Computational Text Analysis

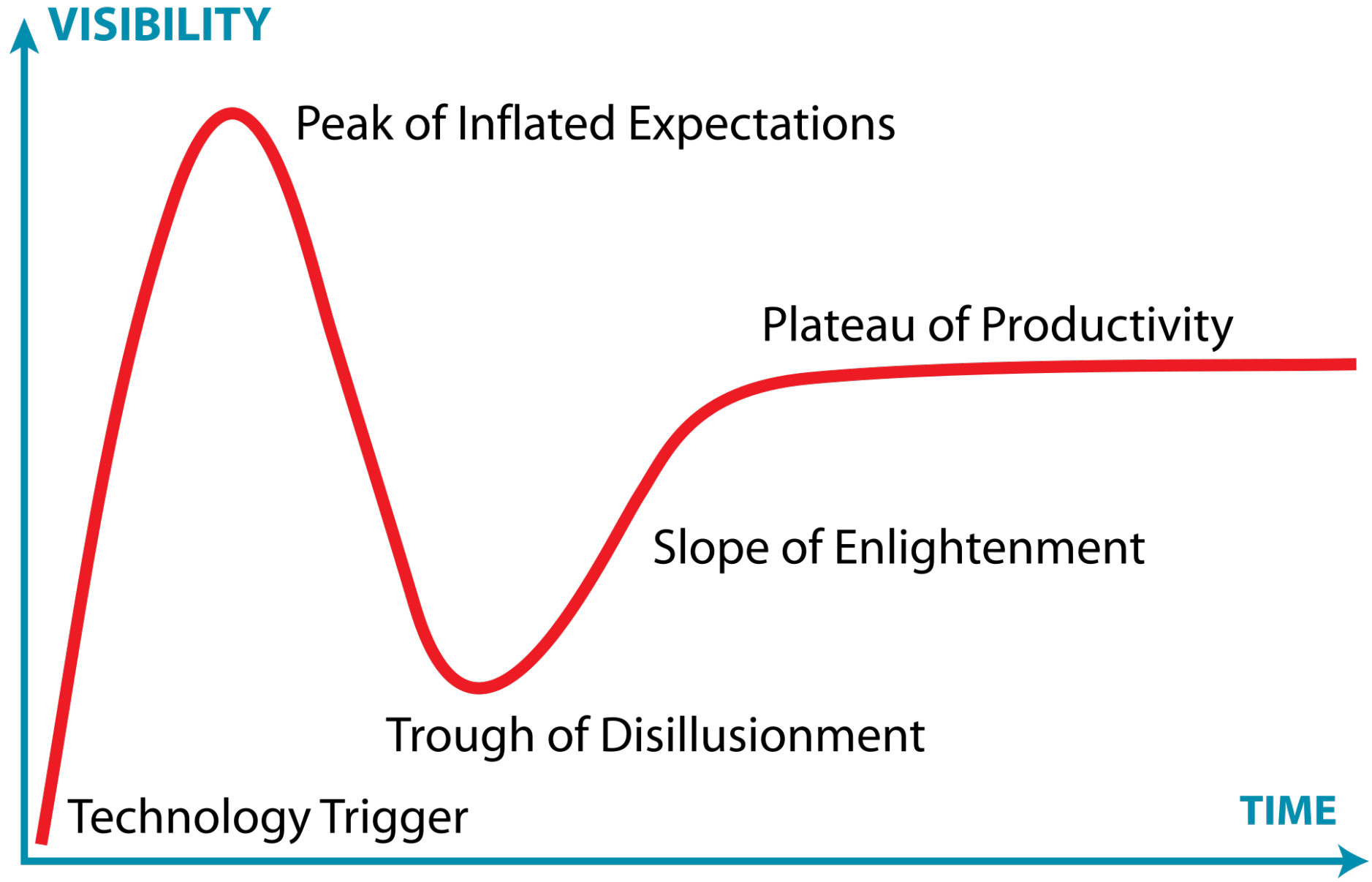
Summer Institute in Computational Social Science
SICSS

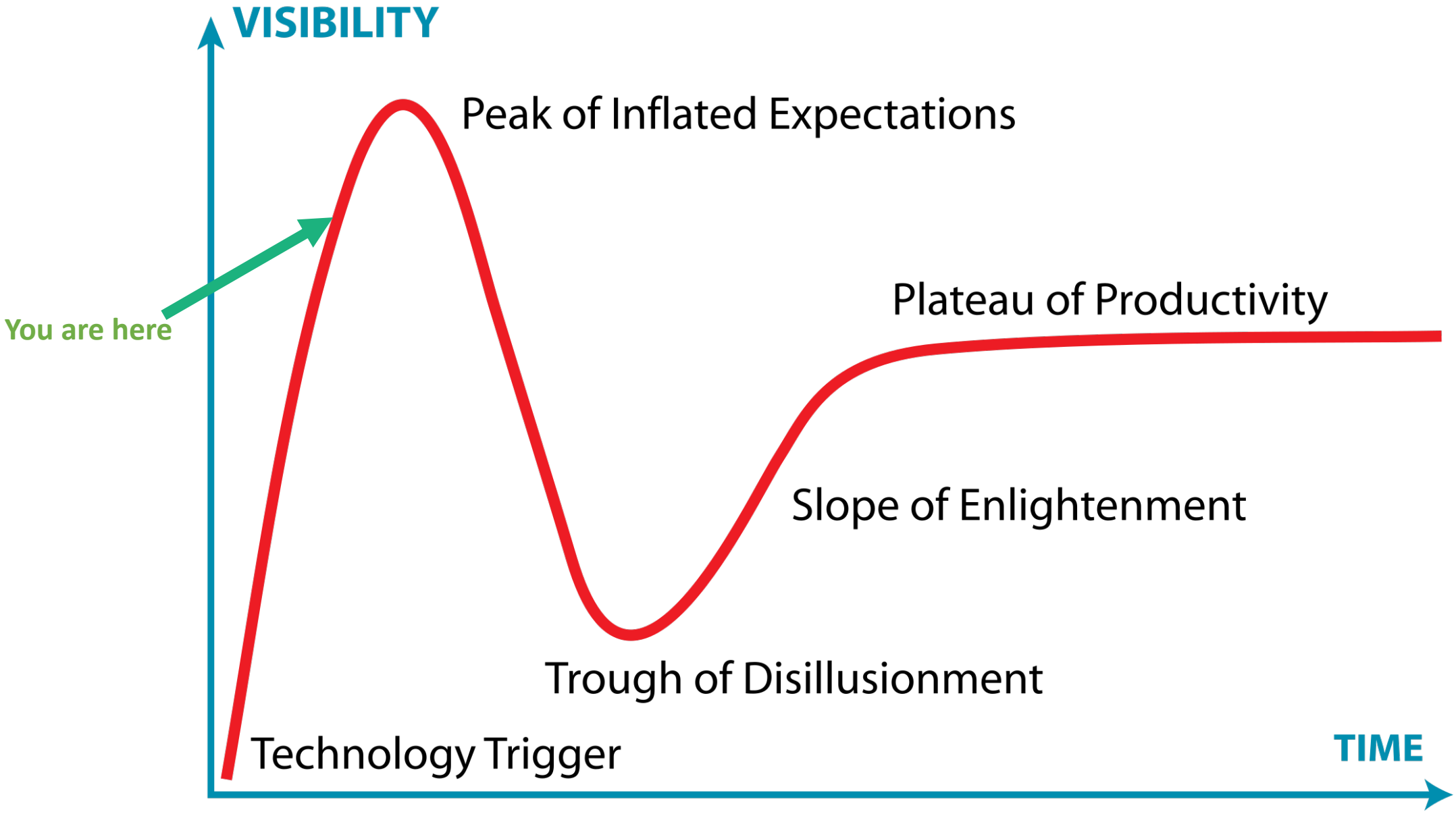
June 11th, 2025, Norrköping, Sverige

Hendrik Erz
Institute for Analytical Sociology (IAS)
<hendrik.erz@liu.se>

Agenda

- Lecture
  - Introduction to text analysis
  - What is text, exactly?
  - An abridged history of text analysis
  - Prominent text analysis methods
- Workshop
  - Working with text corpora
  - Latent Dirichlet Allocation
  - Word Embeddings
  - BERT-Models


VISIBILITY
Peak of Inflated Expectations
Plateau of Productivity
Slope of Enlightenment
Trough of Disillusionment
Technology Trigger
TIME


VISIBILITY
Peak of Inflated Expectations
You are here
Plateau of Productivity
Slope of Enlightenment
Trough of Disillusionment
Technology Trigger
TIME

What is text, exactly?

