Extraction of key influential phrases for reviews in Ukrainian based on explainable AI and review rating estimation

“Entropy”

Contents

Abstract

The aim of the study is research of intelligent systems for information retrieval relevant to Ukrainian text data, with a focus on cross-domain reviews. The aim is to build a system that is easily adaptable and extendable, with a specific emphasis on key-phrases extraction from reviews for customer feedback retrieval automation. Our work addresses the challenge of working with a limited number of pretrained models and datasets in the Ukrainian language and modelling discrepant data, aiming to provide a solution that spans different domains of Ukrainian reviews.

Object of the study is an information retrieval process for Ukrainian language and its automation.

Subject of the study is a cross-domain reviews dataset, artificial intelligence models for reviews scores and sentiment predictions and technology for key-phrases retrieval.

General characteristic of the work:

The work is organized in the following way:

- Chapter 1: Introduction. An overview of the research topic, the proposed approach, and key contributions.
- Chapter 2: Related work. A review of prior work related to text classification, algorithm interpretability, and unsupervised Aspect-Based Sentiment Analysis (ABSA) and aspect extraction.
- Chapter 3: Methodology and experiments. We detail our methodology for key-phrase extraction, covering data collection, analysis, model training, technology development, and an evaluation of various explainable AI methods.
- Chapter 4: Conclusion. The chapter summarizes the work and reviews future enhancements of presented approach.

Total number of pages of the work without including references and abstract is 24. Number of references is 46. Number of figures is 6. Number of tables is 3.

Keywords: NLP, deep-learning, LSTM, Attention, LIME, text classification, sentiment analysis.

1 Introduction

Recent advances in NLP sphere, which is primary relevant to neural network based approaches provided researches with a possibility to tackle large variety of difficult tasks (NER[1], NEL[2], QA[3], etc.) and pushed limits for machine text comprehension. Those technologies allow companies to transform unstructured text data to the structured output that is easier to understand and analyze. Text analysis is very much relevant to the B2B companies which are monitoring mass media towards specific businesses for the sake of analytical reports creation and business insights provision. One of the key features that is often included in analytical reports is sentiment analysis w.r.t specific company and predefined time range. Although sentiment analysis provides general insights about company's well-being, it doesn't address the question of causes that influenced such a result. The task relevant to extraction of the causes of sentiment is called Aspect-Based Sentiment Analysis[4]. Despite of the fact that pretrained models for solving the task do exist, most of them are relevant only to one domain. What is more, open-source solutions to ASBA are mostly based on processing of English language and creation of a new labeled dataset requires much amount of time and lots of manual work. The task of ASBA is relevant to classifying sentiment towards identified aspects. If to unite task of ASBA and aspects identification, the overall task can be reformulated in the following manner: retrieve key aspects and classify them with respect to sentiment labels. If to consider that overall sentiment of the sentence is a composite of aspects sentiments, the other reformulation of the task appears: retrieve key aspects that influenced predicted sentiment label the most. Other problem where such a formulation is applicable is relevant to summarization of reviews relevant to specific entity based on extraction of key phrases that influenced explicit ratings. The only difference in formulation for this task is that instead of sentiment label, the retrieval is done towards rating. Generally, the task can be formulated in an abstract way: retrieve key textual features that influenced predicted label the most.

In our work we propose a method for solving the task of extraction of key influential textual features with respect to the overall predicted label. We primarily focus

on processing of Ukrainian language and solving the task in the bounds of cross-domain reviews. For this purpose, a new dataset including reviews for three domains: hotels, restaurants and products, is collected. The data was scrapped from two websites: TripAdvisor and Rozetka. Due to the fact that TripAdvisor doesn't support Ukrainian language in terms of reviews, while part of reviews on Rozetka are in Russian, it was decided to translate scrapped data using Microsoft translator. To get rid of possible anomalies and incorrect translations, a specific data processing was used. In order to remove incorrectly estimated reviews, an automatic machine-learning based approach was utilized. The proposed method to explaining reviews can be described in two stages: training a machine learning model to predict reviews based on textual features and extraction of textual features that are the most influential during decision making of an aforementioned model based on explainable AI techniques. In terms of model, the experiments include both classical machine learning and deep learning-based methods. Due to the class imbalance f1-macro averaged w.r.t introduced domains (hotels, restaurants, products) was used as the main metric. Due to the subjective nature of reviews and their ratings, rating scores were mapped to sentiment labels and additional model was trained. In both rating estimation and sentiment prediction, the architecture of best model was based on LSTM (Long-Short Term Memory)[5] and attention mechanism. To account for noisiness of data, a noise-tolerant training was used, namely specific losses including: Huber[6] and Log-Cosh[7] loss. The experiments relevant to extraction of most influential features included comparison of two methods w.r.t optimal model on two sets of problems (rating estimation and sentiment prediction): LIME (Local-Interpretable Model-Agnostic Explanations)[8] and Dot-product Attention mechanism. The results of both methods were manually validated in terms of both comprehension and suitability to the predicted metric. To numerically evaluate methods, Precision at K metric was used. An algorithmic approach for aggregation of most influential textual features w.r.t entity was introduced, which simplified the usage of overall flow in production setting.

The main contributions of the work, can be summarized as follows:

1. Introduction of a new cross-domain dataset with Ukrainian reviews: the paper presents a novel dataset including three different domains with reviews in Ukrainian language and consisting of 662907 rows. The dataset can be used both for sentiment analysis and reviews estimation.

2. Exhaustive experiments summarizing techniques mandatory for working with noisy textual data: the paper showcases utilization of different techniques relevant to working with noisy data, including automized filtering of mislabeled samples and specific noise-tolerant losses.

