# Classifying fake news using supervised learning with NLP

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN PYTHON

**Katharine Jarmul**
Founder, kjamistan

datacamp

# What is supervised learning?

- Form of machine learning
  - Problem has predefined training data
  - This data has a label (or outcome) you want the model to learn
  - Classification problem
  - Goal: Make good hypotheses about the species based on geometric features

| Sepal length | Sepal width | Petal length | Petal width | Species |
|---|---|---|---|---|
| 5.1 | 3.5 | 1.4 | 0.2 | I. setosa |
| 7.0 | 3.2 | 4.77 | 1.4 | I.versicolor |
| 6.3 | 3.3 | 6.0 | 2.5 | I.virginica |

# Supervised learning with NLP

- Need to use language instead of geometric features
- `scikit-learn` : Powerful open-source library
- How to create supervised learning data from text?
  - Use bag-of-words models or tf-idf as features

# IMDB Movie Dataset

| Plot | Sci-Fi | Action |
|---|---|---|
| In a post-apocalyptic world in human decay, a ... | 1 | 0 |
| Mohei is a wandering swordsman. He arrives in ... | 0 | 1 |
| #137 is a SCI/FI thriller about a girl, Marla,... | 1 | 0 |

- Goal: Predict movie genre based on plot summary
- Categorical features generated using preprocessing

# Supervised learning steps

- Collect and preprocess our data
- Determine a label (Example: Movie genre)
- Split data into training and test sets
- Extract features from the text to help predict the label
  - Bag-of-words vector built into `scikit-learn`
- Evaluate trained model using the test set

# Let's practice!

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN PYTHON

# Building word count vectors with scikit-learn

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN PYTHON

**Katharine Jarmul**
Founder, kjamistan

datacamp

# Predicting movie genre

- Dataset consisting of movie plots and corresponding genre
- Goal: Create bag-of-word vectors for the movie plots
  - Can we predict genre based on the words used in the plot summary?

# Count Vectorizer with Python

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.feature_extraction.text import CountVectorizer
df = ... # Load data into DataFrame
y = df['Sci-Fi']
X_train, X_test, y_train, y_test = train_test_split(
                                                df['plot'], y,
                                                test_size=0.33,
                                                random_state=53)
count_vectorizer = CountVectorizer(stop_words='english')
count_train = count_vectorizer.fit_transform(X_train.values)
count_test = count_vectorizer.transform(X_test.values)
```

# Let's practice!

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN PYTHON

# Training and testing a classification model with scikit-learn

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN PYTHON

**Katharine Jarmul**
Founder, kjamistan

datacamp

# Naive Bayes classifier

- Naive Bayes Model
  - Commonly used for testing NLP classification problems
  - Basis in probability
- Given a particular piece of data, how likely is a particular outcome?
- Examples:
  - If the plot has a spaceship, how likely is it to be sci-fi?
  - Given a spaceship **and** an alien, how likely **now** is it sci-fi?
- Each word from `CountVectorizer` acts as a feature
- Naive Bayes: Simple and effective

# Naive Bayes with scikit-learn

```
from sklearn.naive_bayes import MultinomialNB
from sklearn import metrics
nb_classifier = MultinomialNB()

nb_classifier.fit(count_train, y_train)
pred = nb_classifier.predict(count_test)
metrics.accuracy_score(y_test, pred)
```

