

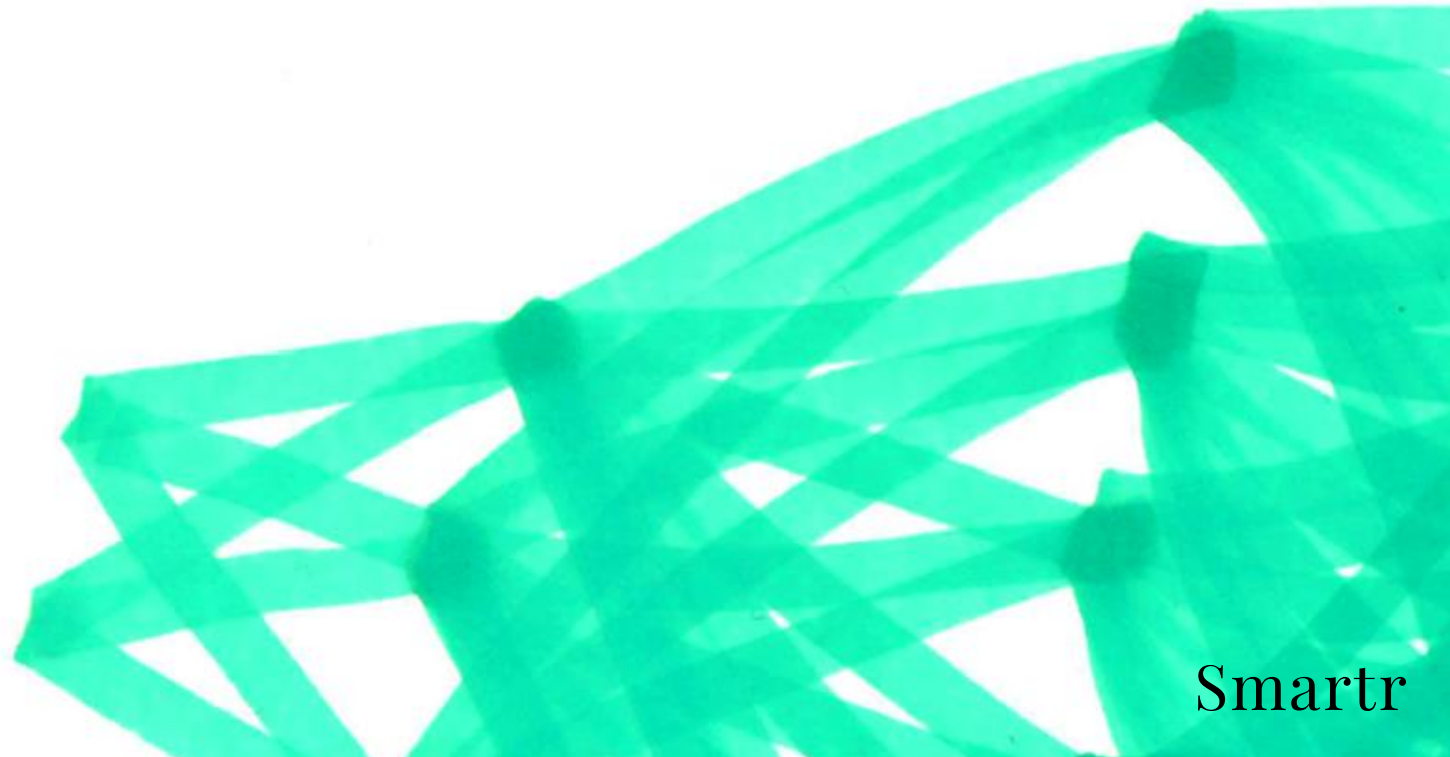
Deep Learning

2023-12-13

Ämne: NLP och RNN

Agenda

- Natural language processing (NLP)
 - Hur jobbar vi med text?
- Recurrent Neural Networks
- Encoder/Decoder
- Tidsserier



NLP

Natural Language Processing handlar om att kombinera språkvetenskap med AI.

Med internet har vi explosionsartat ökat mängden text som är tillgänglig att träna modeller på.