3. Trained models that can be used for both sentiment analysis and reviews estimation in Ukrainian: models that achieved the highest accuracy during experiments were open-sourced and can be utilized as solutions for sentiment analysis and reviews estimation tasks or/and used for transfer learning and further research.

4. Introduction of a method based on explainable AI for key phrases retrieval: based on trained models, a method for key phrases retrieval is introduced. The method can be utilized for advanced analysis and summarization of reviews and as a possible solution to unsupervised aspect-based sentiment analysis.

Overall, the contributions of this work have the potential to advance NLP for Ukrainian language, in particular in domain of reviews and sentiment analysis. Nevertheless, current work focuses specifically on Ukrainian language, developed algorithm is language agnostic. For reproducibility and enhancement of future research in the area, all the code starting with data collection and processing and ending with models training and validation is open-sourced on a GitHub[9].

2 Related work

2.1 Classification with textual data

Recent research leverages plenty of methods for solving the tasks of classification based on textual data. The approaches can be divided into two groups based on utilized algorithms: classical machine learning and deep learning ones.

Classical machine learning algorithms require thorough data preprocessing, which often includes words normalization based on lemmatization or stemming; stopwords removal and vectorization of data using TF-IDF[10]. Then processed features are used as an input to a classifier, such as Gradient boosted trees[11], SVM[12] or Logistic Regression[13]. Nevertheless, such approaches are inferior to deep-learning ones in terms of accuracy, they are still utilized due to speed of training and inference and high interpretability. For instance, Utsha et al.[14] apply extreme gradient boosted trees along with TF-IDF to tackle the task of multiclass fake news detection; Das et al.[15] utilize classical machine learning models on the task of sentiment analysis, showing that TF-IDF text vectorization along with NWT (Next word negation) preprocessing step and SVM achieves pretty high accuracies w.r.t three datasets. There is also a tendency of using additional textual features such as POS (Part of speech) tags[16] or NER (Named entity recognition)[17] to boost performance of models. Other approaches suggest usage of word embeddings as a text vectorization method[18], however usage of embeddings make classical machine learning models less interpretable.

Deep-learning based methods achieve state of the art results on many benchmarks relevant to textual data input. Such methods work well especially when big data is available, as they tend to find hidden structures in text and generalize well. Embedding layer is a basis for deep-learning based approaches, as it’s used to map token identifiers to real value vectors. Embeddings allowed researchers to use transfer-learning and leverage knowledge of models trained on big textual corpus for downstream tasks. For instance, Yoon Kim[19] applied convolutional neural network on top of

Word2Vec[20] embeddings for text classification. Each convolutional layer was applied to embeddings in parallel, where number of filters was relevant to n-gram size. Other approaches utilized more sophisticated models which are based on recurrency. LSTM and its variations are widely used for text classification nowadays. Sachan et al.[21] used simple one-layer Bidirectional LSTM along with mixed objective for training to achieve state-of-the-art results on various datasets.

At the same time many researchers tend to combine CNNs with LSTMs to enhance the performance of overall model. Chunting Zhou et al.[22] proposes a C-LSTM, model which applies one dimensional convolution right after embeddings layer to extract high-level representations, which are then fed into LSTM layer, showing superior results w.r.t other methods. CNNs are also used right after LSTM layer, in order to aggregate and process hidden states in a non-linear way instead of just retrieving the last one. For instance, Peng Zhou et al.[23] utilize Bidirectional LSTM with two-dimensional CNN layers, which outperforms C-LSTM on five datasets. Other researchers tend to aggregate hidden states from LSTM layer using attention mechanism. Wang et al.[24] propose model which uses LSTM along with attention mechanism to tackle the problem of aspect-based sentiment analysis. The attention combines both hidden representations of sentence tokens and aspect embeddings to produce the final output vector which is then fed into classification layer. Recent research features approaches, based on transformers and self-attention, which are superior to others in cases when big datasets are available. Nevertheless, performance of such models is pretty stunning, they are way less explainable than those based on LSTMs and CNNs.

In our work we experimented with both classical machine learning algorithms including SVM and gradient-boosting trees and deep learning ones, which are based on LSTMs, CNNs and attention.

2.2 Explainable AI for text classification

Explainable AI is very important field, main goal of which is to interpret predictions made by machine learning models. Explainable AI techniques are often used to

monitor performance of model w.r.t biases and promote end user trust. Explainable AI methods can be classified into three categories: Intrinsically Interpretable Method, and Model Agnostic Methods and Example-Based Explanations. One of the methods to achieve explainable AI is to use intrinsically explainable methods like logistic regression, decision trees and their ensembles. However, such explainability comes with a cost of performance. Attention mechanism can also be considered as an intrinsically explainable method, even though it only partially explains model's results. While logistic regression and decision trees explain model's decision globally, attention mechanism provides a local perspective. Model-agnostic methods separate explanation from a machine learning model, allowing it to be compatible with a variety of models. Model-agnostic method that is often used is surrogate-based explanations. The main idea of it is to train a simpler model on top of original model's predictions and explain the simpler one, which is called a "surrogate". Surrogate-based methods are also divided into global and local ones, as in the example regarding logistic regression and attention mechanism.

One of the famous algorithms that is build on local explainability is LIME (Local Interpretable Model-Agnostic Explanations). LIME trains an inherently interpretable model on the new dataset constructed from the permutation of samples and corresponding predictions of the model. Trained "surrogate" model can be a good approximator of global behavior, it doesn't provide a good approximation for a global one. Shapely is another local explanation method, which is based on game theory. Main idea behind the method is based on an assumption that each feature value is a player in a game and the prediction is an overall payout that is distributed among players. Example-Based explanations are mostly model-agnostic [25] and explain model predictions by selecting instances of the dataset and not by creating summaries of features. There also exist approaches relevant to specifically analyzing neural networks outputs using gradient-based attribution methods [26]. However, Wang et.al [27] showed that gradient-based analysis of NLP models is manipulable, leaving a space for possible adversarial attacks.

