Natural Language Processing
Sentiment Analysis of Yemeksepeti comments in Turkish

Toygar Tanyel
18011094

17.12.2021

Abstract

Sentiment Analysis has an important role in today's world especially for private companies which hold lots of data. The massive amount of data generated by internet present a unique opportunity for sentiment analysis. However, it is challenging to build an accurate predictive model to identify sentiments, which may lack sufficient context due to the length limit. In addition, sentimental and regular ones can be hard to separate because of word ambiguity. In this project, I will be proposing the phases of text pre-processing, visual analysis and modeling.

1 Introduction

Mobile Apps has been increasingly popular for people to share instant feelings, emotions, opinions, stories, and so on. As a leading food delivery platform, Yemeksepeti has gained tremendous popularity since its inception. People comment on the comments section of the restaurants which they order meal to spread goodness or badness of the restaurant. Therefore, enough data is generated for sentiment analysis to give an intuition to people who are seeking delicious food. The comments are not clean. That means we will start with the data cleaning, and continue with the changing dataset in order to make it useful for our purpose. Word2Vec, CNN and Bidirectional LSTM will be used and explained later in the paper.

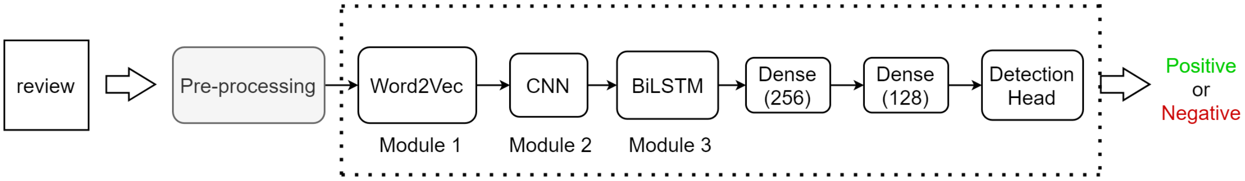


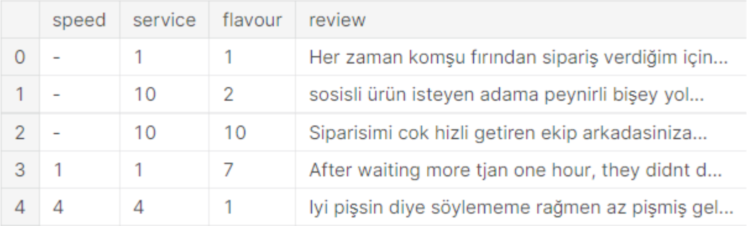
Figure 1: The proposed Word2Vec-CNN-BiLSTM learning pipeline for sentiment prediction.

The rest of this paper is organized as follows: Section 2 covers the dataset description and shows handling with imbalanced data; Section 3 explain the technical details of the proposed learning model; Section 4 provides experimental validation with result analysis; Section 5 summarizes my work.

2 Material and Methods

Entire code in notebook for this task can be found here.

The dataset was created by Doğukan Arslan from Turkish restaurant reviews taken from the food ordering site yemeksepeti.com. The dataset also includes scores in speed, service and flavour categories. In this respect, it is expected to be used in aspect-based sentiment analysis studies. Therefore, for sentiment extraction, dataset named Yorumsepeti choice is reasonable and challenging.

