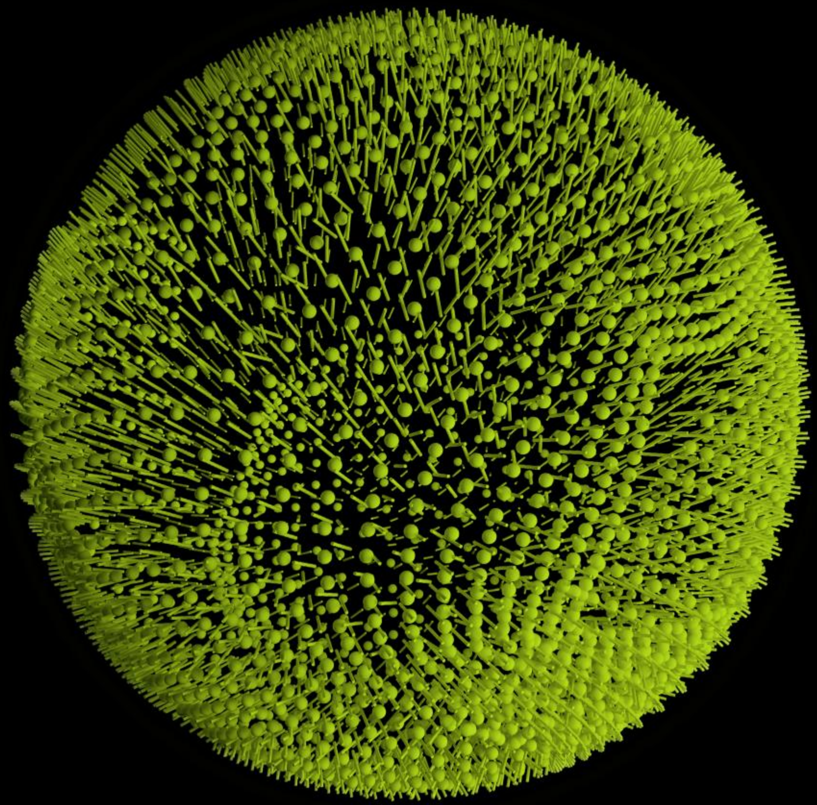
Deloitte.



NLP and Generative AI

Spikeup.ai

16 August 2023

Nice to meet you!

Your facilitators for this training

Anshul Aggarwal
Consultant - Artificial Intelligence

~1 year at Deloitte
Background in
Computer Science / AI

Alex Jang
Analyst - Artificial Intelligence

~1 year at Deloitte
Background in
Computer Science / AI

1 NLP tasks and feature representation

1. NLP preprocessing tasks
2. NLP feature representation
3. Neural networks

NLP has been split up in so-called “tasks”

Each task has specific use cases, models and evaluation metrics

Information Retrieval
Ranking a list of documents or search results in response to an input query

Sentiment Analysis
Classifying the polarity or “happiness” of a given text

Machine Translation
Translating a sentence in a source language to a different target language

Natural language inference
Determining whether a “hypothesis” is true (entailment), false (contradiction), or undetermined (neutral) given a “premise”

Reading comprehension
Answering questions on a given text passage

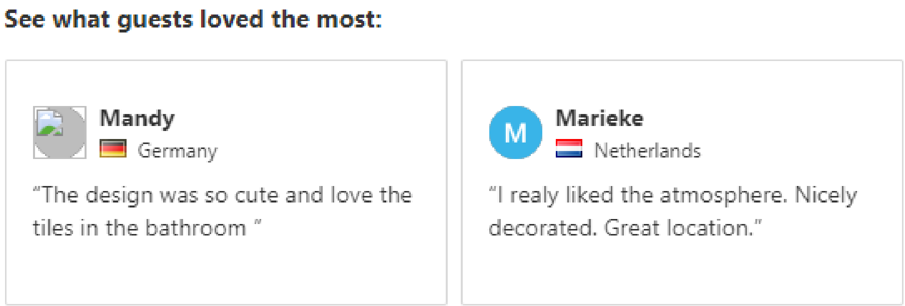
Summarization
Producing a shorter version of one or several documents that preserves most of the input’s meaning

Topic modelling
Discovering the abstract "topics" that occur in a collection of documents

Relation prediction
Recognizing a named relation between two named semantic entities

Sentiment analysis

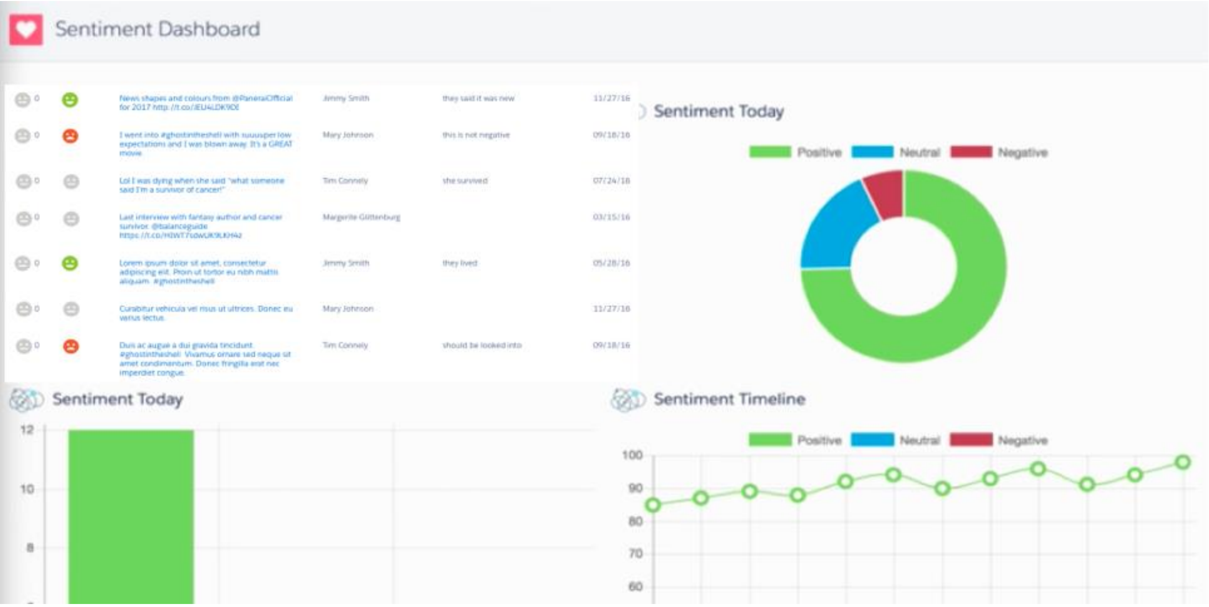
Given a typically short piece of text, classify the emotional tone of the message: positive, negative, or neutral



See what guests loved the most:

Mandy
Germany
"The design was so cute and love the tiles in the bathroom "

Marieke
Netherlands
"I realy liked the atmosphere. Nicely decorated. Great location."

