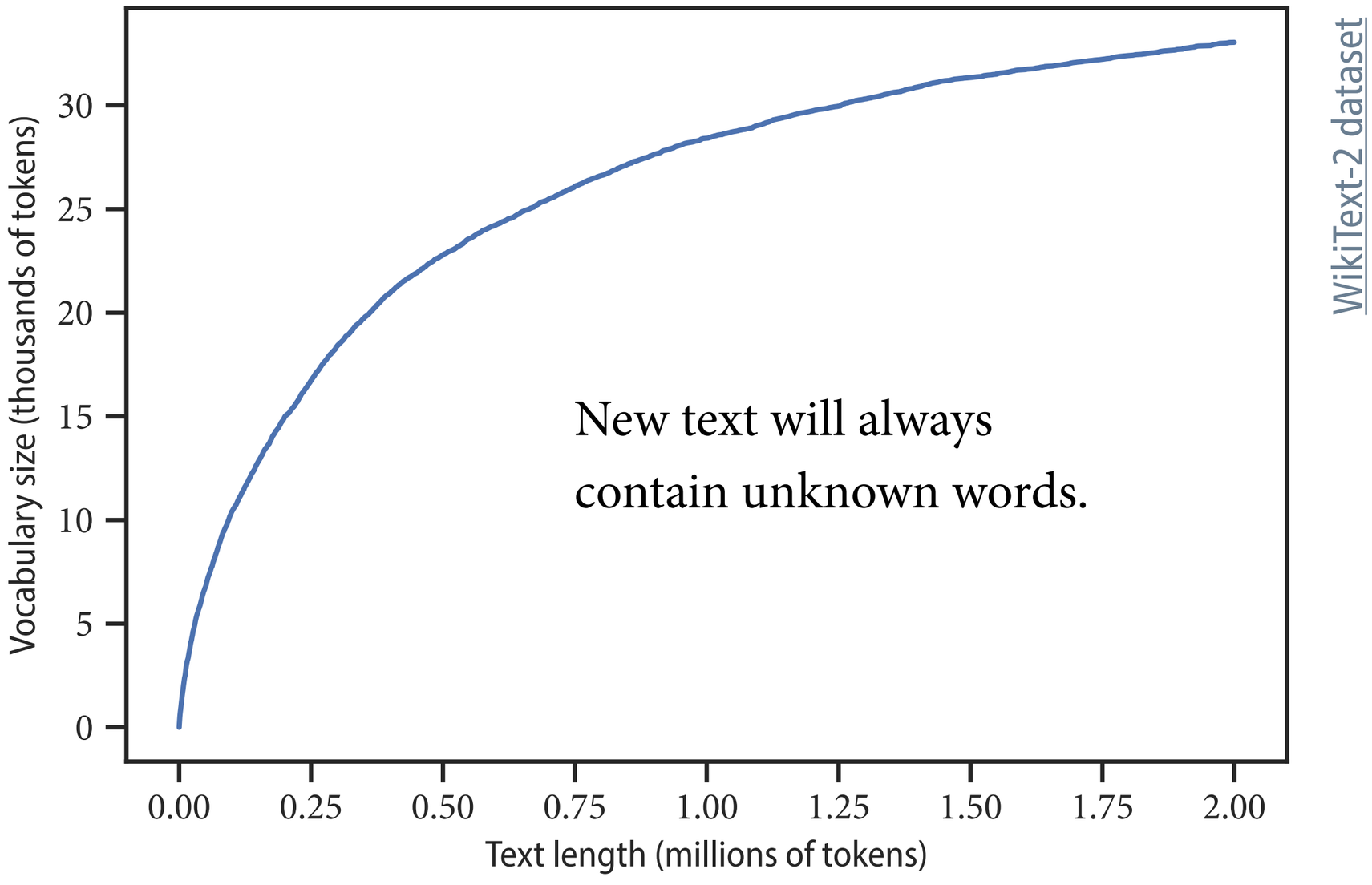
Text normalisation

- **Text normalisation** refers to the process of converting text into a more useful, standard form.
- Standard techniques include case normalisation, harmonisation of spelling variants, lemmatisation, and removing punctuation.

  Harmonisation: *color* → *colour*. Lemmatization: *runs*, *ran*, *running* → *run*

- Text normalisation was once a critical step in NLP tasks but is no longer as widely used today.

The challenge of unknown words – Heaps' law



Dealing with unknown words

- **Step 1:** Build the vocabulary as usual.

  often combined with a frequency threshold

By Ena Music – Own work, CC BY-SA 4.0, Link

- **Step 2:** Augment the vocabulary with a special token, such as [UNK], to represent unknown words.

- **Step 3:** When processing new text, replace any out-of-vocabulary (OOV) word with the special [UNK] token.

  The quokka is adorable. → *The [UNK] is adorable.* (Assuming *quokka* is OOV.)

But what is a word, anyway?

There are many languages that do not adhere to the same concept of a “word” as English and Swedish.

- **Chinese** is written without spaces between characters. Identifying word boundaries is challenging.

  姚明进入总决赛 – “Yao Ming reaches the finals.”

- **Inuktitut** allows entire sentences to be expressed as single words by combining multiple morphemes.

  tusaatsiarunnanngittualuujunga – “I cannot hear very well.”

Target representations for tokenisation

- **Option 1:** Tokenise into words

  But: concept of "word" not universal; unknown words

- **Option 2:** Tokenise into individual characters

  But: may be too small a unit for learning

- **Option 3:** Tokenise into subwords

  Intuition: words are composed of morphemes