Practice with NLP

Review Rating Prediction

Outline

Motivation behind NLP and sentiment analysis

Introduction to Yelp dataset and our research question

Methodology workflow

Exploratory Data Analysis

Overview of applied NLP techniques

Predictive modelling and evaluation

Our model from a practical perspective

Visualizing word embeddings in feature space

Conclusion and future work

Unstructured Data Is Sexy – You Just Don't Know It

Massive boom in the amount of NLP startups

"Since 2011 nearly 800 Natural Language Processing startups have been established worldwide with an average valuation of $4.8 million"[1]

Market trends indicates growing demand for NLP

"NLP market was at $280 million in 2015 and expected to reach $2.1 billion by 2024"[2]

"According to Time.com, Natural Language Processing skills are expected to boost your salary by 18%"[3]

Unstructured data is by far the biggest source of data

"Unstructured data is growing at the rate of 62% per year"[4]

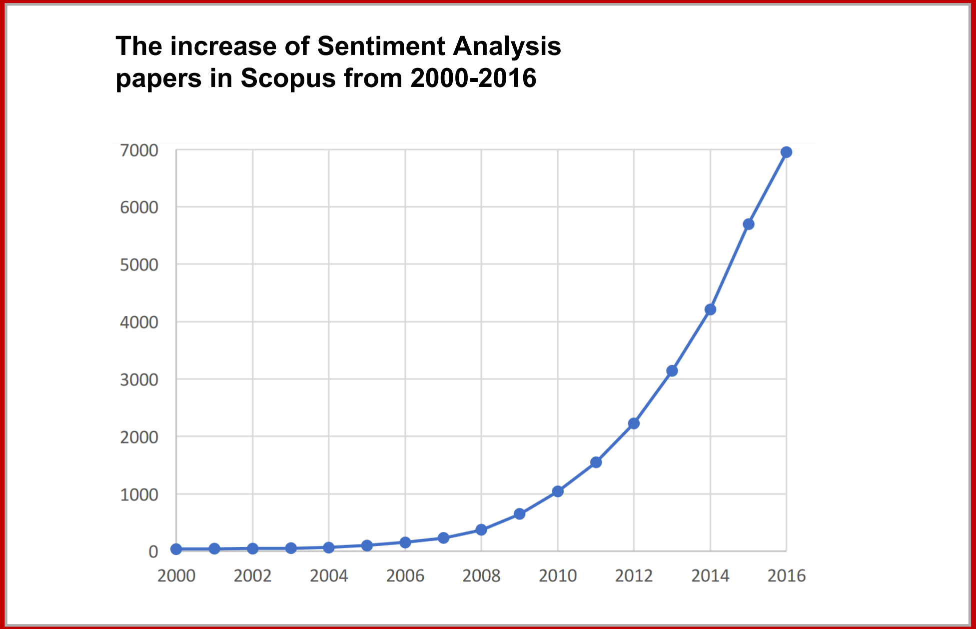
"Data volume is set to grow 800% over the next 5 years and 80% of it will reside as unstructured data"[5]

1: Angel.co 2017, 2: Gartner 2015, 3: Time.com, 2017 4: IDG 2016, 5: Gartner 2017

… even in the Academia

"Nearly 7,000 papers of sentiment analysishave been published … 99% of the papers have appeared after 2004 making sentiment analysis one of the fastest growing research areas"[1]

"Searches made with a search string "sentiment analysis" in Google search engine have increased nearly 800% since 2014"[2]



1, 2: The Evolution of Sentiment Analysis, 2016 – Kuutila et al.

Introduction to the Yelp Official Dataset

Overview

Yelp Open Dataset from September **2017**

More than **1.1** million unique users

4.7 millions reviews of local businessess across **4** countries

More than **10** different categories of services

Motivation

Scientifically bulletproof
More than **38,000** scientific papers have been published based on the Yelp dataset

Learning
Excellent toy dataset to conduct experiments via machine learning techniques

Reviews
The variety of the sentiment and the length of the reviews across different businesses makes the dataset highly useful