It’s not just data

What is text, exactly?

Source: UPenn Museum of Archeology

What is text, exactly?

- Text has been invented long after speech
  - It did not yet exist for the Greeks when the Odyssey or Iliad were told!
- It was born out of necessity to keep a record of things for the future
- Some (e.g., Graeber, 2011) even argue that writing was invented to record debts
- Today, text is everywhere, and with speech transcription tools, we can transform any spoken language into text

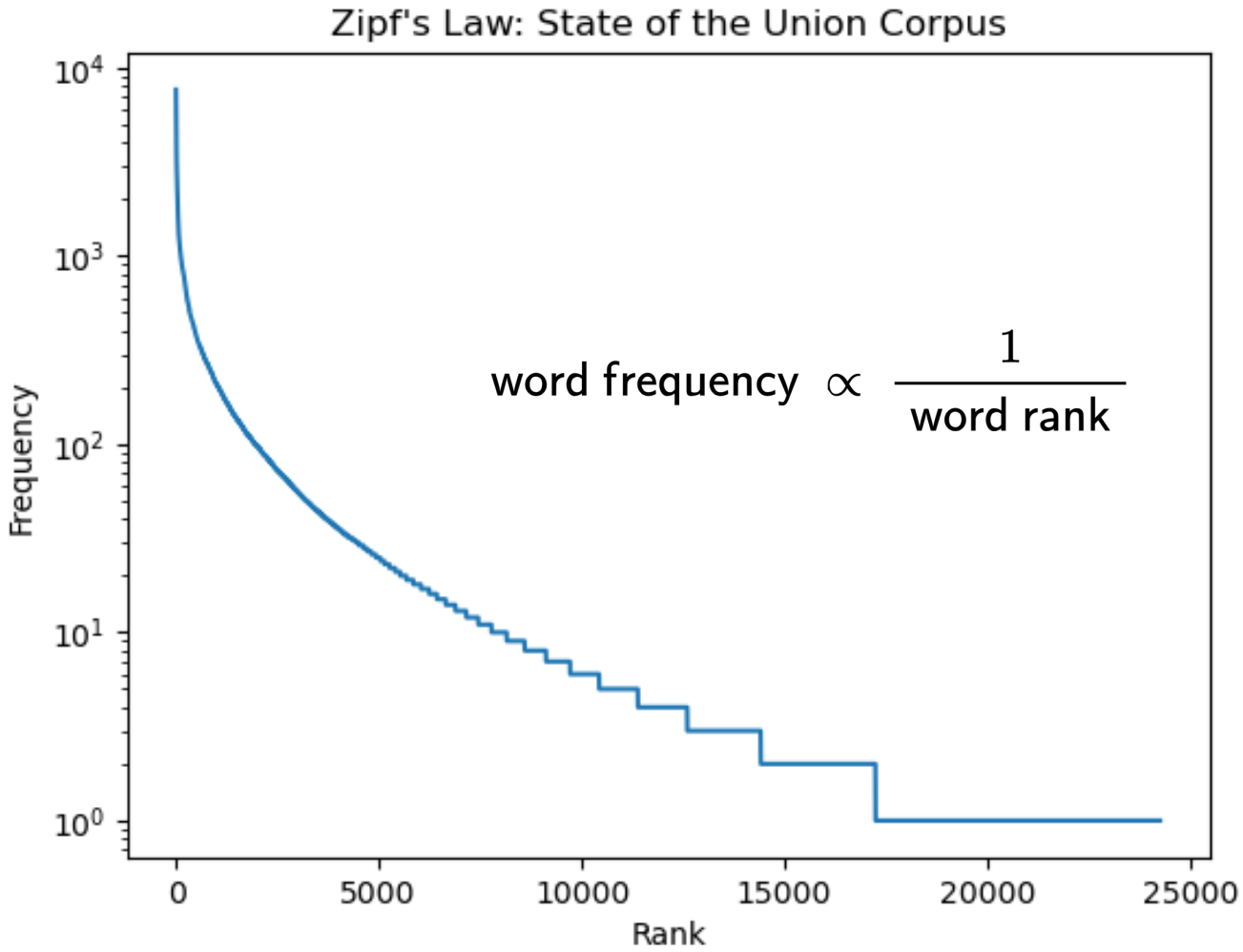
The Linguistic Perspective

- According to Eisenstein (2018), Text is …
  - … fundamentally discrete
  - … constantly changing
  - … compositional

The Linguistic Perspective

- According to Eisenstein (2018), Text is …
  - … fundamentally discrete
    - → There are no “50 shades of” some word
  - … constantly changing
    - → New words are constantly being invented, and existing ones redefined
  - … compositional
    - → One can re-arrange a set of words to form completely different sentences
- In short: Text is utterly high-dimensional (Shannon 1948)

The Linguistic Perspective



The Social Scientific Perspective

- For the social sciences, text is …
  - … a written representation of human (read: natural) language
  - … used to convey information, intentions, feelings, dreams, misinformation, etc. to other people
  - … a record of things
  - … created purposefully or incidentally, public or private
- For our purposes, think of text as a **reservoir of variables** that can be extracted using text analysis methods

The Social Scientific Perspective

"Yesterday, Lilian was feeling unwell and was annoyed when her mother dragged her to the doctor's office."

- Information contained in this sentence:
  - Temporality
  - Spatiality
  - Agency
  - Behavior
  - Sentiment
  - Gender
  - Relationships/Actors
  - …

What is Text?

- As text is an incredibly complex form of data, text analysis sits at the nexus between several research disciplines:
  - **Linguistics** help us understand the **structure** of text
  - **Information theory** helps us **quantify** text
  - **Computational Linguistics** and **Computer Sciences** provide **methods** to work with text
  - **Computational Social Science** uses these methods to answer social scientific research questions

Linguistic Dimensions of Text

Linguistics defines five dimensions of text:

Phonemes	Phonetics is the study of the sounds that make up our speech.
Morphemes	Morphology is the study of the smallest meaningful units of text (similar, but not equal, to syllables).
Syntax	Words are ordered in syntactic relationships (subject, verb, object)
Semantics	Semantics deals with the meaning of words (river bank vs. financial bank), sentences (e.g., ambiguity), and larger units of discourse.
Pragmatics	Pragmatics is the study of the meaning of the context in which speech or text is being produced.

Linguistic Dimensions of Text

Social Scientists can use most dimensions for analysis:

Phonemes	are usually not relevant to text analysis.
Morphemes	are the fundamental unit of analysis (UoA) for most methods.
Syntax	can be studied with part-of-speech (POS) taggers.
Semantics	are the most commonly analyzed part of text.
Pragmatics	can not (yet) be analyzed computationally.

Linguistic Dimensions of Text

Social Scientists can use most dimensions for analysis:

Phonemes	are usually not relevant to text analysis.
Morphemes	are the fundamental unit of analysis (UoA) for most methods.
Syntax	can be studied with part-of-speech (POS) taggers.
Semantics	are the most commonly analyzed part of text.
Pragmatics	can not (yet) be analyzed computationally.

"World Knowledge Problem"

Despite its impressive output, generative AI doesn't have a coherent understanding of the world

Researchers show that even the best-performing large language models don't form a true model of the world and its rules, and can thus fail unexpectedly on similar tasks.

Adam Zewe | MIT News
November 5, 2024

methods.

Syntax ... can be studied with part-of-speech (POS) taggers.

Semantics are the most commonly analyzed part of text.

Pragmatics can not (yet) be analyzed computationally.

"World Knowledge Problem"

Despite its impressive output, generative AI doesn't have a coherent understanding of the world

Researchers show that even the best-performing large language models don't form a true model of the world and its rules, and can thus fail unexpectedly on similar tasks.

