

# Przetwarzanie Języka Naturalnego

**Porównanie różnych metod identyfikacji  
podobnych tekstów**

Mikhail Ermolaev  
David Burchakov

# Agenda

01

## Definicja problemu

kroki do sukcesu

02

Znajdowanie podobieństw w zbiorze danych

Harry Potter VS Star Wars

03

Identyfikacja podobnych tekstów

Gold fish VS fish gold

04

Generacja podobnych sentencji

Tworzenie nowych hitów literackich

# Definicja problemu

## Identyfikacja podobnych tekstów

- **Wyszukiwanie Podobnych Zdań z Bazy Danych**
- **Wyliczenie Stopnia Podobieństwa Dwóch Podanych Tekstów**
- **Generacja podobnych sentencji**

# Agenda

01

**Definicja problemu**

kroki do sukcesu

02

**Znajdowanie podobieństw w zbiorze danych**

Harry Potter VS Star Wars

03

**Identyfikacja podobnych tekstów**

Gold fish VS fish gold

04

**Generacja podobnych sentencji**

Tworzenie nowych hitów literackich

# Przetwarzanie wstępne

- **Remove Stopwords**
- **Expand Contractions**
- **Remove possessive endings**
- **Lemmatization**
- **Remove special characters**
- **Preserve capitalizations**
- **Capitalize Named Entities**
- **Tokenization**

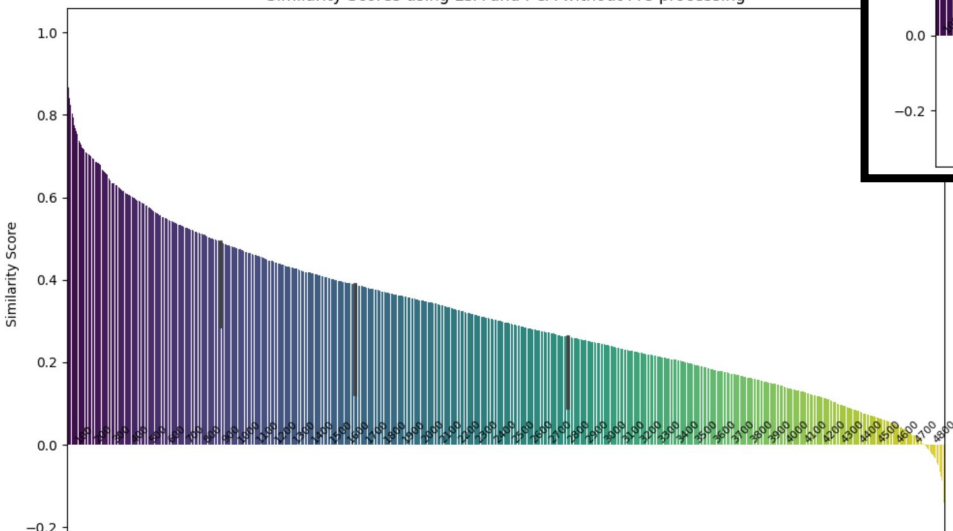
```
sentence = "This movie was truly amazing and had a great story."  
processed_sentence = process_sentence(sentence)  
print(processed_sentence)
```



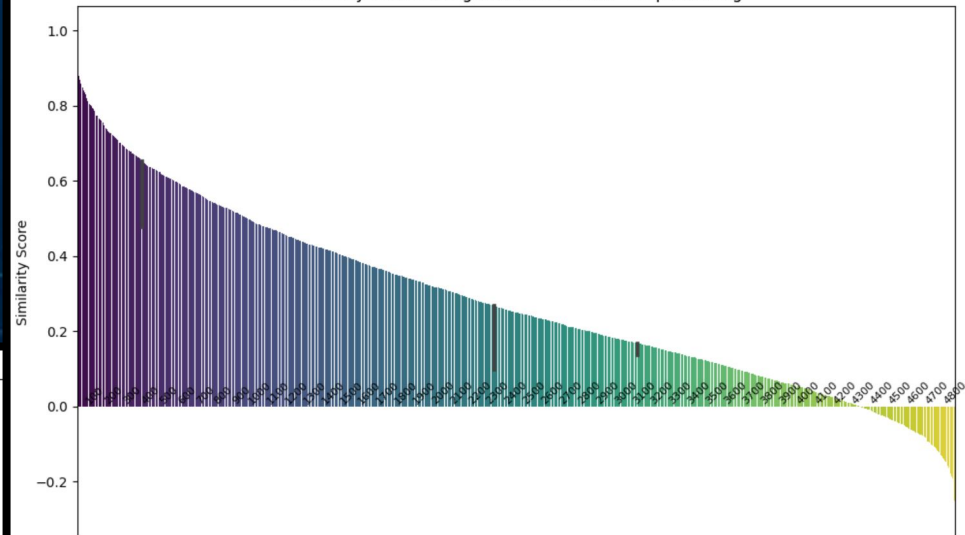
```
movie truly amazing great story .
```