NLP

- Språk är väldigt komplexa med massor av regler, vilket gör dem svåra att jobba med.
- Meningar och ord kan betyda flera olika saker.
- Det finns väldigt många språk (och dialekter).
- Svårt att förstå sarkasm och ironi i text.
- Kultur och kontext har stor betydelse.



NLP

- Vad räknas som en bra text?
 - Text på internet har väldigt varierande kvalitet.
 - Hur hanterar vi felstavningar?
- Hur vi skriver förändras över tid.
 - En bok från 1700-talet behöver inte vara bra data för modeller som ska jobba med nutida text.



NLP

Mentimeter - Vad använder ni
som använder natural language
processing?

NLP

- Språköversättning
- Generera text
 - Chatbotar
- Sentimentanalys
 - Är texten positiv eller negativ?
- Text till tal och tal till text
 - Exempelvis generera undertexter



NLP

Textkorpus, eller Text coprus på engelska, är en ett dataset med text.

Det finns flera olika stora dataset som innehåller data från bland annat tidningsartiklar, forskningsartiklar, foruminlägg och kundrecensioner.

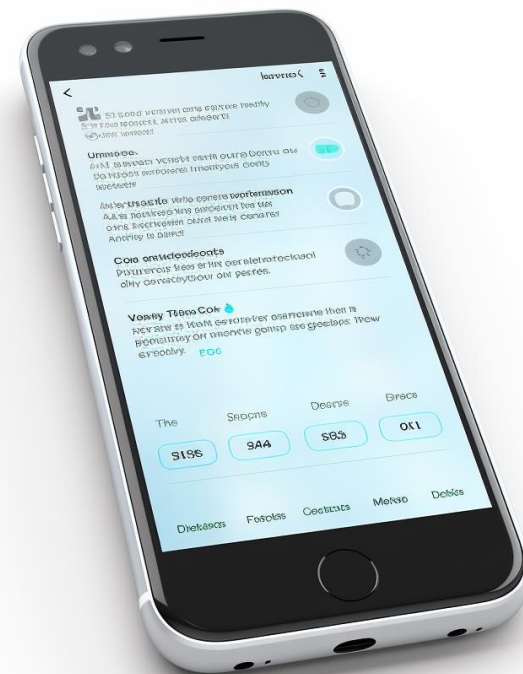
Vilken typ av dataset man vill använda beror på vad för typ av text modellen ska jobba med.



NLP

För att kunna träna modeller på text behöver vi jobba en del med texten först.

- Tokenization
 - Bryta ner texter i ord
- Stoppord (Stop words)
 - Ta bort ord som inte bidrar till meningens betydelse
- Lemmatization
 - Ta bort böjningen av ord
- Word embeddings
 - Göra om ord till vektorer



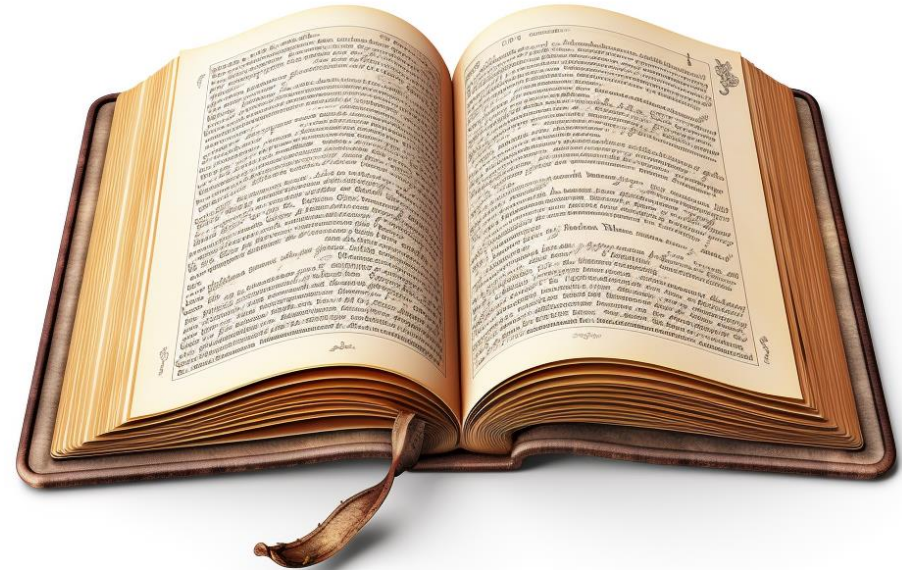
NLP

Tokenization

Vi vill bryta ner meningar till ord.
På så vis kan vi lära oss ordens
betydelse och hur de hänger
ihop.