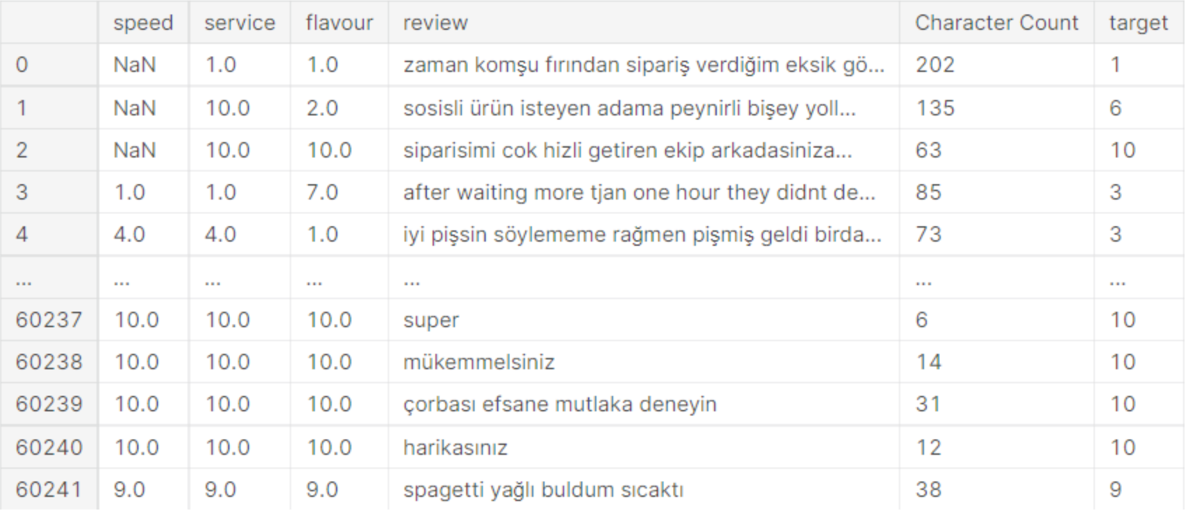


	speed	service	flavour	review
0	-	1	1	Her zaman komşu fırından sipariş verdiğim için...
1	-	10	2	sosisli ürün isteyen adama peynirli bişey yol...
2	-	10	10	Siparisimi cok hizli getiren ekip arkadasiniza...
3	1	1	7	After waiting more tjan one hour, they didnt d...
4	4	4	1	Iyi pişsin diye söylememe rağmen az pişmiş gel...

Figure 2: Raw Dataset

2.1 Data Pre-Processing

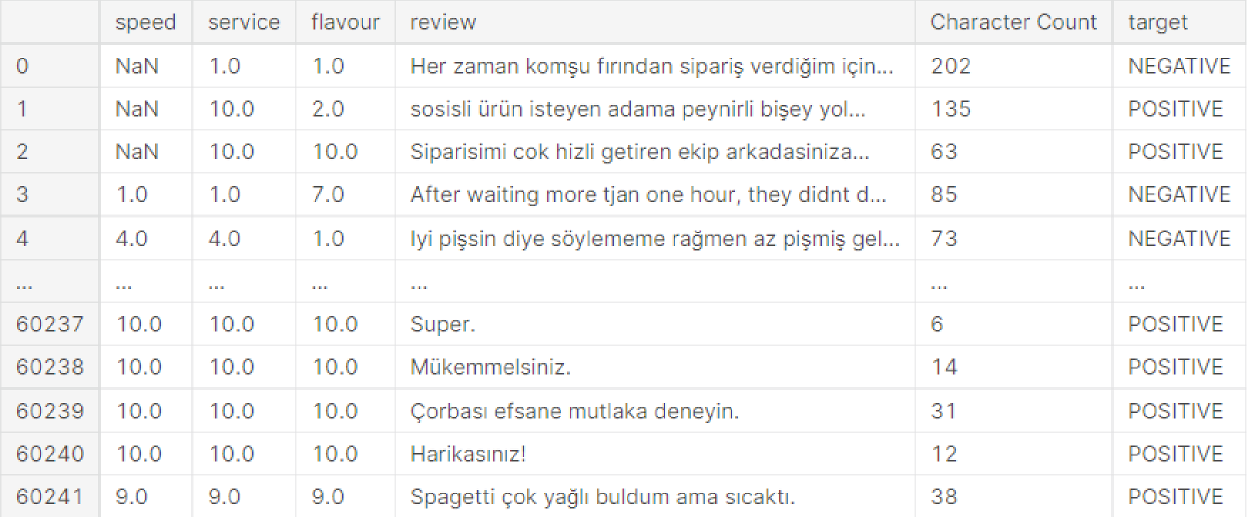
As explained before dataset includes 3 features labelled as speed, service and flavour out of 10 points. For sentiment analysis, it is difficult to reach state-of-art results by using 10 classes for every point due to lack of size of dataset. Therefore, as pre-processing step, the '-' character is deleted from the dataset and inserted NaN instead. This is done because, for every row, we will take the mean of 3 features and create new column as target to hold rounded mean scores(int). Meanwhile, dataset has NaN values for reviews which mean that we cannot use those rows for any sentiment analysis. Therefore, we will drop NULL review rows. In raw dataset, 33 rows has no reviews, and the latest dataset size is 60206 rows.



