UKRAINIAN CATHOLIC UNIVERSITY

MASTER THESIS

# Determining sentiment and important properties of Ukrainian-language user reviews


*Author:*
Dmytro BABENKO

*Supervisor:*
Vsevolod DYOMKIN


*A thesis submitted in fulfillment of the requirements for the degree of Master of Science*

*in the*


Department of Computer Sciences
Faculty of Applied Sciences


Lviv 2020

# Declaration of Authorship

I, Dmytro BABENKO, declare that this thesis titled, “Determining sentiment and important properties of Ukrainian-language user reviews” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Master of Science

**Determining sentiment and important properties of Ukrainian-language user reviews**

by Dmytro BABENKO

# *Abstract*

Every day a lot of visitors leave countless reviews about hotels, restaurants, cafes, attractions or other services. In most cases, they set the rate about this service, sometimes they also set the rate about the specific topic if service provides this possibility. However, the main information about user opinion is hidden inside the body of review text. Thereby, in this work, we propose a solution to analyze one or several user reviews, determine sentiments and acquire important characteristics for these reviews. We determine which characteristics were influenced by such reviews. In this case, the proposed solution can detect sentiments from text and classify for positive and negative. Then it acquires top positive and negative phrases, which can explain why the user left such review. Besides, we analyze all reviews about one hotel or just several reviews and summarize the most important positive and negative properties for a specific hotel.

# *Acknowledgements*

Firstly, I want to thank my supervisor Vsevolod Dyomkin (Ukrainian Catholic University), who directed me for research of this thesis and provided a lot of advice regarding approaches that solve the specified problem.

Secondly, I would like to thank my seminar mentor, Artem Chernodub (Ukrainian Catholic University) who was getting in touch about the process of the work in each seminar and prompted helpful ideas.

Finally, I am grateful to Ukrainian Catholic University and Oleksii Molchanovskyi, the coordinator of the Data Science Master Program, which was an important step in my career and life experience.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| **NLP** | **N**atural **L**anguage **P**rocessing |
| **RNN** | **R**ecurrent **N**eural **N**etwork |
| **LSTM** | **L**ong **S**hort-**T**erm **M**emory |
| **RCNN** | **R**ecurrent **C**onvolution **N**eural **N**etwork |
| **MI** | **M**utual **I**nformation |
| **PMI** | **P**ointwise **M**utual **I**nformation |
| **TF-IDF** | **T**erm **F**requency-**I**nverse **D**ocument **F**requency |

*For/Dedicated to/To my…*

# Chapter 1

# Introduction

## 1.1 Motivation

A lot of services would like to understand whether their clients are satisfied or not. For this reason, they usually ask to leave a review about their service or some product. In many cases when selecting a service or choosing a product, the amount of information available in user reviews can be or greater volume and more trustworthy than the official product description provided by the vendor. The reviews left by clients can help services to become better and attract new consumers. Usually, the number of reviews for specific service items (i.e hotel, restaurant) can be overwhelming. The review reader can gain an understanding of overall satisfaction by looking at summary statistics such as the average rating or score, but the details about what makes service good or not are hidden inside the body of the reviews. With such a large amount of information to process, it would be beneficial to automate the analysis of readers reviews and generate a summary of the reviews. Furthermore, many reviews can help services know which points clients are most interested in.

## 1.2 The proposed method

Considering all of the above and having made several experiments we propose the following:

- classify positive and negative sentences in the review
- determine important phrases in the review text
- summarize important properties for all reviews about a specific hotel

The dataset which contains Ukrainian-language user reviews was parsed from Booking.com[1] and TripAdvisor[2]

## 1.3 Goals of the master thesis

The following are the main goals of this master thesis:

1. Monitor available solutions for sentiment analysis in English language
2. Collect Ukrainian-language user reviews dataset from available sources

[1] booking.com
[2] tripadvisor.com.

3. Use deep learning techniques based on the collected dataset to estimate the rating of the review and classify review text for positive, negative, or neutral
4. Apply NLP and Machine Learning techniques to extract important properties from one or several reviews

## 1.4 Thesis structure

To begin, we provide an overview of related work to our task for English-language, and some information about sentiments for Ukrainian-language in chapter 2. Then, in chapter 3, we give a dataset description and explain how it was collected. Moreover, in chapter 4, we introduce background information about several Machine Learning and Natural Language Processing approaches which we used and describe them. The results of applying the processed methods are presented and described in chapter 5. To conclude, we sum up our contributions and list of directions for future work in chapter 6.

# Chapter 2

# Related work

## 2.1 Pre-process data

In most works with sentiment analysis, before using some model, the data is reviewed and pre-processed. For example, in one paper (Kaur, Sehra, and Sehra, 2017) of IJCSE[1] related to literature review of sentiment analysis, the authors proposed several steps for prepossessing data. The following pre-processing steps have been incorporated in their work to improve accuracy while using any sentiment analysis methods:

- Punctuation erasure. It removes punctuation marks like the period, exclamation point, comma, apostrophe, question mark, quotation mark, and hyphen.
- Number Filter. It allows filtering the numbers.
- N chars filter. It removes words with less than the pre-specified number of 3 characters.
- Case Converter. Lowercase all presented words.
- Stemmer. It allows stemming the terms present in the text.
- Filtering stop words. It allows removing all the terms that represent stopwords like *the*, *is*, *on*, *at* etc

In another paper (Hemalatha, Varma, and Govardhan, 2012) of IJCSE, the authors proposed a deeper explanation of pre-processing data for efficient sentiment analysis. They suggested several additional tasks which data pre-processing involves:

- Removing URLs. URLs do not contribute to analyzing the sentiment in the informal text.
- Filter repeated letters in the words. For example, in some positive sentence user can write words like *happyyyyy* to show his emotions.
- Questions words. The words like which, what, how, etc do not influence polarity.
- Removing special characters. Special characters like.,[]()/' should be removed to remove discrepancies during the assignment of polarity.
- Removal of Retweets. This usually happens if a user likes another user's tweet.

[1] https://www.ijcseonline.org

## 2.2 Predicting rating of user review

In most cases, rating is set by users, but sometimes it is not mentioned and we need a model which can predict this rating.

One paper of the 19th Conference on Information and Communication Technologies in Tourism (Gräbner et al., 2012), proposed a system that performs the classification of customer reviews of hotels using sentiment analysis. Here, they use 5 class labels (1 star, 2 stars, 3 stars, 4 stars, and 5 stars). They elaborate on a process to extract a domain-specific lexicon of semantically relevant words based on a given corpus. The lexicon in the present study was generated on the base of the vocabulary in the training set only. The resulting lexicon backs the sentiment analysis for generating a classification of the reviews

On the other hand, in the study of Saumya et al. (2018) they use their predicted helpfulness score to rank the overwhelming number of product reviews. The helpfulness score is predicted using features extracted from review text data, product description data and customer question-answer data of a product using random-forest classifier and gradient boosting regressor.

Of course, there are a lot of research for multi-class text classification using Neural Nets. One of them, Zhou et al. (2015) proposed C-LSTM for sentiment classification. They consider two classification tasks on the movies review dataset: fine-grained classification with 5 labels (very positive, positive, neural, negative, very negative) and binary classification by removing neural labels.

## 2.3 Sentiment analysis

Sentiment is the technique for measuring the polarity of input text, i.e. how much positive or negative content the text has. *Oh! This is a beautiful hotel*. That shows positive sentiment. *It's the most terrible breakfast, which I have ever eaten before.* That shows negative sentiment. Nowadays, sentiment analysis is a popular topic in NLP, and a lot of machine learning techniques are also being used.