Exmepel:

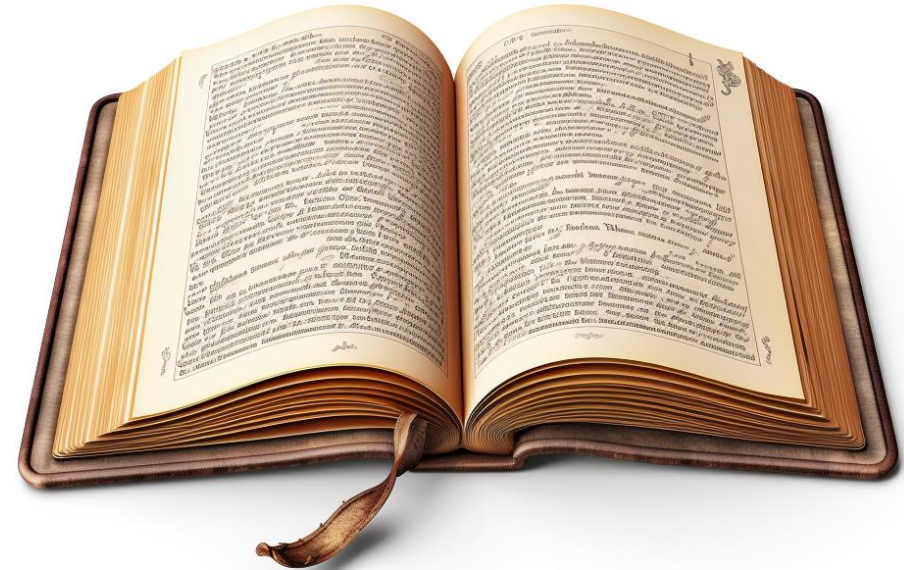
"Jag läser en bok" bryts ner till
"Jag", "läser", "en", "bok".



NLP

Tokenization

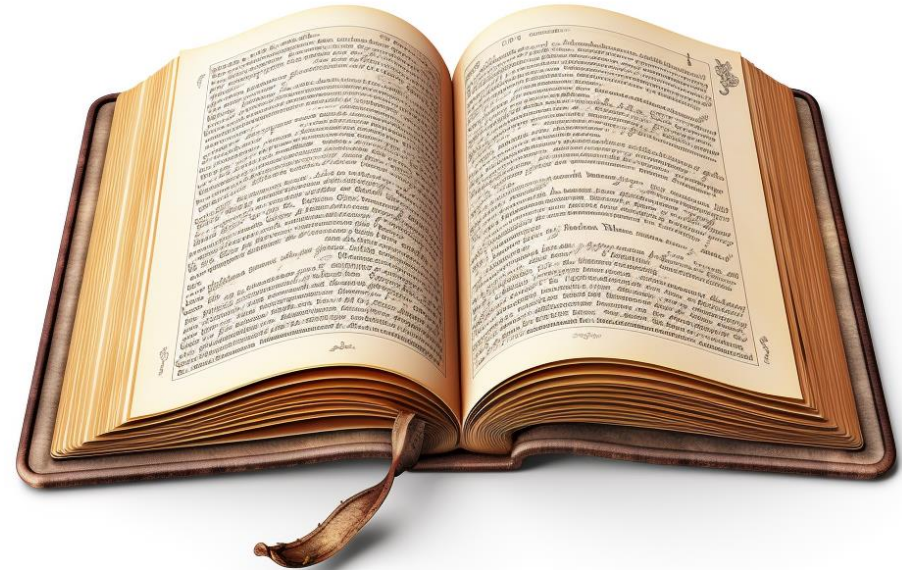
Vi kan få in ord som inte ingår i träningsdatan och därför modellen inte har någon kännedom om. För att hantera detta skapar man ofta ett unknownn ord ("UNK", "<UNK>", eller liknande) som nya ord går under.



