Natural Language Processing

About me

- Graduated Koźmiński University in Finance&Accounting
- Self-taught Data Scientist
- Currently working as Data Scientist at Samba TV
- Developing Real Estate comparison engine Resider.pl as a side project
- Focus on similarity exploration, pricing and working with non-homogenous data

Course Goals

What to expect?

- Exploring NLP use cases and challenges
- Introduction to basic NLP theory
- Introduction to Large Language Models
- Learning how to build first models with popular frameworks such as spacy
- Improving Python skills
- Group assigment to explore one of NLP techniques at home
- Building foundations and a roadmap for further learning

What not to expect?

- Extensive NLP and Machine Learning Theory lectures
- Learning to code in Python from scratch
- Becoming NLP expert in 20 hrs
- Learning from slides alone

Course Agenda

- **W1 – Introduction to NLP**
- **W2 – NLP Theory fundamentals**
- **W3 – Named Entity Recognition and Classification**
- **W4 – Introduction to Large Language Models**
- **W5 – Course summary**

W1 Agenda

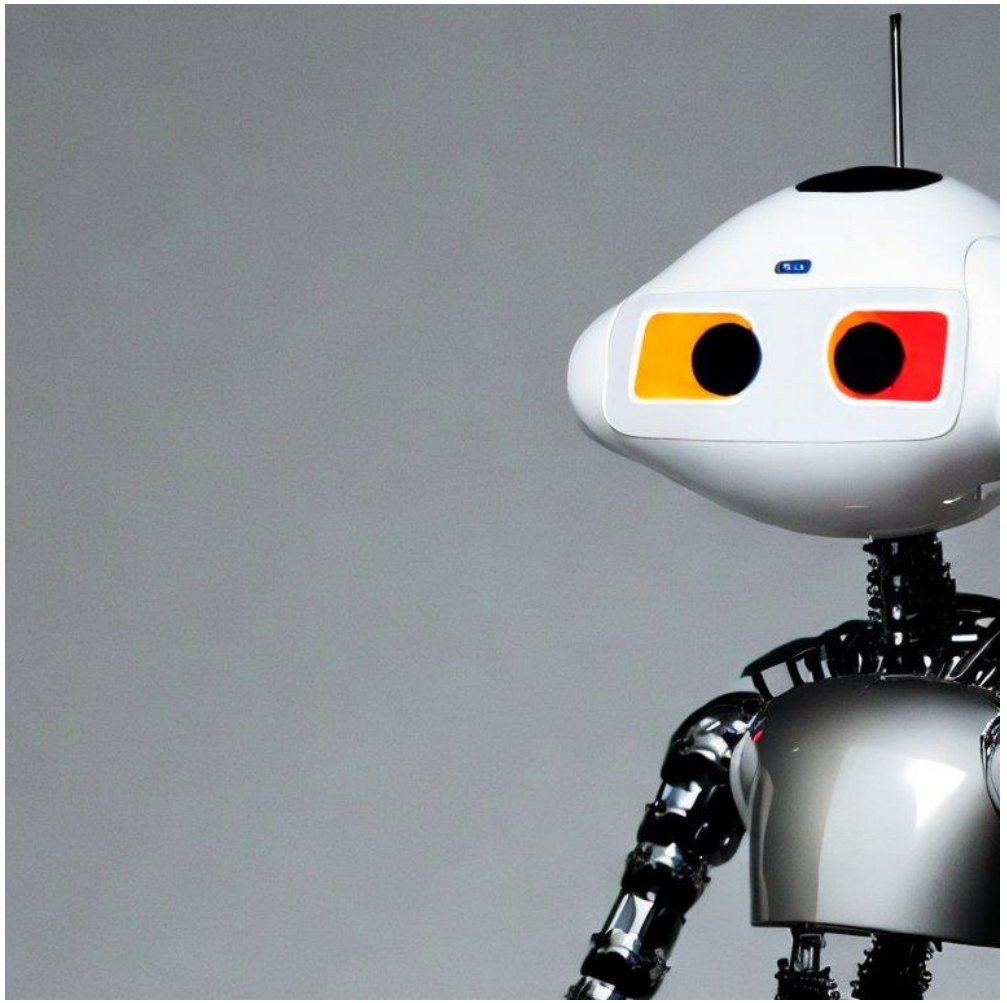
- What is NLP?
- Between the Hype and everyday use cases
- Most common NLP use cases
- Python kick-off assignment
- NLP in business – key opportunities and pitfalls

What is NLP?