Adam Zewe | MIT News
November 5, 2024

Pragmatics

"World Knowledge

The Illusion of Thinking:
Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity

Parshin Shojaee*† Iman Mirzadeh* Keivan Alizadeh
Maxwell Horton Samy Bengio Mehrdad Farajtabar

Apple

“Text as Data”

- One of the seminal books for CTA (Grimmer et al., 2022)
  - This is a misnomer!
- Text is, in fact, data — but only for your computer
  - Information Theory mathematically proves that storing, transmitting, and compressing text is incredibly efficient (see Shannon, 1948), but none of this yields useful information for social scientific research
- A better metaphor: “Mapping Text” (Stoltz and Taylor, 2024)

"Text

- One of
  - This
- Text is,
  - Infor
    com
    yield
- A bette

, and
e of this

"Low-Resource" Text Classification: A Parameter-Free Classification Method with Compressors

Zhiying Jiang[1,2], **Matthew Y.R. Yang**[1], **Mikhail Tsirlin**[1], **Raphael Tang**[1], **Yiqin Dai**[2] and **Jimmy Lin**[1]

[1] University of Waterloo [2] AFAIK

{zhiying.jiang, m259yang, mtsirlin, r33tang}@uwaterloo.ca
quinn@afaik.io jimmylin@uwaterloo.ca

Abstract

Deep neural networks (DNNs) are often used for text classification due to their high accuracy. However, DNNs can be computationally intensive, requiring millions of parameters and large amounts of labeled data, which can make them expensive to use, to optimize, and to transfer to out-of-distribution (OOD) cases in practice. In this paper, we propose a non-parametric alternative to DNNs that's easy, lightweight, and universal in text classification: a combination of a simple compressor like *gzip* with a k-nearest-neighbor classifier. Without any training parameters, our method achieves results that are competitive with non-pretrained

(2018) further show that even word-embedding-based methods can achieve results comparable to convolutional neural networks (CNNs) and recurrent neural networks (RNNs).

Among all the endeavors for a lighter alternative to DNNs, one stream of work focuses on using compressors for text classification. There have been several studies in this field (Teahan and Harper, 2003; Frank et al., 2000), most of them based on the intuition that the minimum cross entropy between a document and a language model of a class built by a compressor indicates the class of the document. However, previous works fall short of matching the quality of neural networks.

A Note on Non-Western Languages

- Most NLP-methods are biased in favor of western languages, often just English.
- They assume that text consists of characters that can be combined to form words, separated by whitespace.
- Some methods are generalizable and only require, say, a specific tokenizer to work with non-English scripts.
- Even then, many non-western languages suffer from resource-sparsity
- These problems continue into the era of Large Language Models (LLMs), with many being primarily able to generate English, Chinese, or similarly resource-heavy languages.

A History of Text Analysis

From the Federalist Papers to the Congressional Record

A History of Text Analysis

- The social scientific approach to quantitative text analysis has lived through three distinct phases:
  - The pre-computing era (ca. 1930–2000)
  - Topic Modeling (ca. 2000–2019)
  - Large Language Models (since 2019)

From Counting Words to Grammar

- Until the 2000s, quantitative text analysis was restricted to simple measures
- Most text analysis was qualitative
  - 1938: Harold Lasswell manually codes newspaper articles to understand propaganda
  - 1963: Mosteller and Wallace de-anonymize the Federalist Papers by counting function words
  - 1989: Roberto Franzosi describes Quantitative Narrative Analysis
- In the 1990s, quantitative text analysis slowly changed with the proliferation of personal computers and new measures, such as TF-IDF

From Counting Words to Grammar

While we wait for artificial intelligence to provide a solution to content analysis that would not require human intervention, I provide a new approach to text coding based on linguistics, in particular, on semantic text grammars. (Franzosi, 1989, p. 264)

The Topic Modeling Phase

- In 2003, David Blei developed Latent Dirichlet Allocation (Blei et al., 2003)
- It allowed researchers to quickly sort documents into topics (admixture process, see Grimmer et al. 2021)
- Since it had well-defined assumptions over the data-generating process, adapting it to various theoretical and methodological approaches was straight-forward
- Between 2003 and 2020, dozens (hundreds?) of papers have used topic modeling for various purposes

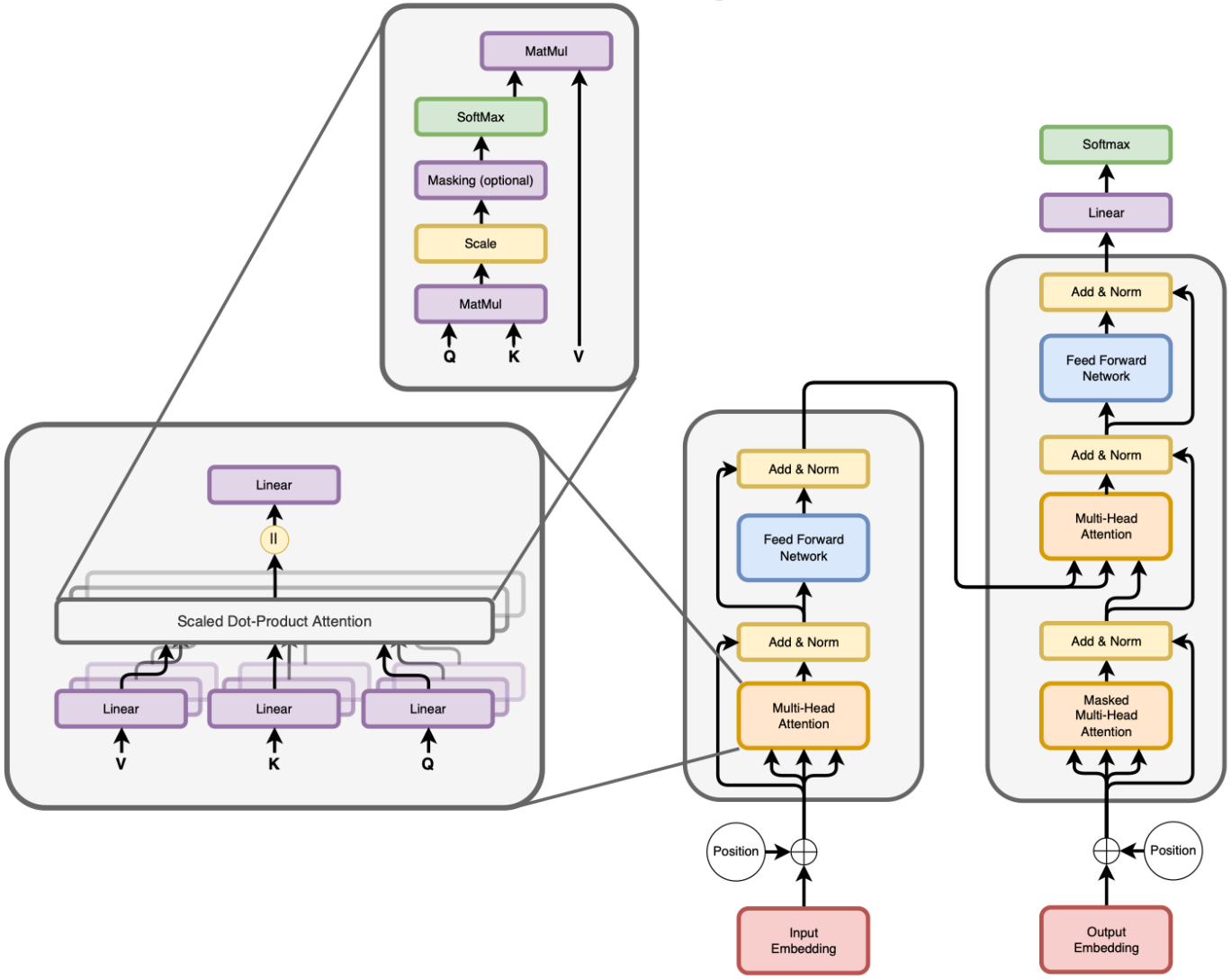
The Era of Artificial Intelligence

- In 2013, a research team of Google invented the word2vec algorithm
- It was not widely used in social sciences due to the prevalence of topic models and the computational difficulty of implementing word2vec
- First papers using word2vec began to appear at around 2019
- At that point, however, another type of text analysis took off: large language models (LLMs)

The Era of Artificial Intelligence

- In 2017, Vaswani et al. developed the **transformer** model
- It consisted of an encoder-decoder structure and was intended for translating text
- Soon, researchers realized that one could also use **only the encoder-stage** of a transformer for text classification: **BERT** (Bidirectional Encoder Representations from Transformers)
- Not long thereafter, researchers also began to use **only the decoder-stage** of a transformer for text generation: **GPT** (Generative Pretrained Transformers)

The Era of Artificial Intelligence



The Transformer Architecture
(source: https://www.hendrik-erz.de/post/the-transformer-architecture-a-visual-guide-pdf-download)

Introduction to Text Analysis

Discovery, Measurement, Inference

Research Design

Classical Statistics

1. Develop a research question
2. Get data that helps you answer this question
3. Create descriptive analyses to gain an overview over the data
4. Choose the appropriate model
5. Transform the data as necessary
6. Run inference

