

# Advanced Methods in Text Analytics

## Introduction



# Hello!

- [Daniel Ruffinelli](#) (PostDoc)
  - Got a PhD with Prof. Rainer Gemulla
  - Research focus *was* machine learning applied to knowledge graphs
  - Experience with ML research, ML/DL teaching
  - *Now* doing PostDoc in NLP with Prof. Simone Ponzetto
  - I will handle all lectures and all tutorials in this course
- **Focus Group:** Natural Language Processing and Information Retrieval
  - We offer the Information Retrieval course in other semesters



# When do we meet?

- Lectures
  - **When:** Tuesdays at 13:45
  - **Where:** A5 6, Room C015
- Tutorials
  - **When:** Wednesdays at 8:30
  - **Where:** A5 6, Room C015
- **First tutorial next week (Feb. 19th)**
  - Basic Python
  - Basic PyTorch
  - Feel free to skip it if you're already familiar with these

# About this course (1)

- Follow-up from IS 661 Text Analytics by Prof. Strohmeier
  - We might briefly cover some concepts from that course
  - But we generally assume the content in that course is known
- **Official requirements** of this course:
  1. Basic: linear algebra, probability theory, calculus (basic = BSc level)
  2. Having finished one of the following courses: Text Analytics (TA), Machine Learning (ML) or Deep Learning (DL) (mostly one of the latter two)
- **Examples of knowledge assumed/required for this course:**
  - **How ML models are trained** (i.e. the basic training loop, which we will ask you to implement at some point in this course), plus algorithms like **gradient-descent** and the use of **optimizers**
  - **How ML models are evaluated** (i.e. the evaluation loop, which you will be asked to implement as well), metrics like **accuracy**, **precision**, **recall**, etc.
- Additionally: you will need to **code in Python/PyTorch**
  - We provide a brief introduction in the first tutorial

# About this course (2)

- **What you will learn:**
  - Fundamentals of Deep Learning for Natural Language Processing (NLP)
  - Concepts/methods in latest developments in NLP research
  - In other words, state-of-the-art methods for NLP (active field!)
- This should allow you to:
  - Understand large language models (LLMs) “under the hood”
  - Read NLP research papers
  - Potentially work in NLP engineering
- There will be **some overlap with ML/DL courses**
  - In general, basic ML/DL is repeated in many courses (unavoidable)
  - We’ll point out when this is so
  - We’ll try to keep it to a minimum
  - But this is NOT a replacement for ML/DL courses
  - We encourage all of you to take those courses
- **(1) Intuition and (2) mathematical details are equally important here!**

# About this course (3)

- **I tend to speak very fast**
  - This is not good for teaching!
  - Raise your hand and politely ask me to slow me down if necessary
  - This works well in practice
- The course has quite a bit of content
  - 14 weeks: 13 lectures + Final QA (or optional lecture, finishing last lecture)
  - Exercises almost every week
- **Main challenge:** strike a balance between delivering *all of the content* and delivering it *with clarity*
  - This is why taking requirement courses is important
  - If you haven't, clarity of the content may suffer
- **Warning:** this course and the Information Retrieval course are very different
  - In this course, we go into more depth and details



# Course Logistics

- **Final grade:**
  - 100% of the grade comes from the final exam
- **Lectures**
  - **Goal:** to introduce and discuss concepts and methods
  - **Format:** references provided in last slide of each set of lecture slides, useful references linked [like this](#) throughout the course
- **Tutorials**
  - **Goal:** support lectures with deeper dives into same or new but related concepts/methods
  - **Format:** released one week, **you work on it at home**, solutions are discussed the week after

# Tentative List of Topics

- Most of **our focus** is on *methods*
  - Basics of ML and DL, feed-forward neural networks
  - Word representations, static and contextual, pre-training
  - Recurrent neural networks, attention
  - Transformers
  - Tokenization
  - Transfer learning (pre-training + fine-tuning)
  - Large language models (LLMs): architecture, tuning, applications
  - NLP Applications (LLM evaluation, common tasks)
  - Multilingual NLP
- **Less focus on tasks** (we discuss some of them when needed)
  - Language modeling (discussed extensively throughout the course)
  - Machine Translation
  - Question Answering
  - Dialogue Systems



# Reference Books

- Speech and Language Processing (3rd ed. draft) by Jurafsky and Martin
  - <https://web.stanford.edu/~jurafsky/slp3/>
- NLP, An ML perspective, Zhang et al.
  - Online access via the University library
- NLP by Eisenstein et al.
  - <https://cseweb.ucsd.edu/~nnakashole/teaching/eisenstein-nov18.pdf>
- Most of the content in 2nd half of the course comes from papers released in last 5 years or so
  - They will be referenced via links in these slides

# Outline

What is Text Analytics?