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data. Natural language processing tasks include tasks such as sentiment analysis, part-of-speech tagging, and named entity recognition. These tasks are typically based on machine learning algorithms that are trained on large amounts of data in order to learn the rules of the language.


Can you image Artificial Intelligence without a conversation?

- As speech is our primary form of communication we require this ability from an Artificial Intelligence
- For decades you could communicate with a computer only by knowing how to code
- Recent breakthroughs in NLP help computers grasp the human language, which increases their accessibility

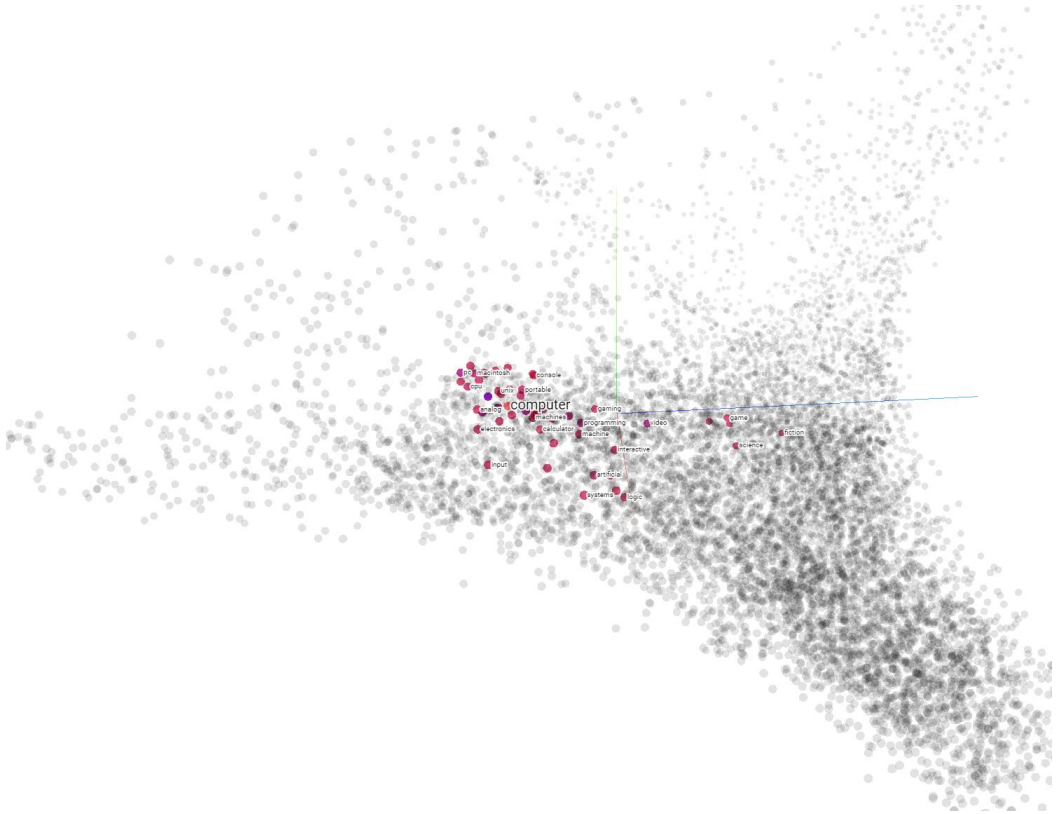


Source: Stable Diffusion

At the end of the day a computer can only understand vectors

- Machines rely on numbers and are not able to understand characters and words
- Converting words and sentences to vectors is the foundation of NLP
- Even current State-of-the-art solutions, which often reach near human performance are still based on vectors
- Enormous increase in computing power over the last few years fuelled rapid development of NLP in last decade

Word2Vec vectors



How do machines learn to convert language to vectors?