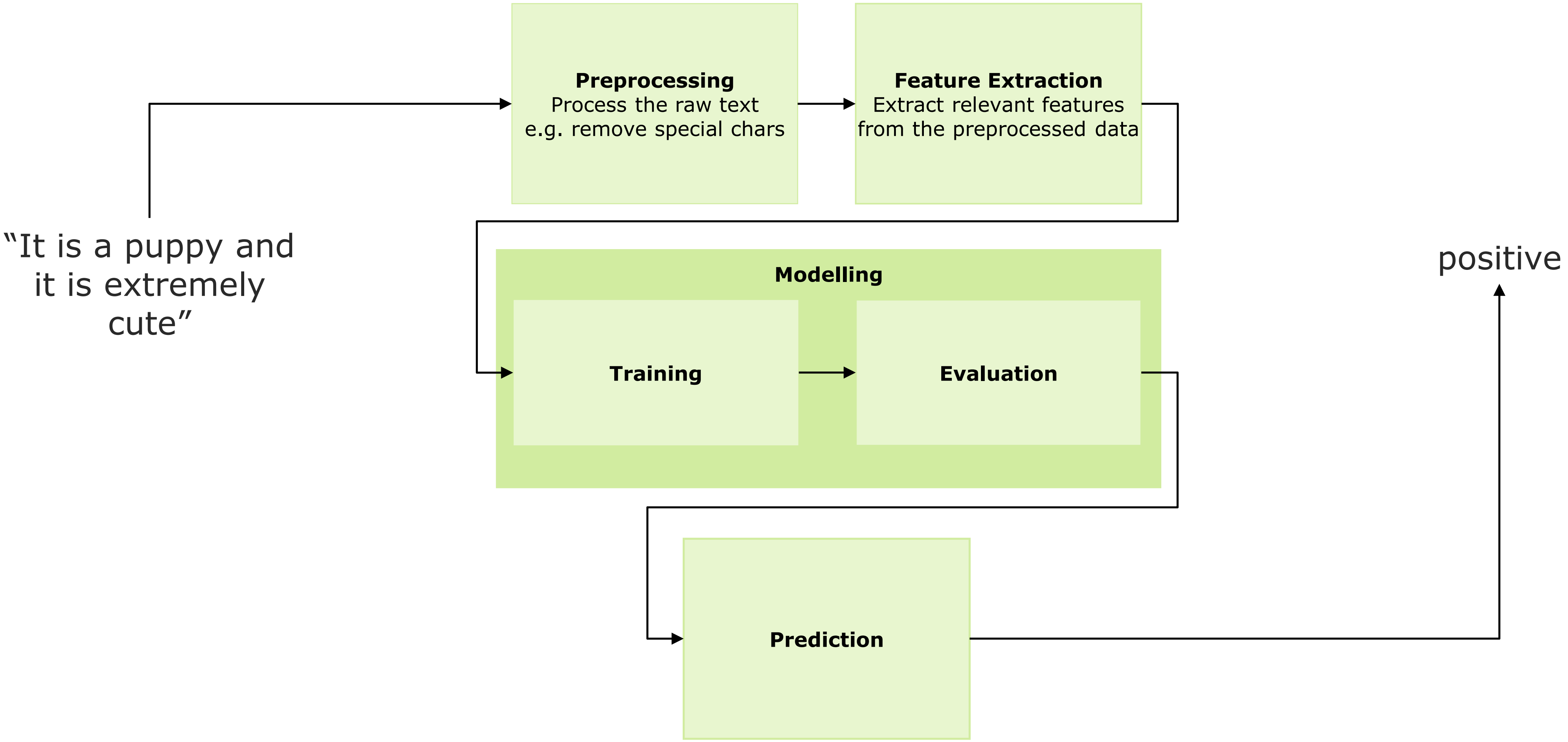


Sentiment Dashboard

😐 0	🙂	News shapes and colours from @PaneraiOfficial for 2017 http://t.co/JEU4LDK9DE	Jimmy Smith	they said it was new	11/27/16
😐 0	☹	I went into #ghostintheshell with suuuuper low expectations and I was blown away. It's a GREAT movie	Mary Johnson	this is not negative	09/18/16
😐 0	😐	Lol I was dying when she said "what someone said I'm a survivor of cancer!"	Tim Connely	she survived	07/24/16
😐 0	😐	Last interview with fantasy author and cancer survivor @balanceguide https://t.co/HIWT7s0WUK9LKH4z	Margerite Glittenburg		03/15/16
😐 0	🙂	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin ut tortor eu nibh mattis aliquam. #ghostintheshell	Jimmy Smith	they lived	05/28/16
😐 0	😐	Curabitur vehicula vel risus ut ultrices. Donec eu varius lectus.	Mary Johnson		11/27/16
😐 0	☹	Duis ac augue a dui gravida tincidunt. #ghostintheshell. Vivamus ornare sed neque sit amet condimentum. Donec fringilla erat nec imperdiet congue.	Tim Connely	should be looked into	09/18/16

Sentiment analysis

Given a typically short piece of text, classify the emotional tone of the message: positive, negative, or neutral



“It is a puppy and it is extremely cute”

Preprocessing
Process the raw text
e.g. remove special chars

Feature Extraction
Extract relevant features
from the preprocessed data

Modelling

Training

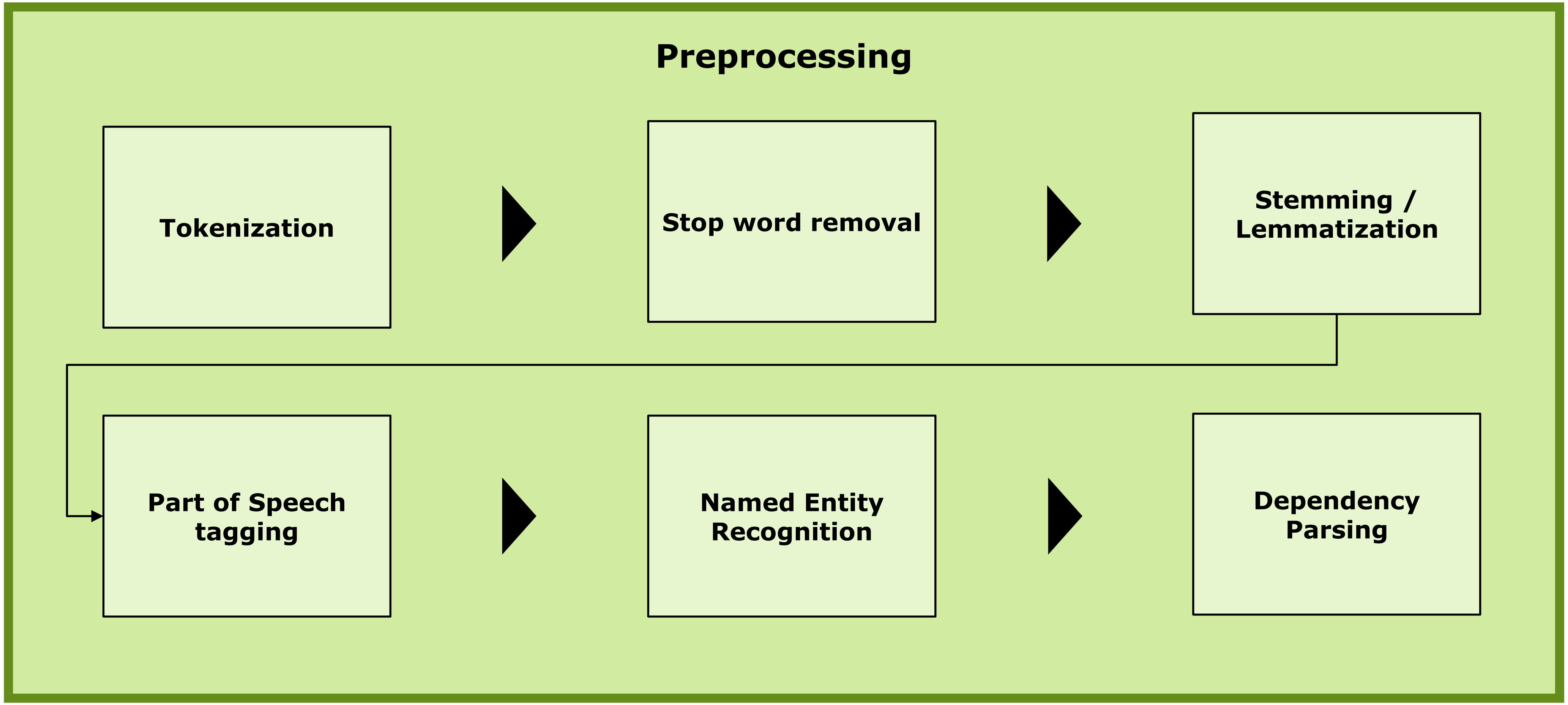
Evaluation

Prediction

positive

NLP preprocessing pipeline

What NLP preprocessing would you apply to get the sentiment per sentence?



**List is not exhaustive, reference:* [NLP is fun! Blog, Medium](),

NLP Preprocessing tasks

Several basic NLP subtasks exist to transform raw data to information

Tokenization

Replace a sequence of characters with a sequence of tokens (words and punctuation)

Stop word removal

Remove ubiquitous words from a sentence

Stemming

Cutting “inflected” words to their root forms by removing suffixes

Lemmatization

Group a so-called “inflected” form of a word to it’s dictionary form (“lemma”)

Part of Speech (PoS) tagging

Assigning parts of speech to individual words in a sentence

Named Entity Recognition (NER)

Identifying important words in a sentence

Dependency parsing

Identify how words relate to each other.

NLP subtasks and pipeline

Example outputs

Tokenization

“I like London,”

Stop word removal

“The city is large”

Stemming

“prettiest”

Lemmatization

“is”

Part of Speech (PoS) tagging

“Large city”

Named Entity Recognition (NER)

“London”

Dependency parsing

“London is the capital”

NLP subtasks and pipeline

Example outputs

Tokenization

"I like London,"

["I", "like", "London", ","]

Stop word removal

"The city is large"

["City", "is", "large"]

Stemming

"prettiest"

"pretti"

Lemmatization

"is"

"be"

Part of Speech (PoS) tagging

"Large city"

{"Large": "adjective", "city": "noun"}

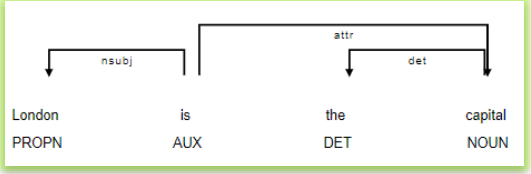
Named Entity Recognition (NER)

"London"

{"London": "geographical entity"}

Dependency parsing

"London is the capital"



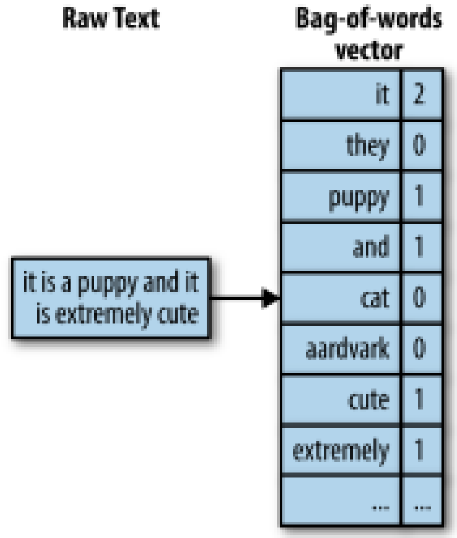
NLP and features

Encoding natural language into numerical values for computation

Bag of Words (BOW): Represent using presence/count of words in a piece of text with the idea that documents are similar if they contain the same words

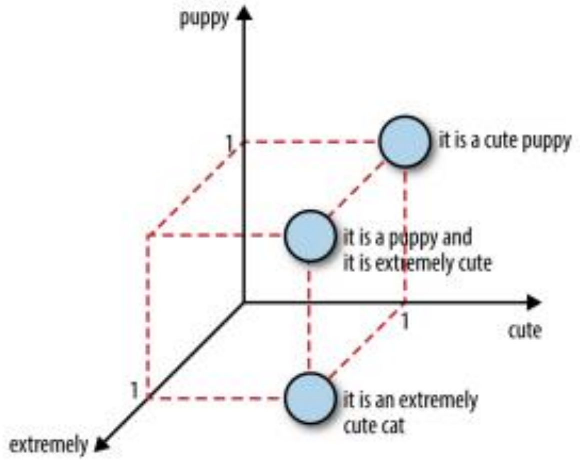
Bag of n-grams: Presence/count of bigrams/trigrams/n-grams; more informative since they capture more context around each word

Vector representation: These gives us a vector representation of natural language text that we can perform computations with.



it	2
they	0
puppy	1
and	1
cat	0
aardvark	0
cute	1
extremely	1
...	...