Since in this work we focus on user reviews and comments about hotels, we found several similar projects for English and even for non-English language. For instance, in paper (Kasper and Vela, 2011), the authors present a system that collects such comments from the web and creates classified and structured overviews of such comments and facilitates access to that information for German language. They provided the BEASOT system - an interactive web application, where core system on server-side handles *data acquisition*, *analysis*, and *storage* as shown in Fig. 2.1

In earlier paper, German scholars uses statistical polarity classifier for assigning to each text segment a polarity value. As a basis for statistical polarity classification they used the classification engine, provided by German Research Center for Artificial Intelligence (Steffen, 2004) which is based on *character n-gram* instead of terms. In this study (Steffen, 2004) they research the model used for the multi-lingual classification of documents according to the topics of the MEMPHIS[2] domains. They use a classification approach based on character-level n-grams. It is also very robust when working on "noisy" texts with spellings errors. However, they argument that using character-level n-grams results in less sparse data, because there are far fewer possible n-grams than there are possible terms and show good performance of this classification approach.

[2] http://www.ist-memphis.org

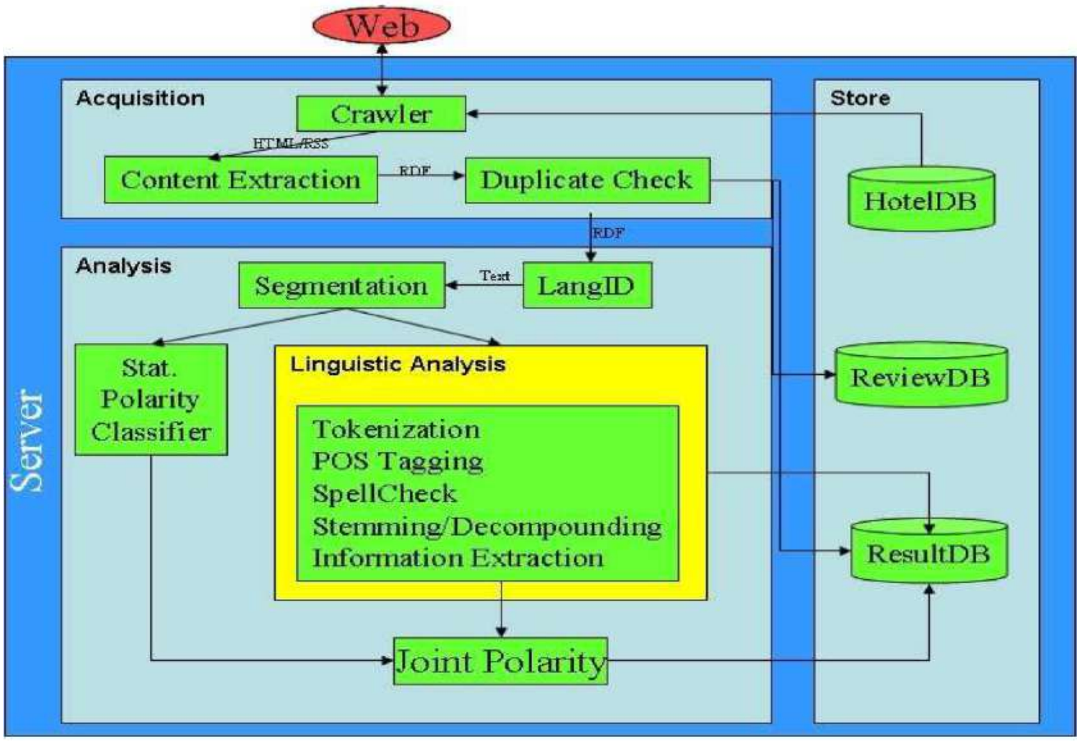



FIGURE 2.1: BESAHOT system overview, provided by German scientists (Kasper and Vela, 2011)

In the following paper (Shi and Li, 2011), authors pay attention to online hotel reviews, and propose a supervised machine learning approach using unigram feature with frequency and term frequency-inverse document frequency (TF-IDF) to realize polarity classification of documents. They show experimental results, where the information of TF-IDF is more effective than frequency. Furthermore, members of IEEE[3] (Das and Chakraborty, 2018) propose a technique for text sentiment classification using TF-IDF along with Next Word Negation (NWN). In this article, they ran several experiments using movie and product review dataset and claim that when TF-IDF is coupled with Next Word Negation then the performance of the sentiment classifier increases by a good percentage compared to a simple bag of words model or common TF-IDF model.

Moreover, there was some research for sentiment analysis of reviews in Ukrainian language conducted by Romaniuk and Romanyshyn, 2013. They describe the necessity of named-entity recognition for the implementation of sentiment analysis and presents methods and tools for recognition of appropriate named entities in Ukrainian restaurant reviews. The authors identified types of entities commonly used in Ukrainian restaurant reviews. The stages of named-entity recognition have been defined: named entities identification and categorization.

[3] https://www.ieee.org

# Chapter 3

# Dataset description

As this work mostly focuses on Ukrainian-language reviews about hotels, here we describe how we collect this data, pre-process it and show how it is distributed. A lot of reviews about Ukrainian hotels can be found on Google reviews, TripAdvisor and of course on Booking.com. Unfortunately, there is no available Google API to parse the reviews and it is quite complicated to implement script for parsing google reviews data. At the same time, it is quite easy to parse the data from Booking.com and TripAdvisor and automate this process.

## 3.1 Collect data

Firstly, we considered Booking.com. There is a very good possibility to select reviews by the author's original language. Additionally, Booking.com asks clients to split their impressions into positive and negative text (see example in Fig. 3.1 ). This feature helps us a lot because we have already annotated dataset for text classification into positive and negative.

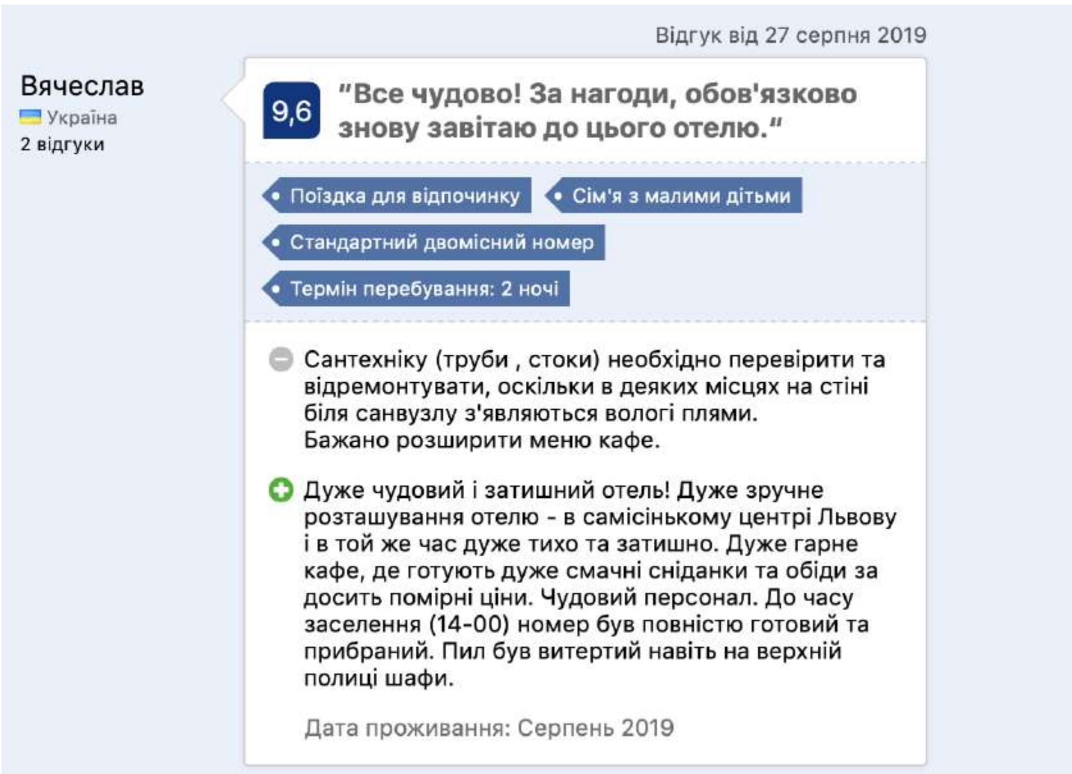



FIGURE 3.1: Example of user review in Booking.com

As a result, we collected approximately 28 000 Ukrainian-language reviews from Booking.com for hotels in Kyiv, Lviv, Odesa, Kharkiv, Dnipro, Ivano-Frankivsk, and Uzhgorod. Besides, Booking.com contains approximately 100 000 Russian-language reviews for the same hotels in these cities. To fill up our Ukrainian-language dataset, we translated Russian-language reviews for these hotels using google translate API into Ukrainian. Therefore, we have approximately 128 000 Ukrainian-language reviews.

