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

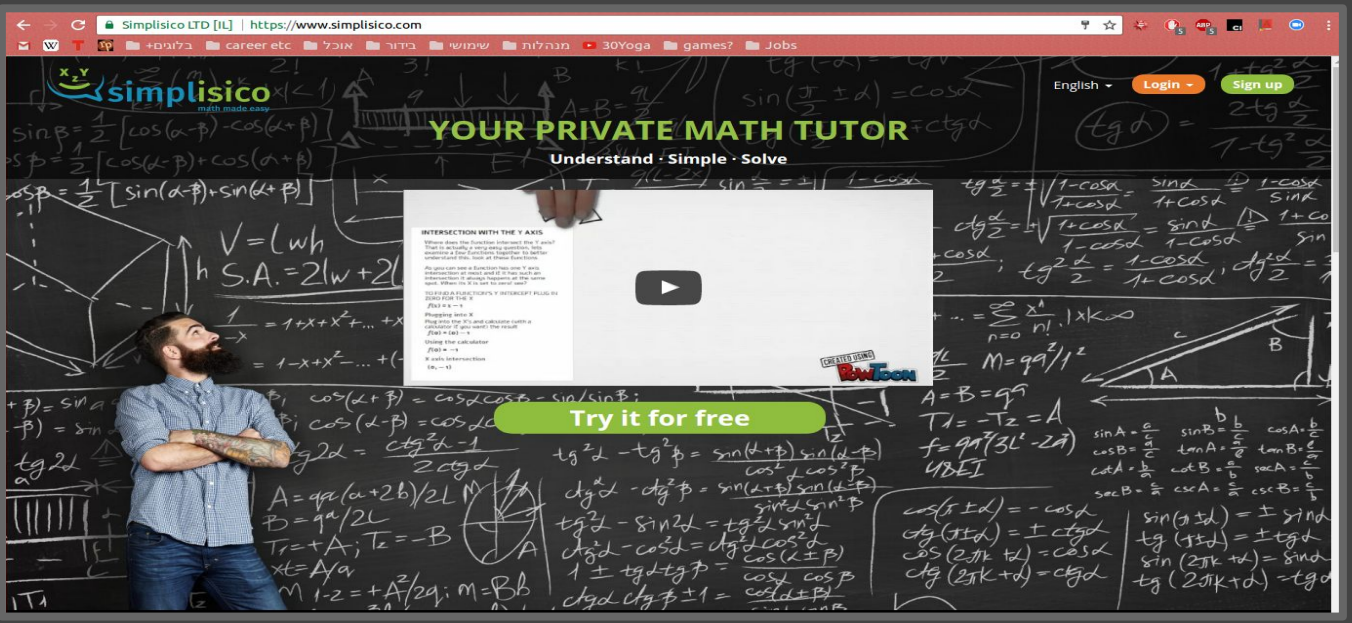
seematics
VISIONARY
DEEP LEARNING