NLP

Tokenization

```
{  
  "<UNK>": 0  
  "jag": 1,  
  "läser": 2,  
  "en": 3,  
  "bok": 4  
}
```

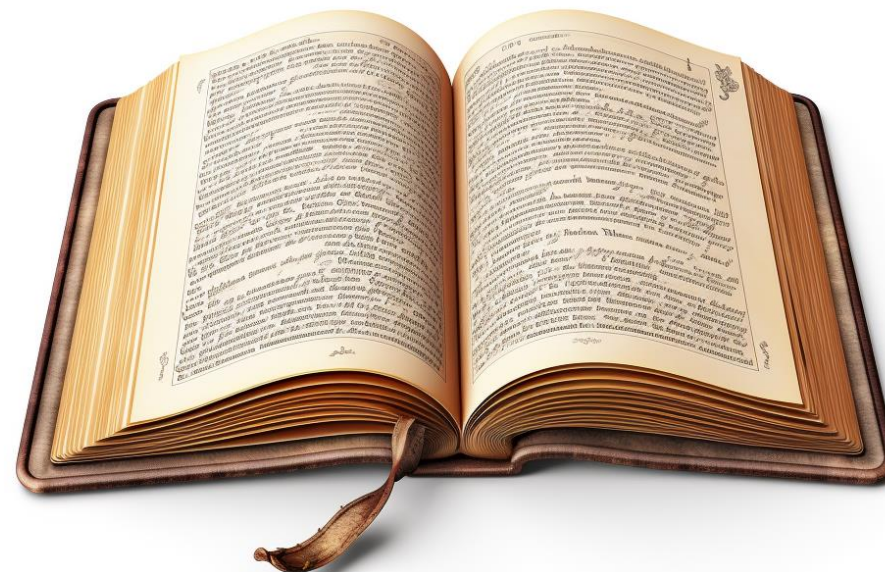


NLP

N-grams

Vi kan även bryta ner meningar i N-grams. Där vi bryter ner meningen i sekvenser om innehåller N ord.

Detta kan användas för att få fram sannolikheten att ett ord kommer efter en ordföljd.



NLP

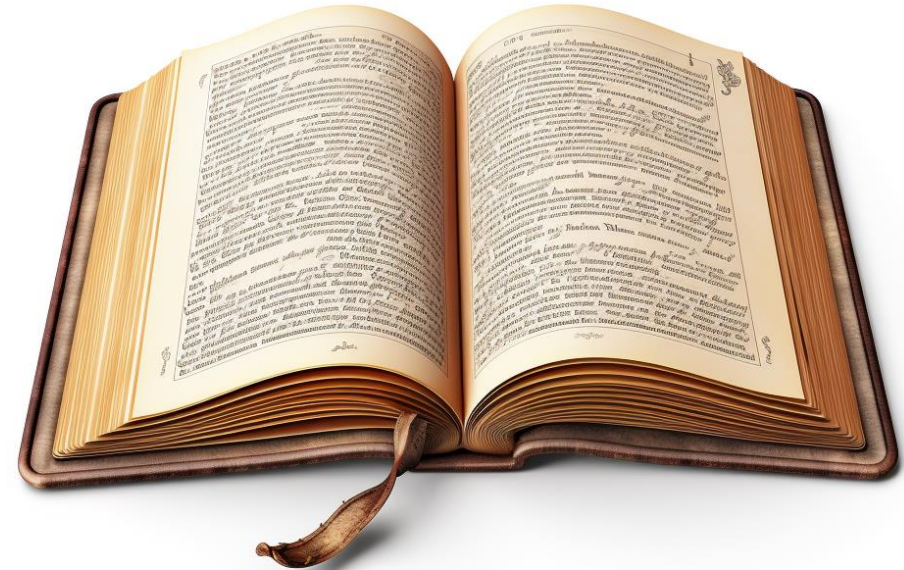
"Jag läser en bok" kan brytas ner i flera N-gram.