	speed	service	flavour	review	Character Count	target
0	NaN	1.0	1.0	zaman komşu fırından sipariş verdiğim eksik gö...	202	1
1	NaN	10.0	2.0	sosisli ürün isteyen adama peynirli bişey yoll...	135	6
2	NaN	10.0	10.0	siparisimi cok hizli getiren ekip arkadasiniza...	63	10
3	1.0	1.0	7.0	after waiting more tjan one hour they didnt de...	85	3
4	4.0	4.0	1.0	iyi pişsin söylememe rağmen pişmiş geldi birda...	73	3
...	...	...	...	...	...	...
60237	10.0	10.0	10.0	super	6	10
60238	10.0	10.0	10.0	mükemmelsiniz	14	10
60239	10.0	10.0	10.0	çorbası efsane mutlaka deneyin	31	10
60240	10.0	10.0	10.0	harikasınız	12	10
60241	9.0	9.0	9.0	spagetti yağlı buldum sıcaktı	38	9

Figure 3: New Dataset

To convert task to binary sentiment classification, we map target column as 1-5 points correspond to negative, and 6-10 points correspond to positive.



	speed	service	flavour	review	Character Count	target
0	NaN	1.0	1.0	Her zaman komşu fırından sipariş verdiğim için...	202	NEGATIVE
1	NaN	10.0	2.0	sosisli ürün isteyen adama peynirli bişey yol...	135	POSITIVE
2	NaN	10.0	10.0	Siparisimi cok hizli getiren ekip arkadasiniza...	63	POSITIVE
3	1.0	1.0	7.0	After waiting more tjan one hour, they didnt d...	85	NEGATIVE
4	4.0	4.0	1.0	Iyi pişsin diye söylememe rağmen az pişmiş gel...	73	NEGATIVE
...	...	...	...	...	...	...
60237	10.0	10.0	10.0	Super.	6	POSITIVE
60238	10.0	10.0	10.0	Mükemmelsiniz.	14	POSITIVE
60239	10.0	10.0	10.0	Çorbası efsane mutlaka deneyin.	31	POSITIVE
60240	10.0	10.0	10.0	Harikasınız!	12	POSITIVE
60241	9.0	9.0	9.0	Spagetti çok yağlı buldum ama sıcaktı.	38	POSITIVE

Figure 4: Mapped Dataset

Dataset has imbalanced distribution as follows:

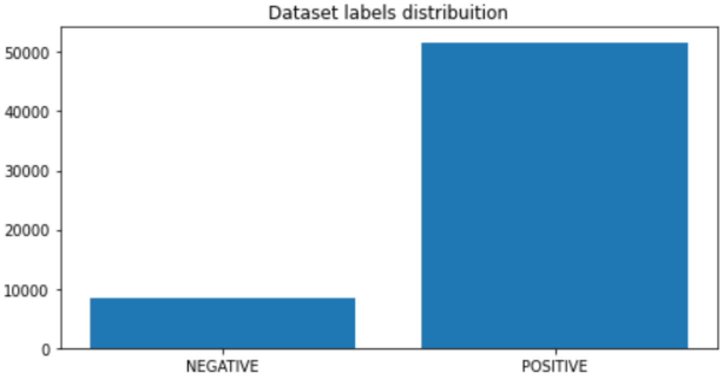


Figure 5: Distribution of Dataset

The imbalance problem is solved by SMOTE algorithm using tokenized dataset. It will be explained later on. To stemming text of reviews, TurkishStemmer library is used, and for stopwords, NLTK('turkish') is preferred.

2.2 Tokenize

Tokenization refers to splitting up a larger body of text into smaller lines, words or even creating words for a non-English language. The various tokenization functions in-built into the nltk module itself and can be used in programs.

2.3 Imbalanced Data Distribution Handling

SMOTE [1] (synthetic minority oversampling technique) is one of the most commonly used oversampling methods to solve the imbalance problem. SMOTE synthesises new minority instances between existing minority instances. It generates the virtual training records by linear interpolation for the minority class. These synthetic training records are generated by randomly selecting one or more of the k-nearest neighbors for each example in the minority class. After the oversampling process, the data is reconstructed and several classification models can be applied for the processed data. We had 1: 46502, 0: 7683 distribution in training set. After SMOTE it becomes to 0: 46502, 1: 46502.