# LSA & PCA

Similarity Scores using LSA and PCA without Pre-processing

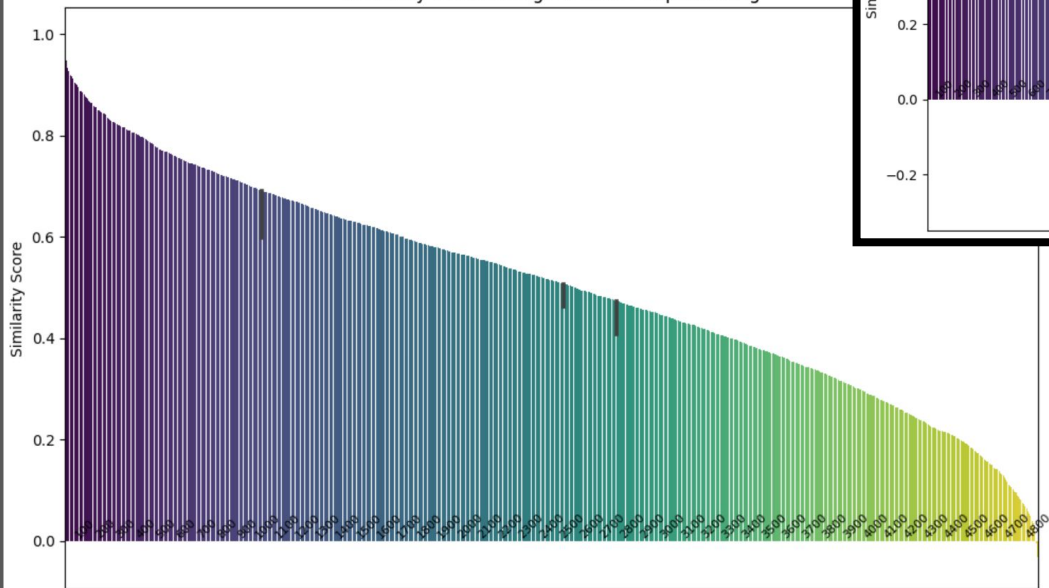


Similarity Scores using LSA and PCA with Pre-processing

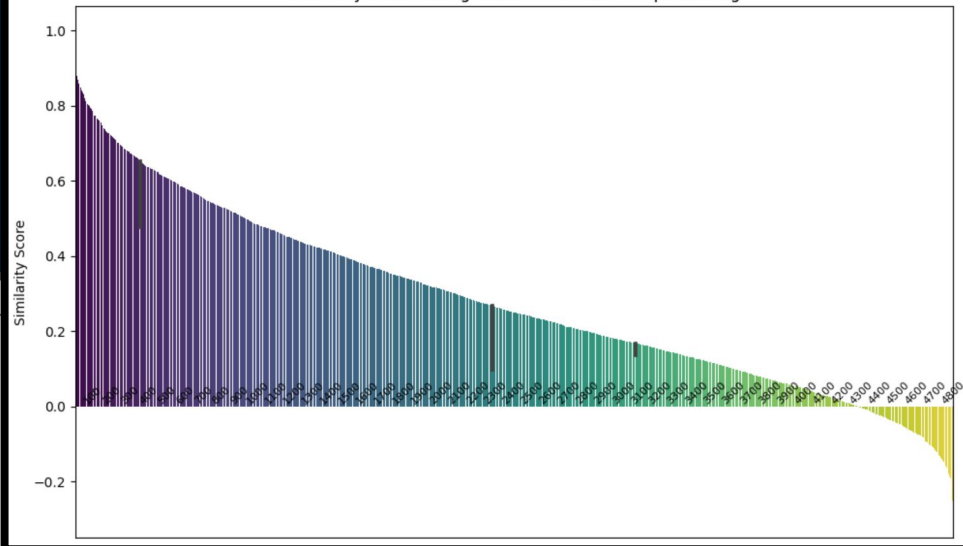


# LSA vs LSA & PCA

Similarity Scores using LSA with Pre-processing



Similarity Scores using LSA and PCA with Pre-processing



# Agenda

01

**Definicja problemu**

kroki do sukcesu

02

**Znajdowanie podobieństw w zbiorze danych**

Harry Potter VS Star Wars

03

**Identyfikacja podobnych tekstów**

Does M similar to W?

04

**Generacja podobnych sentencji**

Tworzenie nowych hitów literackich



# Identyfikacja podobnych tekstów

## GloVe: Global Vectors for Word Representation

Probability and Ratio	$k = \text{solid}$	$k = \text{gas}$	$k = \text{water}$	$k = \text{fashion}$
$P(k \text{ice})$	$1.9 \times 10^{-4}$	$6.6 \times 10^{-5}$	$3.0 \times 10^{-3}$	$1.7 \times 10^{-5}$
$P(k \text{steam})$	$2.2 \times 10^{-5}$	$7.8 \times 10^{-4}$	$2.2 \times 10^{-3}$	$1.8 \times 10^{-5}$
$P(k \text{ice})/P(k \text{steam})$	8.9	$8.5 \times 10^{-2}$	1.36	0.96

GloVe method looks for co-occurrence and relationships between words in the entire corpus.

For example, considering how frequently words like "ice" and "steam" co-occur with other words like "solid" or "gas".

The probe word "solid" is expected to co-occur more often with the word "ice", than the words "steam".

If the probe word is related to both or neither, e.g "water" and "fashion", the expected occurrence probability ratio is expected to be close to 1.