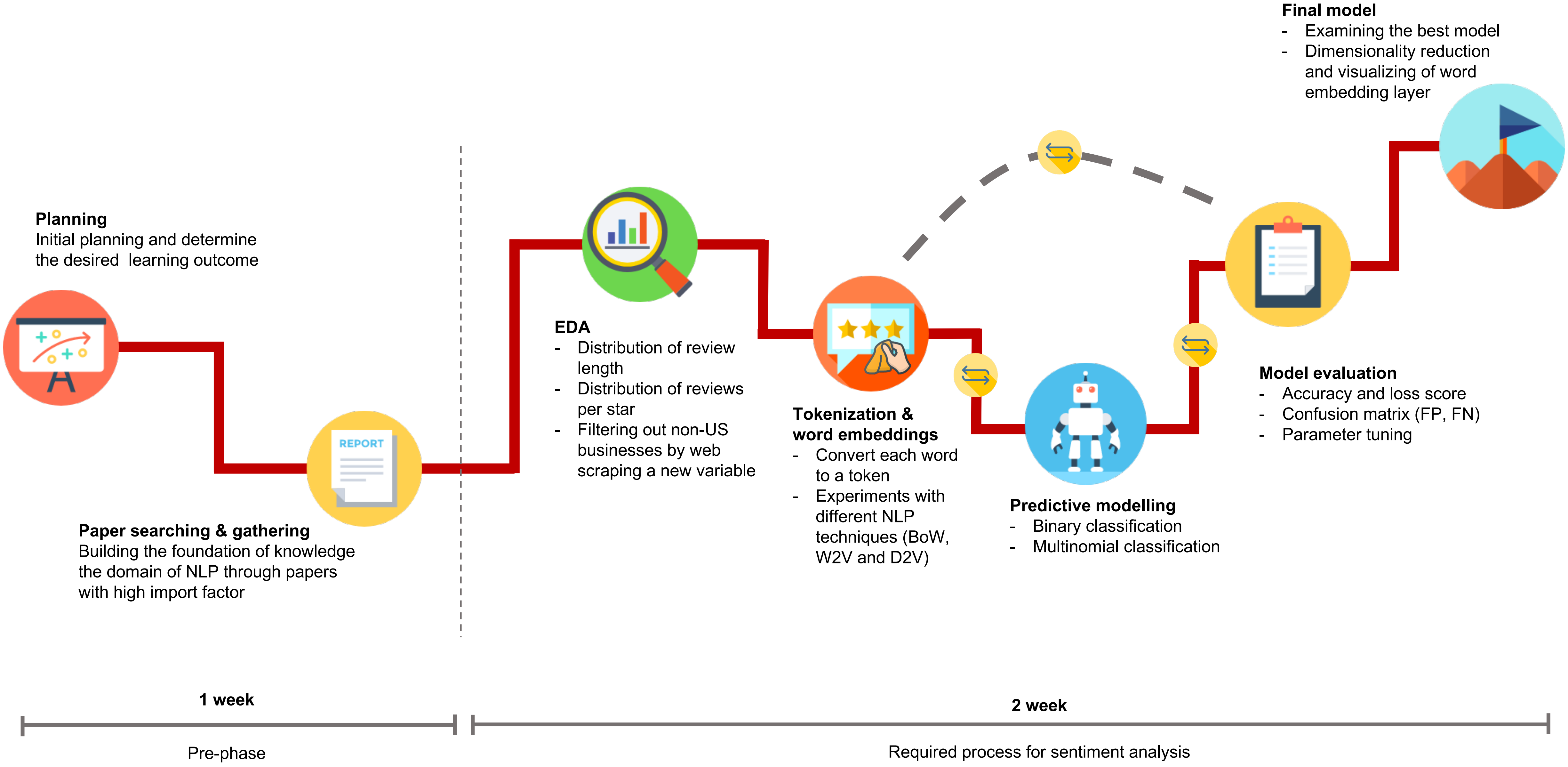
Problem statement

*Conduct several experiments to capture the **semantic** relationships between reviews, and we use different **techniques** to predict the **sentiment** and star rating*

*A **1** star increase in rating results in **5-9%** increase in revenue*

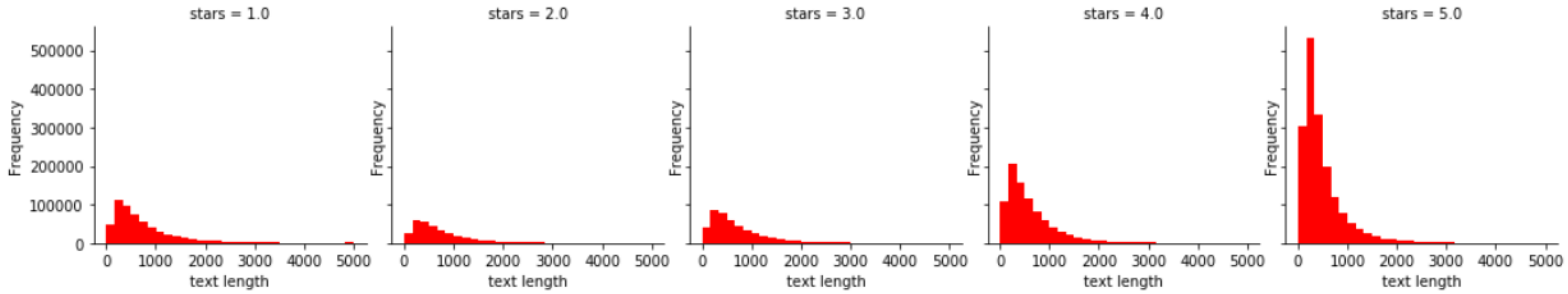
__1__ negative review can cost 30 customers