Secondly, we monitored reviews about hotels for the same Ukrainian cities on TripAdvisor. Unfortunately, there are no reviews in Ukrainian language. However, there are a lot of Russian-language user reviews about Ukrainian hotels. One of them is illustrated in Fig.3.2. In comparison to Booking.com there is no split text into positive and negative, but TripAdvisor has 5 stars rankings that allow us to predict positive or negative reviews. So, we parsed approximately 36 000 Russian-language reviews and translated them into Ukrainian.

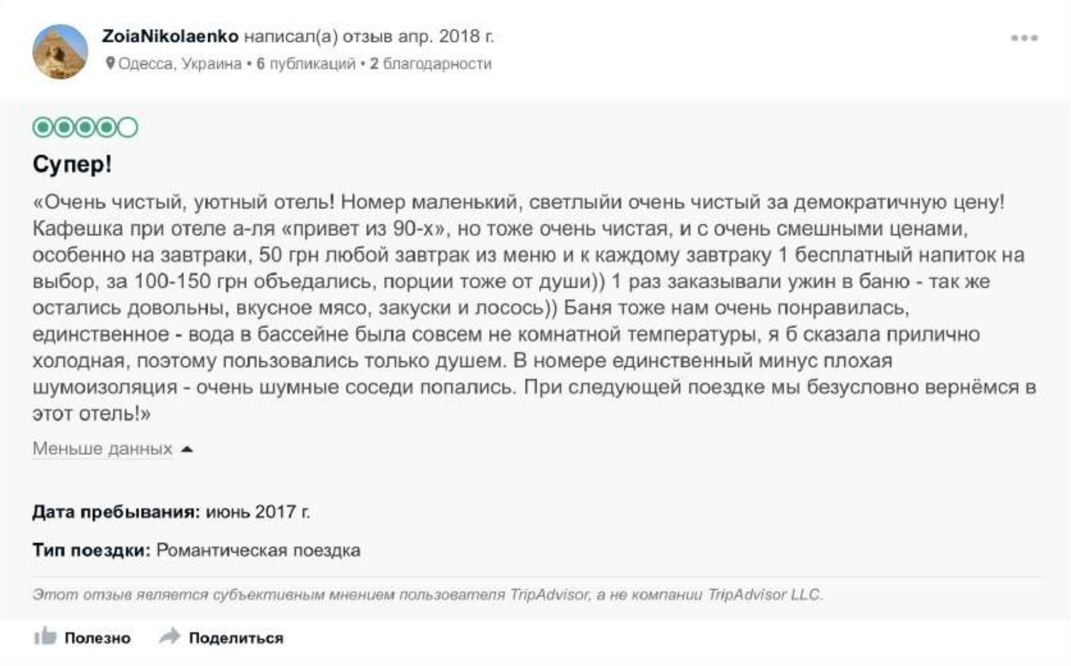



FIGURE 3.2: Example of user review in TripAdvisor

To summarise, we made a first ever attempt to create a deep learning-ready corpus of Ukrainian language about hotels, using Booking.com and TripAdvisor. In general, there are more than 164 000 reviews, where 28 0000 reviews are originally in Ukrainian, and 136 000 are Russian-language reviews translated into Ukrainian via google translate API.

## 3.2 Clean data

As we use the parsed data written by a lot of different users, it is not a secret that this data contains a lot of mistakes. Besides, we translated a lot of reviews from Russian into Ukrainian. In this case, we fixed incorrectly written words automatically. As in our work we use word embedding (Word2Vec), we implemented a simple word fixer which finds the most similar word in word embedding dictionary to the uncorrected written word. As a result, the numbers of fixed words are presented in Table 3.1 for three types of data: original Ukrainian reviews from Booking.com, translated from

Russian into Ukrainian reviews on Booking.com and also translated reviews from TripAdvisor.

| *Data* | *Fixed words* | *All words* |
|---|---|---|
| Booking.com uk | 45014 | 1034651 |
| Booking.com ru $\rightarrow uk$ | 136131 | 4346431 |
| TripAdvisor ru $\rightarrow uk$ | 76385 | 2857254 |
| *All* | 257530 | 8238336 |

TABLE 3.1: Number of fixed words in user reviews

To sum up, 257530 out of 8238336 words have bee corrected, which constitutes approximately 3%.

## 3.3 Split data

In common Machine Learning tasks, we split our data into train, test, and validation. In most cases, the ratio is 60:20:20 train/test/validation respectively. In our cases, we have a lot of reviews about hotels and each hotel has several reviews. We want to avoid a situation where reviews about the same would be in the train and in the test or validation data simultaneously. As a result, we split the unique hotel list into train, test, and validation using ration 60:20:20 respectively. Then, we use the data for these hotels like train, test and validation data. As a result, we have approximately a ratio between train test and validation data.

## 3.4 Data for text classification

Since Booking.com asks a user to split their impressions into positive and negative ones, it allows us to get directly annotated data for text classification into positive and negative. Furthermore, as we want to classify exact sentences into positive or negative, we tokenize existing data into sentences, and as a result, we get an expanded dataset for sentence classification.

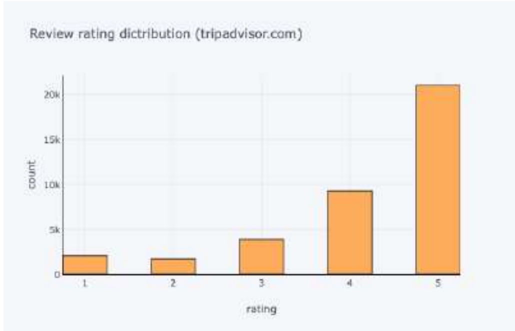
Bellow, we describe the dataset which we want to use for the text classification (positive or negative). Besides, we formulate a dataset for detecting the most valuable n-gram in the sentence. In this case, we have features such as a sentence and label as the most important n-gram. More details about how we formulate this dataset will be described describe in section 5.2

## 3.5 Visualize data

Currently, visually representing the content of a text is a quite difficult and important task in NLP. Nonetheless, visualizing unstructured (text) data and structured data have some gaps between each other. For instance, there is no representation of the text directly in many text visualizations, there is an output of a language model (word count, character length, word sequences, etc.). Inspired by the article of Li, 2019, we try to visualize and explain data as much as we can.

First of all, we would like to present a review rating distribution. In Fig. 3.3 we can see the distribution of Booking.com data and TripAdvisor data respectively. As we can see from Fig. 3.3 bellow, reviews with positive rating are significantly more frequent than with negative.