An unsupervised method

## Identyfikacja podobnych tekstów

GloVe

Ice



Steam



Solid

 $\rightarrow \infty$  $\rightarrow 0$ 

Gas

 $\rightarrow 0$  $\rightarrow \infty$ 

Water

 $\rightarrow 1$  $\rightarrow 1$

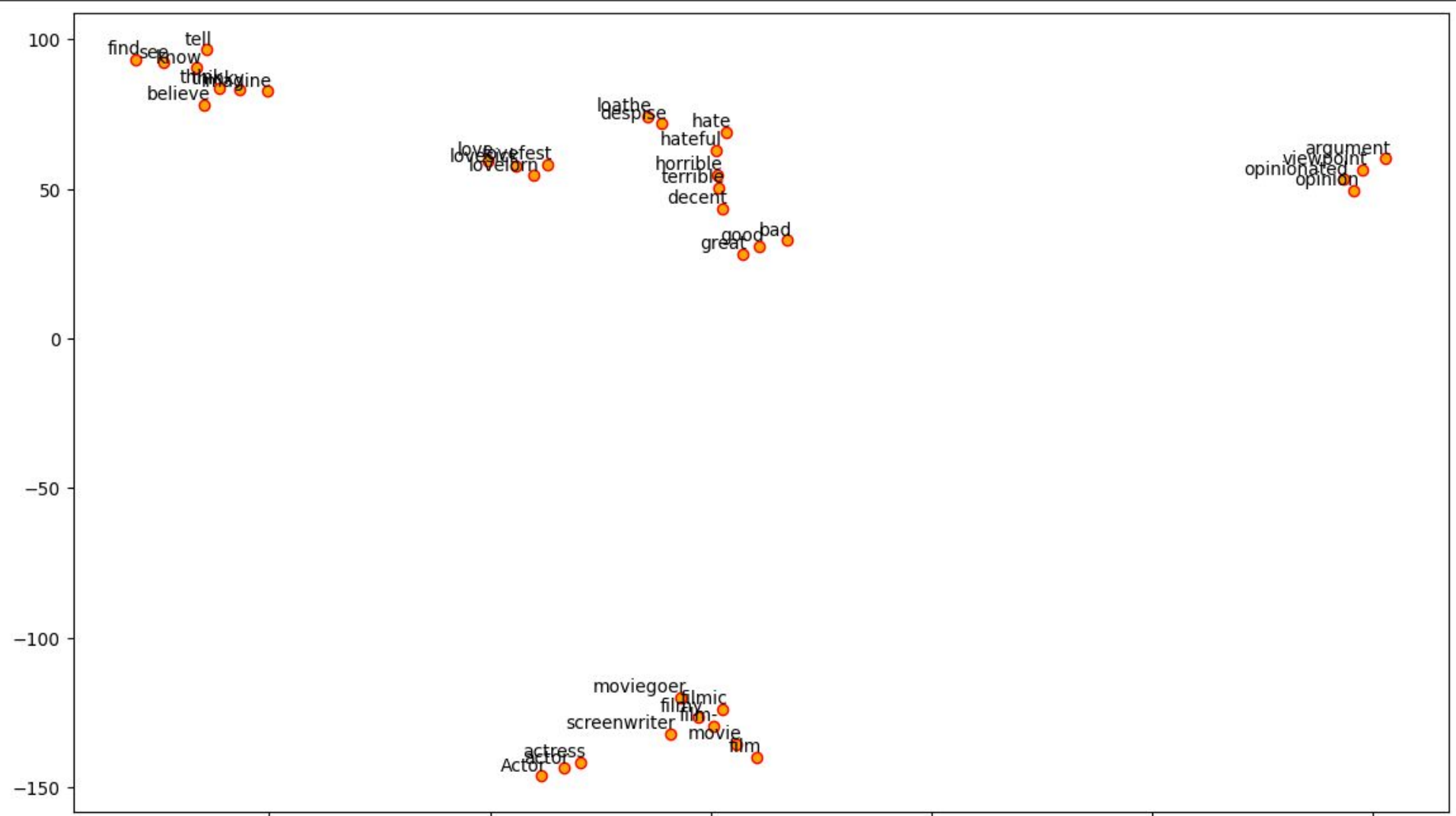
# Identyfikacja podobnych tekstów

GloVe

Użyty dataset



Freshness		Review
0	1	Manakamana doesn't answer any questions, yet makes its point: Nepal, like the rest of our planet, is a picturesque but far from peaceable kingdom.
1	1	Wilfully offensive and powered by a chest-thumping machismo, but it's good clean fun.
2	0	It would be difficult to imagine material more wrong for Spade than Lost & Found.
3	0	Despite the gusto its star brings to the role, it's hard to ride shotgun on Hector's voyage of discovery.
4	0	If there was a good idea at the core of this film, it's been buried in an unsightly pile of flatulence jokes, dog-related bad puns and a ridiculous serial arson plot.
5	0	Gleeson goes the Hallmark Channel route, damaging an intermittently curious entry in the time travel subgenre.
6	1	It was the height of satire in 1976: dark as hell, but patently absurd and surely nowhere close to objective reality. Objective reality surpassed it somewhere in the Jerry Springer era.
7	0	Everyone in "The Comedian" deserves a better movie than "The Comedian."
8	0	Actor encourages grumpy Christians to embrace the season.
9	1	Slight, contained, but ineffably soulful.



# Identyfikacja podobnych tekstów

GloVe

```
sentence1 = "This movie was truly amazing and had a great story."  
sentence2 = "The film was incredibly good and the plot was outstanding."
```

The similarity between the sentences is: 85.60%

	.	film	good	incredibly	outstanding	plot
.	1.000000	-0.009095	0.288674	0.266851	0.326775	0.109322
amazing	0.225753	0.084596	0.590898	0.567639	0.483716	0.051988
great	0.300736	0.082885	0.755130	0.475666	0.592463	0.069649
movie	0.023211	0.833277	0.139775	0.183301	0.016186	0.445783
story	0.096732	0.524190	0.114741	0.297797	0.233712	0.598915
truly	0.173340	0.081420	0.387853	0.652918	0.313164	0.179490
Average Similarity:	0.32					



