

**Dr. Keras or:
How I Learned to Stop Worrying
and Love NLP**

Dalya Gartzman

Dr. Keras or: How I Learned to Stop Worrying and Love NLP

Dalya Gartzman

**TeMS -
Textual Math Solver**

Simplisico

Dr. Keras or: How I Learned to Stop Worrying and Love NLP

Dalya Gartzman

 **seematics**

VISIONARY
DEEP LEARNING


**TeMS -
Textual Math Solver**

Simplisico

Dr. Keras or: How I Learned to Stop Worrying and Love NLP

Dalya Gartzman



seematics
VISIONARY
DEEP LEARNING

**TeMS -
Textual Math Solver**
Simplisico

PART I - What is TeMS?

PART I - What is TeMS?

The image shows a screenshot of the Simplisico website, which is a math tutor. The website has a red header with the Simplisico logo and navigation links. The main content area features a large video player with a play button. To the left of the video player, there is a sidebar with a list of topics and a man standing in front of a chalkboard. The chalkboard is filled with various mathematical formulas and diagrams. A green button with the text "Try it for free" is located at the bottom of the video player.

Simplisico LTD [IL] | https://www.simplisico.com

English Login Sign up

YOUR PRIVATE MATH TUTOR

Understand · Simple · Solve

INTERSECTION WITH THE Y AXIS

When does the function cross the y-axis? That is, what is the y-intercept of the function? This is the value of the function when the input is zero. In other words, we are looking for the value of the function when $x=0$.

As you can see, a function has one y-intercept. This is the point where the function crosses the y-axis. This is the point where the function has the same x and y coordinates. This is the point where the function has the same x and y coordinates. This is the point where the function has the same x and y coordinates.

Plugging into X

Plug into the calculator with a calculator if you want the result.

Using the calculator

$2x+1=0$

$x=-0.5$

x -intercept

$(-0.5, 0)$

Try it for free

PART I - What is TeMS?



The screenshot shows the Simplisico website interface. At the top, there's a navigation bar with the logo "simplisico math made easy" and a user profile icon. Below the navigation bar, there's a prominent orange button that says "Subscribe now" with a text overlay: "to get unlimited access to Simplisico's extended explanations, a full month costs much less than a private tutoring hour!".

Below the button, there's a text prompt: "Enter the equation you have to solve using the keyboard or on-screen buttons, when finished press 'Show Solution' and learn how to explore your function.".

The main area features a large input field where the equation $f(x) = \frac{5x+6}{2x-3}$ has been entered. To the right of the input field is a directional pad icon. Below the input field is a calculator interface with a grid of buttons. The buttons include mathematical symbols like x , $\frac{\square}{\square}$, \square^2 , $\sqrt{\square}$, \square^3 , $\sqrt[3]{\square}$, and a row of numbers 7, 8, 9, 4, 5, 6, 1, 2, 3, 0. There are also buttons for basic operations: \times , $-$, $+$, and a row of functions: \sin , \cos , \ln , \log .

At the bottom right of the calculator interface is a blue button labeled "Show Solution >".

PART I - What is TeMS?

The screenshot shows a web browser window with the URL <https://www.simplisico.com>. The page features the Simplisico logo and navigation links. A math problem is displayed on a slide:

**I have 10 minutes to talk and I prepared 42 slides.
How long can I spend on each slide?**

The equation $42 * x = 10$ is shown in a blue box.