Methodology workflow

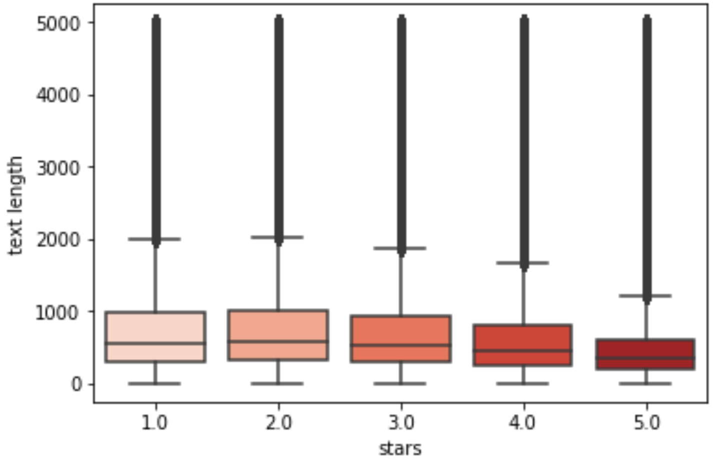


Exploratory Data Analysis

Distribution of review length across different star ratings



Quartile overview of star ratings



Data preparation

Focusing on all types of reviews independent of the venue category

Filtering out non-US reviews by scraping all states from wiki resulting in 3.9 million reviews

Downsampling to 100k reviews for each class to avoid overrepresentation for same star ratings

Overview of applied NLP techniques

Applied NLP techniques in our experiments

We applied **3** different NLP techniques to the Yelp dataset by starting with the most **simple** technique and proceeded with more **complex** word embedding models

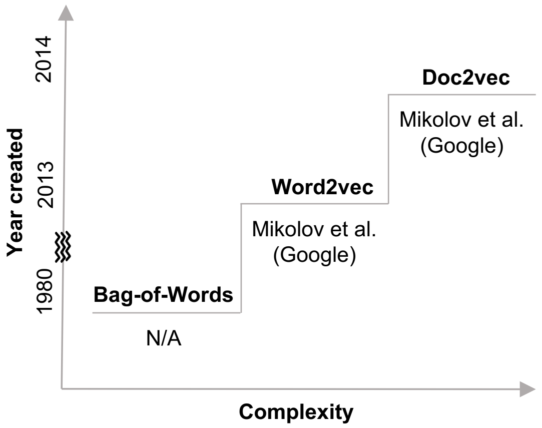
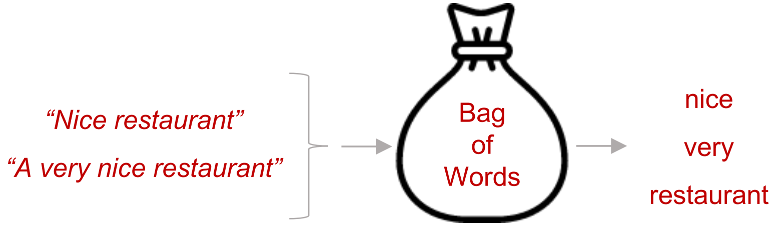
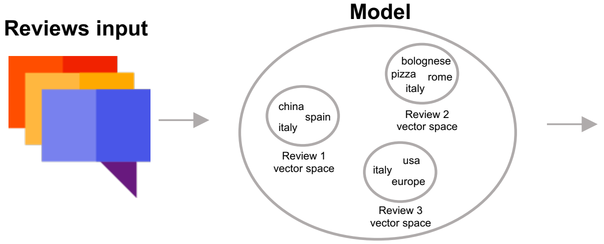


Illustration of the Bag-of-Words model



- The simplest method of representing text when modelling text with ML algorithms
- Cannot capture the semantics relationship between reviews since it ignores the context
- Select top words, letters only, lowercase, remove stop words and do stemming

Illustration of the Doc2vec model



Output

Most_similar("italy"):

rome	0.87
bolognese	0.76
pizza	0.65
…	

Most_similar("review1"):

review3	0.90
review2	0.60*
…	

- Learns representations for words and labels simultaneously
- Can be used to identify similar reviews or restaurants with similar reviews

Two methods:

Paragraph Vector Distributed Memory (PV-DM)

Paragraph Vector Distributed Bag of Words (PV-DBOW)