```
0.8584184938982042
```

# Confusion matrix

```
metrics.confusion_matrix(y_test, pred, labels=[0,1])
```

```
array([[6410,  563],
       [ 864, 2242]])
```

| | Action | Sci-Fi |
|---|---|---|
| Action | 6410 | 563 |
| Sci-Fi | 864 | 2242 |

# Let's practice!

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN PYTHON

# Simple NLP, complex problems

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN PYTHON

**Katharine Jarmul**
Founder, kjamistan

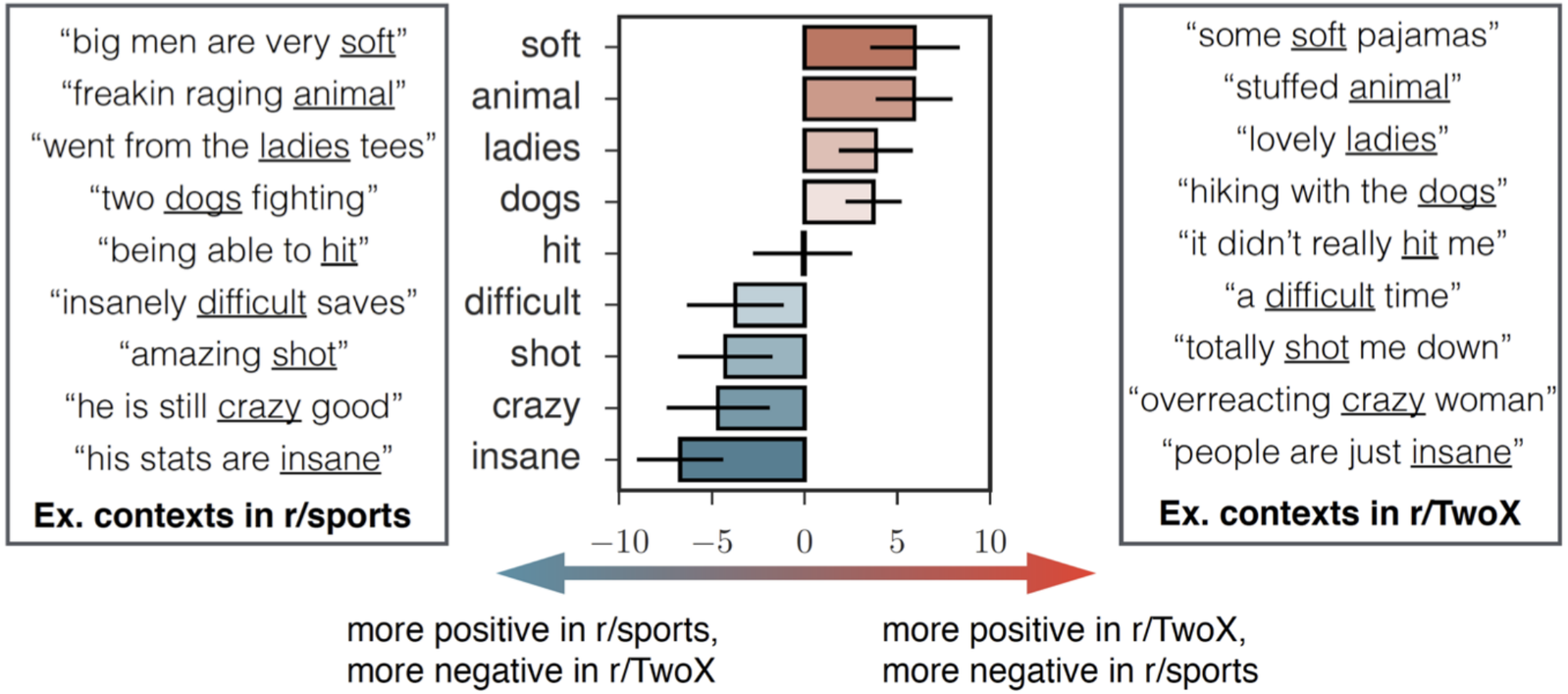
datacamp

# Translation

source:

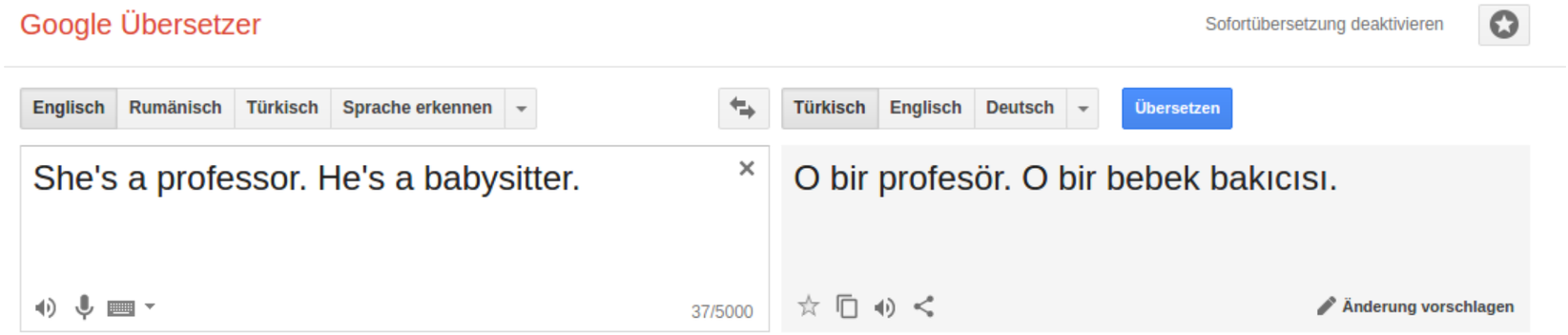
(https://twitter.com/Lupintweets/status/865533182455685121)

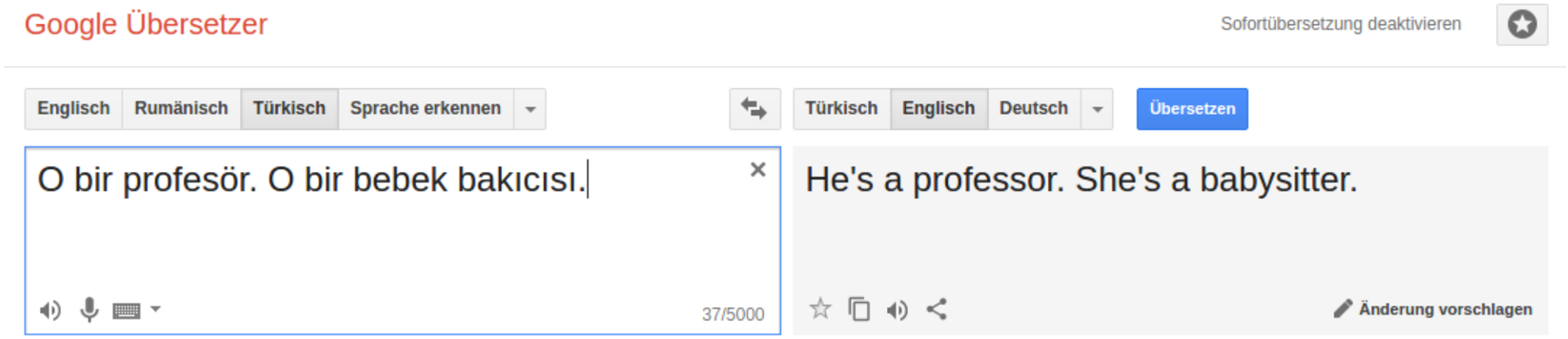
# Sentiment analysis




*(source: **https://nlp.stanford.edu/projects/socialsent/**)*

# Language biases




*(related talk: **https://www.youtube.com/watch?v=j7FwpZB1hWc**)*

# Let's practice!

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN PYTHON