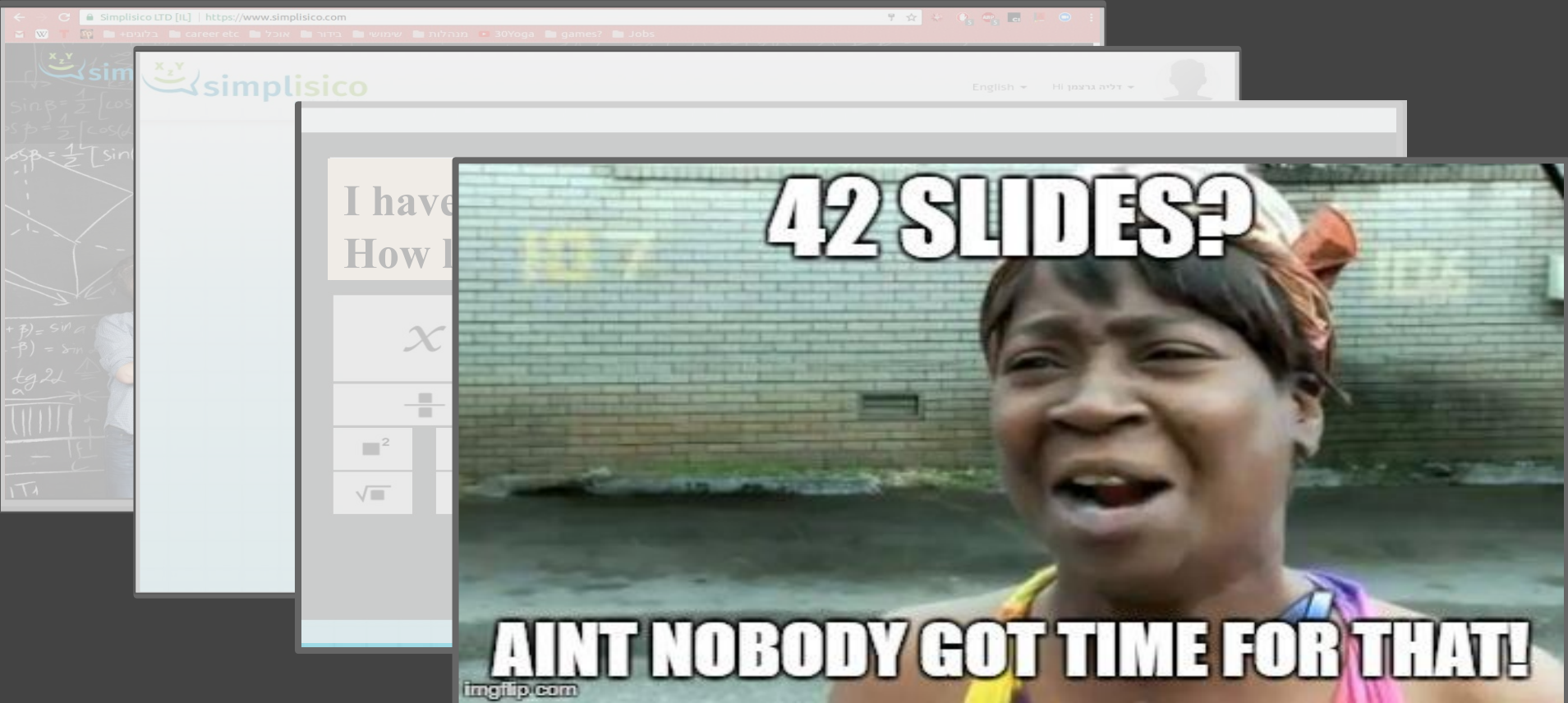
Below the problem is a calculator interface with the following layout:

C		$\leftarrow \times$	()	\div
7	8	9	\times	
4	5	6	-	
1	2	3	+	
0	sin / cos / ln / log		.	

On the left side of the calculator, there are additional buttons for x , $\frac{\square}{\square}$, \square^2 , $\sqrt{\square}$, and $\sqrt[n]{\square}$.

A blue button labeled **Show Solution >** is located at the bottom right of the slide.

PART I - What is TeMS?



Preprocessing

Preprocessing

I have 10 minutes to talk and I prepared 42 slides. How long can I spend on each slide?

Preprocessing

I have **A** minutes to talk and I prepared **B** slides. How long can I spend on each slide?

Preprocessing

I have **A** minutes to talk and I prepared **B** slides. How long can I spend on each slide?

TOKENIZE [10, 42, 0] +
[... 'have', '**varA**', 'minutes', ...
'prepared', '**varB**', 'slides', ...]

Preprocessing

I have **A** minutes to talk and I prepared **B** slides. How long can I spend on each slide?

TOKENIZE [10, 42, 0] +
[... 'have', '**varA**', 'minutes', ...
'prepared', '**varB**', 'slides', ...]

