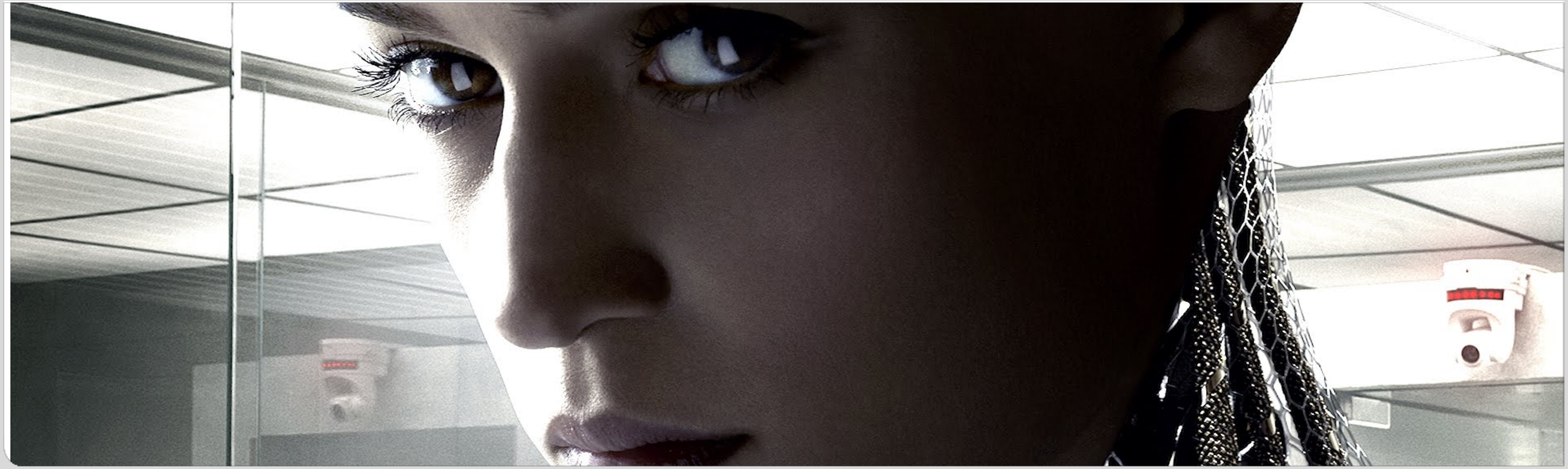
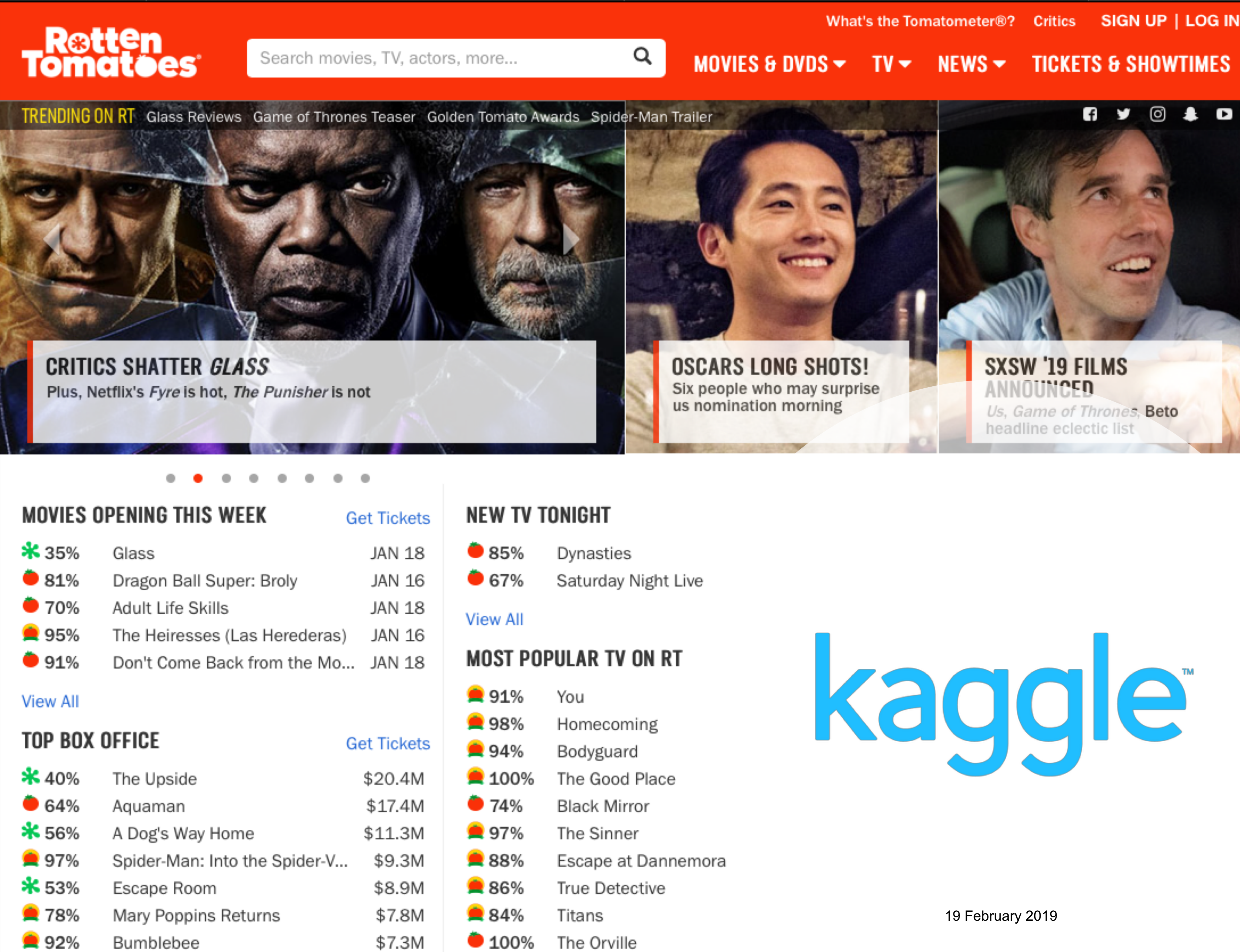
Praktikum: Neural Networks

Movie Sentiment Analysis – Daniela Ruchser, Daniel Betsche, Alan Mazankiewicz

Institut für Anthropomatik und Robotik, Interactive Systems Lab (Lst. Waibel)





kaggle™

19 February 2019

Outline

Data

Preprocessing

Models

Data Exploration

Karlsruhe Institute of Technology

Train Data Set

	PhraseId	SentenceId	Phrase	Sentiment
0	1	1	A series of escapades demonstrating the adage that what is good for the goose is also good for the gander , some of which occasionally amuses but none of which amounts to much of a story .	1
1	2	1	A series of escapades demonstrating the adage that what is good for the goose	2
2	3	1	A series	2
3	4	1	A	2
4	5	1	series	2
5	6	1	of escapades demonstrating the adage that what is good for the goose	2
6	7	1	of	2

	PhraseId	SentenceId	Phrase	Sentiment
76	77	2	is worth seeking .	3
77	78	2	is worth seeking	4
78	79	2	is worth	2

	PhraseId	SentenceId	Phrase	Sentiment
100	101	3	would have a hard time sitting through this one .	1
101	102	3	would have a hard time sitting through this one	0
103	104	3	have a hard time sitting through this one	0

Data Exploration

Karlsruhe Institute of Technology

Train Data Set

	PhraseId	SentenceId	Phrase	Sentiment
0	1	1	A series of escapades demonstrating the adage that what is good for the goose is also good for the gander , some of which occasionally amuses but none of which amounts to much of a story .	1
1	2	1	A series of escapades demonstrating the adage that what is good for the goose	2
2	3	1	A series	2
3	4	1	A	2
4	5	1	series	2
5	6	1	of escapades demonstrating the adage that what is good for the goose	2
6	7	1	of	2

	PhraseId	SentenceId	Phrase	Sentiment
76	77	2	is worth seeking .	3
77	78	2	is worth seeking	4
78	79	2	is worth	2

	PhraseId	SentenceId	Phrase	Sentiment
100	101	3	would have a hard time sitting through this one .	1
101	102	3	would have a hard time sitting through this one	0
103	104	3	have a hard time sitting through this one	0

Data Exploration

Karlsruhe Institute of Technology

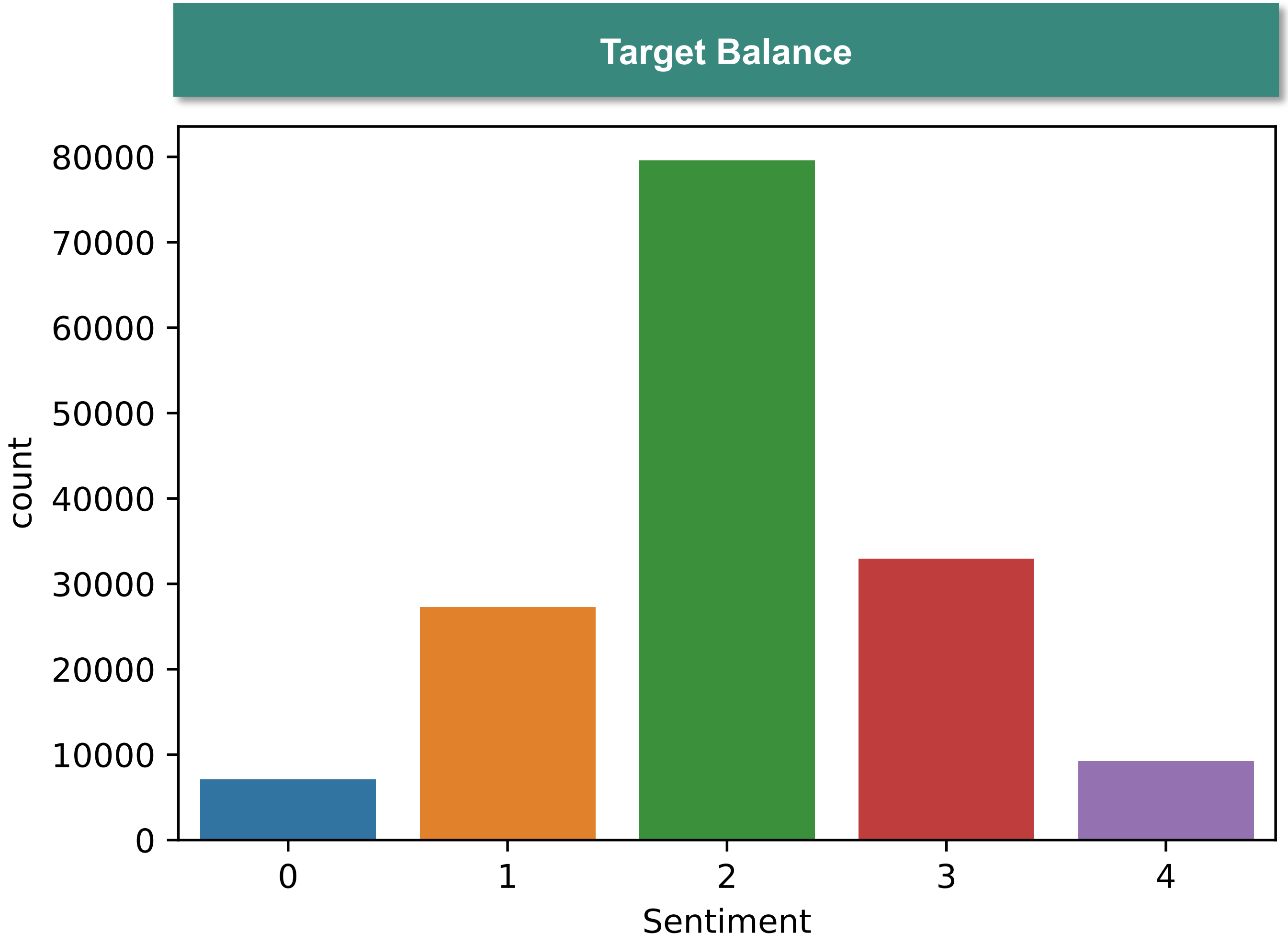
Train Data Set

	PhraseId	SentenceId	Phrase	Sentiment
0	1	1	A series of escapades demonstrating the adage that what is good for the goose is also good for the gander , some of which occasionally amuses but none of which amounts to much of a story .	1
1	2	1	A series of escapades demonstrating the adage that what is good for the goose	2
2	3	1	A series	2
3	4	1	A	2
4	5	1	series	2
5	6	1	of escapades demonstrating the adage that what is good for the goose	2
6	7	1	of	2

