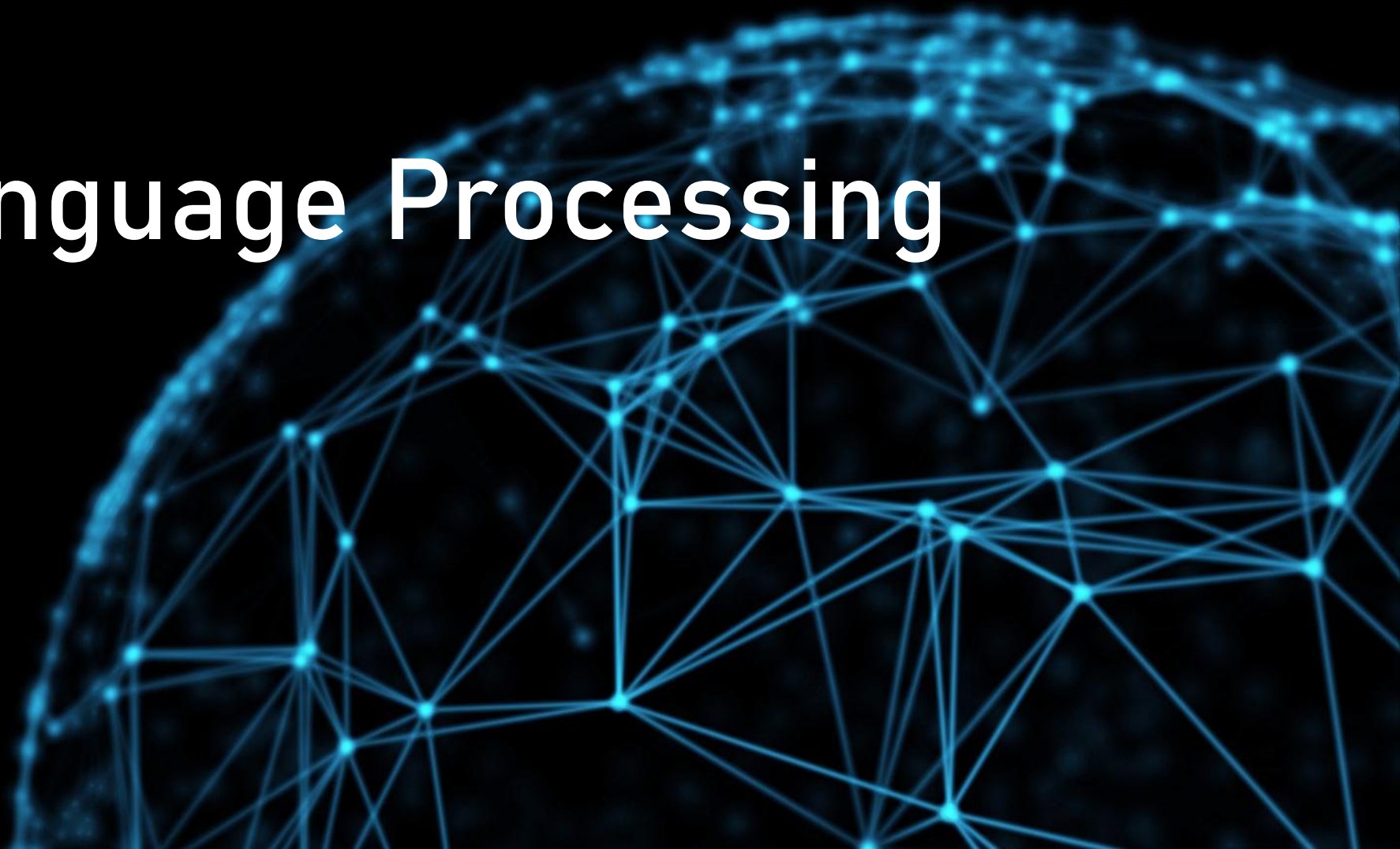
Auxiliary AI GmbH

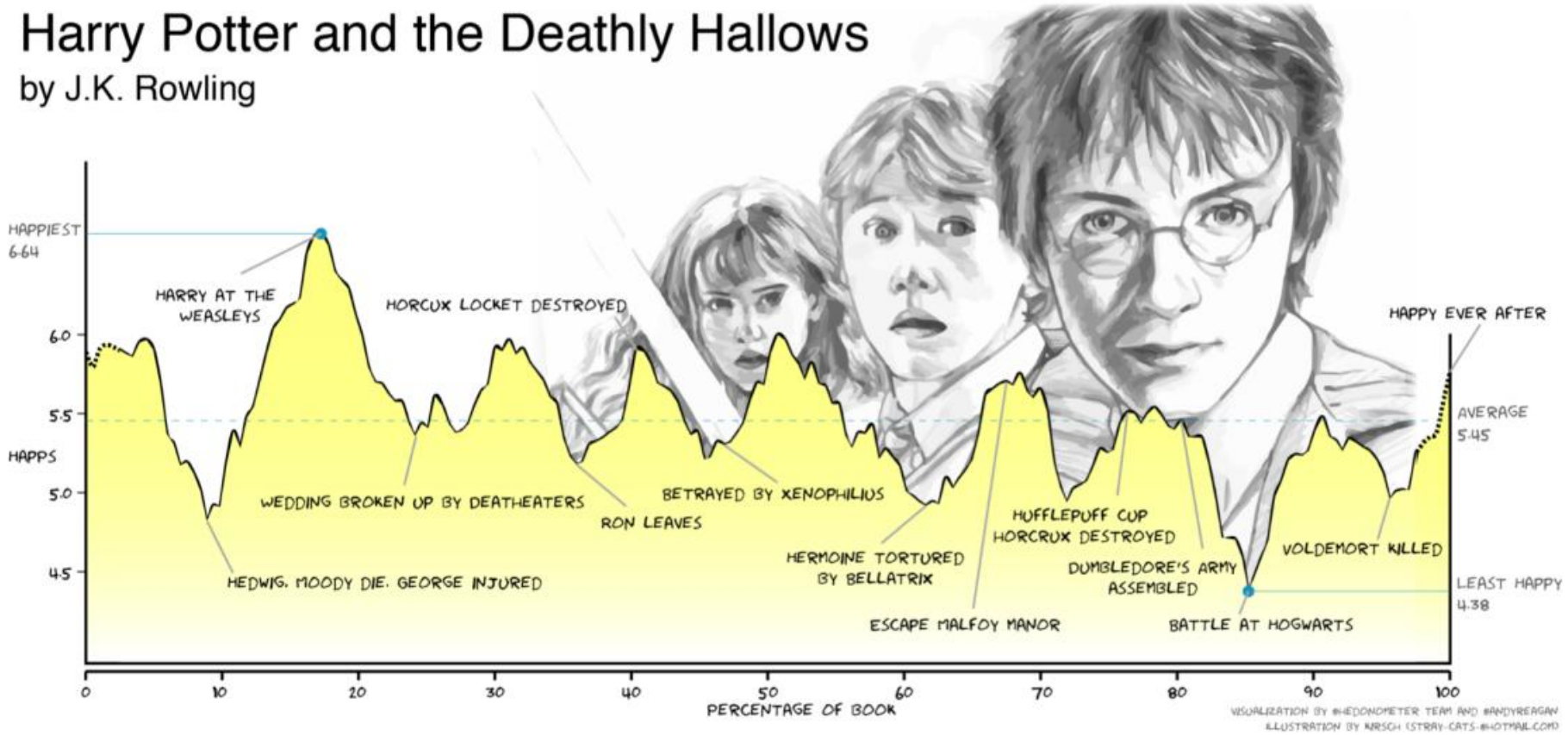
Natural Language Processing

- Block 01 -



Auxiliary AI GmbH

Wie und was bringt es uns, Texte zu verarbeiten?