EMBED

tree → park
slides → talk

TeMS: a Sequence-to-Permutation RNN with LSTM and Attention

TeMS: a Sequence-to-Permutation RNN with LSTM and Attention

**I have 10 minutes to talk and
I prepared 42 slides. How
long can I spend on each
slide?**

TeMS: a Sequence-to-Permutation RNN with LSTM and Attention

I have 10 minutes to talk and
I prepared 42 slides. How
long can I spend on each
slide?

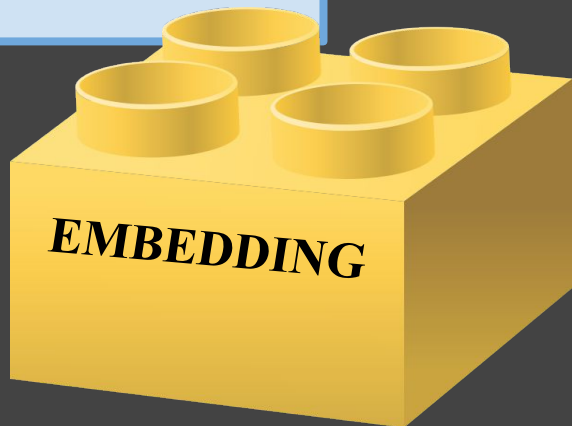
[42, 10, 0]



$42 * x = 10$

TeMS: a Sequence-to-Permutation RNN with LSTM and Attention

I have 10 minutes to talk and
I prepared 42 slides. How
long can I spend on each
slide?



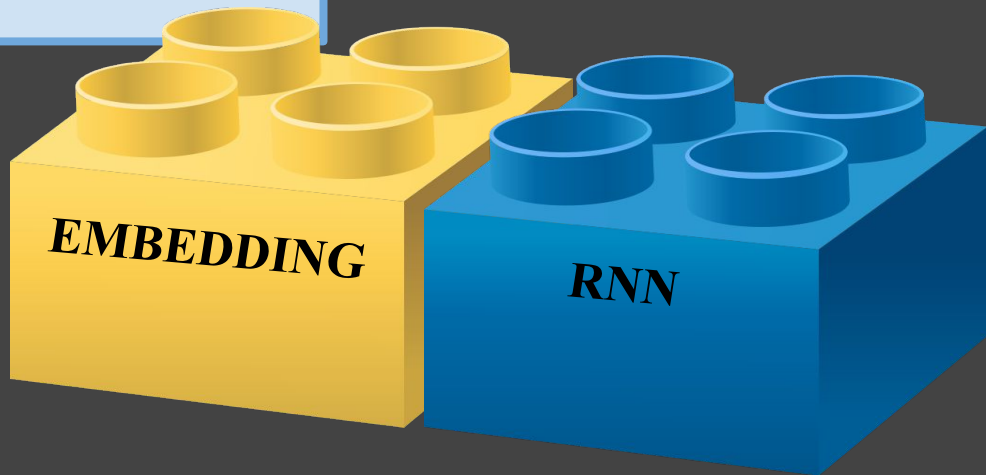
[42, 10, 0]



$42 * x = 10$

TeMS: a Sequence-to-Permutation RNN with LSTM and Attention

I have 10 minutes to talk and
I prepared 42 slides. How
long can I spend on each
slide?