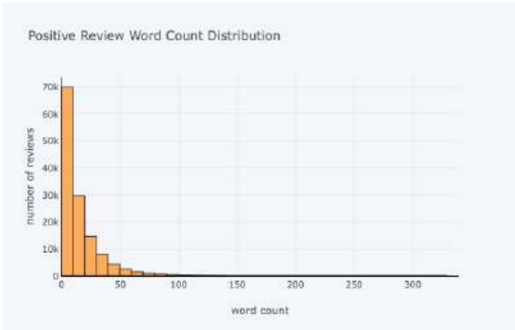

(A) booking.com



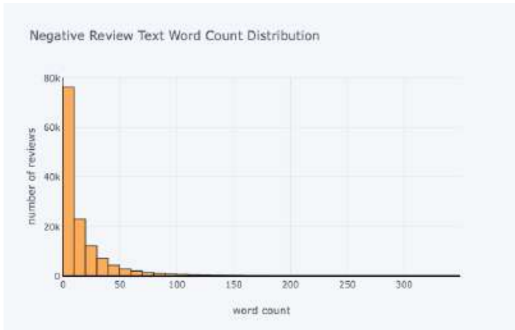

(B) TripAdvisor

FIGURE 3.3: Review rating distribution of available dataset

Secondly, we want to estimate word count distribution. From Fig. 3.4 we can see word count distributions for positive and negative texts from Booking.com reviews.



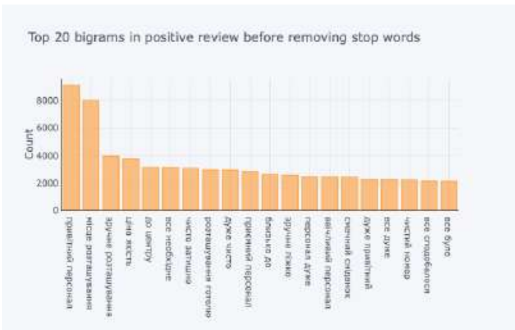

(A) in positive text



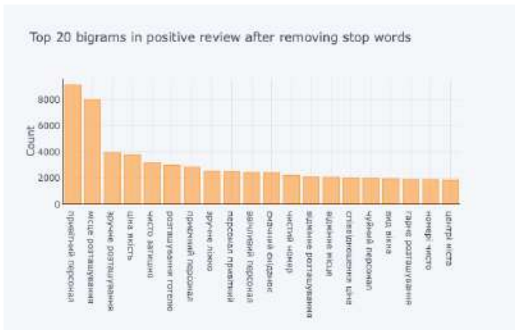

(B) in negative text

FIGURE 3.4: Word count distribution

Now we are going to explore the most popular N-grams. N-grams are used to describe the number of words used as observation points, e.g., unigram means singly-worded, bigram means the 2-worded phrase, and trigram means a 3-worded phrase. From figures 3.5 and 3.6 we see the most popular bigrams in positive and negative review text respectively. Furthermore, we compare the difference between bigrams using full text and bigrams using text after removing stop words.

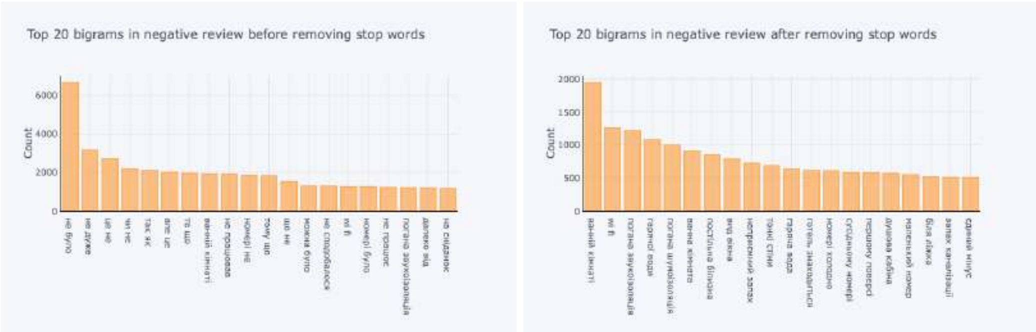



(A) before removing stop word




(B) after removing stop words

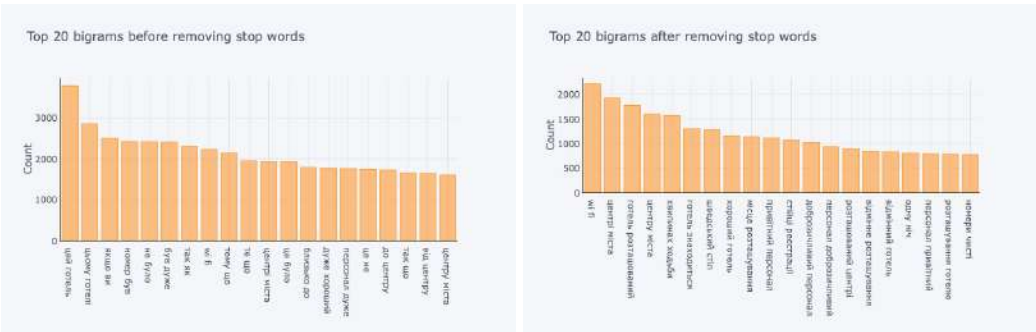
FIGURE 3.5: Top 20 bigrams in positive review text




(A) before removing stop words (B) after removing stop words

FIGURE 3.6: Top 20 bigrams in negative review text

Fig. 3.7 illustrates top 20 bigrams for full-text reviews from TripAdvisor. As there is no separation into the positive and negative text we can notice that most words in bigrams are similar to words in positive bigrams from Booking.com. It means that TripAdvisor contains probably more positive reviews than negative.




(A) before removing stop words (B) after removing stop words

FIGURE 3.7: 20 bigrams TripAdvisor review text

# Chapter 4

# Background information

In this chapter, we describe briefly basic theoretical knowledge about methods used in our experiments for determining sentiment and important properties of user review. The experiment results are described in chapter 5.

## 4.1 Unsupervised NLP

In this section, we provide short descriptions of several methods based on unsupervised learning which we use for determining important phrases in user reviews.

### 4.1.1 Bag of words

The bag-of-words model is a simplifying representation used in natural language processing and information retrieval. It is a good approach to start as a baseline and sometimes this model can give good results for simple data.

### 4.1.2 *n*-gram

N-gram is a continuous sequence of several items (numbers, digits, words, letters, etc.). There can be a unigram (one word), bigram is a sequence of two words, trigram is a sequence of three words. N-grams can help to find keywords or important phrases and also can be useful as a feature for deep learning models.

### 4.1.3 Word Impotency

Below we describe algorithms that help to identify word or n-gram impotency on an existing corpus.

#### Word frequency

Word frequency is one the most simple method to estimate how the specific word or n-gram is important. It is a simple algorithm that counts the occurrence of the specific n-gram in the text or full corpus.

#### Mutual information

*Mutual information* tells us how much we learn about X knowing the value of Y (on average over the choice of Y), Church and Hanks, 1990

#### Pointwise mutual Information

*Pointwise Mutual Information* (PMI) is a measure of association used in information theory and statistics. PMI was introduced into lexicography by (Church and Hanks, 1990). Confusingly, in the computational linguistics literature, PMI is often referred to as simply MI, whereas in the information-theoretic literature, MI refers to the averaged measure.

#### TF-IDF

*Term Frequency–Inverse Document Frequency* (TF-IDF) is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. It is often used in NLP to extract keywords from the specified document.

### 4.1.4 Word representation in vector space

Inspired by article (RANA, 2018), the author considers several methods of word vector representation:

- One-hot representation
- Distributed Representations
- Singular Value Decomposition (SVD)
- Continuous bag of words model
- Skip-Gram model
- Glove Representations

Using either Skip-Gram models or Continuous bag of words model there can be trained *word2vec* representation. Word2vec is a group of related models that are used to produce word embeddings. The big advantage of word2vec representation is latent semantic analysis compared to the earlier word vector representation algorithm.