1-gram (unigrams) - "Jag", "läser", "en", "bok"

2-gram (bigrams) - "Jag läser", "läser en", "en bok"

3-gram (trigram) - "Jag läser en", "läser en bok"

Osv



NLP

Stop words

Stop words är vanligtvis de vanligaste orden i ett språk.

Några engelska exempel är "and", "is", "the", "in", "to" och "it".

Orden förekommer väldigt ofta utan att alltid bidra till meningens innebörd.

Genom att plocka bort den typen av ord får vi många fördelar utan att tappa information.



NLP

Stop words

Genom att plocka bort ord kan vi

- Reducera antalet dimensioner på vår data.
- Snabba på beräkningarna (mindre ord att jobba med).
- Få brusreducering (noise reduction). Genom att plocka bort ord som inte bidrar lyfter vi istället de orden som innehåller mer information.



NLP

Lemmatization

Med lemmatization förenklar vi meningar genom att "normalisera" orden.

Exempelvis ändras "hunden", "hundar" och "hundarna" till "hund".



NLP

Lemmatization

Även här spelar språks komplexitet in och detta behöver hanteras olika i olika språk.

För stora språk (exempelvis Engelska) finns det verktyg för att göra detta.



NLP

Lemmatization

Även här spelar språks komplexitet in och detta behöver hanteras olika i olika språk.

För stora språk (exempelvis Engelska) finns det verktyg för att göra detta.



NLP

Lemmatization

- Minska risk för over fitting
 - Vi behandlar inte ord med olika böjelser som helt unika ord
- Minskar antalet ord som vi behöver lära modellen
- Får en korrekt räkning av hur ofta ett ord förekommer
 - Oavsett om det står i en annan form från början
- Utan kontext kan ord som normaliseras bli amma ord, utan att egentligen betyda samma sak
- Verktyg uppdateras inte lika snabbt som nya ord.