- Computers learn to „understand" language by imitating humans – similar to a child learning new words
- Predicting the most probable word in a sequence is how transformer-based NLP models are trained to convert language to vectors
- This ability to learn requires complex architecture – GPT3 has about 175B parameters, and GTP4 will be 500X larger

GPT3 model architecture

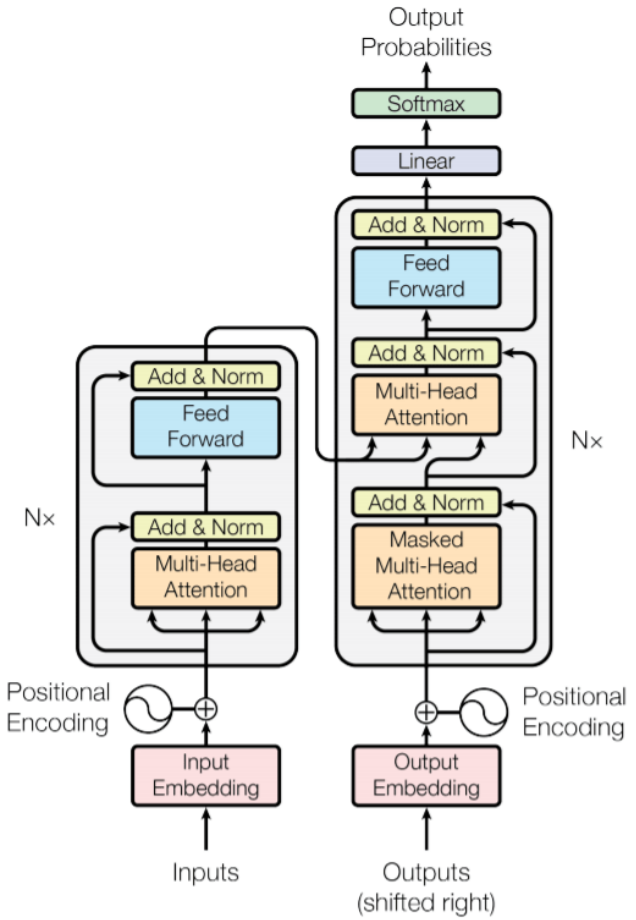


Figure 1: The Transformer - model architecture.

Between the Hype and everyday use cases?

What are your first thoughts when you hear Natural Language Processing?

Photo by Kelly Sikkema on Unsplash

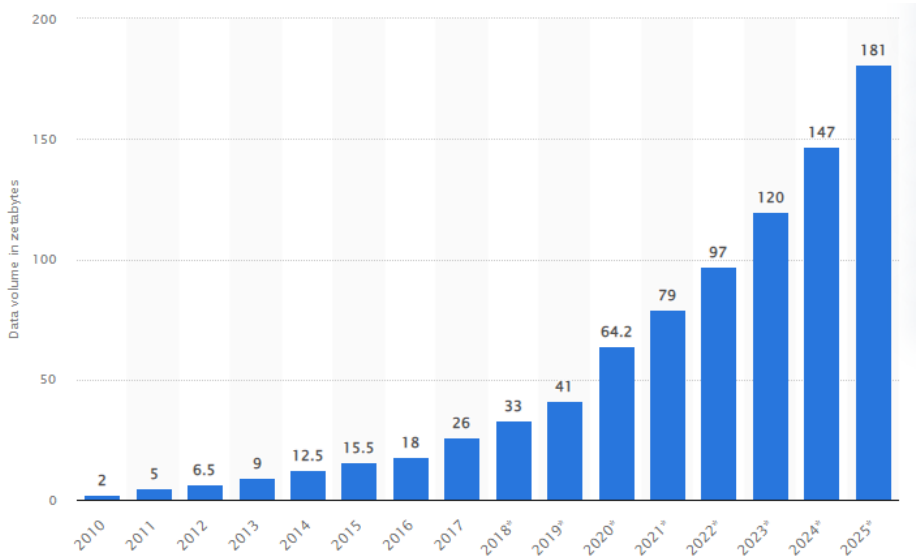
What are the NLP use cases you can think of?