Text Analysis

1. Develop a research question
2. Define the concept(s) that correspond to this research question
3. Search for a text corpus which allows you to measure this concept
4. Create descriptive analyses to gain an overview over the data
5. Identify how and where the concept(s) manifest(s) in this corpus
6. Choose an appropriate model to extract this information
7. Add statistical data to that (see left side)
8. Transform the text data as necessary
9. Run inference

Research Design

Classical Statistics

1. Develop a research question
2. Get data that helps you answer this question
3. Create descriptive analyses to gain an overview over the data
4. Choose the appropriate model
5. Transform the data as necessary
6. Run inference

Text Analysis

1. Develop a research question
2. **Define the concept(s) that correspond to this research question**
3. **Search for a text corpus which allows you to measure this concept**
4. **Create descriptive analyses to gain an overview over the data**
5. **Identify how and where the concept(s) manifest(s) in this corpus**
6. **Choose an appropriate model to extract this information**
7. Add statistical data to that (see left side)
8. Transform the text data as necessary
9. Run inference

Research Design

Four Steps Towards Text Analysis

1. Discovery
2. Concept Definition
3. Measurement
4. Inference

Discovery

- Before diving into any analysis, you need to understand your corpus
- How is the text being generated?
  - Purposefully or incidental?
  - Public or private?
- Who wrote the text? Who is “speaking”? Are they speaking in equal amounts?
- What are they talking about? What topics, concepts, “things”?
- → Understand what your corpus is about by **deep reading**

Concept Definition

- What concept(s) are you interested in?
  - Gender
  - Topics
  - Tone
  - Discourse
- How do you (a human) find these concepts?
- How can you automate finding them (i.e., tell a computer to find them)?
- What are potential edge cases? Can you avoid them? Or are they inconsequential?

Concept Definition

- Example: Sentiment Analysis
- Many assume sentiment is simply a positive/negative number associated with words
- However, it's not!
- Having a sentiment usually involves identifying a target (towards which this sentiment is directed), a dimension (not just pos/neg, but various emotions, or stances), and accounting for shifts

Measurement

- Find a text analytical model that can appropriately map your corpus onto these concepts
  - Pay close attention to the implicit assumptions built into the model:
    - Biases, training data, linguistic assumptions, theoretical shortcuts, training task targets
- Build verification into these models (Stuhler, Ollion, and Ton, 2025)
  - Metrics such as loss, perplexity, accuracy, F1 scores, coherence, exclusivity, …
  - Held-out test data
  - Cross-Validation
  - Human inspection

Inference

- Inference usually maps onto regular statistical inference tactics
- At this point, you should have transformed your corpus in such a way that you end up with regular, continuous, categorical, or binary variables that you can use like any other

Introduction to Text Analysis Methods

TF-IDF, LDA, word2vec, GloVe, MfG, BERT, GPT, etc. …

Text Preprocessing

- Out “in the wild,” text is incredibly dirty
- Text preprocessing describes the necessary steps before even using a method to clean the text
- Some preliminaries:
  - Optical Character Recognition (OCR)
  - Character Encoding
  - Splitting/Joining the corpus
  - Removing control characters

Text Preprocessing

- Depending on research question and method used, one may…
  - Remove Whitespace
  - Remove Numbers
  - Remove Punctuation
  - Remove Stopwords
  - Drop the most frequent and least frequent words
  - Convert text to lowercase
  - Stem or lemmatize

Text Representation

- As mentioned initially, text is not (initially) data for social scientific purposes
- One must find a suitable *representation* for text
  - (Weighted) Frequency Counts (incl. TF-IDF)
  - Expert-assigned values (e.g., LIWC, VADER)
  - Document-Term-Matrix (DTM)
  - Word Embeddings (static or dynamic, BPE)
  - BOW or CBOW

TF-IDF

- td-idf is a measure that aims to improve word frequency by weighting words according to their frequency and scale them appropriately
- It is a direct response to “Zipf’s Law” and attempts to flatten the frequency curve
- tf-idf scores are often used as weights to pass text data into regressions or to decide which words resemble stopwords in a given, specific corpus

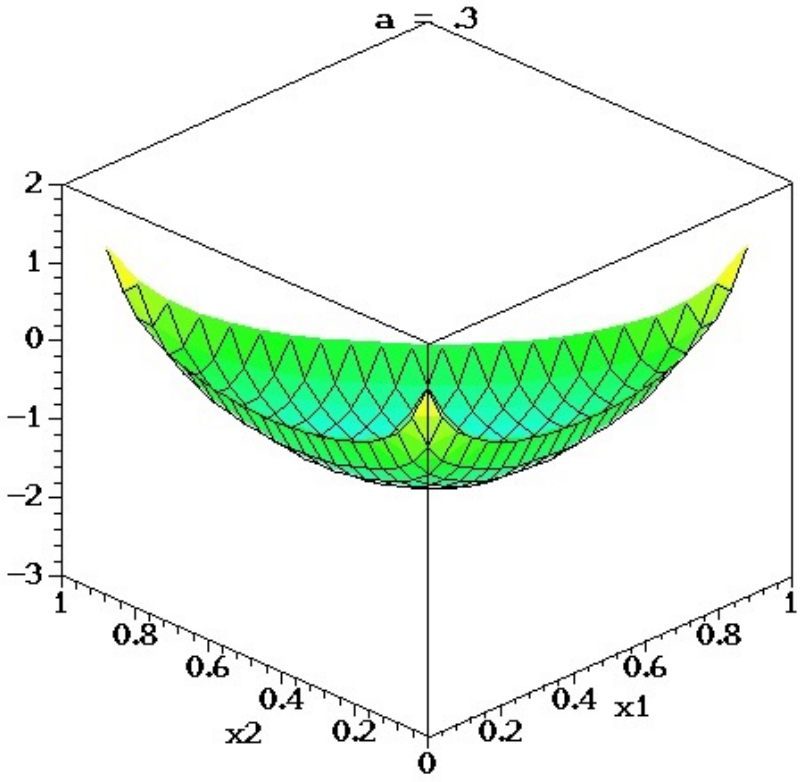
$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \times \text{idf}(t, D)$$

$$\text{tf}(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

$$\text{idf}(t, D) = \log \frac{N}{1 + D}$$

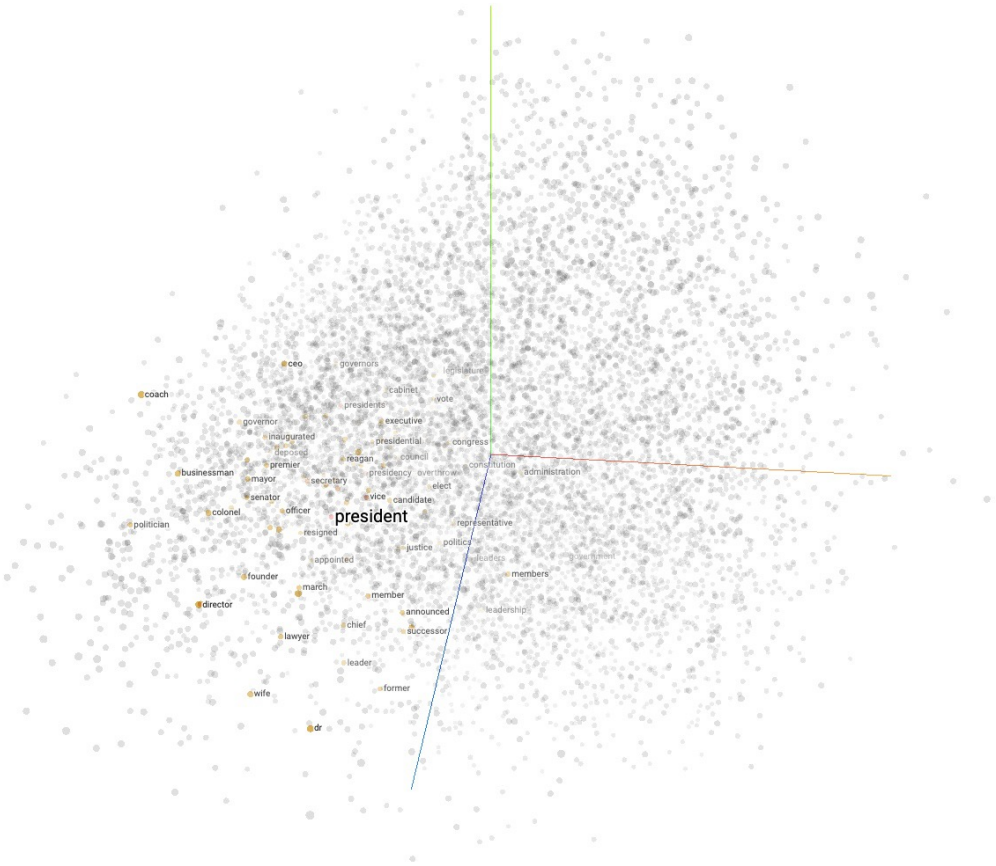
Topic Models (LDA)