## 4.2 Deep Learning for NLP

### 4.2.1 Recurrent neural network

Recurrent Neural Network (RNN) is a type of Neural Network where the output from the previous step are fed as input to the current step. The main and most important feature of RNN is the hidden state, which remembers some information about a sequence. All recurrent neural networks have the form of a chain of repeating modules of the neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single *tanh* layer, see Fig. 4.1

#### Long short-term memory

Long Short Term Memory (LSTM) networks are a special kind of RNN capable of learning long-term dependencies. LSTM has four neural network layer instead of single like RNN has, it is interacting in a very special way as illustrated in Fig. 4.2

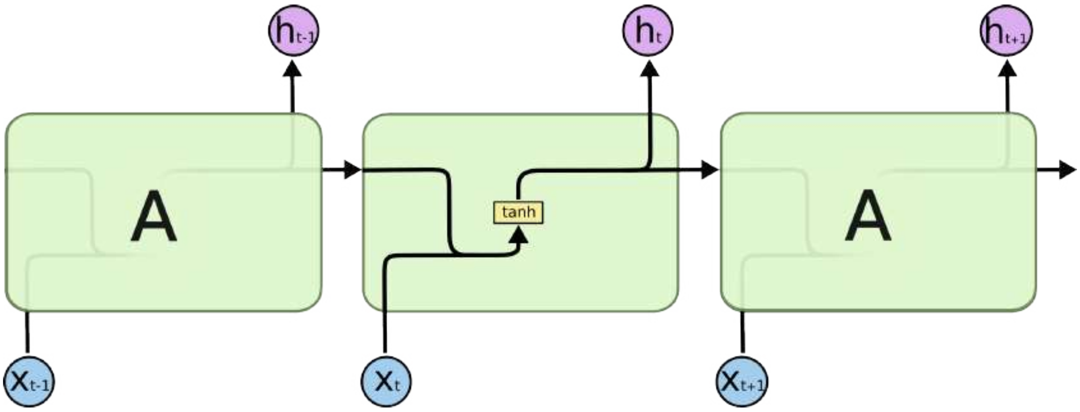



FIGURE 4.1: The repeating module in a standard RNN contains a single layer. Source: Olah, 2015

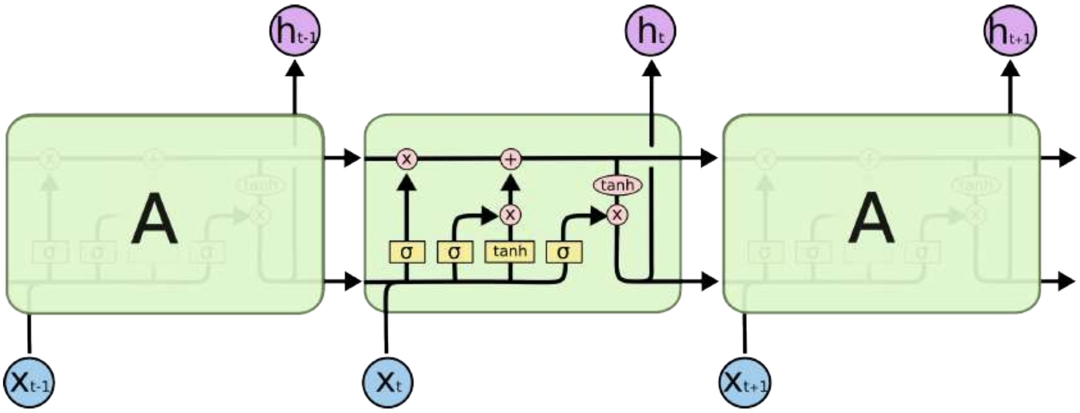



FIGURE 4.2: The repeating module in an LSTM contains four interacting layers. Source: Olah, 2015

### 4.2.2 fastText

fastText is a library for the learning of word embeddings and text classification created by Facebook's AI Research. This model can be used as a simple baseline for sentence classification. The simple illustration of this model is in Fig. 4.3, where the features presented as embeddings because fatsText uses a neural network or word embedding.

## 4.3 Classification

Classification is a machine learning problem of identifying to which a set of categories belongs. Andrew Ng, in his lectures Ng, 2012 provided several examples of classification:

- distinguish whether a tumor is malignant or benign by its size,
- identify spam and non-spam emails by the words

Besides, in our work, we consider the text classification problem (positive or negative).

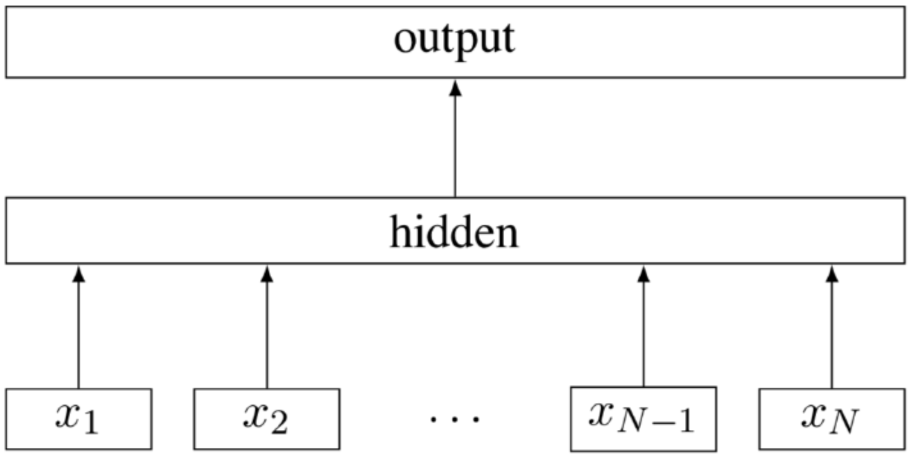



FIGURE 4.3: fastText architecture for a sentence with N n-gram features. Source: Joulin et al., 2016

## 4.4 Clustering

Clustering is a basic of unsupervised machine learning. It is the task of grouping a set of objects in such a way that objects in the same group.

### 4.4.1 K-means

K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). (Trevino, 2016). The main purpose of this algorithm is to find *k* groups in the data concentrated on their centroids.

# Chapter 5

# Experiments

## 5.1 Text classification

Inspired by the project (Gupta, 2019) where the author implemented several models for text classification, we forked it, made some changes to adapt for our task and trained some models using our available dataset, described in chapter 3. As we have a dataset which contains positive and negative user review sentences, we trained several models for review text classification:

- *fastText*. The model was created by Facebook AI Research (FAIR) lab and the algorithm is described in paper of Joulin et al., 2016 where authors explore a simple and efficient baseline for text classification
- *Seq2Seq*. It turns one sequence into another sequence. The implementation was explored by Bahdanau, Cho, and Bengio, 2014 and Du and Huang, 2018
- *TextCNN*. CNN for text classification proposed in New York University by Kim, 2014
- *TextRNN*. Bi-directional LSTM network for text classification.
- *RCNN*. Recurent Convolution Neural Network for text classification proposed by Lai et al., 2015. Here we also use LSTM while implementing this model.

As a result, we trained these proposed model on existing data parsed from Booking.com for text classification into positive or negative. The accuracy of each model is presented in Table 5.1. Besides, we measure the accuracy on data parsed from TripAdvisor, which are not used as training data and it shows us how this model can work on the data from other cites. All the models were run on a 64GB machine with 2 GeForce RTX 2080Ti GPU. Runtime in the table below includes only the time for training the model.

| Model | Accuracy (Booking.com) | Accuracy (TripAdvisor) | Runtime |
|---|---|---|---|
| *fastText* | 89% | 71% | 10 min |
| *Seq2Seq* | 92% | 77% | 17 min |
| *TextCNN* | 86% | 47% | 8 min |
| *TextRNN* | 92% | 58% | 9 min |
| *RCNN* | 90% | 85% | 38 min |

TABLE 5.1: Text classification model performance on existing.

As we can see from the Table 5.2 above, *RCNN* models give us the best accuracy on available dataset.