*What is Advanced Text Analytics?*

**Recap:** Basic NLP Concepts

# What is Text Analytics?

# Natural Language Processing

- **Eisenstein:** set of methods for making human language accessible to computers.
- **Zhang and Teng:** the study of automatically processing or synthesizing human languages.
- **Generally:** interdisciplinary area, e.g. some linguistics, some computer science, some machine learning.
- Similar/related terms:
  - **Computational linguistics:** main focus is language, not computational methods
  - **Speech processing:** main focus is processing audio into text
  - **Text Analytics:** drawing value from text (we treat it as synonym to NLP)



# Challenges

- Started in the 1950s as part of artificial intelligence research
  - Initially thought to be easy, challenges quickly emerged
- **Main challenge:** ambiguity in language
- Semantic ambiguity: “They can fish here.”
  1. They are allowed to fish in that location.
  2. They put fish in cans in that location.
- Lexical ambiguity: “L’avocat est juste là.”
  - Is the *lawyer* right there?
  - Or is the *avocado* right there?
- Named entity ambiguity:
  - “Michael Jordan is the Miles Davis of machine learning.”
  - Who? The basketball player or the professor at UC Berkeley?
- **Other challenges:** idioms, e.g. he is “out of his mind”; multilingualism, e.g. many “low-resource” languages; basic knowledge not given in text

# NLP Tasks

- NLP is a broad area that studies a **wide range of tasks**.
- Some examples of NLP tasks:
  - Question answering (QA)
  - Machine translation (MT)
  - Text summarization
  - Assisted writing
- Useful to distinguish them by their inputs and outputs

# Question Answering (QA) Machine Translation

×  

Bilder Videos News Bücher Maps Flüge Finanzen

Ungefähr 39.600.000 Ergebnisse (0,65 Sekunden)