Nevertheless, many approaches to explainable AI exist, we focused on analyzing of LIME and attention-based explanations.

2.3 Aspects ranking and unsupervised aspect-based sentiment analysis

Several works similar to ours in terms of task exist. Aspect ranking is a process of identifying important product aspects from online consumer reviews. Yu et. al[28] presented an approach which consisted of three steps: aspect identification, aspect sentiment classification and aspect ranking. Nevertheless, the approach seemed to be effective in comparison with methods of Hu et. al [29] and Wu et. al [30], it includes the estimation of parameters for three models (2 SVMs and parameters for Gaussian distribution), which is hard to adopt to new data and can be slow during inference. Approach was shown to work for English language. In comparison, our approach only needs to train model ones and then apply explainable AI techniques to identify important aspects w.r.t labels model was trained on.

As it was already mentioned, our approach can be thought of as the instance of unsupervised aspect-based sentiment analysis. Garcia-Pablos et.al [31] presented an unsupervised approach to aspect-based sentiment analysis, that utilized graphs and Word2Vec model to identify aspects and detect their polarity. Hercig et.al [32] tackled the problem of unsupervised aspect-based sentiment analysis for Czech language, by breaking the task into 4 separate problems: aspect term extraction, aspect term polarity, aspect category extraction and aspect category polarity. Once again, out approach can be easily adopted for unsupervised aspect-based sentiment analysis, has fewer number of steps and is much easier to use.

So far, the most similar research to ours is the master's thesis of Dmytro Bobenko [33]. In his work, the author tackled the problem of determining sentiment and most influential phrases for each review. The data was collected from TripAdvisor and Booking websites, resulting into the dataset of 164k reviews. The author trained models for sentiment detection and used PMI (pointwise mutual information) to globally create dictionary of negative/positive phrases, which is then used to determine most influential phrases for each classified review. In comparison, the dataset collected in our work is cross-domain and is much bigger (662k reviews); the key phrases extraction works locally which makes it more contextualized and applicable for new data;

similarly to authors we used f1-score as a main metric, however due to imbalance nature of data the “macro” averaging was applied in contrast to “weighted”, which assigns greater contribution to classes with more examples and is not representative of model performance w.r.t all the classes. Other differences are depicted further throughout the article.

3 Methodology and experiments

This section describes methodology used to tackle the problem of key influential phrases extraction including information about data collection, processing and filtering, models training and workflow of an algorithm for key phrases retrieval along with empirical results. As it was already mentioned, our method comprises of two steps: training a discriminative model to predict specific label based on input text and applying local methods for prediction explanation based on input. The second step allows to score phrases and words w.r.t predefined categories of a target variable. Those phrases which have the biggest impact on model's output for predicted label are considered to be most influential.

3.1 Data collection

As it was already mentioned, the collected dataset included three different domains: restaurants, hotels and products. The data was parsed from two websites TripAdvisor and Rozetka. In order to parse big amounts of data without being banned, a number of techniques were used, including: user-agent rotation, proxy server and different time intervals between scrapping. To speed up data collection, multiprocessing was applied. For TripAdvisor the information of only Ukrainian hotels and restaurants was parsed.

In result, the dataset containing 671k reviews was collected. Nevertheless, many complementary information was parsed, we primarily focused on the following columns:

- reviews_text – parsed text of original reviews.
- dataset_name – name of domain dataset.
- entity_name – name of unique hotel, restaurant or product for which review was written.
- rating – rating of review.

A few records from resulting dataset are shown (see Fig. 1).

	review	dataset_name	entity_name	rating
59204	Мягкий, натуральный. На заявленные размеры мат...	rozetka	andersen_cotton_plus_cmp204	5.0
103374	Сподобався. Поки ще не пройшов випробування мо...	rozetka	electro_tf320s	5.0
49760	Была в этом месте первый раз, впечатления хоро...	tripadvisor_restaurants_ukraine	Restaurant_Review-g681193-d12182382-Reviews-La...	5.0
39347	Качественная и очень плотная.Но с нее тяжелее ...	rozetka	freken_bok_14801080	4.0
261413	Наушники супер. Звук очень достойный, басы и в...	rozetka	35734	5.0
149506	L'endroit est tr?s bien situ? , tr?s sympa , l...	tripadvisor_restaurants_ukraine	Restaurant_Review-g294474-d10717273-Reviews-Ko...	2.0
77123	Огромный плюс расположение. Выходим на улицу, ...	tripadvisor_hotels_ukraine	Apartment Club ZimaSnow Ski & Spa	5.0
77560	Дуже класний набір.\nЧашка бомба, чаї дуже сма...	rozetka	lovare_4820198877231	5.0
167806	Отличные носовые платки!Из доступных наверное ...	rozetka	zewa_7322540352313	5.0
238380	Понравилось качество товара. Добротно, без нар...	rozetka	255970641	5.0
88662	Кислятина, при этом абсолютно неострый. Даже и...	rozetka	tabasco_11210009493	2.0
36040	Все добре.\n	rozetka	48229646	5.0
38342	Із заявленого комплектування не було подушки-р...	rozetka	evo_kids_evo_18_bl	2.0
89807	Непоганий протеїн, шоклоладний досить добрий н...	rozetka	ab_pro_pro2000abva79	4.0
86247	Купил его летом 2017 года, ровно год пользовал...	rozetka	9949118	2.0
8650	Патрик Паб - хороший паб в жилом доме недалеко...	tripadvisor_restaurants_ukraine	Restaurant_Review-g294474-d10209611-Reviews-Pa...	4.0
119898	I was dreaming for macaroons and eclers, we ca...	tripadvisor_restaurants_ukraine	Restaurant_Review-g295377-d11827697-Reviews-Do...	5.0
145818	We spent few days here in Kyiv and one place w...	tripadvisor_restaurants_ukraine	Restaurant_Review-g294474-d10593831-Reviews-La...	5.0
18707	Дуже сподобався напій,м'який з насиченим смако...	rozetka	jacobs_4820187049359	5.0
71476	Мышкой пользовался больше года, год активного ...	rozetka	hator_htm_310	3.0