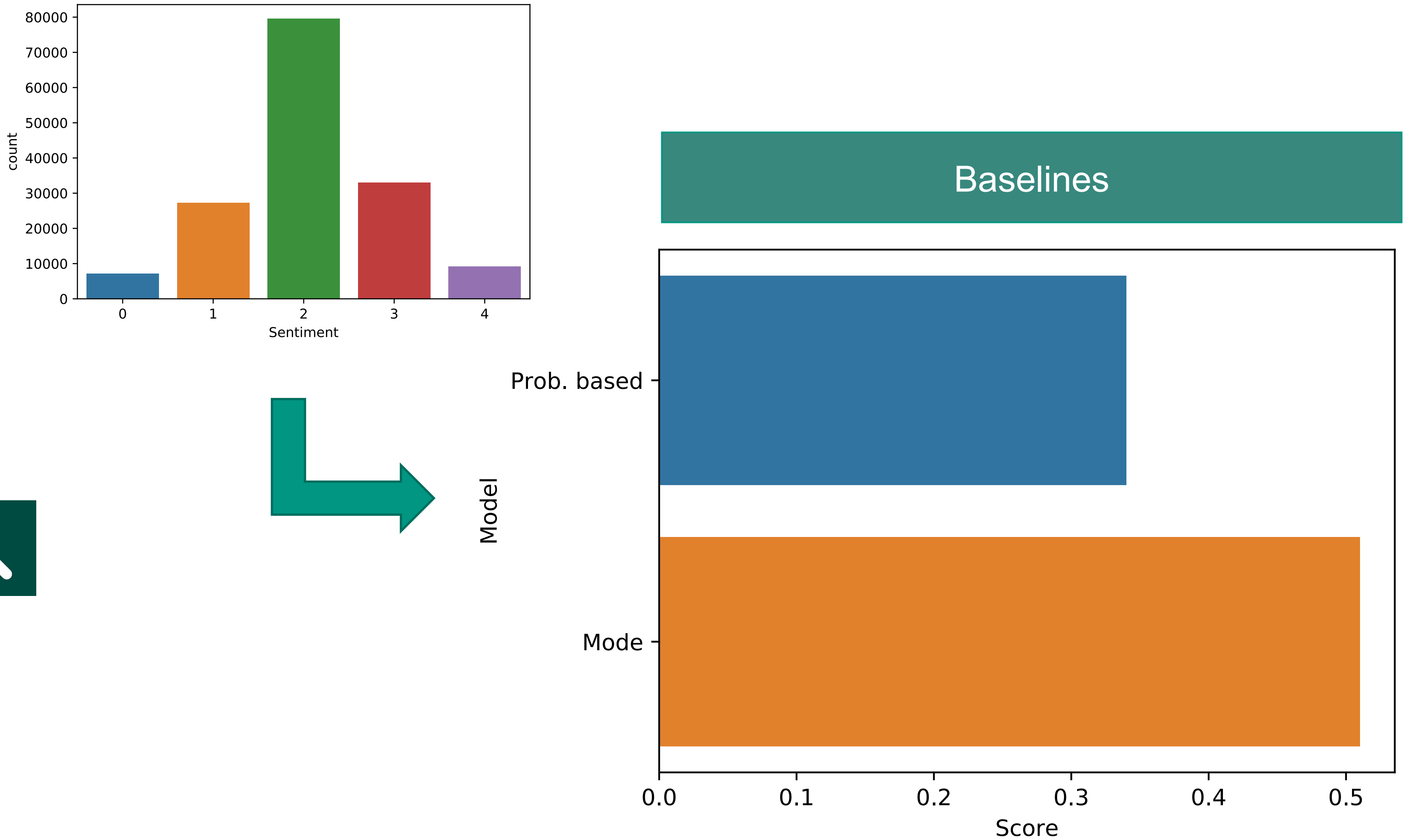
	PhraseId	SentenceId	Phrase	Sentiment
76	77	2	is worth seeking .	3
77	78	2	is worth seeking	4
78	79	2	is worth	2

	PhraseId	SentenceId	Phrase	Sentiment
100	101	3	would have a hard time sitting through this one .	1
101	102	3	would have a hard time sitting through this one	0
103	104	3	have a hard time sitting through this one	0