Picture from ojosujono96 (Freepik)

Zerlegung -> split

```
sentence = "The goal of this lecture
isn't to explain complex free text
processing"

tokens = nltk.word_tokenize(sentence)
# ['The', 'goal', 'of', 'this',
'lecture', 'is', "n't", 'to', 'explain',
'complex', 'free', 'text', 'processing']
```

Auxiliary AI GmbH

Part-of-Speech Tagging

```
pos = nltk.pos_tag(tokens)
# [('The', 'DT'), ('goal', 'NN'), ('of',
'IN'), ('this', 'DT'), ('lecture', 'NN'),
('is', 'VBZ'), ("n't", 'RB'), ('to',
'TO'), ('explain', 'VB'), ('complex',
'JJ'), ('free', 'JJ'), ('text', 'NN'),
('processing', 'NN')]
```

NN*	Noun
VB*	Verb
JJ*	Adjective
RB*	Adverb
DT	Determiner
IN	Preposition

Tokenization

```
>>> import nltk
>>> text = "New York City is the largest city
in the United States."
>>> words = nltk.word_tokenize( text )

>>> nltk.ne_chunk( nltk.pos_tag( words ) )
Tree('S', [Tree('GPE', [('New', 'NNP'), ('York', 'NNP'),
('City', 'NNP')]), ('is', 'VBZ'), ('the', 'DT'),
('largest', 'JJS'), ('city', 'NN'), ('in', 'IN'), ('the',
'DT'), Tree('GPE', [('United', 'NNP'), ('States',
'NNPS')]), ('.', '.')])
```

ORGANIZATION	Georgia-Pacific Corp., WHO
PERSON	Eddy Bonte, President Obama
LOCATION	Murray River, Mount Everest
DATE	June, 2008-06-29
TIME	two fifty a m, 1:30 p.m.
MONEY	GBP 10.40
PERCENT	twenty pct, 18.75 %
FACILITY	Washington Monument, Stonehenge
GPE (geo-political entity)	South East Asia, Midlothian

Auxiliary AI GmbH

Is ja ganz nett ...

- aber wie kann Wörter in meinen Decision Tree, Random Forest oder Support Vector Machine werfen?

- Kein KI/ML Modell dieser Welt, arbeitet auf Text – auch keine LLM!

- **Wie kann Semantik über Syntax ausgedrückt werden?**
  Wie kann Bedeutung über Zeichen ausgedrückt werden?

Auxiliary AI GmbH

Bag of Word (Count)

Raw Text

It is a puppy and it is extremely cute.

Bag of words

```
the 0
beer 0
is 2
cold 0
it 2
a 1
puppy 1
and 1
extremely 1
cute 1
cat 0
guitar 0
```

Auxiliary AI GmbH

Bag of Word (Exists)

Raw Text

It is a puppy and it is extremely cute.

Bag of words

```
the 0
beer 0
is 1
cold 0
it 1
a 1
puppy 1
and 1
extremely 1
cute 1
cat 0
guitar 0
```

Auxiliary AI GmbH

Bag of Word

- Bag of Word berücksichtigt die gesamte Syntax, aber verzichtet auf die Reihenfolge von Sätzen und verliert dadurch sehr viele entscheidende Informationen.

- „boring movie and not great“ <-> „great movie and not boring“
- „schöner Garten und nicht gepflegt“ <-> „gepflegter Garten und nicht schön“
- „schnelles Auto und nicht teuer“ <-> „teures Auto und nicht schnell“
- interessantes Buch und nicht langweilig“ <-> „langweiliges Buch und nicht interessant“

Auxiliary AI GmbH

N-Gram Models

```
list(nltk.ngrams(tokens, 3))
# [('The', 'goal', 'of'), ('goal', 'of',
'this'), ('of', 'this', 'lecture'), ('this',
'lecture', 'is'), ('lecture', 'is', "n't"),
('is', "n't", 'to'), ("n't", 'to', 'explain'),
('to', 'explain', 'complex'), ('explain',
'complex', 'free'), ('complex', 'free',
'text'), ('free', 'text', 'processing')]
```

N-Gram Models

- Zählt die Häufigkeit von Wörtern in einem Dokument -> Term Frequency (TF)
- Anzahl der Dokumente geteilt durch die Anzahl der Dokumente mit einem ausgewählten Term (IDF)

- TF + IDF = TFIDF (Term Frequency Inverse Document Frequency)

- Bisschen nervig, wenn ich das zu Beginn eine ML/DS Projektes für meine Dokumente erstellen muss! Geht das nicht einfacher?
- ***-> Ja und nein ... Word Embeddings***

Auxiliary AI GmbH

Word Embeddings

- **word2vec**, **spaCy, GoVe, flairNLP, bert** und viele weitere

- **Vektorisierung von Wörtern**:
  - Word Embeddings wandeln Wörter in numerische Vektoren um, wobei ähnliche Wörter ähnliche Vektoren haben.
- **Vortrainierte Modelle**:
  - SpaCy bietet Zugriff auf mehrere vortrainierten Embedding-Modelle, die auf großen Textkorpora trainiert und können in Anwendungen direkt verwendet werden können. Diese gibt es in vielen unterschiedlichen Sprachen.