- Topic models are a measure to group documents together which use similar mixtures of words
- Most commonly used: Latent Dirichlet Allocation (Blei et al., 2003)
- Extensions:
  - Structural Topic Models (Roberts, 2014)
  - Seeded Topic Models
- Assumes topics to be Dirichlet-distributed across a corpus, based on a multinomial word-distribution
- Requires hyperparameters: Alpha, Beta, K
- Metrics to estimate quality, e.g., cohesiveness, or exclusivity



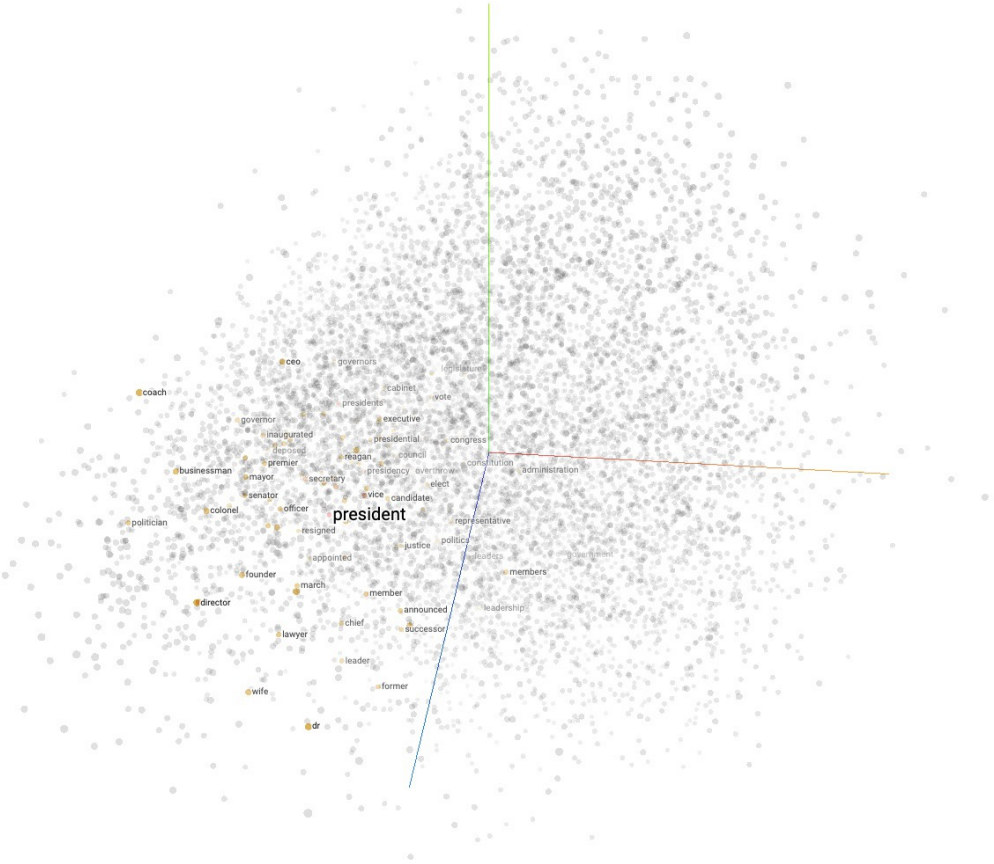
Word embeddings (word2vec)

- Word embeddings assign each individual token in a vocabulary to an n-dimensional numerical vector
- Most common static method: word2vec (Mikolov et al., 2013)
- Based on John Rupert Firth (1953): “You shall know a word by the company it keeps”
- Can also be performed with SVD (see GloVe, Pennington et al., 2014)



Word embeddings (word2vec)

- One important parameter to keep in mind: window size (Stoltz et al., 2021)
  - Small window sizes (~ 5 words) capture more synonymous words (i.e., “problem” will be very close to “issue”)
  - Larger window sizes (> 10 words) capture more topically smimilar words (i.e., “problem” will be very close to “solution”)



POS-tagging & Dependency Parsing

- Part-of-Speech (POS) tagging is a method that can extract grammatical functions from text (e.g., subjects, verbs, or objects, see Franzosi, 1989)
- Dependency parsing more generally describes tools that extract the grammatical structure of words
- Also possible: Named Entity Recognition (NER)
- Relies on LSTM-networks (Hochreiter & Schmidthuber, 1997; Qi et al., 2020)

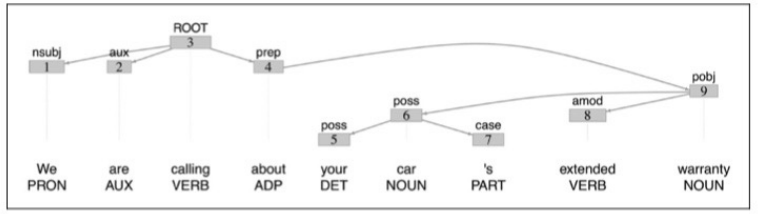


Figure 1. A sentence and corresponding dependency tree.
Note: Syntactic relationships between tokens are directed as indicated by the arrows pointing from the head to the dependent token. The depiction of dependency trees in this paper follows the convention of placing the tag of a dependency relation above the dependent node. For instance, "We" is the nominal subject (nsubj) of "calling.".

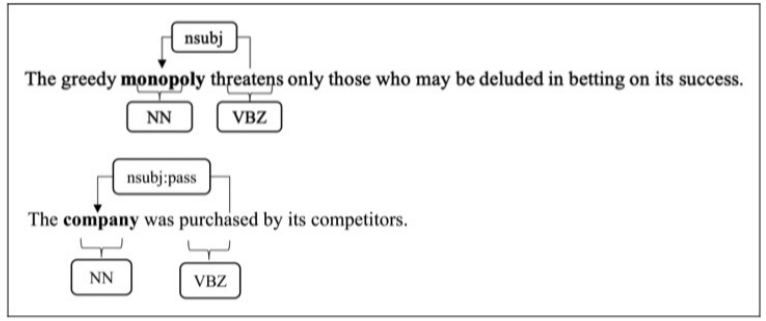
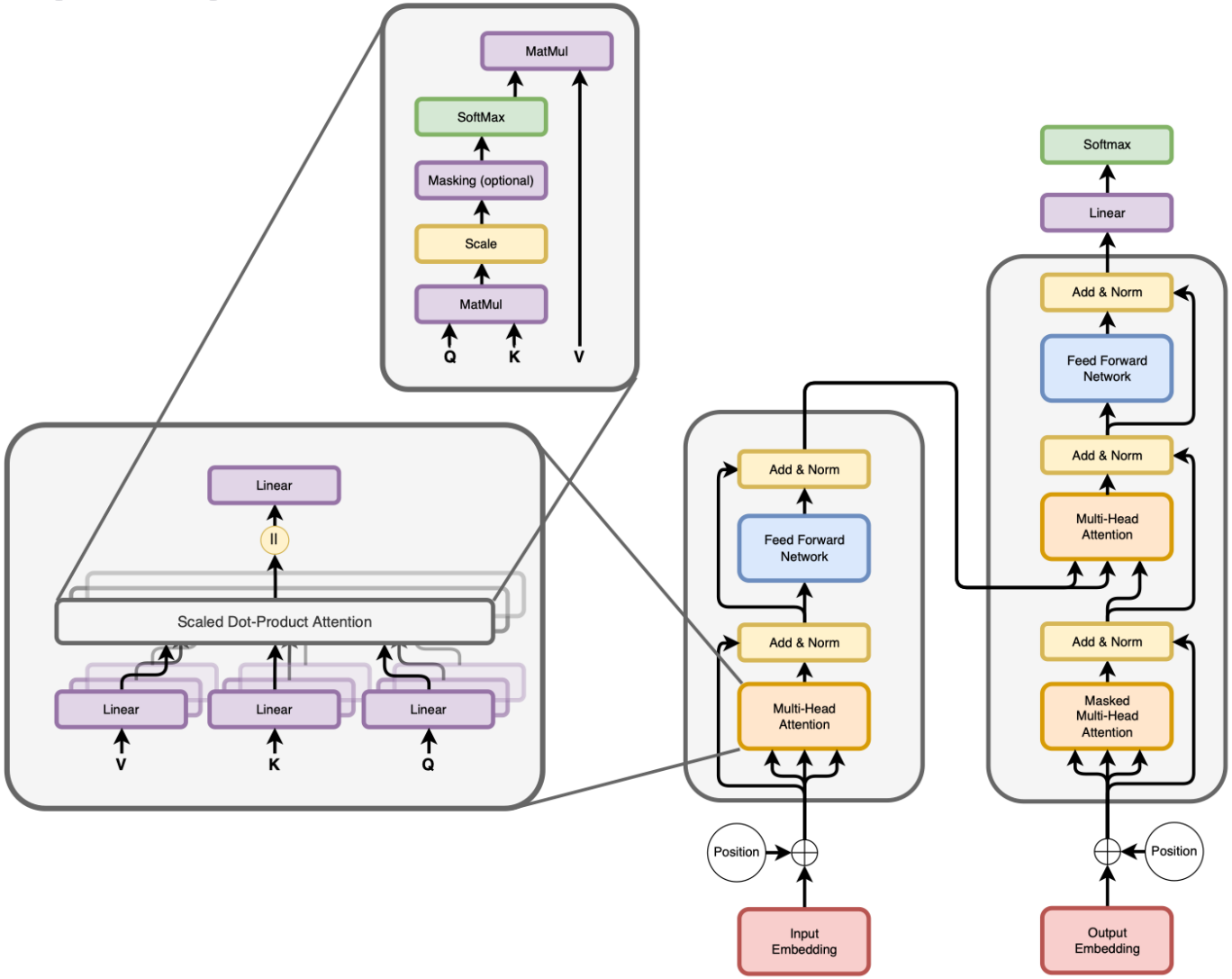