TeMS -
Textual Math Solver

Simplisico

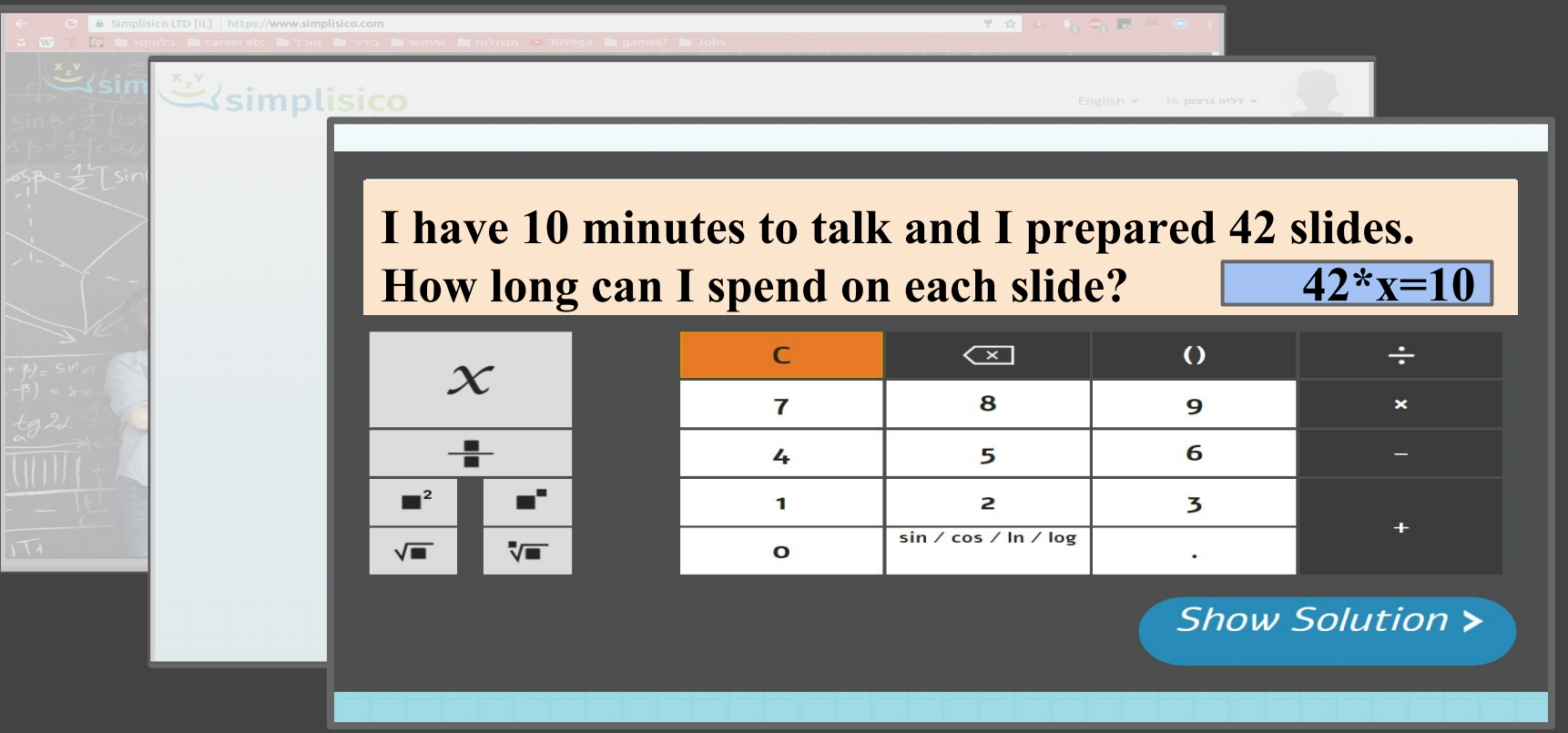
PART I - What is TeMS?

PART I - What is TeMS?

