1 Introduction to Text Mining

Text mining is the discipline that tries to extract knowledge (structured) from text (unstructured). It uses methods from Natural Language Processing but is not the same as NLP also does translation, speech recognition and semantic parsing. Another discipline, information retrieval, is the first step in text mining processes as to filter the documents to actually mine. But text mining generally does not use approaches like ranking models or user modeling and evaluation.

1.1 Challenges

Some challenges that text mining projects generally face are errors in the text as it might be generated through OCR or ASR, the text might not be labelled or might miss punctuation and capitalisation, different languages might be used in the text and the terms in the text might have synonyms or could be polysyms (multiple meanings and uses for the same word).

- **Unstructured**
  Missing capitalisation, punctuation or unlabelled plain text that has no metadata attached to it whatsoever.
- **Noise**
  The **source** text can contain non-content information (i.e. HTML), encoding errors, page numbers or other noise. **Content** might contain OCR and ASR artifacts, spelling errors, typos or (lost) formatting. **Labelling** might be incomplete or contain wrong labels.
- **Infinity**
  Every new document is likely to introduce new terms, the first moreso (at a very rapid rate) than latter documents (at a slower rate):

  Definition 1.1 (Heaps' Law)**.** As more instance text is gathered, there will be diminishing returns in terms of discovery of the full vocabulary from which the distinct terms are drawn.
- **Ambiguity**
  Phrases might have a different meaning in a different context. For example, 2 stars is poor for a hotel, but excellent in a michelin rating.

1.2 Tasks

There are three main types of text mining tasks: text classification (or clustering), sequence labelling or text-to-text generation.

1.2.1 Classification

Classification (or clustering) assigns labels to a document. The document can be many things, from a sentence or a tweet to entire articles or maybe even a book. The labels can also be many things, such as the topic of the document, relevance, the author of the document, sentiment analysis, et cetera.

In classification, the order of words is not as relevant, and traditionally represent the text as a bag of words. Each word in the collection becomes a machine learning feature. This results in high-dimensional sparse vectors (i.e. dimension is the distinct words, and the values are counts).

Definition 1.2 (Bag of words approach)**.** Each word in a document is counted without punctuation, capitalization or order. A vector with every word in the corpus as the dimension and the respective count in the document as values represents the text. For example: The brown fox $\rightarrow$ [the, brown, some, fox] $\rightarrow [1, 1, 0, 1]$.

Only very few words are very frequent in documents. There is an strong inverse relation between the word frequency and the word rank, also called the long tail distribution. Words in the tail might be noise or errors and we want to get rid of them.

Definition 1.3 (Zipf's law)**.** When a list of measured words is sorted in decreasing order, the value of the nth entry is often approximately inversely proportional to n.

An alternative representation of words that is dense and has a lower dimension is word embeddings. Here words are represented in the vector space close to relevant words.

1.2.2 Sequence labelling

Sequence labelling identifies named entities in a text (such as persons, places, actions). Here the order matters, punctuation and capitalization is important.

1.2.3 Text-to-text generation

Text is given as an input, and resulting text is generated as an output. Use cases are things like summarization, translation.

1.3 Transformers

Transformers take an input text and generate an output text, they consist of two parts, an encoder that processes the input text and a decoder that generates the output text. Initially they were designed for translation tasks and were used in Google Translate.

BERT is a transformer model that only gives the encoder, providing word embeddings for input text. GPT is a decoder-only transformer, where the input is a text (or a prompt) and the output is text.

- **Encoder** (BERT)
  Classification tasks, sequence labelling.
- **Decoder** (GPT)
  Text generation (i.e. autocomplete).
- **Encoder-decoder** (T5)
  Summarization and translation.

1.4 Paradigms

- **Supervised Learning**
  Train feature-based models (i.e. bag of words in SVM). Lightweight and explainable, important to understand the model.
- **Transfer Learning**
  Use a pre-trained model and fine-tune it for a task. Best option when there is sufficient labelled data.
- **In-context Learning**
  Prompt a large language model with instructions and examples. Can be used without labelled data or undefined outcomes.