Photo by Kelly Sikkema on Unsplash

Most Big Data is not structured, that's why we need NLP

- Data extraction is the most common use case of NLP as it is easiest to implement with existing analytics
- In most businesses NLP main goal is to extract value from the chaos of text data like emails, chats, or descriptions
- In more advanced use cases NLP also helps automating repeatable tasks such as performed by customer service bots

Data created, captured, copied, and consumed worldwide [zettabytes]



Source: Statista

Majority of this data is either text, audio, video or images. NLP helps us extract information from both text and audio

Big Tech gets closer to human performance...

Translation

Google Translate

Speech recognition

Siri

Chatbots

ChatGPT

Creating NLP models, which are hard to distinguish from humans requires enourmous resources in terms of data, computing power and engineering.

Majority of tech breakthrough come from Big-Tech companies. Fortunately they are often open-sourced like Google BERT, which accelerates further research and enables leveraging them in different use-cases.

.. while most companies struggle with extracting personal data from a chat

- Most complex NLP use cases resembling Artificial Intelligence get most of the media attention
- In real life business most value is brought by much simpler solutions
- Well-written REGEX can often bring more value than building a model from scratch
- Extracting structured data from text is still the most common use-case

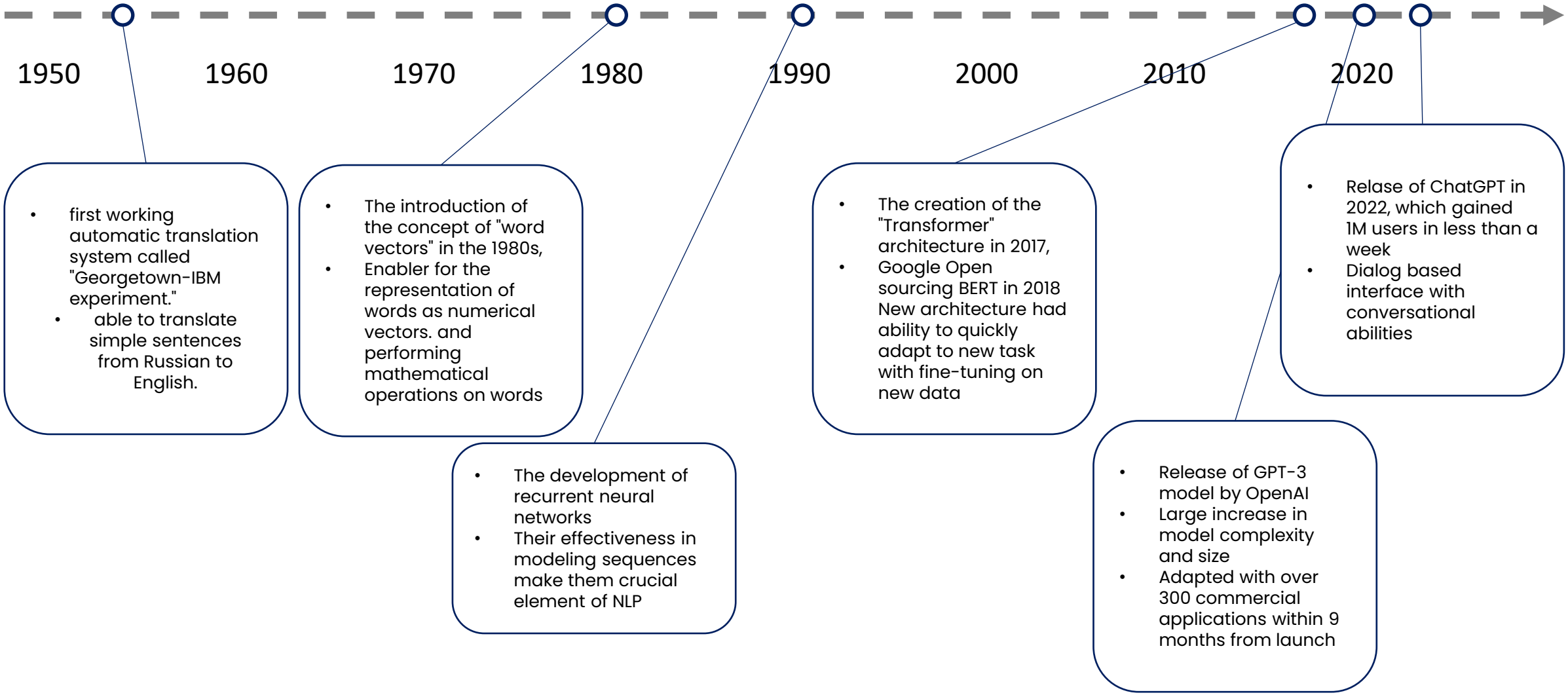
Models performing simple, single-purpose actions are the backbone of business use cases