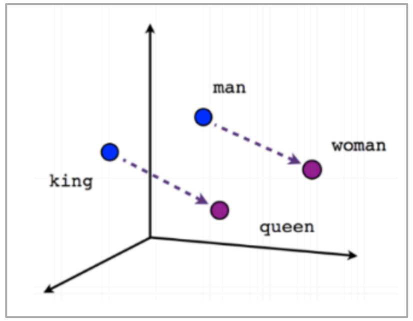
*Based on cosine similarity

Illustration of the Word2vec model

Vectors that represent similar words are close by different distance measures

Two methods:
CBOW
Skip-gram

Word2vec will be elaborated on the next slide…



It's illegal to talk about word2vec without attaching this plot

Using word2vec model for learning word embeddings from raw text

Pre-processing to build word2vec model

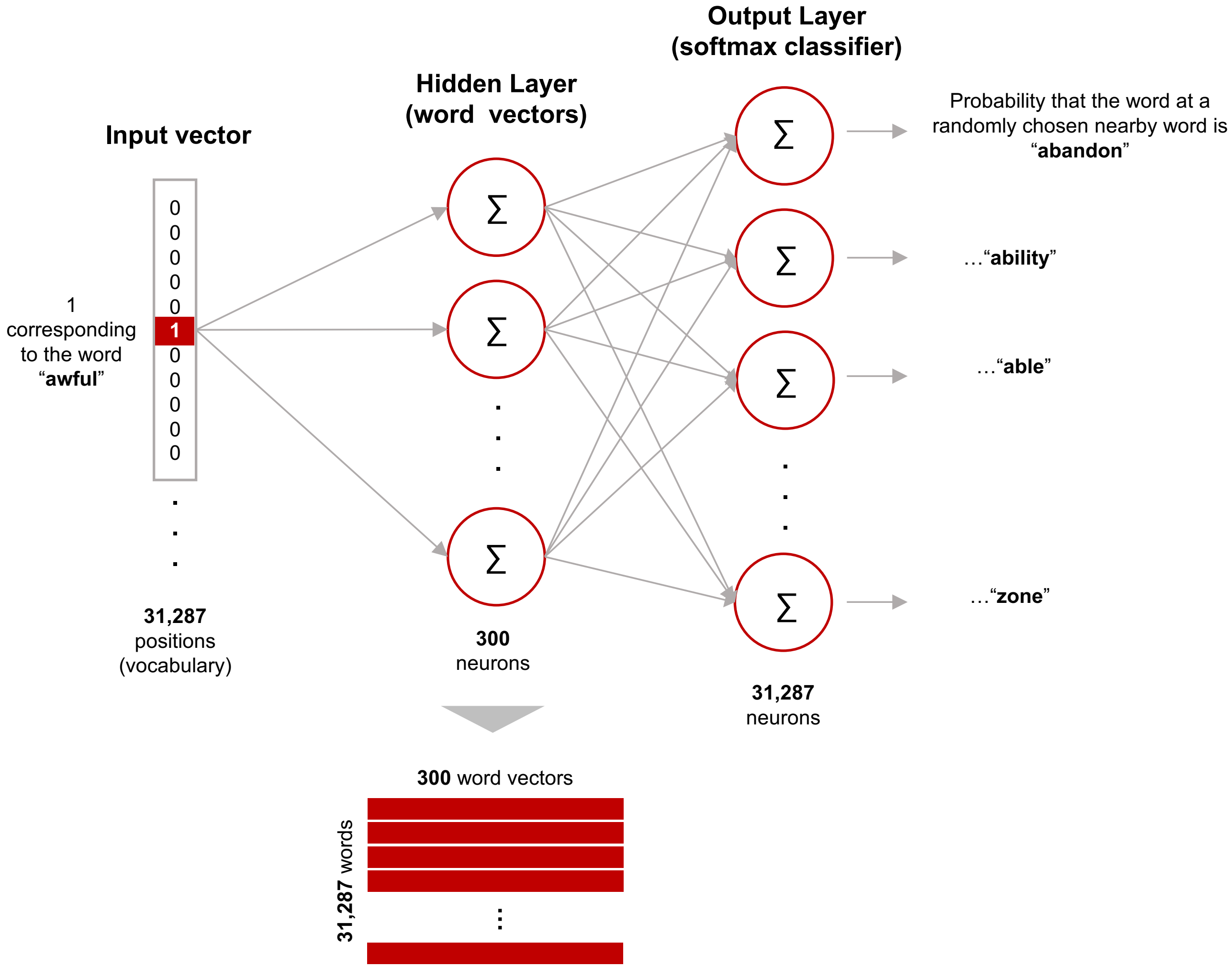
1. Build iterator to import 1 review at a time
2. Make the review to lowercase
3. Remove symbols and numbers
4. Convert each word to a token
5. Append the tokenized word to a list
6. Discard the original review and import next review

"Decent customer service but the food was awful. It was cold and had no sauce at all. I was expecting it to be good but this place really went down hill. I will never eat here again."

['decent', 'customer', 'service', 'but', 'the', 'food', 'was', '**awful**', 'it', 'was', 'cold', 'and', 'had', 'no', 'sauce', 'at', 'all', 'i', 'was', 'expecting', 'it', 'to', 'be', 'good', 'but', 'this', 'place', 'really', 'went', 'down', 'hill', 'i', 'will', 'never', 'eat', 'here', 'again']

"In my experience, it is usually good to disconnect (or remove) punctuation from words, and sometimes also convert all characters to lowercase" – Mikolov

Skip-gram neural network



Binary classification

Predictive modelling for sentiment analysis

Logistic regression

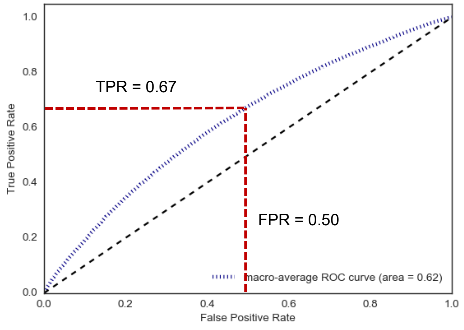
Embeddings	Test score
Bag-of-Words	58.95
Word2vec	64.76
Doc2vec	64.32