William Shakespeare / Geburtsdatum

**April 1564**

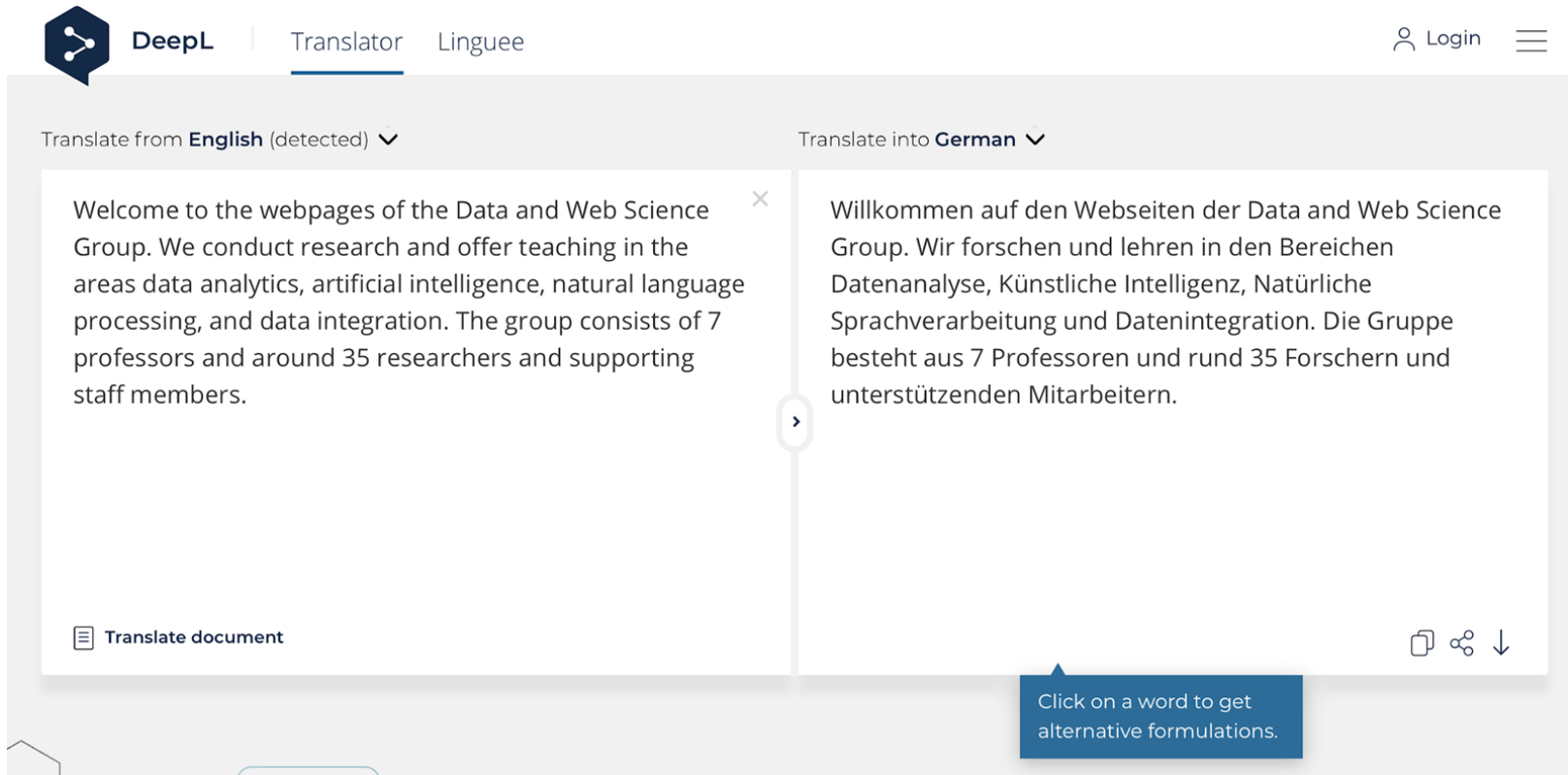


William Shakespeare ['wɪljəm 'ʃeɪkspɪə] (getauft am 26. April 1564<sup>jul.</sup> in Stratford-upon-Avon; gestorben am 23. April<sup>jul.</sup> / 3. Mai 1616<sup>greg.</sup> ebenda) war ein englischer Dichter, Theaterunternehmer und Schauspieler, dessen Dramen zu den bedeutendsten Werken der Weltliteratur gehören.



# Machine Translation (MT)

## Machine Translation

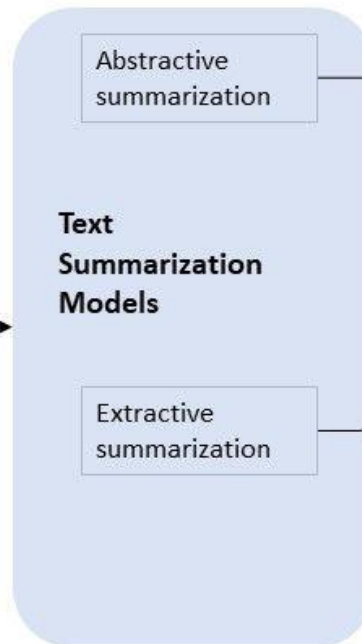


The screenshot displays the DeepL Translator web interface. At the top, the DeepL logo is on the left, and 'Translator' and 'Linguee' are in the center. On the right, there is a 'Login' button and a menu icon. Below the header, the interface is split into two main sections. The left section is labeled 'Translate from English (detected)' and contains a text box with the English text: 'Welcome to the webpages of the Data and Web Science Group. We conduct research and offer teaching in the areas data analytics, artificial intelligence, natural language processing, and data integration. The group consists of 7 professors and around 35 researchers and supporting staff members.' Below this text box is a 'Translate document' button. The right section is labeled 'Translate into German' and contains the German translation: 'Willkommen auf den Webseiten der Data and Web Science Group. Wir forschen und lehren in den Bereichen Datenanalyse, Künstliche Intelligenz, Natürliche Sprachverarbeitung und Datenintegration. Die Gruppe besteht aus 7 Professoren und rund 35 Forschern und unterstützenden Mitarbeitern.' At the bottom right of the German text box, there are icons for copying, sharing, and downloading. A blue tooltip box at the bottom right of the interface says 'Click on a word to get alternative formulations.'

# Text Summarization

## Input Article

Marseille, France (CNN) The French prosecutor leading an investigation into the crash of Germanwings Flight 9525 insisted Wednesday that he was not aware of any video footage from on board the plane. Marseille prosecutor Brice Robin told CNN that "so far no videos were used in the crash investigation." He added, "A person who has such a video needs to immediately give it to the investigators." Robin's comments follow claims by two magazines, German daily Bild and French Paris Match, of a cell phone video showing the harrowing final seconds from on board Germanwings Flight 9525 as it crashed into the French Alps. All 150 on board were killed. Paris Match and Bild reported that the video was recovered from a phone at the wreckage site. ...