3 Model

Figure 1 shows the proposed Word2Vec-CNN-BiLSTM learning pipeline, which consists of three sequential module.

Recently, Word2Vec model still produces competitive results by using as embedding matrix of deep learning models.

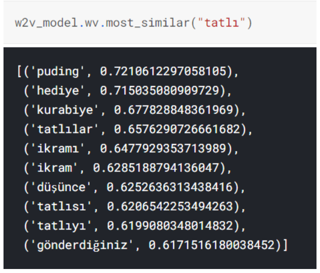
The combination of CNN and BiLSTM models requires a particular design, since each model has a specific architecture and its own strengths:

- CNN is known for its ability to extract as many features as possible from the text.
- BiLSTM keeps the chronological order between words in a document, thus it has the ability to ignore unnecessary words using the delete gate.

The purpose of combining these two models is to create a model that takes advantage of the strengths of CNN and BiLSTM, so that it captures the features extracted using CNN, and uses them as an LSTM input. Therefore, I develop a model that meets this objective, such that the vectors built in the word embedding part are used as convolutional neural network input.

3.1 Word2Vec

Word2Vec [4] is a technique for natural language processing. Word2vec is a group of related models that are used to produce word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located close to one another in the space. Here an example of closest words to chosen one:



```
w2v_model.wv.most_similar("tatlı")
```

```
[('puding', 0.7210612297058105),
 ('hediye', 0.715035080909729),
 ('kurabiye', 0.677828848361969),
 ('tatlılar', 0.6576290726661682),
 ('ikramı', 0.6477929353713989),
 ('ikram', 0.6285188794136047),
 ('düşünce', 0.6252636313438416),
 ('tatlısı', 0.6206542253494263),
 ('tatlıyı', 0.6199080348014832),
 ('gönderdiğiniz', 0.6171516180038452)]
```

Figure 6: Most similar words for ”tatlı”

The architecture of implementation above:

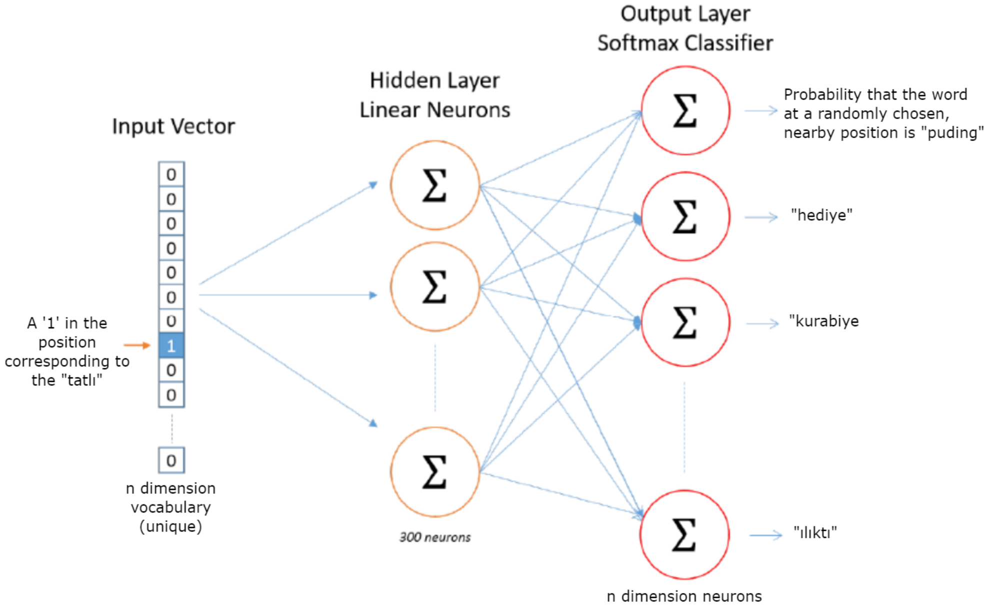

Input Vector
Hidden Layer
Linear Neurons
Output Layer
Softmax Classifier
Σ
0
1
A '1' in the position corresponding to the "tatlı"
n dimension vocabulary (unique)
300 neurons
Probability that the word at a randomly chosen, nearby position is "puding"
"hediye"
"kurabiye
"ılıktı"
n dimension neurons