Bigram	
it is	1
is a	1
…	…
extremely cute	1
not cute	0



Embeddings

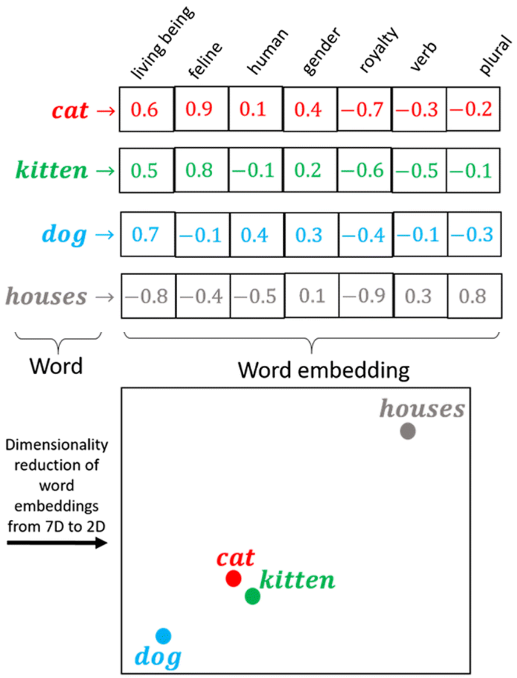
A special type of feature representation

Embeddings: Latent representations that preserve semantics.

- Compressed feature representation of data that contain semantic information
- Features along the dimensions are directly interpretable.
- Convert **words or sentences** into **vectors that preserves its meanings and structure**.
- Does not create deeper features. They are **generated using simpler architectures**.
- Can be used to compare pairs of words or sentences – do they mean something similar?
    - You can use cosine similarity!
- Are used in all LLMs.

Embeddings (e.g. word2vec)

let NN identify patterns

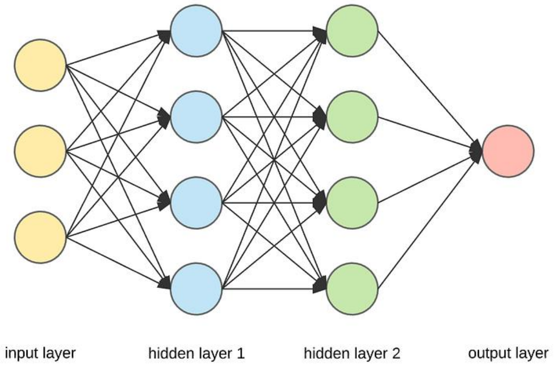


	living being	feline	human	gender	royalty	verb	plural
cat →	0.6	0.9	0.1	0.4	−0.7	−0.3	−0.2
kitten →	0.5	0.8	−0.1	0.2	−0.6	−0.5	−0.1
dog →	0.7	−0.1	0.4	0.3	−0.4	−0.1	−0.3
houses →	−0.8	−0.4	−0.5	0.1	−0.9	0.3	0.8

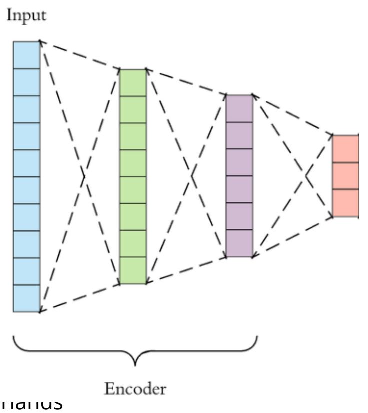
Neural networks

Artificial neurons that mimic the human brain

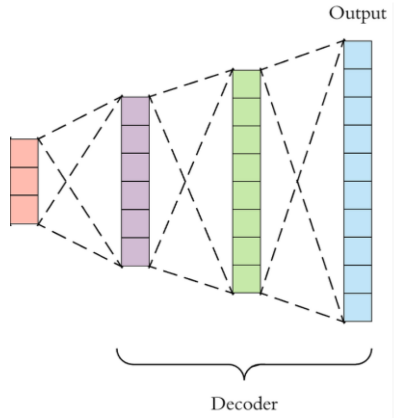
- **Neural network:** neurons organised in layers; information flows through the layers to give final output



- **Encoders:** produces a vector representation **encodes** relevant information from the input

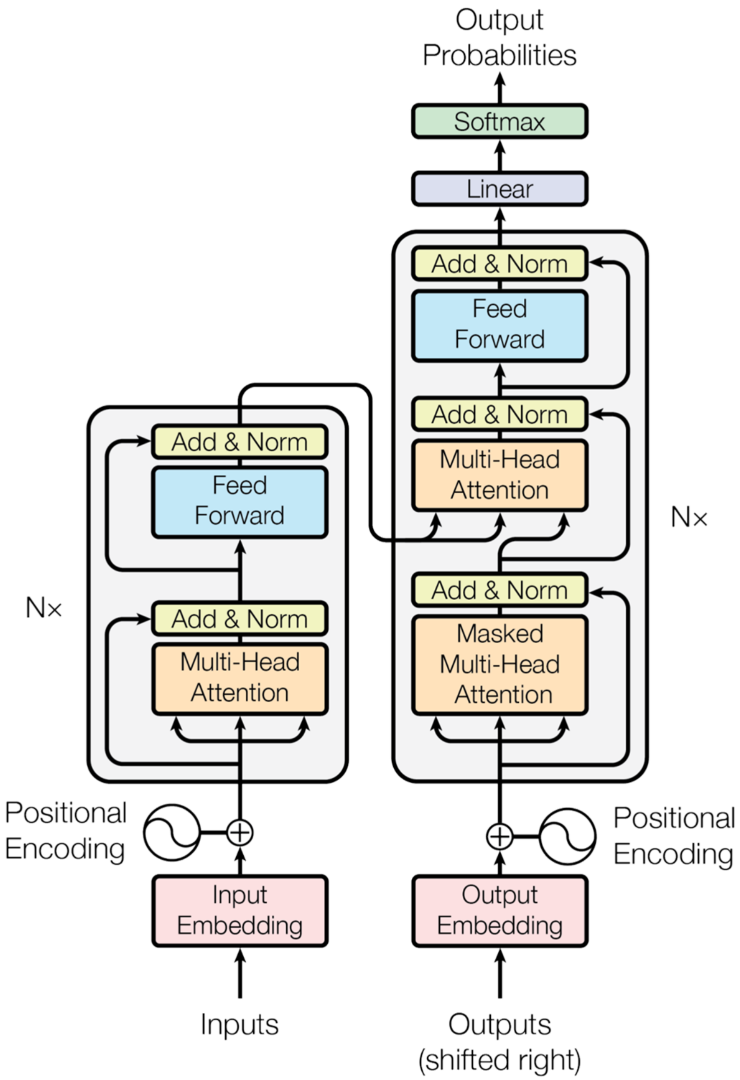


- **Decoders**: produces an output result for the task by **decoding** the information given