Data Exploration

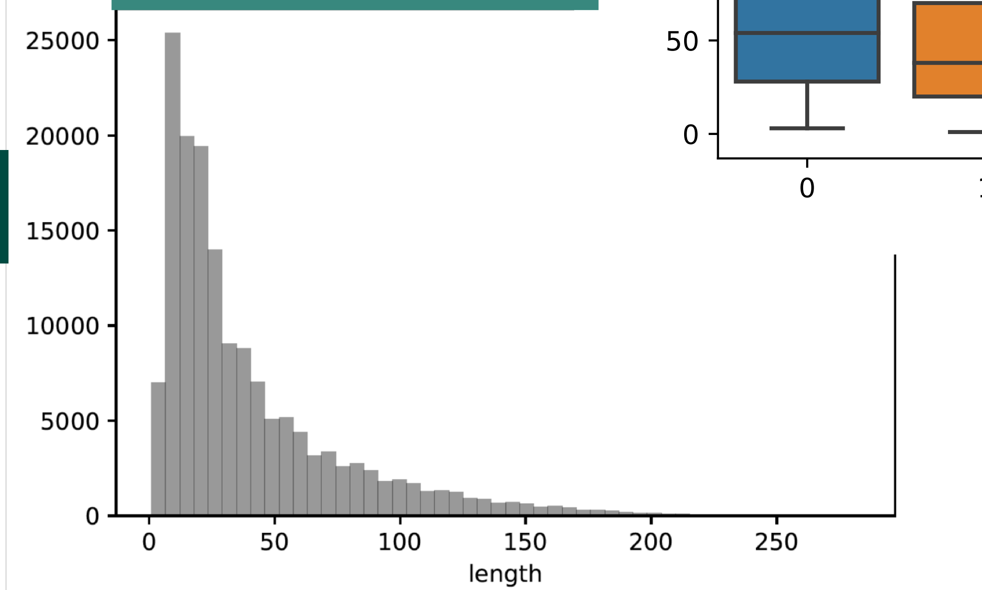


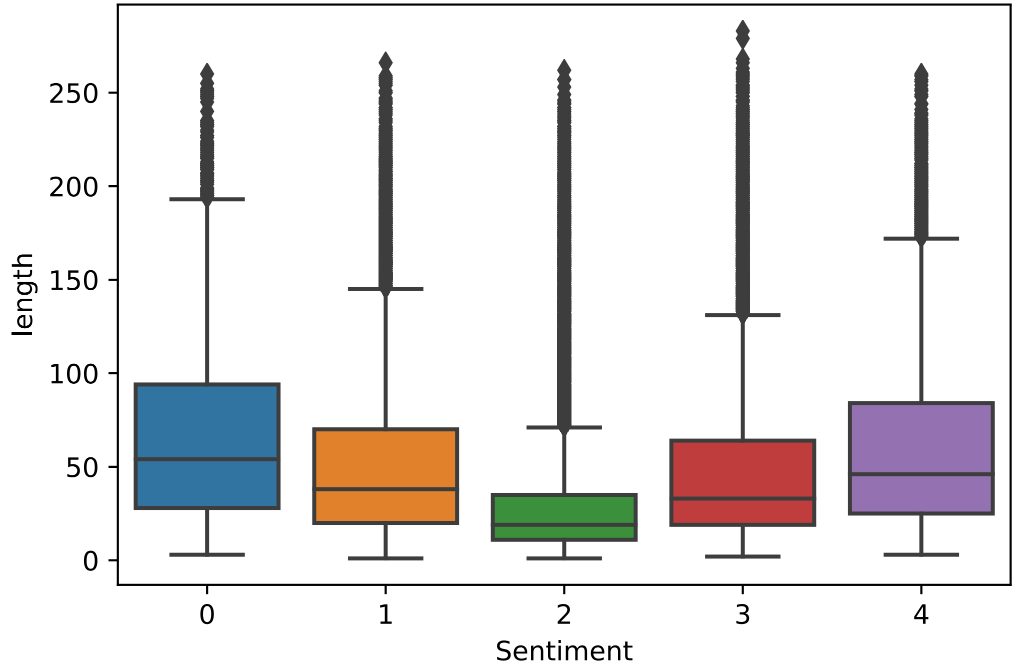
Data Exploration



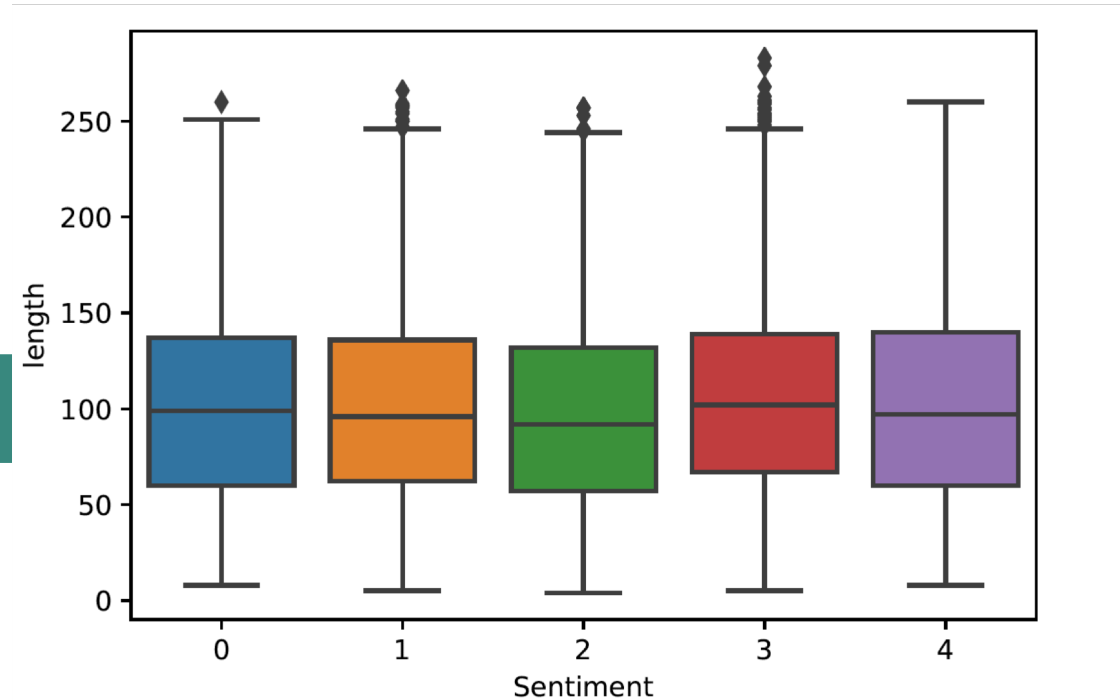
Data Exploration

Text Length

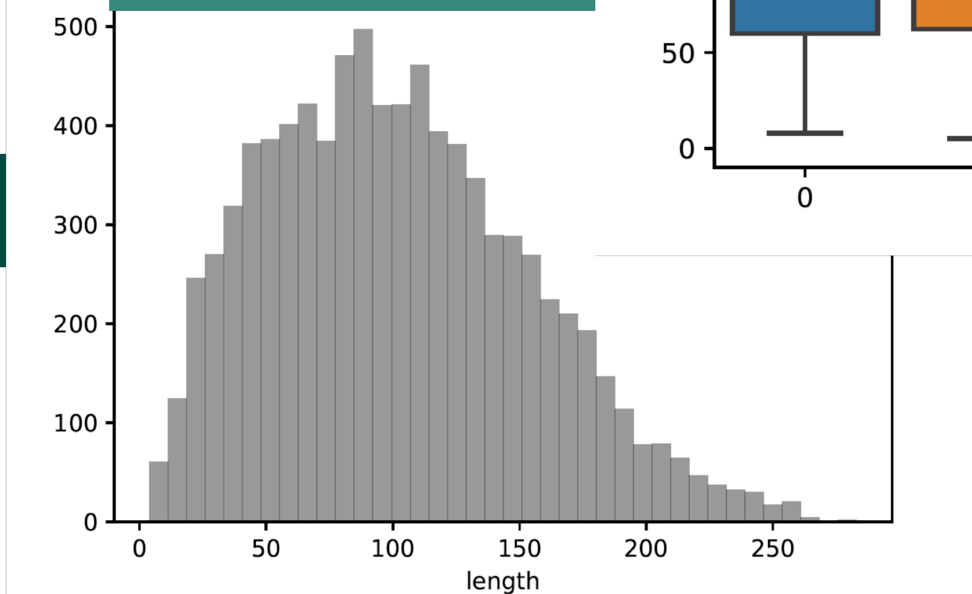




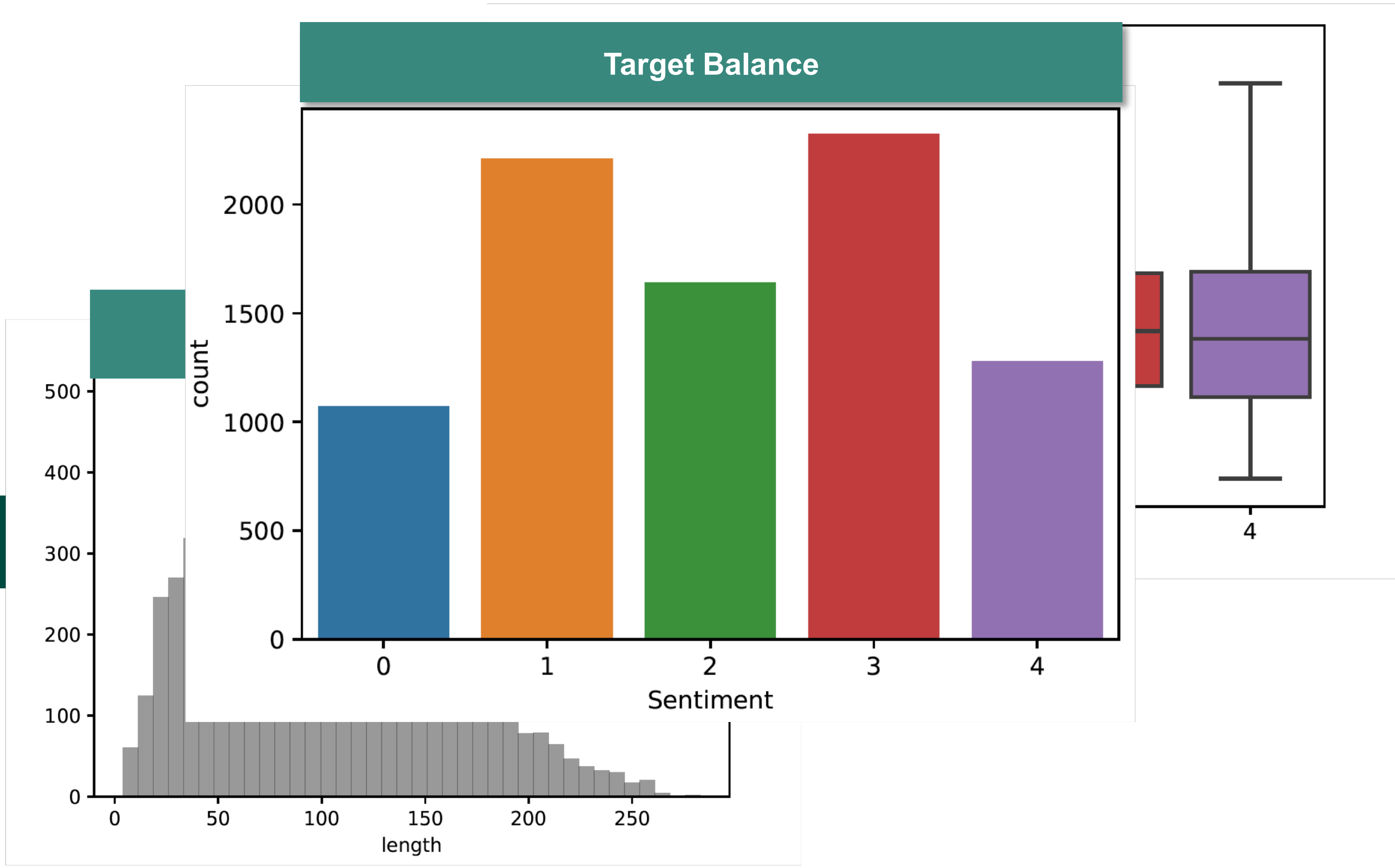
Data Exploration – Sentence Level



Text Length



Data Exploration – Sentence Level



Leaderboard

Top 15

1	—	Mark Archer	0.76526	22	4y
2	—	Armineh Nourbakhsh	0.76096	7	4y
3	—	Merlion	0.70936	9	5y
4	—	Puneet Singh	0.70789	14	4y
5	—	Yoon	0.68765	4	4y
6	▲ 18	DrStrangelove	0.67931	42	4y
7	▲ 3	akqwerty	0.67931	55	4y
8	▲ 1	MDAKMlab	0.67854	200	4y
9	▼ 3	st_sopov	0.67590	55	4y
10	▼ 3	JR	0.67496	45	4y
11	▼ 3	Dmitry Ulyanov	0.67205	11	4y
12	▲ 186	Phil Culliton	0.67133	42	4y
13	—	UCL_yocodima*	0.67066	34	4y
14	▲ 1	UCL Gaussian Priors	0.67041	37	4y
15	▼ 4	UCLOL_It's_been_a_hard_Baye...	0.66973	41	4y

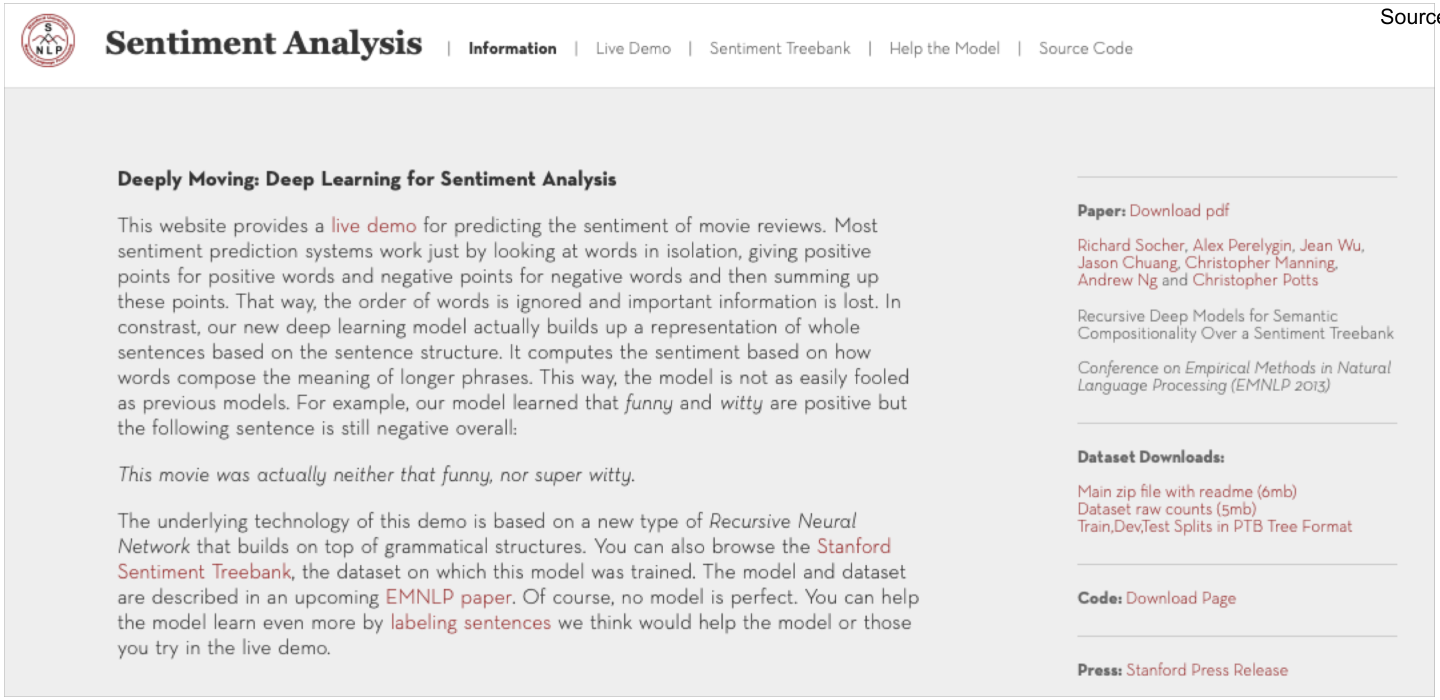
Stanford CoreNLP

Socher et al. - Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, 2013

Wrapper to access Stanford Core NLP Server available in Python

Performance: 80.7 %

Source: hackernoon.com



Sentiment Analysis | **Information** | Live Demo | Sentiment Treebank | Help the Model | Source Code

Deeply Moving: Deep Learning for Sentiment Analysis

This website provides a live demo for predicting the sentiment of movie reviews. Most sentiment prediction systems work just by looking at words in isolation, giving positive points for positive words and negative points for negative words and then summing up these points. That way, the order of words is ignored and important information is lost. In constrast, our new deep learning model actually builds up a representation of whole sentences based on the sentence structure. It computes the sentiment based on how words compose the meaning of longer phrases. This way, the model is not as easily fooled as previous models. For example, our model learned that *funny* and *witty* are positive but the following sentence is still negative overall:

This movie was actually neither that funny, nor super witty.

The underlying technology of this demo is based on a new type of *Recursive Neural Network* that builds on top of grammatical structures. You can also browse the Stanford Sentiment Treebank, the dataset on which this model was trained. The model and dataset are described in an upcoming EMNLP paper. Of course, no model is perfect. You can help the model learn even more by labeling sentences we think would help the model or those you try in the live demo.

Paper: Download pdf

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher Manning, Andrew Ng and Christopher Potts

Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank

Conference on *Empirical Methods in Natural Language Processing (EMNLP 2013)*

Dataset Downloads:

Main zip file with readme (6mb)
Dataset raw counts (5mb)
Train,Dev,Test Splits in PTB Tree Format

Code: Download Page

Press: Stanford Press Release

Outline

Data

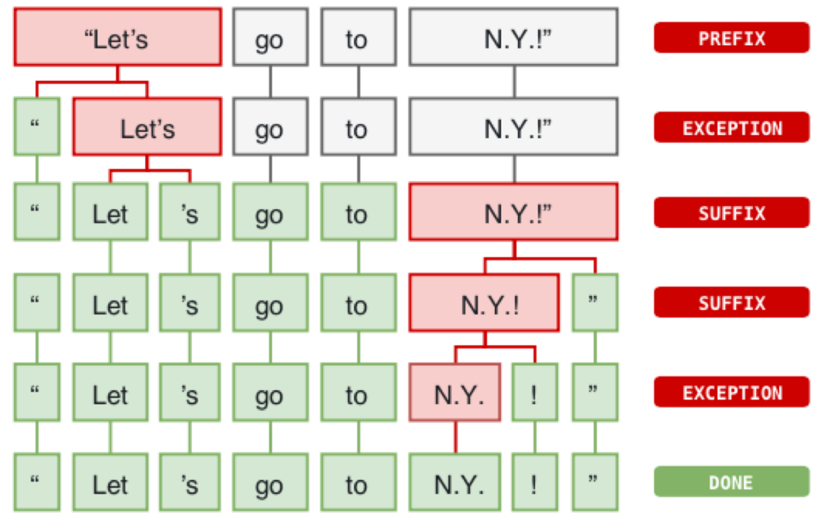
Preprocessing

Models

Data Preprocessing

spaCy

- Tokenization
  - Segmenting text into words, punctuations marks etc.



Source: spacy.io

PYTORCH

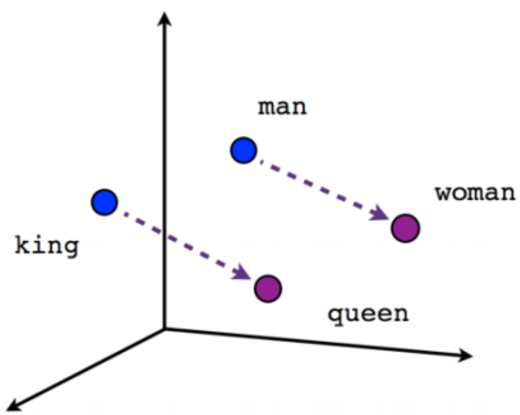
- Utilities e.g.
  - Loading, numericalizing, batching, padding, data set splits / shuffle

Batch

This	movie	is	great	**0**	**0**
Johny	Depps	performance	was	very	poor
...	...	...	...	...	**0**