In a general case, when we don't have separate positive or negative text, we want to analyze which sentence is positive and which is negative of this text. As we have already had dataset for text classification, we tokenized these texts into sentences and got dataset for sentence classification. Using the same approaches as for text classification, we trained the same model and got some results which are presented in Table 5.1.

| Model | Accuracy | $F_1$ − score | Runtime |
|---|---|---|---|
| *fastText* | 82% | 82% | 6 min |
| *Seq2Seq* | 79% | 79% | 14 min |
| *TextCNN* | 82% | 82% | 7 min |
| *TextRNN* | 86% | 86% | 20 min |
| *RCNN* | 86% | 86% | 31 min |

TABLE 5.2: Sentence classification model performance on existing Booking.com dataset)

## 5.2 Analyze important phrases

In this section, we present several results for creating a dictionary of important positive and negative phrases. Here we determine important positive and negative words and phrases based on existing ready Booking.com corpus as it has a strict separation into positive and negative text for each review.

### 5.2.1 TF-IDF

TF-IDF is a natural language processing technique useful for the extraction of important keywords within a set of documents or chapters. For this reason, we fitted TF-IDF vectorizer based on existing positive and negative corpus. As a result we got vocabulary for positive and negative unigram, bigram and trigram and using TF-IDF we can estimate how importnat the specified n-grams are in the existing corpus. As an example, in Fig. 5.1 we would like present how TF-IDF acquire the most important n-grams for negative review.

```
===Review text===
Хороше розташування. Безпечна парковка на території готелю. Наявність холодильника та чайника в номері.
Щоденне прибирання.

===Keywords===
парковка території 0.532
щоденне прибирання 0.506
території готелю 0.487
хороше розташування 0.472
```

FIGURE 5.1: Example of TF-IDF usage for acquiring important phrase

### 5.2.2 Pointwise Mutual Information

The dictionaries created by *pointwise mutual information* (PMI) method presented in Fig. 5.2 for positive and negative phrases respectively.

In contrast to *mutual information* (MI) which builds upon PMI, it refers to single events, whereas MI refers to the average of all possible events. By reading the n-gram tokens with high PMI scores for the target variable, we can get a sense of how

| | uni-gram | score |
|---|---|---|
| 0 | торговий | 0.251974 |
| 1 | стильний | 0.251648 |
| 2 | просторо | 0.250339 |
| 3 | шикарне | 0.249222 |
| 4 | відмінне | 0.249179 |

| | bi-gram | score |
|---|---|---|
| 0 | місцезнаходження супер | 0.255065 |
| 1 | привітна господиня | 0.255065 |
| 2 | єр чистота | 0.255065 |
| 3 | чудове співвідношення | 0.255065 |
| 4 | просторі кімнати | 0.255065 |

| | tri-gram | score |
|---|---|---|
| 0 | персонал ввічливий готовий | 0.255065 |
| 1 | приємний персонал смачні | 0.255065 |
| 2 | приємний персонал хороший | 0.255065 |
| 3 | приємний персонал чистий | 0.255065 |
| 4 | персонал смачні ситні | 0.255065 |

(A) positive

| | uni-gram | score |
|---|---|---|
| 0 | зламаний | 0.347339 |
| 1 | плямами | 0.347069 |
| 2 | слабка | 0.347034 |
| 3 | дверцята | 0.344510 |
| 4 | павутина | 0.344362 |

| | bi-gram | score |
|---|---|---|
| 0 | дивний запах | 0.352442 |
| 1 | жахлива звукоізоляція | 0.352442 |
| 2 | номері погана | 0.352442 |
| 3 | номері погано | 0.352442 |
| 4 | скрипучі ліжко | 0.352442 |

| | tri-gram | score |
|---|---|---|
| 0 | відсутність питної води | 0.352442 |
| 1 | запах ванній кімнаті | 0.352442 |
| 2 | звукоізоляція залишає бажати | 0.352442 |
| 3 | порожній міні бар | 0.352442 |
| 4 | слабкий сигнал wi | 0.352442 |

(B) negative

FIGURE 5.2: Top 5 PMI n-grams

much the reviewer did or didn't like the hotels. So, in further approaches, we will use the dictionaries created by PMI. Analogically to the TF-IDF (subsection 5.2.1), in Fig. 5.3 we present how we can extract important n-grams from positive review text using the PMI method with the specified threshold for the score.

```
===Review text===
Хороше розташування. Безпечна парковка на території готелю. Наявність холодильника та чайника в номері.
Щоденне прибирання.

===Keywords===
[('розташування', 0.23),
 ('хороше', 0.23),
 ('щоденне', 0.21),
 ('наявність', 0.17),
 ('парковка', 0.15)]

[('хороше розташування', 0.25),
 ('парковка території', 0.21),
 ('щоденне прибирання', 0.21)]

[('парковка території готелю', 0.17)]
```

FIGURE 5.3: Example of PMI usage for acquiring important phrases

By the way, similar dictionaries were created using word frequency. In particular, we have already illustrated the result based on word frequency in Fig. 3.5 and 3.6 in the chapter 3.) and mutual information methods. However, word frequency is not a good indicator for sentiment analysis, so we focus on the PMI method.

### 5.2.3 Create dataset

In this section, we describe how we created the dataset which can be used for training a model to determine the most important phrase in the sentence. As we have already had the dictionaries with most important positive and negative n-grams, we can determine which n-grams in the sentence are the most important, in the case when we know whether the sentence is positive or negative.

In the section 3.4 we describe how we got the data for sentence classification. Currently, we have 520710 annotated sentences from positive and negative user reviews. To build the label (n-grams) for each sentence we need to do the following:

- Acquire all potential n-grams from sentence
- As we know whether sentence positive or negative, use an appropriate n-gram dictionary and determine which n-gram has the highest score.
- In case there is no n-gram in dictionary, skip this sentence, otherwise use n-gram with the highest score as a label.

Using this approach for each sentence we got 258190, 164660, and 58830 annotated positive sentences with unigrams, bigrams, trigrams labels respectively. Also, we got 240209, 75870 and 13686 annotated negative sentences with unigrams, bigrams, trigrams labels respectively. The example of created dataset for determining important n-gram are illustrated on the Fig. 5.4

| text | unigram | bigram | trigram |
|---|---|---|---|
| Смачний сніданок, хороший номер, гостинний пер... | хороший | смачний сніданок | смачний сніданок хороший |
| Гарний вид з вікна. | гарний | гарний вид | гарний вид вікна |
| Шикарний вигляд з балкону, в номері нові меблі... | необхідне | гарний дизайн | необхідне комфортного проживання |
| гарне місце розташування | гарне | гарне місце | гарне місце розташування |
| мені сподобалося ставлення персоналу, адмініст... | сподобалося | сподобалося ставлення | сподобалося ставлення персоналу |
| Дуже хороший персонал. гарне місце розташування | гарне | персонал гарне | персонал гарне місце |
| Сподобалося розташування готелю в чистому і от... | тиша | сподобалося розташування | сподобалося розташування готелю |
| Ціна/якість відповідає одне одному | якість | ціна якість | ціна якість відповідає |
| Привітний, усміхнений персонал, чистий номер з... | привітний | персонал чистий | привітний усміхнений персонал |

(A) positive

| text | unigram | bigram | trigram |
|---|---|---|---|
| Перебої з гарячою водою, прохолода в номерах, ... | маленькі | гарячою водою | перебої гарячою водою |
| Дуже тонкі стіни, чути що відбувається в сусід... | пружини | стіни чути | тонкі стіни чути |
| душова кабіна протікає, Повелитель теж | протікає | душова кабіна | душова кабіна протікає |
| Не працював Wi-Fi. | працював | працював wi | працював wi fi |
| Також із невеликих мінусів я б відмітив помилк... | двері | двері ванну | двері ванну кімнату |
| Плюс до всього вікна виходили на жваву магістр... | неможливо | спати відкритим | спати відкритим вікном |
| Номер на першому поверсі змусив слухати голосн... | голосно | вхідні двері | номер першому поверсі |
| Великий просторий номер ліжко і матраци дуже з... | матраци | номер ліжко | великий просторий номер |
| Сильна сутність з сусідніх номерів . огидний с... | сильна | сильна сутність | сутність сусідніх номерів |

(B) negative

FIGURE 5.4: Example of sentence to important n-gram dataset

## 5.3 Clustering n-grams

In case when we have several n-grams, some of them are similar or have the same sense. For instance, we want to summarize shortly the information about this hotel, so we need to cluster gotten n-grams and show the most valuable. For this purpose, we use the K-means algorithm, described in subsection 4.4.1, with flexible number of clusters, after that we select k n-gram which are the closest to their centroid. As an example of clustering n-grams, see Fig. 5.5

bellow we ca

```
===Bi-grams===
['ввічливий персонал', 'чудовий готель', 'привітний персонал', 'дуже чисто', 'чиста кімната', 'чудовий персонал']

===Clustered bi-grams===

['дуже чисто', 'чудовий персонал']
```

FIGURE 5.5: Example of clustering n-grams

## 5.4 Instrument pipeline

In this section we show how the full pipeline works to determine important properties of one or several user reviews. Here we consider the reviews about BLUM HOTEL[1] in Lviv at both resources (Booking.com and TripAdvisor).