## Generated summary

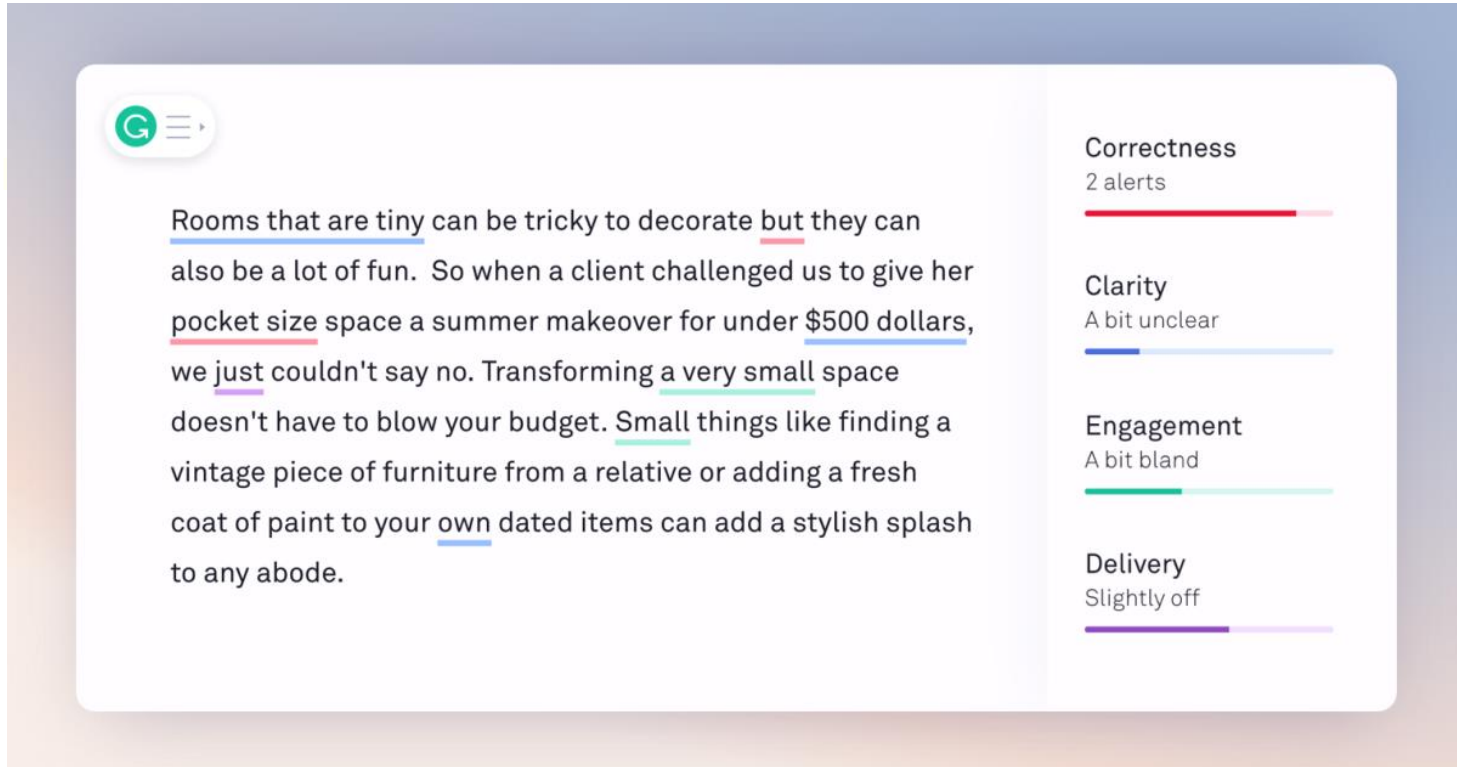
Prosecutor : " So far no videos were used in the crash investigation "

## Extractive summary

marseille prosecutor brice robin told cnn that " so far no videos were used in the crash investigation . " robin \s comments follow claims by two magazines , german daily bild and french paris match , of a cell phone video showing the harrowing final seconds from on board germanwings flight 9525 as it crashed into the french alps . paris match and bild reported that the video was recovered from a phone at the wreckage site .

[Image source](#)

# Assisted Writing



The screenshot displays a writing assistant interface. On the left, a paragraph of text is shown with various words highlighted in different colors (blue, red, purple, green). Above the text is a green circular icon with a white 'G' and a menu icon. On the right, four metrics are listed, each with a horizontal progress bar:

- Correctness**: 2 alerts (red bar)
- Clarity**: A bit unclear (blue bar)
- Engagement**: A bit bland (green bar)
- Delivery**: Slightly off (purple bar)

[Image source](#)

# The Role of Machine Learning in NLP

- Current NLP methods rely almost exclusively on ML techniques to solve tasks.
- ML (specifically, Deep Learning) allows complex solutions to be built from training on large amounts of data and without requiring knowledge about linguistics
- However, there are **fundamental differences with NLP**
  1. Text data is discrete, unlike audio or images.
    - But new words are constantly created
    - And word distribution is highly skewed, meaning it's challenging for algorithms to be robust on less frequent words.
  2. Language is compositional.
    - Meaning units such as words can be combined to create new phrases with new meanings