The Transformer changed the (NLP) world

Source: A. Vaswani et al., “Attention is all you need”, June 2017



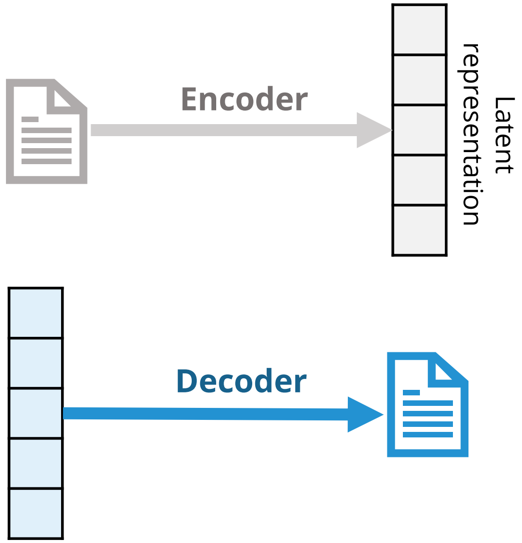
The fundamental structure is familiar

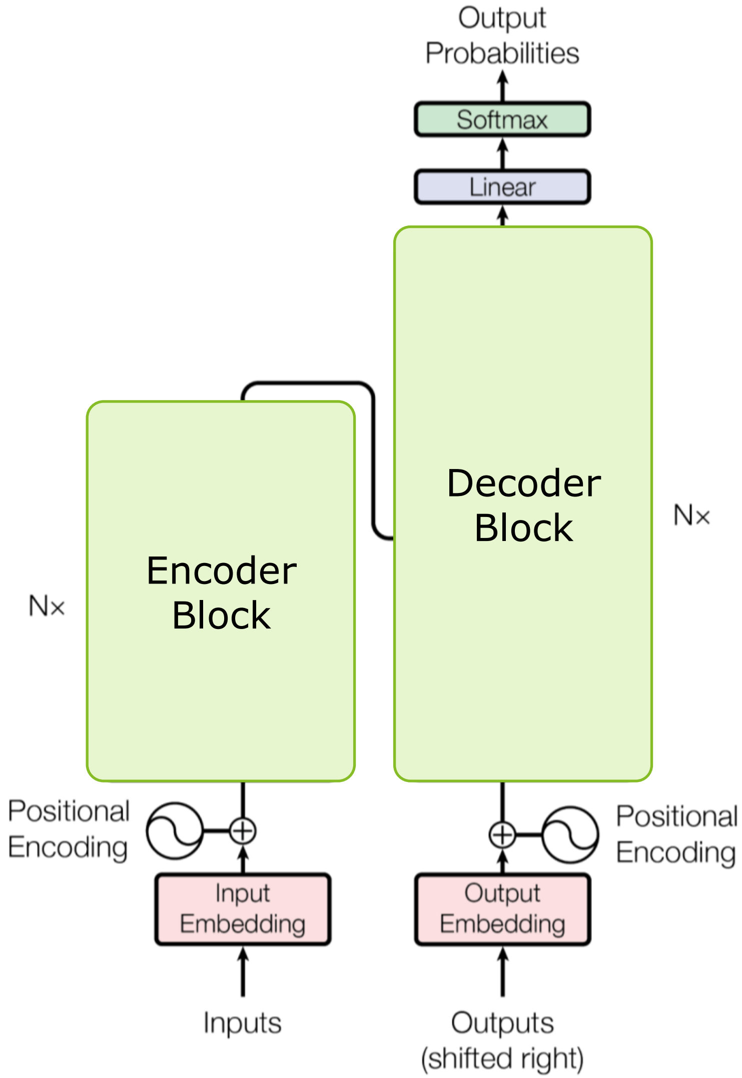
It builds up on the encoder-decoder architecture

Latent representation: Compressed feature representation of data. Called "latent" (*hidden*) as it cannot be directly observed but is inferred from the input.

Encoder: A neural network component that converts input data, into a latent representation, usually a vector of fixed size.

Decoder: The opposite of an Encoder, it converts a latent representation into the desired output form.

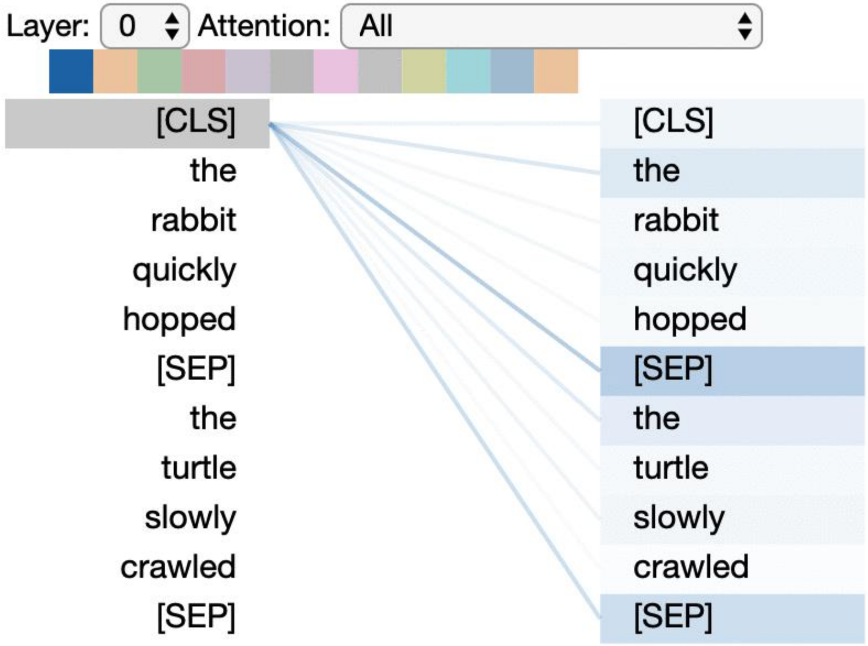




Attention is all you need

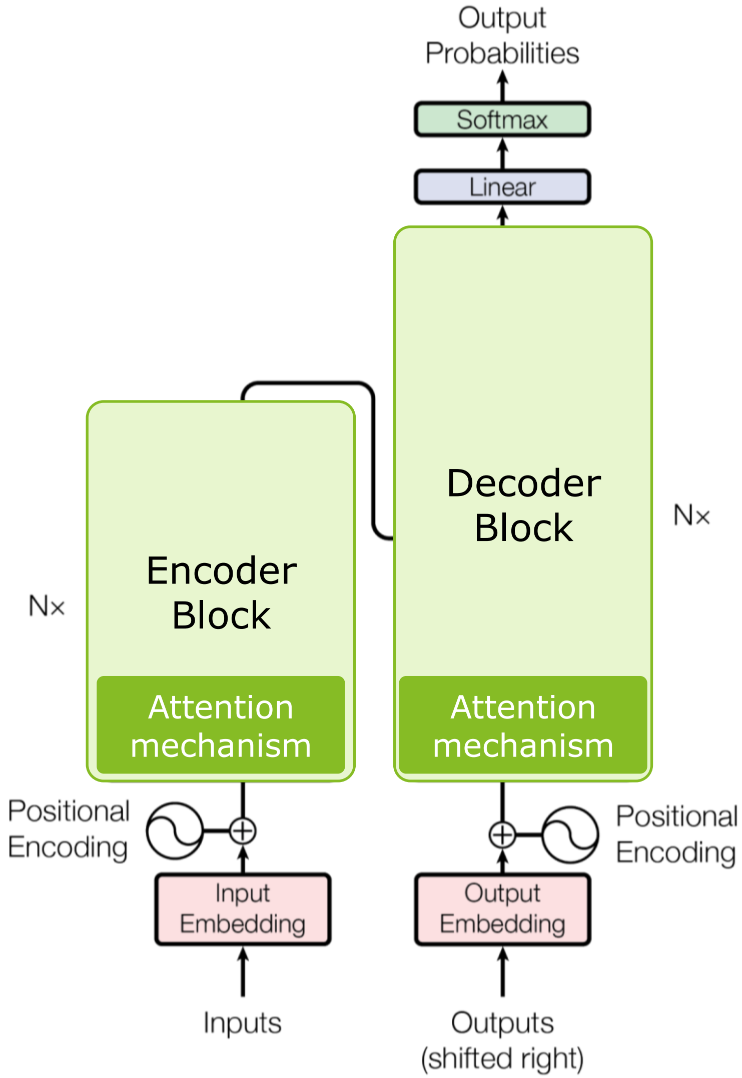
Understanding context in a better way

Attention mechanism: Increase the importance of some parts of the input (*pay attention to these parts*), while diminishing the importance of others for a given token. What parts to focus on and by how much, is learnt during the training process.



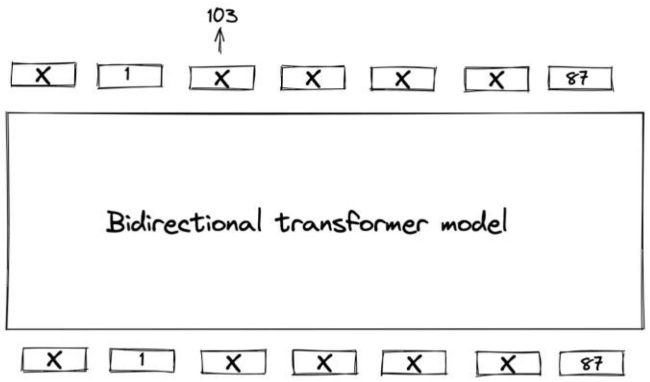
Layer: 0 Attention: All

[CLS]	[CLS]
the	the
rabbit	rabbit
quickly	quickly
hopped	hopped
[SEP]	[SEP]
the	the
turtle	turtle
slowly	slowly
crawled	crawled
[SEP]	[SEP]

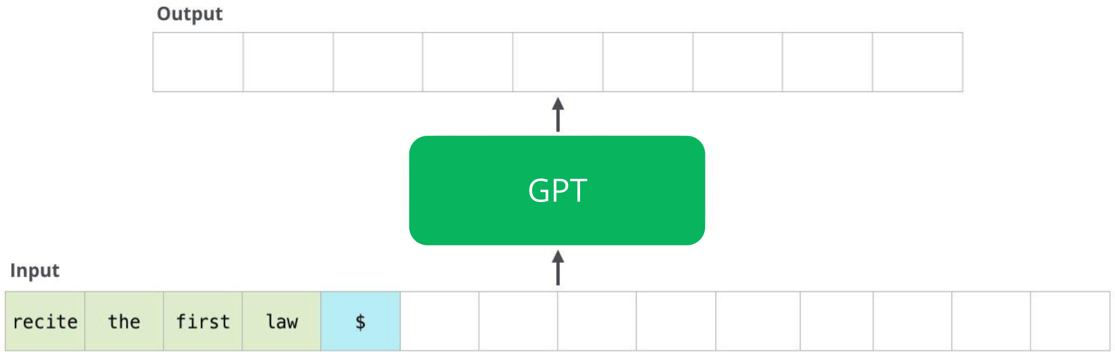


Comparing two different families of LLMs

	BERT – Bidirectional Encoder Representations from Transformers	GPT – Generative Pretrained Transformer
CONTEXT DIRECTION	Bidirectional: Considers **both the left and right contexts** to make predictions.	Autoregressive: Considers only the **context on the left** to make predictions
TASKS	Suitable for **Natural Language Understanding** tasks.	Suitable for **Natural Language Understanding** and **text generation** tasks.
ARCHITECTURE	Contains only the **encoder** blocks of the transformer architecture	Contains only the **decoder** blocks of the transformer architecture
FINE-TUNING	**Necessary** for all task-specific operations.	Only used when **memorisation of private content** or **simulation of writing style/tone of voice** is required.
MODEL DEPLOYMENT	A **task specific model is deployed** for applications	A **general purpose model can be deployed**, using prompts for defining tasks
DELOITTE USE CASES	Classify customer feedback at ING, using RoBERTa	Make internal Knowledge easily accessible using GPT-3 and Search, "QueryGenie" for Shell



