Named Entity Recognition
Topic Modeling
Sentiment Analysis

PREPARED BY: AHMAD ALAA ALDINE

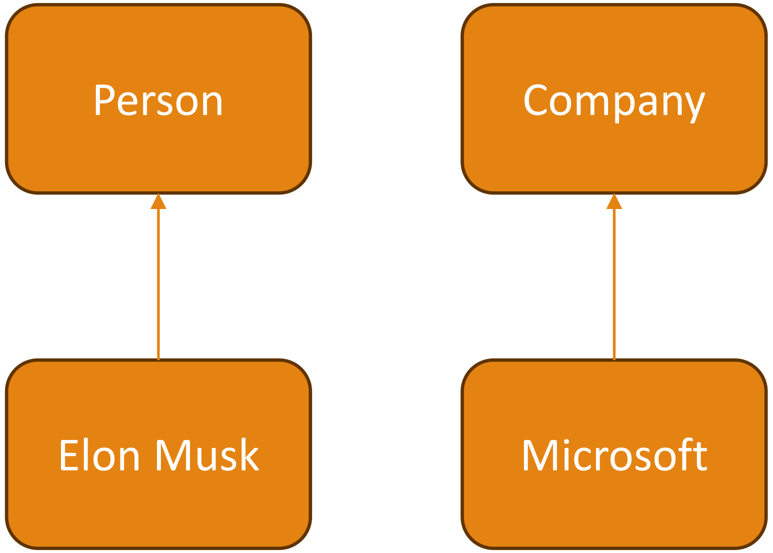
PRESENTED BY: AHMAD ALAA ALDINE

Named Entity Recognition

What is Named Entity?

A named entity is a specific word or phrase that refers to a particular:

- Person
- Place
- Organization
- Money
- Time
- Other real-world values



Named Entity Recognition

Named Entity Recognition (NER) is an NLP technique to find and classify entities from textual data into predefined categories.

Named Entity Recognition Methods:

- Rule-based Methods (Linguistic patterns, regular expressions)
- Machine Learning Methods (SVM, Decision Tree)
- Deep Learning Methods (RNN, Transformers)

Types of Named Entities

Person
- name of an individual, such as Tom Cruise or Lionel Messi.

•Organizations
- name of a company, organization, or institution, such as Microsoft Corporation or Stanford University.

•Locations
- name of places, such as Lebanon, Beirut, or Mount Everest.

•Products
- name of products, such as Macbook.

•Events
- name of events, such as FIFA World Cup.

Where Is NER Used?

Information Extraction

- NER is used to extract specific named entities from text and store them in a structured format. This information is then used for purposes, such as generating reports or building knowledge graphs.

Question Answering

- NER be used to tag answers with relevant entities (e.g. people, organizations, locations). These tags can be used to quickly and efficiently match questions with relevant answers.

Text summarization:

- NER can be used to find important named entities in a text or a document and use them to give a summary of the text with contextual information highlighted.

NER with Python

NER Using Spacy Pre-Trained Model (1/3)

Named Entities Categories

```
import spacy

nlp = spacy.load("en_core_web_sm")

print(nlp.pipe_labels['ner'])
```

```
['CARDINAL', 'DATE', 'EVENT', 'FAC', 'GPE', 'LANGUAGE', 'LAW', 'LOC', 'MONEY', 'NORP', 'ORDINAL', 'ORG', 'PERCENT', 'PERSON',
'PRODUCT', 'QUANTITY', 'TIME', 'WORK_OF_ART']
```

NER Using Spacy Pre-Trained Model (2/3)

```
text ="""Elon Reeve Musk (/ˈiːlɒn/; EE-lon; born June 28, 1971) is a businessman and investor.
He is the founder, chairman, CEO, and CTO of SpaceX; angel investor, CEO, product architect and former chairman of Tesla, Inc.; o
He is the wealthiest person in the world, with an estimated net worth of US$232 billion as of December 2023, according to the Blo

doc = nlp(text)

for ent in doc.ents:
    print(ent.text, "|", ent.label_, "|", spacy.explain(ent.label_))
```

```
Elon Reeve Musk | PERSON | People, including fictional
June 28, 1971 | DATE | Absolute or relative dates or periods
CTO | ORG | Companies, agencies, institutions, etc.
angel investor | PERSON | People, including fictional
Tesla, Inc. | ORG | Companies, agencies, institutions, etc.
CTO of X Corp. | ORG | Companies, agencies, institutions, etc.
the Boring Company | ORG | Companies, agencies, institutions, etc.
Neuralink | ORG | Companies, agencies, institutions, etc.
OpenAI | GPE | Countries, cities, states
the Musk Foundation | ORG | Companies, agencies, institutions, etc.
US$232 billion | MONEY | Monetary values, including unit
December 2023 | DATE | Absolute or relative dates or periods
the Bloomberg Billionaires Index | ORG | Companies, agencies, institutions, etc.
$254 billion | MONEY | Monetary values, including unit
Forbes | ORG | Companies, agencies, institutions, etc.
Tesla | ORG | Companies, agencies, institutions, etc.
```