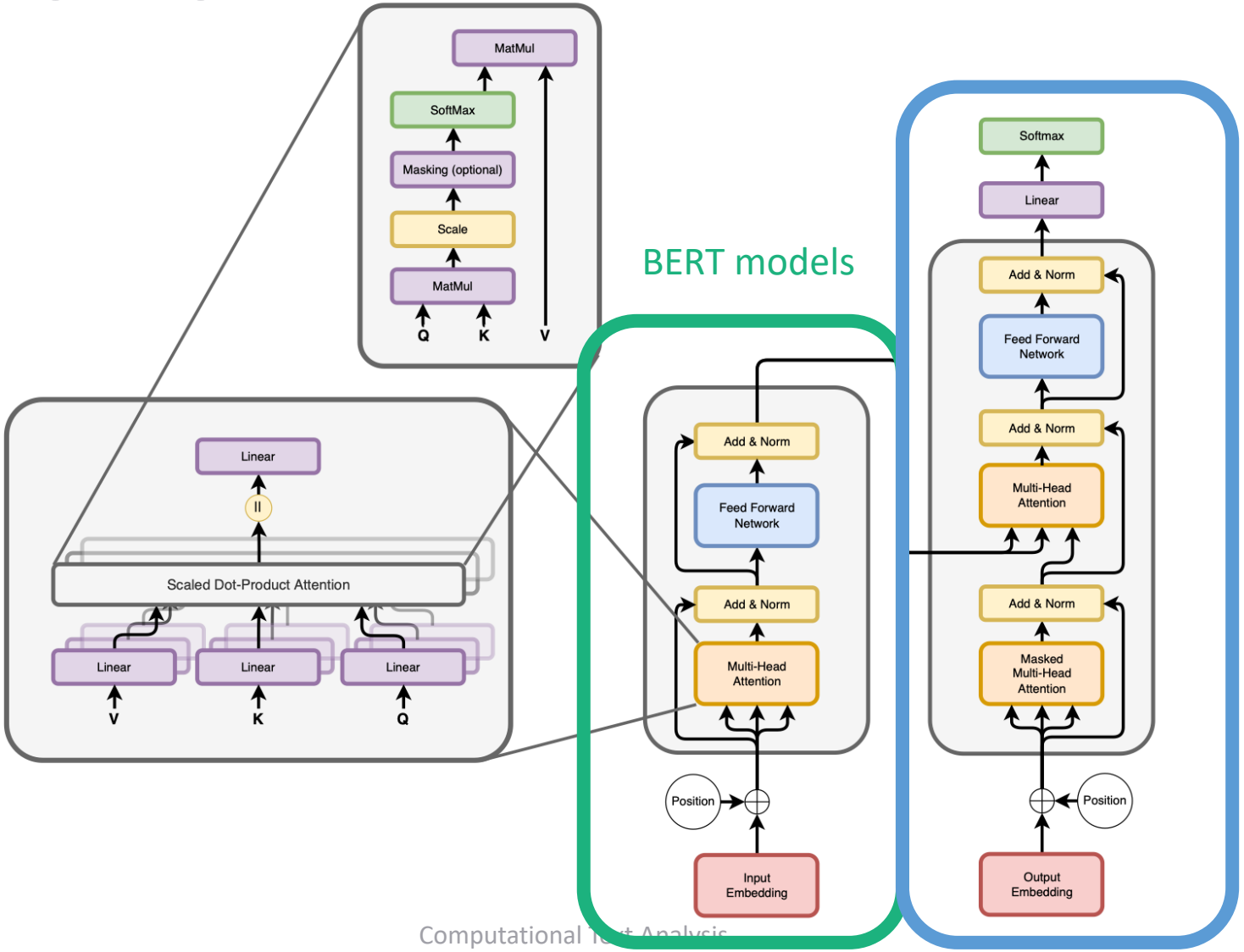
Figure 2. Example of key dependency relations.

Large Language Models



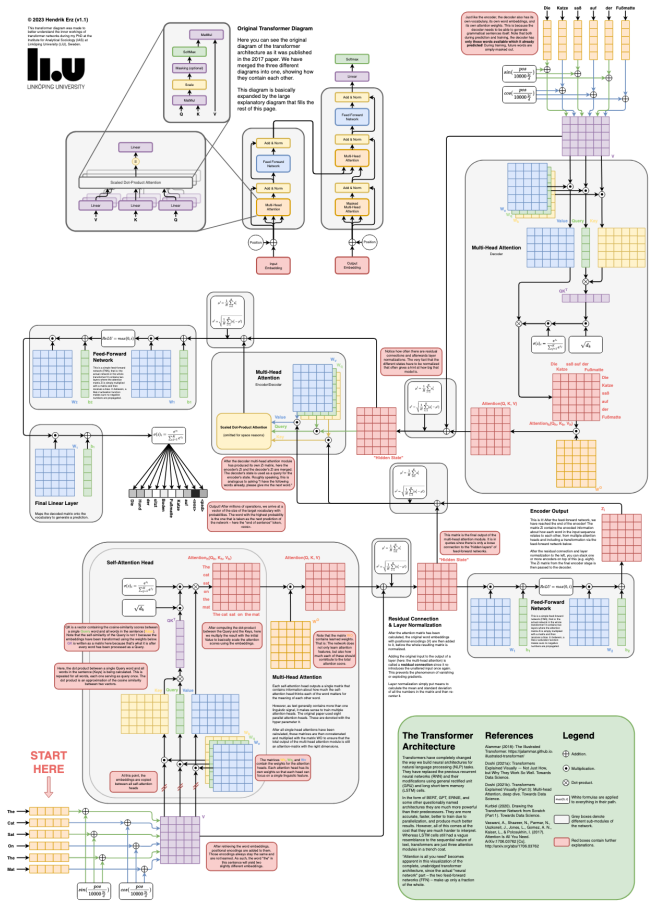
The Transformer Architecture
(source: https://www.hendrik-erz.de/post/the-transformer-architecture-a-visual-guide-pdf-download)