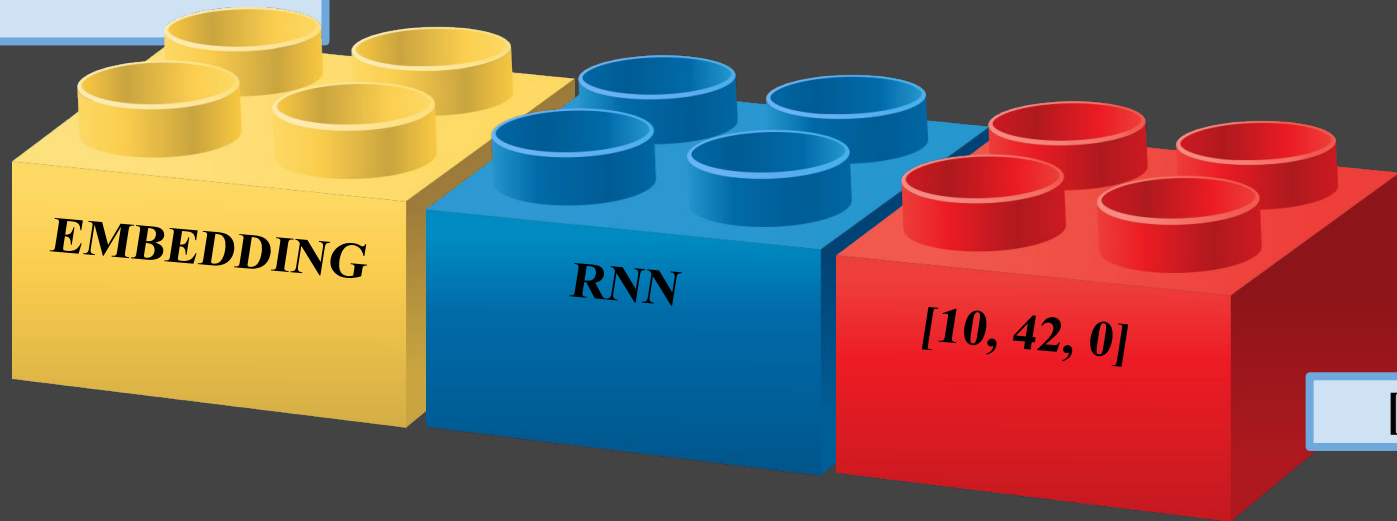
[42, 10, 0]

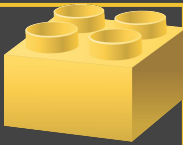


$42 * x = 10$

TeMS: a Sequence-to-Permutation RNN with LSTM and Attention

I have 10 minutes to talk and
I prepared 42 slides. How
long can I spend on each
slide?





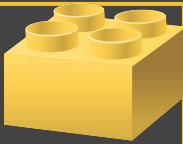
```
# generate embeddings
embedding = w2v.Word2Vec()
embedding.build_vocab(all_questions)
embedding.train(all_questions)
```

```
# Words model
words_model = Sequential()
words_model.add(Embedding(vocab_size, embedding_dim, weights=[embedding]))
words_model.add(LSTM(64, activation='softmax', return_sequences=True, dropout=0.5))
words_model.add(LSTM(64, activation='relu', return_sequences=False, dropout=0.5))
words_input = Input(shape=(max_len,))
processed_words = words_model(words_input)
```

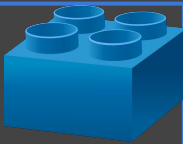
```
# Numbers model
nums_model = Sequential()
nums_model.add(Dense(3, input_dim=3))
nums_input = Input(shape=(3,))
processed_nums = nums_model(nums_input)
```

```
# Concatenate models
merged = keras.layers.concatenate([processed_nums, processed_words])
```

```
# add one dense layer to integrate the merging
hidden = Dense(32, activation='tanh')(merged)
hidden = Dropout(0.5)(hidden)
# finish with a dense layer
output = Dense(3)(hidden)
```



```
# generate embeddings
embedding = w2v.Word2Vec()
embedding.build_vocab(all_questions)
embedding.train(all_questions)
```



```
# Words model
words_model = Sequential()
words_model.add(Embedding(vocab_size, embedding_dim, weights=[embedding]))
words_model.add(LSTM(64, activation='softmax', return_sequences=True, dropout=0.5))
words_model.add(LSTM(64, activation='relu', return_sequences=False, dropout=0.5))
words_input = Input(shape=(max_len,))
processed_words = words_model(words_input)
```

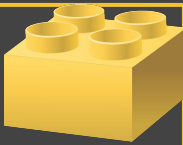
```
# Numbers model
nums_model = Sequential()
nums_model.add(Dense(3, input_dim=3))
nums_input = Input(shape=(3,))
processed_nums = nums_model(nums_input)

# Concatenate models
merged = keras.layers.concatenate([processed_nums, processed_words])

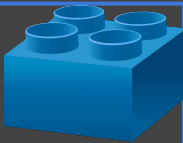
# add one dense layer to integrate the merging
hidden = Dense(32, activation='tanh')(merged)
hidden = Dropout(0.5)(hidden)

# finish with a dense layer
output = Dense(3)(hidden)
```

```
# generate embeddings
embedding = w2v.Word2Vec()
embedding.build_vocab(all_questions)
embedding.train(all_questions)
```