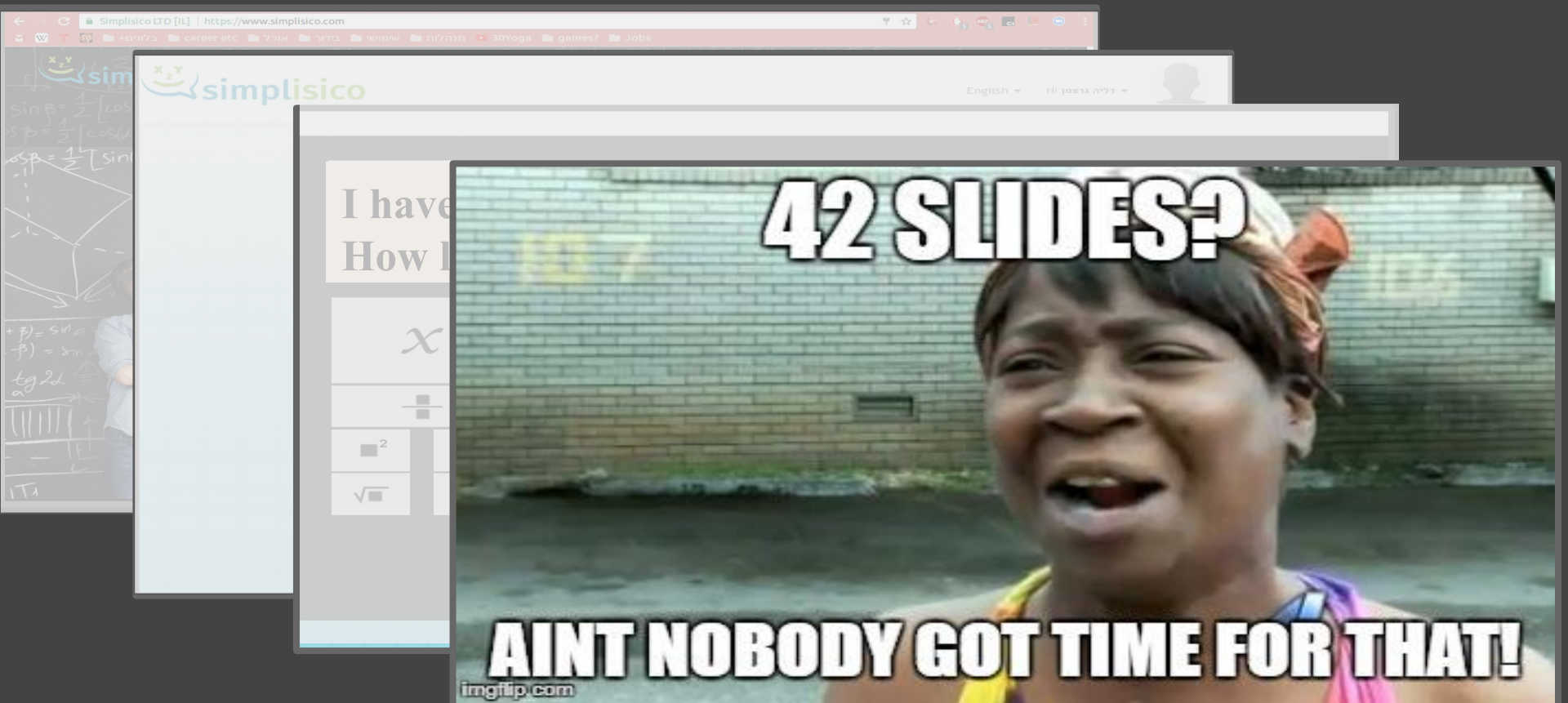


PART I - What is TeMS?

PART I - What is TeMS?



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

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

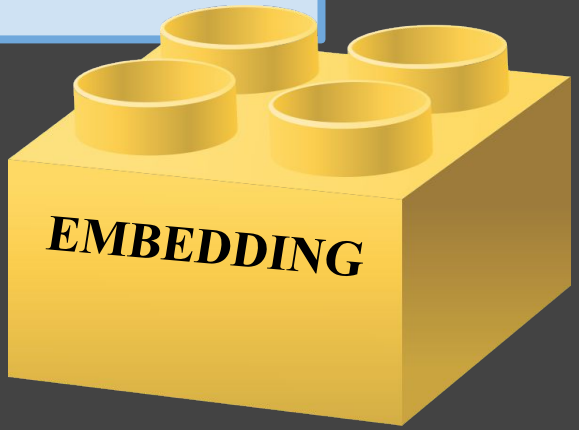
I have 10 minutes to talk and I prepared 42 slides. How long can I spend on each slide?

[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?

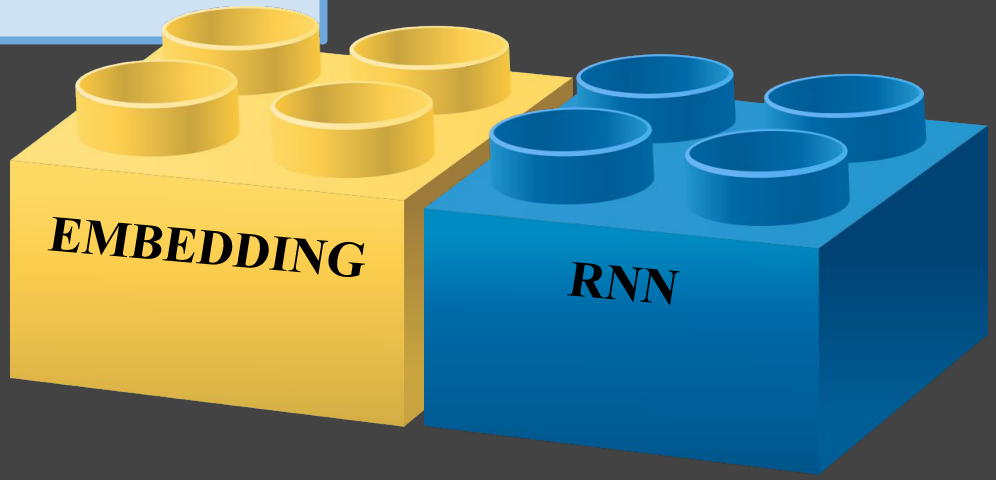


[42, 10, 0]

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?