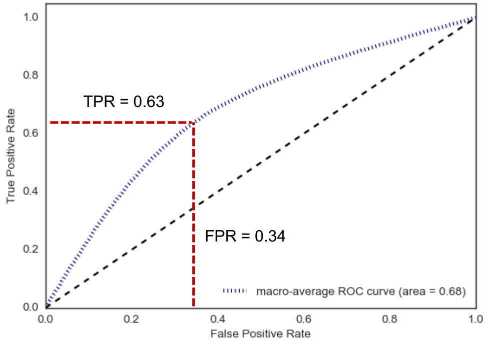
Bag-of-Word

Predicted sentiment \ True sentiment	Positive	Negative
Positive	13430	9994
Negative	6424	10152

Word2vec

Predicted sentiment \ True sentiment	Positive	Negative
Positive	12633	6882
Negative	7205	13258

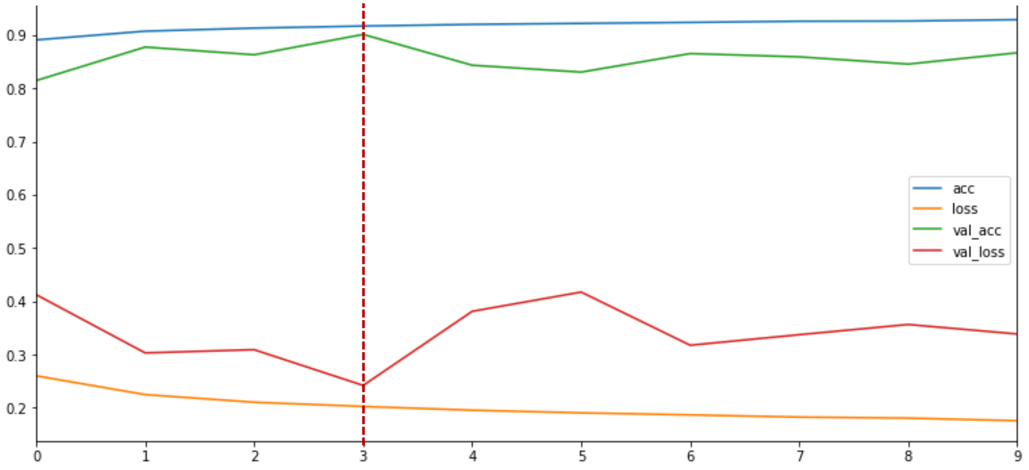




Convolutional Neural Network

Parameters	Value
Embedding	(20000, 300, 300)
Conv1D	(filters = 64, window = 5, activation = ReLu)
Maxpooling1D	4
LTSM	128, 300
Output activation function	Sigmoid
Loss function	Cross entropy
Results (Keras embedding)	**88.36**
Results (word2vec embedding)	**90.13**

Loss and accuracy plot (word2vec embedding)