NER Using Spacy Pre-Trained Model (3/3)

```
from spacy import displacy

displacy.render(doc, style="ent")
```

Elon Reeve Musk PERSON (/ˈiːlɒn/; EE-lon; born June 28, 1971 DATE) is a businessman and investor.

He is the founder, chairman, CEO, and CTO ORG of SpaceX; angel investor PERSON , CEO, product architect and former chairman of Tesla, Inc. ORG ; owner, chairman and CTO of X Corp. ORG ; founder of the Boring Company ORG and xAI; co-founder of Neuralink ORG and OpenAI GPE ; and president of the Musk Foundation ORG .

He is the wealthiest person in the world, with an estimated net worth of US$232 billion MONEY as of December 2023 DATE , according to the Bloomberg Billionaires Index ORG , and $254 billion MONEY according to Forbes ORG , primarily from his ownership stakes in Tesla ORG and SpaceX

Topic Modeling

Topic Modeling (1/3)

It is a statistical approach designed to extract topics present in a set of documents.

It is an unsupervised approach → No need for labeled datasets.

It can be seen as a clustering approach.

- Clusters of words representing documents topics.
- Number of topics -> Number of clusters.
- Topics are abstracts.

Topics can be defined as a repeating pattern of co-occurring terms in a corpus:

- health, doctor, patient, hospital for a topic – Healthcare
- farm, crops, wheat for a topic – Farming

Topic Modeling (2/3)

Topic Modeling can be used:

- Document Clustering
- Information Retrieval
- Recommendation systems

Real applications:

- New York Times is using topic models to boost its user–article recommendation engines.
- Various professionals are using topic models for recruitment industries where they aim to extract latent features of job descriptions and map them to the right candidates.

Topic Modeling (3/3)