# Identyfikacja podobnych tekstów

GloVe

```
sentence1 = "This movie was truly amazing and had a great story."  
sentence2 = "My cat is very biting, so I never pick her up"
```

The similarity between the sentences is: 51.66%

	,	biting	cat	never	pick
.	0.425803	0.378945	0.014464	0.278176	0.183916
amazing	0.115804	0.227776	0.121694	0.324187	0.230786
great	0.269700	0.350155	0.054076	0.357812	0.166819
movie	0.006554	0.066448	0.118379	0.233778	0.071128
story	0.232640	0.246280	0.078144	0.282937	-0.034338
truly	0.055032	0.139608	0.139030	0.448159	0.104447
Average Similarity: 0.19					

# Identyfikacja podobnych tekstów

## *“Generacja podobnych tekstów”*

GloVe

Input:

The movie is a great plot, the director is a good person.  
The actors performed very well. Probably the best film of 2014!

Output:

the moviegoer be a fantastic plotline , the codirector be a  
great personhood . the Actress performative extremely welly .  
Improbably the great filmy of 2014 !

Old sentence: The movie is a great plot, the director is a good person. The actors performed very well. Probably the best film of 2014!

New sentence: the moviegoer be a fantastic plotline , the codirector be a great personhood . the Actress performative extremely welly ! Improbably the great filmy of 2014 !

# Identyfikacja podobnych tekstów

## FastText

```
# Load the pre-trained FastText model
model_en = fasttext.load_model('cc.en.300.bin')

sentence1 = "The movie was fantastic and thrilling."
sentence2 = "I found the film to be exciting and wonderful."
```

Similarity: 68.79260540008545%

	i	found	the	film	to	be	exciting	and	wonderful.
the	0.141125	0.258783	1	0.2078	0.373638	0.261259	0.164036	0.43708	0.062681
movie	0.131207	0.075474	0.191173	0.773174	0.045066	0.074241	0.150432	0.090201	0.072723
was	0.267145	0.490676	0.358188	0.169538	0.248655	0.43198	0.197481	0.422343	0.254178
fantastic	0.132722	0.220809	0.217391	0.164246	0.092706	0.211421	0.559153	0.244111	0.485977
and	0.233203	0.275322	0.43708	0.101709	0.364097	0.314614	0.198858	1	0.220625
thrilling.	0.45481	0.086195	0.099228	0.140663	0.051223	0.213135	0.381754	0.17585	0.668105



# Identyfikacja podobnych tekstów

## *“Generacja podobnych tekstów”*

FastText

Input:

The movie is a great plot, the director is a good person. The actors performed very well. Probably the best film of 2014!

Output:

ththe film It A fantastic Pejeta ththe co-director It A bad individual.  
ththe actresses peformed extremely too. probaby ththe finest films  
ofn 201588

Input:

The movie is a great plot, the director is a good person. The actors performed very well. Probably the best film of 2014!

Output:

ththe film It A fantastic Pejeta ththe co-director It A bad individual. ththe actresses peformed extremely too. probaby ththe finest films of  
n 201588

# Agenda

01

**Definicja problemu**

kroki do sukcesu

02

**Znajdowanie podobieństw w zbiorze danych**

Harry Potter VS Star Wars

03

**Identyfikacja podobnych tekstów**

Gold fish VS fish gold

04

**Generacja podobnych sentencji**

Tworzenie nowych hitów literackich

# Generowanie podobnych zdań

*dataset used: Opusparcus*

NLG

Opusparcus: Open Subtitles Paraphrase Corpus for Six Languages (version 1.0) 📄

	text	paraphrase
0	You 're not alone , Claire .	You are not alone , Claire .
1	Who told you to throw acid at Vargas , hmm ?	Who told you to throw acid at Vargas ?
2	Where the pure angel merges with the antic Sphinx	Where the pure angel merges with the antic sphynx .
3	Where is it written what is it I 'm meant to be	Where is it written what it is I 'm meant to be
4	We 'll find the skipper and then we 'll go home .	We 'll find the skipper and then we go home .
5	Seymour 's Darling is third ... and little Arnie moving fast on the outside .	Seymour 's Darling is third ... and little Arnie moving fast to the outside .
6	Scud , do you read me ?	Scud , you reading me ?
7	Jumby now wants to be born .	Jumby want birth .
8	It was a difficult and long delivery .	The delivery was difficult and long .
9	It 's a shit , but it 's better than nothing , right ?	It 's a shit , but it 's better that nothing , right ?