### 5.4.1 Analyze one review

In general, to analyze one single review about hotel we need to do the following main steps:

1. Translate the review text into Ukrainian if original is not
2. Tokenize text into sentences
3. Detect via a trained model which sentence is positive or negative (in case of Booking.com, this step can be skipped)
4. Acquire all possible positive and negative n-grams from split positive and negative sentences respectively
5. Select the most important recently acquired n-grams using a built dictionary, described in 5.2.2 and specified threshold score or via the TF-DF method, described in 5.2.1.

#### Example for TripAdvisor

Let's consider one example review from TripAdvisor, see Fig. 5.6.

As a result of the presented review in Fig. 5.6, we acquire the next positive and negative n-grams.

#### Example for Booking

It is easier to analyze review from Booking.com than analyzing TripAdvisor, because there is a strict separation into positive and negative text. In this case we skip running detection to know whether the sentence is positive or negative. We acquire important positive and negative n-grams from positive and negative texts respectively.

In Fig. 5.8 above we can see the example of one review from Booking.com and in Fig 5.9 bellow we have the generated summary about this review. As we can see from the Fig. 5.9, sometimes we cannot extract important negative phrases, it means that n-grams in the negative text on Fig. 5.8 do not have valuable score in built dictionary.

[1] https://blum-hotel-lviv.hotelmix.com.ua

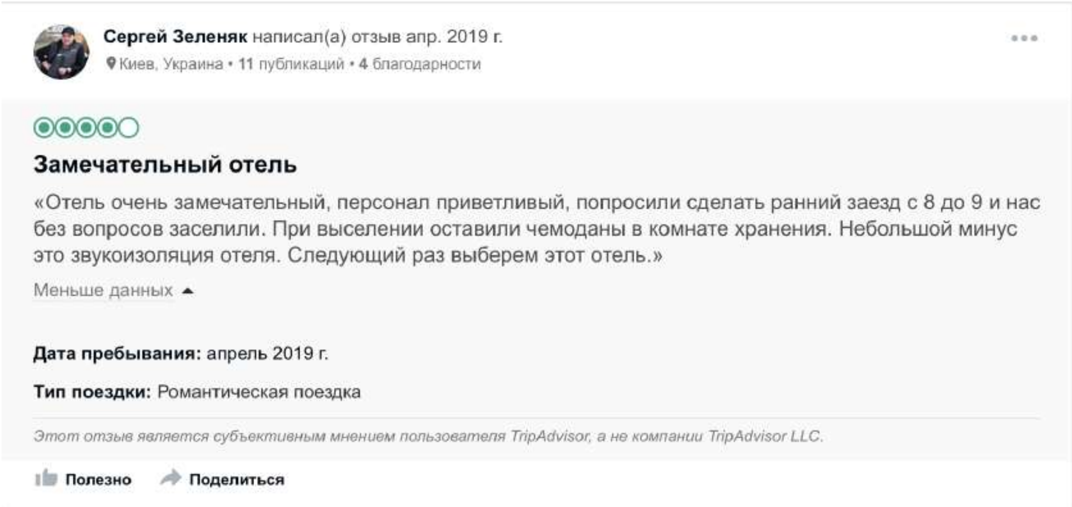



FIGURE 5.6: Example of user review in TripAdvisor

```
====Positive n-gram====
['готель чудовий', 'персонал привітний']
====Negative n-grams====
['звукоізоляція']
```

FIGURE 5.7: Example of summarizing TripAdvisor review provided on Fig. 5.6

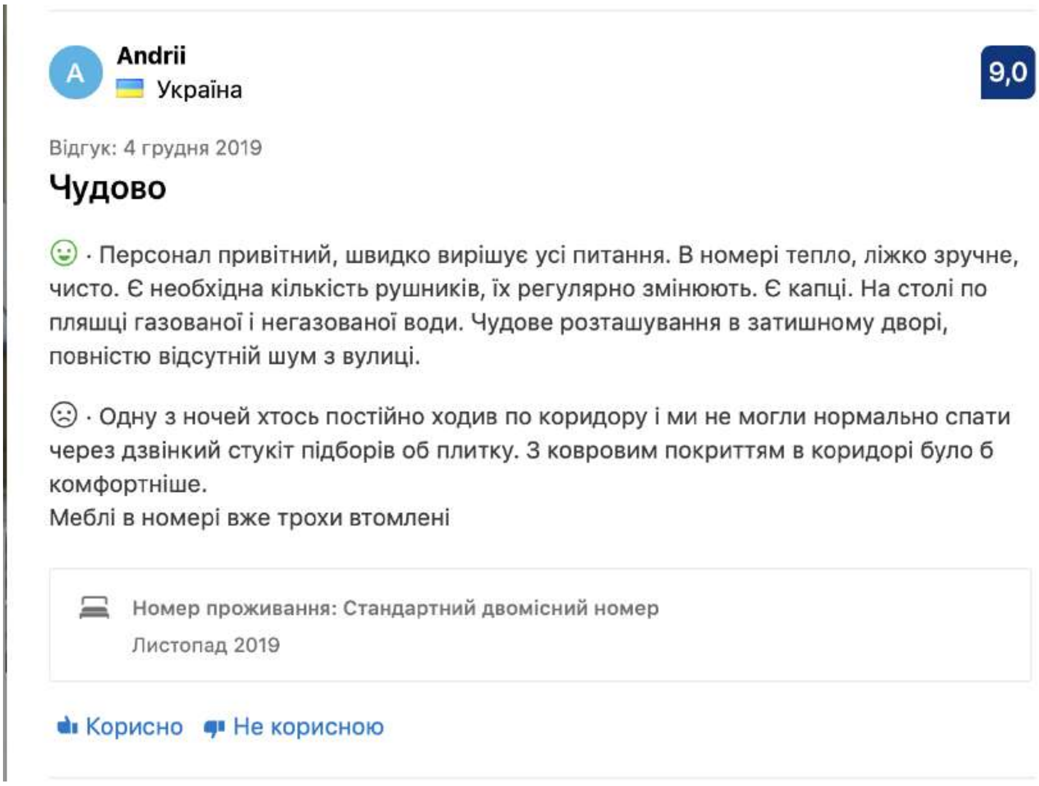



FIGURE 5.8: Example of user review in Booking.com

```
====Positive n-gram====
['персонал привітний', 'ліжко зручне', 'чудове розташування']
====Negative n-grams====
[]
```

FIGURE 5.9: Example of summarizing Booking.com review provided on Fig. 5.8

### 5.4.2 Analyze several reviews

Here we give a summary example about the Blum hotel according to analyzed reviews from Booking.com and TripAdvisor.