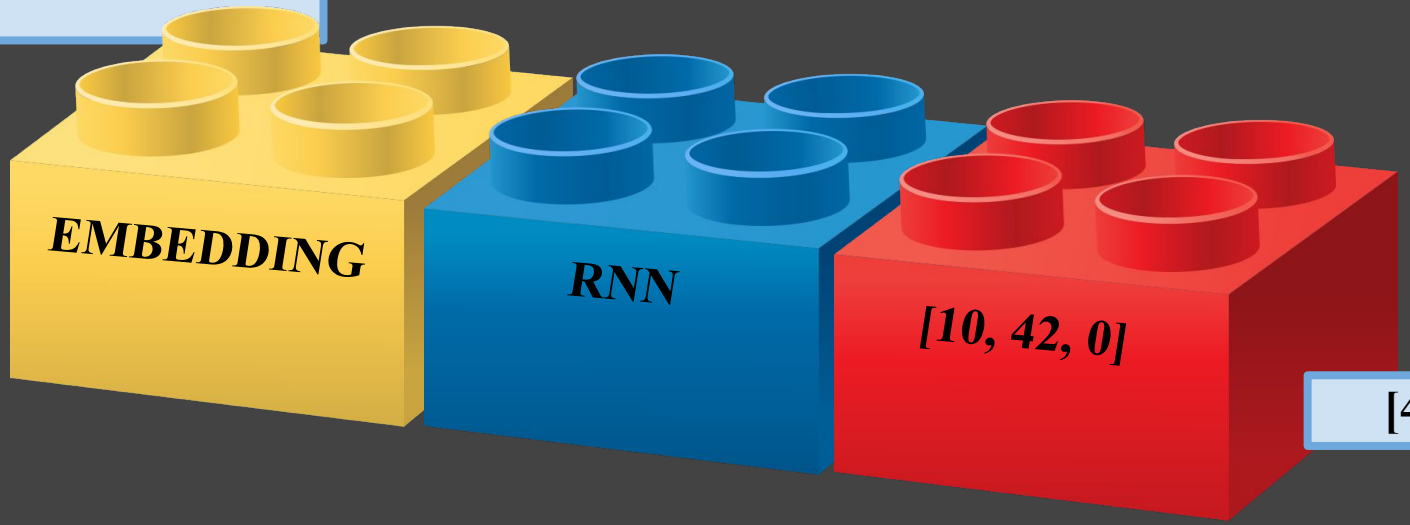


[42, 10, 0]

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?

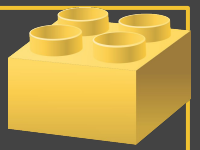


[42, 10, 0]

42*x = 10

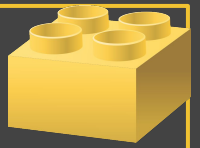
```
# generate embeddings
embedding = w2v.Word2Vec()
embedding.build_vocab(all_questions)
embedding.train(all_questions)

# 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)
```

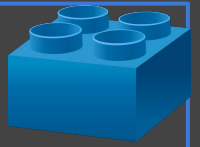


```
# Words model
words_model = Sequential()
words_model.add(Embedding(vocab_size, embedding_dim, weights=[embedding]))
words_model.add(LSTM(64, activation='softmax', return_sequences=True, dropout=0.5))
words_model.add(LSTM(64, activation='relu', return_sequences=False, dropout=0.5))
words_input = Input(shape=(max_len,))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

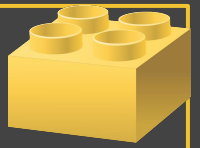


```
# Words model
words_model = Sequential()
words_model.add(Embedding(vocab_size, embedding_dim, weights=[embedding]))
words_model.add(LSTM(64, activation='softmax', return_sequences=True, dropout=0.5))
words_model.add(LSTM(64, activation='relu', return_sequences=False, dropout=0.5))
words_input = Input(shape=(max_len,))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

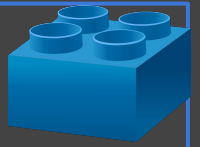


```
# Numbers model
nums_model = Sequential()
nums_model.add(Dense(3, input_dim=3))
nums_input = Input(shape=(3,))
processed_nums = nums_model(nums_input)

# 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)
```