Source: B. Christiaens., Potential applications of discrete diffusion models", August 2022

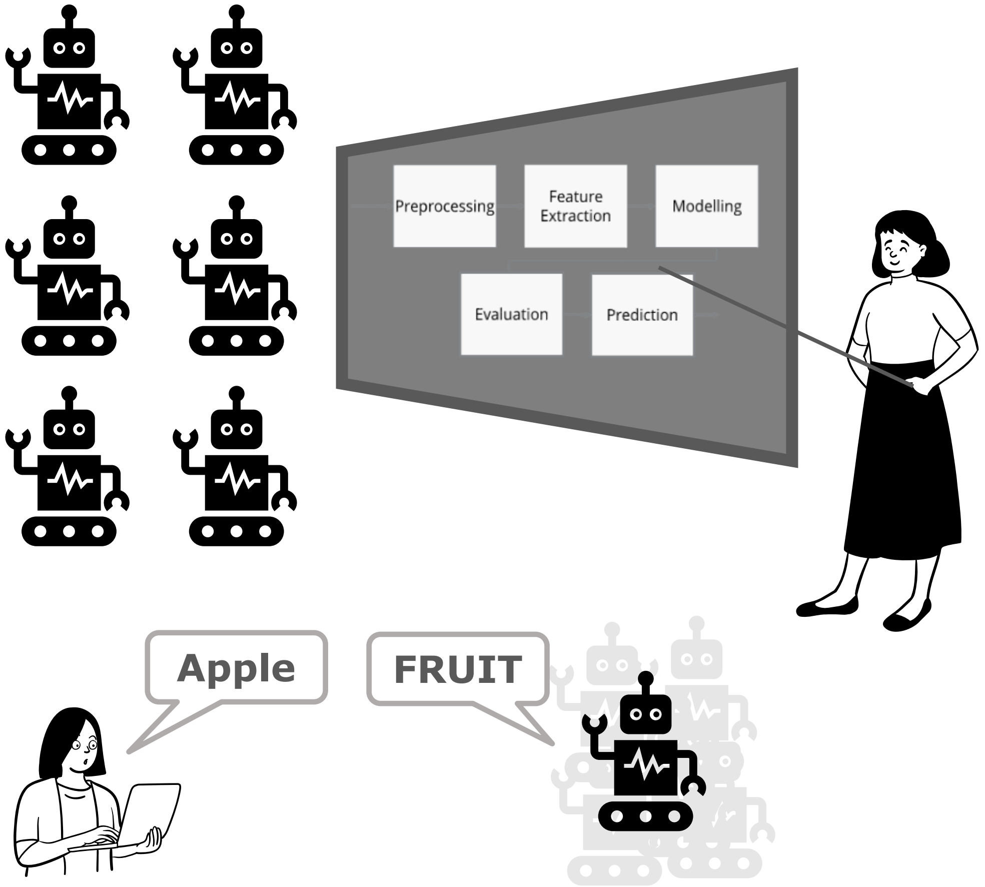


Source: J. Alammar, "The Illustrated GPT-2 (Visualizing Transformer Language Models)", August 2019

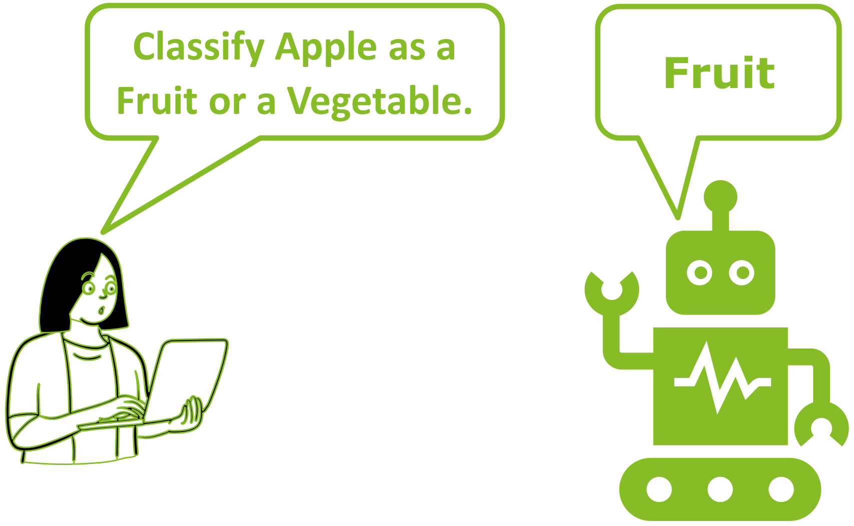
Generative LLMs can take over the entire NLP flow

They can be deployed easily end-to-end, simplifying NLP application development

What NLP looked like before Generative LLMs – a lot of training for every task



What NLP looks like now – just ask!



Prompt engineering

Improving your prompts

Never forget these tips that will help you take your Generative LLM results to a higher level