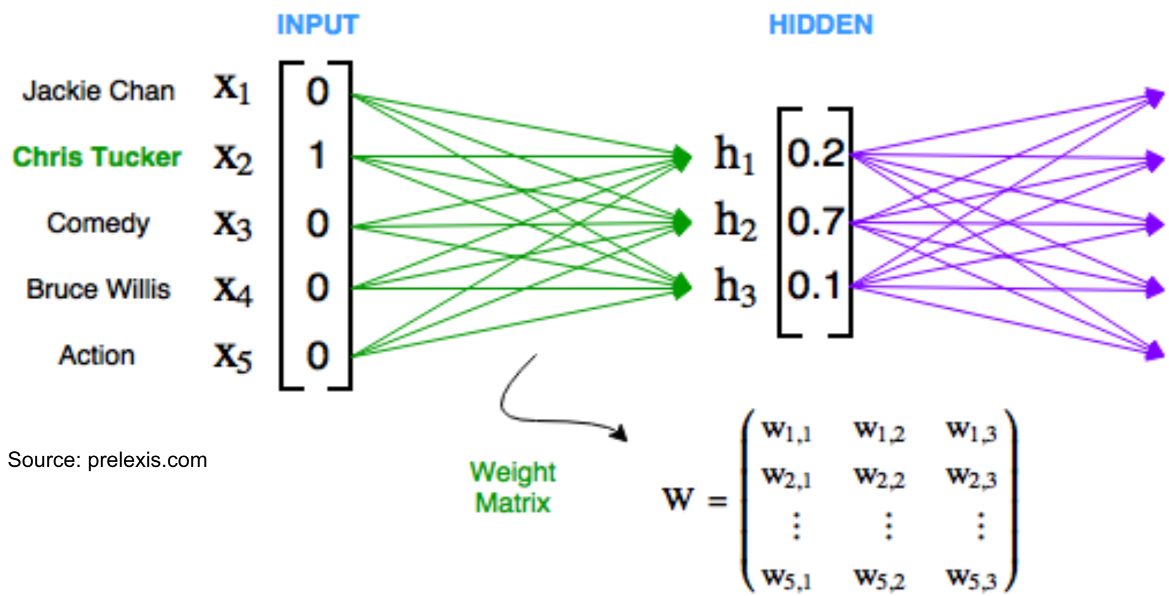
Word Embeddings

Semantic Similarity

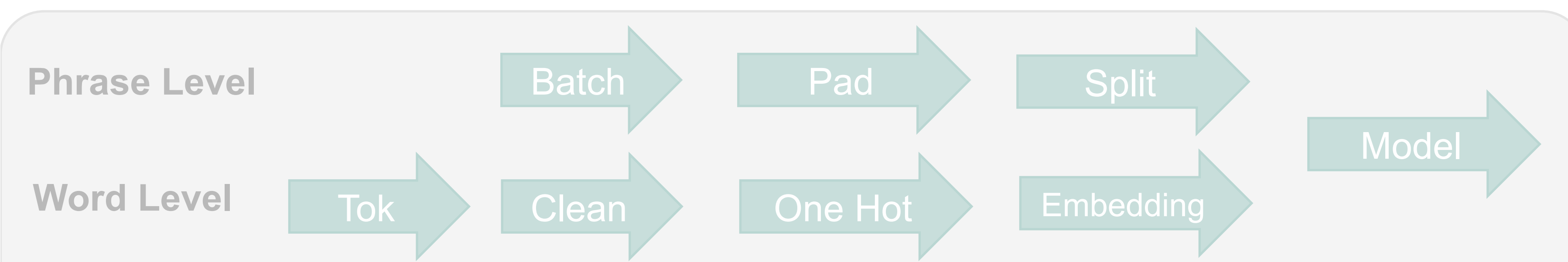


Source: towardsdatascience.com

Pretrained Glove 6B300D



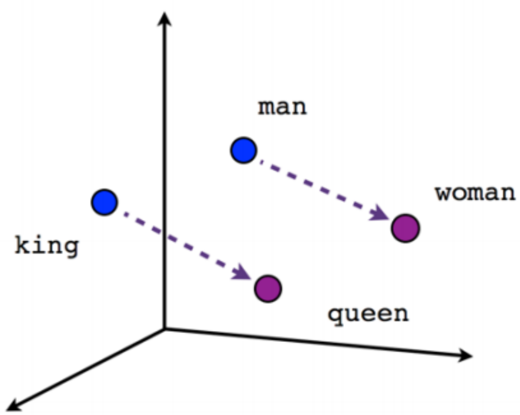
Source: prelexis.com



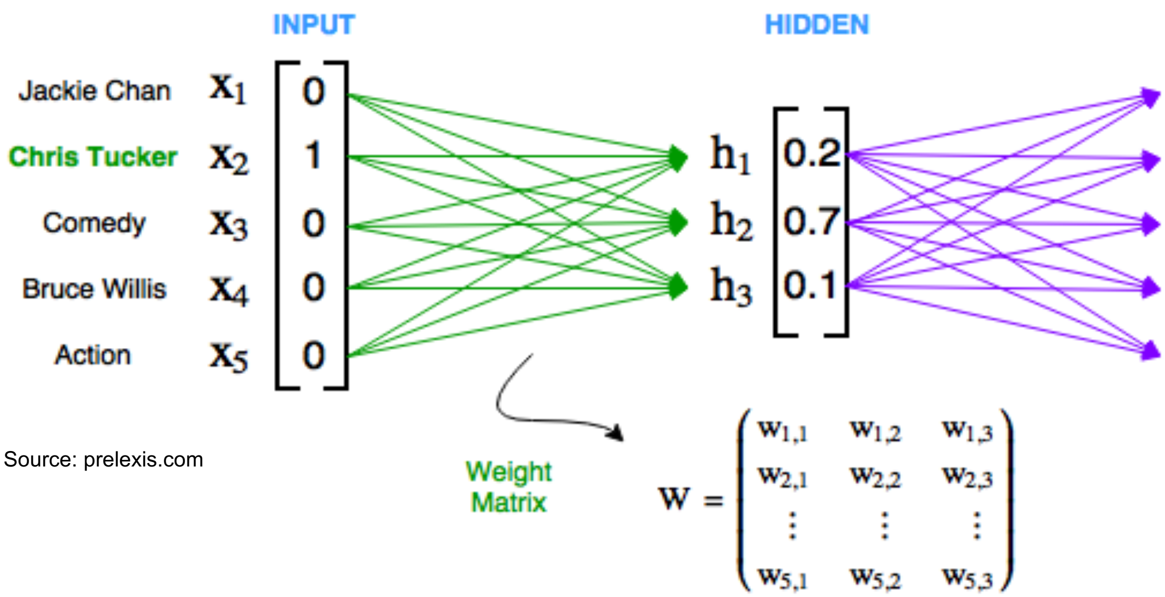
Word Embeddings

Semantic Similarity

Pretrained Glove 6B300D



Source: towardsdatascience.com



$$x = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad h = \begin{bmatrix} 0.2 \\ 0.7 \\ 0.1 \end{bmatrix}$$

$$W = \begin{pmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \\ \vdots & \vdots & \vdots \\ w_{5,1} & w_{5,2} & w_{5,3} \end{pmatrix}$$