Fig. 1. 20 random samples drawing from originally collected data

3.2 Data preprocessing and analysis

Analyzing the collected dataset, it was found that similarly to the work of Bobenko, parsed textual data was multi-lingual, including, Russian, Ukrainian and other languages (19% to Ukrainian and 81% of reviews relevant to other languages). What is more, TripAdvisor don't support Ukrainian language at all, thus all the reviews relevant to hotels and restaurants domains were in other languages. To tackle this problem, we automated the translation process utilizing Microsoft translation API [34]. As full automation could still result in errors and incorrect translation, reviews were automatically filtered. Analyzing the distribution of characters number in the translated reviews, it was found that some of them had only 1 character and thus were filtered out (see Fig. 2).

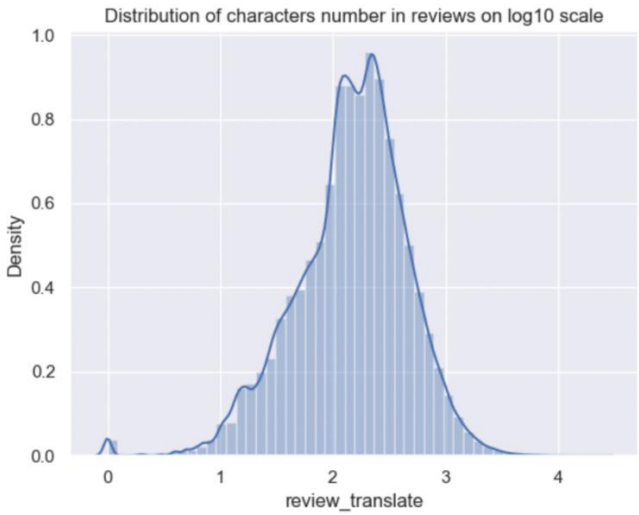

Fig. 2. Distribution of characters number in reviews on log10 scale

Logically if the difference between number of characters in original review and its translation is too big, translated review could be incorrect or incomplete. Those reviews for which the difference was bigger than 200 characters were filtered out. As the possibility of partial translation on the level of sentences existed, translated reviews were tokenized into sentences and for each sentence the language was detected using fasttext model[35]. Based on this information, partially translated reviews were filtered out. Each sentence was tokenized into words using special tokenizer for Ukrainian language that tolerated both apostrophe and hyphen characters. In order to reduce vocabulary and normalize tokens, a specific preprocessing that separated letters from symbols was used. As some of the reviews could be questions about hotels, restaurants or products specific heuristic to determine questions based on POS (part of speech) tags was applied. POS tags were detected using pymorphy2 library. Found questions were filtered out from the dataset. Other preprocessing included deletion of multi-spaces, removal of a newline character, lowercasing and lemmatization that was only used for classical machine learning methods.

Applied preprocessing resulted in a reduced dataset consisting of 662907 reviews. Dataset included 364935 unique words and 205161 unique lemmas. Entity name is an essential categorical feature which is used further for final algorithm of key phrases retrieval. There are more than 28k of unique entities with the median number

of reviews equal to 7. The data can be logically split into subsets w.r.t domains (dataset_name column) and whether the text was translated or not (translated column). In terms of distribution w.r.t domains, 60% of data is relevant to products, 28% to restaurants and 12% to hotels reviews. Analyzing the distribution of ratings, it's clear that it's far from even (see Fig. 3).

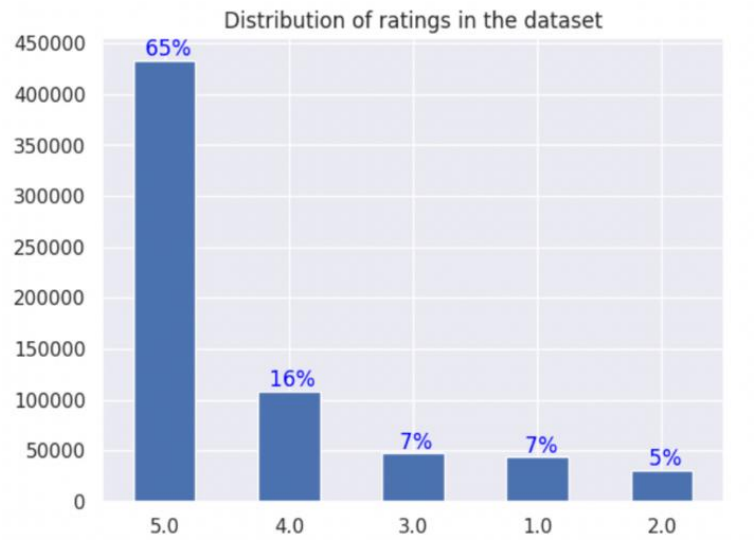

Fig. 3. Distribution of ratings across all domains

As the result of analysis, the following conclusions were made:

1. The fact that number of unique words is pretty huge implies filtering of stopwords for classical machine learning algorithms and usage of specific tokenizers for deep-learning based methods to reduce number of tokens.