Be **Concise** and **Specific**

Include your **own Knowledge**

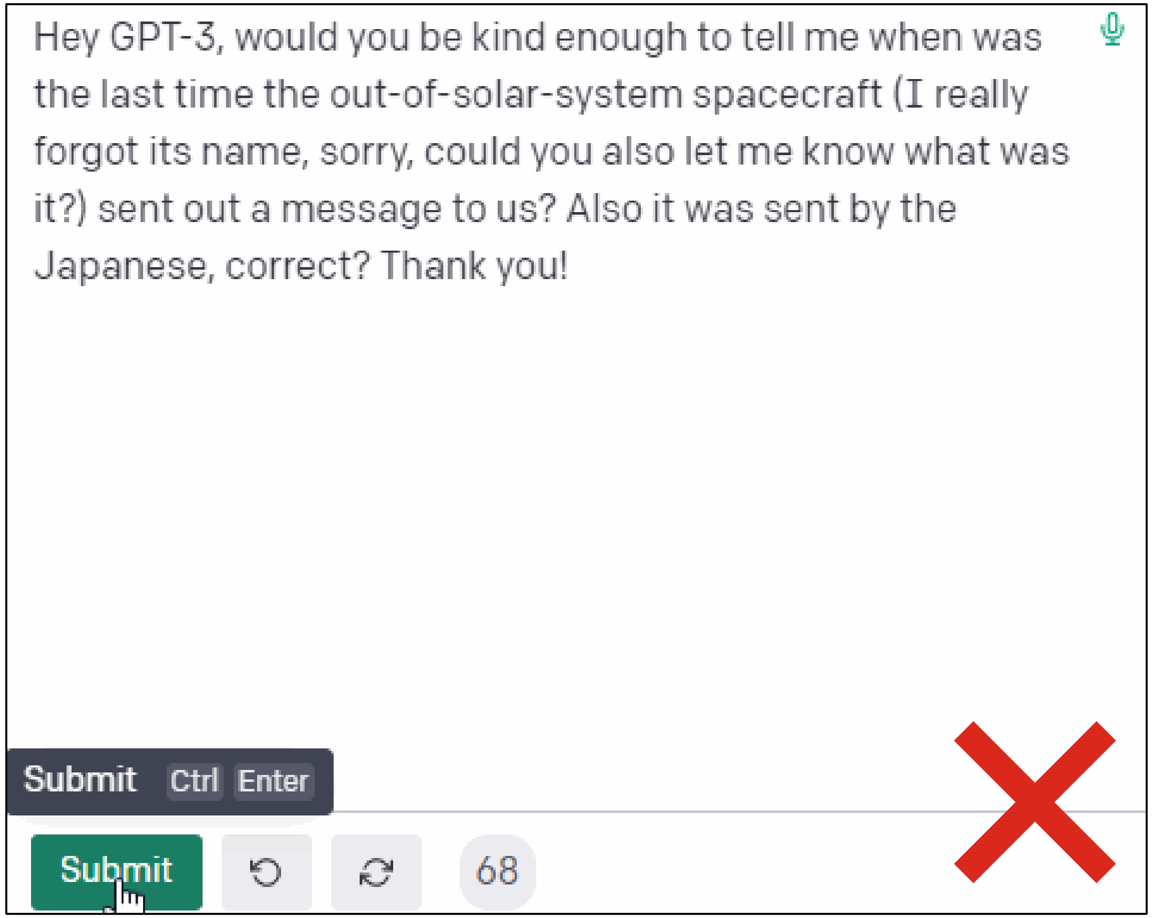
Identify the **Goal**

Create a **Draft Output**

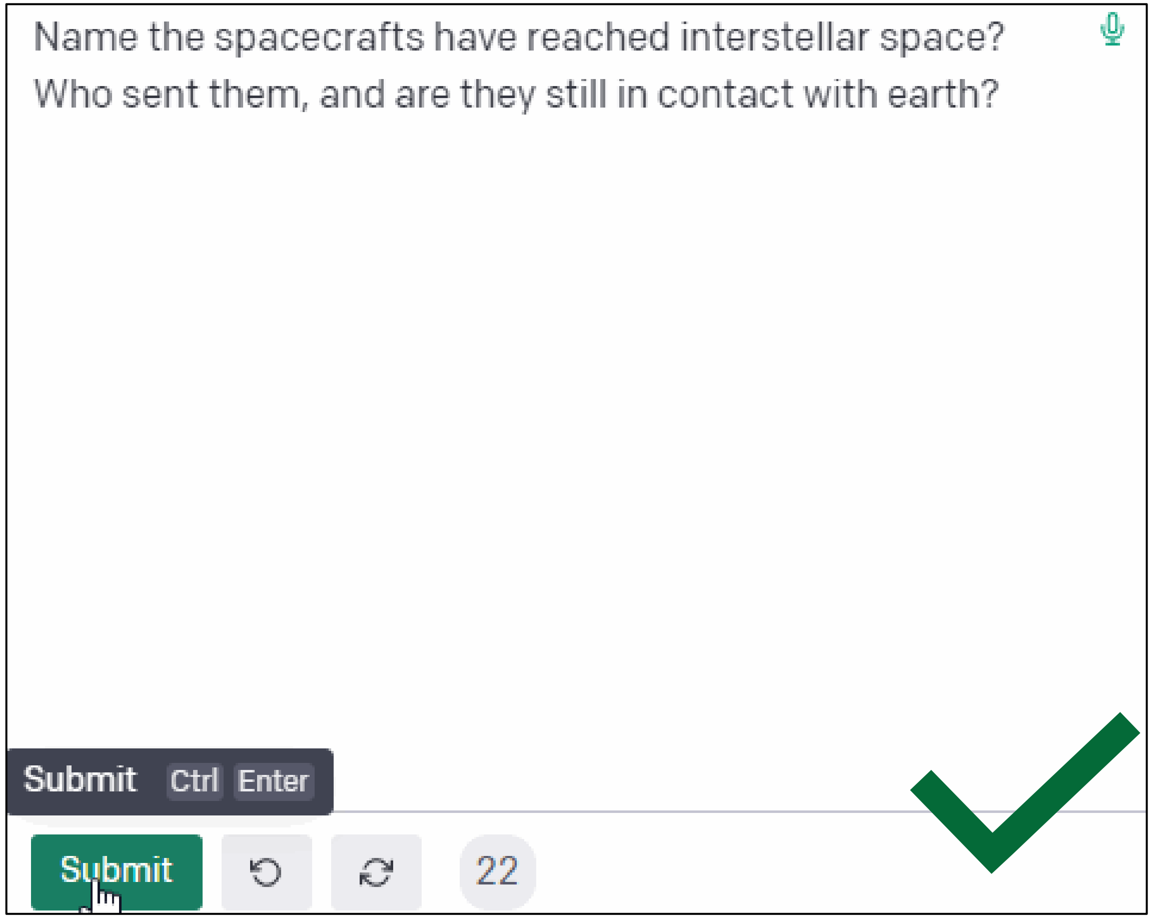
Prime the AI

The Power of Prompting

Be clear and specific to what you expect out of it. Do not provide misleading information that can confuse the model



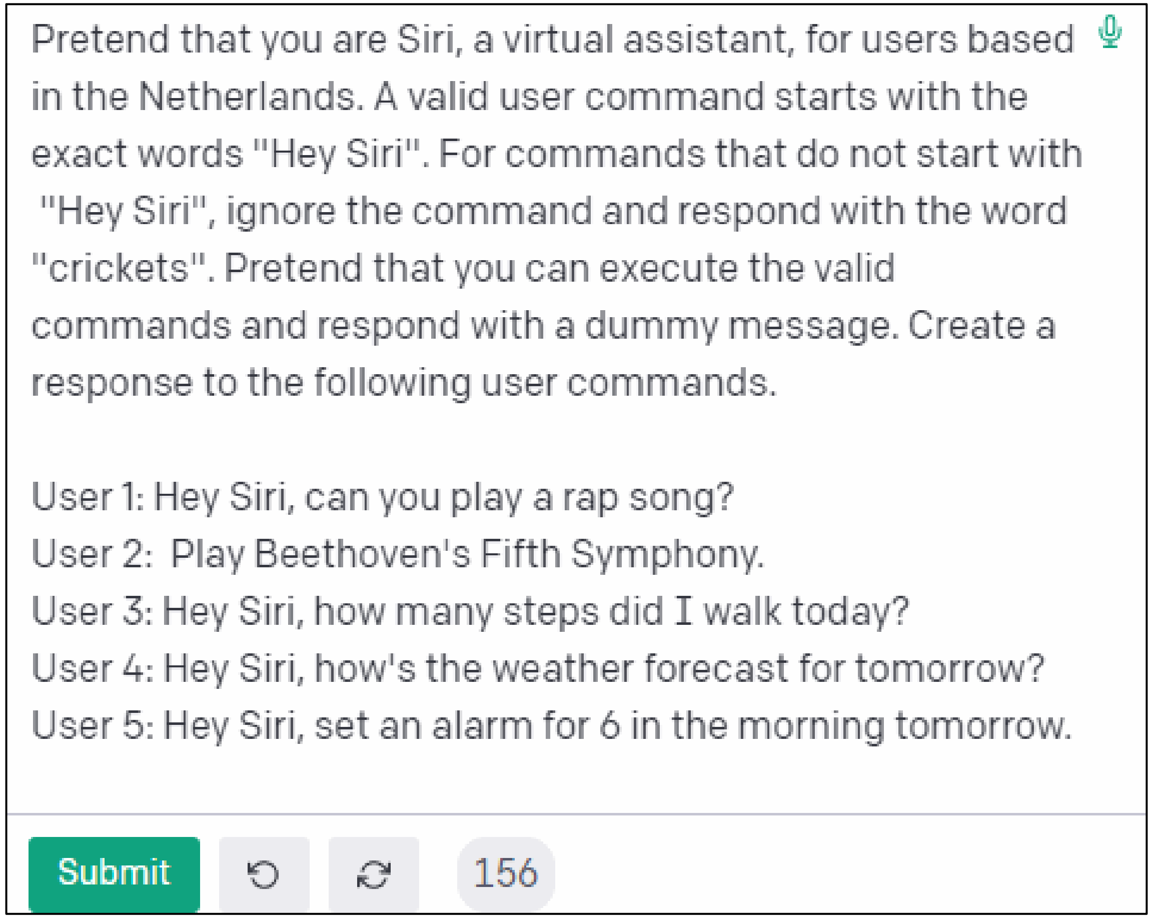
Hey GPT-3, would you be kind enough to tell me when was the last time the out-of-solar-system spacecraft (I really forgot its name, sorry, could you also let me know what was it?) sent out a message to us? Also it was sent by the Japanese, correct? Thank you!



Name the spacecrafts have reached interstellar space? Who sent them, and are they still in contact with earth?