Most popular approach for Topic Modeling

1. Latent Dirichlet Allocation (LDA)
2. Latent Semantic Analysis (LSA)

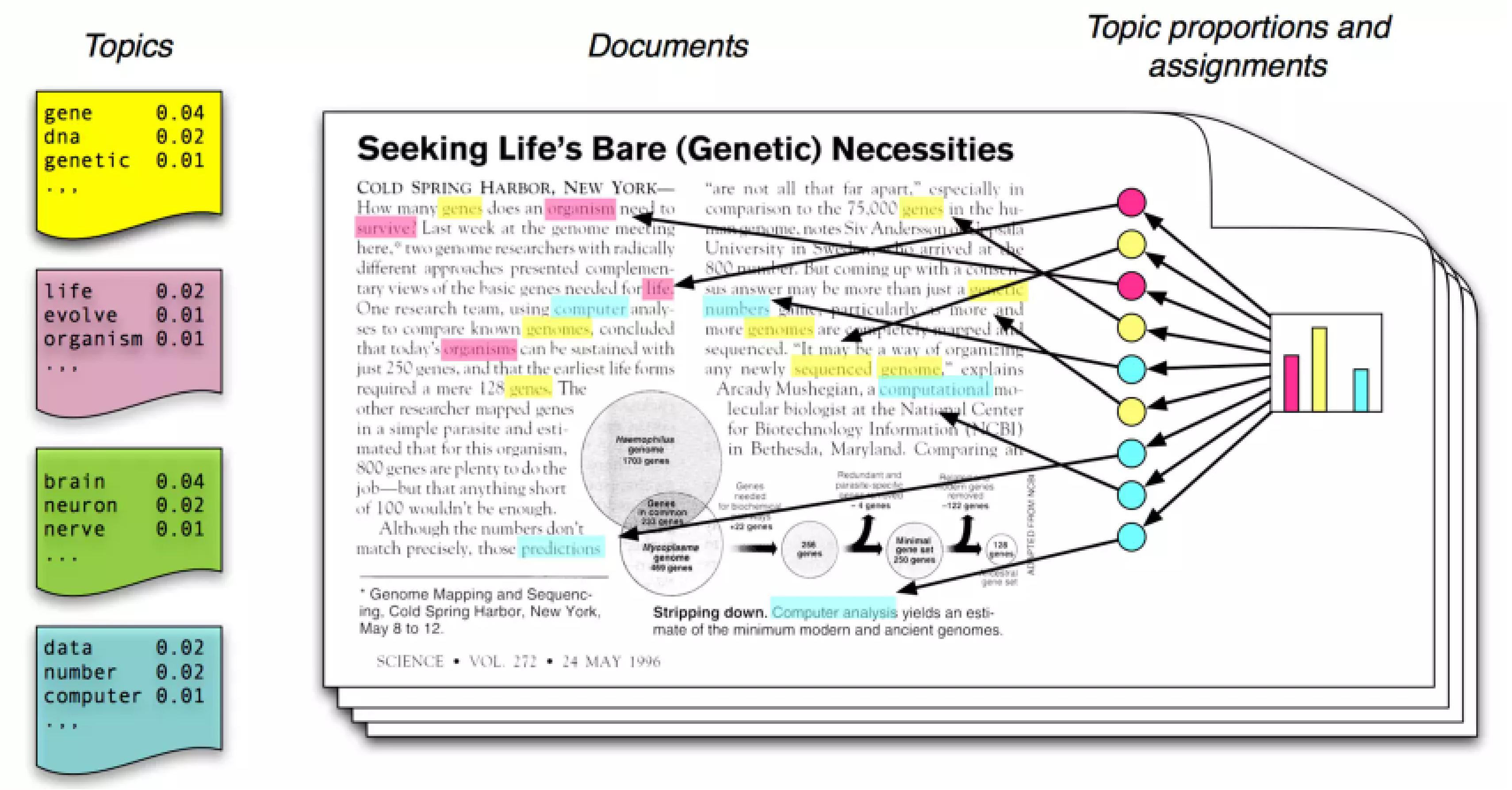
LDA	LSA
Identifying topics and their distribution across documents	Capturing the latent semantic structure and reducing dimensionality
Represents topics as probability distributions over words	Represents documents and terms in a lower-dimensional semantic space

LDA

Each **document** is a mixture of topics

Each **topic** is a distribution over words

Each **word** is drawn from one of those topics



LSA

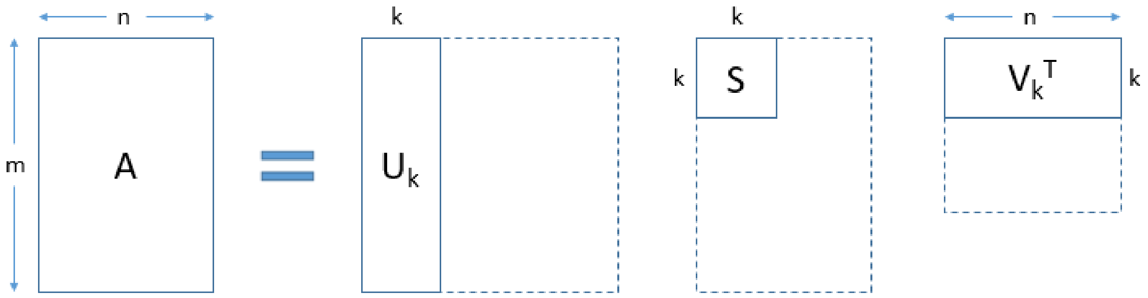
TF-IDF Vectorization

Terms

Documents

	T1	T2	T3	...	Tn
D1	0.2	0.1	0.5	...	0.1
D2	0.1	0.3	0.4	...	0.3
D3	0.3	0.1	0.1	...	0.5
...	...	...	...	...	...
Dm	0.2	0.1	0.2	...	0.1

Dimensionality Reduction using Singular-Value Decomposition (SVD)