2. The fact that distribution of ratings is imbalanced, implies usage of specific techniques to stabilize training procedure and correctly evaluate model performance.

3. The fact that distribution of domains across the dataset is not even and major part of reviews are translated can cause model to overfit to one domain. Thus, it was decided to conduct evaluation of algorithms w.r.t each domain and translated identifier category.

3.3 Data filtering

In our work the experiments were done w.r.t both classical machine learning models, in particular logistic regression and gradient-boosted trees and deep-learning based ones, which utilized convolution, recurrent and attention layers. While training the models, we faced the issue of noisy data that came out from the subjectivity of user's ratings and discrepancy between the actual text of review and its rating. Such a problem typically arises while working with human generated data. Thus, in order to filter out misleading data samples, an automotive approach was used. Models with different architectures were picked and trained on dataset in a cross-validation manner, so that each model could generate predictions for each K fold, while being trained on K-1 folds. For filtration a stratified k-fold strategy was used with K equal to 5. After generating predictions for each sample using M different models, those samples for which all the models made incorrect prediction were analyzed. The logic behind the filtering was in the fact that different models would learn distribution of data w.r.t target differently, but would make same mistakes for outliers. It was empirically discovered that majority of analyzed samples were mislabeled and had discrepancy between review text and rating (see Fig. 4).

Review (Ukrainian)	Review (English)	Rating	Expected rating
Дуже рекомендую !	Highly recommend !	1	4-5
Відмінна ціна))) На акцію ! ! !	Excellent price))) On sale ! ! !	3	4-5
Якщо ви потрапите в це місце , будь те готові до того , що через 15 хвили н після замовлення вам подадуть апе ритив або коктейль , і все буде смач но . Акцій і подарунків дуже багато , тому якщо ви не поспішаете , ви тут !	If you get to this place, be prepared for the fact that 15 minutes after ordering you will be served an aperitif or a cocktail, and everything will be delicious. There are a lot of promotions and gifts, so if you are not in a hurry, you are here!	3	4-5
Місце дійсно цікаве , сподобалося вс е , в тому числі і інтер'єр . Офіціанти доброзичливі і дуже швидко обслуго вуються ! Я рекомендую кубик кокт еля libre . Є безкоштовний Wi - Fi . Музика круто грає ! Також полюбил и їжу і коктейльну карту !	The place is really interesting, I liked everything, including the interior. The waiters are friendly and serve very quickly! I recommend the libre cocktail cube. There is free Wi-Fi. The music plays cool! We also liked the food and cocktail menu!	3	4-5
Відсутні DVI - D (цифрові) адапте ри VGA (аналогові) . Існують тіль ки DVI - I (цифровий аналоговий) до VGA (аналоговий) . БУДЬ ОБЕР ЕЖНИЙ !	DVI-D (digital) VGA (analog) adapters are missing. There are only DVI-I (digital analog) to VGA (analog). BE CAREFUL !	5	1-2

Fig. 4. Example of confusing samples with discrepancy between reviews and rating

It’s important to note that subjectivity of ratings naturally exists in terms of ratings that are close to each other (1 star is pretty similar to 2 stars, whereas same is true for 5 and 4 ones). Thus, only those samples for which the difference between actual rating and predicted was bigger than two were filtered. As the result of filtering, 7437 samples were removed from dataset.

3.4 Modelling

The training procedure can be divided into two categories: classical machine learning algorithms and deep-learning ones. As it was already mentioned, ratings are pretty subjective, thus it was decided to conduct experiments both on the problem of rating estimation and on sentiment prediction one. To convert task from rating estimation to sentiment prediction, rating labels were mapped to sentiment ones using the following rule: ratings equal to 2 and lower mapped to negative, rating of 3 to neutral, and ratings higher than 3 to positive. For sentiment prediction, the experiments were conducted towards 5 deep-learning architectures that achieved best results on ratings estimation and two classical machine learning algorithms. The data was split in a stratified manner w.r.t each domain dataset and ratings. Throughout experiments, f1 score with macro averaging was used as the main metric. To choose between algorithms, averaged f1 macro w.r.t three domains was used.

Firstly, classical machine learning algorithms were trained. Experiments were done towards logistic regression and gradient-boosted trees implementation of xgboost library[36] Stop words were removed from lemmatized tokens, which were then transformed into vectors using tf-idf (term frequency – inverse document frequency) and used as input to models. As the runtime of classical machine learning algorithms is often lower than of deep-learning ones due to fewer number of parameters, a Bayesian search [37] over the hyper-parameters was performed.

As of deep-learning algorithms, the experiments were conducted w.r.t combination of different layers and mechanisms including attention, convolution and recurrency. Considering the fact that real-world text has many typos and number of words

in vocabulary is huge, it was decided to use sub-word tokenization method named BPE (byte-pair-coding)[38]. BPE tokenizer was trained with a min frequency of words equal to 5, which resulted into more than 10 times decrease in a number of tokens (30k). For all the experiments, embeddings with 300 dimensions were used. Due to analysis of median number of tokens in a review, all the sequences of tokens were padded to the length of 300. All the models were trained for 20 epochs and early stopping strategy with a tolerance equal to 5 epochs of training was utilized. As the main technique for regularization the Dropout[39] was applied. Adam optimizer [40] with default parameters was used for model's training. Some of the models utilized embeddings from Word2Vec model, which was pretrained on the BPE tokenized dataset. Throughout the experiments, the same random seed was used to ensure reproducibility. All the architectures were implemented using Tensorflow[41] and Keras[42] frameworks. The following architectures were implemented and tried out:

- Kim-CNN. The architecture proposed by Yoon Kim, which applies parallel convolutional layers to embedding layer and concatenates their output before the classification one. Kernel size is relevant to number of n-gram range that are convolved. In our experiments, we used kernel size range from 3 to 5, pooling window equal to 2 and filters equal to 32.