# NLP Tasks from a ML Perspective

- From an ML perspective, there are fewer types of NLP tasks
- Based on output of task:
  - **Classification:** model produces categorical output, e.g. one of few possible sentiments, one of thousands of possible words
  - **Structured prediction:** model produces structures with inter-related substructures, e.g. POS-tagging and dependency parsing
  - **Regression:** model produces real-valued prediction, e.g. automatic essay scoring
- Based on training data:
  - **Unsupervised:** unlabelled training data
  - **Supervised:** labelled training data
  - **Self-supervised:** in-between both

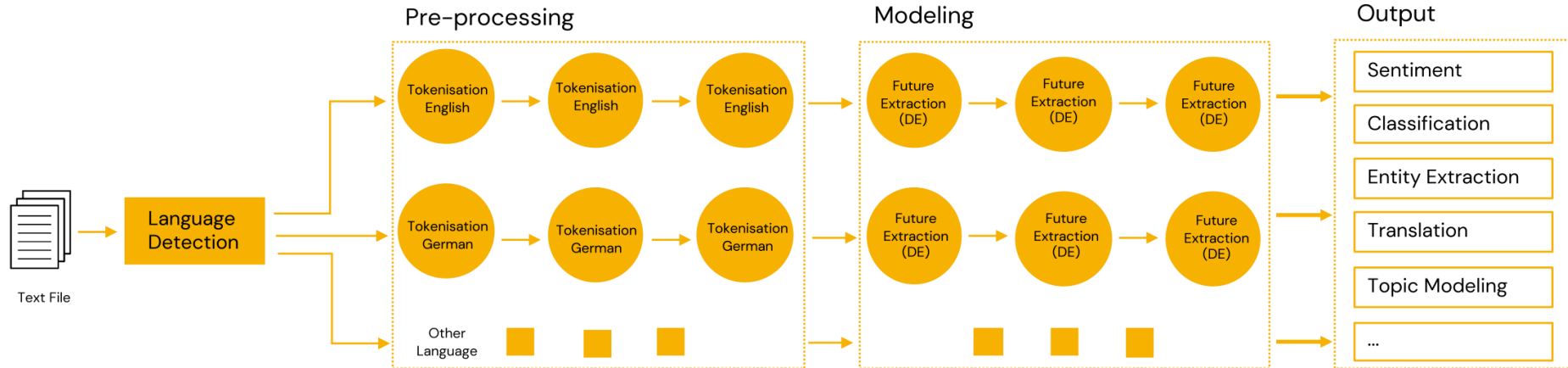
# What is *Advanced* Text Analytics?

# Traditional vs Modern NLP

- The term *advanced* mostly refers to methods
  - Most state-of-the-art methods today share similar foundations in deep learning
- NLP has been through a revolution in the past decade.
  - Mostly driven by deep learning technology
  - Similar revolutions have taken place in Computer Vision and Speech Processing
- **Traditional methods:** relied on expert knowledge manually injected into automated solutions, e.g. structures encoded into regular expressions
- **Modern NLP methods:** big-data and high compute power allowed deep learning solutions to largely out-perform traditional methods in all NLP tasks without requiring expert knowledge (see [The Bitter Lesson](#))



# Traditional NLP

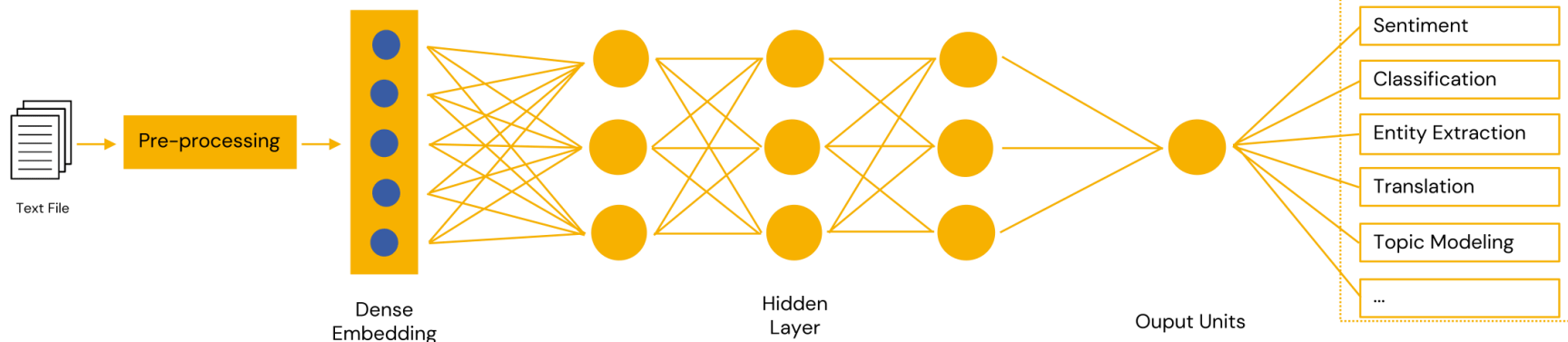


[Image source](#)