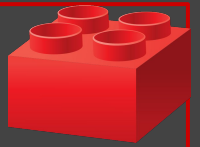
```
# generate embeddings
embedding = w2v.Word2Vec()
embedding.build_vocab(all_questions)
embedding.train(all_questions)
```



```
# 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)
```

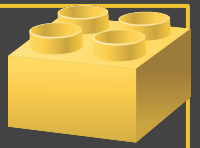


```
# Numbers model
nums_model = Sequential()
nums_model.add(Dense(3, input_dim=3))
nums_input = Input(shape=(3,))
processed_nums = nums_model(nums_input)

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

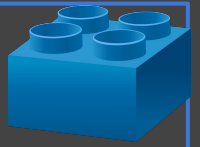


```
# add one dense layer to integrate the merging
hidden = Dense(32, activation='tanh')(merged)
hidden = Dropout(0.5)(hidden)
# finish with a dense layer
output = Dense(3)(hidden)
```

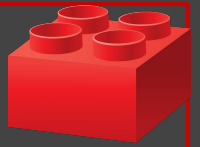
```
# generate embeddings
embedding = w2v.Word2Vec()
embedding.build_vocab(all_questions)
embedding.train(all_questions)
```



```
# 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)
```



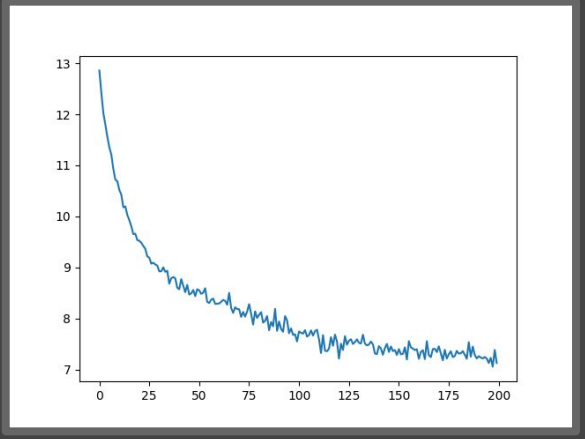
```
# Numbers model
nums_model = Sequential()
nums_model.add(Dense(3, input_dim=3))
nums_input = Input(shape=(3,))
processed_nums = nums_model(nums_input)