Figure 7: Architecture

3.2 CNN (Convolutional Neural Networks)

CNN is a class of deep, feed-forward artificial neural networks (where connections between nodes do not form a cycle) & use a variation of multilayer perceptrons designed to require minimal preprocessing. [3] shows that a simple CNN with little hyperparameter tuning and static vectors achieves remarkable results on multiple benchmarks. Learning task-specific vectors through fine-tuning offers further gains in performance.

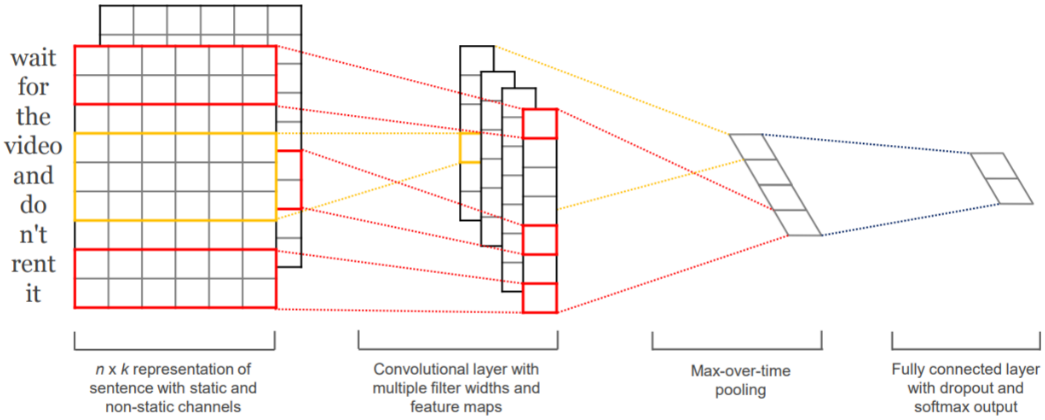

wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 8: Model architecture with two channels for an example sentence.

3.3 Bi-LSTM (Bidirectional Long Short-term Memory)

Original LSTM [2] (Long Short-term Memory) previously published at 1997 to solve problem which called back-propagation through time. To be more clear, learning to store information over extended time intervals by recurrent back-propagation takes a very long time, mostly because of insufficient, decaying error back-flow.

A Bidirectional LSTM, is a sequence processing model that consists of two LSTM, one taking the input in a forward direction, and the other in a backwards direction. Bi-LSTM effectively increase the amount of information available to the network, improving the context available to the algorithm. For further information go through the original LSTM paper.

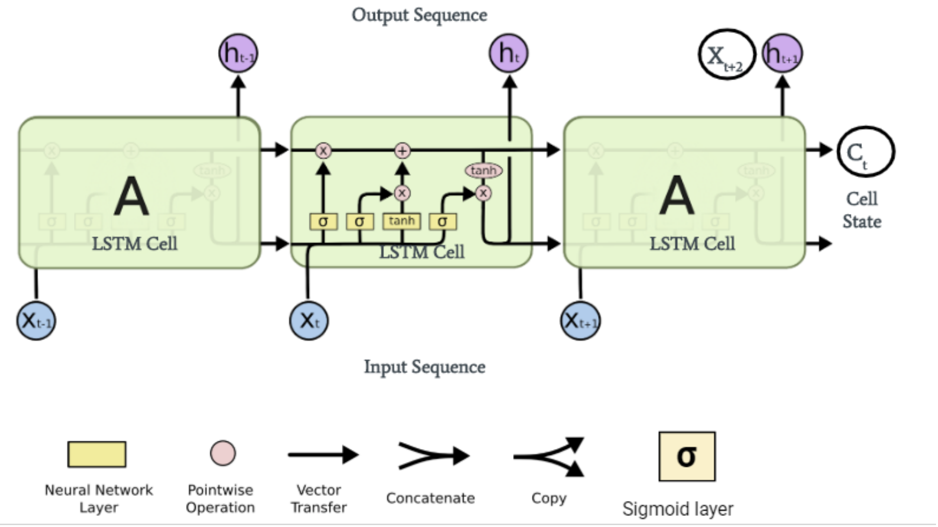


Figure 9: The LSTM cell architecture.