- Kim-CNN with spatial dropout and more layers. In this experiment, the previous architecture was modified to include more convolutional layers. In particular, spatial dropout [43] was applied after the embbeding layer; the kernel size range was extended to the following values: 3,4,5,7,9; after each convolutional layer along with max pooling, the dropout was used.

- LSTM-CNN. Right after the embeddings, LSTM (Long-short-term-memory) layer was utilized. Processed sequences from LSTM were then convolved. This combination would allow to nonlinearly aggregate processed information from the LSTM. Spatial dropout is used through all the next experiments, including this one.

- CNN-LSTM. Right after the embeddings, convolution is applied similarly to Kim-CNN architecture. Number of convolutional filters was increased to 100. After convolution layer, the LSTM one is utilized.

- LSTM-Attention. Attention is applied after the LSTM to aggregate processed representations of words. A dot product attention was used with tanh nonlinearity. Before the classification layer attention output was concatenated with the last state of the LSTM. As it was already mentioned, attention can be used to locally explain model's decision to some degree by analyzing importance weights assigned to each processed word from LSTM. In our experiments, both attention weights matrix and the LSTM one had the same number of shape equal to 128.

- Bi-LSTM. Instead of applying the LSTM after embeddings, a bidirectional version of it is utilized. It allows to access text both from right to left and left to right allowing for richer representation of text.

- Bi-LSTM CNN2D. Architecture proposed by Zhang et. al, which based on utilization of bidirectional LSTM and processing its outputs using two dimensional CNN. In our experiments, we used CNN with 100 filters and kernel size of 3 and for bi-LSTM number of units was set to 300.

- Deep LSTM. Instead of applying one LSTM after embeddings, two LSTMs were stacked. Between LSTMs dropout was utilized along with layer normalization[44]. For first LSTM number of neurons was increased to 128.

- Deep Bi-LSTM. Same logic as for deep LSTM, but with substitution of first level LSTM layer by a bidirectional one. Instead of layer normalization, batch normalization [45] was used.

- Deep LSTM Attention. Similar to the deep LSTM, but with usage of attention for aggregation of all output states of the second level LSTM.

- Deep LSTM Attention with Word2Vec embbedings. Same architecture as before, but instead of training embeddings from scratch, the pretrained ones were finetuned.

- CNN Deep LSTM Attention with Word2Vec embbedings. A forge of two architectures, in particular Kim-CNN with more layers and Deep LSTM attention. Firstly, parallel convolutions for defined kernel sizes were applied, the concatenated result was then passed to LSTM layers and attention. Word2Vec embbedings were utilized as in previous architecture.

- Deep LSTM Attention with Word2Vec embeddings and class weights. Same as deep LSTM attention with Word2Vec embbedings, but class weights were applied to tackle the problem of class imbalance. Class weights were simply computed by scikit-learn library.

- Variations of Deep LSTM Attention with Word2Vec embeddings w.r.t noise-tolerant objectives. Even after automatic data filtration process, biased samples still persisted in the data. Thus, it was decided to try out noise-tolerant training, specifically techniques relevant to altering the objective of a model. First experiment was related to technique named label smoothing [46], the logic of which lies in the fact that for high accuracy of the model, pushing the probabilities for right classes towards 1 (that's what cross-entropy does under the hood) is not always needed. If data is noisy, maximizing the likelihood of labels given the data can be harmful. Label smoothing regularizes the model by converting hard labels into the soft ones, which helps to deal with overconfident predictions and improve generalization. In our experiments we applied label smoothing with label smoothing factor equal to 0.1. While label smoothing alters targets for cross-entropy objective, there are approaches which utilize noise-robust objectives such as log cosh and Huber loss. Log cosh loss is less sensitive to outliers and is simply computed as applying cosh and logarithm to difference between predicted and real vector. Log cosh loss can be viewed as a smoothed out L1 using L2 around origin. Huber loss combines L1 and L2 losses by explicitly using L2 in the vicinity of the origin where the discontinuity lies, and then switching to L1 a certain distance, delta, away from the origin. Both losses are primarily used for robust regression, but can also be adopted to classification problems, by simply computing the difference between predictions probabilities vector and one-hot vector of target classes. In our experiments, we used Huber loss with a delta of 1.

The results of modeling on ratings prediction problem are presented in Table 1, whereas on problem of sentiment analysis – in Table 2.

Table 1. Results on problem of rating estimation

Approach	Test f1 Rozetka	Test f1 TripAdvisor hotels	Test f1 TripAdvisor restaurants	Test f1 translated data	Test f1 original data	Averaged f1 on all domains
logistic_regression	0.378	0.339	0.367	-	-	0.361
gradient boosted trees	0.26	0.256	0.262	-	-	0.259
lstm_attention	0.474	0.555	0.563	0.530	0.483	0.531
lstm_cnn	0.482	0.550	0.546	0.526	0.479	0.526
bilstm_cnn2d	0.497	0.556	0.549	0.534	0.496	0.534
bilstm	0.483	0.532	0.54	0.521	0.480	0.518
cnn_deep_lstm_attention_w2v	0.504	0.549	0.546	0.533	0.514	0.533
cnn_lstm	0.51	0.528	0.541	0.528	0.518	0.526
deep_bilstm	0.492	0.536	0.548	0.527	0.502	0.525
deep_lstm	0.491	0.548	0.554	0.532	0.494	0.531
deep_lstm_attention	0.498	0.553	0.557	0.538	0.496	0.536
deep_lstm_attention_w2v	0.516	0.568	0.572	0.552	0.523	0.5521
deep_lstm_attention_w2v_class_weights	0.493	0.562	0.584	0.546	0.497	0.546
deep_lstm_attention_w2v_huber	0.511	0.574	0.572	0.553	0.511	0.5526
deep_lstm_attention_w2v_label_smoothing	0.498	0.566	0.564	0.543	0.501	0.543
deep_lstm_attention_w2v_log_cosh	0.5	0.57	0.571	0.547	0.505	0.547
kim_cnn	0.517	0.510	0.534	0.528	0.516	0.520
kim_cnn_more_layers_spatial_drop	0.513	0.532	0.546	0.535	0.514	0.530