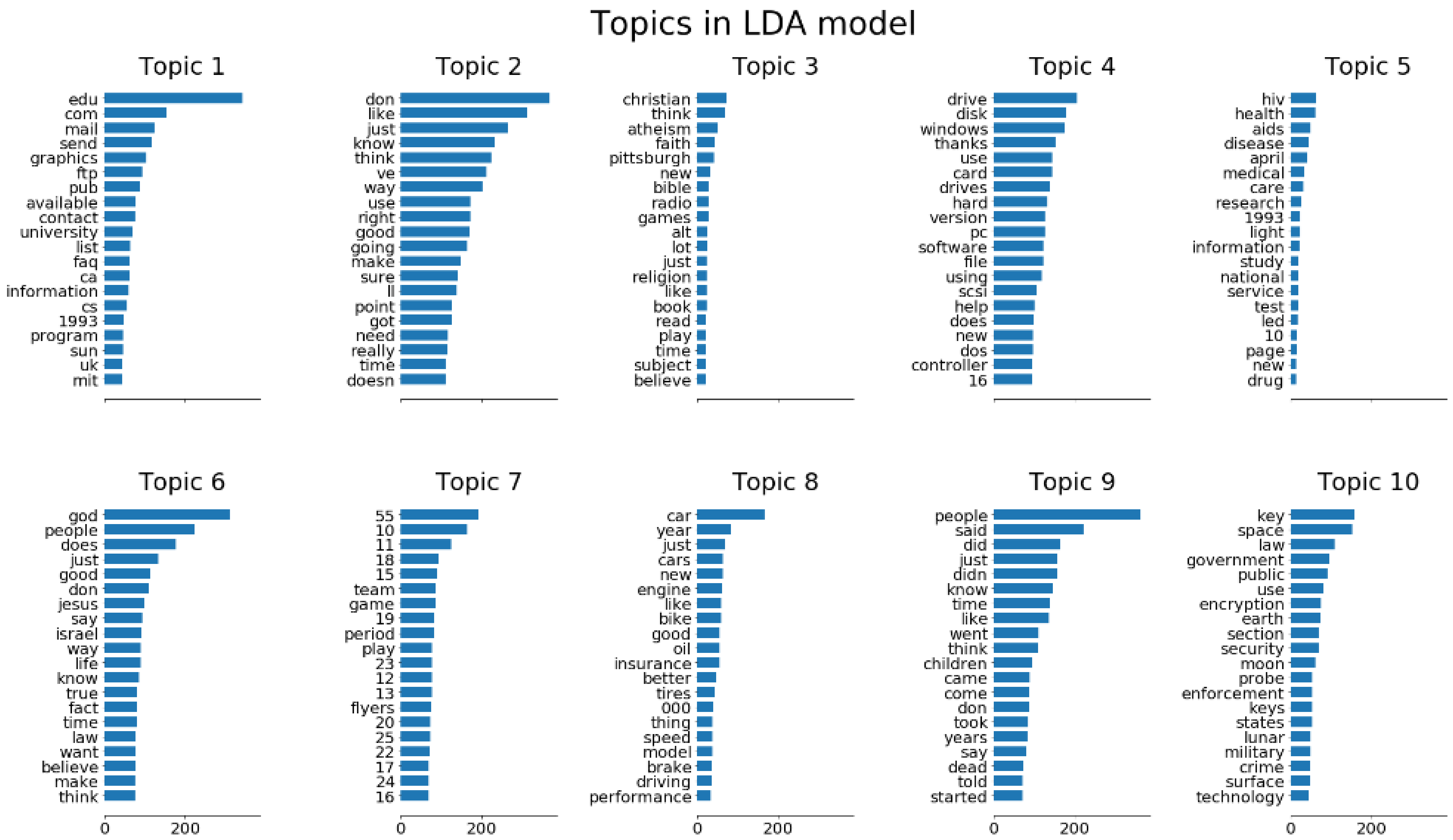
Document – Term matrix

Document – Topic matrix

Topic – Term matrix

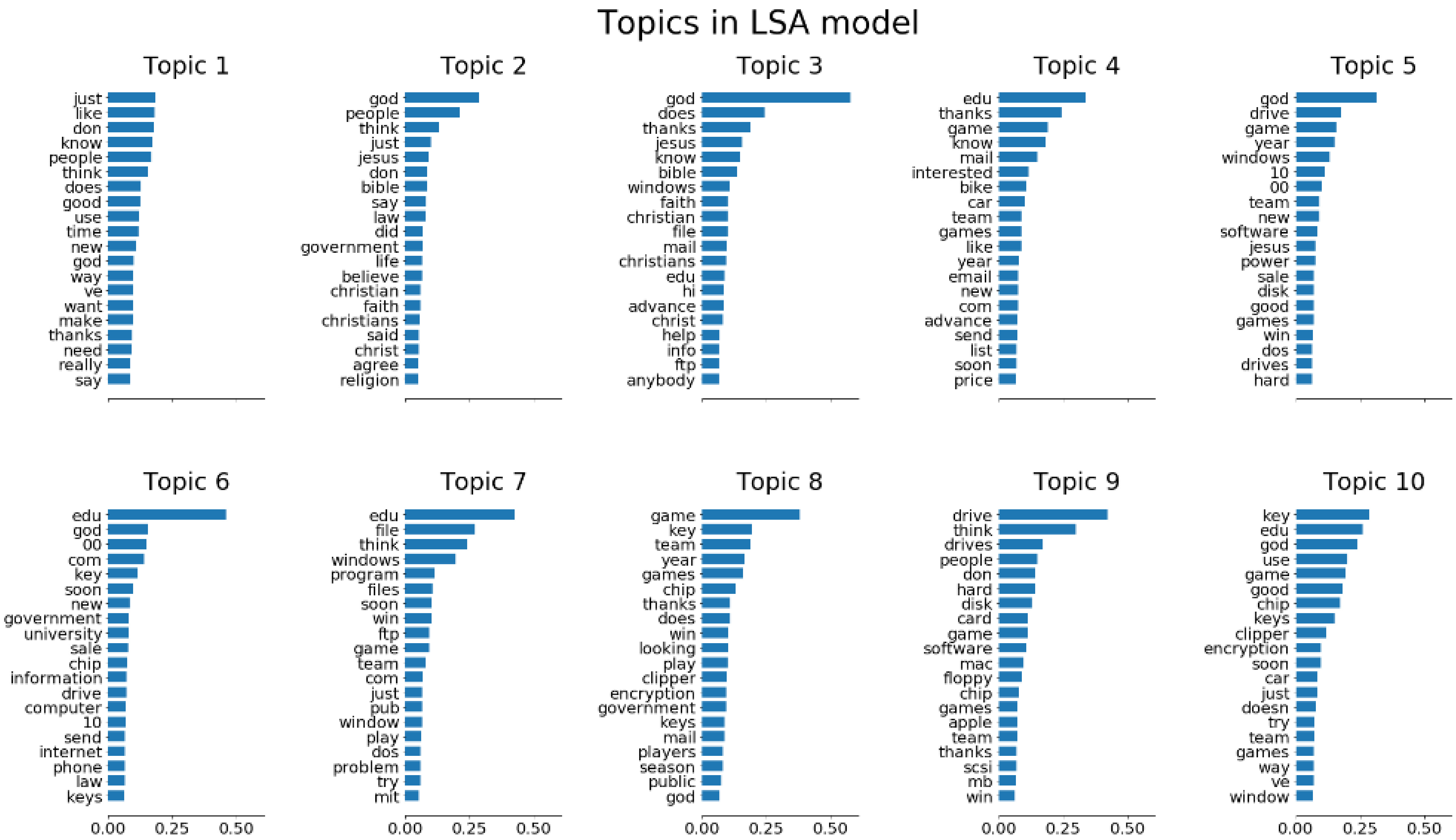
Topic Modeling with Python

```
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_20newsgroups
from sklearn.decomposition import LatentDirichletAllocation
from sklearn.feature_extraction.text import CountVectorizer

n_samples = 2000
n_features = 1000
n_components = 10
n_top_words = 20

data = fetch_20newsgroups(
    shuffle=True,
    random_state=1,
    remove=("headers", "footers", "quotes"),
)["data"]
data_samples = data[:n_samples]


tf_vectorizer = CountVectorizer(
    max_df=0.95, min_df=2, max_features=n_features, stop_words="english"
)
tf = tf_vectorizer.fit_transform(data_samples)

lda = LatentDirichletAllocation(
    n_components=n_components,
    max_iter=5,
    learning_method="online",
    learning_offset=50.0,
    random_state=0,
)

lda.fit(tf)

tf_feature_names = tf_vectorizer.get_feature_names()
plot_top_words(lda, tf_feature_names, n_top_words, "Topics in LDA model")
```