#### Example for Booking

In existing Booking.com dataset, we have 379 reviews about BLUM HOTEL. We analyze each review as we described in previous subsection 5.4.1, then we get approximately 2-3 positive and negative bigrams for each review, after that we run K-mean clustering to get top 5 most important bigrams. As a result, we get a summary illustrated in Fig. 5.10 bellow.

```
====Positive n-gram====
['персонал привітний', 'зручне розташування', 'співвідношення ціни', 'тихе місце', 'смачні сніданки']
====Negative n-grams====
['погана шумоізоляція', 'тонкі стіни', 'душова кабіна', 'відсутність ліфта', 'запах номері']
```

FIGURE 5.10: Summary about BLUE HOTEL generated by reviews from Booking.com

#### Example for TripAdvisor

Here, we have significantly fewer reviews about BLUM HOTEL, it has only 19 reviews. However, it gives us a possibility to generate a similar summary presented in Fig. 5.11.

```
=====Positive summary=====

['привітний персонал', 'гарне розташування', 'комфортний номер', 'смачні сніданки']

=====Negative summary=====

['звукоізоляція', 'wi fi']
```

FIGURE 5.11: Summary about BLUE HOTEL generated by reviews from TripAdvisor

As we can see from Fig. 5.11 TripAdvisor analyzer contains less information than we have from Fig. 5.10. Besides, for the summary from TripAdvisor we can notice that there is negative unigram instead of bigrams, it can be explained that there is no valuable negative bigrams based on these reviews. Furthermore, some information from both Booking.com and TripAdvisor are intersected.

# Chapter 6

# Conclusions

## 6.1 Contributions

In this work, we consider the task of important properties of Ukrainian-language user reviews. Quite a lot of work has been done in this domain, but it is the first such kind of investigation for the Ukrainian language. To conclude, we made the following contributions:

1. For the first time for Ukrainian language, a deep-learning ready corpus of reviews about hotels in Ukrainian was created. In total, approximately 164 000 reviews from Booking.com and TripAdvisor were parsed, with 28 000 of them being original Ukrainian-language reviews and 136 000 Russian-language reviews translated into Ukrainian via Google Translate API. This corpus enables deep learning models to be used for solving sentiment analysis problems for this and related domain industries. In this work we considered hotels in popular cities around Ukraine.

2. Dictionaries for positive and negative n-grams (unigrams, bigrams, and trigrams) were cteated with their scores calculated by the PMI method which allows us to define the most influential phrase in the sentence. Furthermore, using these dictionaries, we generated a dataset for extracting the most important n-gram in positive and negative sentences. This dataset can be used for training advanced models, which can determine important phrases in the sentence.

3. For the first time for Ukrainian language, an instrument for determining important properties in Ukrainian-language user reviews has been created. It allows extracting the most positive or negative topics or phrases about hotels. Several models were trained for text and sentence classification into positive and negative, the results are discussed in section 5.1. The best accuracy was achieved by the Recurrent Convolution Neural Network model, inspired by paper Lai et al., 2015 (85% for text classification and 86% for sentence classification). Overall, it allows generating a summary (including positive and negative topics) about a specific hotel based on its reviews. This instrument can be adapted for other domains like reviews about the restaurants, car rentals, etc.

To sum up, all of the above experiments are presented and available for public access at GitHub[1] repository including additional sub-module[2].

[1] https://github.com/DmytroBabenko/Detect-emotion-sentimental
[2] https://github.com/DmytroBabenko/Text-Classification-Models-Pytorch

## 6.2 Future work

We have several ideas for how to improve existing instrument in the future and what could be next directions for this work:

1. Currently, we work only with data from Booking.com and TripAdvisor. There a lot of reviews about the hotel in *google.com* reviews for Ukrainian-language. It is harder to parse such kinds of reviews but it would help to make the existing instrument more flexible and stable.

2. In this work, we focus only on n-grams as important properties of reviews. We do not consider dependency connection in a sentence (e.g specific adjective relatives noun ) and we do not classify part of languages in the review text. It could be useful to acquire the most valuable adjectives which illustrate the tonal impression of users. Also, we could analyze which nouns are the most important and which tonality (positive or negative) they have.

3. Sometimes, some services like Booking.com ask user to leave their rate about specific individual topic like *food*, *location*, *staff* or something like that. Usually, these topics are strictly pre-defined and the user cannot leave the rate about any topic which is not on the list. In this case, the user can write an impression in the text. Using sentiment analysis approaches, it would be helpful to acquire automatically which topic is important for the user (as an example user cannot set rate about *shower* but the proposed model can estimate this rate based on the analysis of several reviews about *shower*).

4. Here, we analyze only reviews of the hotel, in the future direction we would like to parse more reviews about restaurants or other services and transfer existing models on new data. In general, the existing instrument can be adapted to other domains.

# Bibliography

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473*.

Church, Kenneth Ward and Patrick Hanks (1990). "Word association norms, mutual information, and lexicography". In: *Computational linguistics* 16.1, pp. 22–29.

Das, Bijoyan and Sarit Chakraborty (2018). "An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation". In: *arXiv preprint arXiv:1806.06407*.

Du, Changshun and Lei Huang (2018). "Text classification research with attention-based recurrent neural networks". In: *International Journal of Computers Communications & Control* 13.1, pp. 50–61.

Gräbner, Dietmar et al. (2012). "Classification of customer reviews based on sentiment analysis". In: *ENTER*. Citeseer, pp. 460–470.

Gupta, Anubhav (2019). *Text-Classification-Models-Pytorch*. https://github.com/AnubhavGupta3377/Text-Classification-Models-Pytorch.

Hemalatha, I, GP Saradhi Varma, and A Govardhan (2012). "Preprocessing the informal text for efficient sentiment analysis". In: *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)* 1.2, pp. 58–61.

Joulin, Armand et al. (2016). "Bag of tricks for efficient text classification". In: *arXiv preprint arXiv:1607.01759*.

Kasper, Walter and Mihaela Vela (2011). "Sentiment analysis for hotel reviews". In: *Computational linguistics-applications conference*. Vol. 231527, pp. 45–52.

Kaur, J, SS Sehra, and SK Sehra (2017). "A Systematic Literature Review of Sentiment Analysis Techniques". In:

Kim, Yoon (2014). "Convolutional neural networks for sentence classification". In: *arXiv preprint arXiv:1408.5882*.

Lai, Siwei et al. (2015). "Recurrent convolutional neural networks for text classification". In: *Twenty-ninth AAAI conference on artificial intelligence*.

Li, Susan (2019). *A Complete Exploratory Data Analysis and Visualization for Text Data*. https://towardsdatascience.com/a-complete-exploratory-data-analysis-and-visualization-for-text-data-29fb1b96fb6a.

Ng, Andrew (2012). *CS229 Lecture notes - Supervised learning*. URL: http://cs229.stanford.edu/notes/cs229-notes1.pdf.

Olah, Christopher (2015). "Understanding lstm networks". In:

RANA, ASHISH (2018). *Text-Classification-Models-Pytorch*. https://towardsdatascience.com/art-of-vector-representation-of-words-5e85c59fee5.

Romaniuk, A and M Romanyshyn (2013). "Named-entity recognition for sentiment analysis of Ukrainian reviews". In:

Saumya, Sunil et al. (2018). "Ranking online consumer reviews". In: *Electronic Commerce Research and Applications* 29, pp. 78–89.

Shi, Han-Xiao and Xiao-Jun Li (2011). "A sentiment analysis model for hotel reviews based on supervised learning". In: *2011 International Conference on Machine Learning and Cybernetics*. Vol. 3. IEEE, pp. 950–954.

Steffen, Jörg (2004). "N-Gram Language Modeling for Robust Multi-Lingual Document Classification." In: *LREC*.

Trevino, Andrea (2016). *Introduction to K-means Clustering*. URL: https://blogs.oracle.com/datascience/introduction-to-k-means-clustering.

Zhou, Chunting et al. (2015). "A C-LSTM neural network for text classification". In: *arXiv preprint arXiv:1511.08630*.