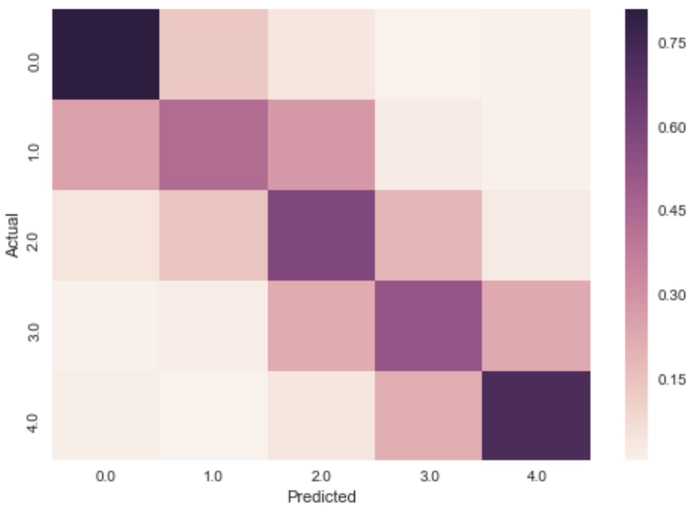


Multinomial classification

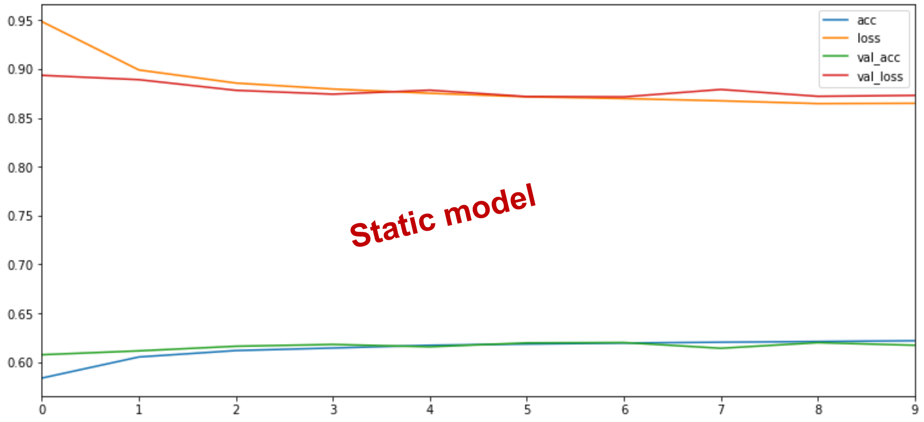
Predictive modelling for multinomial star rating prediction