Word Embeddings

"You shall know a word
by the company it keeps"

One of the most successful ideas of modern statistical NLP

government debt problems turning into **banking** crises as has happened in

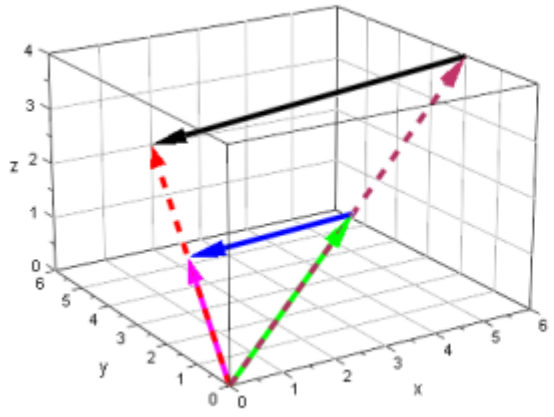
saying that Europe needs unified **banking** regulation to replace the hodgepodge

↖ These words will represent *banking* ↗

Auxiliary AI GmbH

Word Embeddings

- Visuell sieht das so aus:



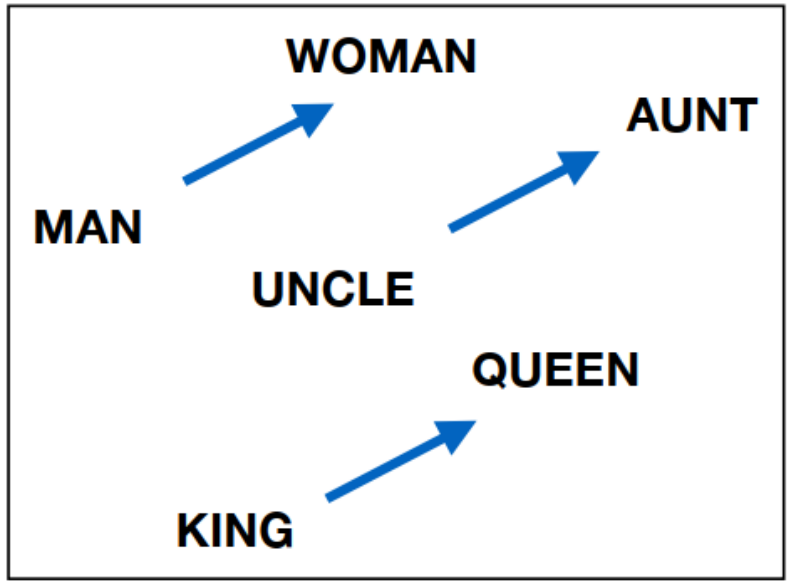
$$linguistics = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{pmatrix}$$

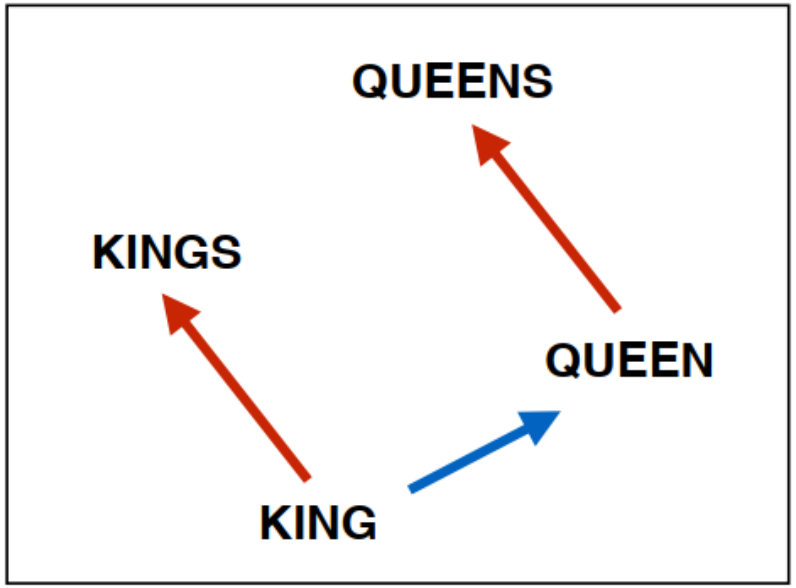
- Die Modelle haben meist 50, 100, 300 oder 1.000 Dimensionen