# Generowanie podobnych zdań

## *NLP with NN*

*First try: untrained*

```
# Building the Model
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(vocab_size, embedding_dim, mask_zero=True),
    tf.keras.layers.GRU(rnn_units, return_sequences=True),
    tf.keras.layers.Dense(vocab_size, activation='softmax')
])

model.compile(optimizer='adam', loss='sparse_categorical_crossentropy')

# Train the Model
model.fit(dataset, epochs=20)
```

# Generowanie podobnych zdań

## *NLP with NN*

**NLG***First try: untrained*

```
input_text = "I 'm at the bridge"  
generated_sentence = generate_similar_sentence(input_text, model, tokenizer, max_seq_length)  
print("Generated sentence:", generated_sentence)
```

```
1/1 [=====] - 0s 21ms/step
```

```
Generated sentence: i am in the bridge the the the the the the the the the the the the the the the the the
```

# Generowanie podobnych zdań

## *NLP with NN*

**NLG***First try: untrained*

```
input_text = "I 'm going to have a son"  
generated_sentence = generate_similar_sentence(input_text, model, tokenizer, max_seq_length)  
print("Generated sentence:", generated_sentence)
```

1/1 [=====] - 0s 28ms/step

Generated sentence: i am gonna to a baby . the the the the the the the the the the the the the the the the the



## First try: untrained

1/1 [=====] - 0s 20ms/step

[illegible]





# Generowanie podobnych zdań

## *NLP with NN*

*Second try: overfitted*

Run

22898.5s - GPU T4 x2

```
# Model with more complexity and regularization
model = tf.keras.Sequential([
    Embedding(vocab_size, embedding_dim, mask_zero=True),
    Bidirectional(GRU(rnn_units, return_sequences=True, recurrent_dropout=0.2)),
    Dropout(0.2),
    Dense(vocab_size, activation='softmax')
])

# Early stopping to prevent overfitting
early_stopping = EarlyStopping(monitor='loss', patience=3)

model.compile(optimizer='adam', loss='sparse_categorical_crossentropy')

# Train with early stopping
model.fit(dataset, epochs=50, callbacks=[early_stopping])
```

# Generowanie podobnych zdań

## *NLP with NN*

**NLG***Second try: overfitted*

```
input_text = "i am at the bridge"  
generated_sentence = generate_similar_sentence(input_text, model, tokenizer, max_seq_length)  
print("Generated sentence:", generated_sentence)
```

```
1/1 [=====] - 0s 54ms/step
```

```
Generated sentence: i am on the bridge
```

# Generowanie podobnych zdań

## *NLP with NN*

**NLG***Second try: overfitted*

```
input_text = "I 'm going to have a son"  
generated_sentence = generate_similar_sentence(input_text, model, tokenizer, max_seq_length)  
print("Generated sentence:", generated_sentence)
```

```
1/1 [=====] - 0s 54ms/step
```

```
Generated sentence: i gonna gonna have son son
```

# Generowanie podobnych zdań

## *NLP with NN*

**NLG***Second try: overfitted*

```
input_text = "Are you in army"  
generated_sentence = generate_similar_sentence(input_text, model, tokenizer, max_seq_length)  
print("Generated sentence:", generated_sentence)
```

```
1/1 [=====] - 0s 54ms/step
```

```
Generated sentence: are you in military
```

# Agenda

01

Definicja problemu  
kroki do sukcesu

02

Znajdowanie podobieństw w zbiorze danych  
Harry Potter VS Star Wars

03

Identyfikacja podobnych tekstów  
Gold fish VS fish gold

04

Generacja podobnych sentencji  
Tworzenie nowych hitów literackich

05

**Podsumowanie**

Word2Vec vs. GloVe vs. FastText vs. LSA vs. PCA vs. NN

# Najlepsza metoda?

*Word2Vec?*

*PCA?*

***BOW?***

*FastTest?*

*GloVe?*

*RNN?*

*LSA?*