Source: prelexis.com

Phrase Level: Batch → Pad → Split → Model

Word Level: Tok → Clean → One Hot → Embedding → Model

Outline

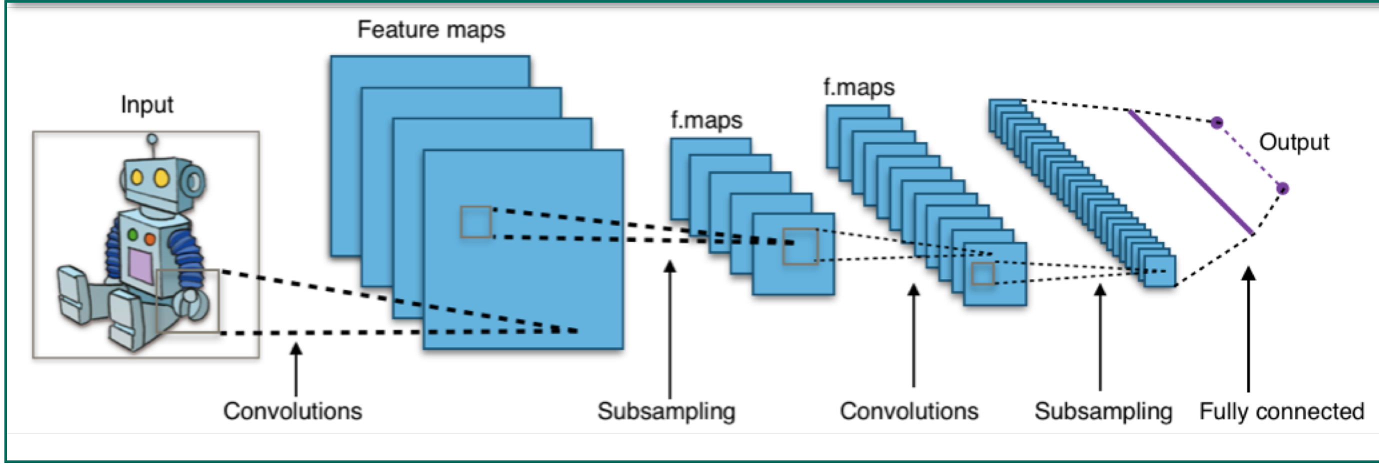
Data

Preprocessing

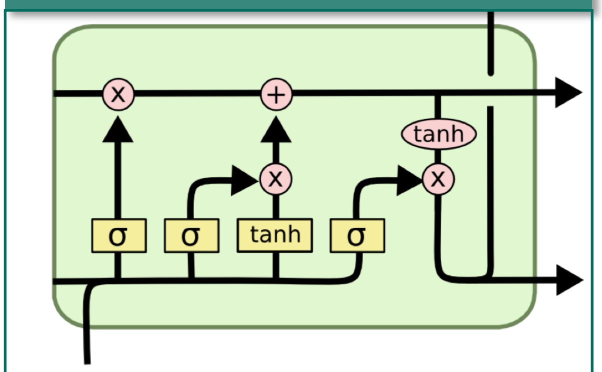
Models

Model Overview

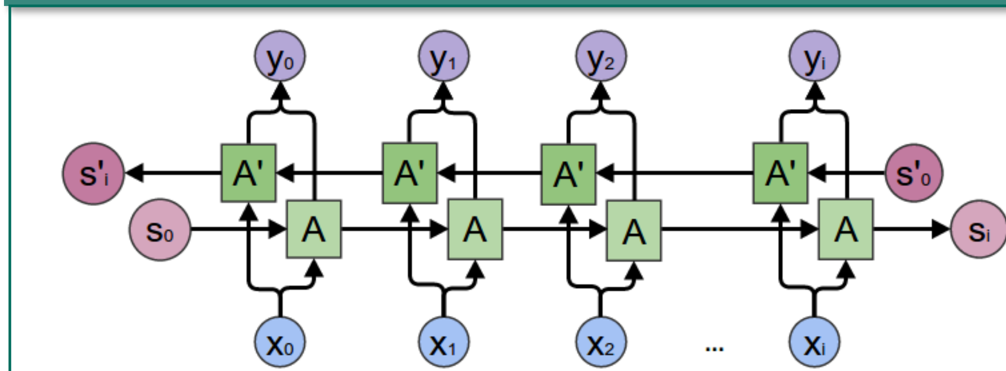
CNN



LSTM

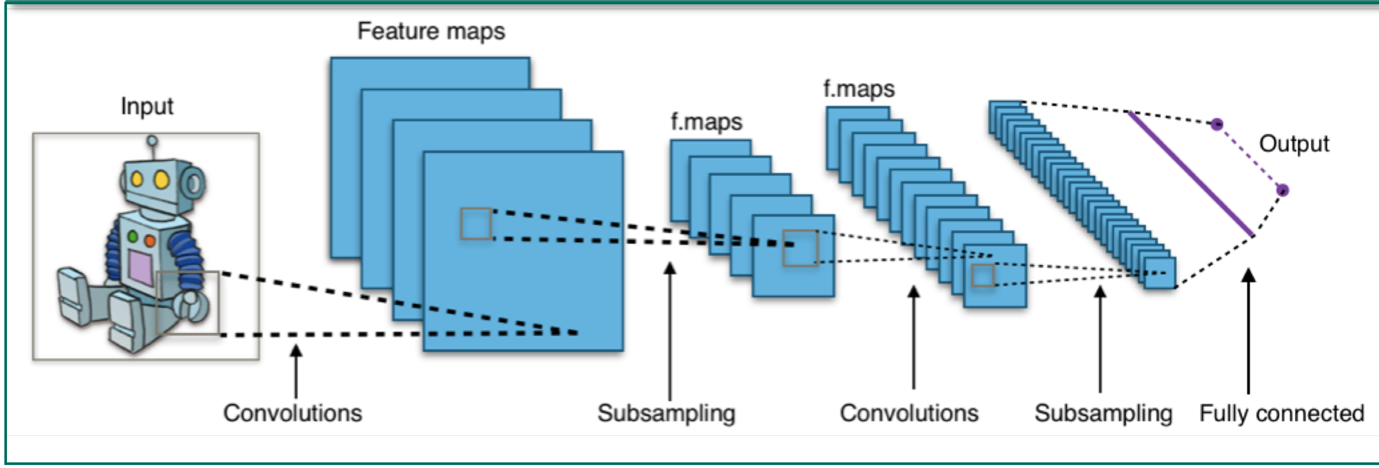


BLSTM

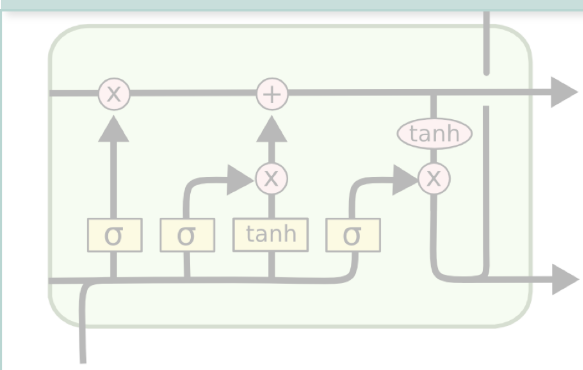


Model Overview

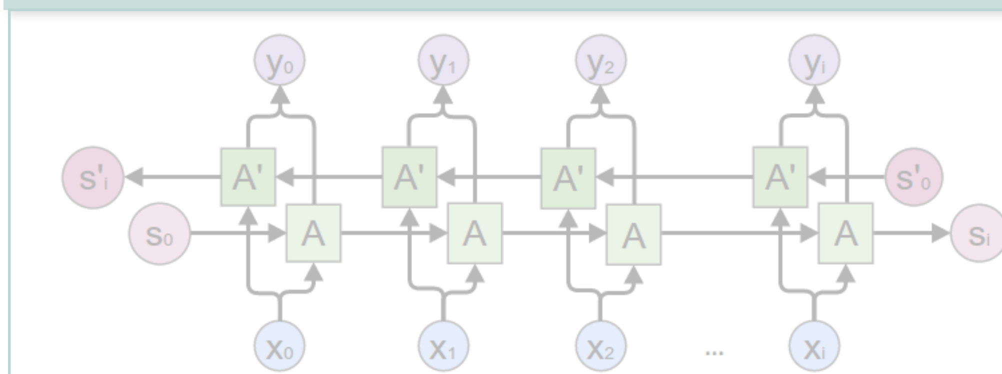
CNN



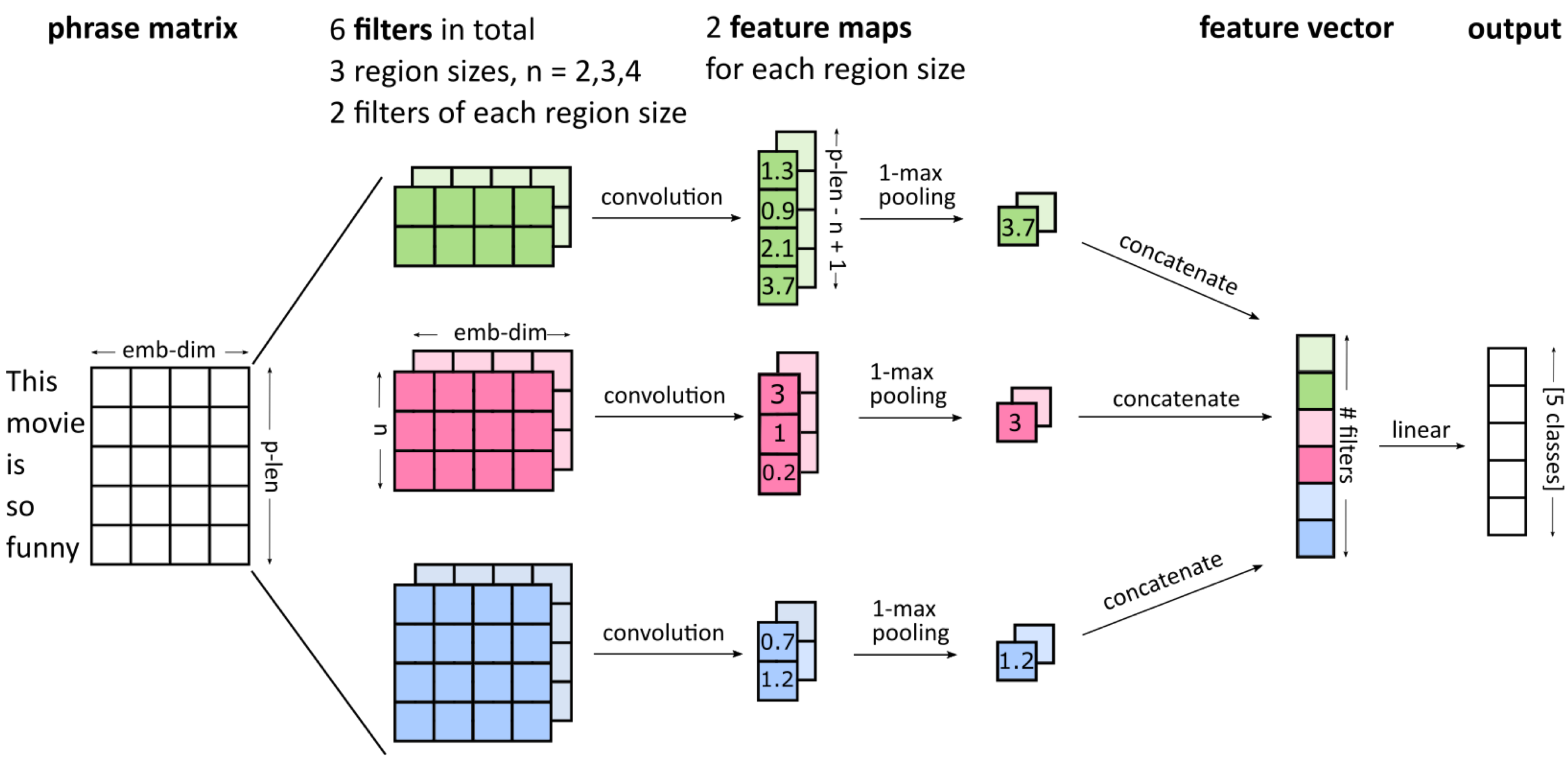
LSTM



BLSTM

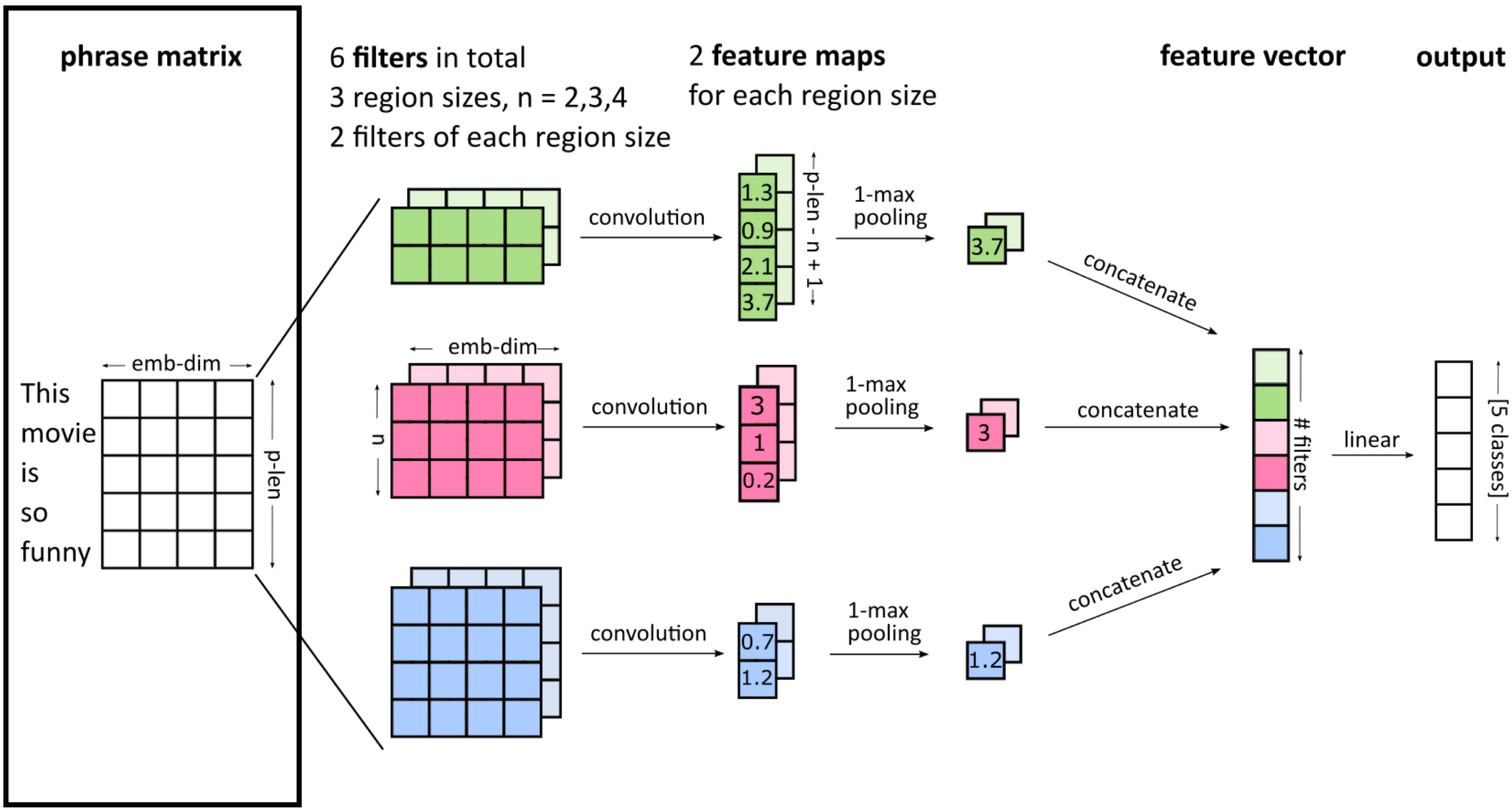


CNN – Convolutional Neural Network

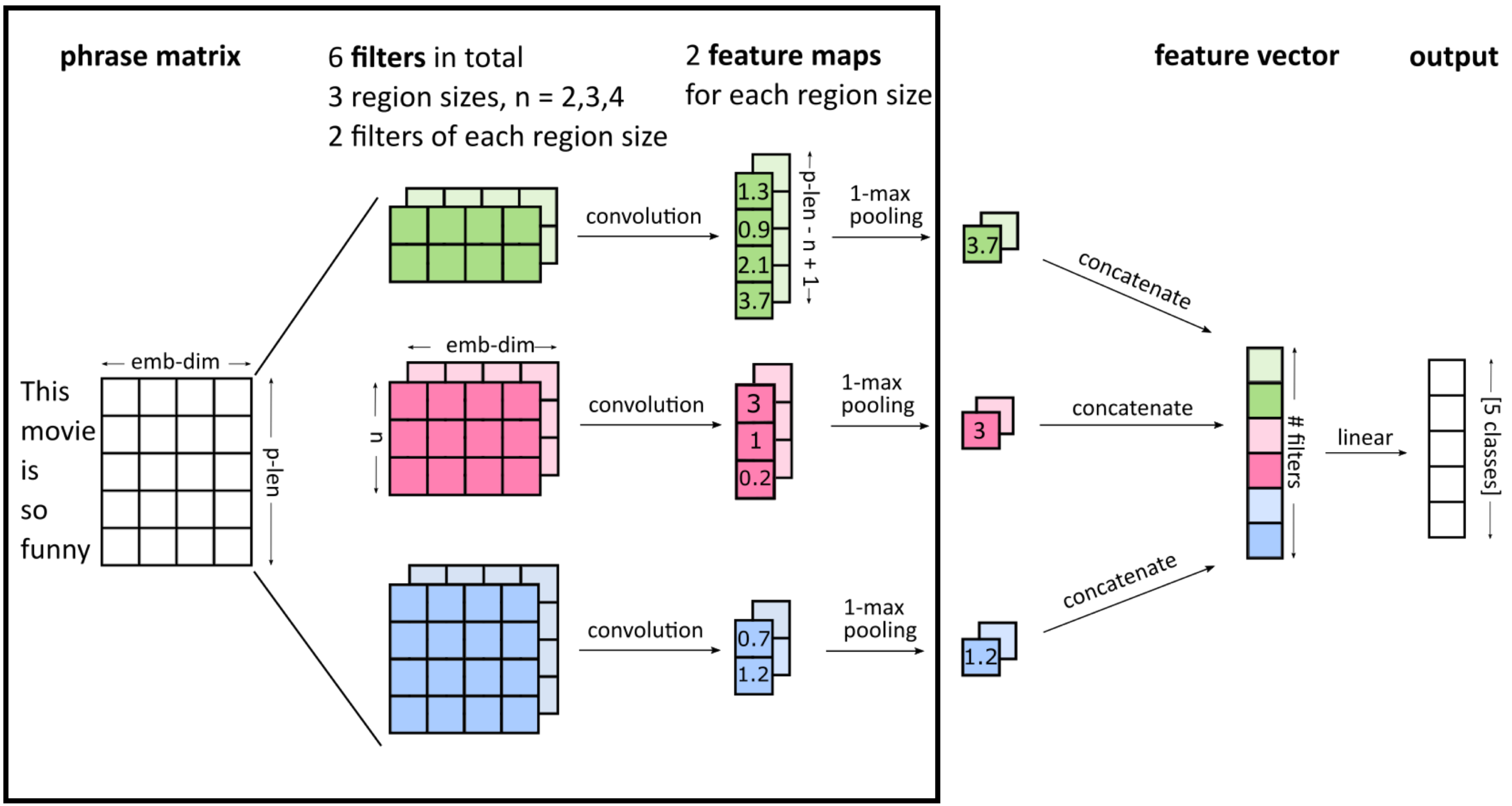


- Architecture is based on ‘Convolutional neural networks for sentence classification‘ by Yoon Kim (2014)