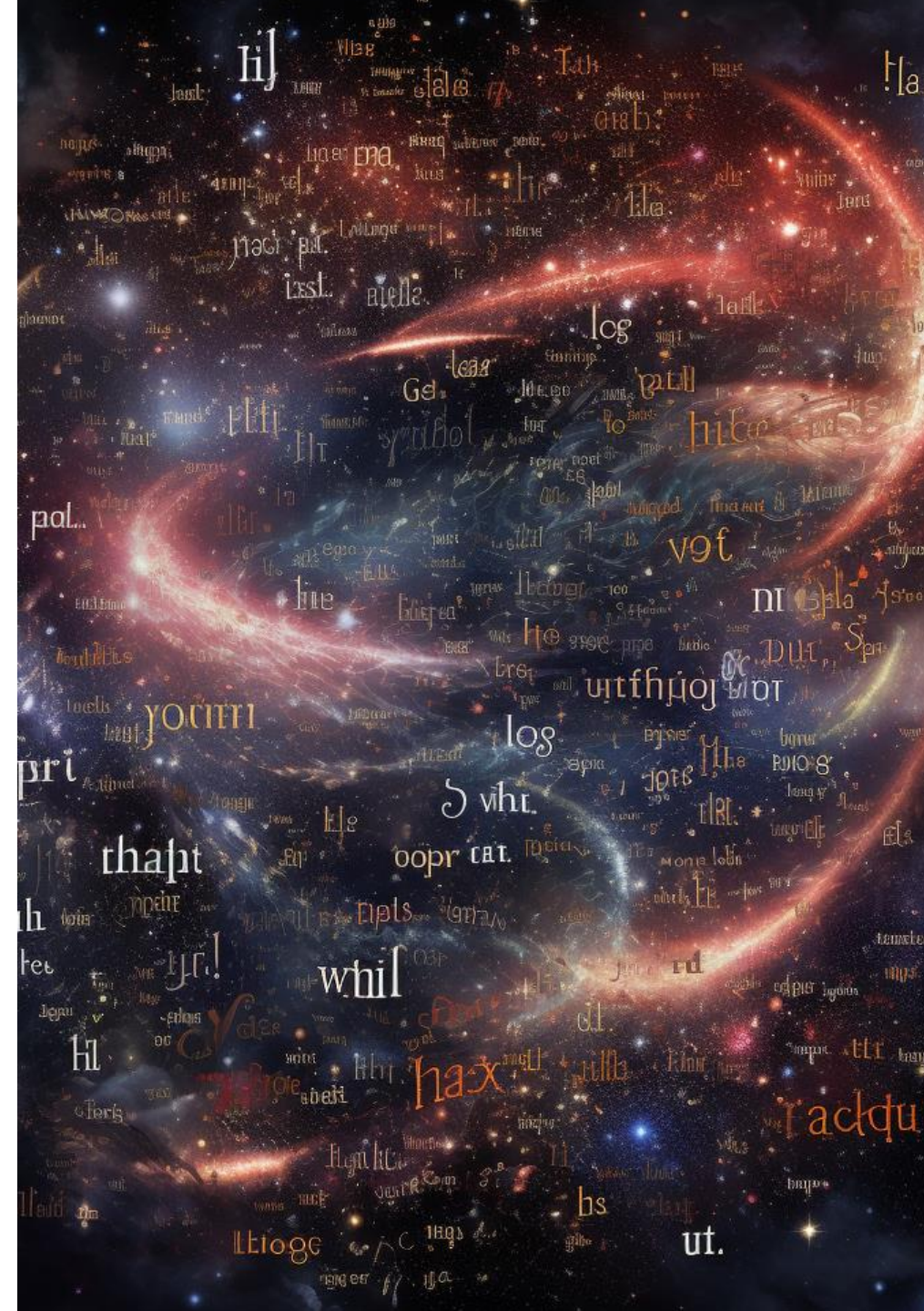


NLP

Word embeddings

Vi kan inte räkna på ord. Därför behöver vi göra om orden till något vi kan använda för att se hur ord hänger ihop, om de är lika osv.

Exempelvis vet vi att "hus" och "sommarestuga" är väldigt lika, men det är för att vi vet vad det är. Bokstäverna och orden i sig säger inget om hur lika de är.