# 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)
```

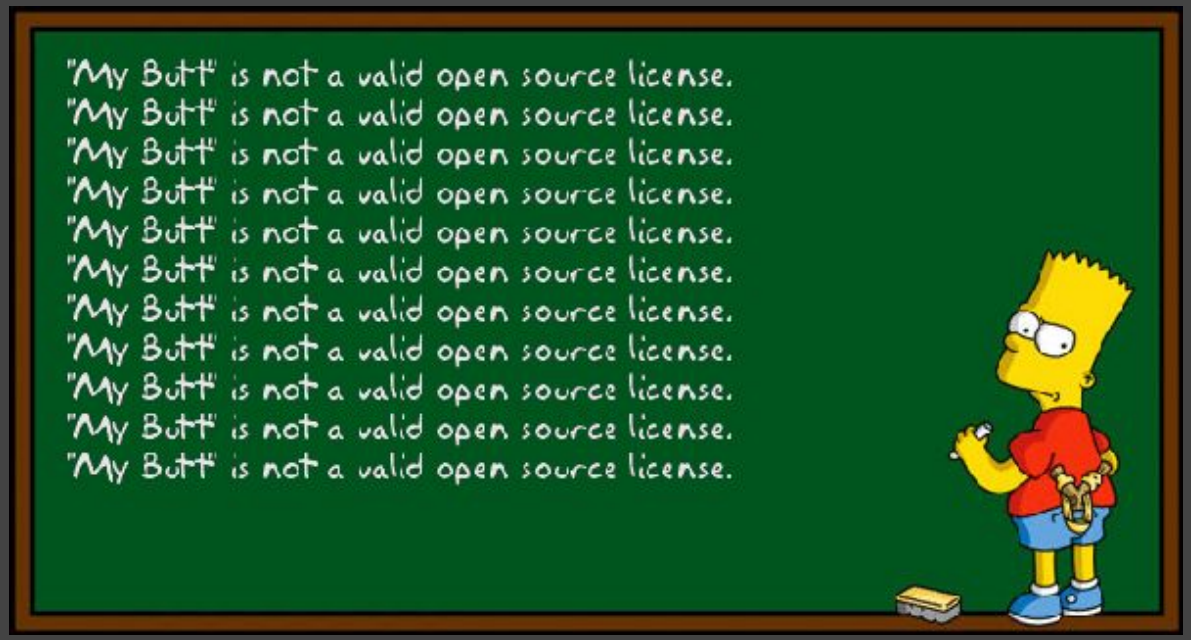
Results

Input:	Prediction:	Truth:
[‘You have 5 apples…’,	5*x=39-19	5*x=39-19
‘23 years from now…’,	2*x=23-16.8	2*x=23-16.8
‘Sean collected 47 stamps…’,	5*x=47-17	5*x=47-17
‘A cup of coffee costs 13.5$ ...’,	2*x=13.5-3.5	2*x=13.5-3.5
‘Fourteen drinks were served …’]	4*x=14-6	-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?!?

Intro/Tutorial

Documentation

Support

Tip #1 - K

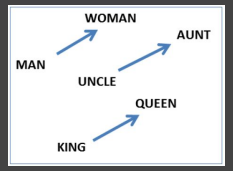
Documentation, Documentation, Documentation?!?

Intro/Tutorial

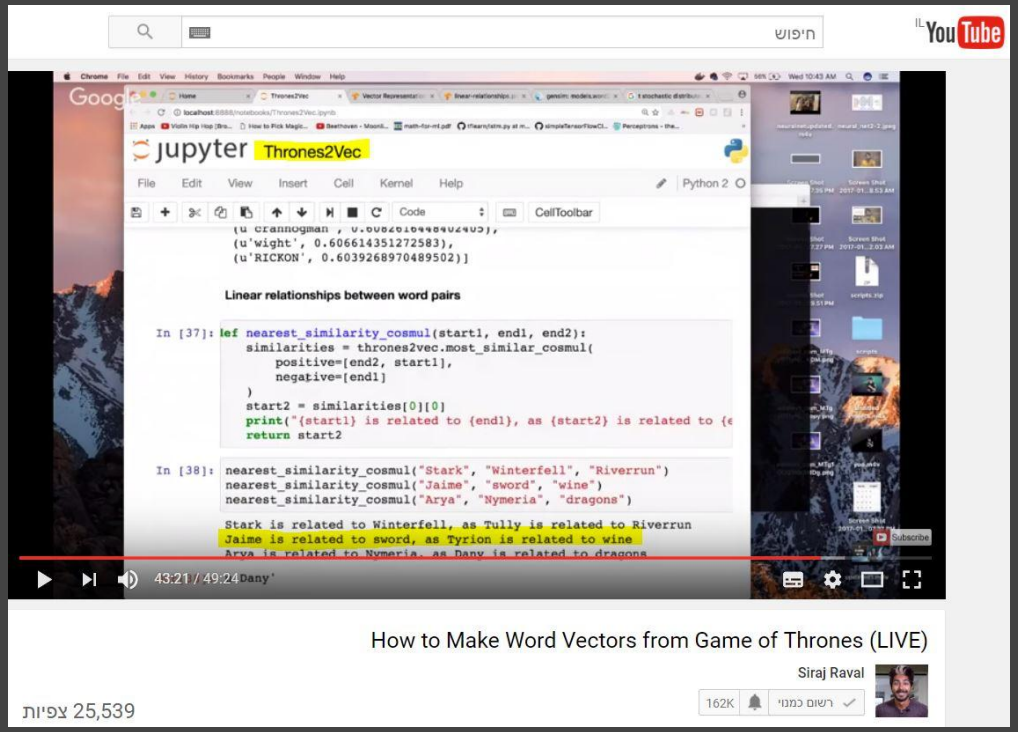
Documentation

Support

Tip #2 -



Think up a killer app



Tip #3 -
Dare to be Mainstream

PART II Recap

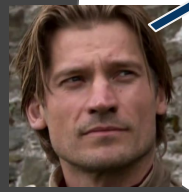
PART II Recap

Friendly

PART II Recap

Friendly

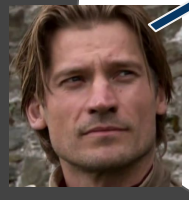
Fun



PART II Recap

Accessible

Fun



Friendly

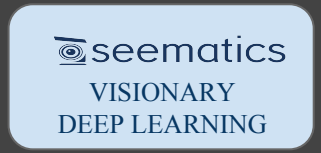
Take Home Message

Thank you :)

Questions?

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

Dalya Gartzman



TeMS -
Textual Math Solver

Simplisico