Large Language Models

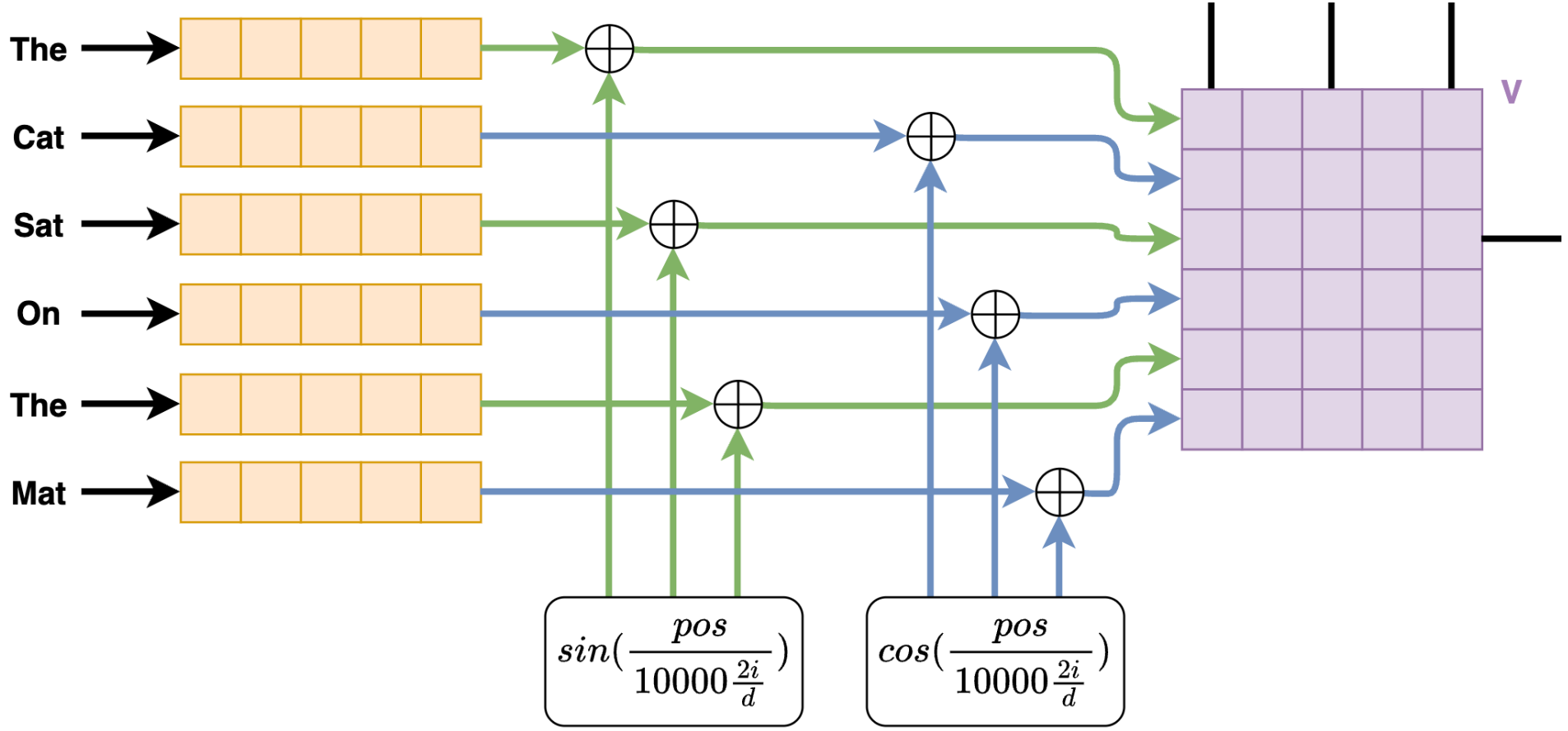


The Transformer Architecture
(source: https://www.hendrik-erz.de/post/the-transformer-architecture-a-visual-guide-pdf-download)