- Relied heavily on pre-processing methods, feature engineering
  - Pre-processing may drop useful information, e.g. punctuation
  - Feature engineering required expensive expert knowledge
  - Steps were often task-dependent
- **Important:** some of this experience could become useful again

# Modern NLP

## Deep Learning



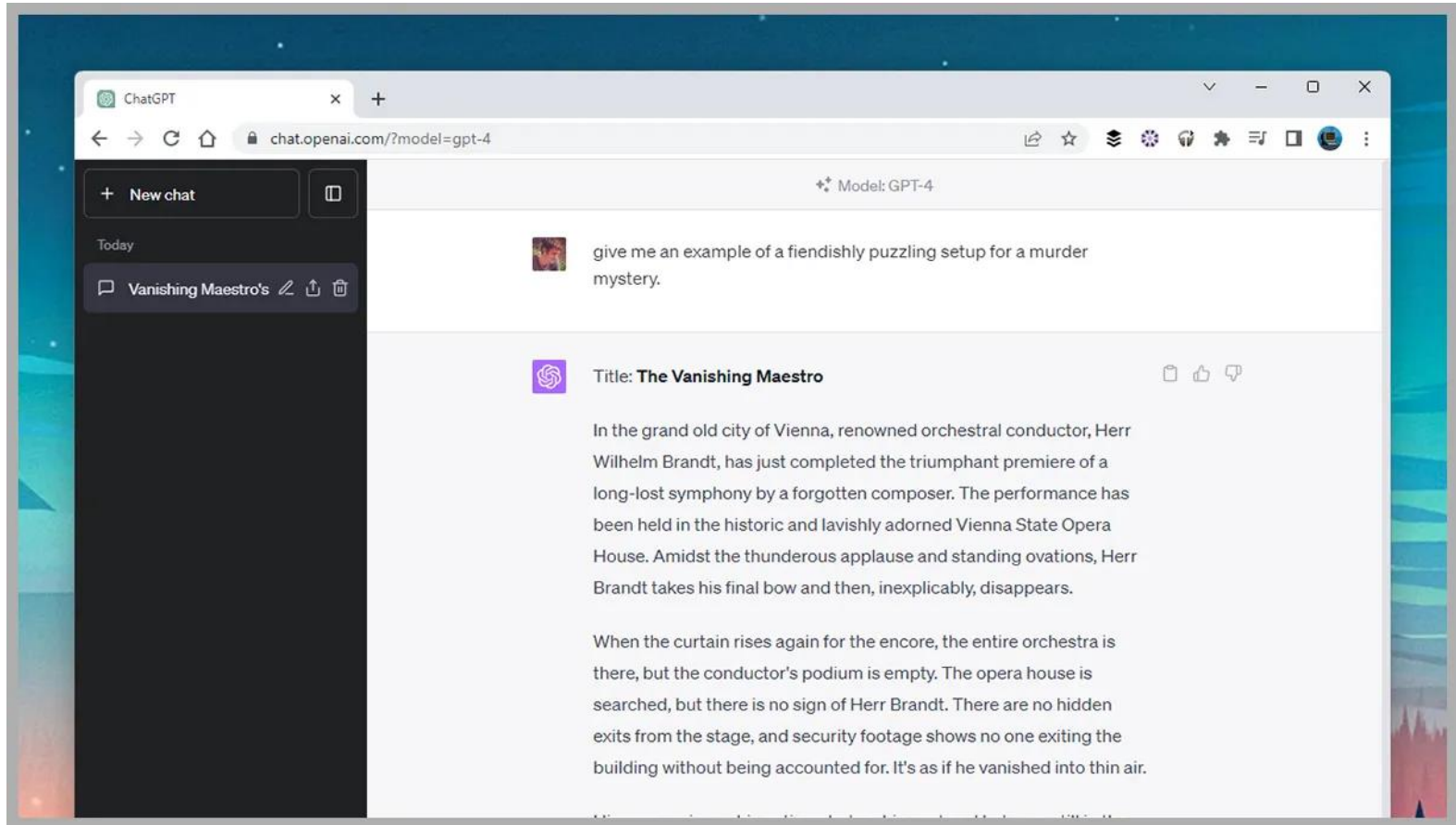
[Image source](#)

- Deep learning techniques based on learning **generally useful representations of words** (tokens), which are then used in several different types of downstream applications (QA, MT, etc.)
- **General intuition:** learned representations encode meaning of words
- Training such representations is prohibitively expensive (more later)

# ChatGPT and Other LLMs

- [ChatGPT](#) and similar models, e.g. [Google's Gemini](#) and [DeepSeek](#), are examples of the success of Deep Learning and the progress of NLP solutions
- These models are **chatbots** based on **large language models (LLMs)**, in turn based on the **transformer architecture** and trained with manually designed **training objectives**
- **In this course, we will cover:**
  - How these models are designed to learn representations
  - How these models are trained to learn representations
  - How these models/representations are used in downstream applications
  - In short, **how LLMs work internally**
- Some of the abilities of these models are remarkable!
  - Let's go over some of them

# ChatGPT for Writing



[Image source](#)

# ChatGPT for Creating Tables

@ChatGPT 1 find the most popular Ford car models, based on US sales figures, between 2010 and 2020, and make a two column table for it.