Auxiliary AI GmbH

Word Embeddings

- Diese Vektoren erfassen semantische Beziehungen, sodass man beispielsweise mathematische Operationen wie „König - Mann + Frau = Königin“ durchführen kann.

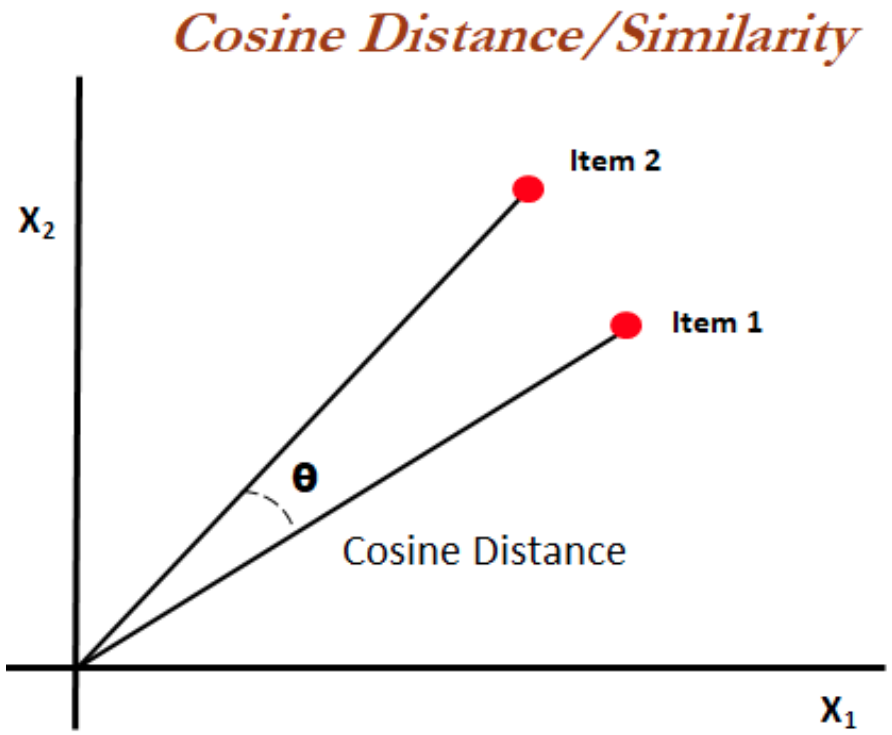




Auxiliary AI GmbH

Word Embeddings - Similarity

- Cosine-Similarity calculates the similarity via the angle of the vectors to the zero point and thus results in a value between 0 & 1.
- Attention, there is also an inverted version of this!