Table 2. Results on problem of sentiments analysis

Approach	Test f1 Rozetka	Test f1 TripAdvisor hotels	Test f1 TripAdvisor restaurants	Test f1 translated data	Test f1 original data	Averaged f1 on all domains
logistic_regression	0.562	0.497	0.546	-	-	0.535
gradient boosted trees	0.422	0.39	0.428	-	-	0.413
bilstm_cnn2d	0.685	0.699	0.732	0.709	0.689	0.705
deep_lstm_attention_w2v	0.691	0.712	0.728	0.712	0.698	0.71
deep_lstm_attention_w2v_class_weights	0.676	0.7	0.738	0.709	0.673	0.705
deep_lstm_attention_w2v_huber	0.691	0.721	0.745	0.721	0.695	0.719
kim_cnn_more_layers_spatial_drop	0.657	0.709	0.734	0.705	0.650	0.7

As it can be seen from results depicted in Table 1, deep_lstm_attention-w2v_huber achieves best results in terms of test f1 for TripAdvisor hotels domain and averaged f1 on all domains. Analyzing the confusion matrix (see Fig. 5) of best approach on

rating estimation, it's easy to notice that most of the errors are relevant to mismatching close categories, that are subjective by nature. This in particular, implies that trained model is representable of data distribution and can be used for further experiments relevant to key phrases retrieval. Interestingly, the effect of noise-robust objective isn't very noticeable in rating estimation experiment. In fact, the difference between average f1 on all domains between Deep LSTM Attention Word2Vec embeddings with cross-entropy and with Huber loss is only 0.0005 points, whereas the gap is much bigger for the task of sentiment analysis (+0.09). It's worth mentioning that results of models could be improved by using automatic hyper-parameters optimization and manual data filtering. The exact configurations of models in terms of their architectures and hyper-parameters are available on GitHub.

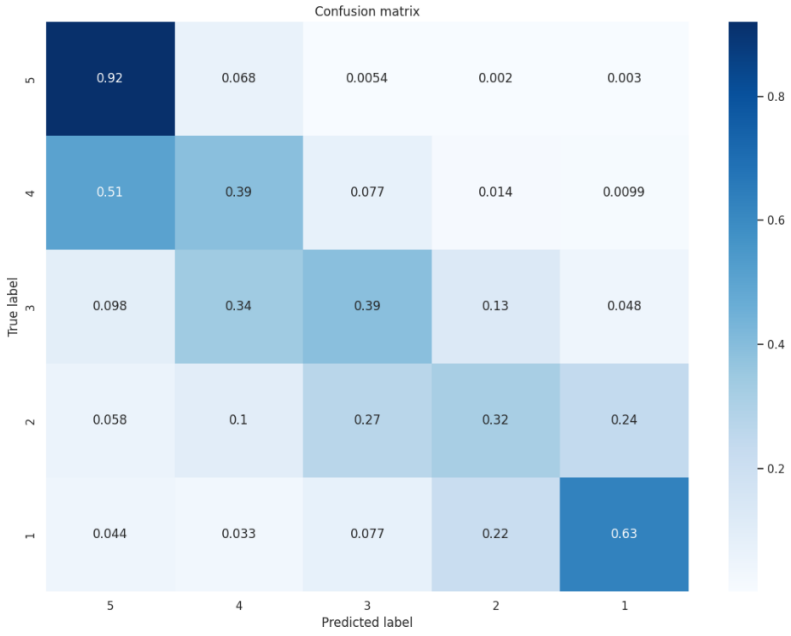

Fig. 5. Confusion matrix produced by best approach for rating estimation

3.5 Algorithm for key phrases retrieval

After training the models, the best one w.r.t chosen metric was picked for explainability experiments and construction of an algorithm for key phrases retrieval. The algorithm works on both on the level of entity (restaurant/hotel/product) and on the level of its review. While working on the level of entity, specific averaging is used to

summarize most influential phrases across all the reviews for the entity. The algorithm for key-phrases retrieval can be logically divided into two steps: retrieval of predictions and scores for each token in each review and aggregation of scores across all the predictions.

Retrieval of scores is the main subject of our experiments. In particular, the experiments were conducted towards two methods: LIME and Attention. As the trained model operated on BPE tokens, which are essentially sub-words, the operation of sub-words merging was implemented. For merged sub-words, corresponding attention scores were summed-up. Main disadvantage of straightforward attention explanation is that its feature scoring gives explanation that is interpreted towards the class with highest probability, although certain features can contribute to increasing of probabilities of other classes. On the other hand, LIME provides explanation that captures contribution of features towards each class. Speaking of LIME, the main disadvantage that we found was disability of using custom tokenizer, which is essential for Ukrainian language. Also, while Attention is an in-built mechanism of model explainability, LIME uses local surrogate models to interpret predictions, which could be not strong enough to understand the data and approximate predictions of much more complex model.