Popular NLP examples

- Find name and phone number in a chat
- Find items with a similar description
- Classify if the review is positive or negative

More complex systems often consist of multiple simpler, single-purpose models. This is also the safest way to start with NLP implementation

NLP breakthroughs timeline



Most common NLP use cases

Sentiment analysis

- Common NLP algorithm which aims to identify subjective information from text
- Usefull for learning purposes, as opinion related words have clear positive/negative vector interpration
- Use cases ranging from customer reviews to trading bots
- Can focus on negative vs positive or focus on more specific emotions or opinions
- Sarcasm, irony is most challenging to handle in Sentiment Analysis cases

„The Hathaway Effect" – sentyment analysis gone wrong or a statistical anomaly?

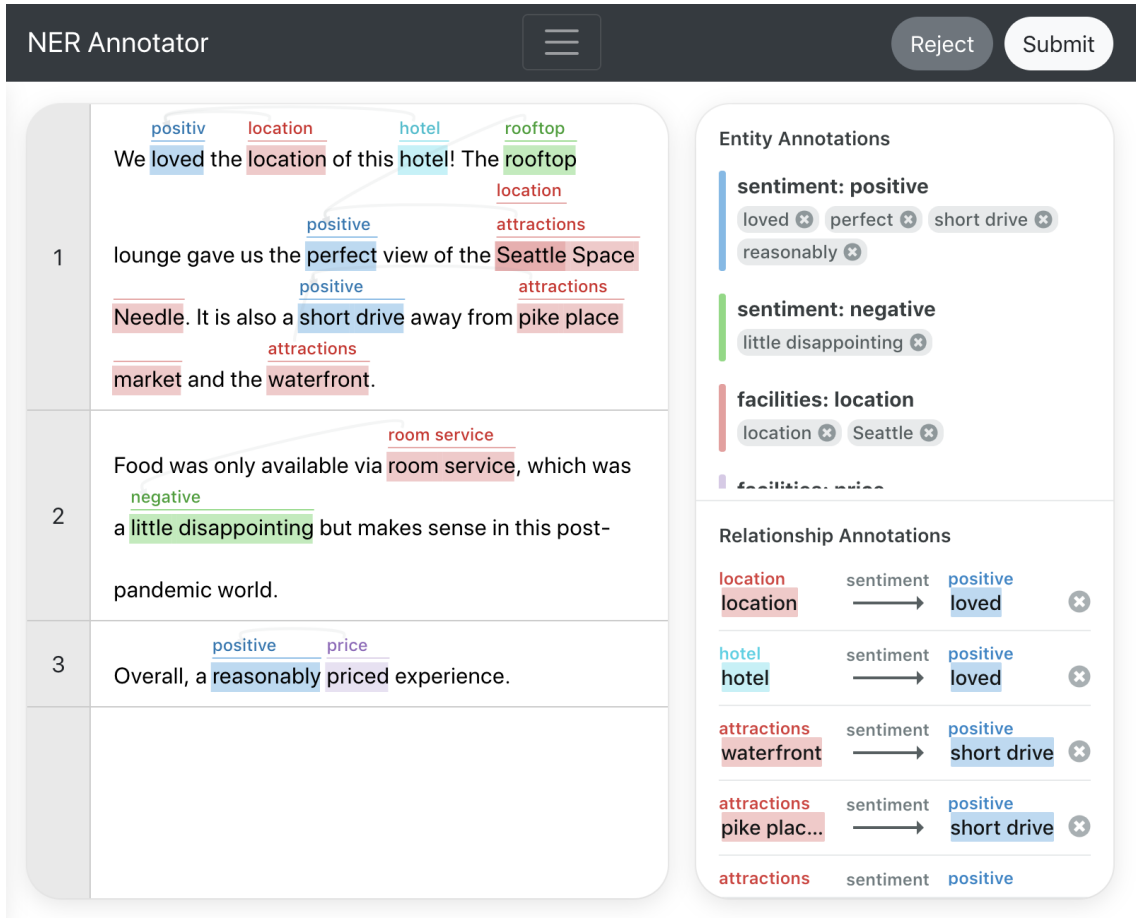
In 2011 Dan Mirvish spotted high correlation between Anne Hathaway's movie premieres and spikes in Berkshire Hathway stock prices.