@ChatGPT 1 find the most popular Ford car models, based on US sales figures, between 2010 and 2020, and make a two column table for it.




ChatGPT 1  BOT Today at 12:44 PM

Here are the most popular Ford car models, based on US sales figures, between 2010 and 2020:


Model	Total Sales
F-Series Pickup	7,695,241
Escape	2,448,437
Fusion	2,138,817
Focus	1,979,730
Explorer	1,840,255
Edge	1,070,098
Fiesta	917,276
Mustang	909,330
Taurus	657,223
C-Max	60,571

[Image source](#)

# ChatGPT for Coding



I apologize for the error in the code that I provided. The `gridify` function is actually part of the `shapely` library, not `geopandas`. Here is the updated code that you can use to create a grid of geodataframe within a polygon:

 Copy code

```
import geopandas as gpd
from shapely.geometry import Polygon

# create a polygon
polygon = gpd.GeoDataFrame({'geometry': gpd.GeoSeries([Polygon([(0,0), (0,1), (1,1), (1,0)])])})

# create a grid of geodataframe within the polygon
from shapely.gridify import gridify
grid = gpd.GeoDataFrame(
    gridify(polygon.geometry[0], cell_size=0.1, geometry='Polygon')
)

# plot the grid
grid.plot()
```

This should create a grid of geodataframe within the polygon, with each cell having a size of 0.1 units. You can adjust the `cell_size` parameter to change the size of the cells in the grid.

[Image source](#)

# Recap: Basic NLP Concepts



# Contents of Text Analytics Course

- Basic concepts
  - Basic text processing, e.g. stemming, segmentation, tokenization, normalization
  - Basic concepts: similarity, relatedness, disambiguation
- Language modeling, POS tagging Intro to ML, DL
- Word senses, word embeddings
- **Today:**
  - Quick recap of some important concepts
- **Part of next few lectures:**
  - Recap the rest as needed
- No recap is exhaustive
  - Again, we assume either the TA, ML or DL course

# Similarity and Relatedness

- Concept of **similarity** *very useful* in many NLP tasks
- **Question:** How to define/measure similarity?
  - Edit distance? E.g. typo *graffe* closest to *giraffe* than *grail*
  - A data base of synonyms?
- Synonymity is usually a yes/no question.
  - Is similarity that simple?
  - There are other types of relations between words
  - E.g. went is *derived from* go
  - This more generally refers to **word relatedness**
- Are all semantically related words similar?
  - **Similarity:** can substitute one word for the other
  - **Relatedness:** semantic correlation, but not interchangeable (more general)
- Further, do we measure similarity between words or senses?
  - **Word:** Bass
  - **Sense 1:** type of fish; **Sense 2:** musical instrument

# Word Sense Disambiguation

- **Identifying** intended **sense** of each word in a document
  - “Drunk gets nine years in violin case”
  - Is it a violin case? Or a legal case?
- Sense is a property of lemmas (roughly, roots of words), not of words themselves
- Word sense disambiguation is thus:
  - Identify lemma
  - Choose correct sense
- **How often do words have many senses?**
  - **WordNet**: big manual effort to encode sense as knowledge base of sets of synonyms (synsets)
  - Example: Wordnet has 8 distinct senses for the word bass
  - Also contains other types of relations between synsets, e.g. antonym, hyponym, hypernym, meronym, etc.
- So, task is not trivial!

# Tokenization (1)

- **Goal:** segmenting text into words.
  - Words from a finite set, a vocabulary
- But **what exactly is a word?** Not always clear
  - Are punctuation marks words?
  - And acronyms such as U.S.A.? How many words is that?
- **Such decisions often made by the tokenizer (TKZ)**
  - Example: “We are the champions, my friends!”
  - TKZ 1: {'We', 'are', 'the', 'champions', 'my', 'friends'}
  - TKZ 2: {'We', 'are', 'the', 'champions', ',', 'my', 'friends', '!'}
  - TKZ 3: {'We', 'are', 'the', 'UNK', ',', 'my', 'friends', '!'}
- Thus, **basic segments** often referred to as **tokens**
  - What these are depends on tokenization method
  - High-level discussion today
  - We cover these in more detail later in the course