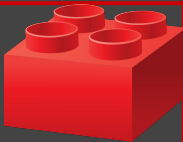


```
# Words model
words_model = Sequential()
words_model.add(Embedding(vocab_size, embedding_dim, weights=[embedding]))
words_model.add(LSTM(64, activation='softmax', return_sequences=True, dropout=0.5))
words_model.add(LSTM(64, activation='relu', return_sequences=False, dropout=0.5))
words_input = Input(shape=(max_len,))
processed_words = words_model(words_input)
```



```
# Numbers model
nums_model = Sequential()
nums_model.add(Dense(3, input_dim=3))
nums_input = Input(shape=(3,))
processed_nums = nums_model(nums_input)

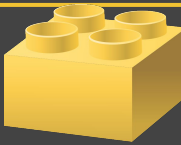
# Concatenate models
merged = keras.layers.concatenate([processed_nums, processed_words])
```



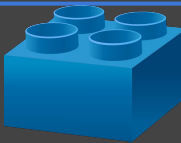
```
# add one dense layer to integrate the merging
hidden = Dense(32, activation='tanh')(merged)
hidden = Dropout(0.5)(hidden)
# finish with a dense layer
output = Dense(3)(hidden)
```



```
# generate embeddings
embedding = w2v.Word2Vec()
embedding.build_vocab(all_questions)
embedding.train(all_questions)
```

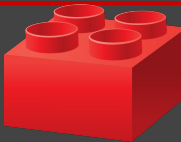


```
# Words model
words_model = Sequential()
words_model.add(Embedding(vocab_size, embedding_dim, weights=[embedding]))
words_model.add(LSTM(64, activation='softmax', return_sequences=True, dropout=0.5))
words_model.add(LSTM(64, activation='relu', return_sequences=False, dropout=0.5))
words_input = Input(shape=(max_len,))
processed_words = words_model(words_input)
```



```
# Numbers model
nums_model = Sequential()
nums_model.add(Dense(3, input_dim=3))
nums_input = Input(shape=(3,))
processed_nums = nums_model(nums_input)

# Concatenate models
merged = keras.layers.concatenate([processed_nums, processed_words])
```



```
# add one dense layer to integrate the merging
hidden = Dense(32, activation='tanh')(merged)
hidden = Dropout(0.5)(hidden)
# finish with a dense layer
output = Dense(3)(hidden)
```

Results

Input:

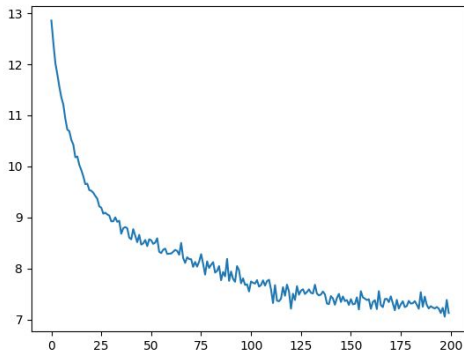
['You have 5 apples...',
'23 years from now...',
'Sean collected 47 stamps...',
'A cup of coffee costs 13.5\$...',
'Fourteen drinks were served ...']

Prediction:

$5 * x = 39 - 19$
 $2 * x = 23 - 16.8$
 $5 * x = 47 - 17$
 $2 * x = 13.5 - 3.5$
 $4 * x = 14 - 6$

Truth:

$5 * x = 39 - 19$
 $2 * x = 23 - 16.8$
 $5 * x = 47 - 17$
 $2 * x = 13.5 - 3.5$
 $-4 * x = 6 - 14$