His theory stated that automated robotic trading, which relies heavily on setiment analysis could pick up all the occurences of positive sentiment related to Hathaway.

Although not proven statistically the theory is probable and was investigated by Nobel Price winner Paul Krugman and discussed at Bekshire Hathaway shareholders meetings

Named Entity Recognition

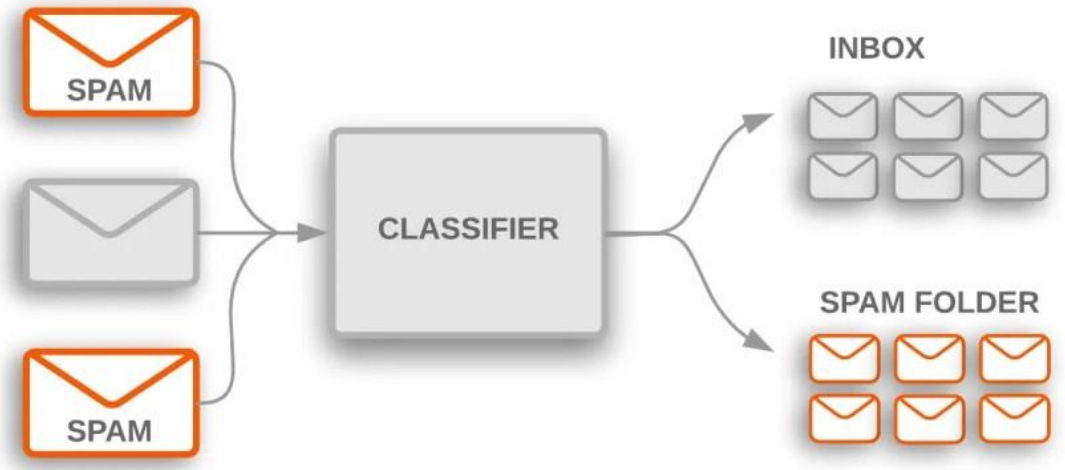
- NLP technique used to identify and classify named entities in text
- Named entities can include names, dates, job titles, brands or any other entity types, which are easily classifiable by humans within a few words
- Easy to integrate with current analytics and used for data enrichment as it extracts structured data from text
- One of the most explainable NLP techniques
- NER data annotation is time consuming and challenging



Source: AWS Sage Maker tutorial

Text classification

- NLP technique used to classify sentences or longer pieces of text
- Use cases range from Spam or hate speech detection to article topic classification
- Data labeling is reasonably easy and can rely on customer feedback loop as seen in Gmail spam classifier