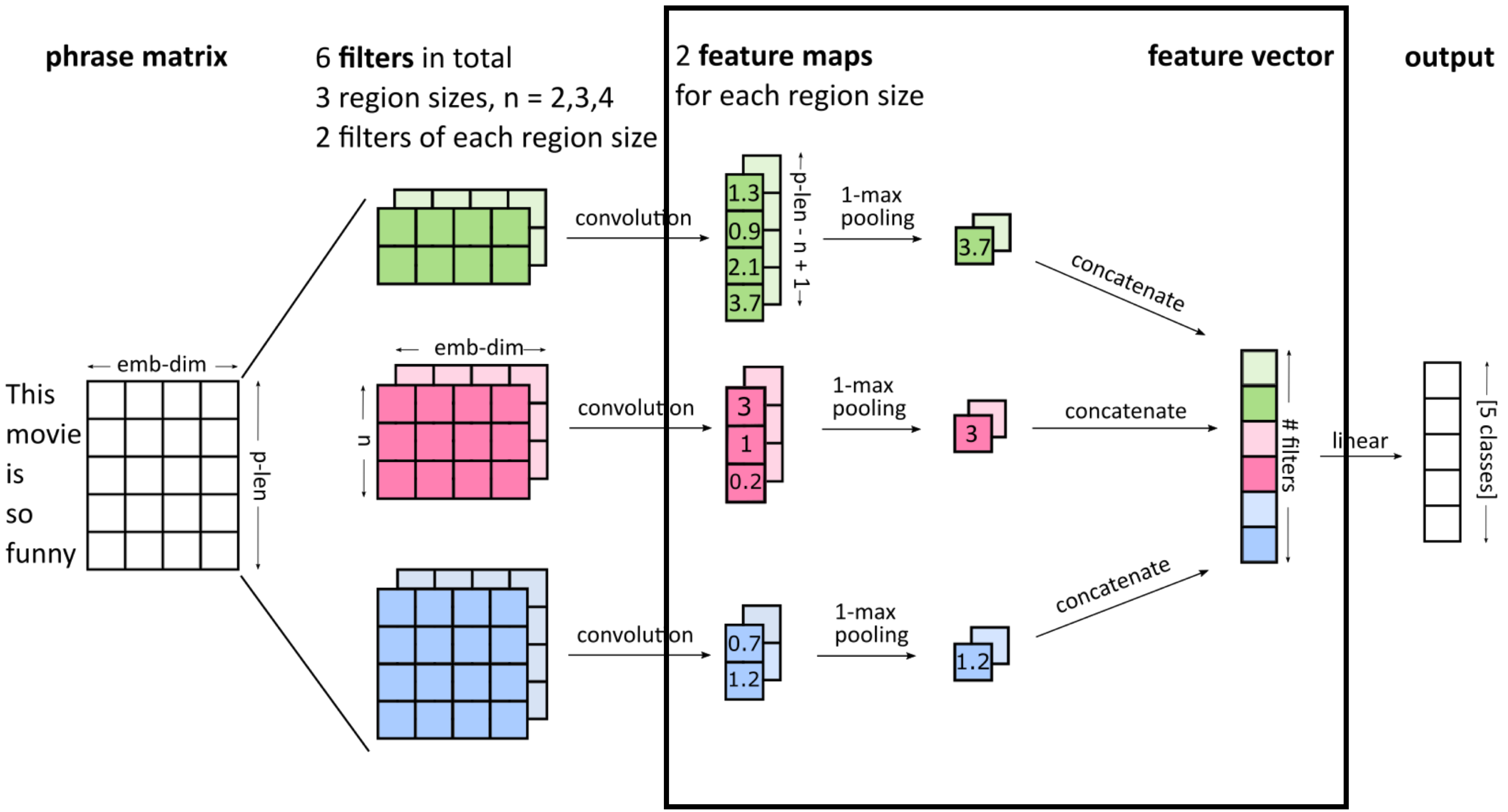
CNN – Embedding Layer



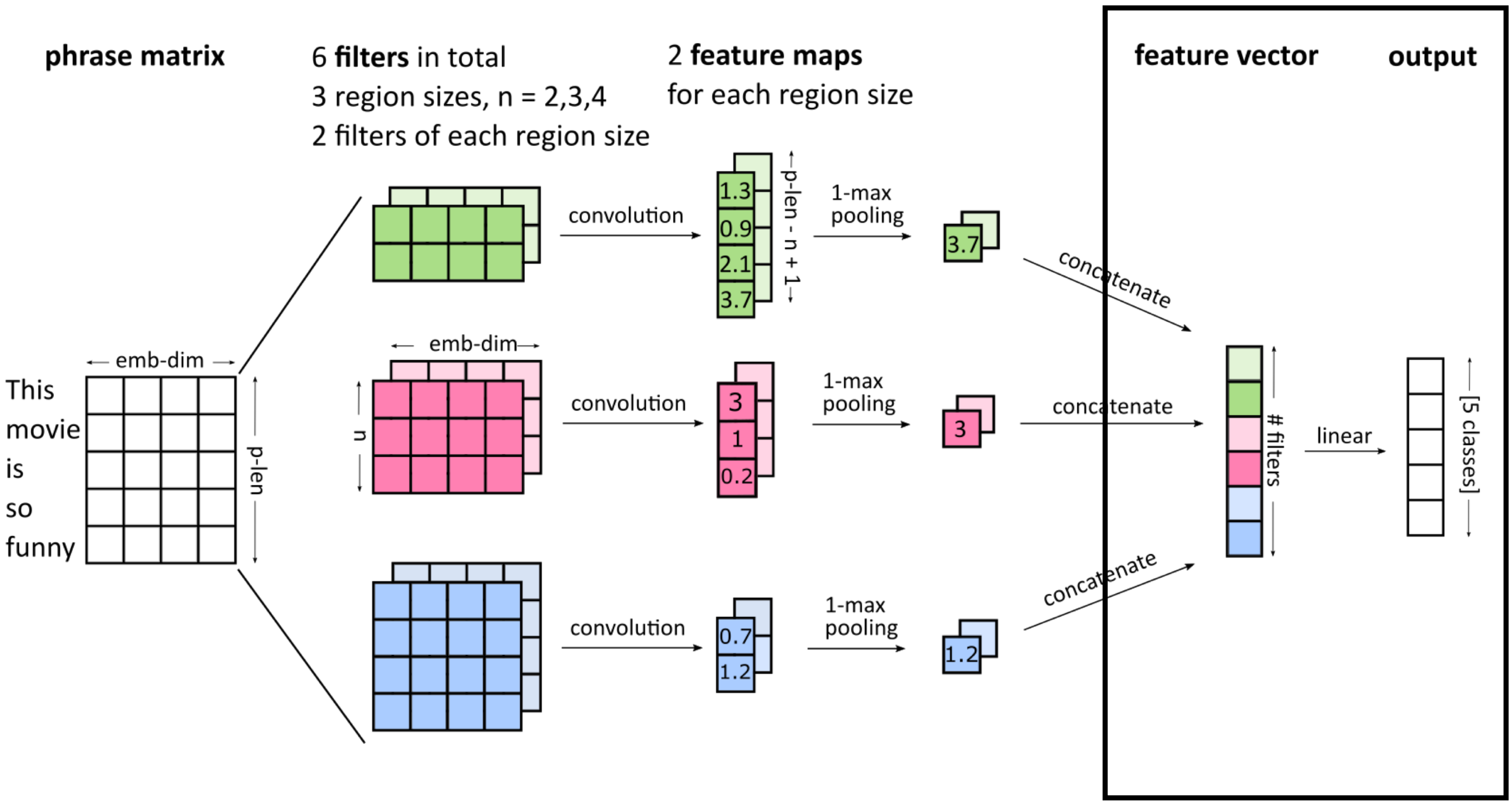
CNN – Convolutional Layer



CNN – Max Pooling Layer



CNN – Output Layer



CNN – Hyperparameters

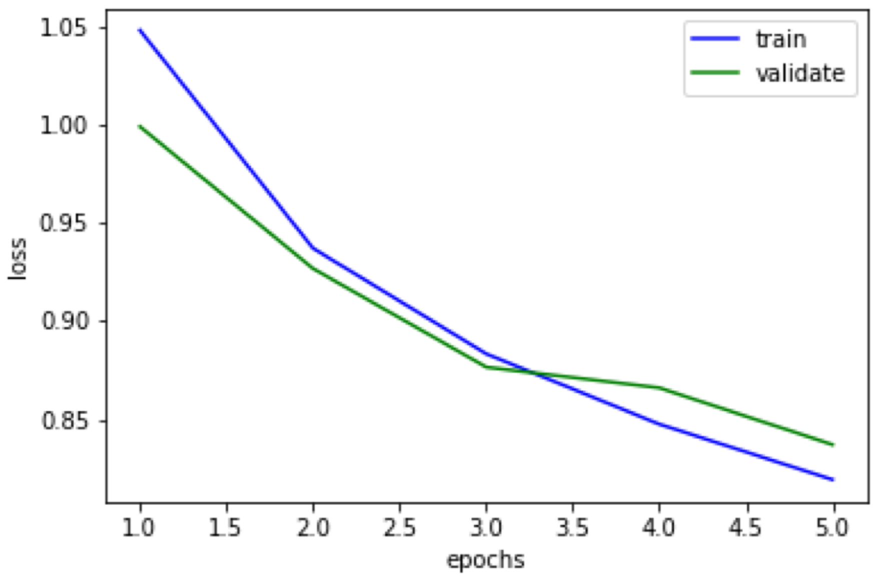
- 5 Epochs
- 32 Batch Size
- 0.6 Dropout
- Region Sizes: 2-10
- 120 Filters per region size
- glove.6B.100d Word Embedding