Topics in LDA model
Topic 1
edu com mail send graphics ftp pub available contact university list faq ca information cs 1993 program sun uk mit
Topic 2
don like just know think ve way use right good going make sure ll point got need really time doesn
Topic 3
christian think atheism faith pittsburgh new bible radio games alt lot just religion like book read play time subject believe
Topic 4
drive disk windows thanks use card drives hard version pc software file using scsi help does new dos controller 16
Topic 5
hiv health aids disease april medical care research 1993 light information study national service test led 10 page new drug
Topic 6
god people does just good don jesus say israel way life know true fact time law want believe make think
Topic 7
55 10 11 18 15 team game 19 period play 23 12 13 flyers 20 25 22 17 24 16
Topic 8
car year just cars new engine like bike good oil insurance better tires 000 thing speed model brake driving performance
Topic 9
people said did just didn know time like went think children came come don took years say dead told started
Topic 10
key space law government public use encryption earth section security moon probe enforcement keys states lunar military crime surface technology
0
200

```
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_20newsgroups
from sklearn.decomposition import LatentDirichletAllocation
from sklearn.feature_extraction.text import CountVectorizer

n_samples = 2000
n_features = 1000
n_components = 10
n_top_words = 20

data = fetch_20newsgroups(
    shuffle=True,
    random_state=1,
    remove=("headers", "footers", "quotes"),
)["data"]
data_samples = data[:n_samples]

tfidf_vectorizer = TfidfVectorizer(
    max_df=0.95, min_df=2, max_features=n_features, stop_words="english"
)

tfidf = tfidf_vectorizer.fit_transform(data_samples)

svd = TruncatedSVD(n_components=10, n_iter=7, random_state=42)
svd.fit(tfidf)

tf_feature_names = tfidf_vectorizer.get_feature_names()
plot_top_words(svd, tf_feature_names, n_top_words, "Topics in LSA model")
```


Topics in LSA model
Topic 1
just
like
don
know
people
think
does
good
use
time
new
god
way
ve
want
make
thanks
need
really
say
Topic 2
god
people
think
just
jesus
don
bible
say
law
did
government
life
believe
christian
faith
christians
said
christ
agree
religion
Topic 3
god
does
thanks
jesus
know
bible
windows
faith
christian
file
mail
christians
edu
hi
advance
christ
help
info
ftp
anybody
Topic 4
edu
thanks
game
know
mail
interested
bike
car
team
games
like
year
email
new
com
advance
send
list
soon
price
Topic 5
god
drive
game
year
windows
10
00
team
new
software
jesus
power
sale
disk
good
games
win
dos
drives
hard
Topic 6
edu
god
00
com
key
soon
new
government
university
sale
chip
information
drive
computer
10
send
internet
phone
law
keys
Topic 7
edu
file
think
windows
program
files
soon
win
ftp
game
team
com
just
pub
window
play
dos
problem
try
mit
Topic 8
game
key
team
year
games
chip
thanks
does
win
looking
play
clipper
encryption
government
keys
mail
players
season
public
god
Topic 9
drive
think
drives
people
don
hard
disk
card
game
software
mac
floppy
chip
games
apple
team
thanks
scsi
mb
win
Topic 10
key
edu
god
use
game
good
chip
keys
clipper
encryption
soon
car
just
doesn
try
team
games
way
ve
window
0.00
0.25
0.50

Sentiment Analysis

What is Sentiment Analysis?

Sentiment Analysis is to identify the view or emotion behind a situation.

It means to analyze and find the emotion or intent behind a piece of text or any mode of communication.

Each communication text has a sentiment associated with it.

It might be:

- Positive
- Negative
- Neutral

Sentiment Analysis Can be Used For

Product or service marketing

- With the launch of a new product, companies can employ sentiment analysis to understand user response to the new product. Based on customer feedback, companies can zero in on speeding up the product production process, identify the features that need to be added, resolve bugs from elements causing problems, and so on.

Efficient data mining practice

- Businesses can use sentiment analysis as a data mining tool that can help them gather competitive intelligence concerning competitor brands. With such data, companies can gain a competitive edge over other brands, allowing them to adjust their business model based on market sentiments.

Supports political analysis

- Sentiment analysis on social media platforms such as Twitter can allow official authorities to keep a check on people's reactions to newly-framed political policies. Political parties can reframe their policies and plan their election manifesto or campaigns based on people's responses, anger, and common trends.

Sentiment Analysis Using spaCy

```python
import spacy
from spacytextblob.spacytextblob import SpacyTextBlob

nlp = spacy.load('en_core_web_sm')
nlp.add_pipe("spacytextblob")

corpus = ["I love this cheese sandwich, it’s so delicious.",
          "This chicken burger has a very bad taste.",
          "I ordered this pizza today."]
for text in corpus:
    doc = nlp(text)
    print(text, "|", 'Polarity:', doc._.polarity) # [-1, 1]
```

```
I love this cheese sandwich, it’s so delicious. | Polarity: 0.75
This chicken burger has a very bad taste. | Polarity: -0.7549999999999999
I ordered this pizza today. | Polarity: 0.0
```