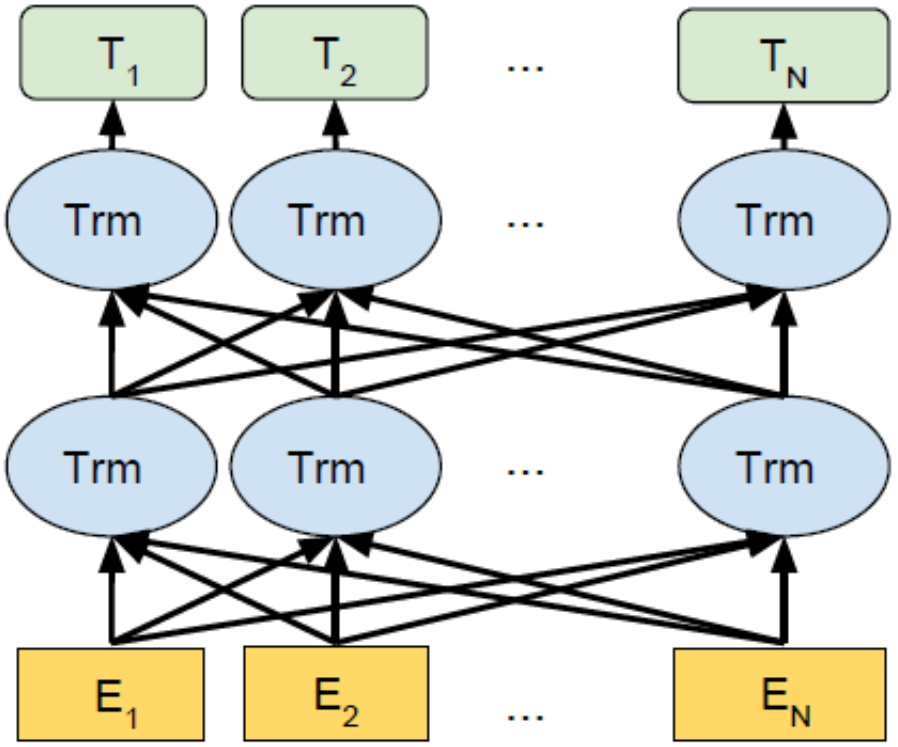


Source: developers.google.com

Language Translation

- One of first large-scale use cases of NLP, which has the benefit of reducing language barriers online
- Efforts by Google resulted in the creation of BERT and accelerated Transformer based NLP development
- Currently there are dozens of different translation tools as development in NLP architectures and easy access to language corpus online make training easier

Bidirectorial approach is crucial translation as different languages vary in syntax and sentence order



Source: Devlin et al. 2019

Python Warm-up Twitter sentyment analysis

jupyter Python-warm-up-setiment-analysis Last Checkpoint: kilka sekund temu (unsaved changes)

Logout

File Edit View Insert Cell Kernel Navigate Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

1 Read data and scores dict

In [237]:
```
df = pd.read_csv("data/warm_up_data.csv")
```

In [238]:
```
with open('data/AFINN-111-scores.json', 'r') as fp:
    scores_dict = json.load(fp)
```

In [239]:
```
df.head()
```

Out[239]:

	tweet_id	airline_sentiment	text	airline
0	570301031407624196	negative	@VirginAmerica it's really aggressive to blast...	Virgin America
1	570300817074462722	negative	@VirginAmerica and it's a really big bad thing...	Virgin America
2	570300767074181121	negative	@VirginAmerica seriously would pay $30 a figh...	Virgin America
3	570300248553349120	neutral	@VirginAmerica Really missed a prime opportuni...	Virgin America
4	570295459631263746	positive	@VirginAmerica it was amazing, and arrived an ...	Virgin America

2 Data preparation

Convert airline sentiment column to numeric class

In [240]:
```
setiment_class_dict = {'negative':-1, 'neutral':0, 'positive':1}
```

In [241]:
```
df["sentiment_class_true"] = df.airline_sentiment.apply(lambda x: setiment_class_dict.get(x))
```

3 Assign text sentiment score based on scores dict

In [242]:
```
def assign_score(text):
    score = 0
    for word in text.split():
        score+=scores.get(word.lower(),0)
    return score
```

In [243]:
```
df["sentiment_score_hat"] = df.text.apply(assign_score)
```

3.1 Split sentiment scores to class

In [244]:
```
def classify_sentiment(score_col, negative_th, positive_th):
    bins = [-100,negative_th, positive_th, 100]
    labels = [-1,0,1]

    score_class = pd.cut(score_col,bins = bins, labels = labels).astype(int)
    return score_class
```

In [245]:
```
df.sentiment_score_hat.describe()
```

Out[245]:
```
count    9489.000000
mean        0.284540
std         2.853243
min       -13.000000
25%        -2.000000
50%         0.000000
75%         2.000000
max        16.000000
Name: sentiment_score_hat, dtype: float64
```

In [246]:
```
df["sentiment_class_hat"] = classify_sentiment(df.sentiment_score_hat, -1, 1)
```

In [247]:
```
df["correct_classification"] = df.sentiment_class_hat == df.sentiment_class_true
```

GH link

- Setup a folder for the whole course
- Clone course materials: https://github.com/Jan-Majewski/ALK-NLP-course.git
- Open Jupyter Notebook
- Open Python-warm-up-setiment-analysis.ipynb BLANK version
- Follow instructions within the notebook