4 Results and Observations

I also conducted experiments to evaluate a set of models, and present a performance comparison of all evaluated models in table 1. The set of models CNN, BiLSTM, CNN-BiLSTM, and CNN-BiLSTM-Dense forms an ablation study, from which we can evaluate the performance of each individual module and the combined versions. It can be seen that the pure CNN model performs the worst since a single-layer CNN cannot learn any contextual information. Therefore, could not obtain any results. BiLSTM present an obvious improvement. My final model, BiLSTM-CNN-Dense tops every other model, showing its power to combine the strength of each individual building block. Another observation is that even though the its accuracy is high, I see pure BiLSTM model does not perform robust results in terms of hard cases. You can confirm that from table 3.

Table 1: A performance comparison of models.

Model	Precision	Recall	Accuracy
BiLSTM	90.01	86.32	86.41
CNN-BiLSTM	90.23	85.74	85.03
CNN-BiLSTM-Dense	90.35	87.42	86.51

True/false prediction from random sentences:

Table 2: Prediction performance of models.

Model	Sentence	Prediction
BiLSTM	”Kuryeniz çok saygısız”	’label’: ’POSITIVE’, ’score’: 0.72
CNN-BiLSTM-Dense	”Kuryeniz çok saygısız”	’label’: ’NEGATIVE’, ’score’: 0.45
BiLSTM	”EFFFFSANEYDİ BEEE”	’label’: ’POSITIVE’, ’score’: 0.80
CNN-BiLSTM-Dense	”EFFFFSANEYDİ BEEE”	’label’: ’POSITIVE’, ’score’: 0.85,
BiLSTM	”Bi tantuni yiyelim dedik kusacaktık reziller sizi”	’label’: ’NEGATIVE’, ’score’: 0.56
CNN-BiLSTM-Dense	”Bi tantuni yiyelim dedik kusacaktık reziller sizi”	’label’: ’NEGATIVE’, ’score’: 0.46

One of the most important observation is that the problem with imbalanced raw dataset. The problem shows up after training phase. Even though the model training and test accuracy makes us think it is convenient model to use, the model learns positive overmuch. Overfeed leads to model cannot predict properly random sentences. I solved problem by synthetic data generation using SMOTE algorithm.

Example:

Table 3: Raw Dataset Prediction performance of models.

Model	Sentence	Prediction
BiLSTM	”Kuryeniz çok saygısız”	’label’: ’POSITIVE’, ’score’: 0.69
CNN-BiLSTM-Dense	”Kuryeniz çok saygısız”	’label’: ’POSITIVE’, ’score’: 0.68
BiLSTM	”İşini aşkla yapan bir mekan daha sarrrrdı”	’label’: ’POSITIVE’, ’score’: 0.91
CNN-BiLSTM-Dense	”İşini aşkla yapan bir mekan daha sarrrrdı”	’label’: ’POSITIVE’, ’score’: 0.99

5 Summarize

Sentiment analysis is highly related to people’s daily lives, and recent years have seen more research efforts dedicating to this field. Research on sentiment prediction helps augment people’s awareness, and improve the mechanism of a service. This paper investigates a novel model for sentiment prediction using a dirty comment data. My model, Word2Vec-BiLSTM-CNN-Dense extract high-quality linguistic features for sentiment prediction. Although the proposed model is trained and validated on an Turkish dataset, it can be applied to datasets in other languages.

References

[1] Kevin W. Bowyer et al. “SMOTE: Synthetic Minority Over-sampling Technique”. In: *CoRR* abs/1106.1813 (2011). arXiv: 1106.1813. URL: http://arxiv.org/abs/1106.1813.

[2] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-term Memory”. In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[3] Yoon Kim. “Convolutional Neural Networks for Sentence Classification”. In: *CoRR* abs/1408.5882 (2014). arXiv: 1408.5882. URL: http://arxiv.org/abs/1408.5882.

[4] Tomas Mikolov et al. “Distributed representations of words and phrases and their compositionality”. In: (2013), pp. 3111–3119.