Convolutional Neural Network – word2vec embedding

Confusion matrix

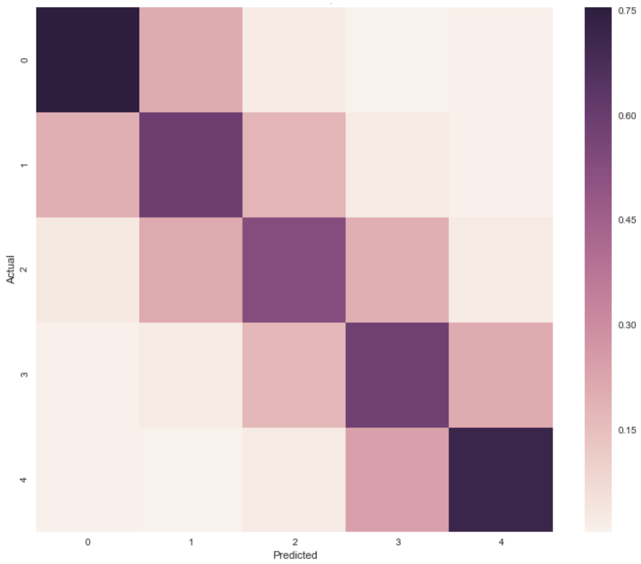


Loss and accuracy plot

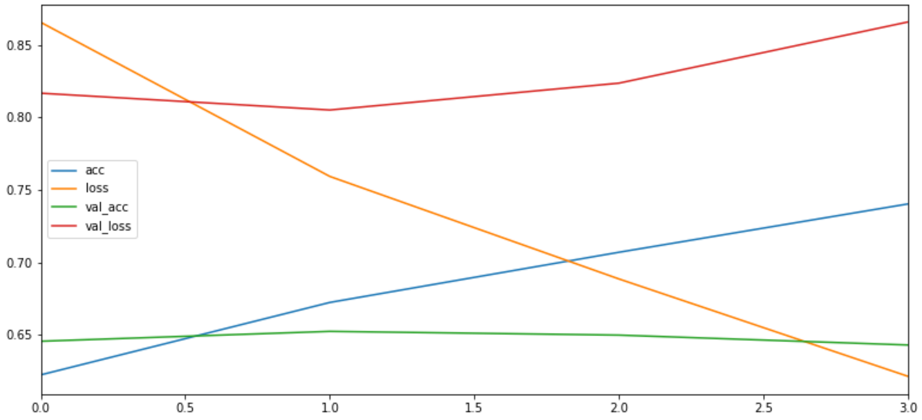


Convolutional Neural Network – Keras embedding

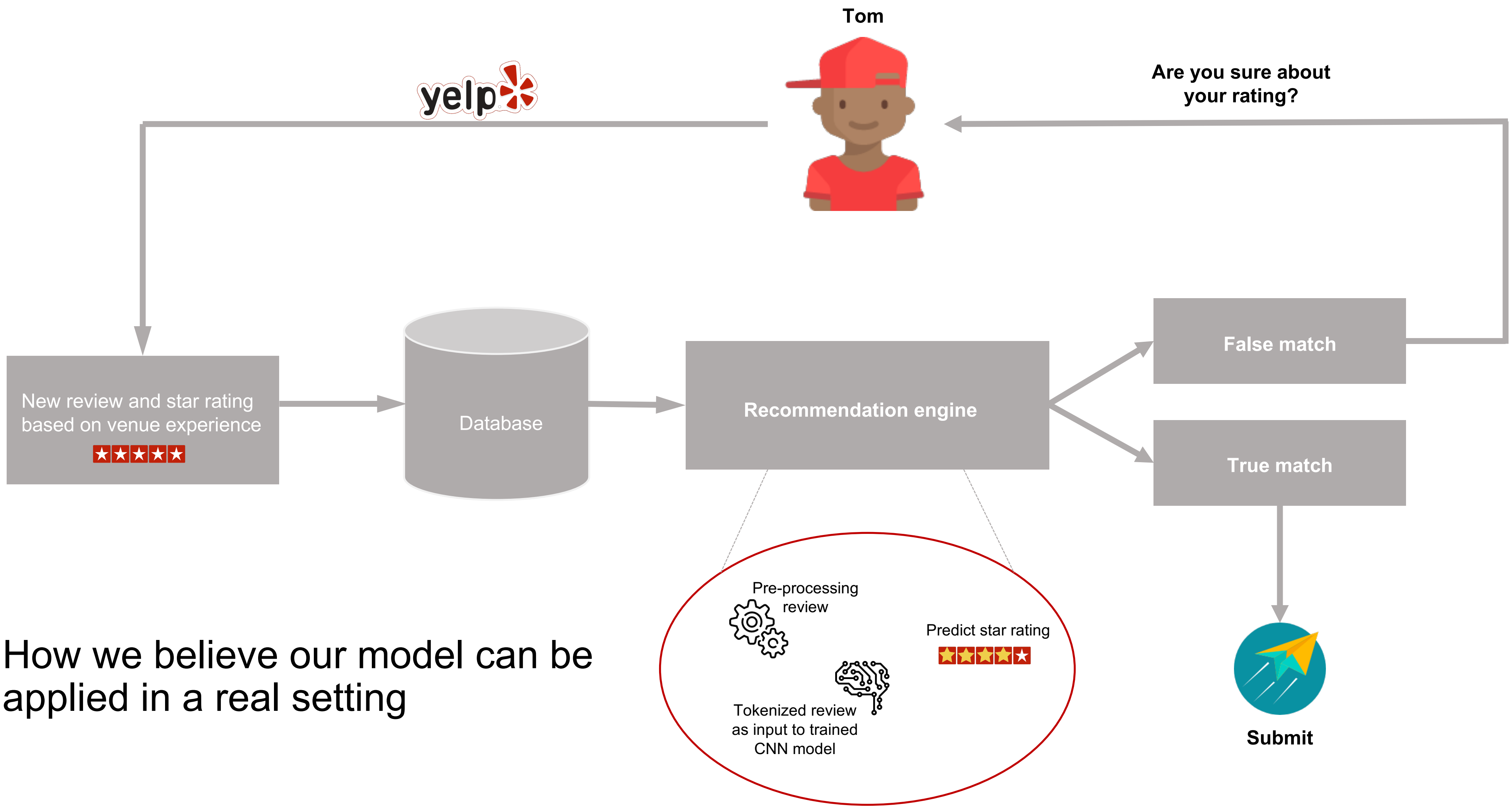
Confusion matrix



Loss and accuracy plot

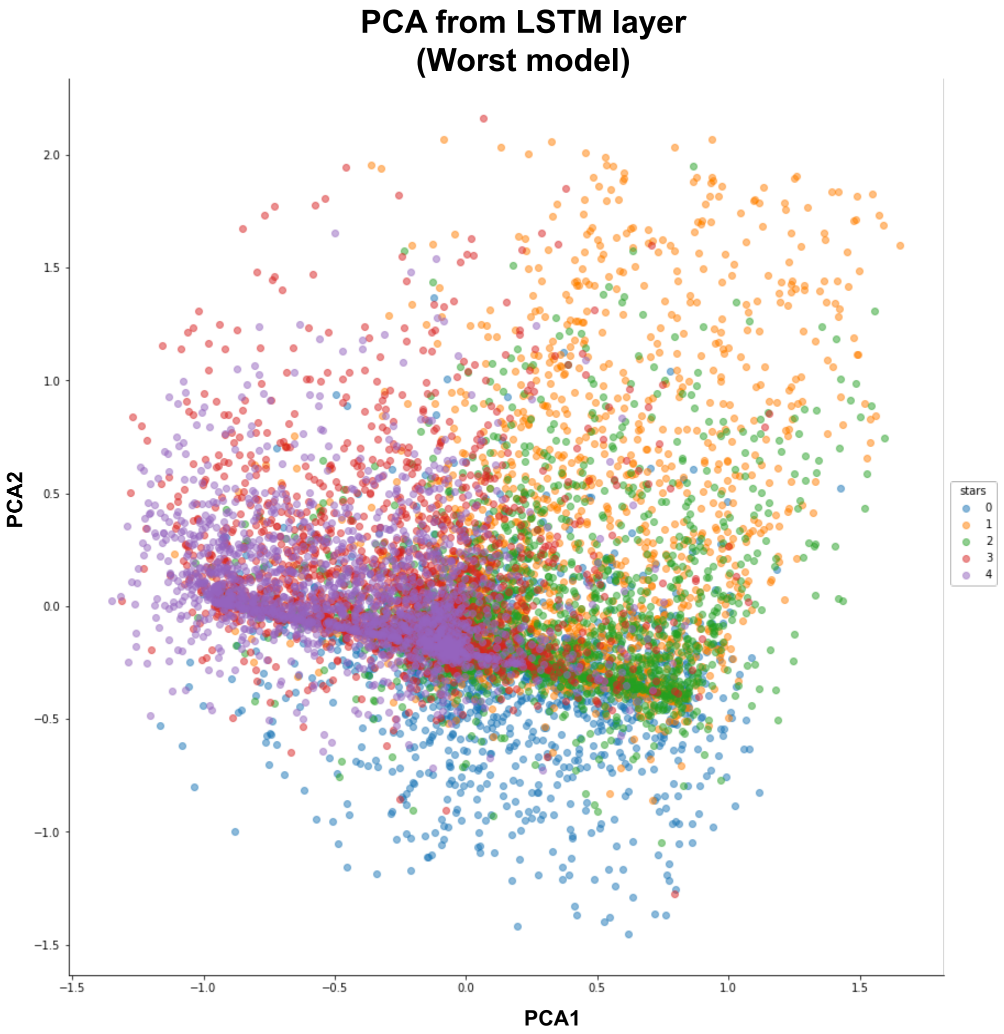


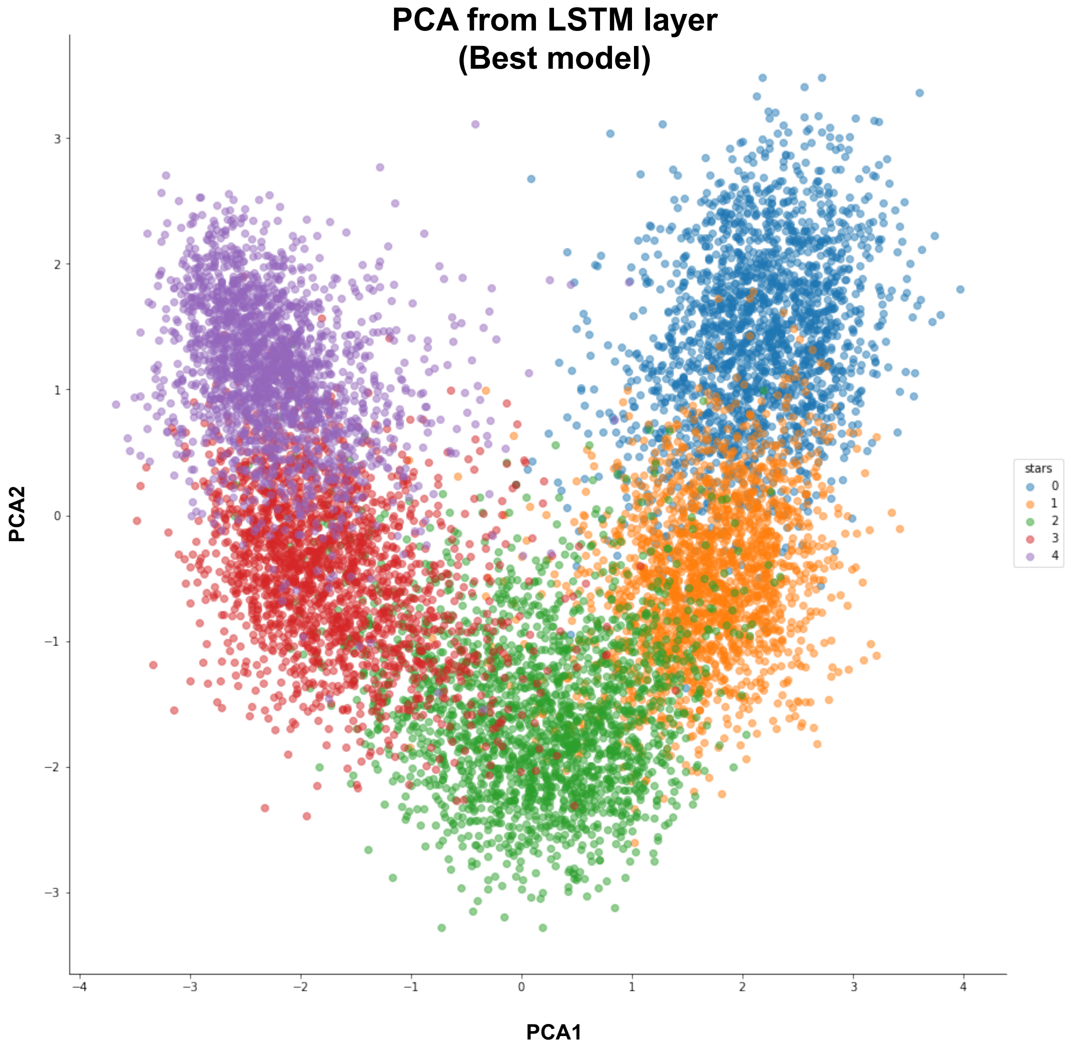
Our model from a practical perspective



How we believe our model can be applied in a real setting

Visualizing word embeddings in feature space





Summing up

Conclusion

- Difference between self and pre-trained embedding models
- Length of word vectors has a significant impact on the performance of the predictive models
- Acknowledge that performance relies significantly on the specific applied context

Future work

- Combine different embedding techniques (e.g. PV-DM + PV-DBOW)
- Conduct experiments with GloVe
- Increase the number of reviews
- Play around with the CNN hyper parameters
- Apply problem to other business settings

Thank you!