CNN – Results

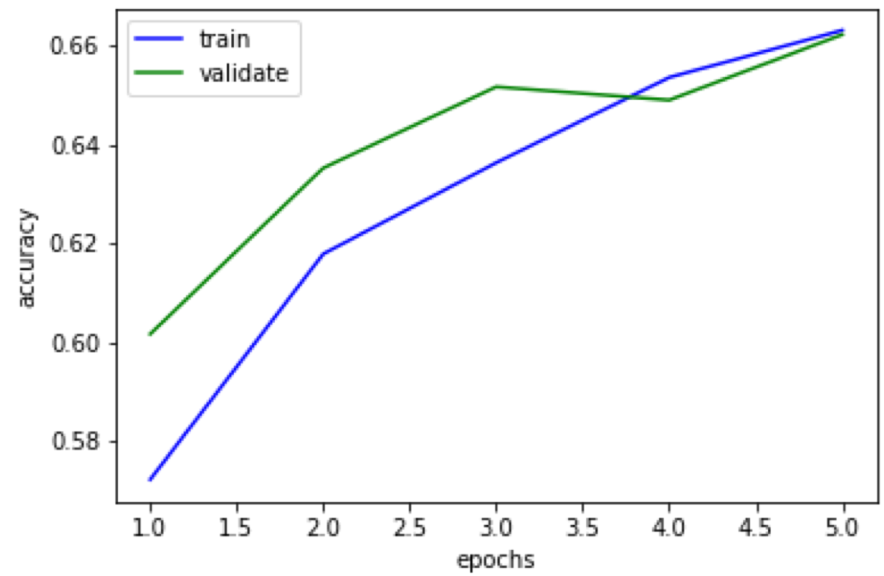
Test accuracy: 66.11%

Kaggle test accuracy: 64.28 %

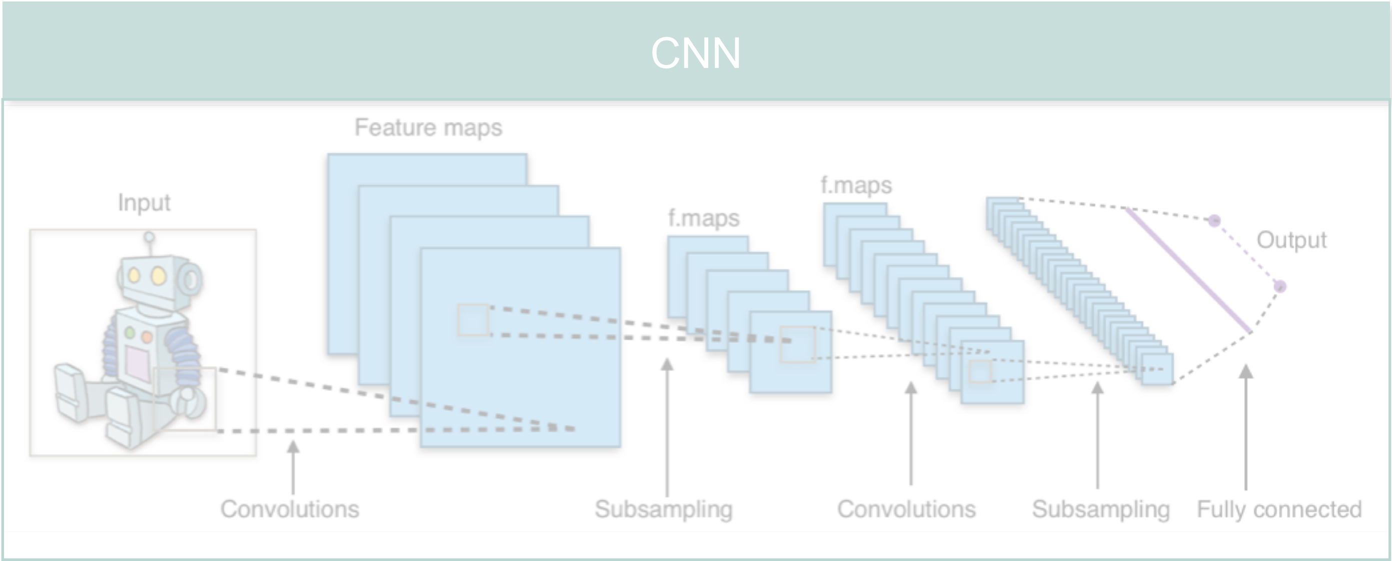
Loss

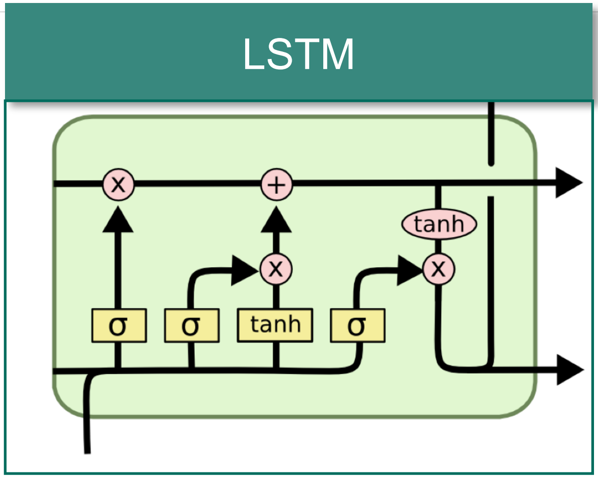


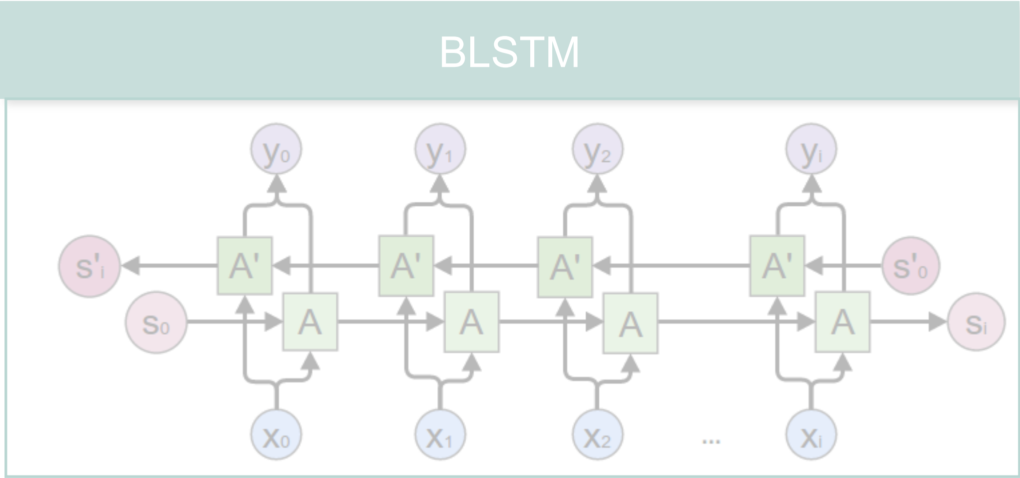
Accuracy



Model Overview

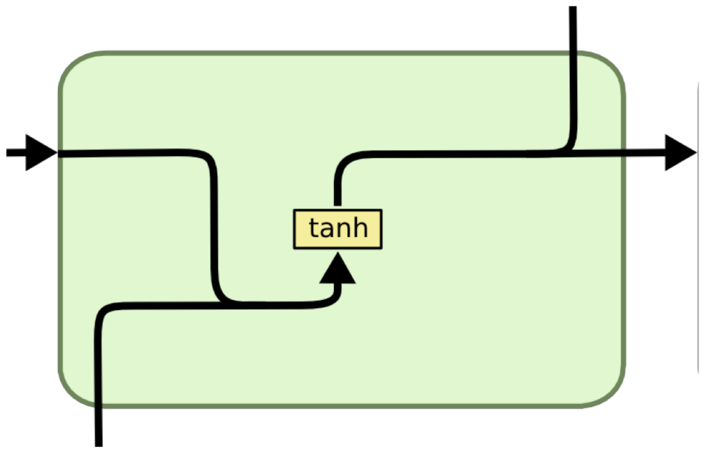




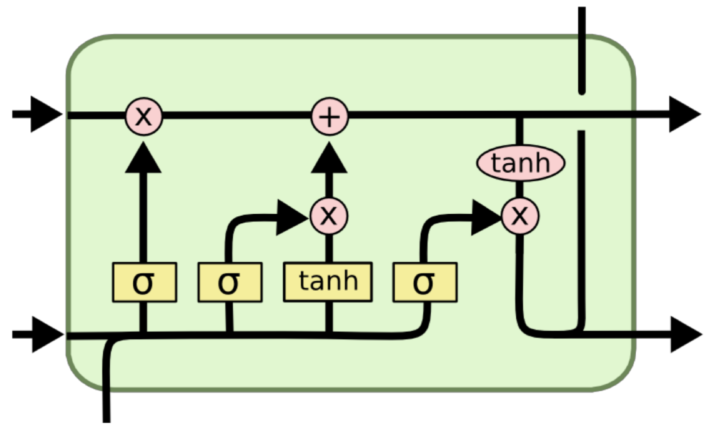


LSTM – Long short-term memory

RNN



LSTM



Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM – Hyperparameters

6 Epochs

1 Layers

32 Batch Size

0.5 Dropout

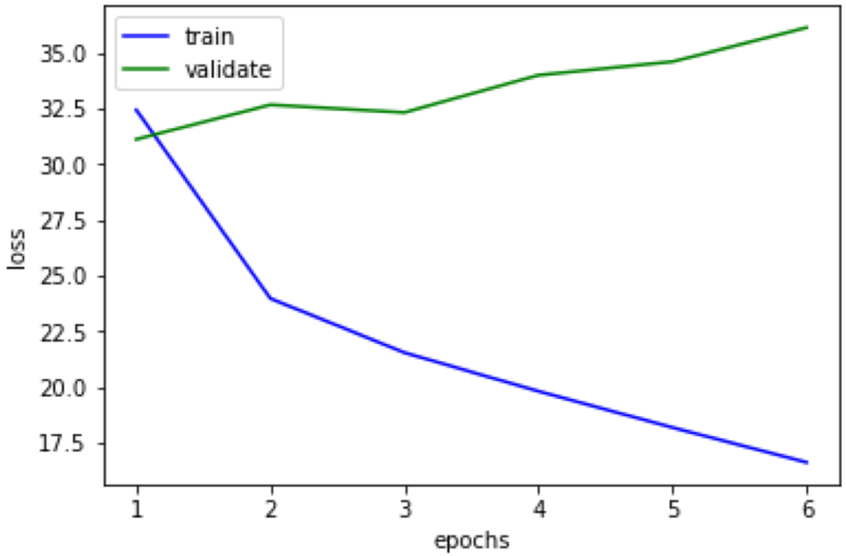
256 Hidden Dimensions

glove.6B.300d Word Embedding

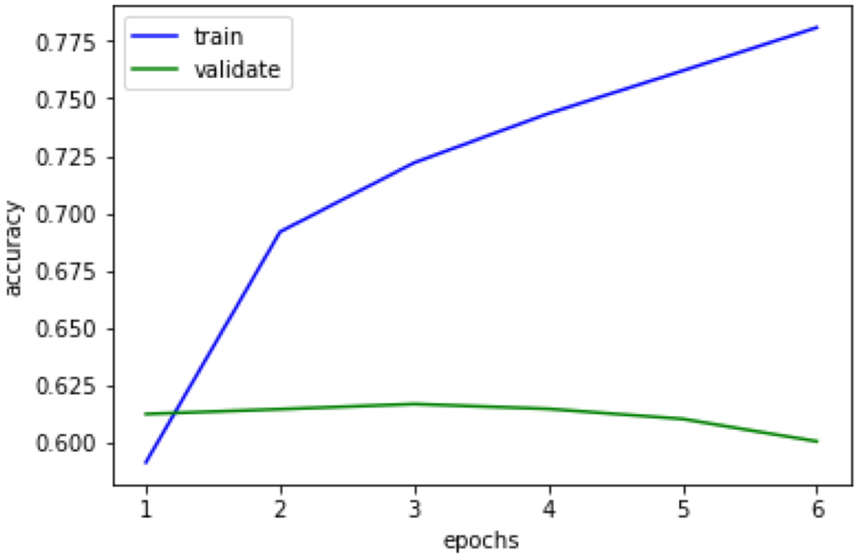
LSTM – Results

- Test accuracy: 59,18%
- Kaggle test accuracy: 57,73%

Loss

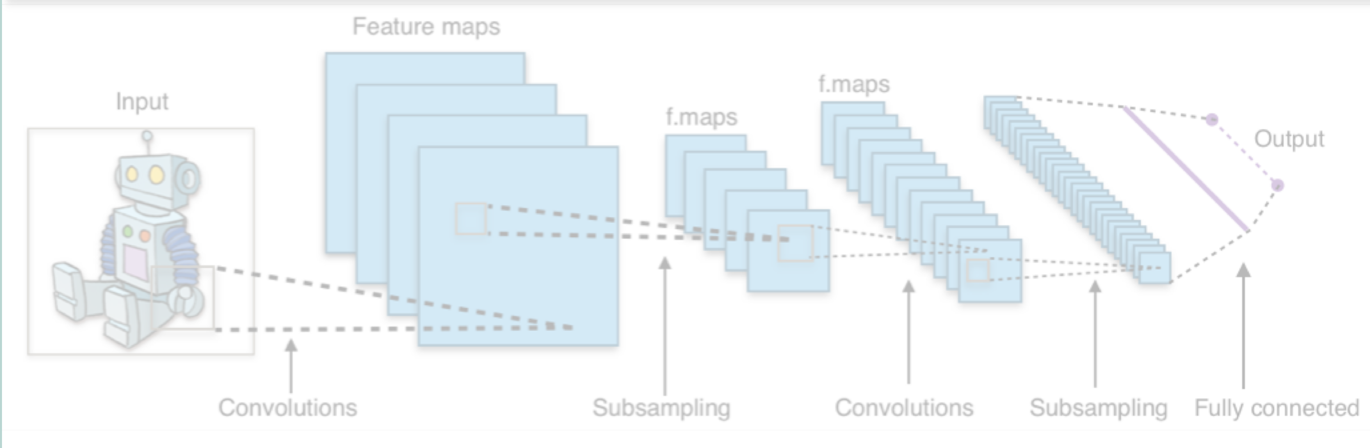


Accuracy

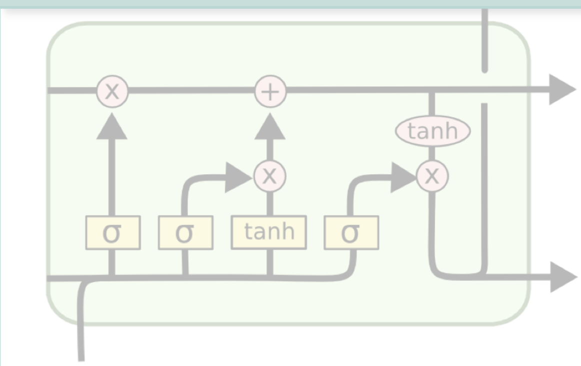


Model Overview

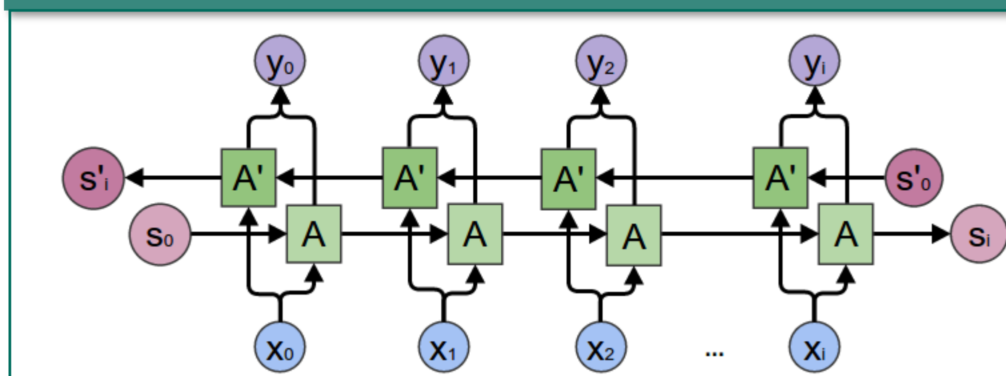
CNN



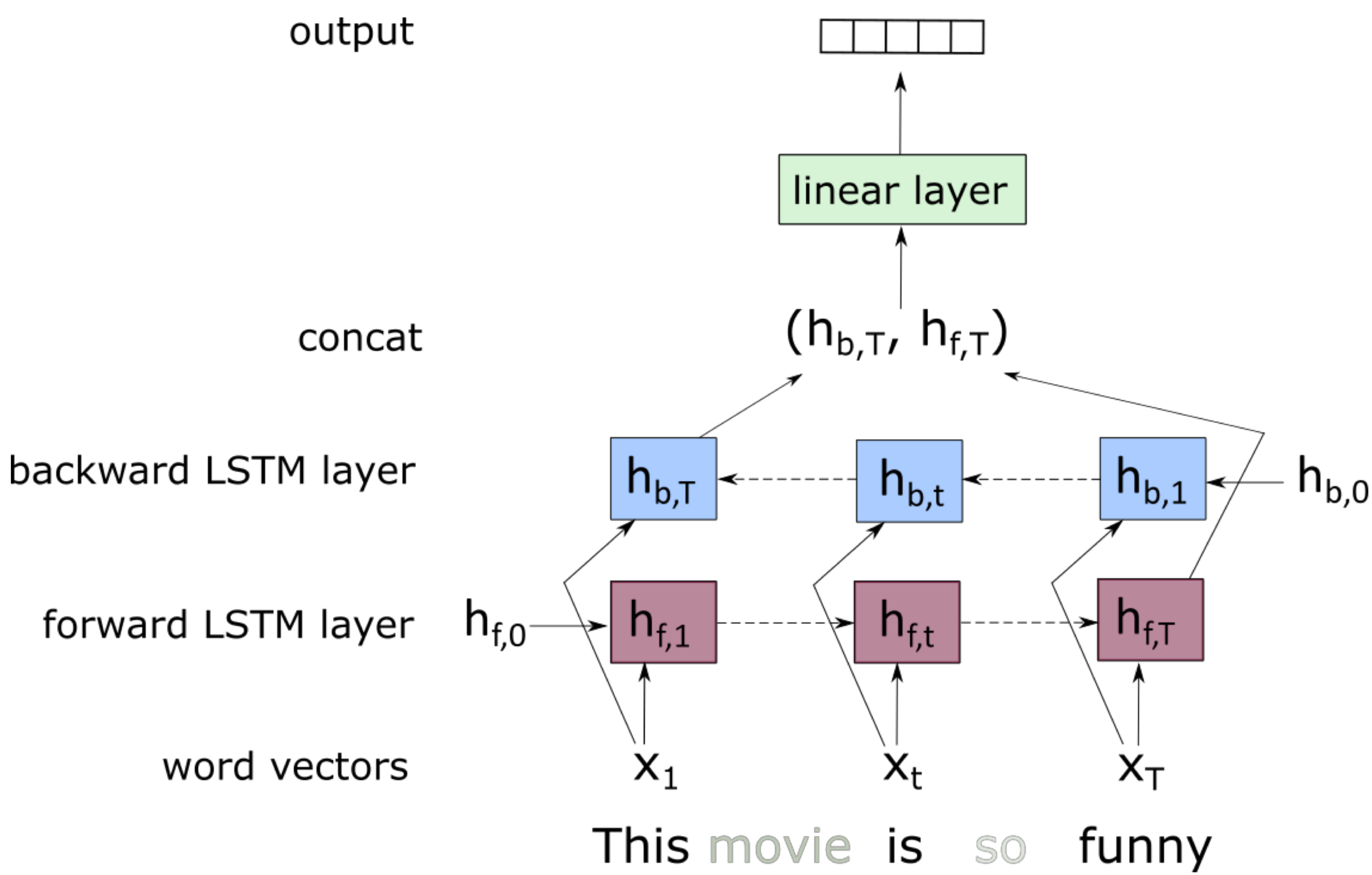
LSTM



BLSTM



BLSTM – bidirectional LSTM



BLSTM – Hyperparameters

5 Epochs

2 Layers

32 Batch Size

0.5 Dropout

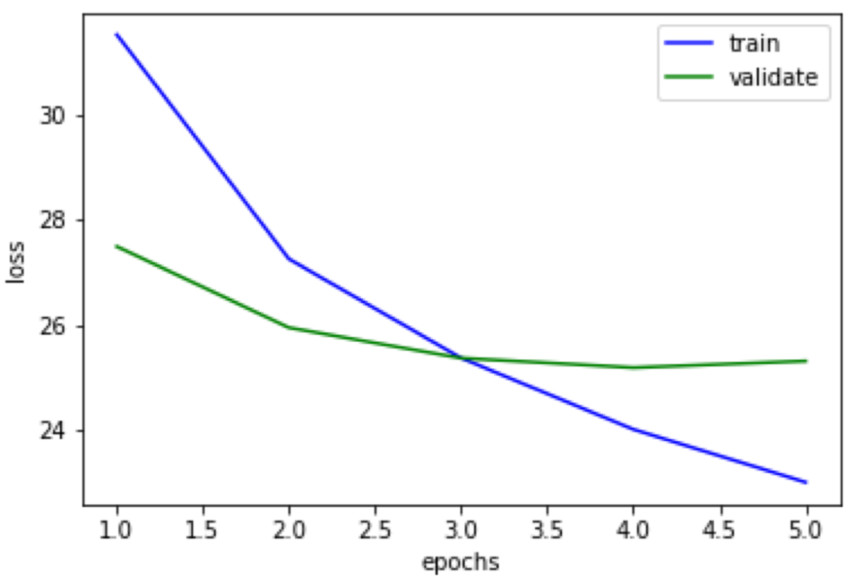
512 Hidden Dimensions

glove.6B.100d Word Embedding

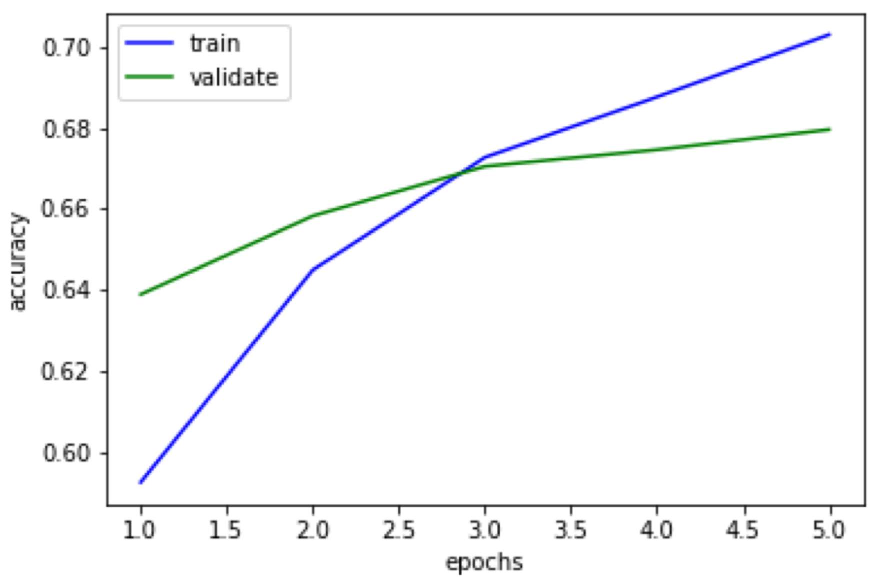
BLSTM – Results

- Test accuracy: 64.02%