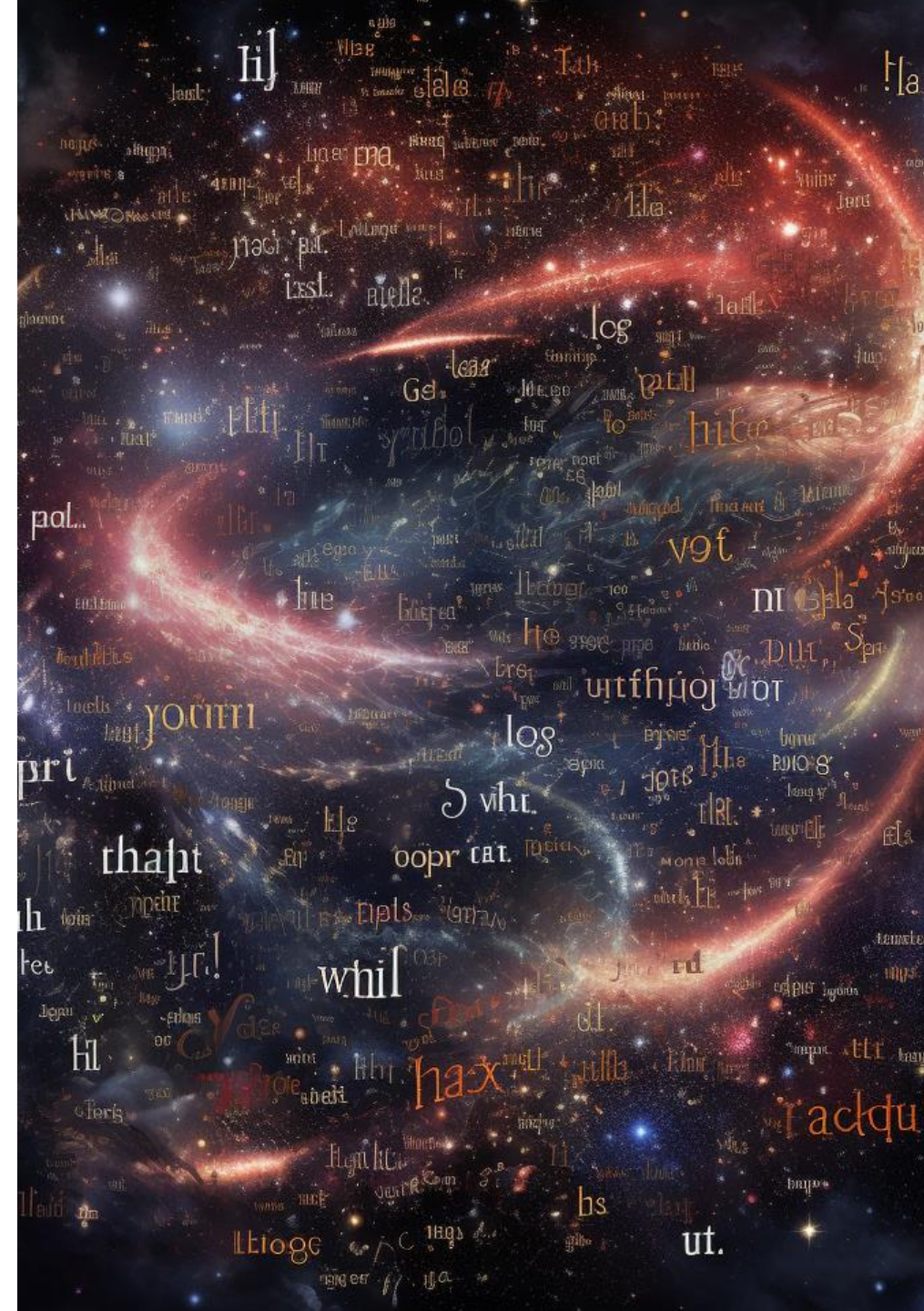
NLP

Word embeddings

Att förvandla orden till högdimensionella vektorer (hundratala dimension) kan vi börja räkna på orden och hitta samband mellan dem.

Detta kallas **Word2Vec** och har hjälpt till med stora framgångar inom machine learning och språk.


Det finns bättre metoder nu som Large Language Models (exempelvis ChatGPT) använder. Men det är fortfarande en väldigt bra teknik för flera användningsområden.



NLP

Word embeddings

king




man



woman



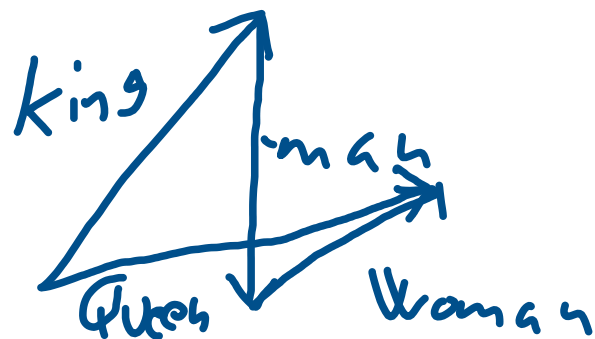
queen



NLP

Word embeddings

King - Man + Woman = Queen



NLP

Word embeddings

Flera av världens största techbolag har använt Word2Vec för att förbättra befintliga produkter eller skapa helt nya.



NLP

När vi jobbar med meningar så vet vi inte storleken på vår input. Vi behöver sätta en maxgräns på antal ord.

Alla meningar kommer inte inne hålla så många ord, men vi måste fortfarande fylla inputen. Detta gör med hjälp av **padding**. Det är ytterligare ett specialord, men som inte betyder någonting.

NLP

```
{  
  "<PAD>": 0  
  "<UNK>": 1  
  "jag": 2,  
  "läser": 3,  
  "en": 4,  
  "bok": 5  
}
```

| jag | läser | en | bok |
|-----|-------|----|-----|
| 2 | 3 | 4 | 5 |

| jag | läser | tidningen | <PAD> |
|-----|-------|-----------|-------|
| 2 | 3 | 1 | 0 |