# Tokenization (2)

- Ideally:
  - Tokens encode meaning
  - Tokenization is fast! (Basic pre-processing step)
  - Memory efficient (often required on GPU memory)
  - Coverage (less chance of finding unknown token)
- **Question:** What should our basic tokens be?
- 1. **Words?** Obvious suggestion, but again, what are words?
  - And what about written languages that don't delimit words?
  - And words composed of subwords, e.g. *Wahrscheinlichkeitstheorie*?
- 2. **Characters?** Above problems solved!
  - Also efficient (small vocab.), even if including ALL languages
  - But what meaning do single characters encode? Often none
- 3. **Subword?** Often the best trade-off.
  - Encodes meaning, e.g. words **token** and **tokenizer** are related
  - Reasonably sized vocabulary

# Language Modeling (1)

- Predict the next word:
  - “Every Thursday there is a Schneckenhof ...”
- Which word is more likely to follow?
  - Meeting?
  - Party?
  - Notebook?
- This suggests:
  - **Some words are more likely to appear than others given some context**
- Probabilistically, a language model (LM) computes the following:
  - $p(\text{meeting} / \text{“Every Thursday there is a Schneckenhof”})$
- **Generally:  $p(w_{n+1} / w_1, w_2, \dots, w_n)$** 
  - Conditional probability of  $w_{n+1}$  given joint distribution of  $w_1, w_2, \dots, w_n$
- **Such a model can predict entire sequences with the chain rule**
  - $p(w_1, w_2, \dots, w_n) = p(w_1)p(w_2/w_1) p(w_3/w_1, w_2) \dots p(w_n/w_1, w_2, \dots, w_{n-1})$

# Language Modeling (2)

- In many NLP applications, goal is producing word/token sequences
  - Machine translation
  - Summarization
  - Dialogue systems
- The following task is therefore useful:
  - Given vocabulary  $V = \{aardvark, abacus, \dots, zither\}$ , **predict probability of sequence of words**  $p(w_1, w_2, \dots, w_M)$ , with  $w_m \in V$
- For example, in machine translation
  - **Input:** *El café negro me gusta mucho.*
- Say a translation system provides the following possible answers
  1. *The coffee black pleases me much.* (word-for-word translation)
  2. *I love dark coffee.*
- A good language model should say:
  - $p(\textit{The coffee black pleases me much}) < p(\textit{I love dark coffee})$



# Language Modeling with N-Grams

- **N-Gram:** sequence of  $n$  words
  - **2-gram:** sequence of two words, **3-gram:** sequence of three words
- Example: extract all 3-grams from the following toy text corpus
  - Corpus: "This is the example sentence."
  - 3-grams: "This is the", "is the example", "the example sentence"
- N-gram **models can model the task by counting n-grams** (e.g. 3-grams):
 

number of times we see "is the example"

  - $p(\text{example} | \text{"is the"}) = \frac{\text{number of times we see "is the example"}}{\text{number of times we see "is the"}}$
- Answer depends on text corpus used to count n-grams:
  - **Toy Corpus 1:** "This *is the* example of the best scenario, this *is the* outcome of that same scenario, and this *is the* example of the worst scenario."
  - **Toy Corpus 2:** "This *is the* example, while this *is the* analogy."
- **Ideally, probabilities estimated from large corpus of natural language**
  - Would be nice if it includes multiple domains (e.g. history, medicine, etc.)

# LLMs: Large Language Models

- **Language models:** they model the same task as n-gram models
  - In reality, there are variants of the language modeling task
  - They all relate to predicting new words given other words in same context
  - We'll cover the most common ones in this course
- **Large:** they have billions of parameters that are used to estimate those probabilities (i.e. model training)
  - The more parameters we have, the more data we need to avoid overfitting
  - Where does this data come from?
- **LLMs are trained on corpora with trillions of words**
  - E.g. the entire internet up to October 2023 ([GPT-4o cutoff date](#))
- This is why **developing LLMs today is prohibitively expensive**
  - Billions of parameters require lots of memory and data, thousands of GPUs
  - E.g. estimated cost of training Google's Gemini Ultra: 191 million USD ([Stanford's 2024 AI Index Report](#))

# Summary: Introduction

- We went over **structure of the course**
  - Goals
  - Requirements
  - Tentative topics
  - References
- We had an **overview of text analytics**
  - NLP
  - NLP Tasks
  - Role of ML, DL
  - Modern NLP Applications
- We went over some basic and **relevant NLP concepts**
  - Similarity vs relatedness
  - Word sense disambiguation
  - Tokenization
  - Language modeling

# References

- Jurafsky et al., Chapter 2
- Zhang et al., Chapter 1
- Eisenstein, Chapters 1 and 4