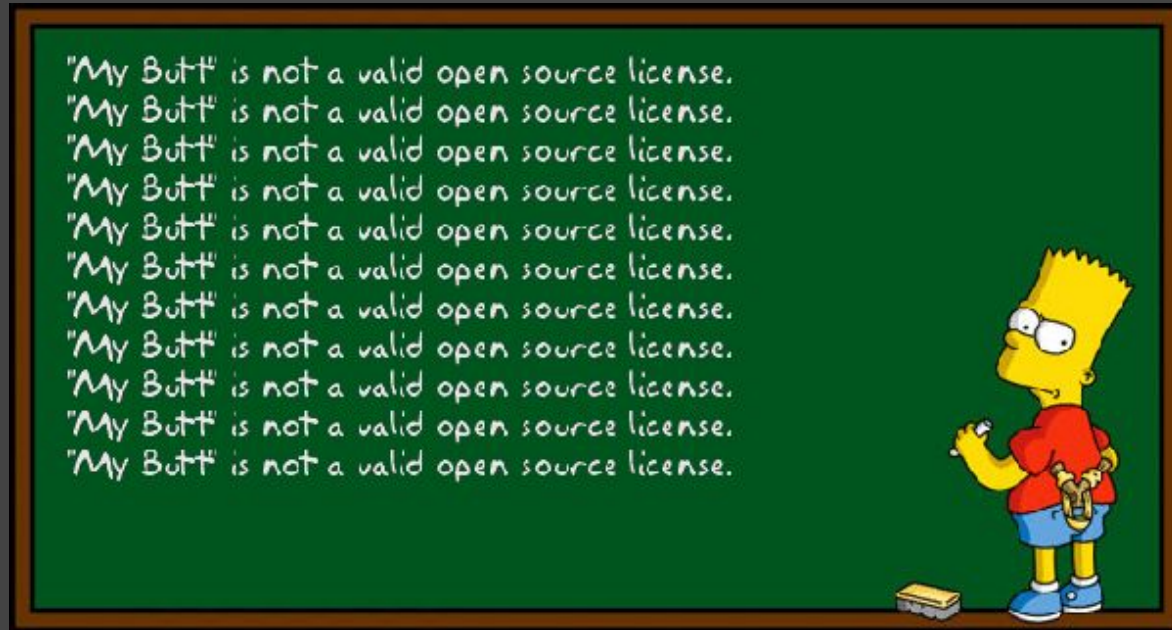


PART I Recap



PART II -

How to Reach a Wider Audience



Tip #1 -



Documentation, Documentation, Documentation?!?

Tip #1 -

Documentation, Documentation, Documentation?!?

Intro/Tutorial



Tip #1 -

Documentation, Documentation, Documentation?!?

Documentation



Intro/Tutorial



Tip #1 -



Documentation, Documentation, Documentation?!?

Support



Documentation



Intro/Tutorial



Tip #1 -

Documentation, Documentation, Documentation?!?

Intro/Tutorial



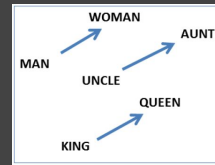
Documentation



Support



Tip #2 - Think up a killer app



Linear relationships between word pairs

```
In [37]: def nearest_similarity_cosmul(start1, end1, end2):
similarity = thrones2vec.most_similar_cosmul(
    positive=[end2, start1],
    negative=[end1]
)
start2 = similarity[0][0]
print("{} is related to {}, as {} is related to {}".format(start1, end1, start2, end2))
return start2

In [38]: nearest_similarity_cosmul("Stark", "Winterfell", "Riverrun")
nearest_similarity_cosmul("Jaime", "sword", "wine")
nearest_similarity_cosmul("Arya", "Nymeria", "dragons")
Stark is related to Winterfell, as Tully is related to Riverrun
Jaime is related to sword, as Tyrion is related to wine
Arya is related to Nymeria, as Dany is related to dragons
```

How to Make Word Vectors from Game of Thrones (LIVE)

Siraj Raval

162K

רשום כענני

25,539 צפיות

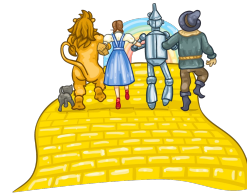
Tip #3 - Dare to be Mainstream



PART II Recap

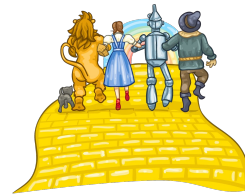
PART II Recap

Friendly



PART II Recap

Friendly



Fun

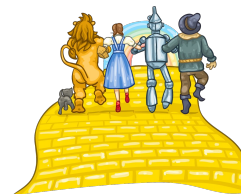


PART II Recap

Accessible



Friendly



Fun



Take Home Message




Thank you :)
Questions?

Dr. Keras or:
How I Learned to Stop Worrying and Love NLP

Dalya Gartzman



 **seematics**
VISIONARY
DEEP LEARNING

TeMS -
Textual Math Solver
Simplisico