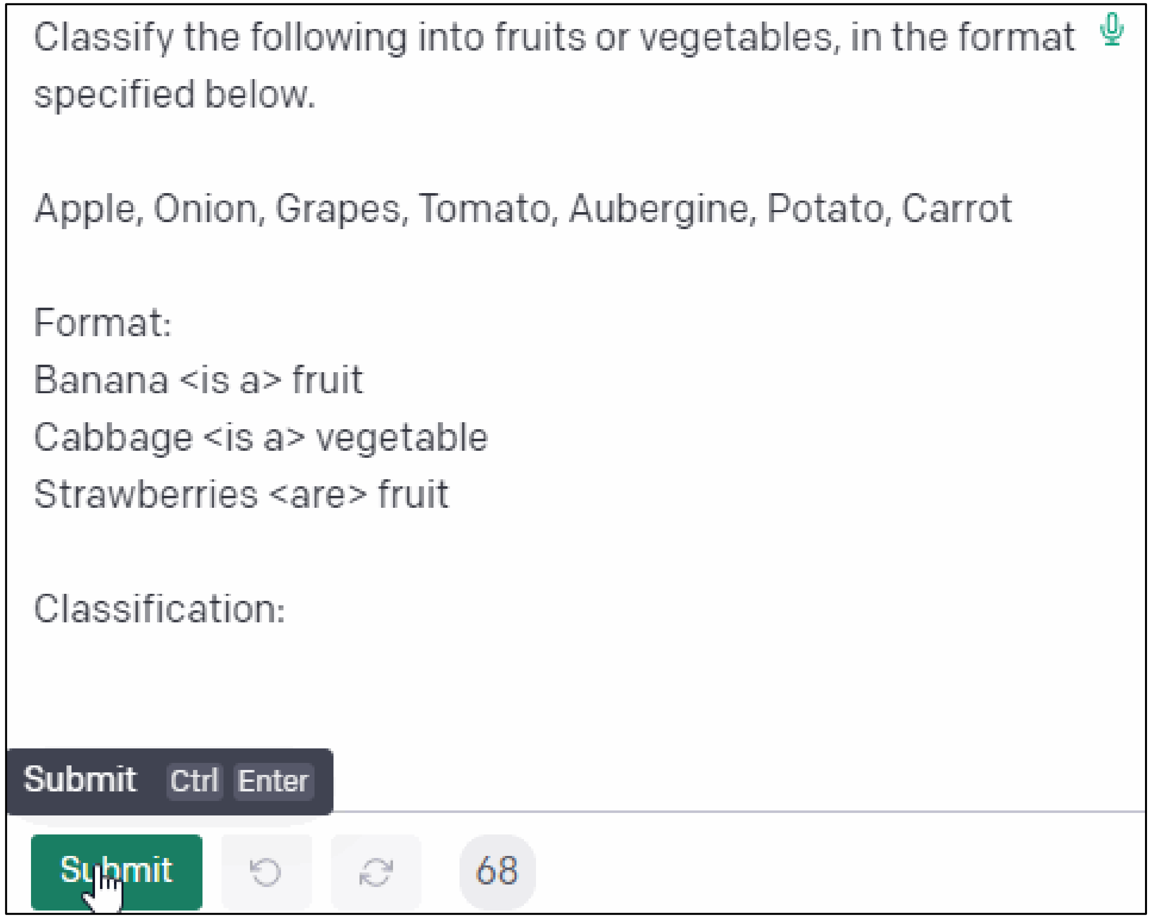
The Power of Prompting

Set the scene, or provide some examples



Pretend that you are Siri, a virtual assistant, for users based in the Netherlands. A valid user command starts with the exact words "Hey Siri". For commands that do not start with "Hey Siri", ignore the command and respond with the word "crickets". Pretend that you can execute the valid commands and respond with a dummy message. Create a response to the following user commands.

User 1: Hey Siri, can you play a rap song?
User 2: Play Beethoven's Fifth Symphony.
User 3: Hey Siri, how many steps did I walk today?
User 4: Hey Siri, how's the weather forecast for tomorrow?
User 5: Hey Siri, set an alarm for 6 in the morning tomorrow.



Classify the following into fruits or vegetables, in the format specified below.

Apple, Onion, Grapes, Tomato, Aubergine, Potato, Carrot

Format:
Banana <is a> fruit
Cabbage <is a> vegetable
Strawberries <are> fruit

Classification:

LLMs in practice

LLMs can perform all NLP “tasks”

And then some more

Information Retrieval
Ranking a list of documents or search results in response to an input query

Sentiment Analysis
Classifying the polarity or “happiness” of a given text

Machine Translation
Translating a sentence in a source language to a different target language

Natural language inference
Determining whether a “hypothesis” is true (entailment), false (contradiction), or undetermined (neutral) given a “premise”

Reading comprehension
Answering questions on a given text passage

Summarization
Producing a shorter version of one or several documents that preserves most of the input’s meaning

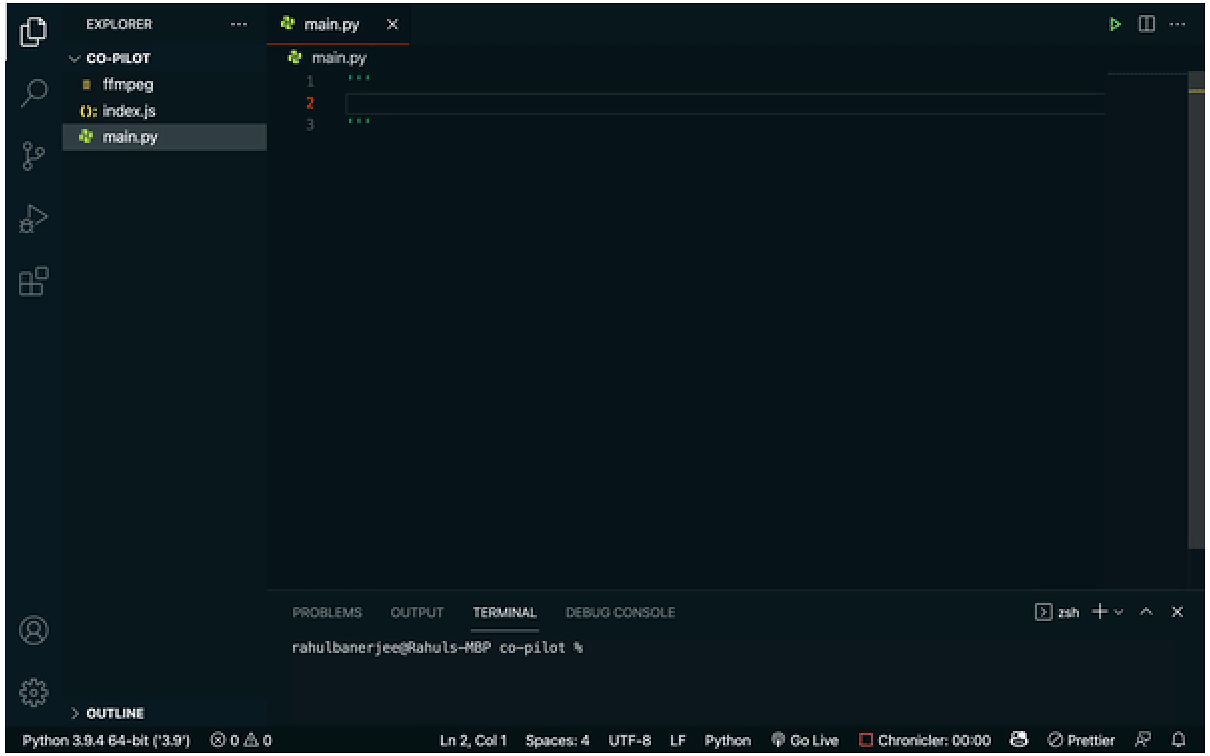
Topic modelling
Discovering the abstract "topics" that occur in a collection of documents

Relation prediction
Recognizing a named relation between two named semantic entities

Generative LLM use cases

Once you have a strong language model, the number of use cases is endless

WRITING CODE



```
'''

'''
```

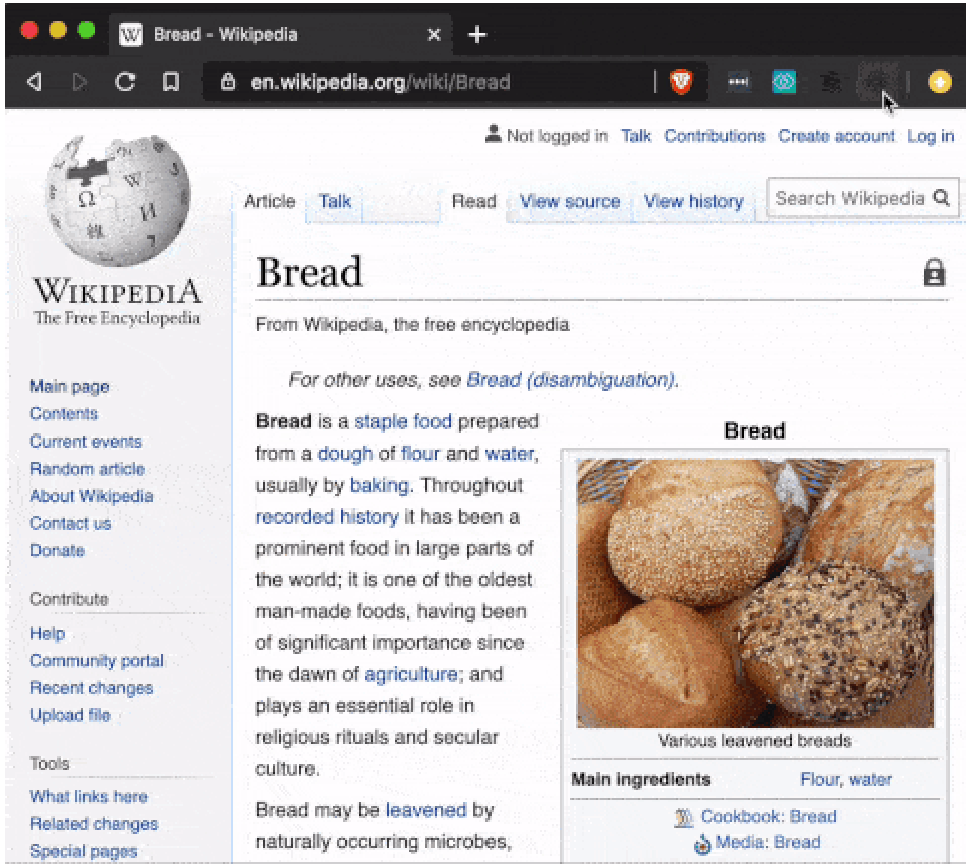
```
rahulbanerjee@Rahuls-MBP co-pilot %
```

Generative LLM use cases

Once you have a strong language model, the number of use cases is endless

WRITING CODE

ENHANCED ON-PAGE SEARCH



Bread

From Wikipedia, the free encyclopedia

For other uses, see Bread (disambiguation).

Bread is a staple food prepared from a dough of flour and water, usually by baking. Throughout recorded history it has been a prominent food in large parts of the world; it is one of the oldest man-made foods, having been of significant importance since the dawn of agriculture; and plays an essential role in religious rituals and secular culture.