NLP in business – key opportunities and pitfalls

Non-technical people have a tendency to overestimate NLP models „intelligence”

Pros

- Opportunity to shine, as even easy NER seems as a model that „understands”
- A lot of low hanging fruits, which cannot be extracted even with unlimited hours of Data Analysts
- If NLP is not yet widely used even basic knowledge and out-of-the-box solutions can bring visible value

Cons

- As a data professional you need to control their enthusiasm
- Good communication is essential to avoid overpromising
- Businesses might expect that NLP will handle data quality issues better than humans
- LLMs like Chat GPT raised the bar in terms of expectations but are still too slow and too expensive for majority of everyday use cases

Majority of models you will work on will be far from „understanding” anything

At their core NLP models work as any other model

- They optimize for a given loss function
- They can find common relations
- Usually human performance is still far way off
- They need good data to learn from
- If we want to create an appearance of logical reasoning, we will probably need multiple steps guiding the logic

Can we compare similar items within same brand based on description?

What steps do we actually need to Conduct?

1. We don’t have structured data so we need to train brand detection NER
2. Use NLP to create meaningful vector and define similarity – should it be price or user interest?
3. Implement Nearest Neighbor search algorithm, choose distance metric
4. Evaluate model performance, do we even have annotated data?

NLP accessibility exploded over last few years

- While a few years back any real-life implementation needed a strong research team and tonnes of data
- Now out-of-the-box solutions like Spacy make it 10x easier
- Multiple models on Hugging Face with ability to easily retrain them for domain-specific data
- Less popular languages such as polish finally catch-up in performance to English or Chinese

Loading HF model takes 5 lines of code

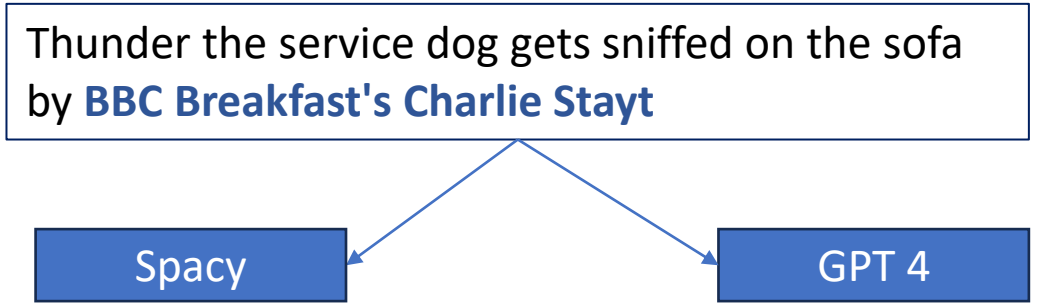
```
>>> from transformers import HerbertTokenizer, RobertaModel

>>> tokenizer = HerbertTokenizer.from_pretrained("allegro/herbert-klej-cased-tokenizer-v1")
>>> model = RobertaModel.from_pretrained("allegro/herbert-klej-cased-v1")

>>> encoded_input = tokenizer.encode("Kto ma lepszą sztukę, ma lepszy rząd – to jasne.", return_tensors='
>>> outputs = model(encoded_input)

>>> # HerBERT can also be loaded using AutoTokenizer and AutoModel:
>>> import torch
>>> from transformers import AutoModel, AutoTokenizer

>>> tokenizer = AutoTokenizer.from_pretrained("allegro/herbert-klej-cased-tokenizer-v1")
>>> model = AutoModel.from_pretrained("allegro/herbert-klej-cased-v1")
```

Learning how to leverage these resources is one of key goals of this course

Do we need other NLP methods if LLMs can handle the majority of tasks?

- LLMs are very large and using them for simple tasks is a complete overkill
- They are slower and more expensive than streamlined, task-specific models
- In many cases precise prompt engineering takes too many tokens to be cost efficient
- LLMs can be used to speed up training data for smaller models

Find named entities within text



- Response: 6ms
- Costs: 6ms of compute time (next to zero)

- Response: 2380 ms (400x longer)
- Costs: around 1 cent