- Kaggle test accuracy: 62.07%

Loss



Accuracy



Predict sentiment of sentences

Predict sentiment of all phrases of a sentence and compute:

- median
- mean
- truncated mean
- weighted mean

Sentiment of the sub-phrase with maximum length

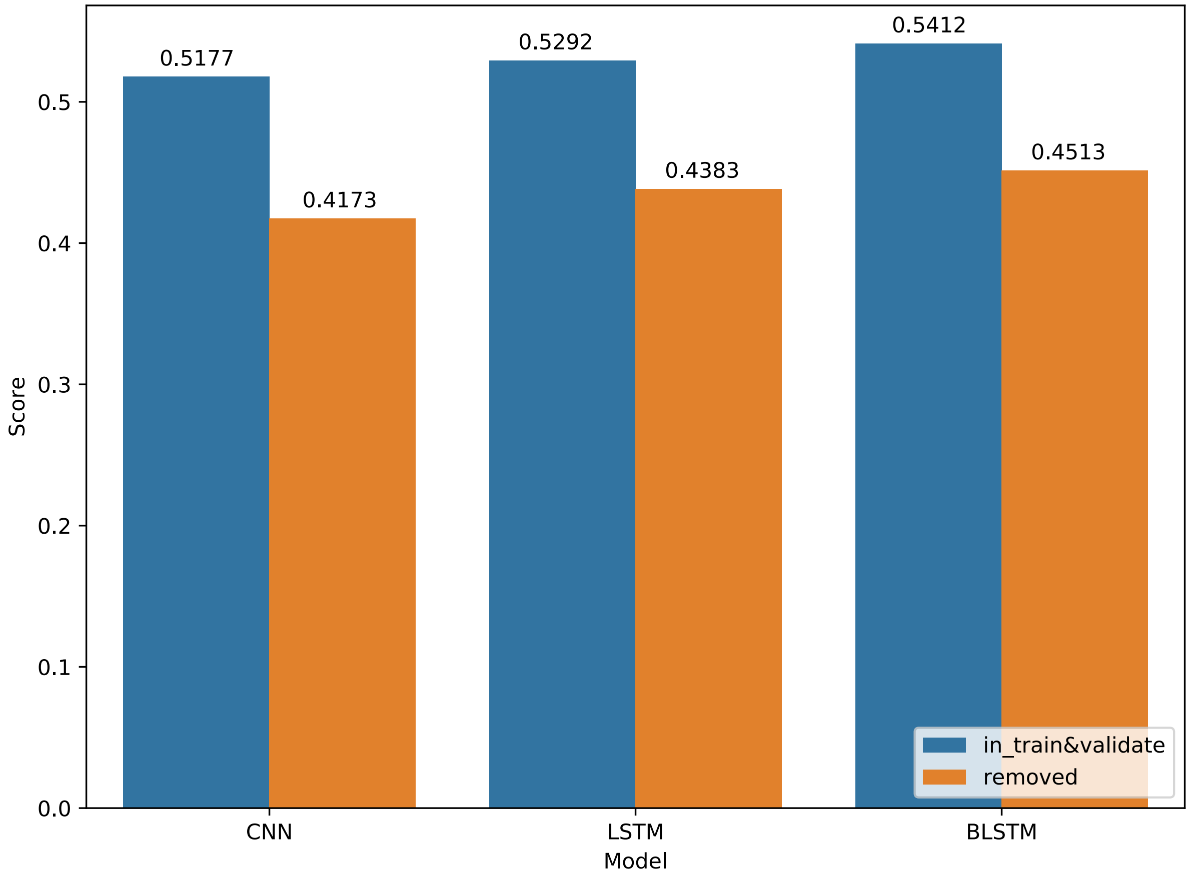
Use model trained on phrases

Predict Sentiment of Sentences

- Predict sentiment of all phrases of a sentence and compute:
  - median
  - mean
  - truncated mean
  - weighted mean
- Sentiment of the sub-phrase with maximum length
- **Use model trained on phrases**

Predict Sentiment of Sentences

Sentence Level Performance



Summary

CNN

- Good for Classification
- Performs well on short sentences and thus phrases
- Performance heavily dependent on filter and input size

LSTM

- Good for sequence learning
- Good on short sentences
- Bad for inconsistent labeled data

BLSTM

- Actually regards future context
- Important for sentiment due to contextual dependencies

Conclusion

- More Data
  - < 10.000 full sentences available for all sentiments
  - ~ 1.000 – 2.500 per sentiment
- Consistent labeling
  - A single fullstop should not alter the sentiment of a sentence
- Full sentence vs. phrases as training data

QUESTIONS?