Bread may be leavened by naturally occurring microbes,

Bread

Various leavened breads

Main ingredients	Flour, water

Cookbook: Bread
Media: Bread

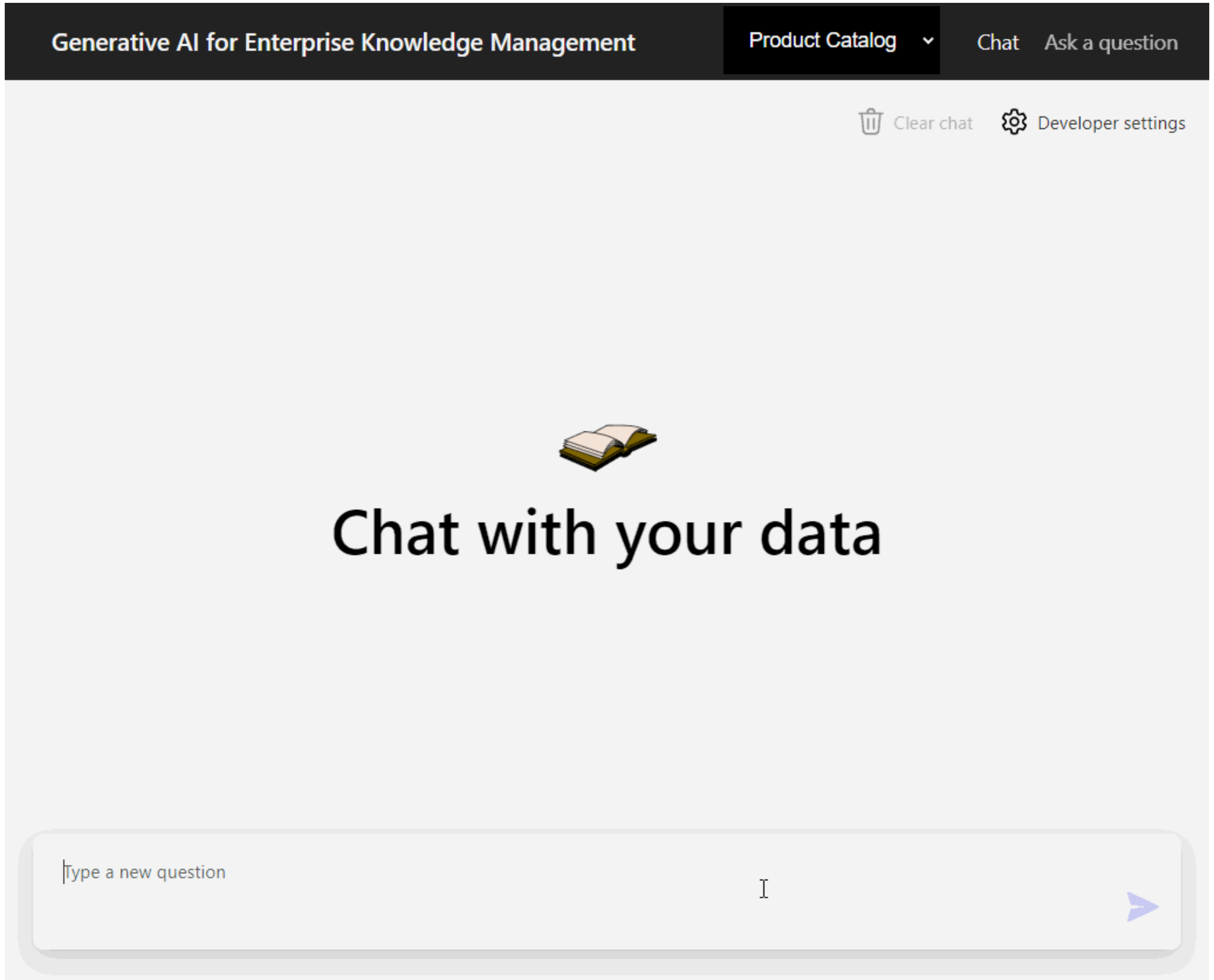
Generative LLM use cases

Once you have a strong language model, the number of use cases is endless

WRITING CODE

ENHANCED ON-PAGE SEARCH

FINDING INTERNAL INFORMATION



Risks and limitations

Limitations

There are some limitations to consider when using LLMs

Cost

The cost per use of sophisticated models is materially significant. Fine-tuning the biggest models is even more expensive, potentially costing thousands of Euros a month.

Ethical Use

Is the AI being used in a manner consistent with the purpose of the overall exercise? For e.g., submitting an AI-generated essay as your own.

Hallucination and Confabulation

Models might output statements with confidence that are factually false. Sources and citations are unavailable for most models.

Latency

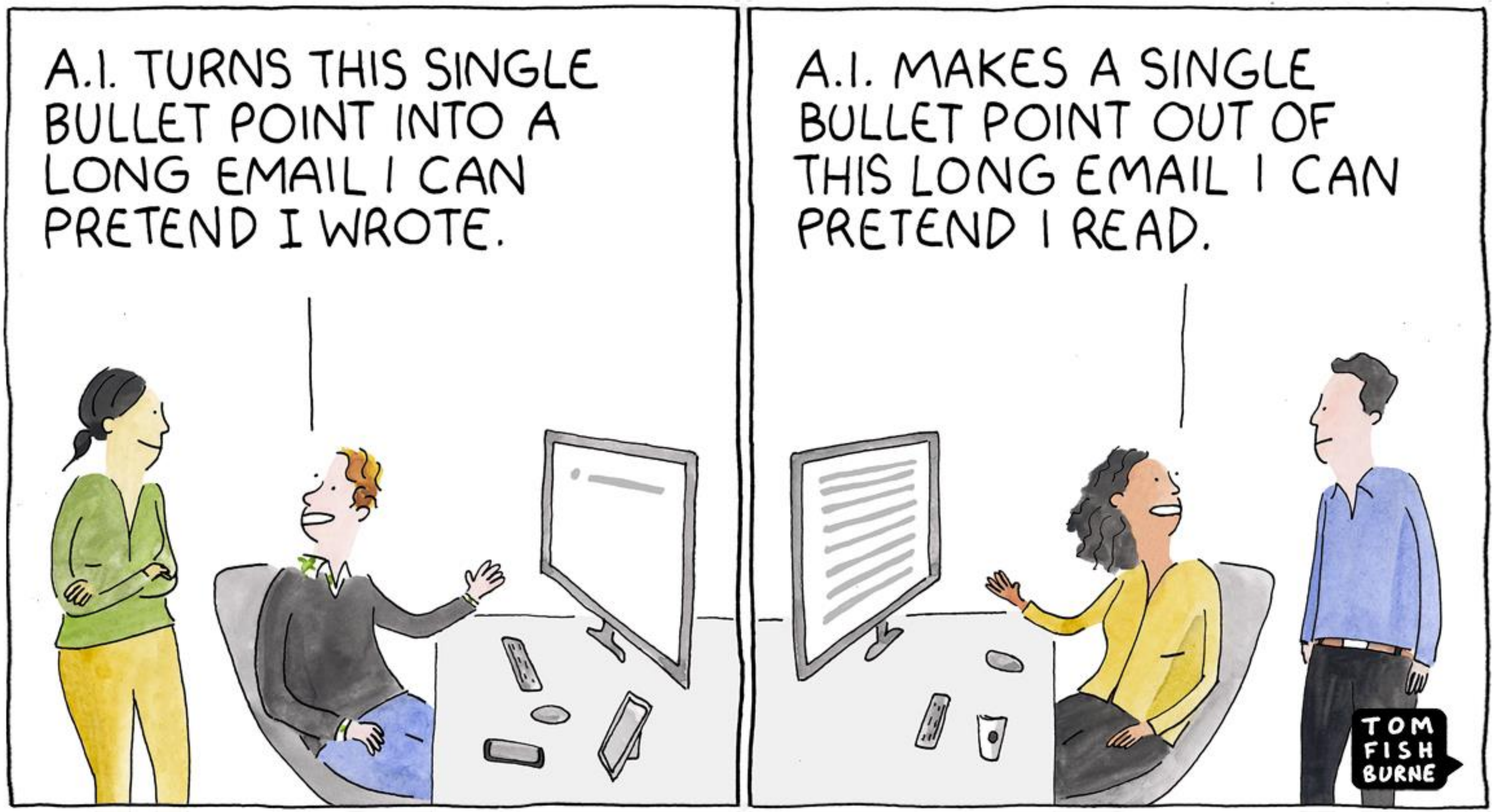
LLMs are comprised of billions of parameters. In theory, the larger the model, the better the output, but the compute time also increases.

Confidentiality & Privacy

Consent for data used (confidential information, personally identifiable information) is necessary, and the residency of data in geo-locations should also be compliant with legal and contractual requirements

IP Protection & Infringement

SaaS-AI companies may use prompt payloads to train future versions of the base model, potentially including confidential data that could expose the user to IP infringement.


A.I. TURNS THIS SINGLE BULLET POINT INTO A LONG EMAIL I CAN PRETEND I WROTE.
A.I. MAKES A SINGLE BULLET POINT OUT OF THIS LONG EMAIL I CAN PRETEND I READ.
TOM FISH BURNE

Tips and tricks

Some tips and tricks when applying NLP in practice

01 **Dutch is more and more supported**
Although English is best supported by NLP models, more and more multi-lingual and Dutch-based models are showing up, while translations are getting better as well

02 **Use available libraries**
Do not reinvent the wheel, reuse available libraries such as SpaCy, NLTK and LangChain

03 **Use existing pipelines**
Use the existing pipelines in e.g. Spacy and LangChain to deliver code that is easy to maintain and reuse

04 **Sufficient data required**
Consider if there's enough data to build a machine learning or deep learning model (rule of thumb: RB: hundreds, ML: thousands, DL: tens of thousands of labels, LLM: it depends)

05 **Develop in an agile way**
Start with the simplest approach that could possibly work (e.g. rule-based) and iterate on bigger and bigger models