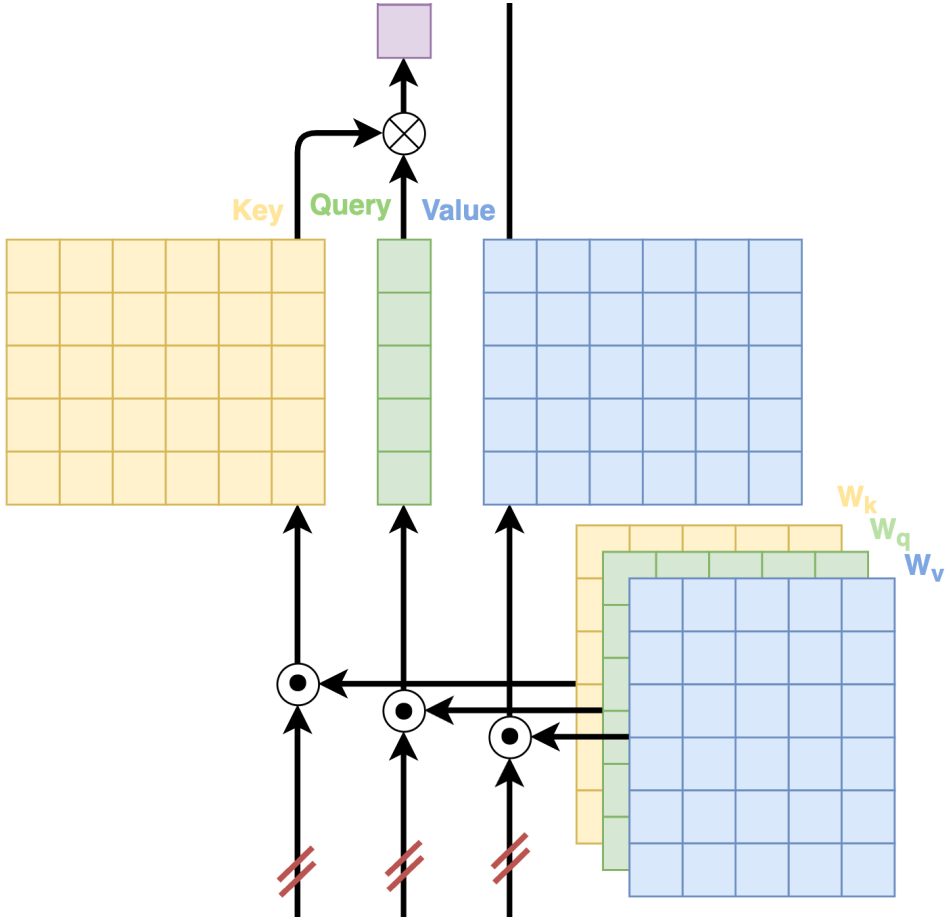
The Transformer Architecture



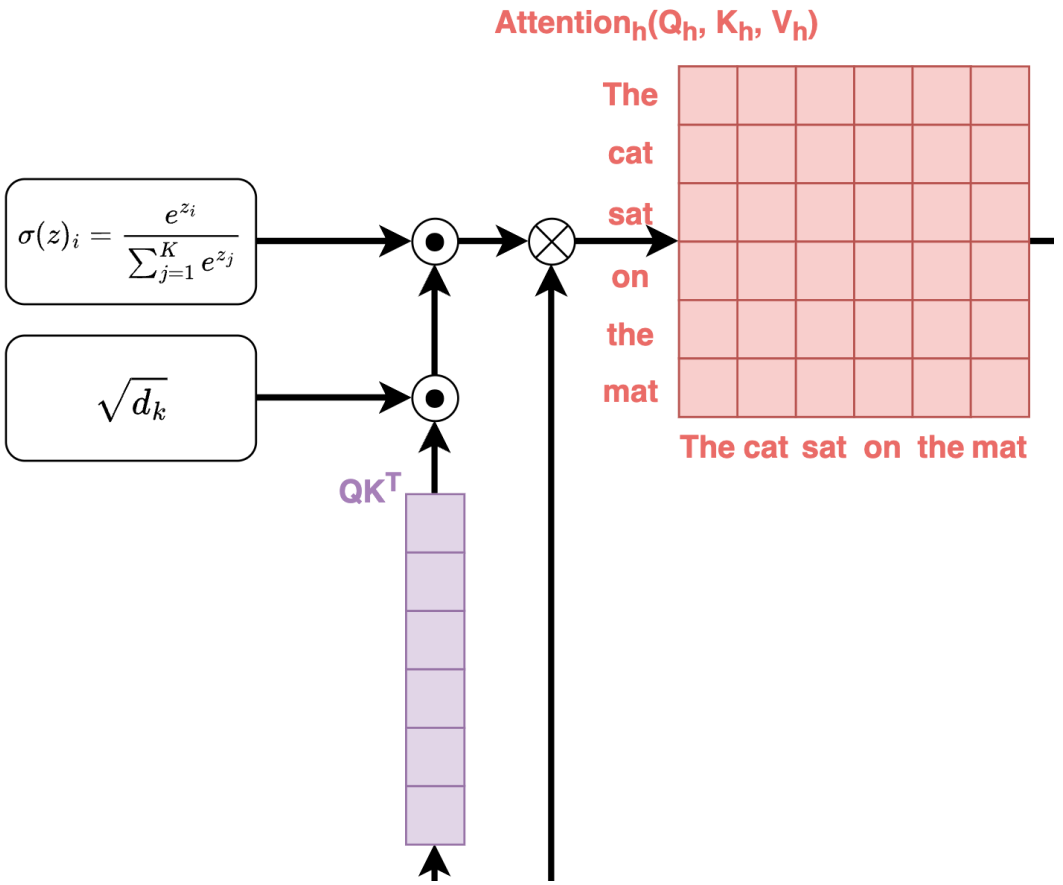
The Transformer Architecture



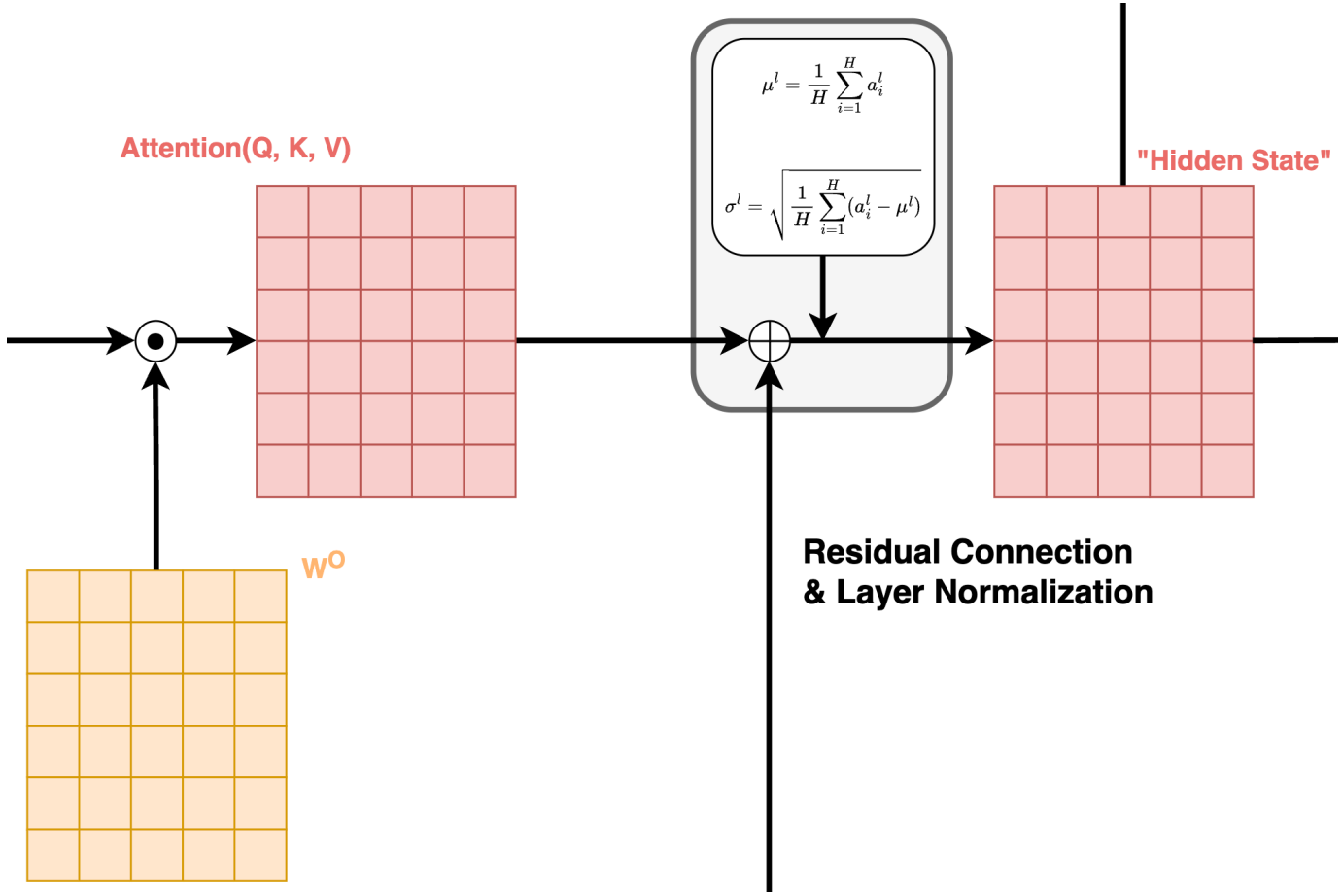
The Transformer Architecture



The Transformer Architecture



The Transformer Architecture



The Transformer Architecture

$$\mathbf{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

Active Learning

- You can use BERT models to classify text
  - This is called Active Learning (Ein-Dor 2020)
- Usually used to assign class labels to sentences/paragraphs
  - Always dependent on your research question
- For example: Whom do students acknowledge in their theses?
  - Categories: Advisors, Family, Institute, Library, …
  - Example: “For my Mom, Litaifah, and my Dad, Omar.” → Family support

Active Learning

1. Preparation:
    1. Define class labels
    2. Design a code book (what counts, what doesn't?)
    3. Prepare your corpus (clean, preprocess, split up into sentences)
    4. Draw a random sample (e.g., 100 sentences)
    5. Manually annotate these yourself; split into train/test
2. Active Learning Step:
    1. Fine-Tune a BERT-variant on your training data
    2. Calculate the F1 on the test set
    3. Let it annotate the entire corpus
    4. Sample new sentences, correct their labels, add to training data, repeat

Generative AI

- After the launch of ChatGPT, social scientists have started using decoder-models to generate text
- Two general approaches:
  - "In silico sampling" (Argyle et al., 2023)
  - Text generation (Törnberg et al., 2023)
- *However*:
  - "In silico sampling" must assume that the model possesses extensive "world knowledge"
  - Text generation must assume that the model's training data is similar to the target text generation process

Coda

State of the Art in Computational Text Analysis

CTA is not yet standardized

- Many new ways to utilize CTA are explored every day
- Creative re-using of methods
  - E.g., using perplexity scores (Vicinanza et al., 2020)
  - Using generative AI to mimic online discussions (Törnberg et al., 2023)
  - Using topic-models for other data types
  - Vary word embeddings (Levy & Goldberg, 2014)
  - etc. …

Remaining Issues With CTA

- CSS often forgets theoretical rigor → weak results
- CTA scholars have little understanding of linguistics → bad mapping of methods and theory
- The hype curve incentivizes novelty, not rigor
- But: We are only at the beginning; many new opportunities

Some Final Tips

- Don’t let yourself be guided by the hype curve
  - While we are still approaching its peak, do not jump onto every “new cool kid on the block”
  - As we approach the valley of despair, don’t be forlorn about methods as people dissuade you
- Always make sure that your theoretical and methodological assumptions map onto the theoretical and methodological assumptions of your method, and control for any biases. Then you’ll be good!

Afternoon Workshop

- Experiment with CTA methods yourself
- We provide a set of Python and R notebooks
  - Note that many CTA methods only work in Python!
- We will help you
  - to set up Python or Colab
  - in guiding you through the exercises
  - to apply these methods to your own research data
- Staff: Me, Diletta Goglia, Alexandra Rottenkolber
- Kopparhammaren 2; KO22, 3.00pm–5.30pm

Afternoon Workshop

SICSS-2025 / 03_text_analysis /

Add file

dilettagoglia AI notebook upload — c8a1886 · 5 minutes ago — Previous — History

Name	Last commit message	Last commit date
..		
.gitignore	Rename folders	15 hours ago
AL_Workshop.ipynb	AI notebook upload	5 minutes ago
AL_corpus.tsv	Rename folders	15 hours ago
AL_gold_data.tsv	Rename folders	15 hours ago
NLP_Workshop.ipynb	Rename folders	15 hours ago
NLP_Workshop_solutions.Rmd	Rename folders	15 hours ago
NLP_Workshop_solutions.ipynb	Rename folders	15 hours ago
NLP_workshop.Rmd	Rename folders	15 hours ago
README.md	Rename folders	15 hours ago
sotu.tsv	Rename folders	15 hours ago

Thank you!

Email: hendrik.erz@liu.se

Website: www.hendrik-erz.de

Bluesky: @hendrik-erz.de

Mastodon: @hendrikerz@scholar.social