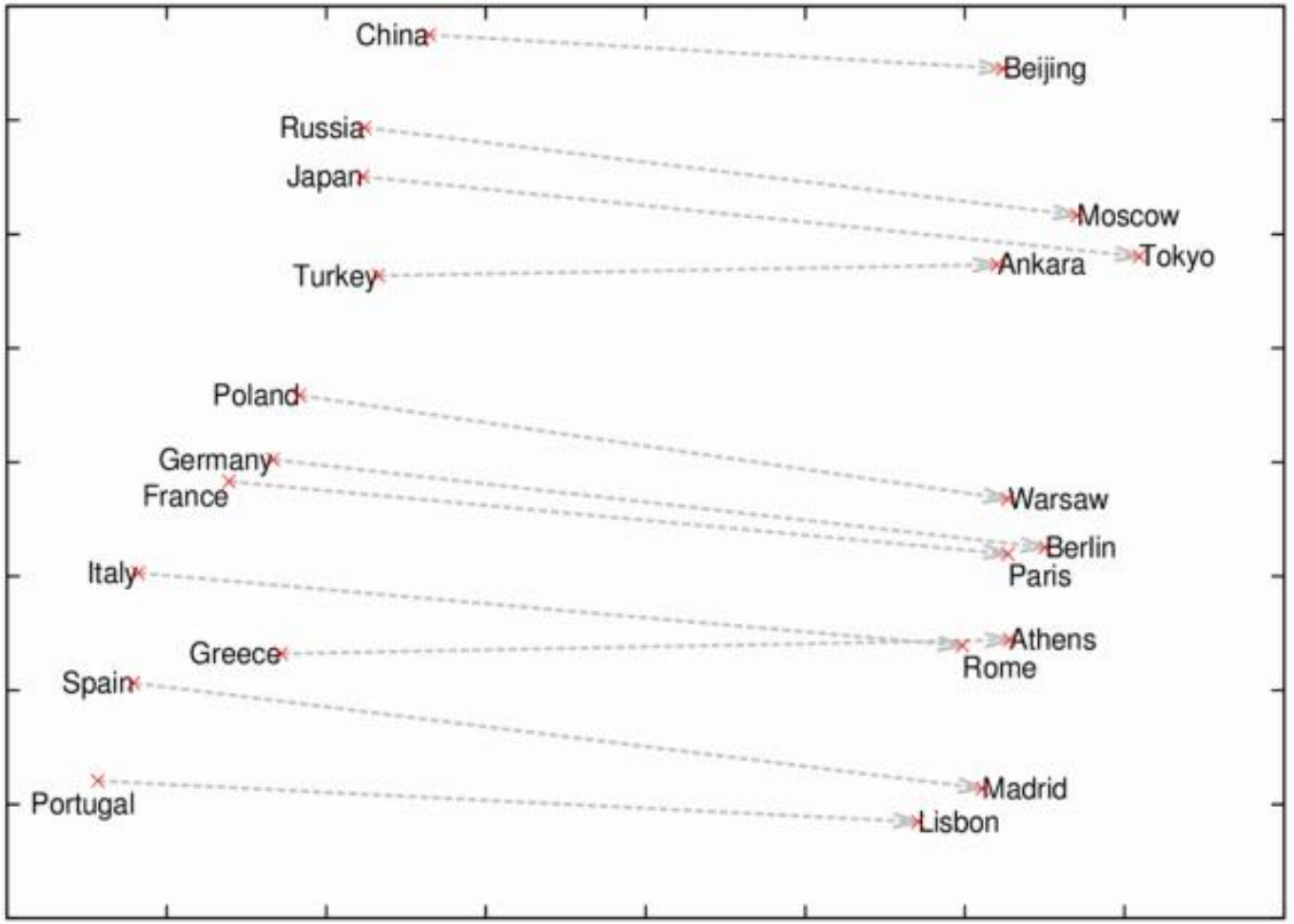


Picture from ojosujono96 (Freepik)

Auxiliary AI GmbH

Word Embeddings

- Entfernungen von Ländern zu ihren Hauptstädten



- nlp("Deutschland").similarity(nlp("Berlin"))
  -> 0.5120840072631836
- nlp("Deutschland").similarity(nlp(„Hamburg"))
  -> 0.4778415560722351
- nlp("Deutschland").similarity(nlp(„Copenhagen"))
  -> 0.27788498997688293
- nlp("Ich").similarity(nlp("ich"))
  -> 0.8771186470985413

Picture from ojosujono96 (Freepik)

Auxiliary AI GmbH

Auxiliary AI GmbH

Geschäftsführer	**Marten Borchers & Benjamin Klinkigt**
Anschrift	Am Ziegelteich 74 22525 Hamburg Deutschland
Handelsregister	HR B 185519
Registergericht	Amtsgericht Hamburg
Umsatzsteuer-ID	DE 366 814 276
Kontakt	info@auxiliary-ai.de
Webseite	https://auxiliary-ai.de/