Having obtained the scores for each token and actual predictions for each review, the aggregation of results was done. The aggregation step works for phrases of varying size, supports aggregations relevant to sum and mean and has a functionality for diversification of results based on input tokens. For n-grams other than unigrams, the scores are summed up or averaged, depending on aggregation algorithm's settings. It's worth mentioning that aggregation algorithm is agnostic towards the method used for scoring tokens and is pretty simple in nature, which makes it easier to extend and enhance.

The full pipeline of extraction for key phrases extraction and reviews summarization works in the following way:

1. Process the data in the same way, that was used to process the data for training the models (add spaces between punctuations, remove next-line character, etc.).
2. Tokenize the data using trained BPE tokenizer.

3. Make predictions and explanations based on trained model using LIME or Attention for each review/text.
4. Summarize results using aggregation algorithm.

The experiments to decide which method is more suitable for key-phrases retrieval were conducted. During experiments the aggregation method's parameters were the following: n-gram was set to be equal to 4, both aggregation for phrase scores and overall scores were set to mean, diversification was used with overlap of 2 words, for LIME method top-15 phrases per label were retrieved, whereas for Attention – top-20. The experiments were made towards best model trained for rating estimation. For experiments, a new dataset of diverse entities in terms of their average rating across three domains was constructed. To compare results of LIME and Attention explanation, Precision at K metric was used. The phrase was considered relevant if it was comprehensive and reflective of predicted category (see Fig. 6).

Fig. 6. Example of LIME and Attention explanation for one of the entities. In green – positive phrases are shown, in red – negative. For ratings <3 only negative phrases are relevant, for >4 – only positive, for 3 – both negative and positive. Results were validated towards summarization of all reviews w.r.t specific entity and categorized by averaged rating groups (<3, 3 and >=4).

Results were validated towards summarization of all reviews w.r.t specific entity and categorized by averaged rating groups (<3, 3 and >=4).

As it can be seen from Table 3, Attention method achieves better Precision at K averaged on all rating groups, which was used as main metric. It's worth mentioning that LIME has better coverage in terms of number of phrases, thus it's recommended to use the combination of methods while retrieving key phrases. The algorithm for phrases retrieval can easily be enhanced based on POS tags, which could help to obtain only those phrases which suit specific patterns (e.g. Noun-Adjective, Adjective-Noun-Verb, etc.).

Table 3. Results on problem of key-phrases retrieval

Approach	Precision at K for average rating <3	Precision at K for average rating 3	Precision at K for average rating >=4	Average Precision at K
LIME	0.2806	0.3292	0.221	0.276
Attention	0.308	0.3009	0.266	0.291

4 Conclusion

A novel method for key-phrases retrieval, based on training discriminative model and applying explainable AI on top of it was presented. The new dataset which can be used for further research of key-phrases retrieval and pretraining of models in Ukrainian, was collected. In order to tackle the noisiness, the chain of methods was described, showing that substitution of cross-entropy loss with Huber one, improves f1 score.

Trained models can be utilized solely to tackle the problem of sentiment analysis and rating estimation in 3 domains. There is also a room for using trained models for transfer learning, therefore helping to tackle other problems in Ukrainian NLP. Although, the comparative study has shown that Attention-based phrases retrieval is better than LIME ones, in practice it's recommend to experiment with both or even combine them. The proposed method for key-phrases retrieval is simple, easy to extend and enhance. Nevertheless, there are still many things to improve, including enhancement of model's quality, application of other methods for explainable AI (including gradient based explanations), POS-tags based key-phrases filtration and others. In the future work, we plan to adopt our method to unsupervised aspect-based sentiment analysis and compare it to other methods in the field. Even though the experiments with key-phrases retrieval algorithm were conducted in Ukrainian language, it can easily be adopted to any other. All the models, data and code are open-sourced for future research.

5 Acknowledgements

We express our heartfelt gratitude to LOOQME company for generously furnishing us with the essential computational resources vital to the execution of our research endeavors.

6 References

1. Vikas Yadav and Steven Bethard. A Survey on Recent Advances in Named Entity Recognition from Deep Learning models. In Proceedings of the 27th International Conference on Computational Linguistics, pages 2145–2158, Santa Fe, New Mexico, USA. Association for Computational Linguistics. (2018)
2. Sevgili, Özge & Shelmanov, Artem & Arkhipov, Mikhail & Panchenko, Alexander & Bie-mann, Chris. Neural entity linking: A survey of models based on deep learning. Semantic Web. 13. 1-44. 10.3233/SW-222986. (2022).
3. Qin Zhang, Shangsi Chen, Dongkuan Xu, Qingqing Cao, Xiaojun Chen, Trevor Cohn, and Meng Fang. A Survey for Efficient Open Domain Question Answering. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14447–14465, Toronto, Canada. Association for Computational Linguistics. (2023).
4. Gianni Brauwers and Flavius Frasincar. A Survey on Aspect-Based Sentiment Classification. ACM Comput. Surv. 55, 4, Article 65, 37 pages. https://doi.org/10.1145/3503044 (2023).
5. Hochreiter, S., & Schmidhuber, J"urgen. Long short-term memory. Neural Computation, 9(8), 1735–1780 (1997).
6. Kaan Gokcesu, Hakan Gokcesu. Generalized Huber Loss for Robust Learning and its Efficient Minimization for a Robust Statistics. URL: https://arxiv.org/pdf/2108.12627.pdf
7. Resve A. Saleh, A.K.Md. Ehsanes Saleh. Statistical Properties of the log-cosh Loss Function Used in Machine Learning. URL: https://arxiv.org/pdf/2208.04564.pdf
8. Marco Tulio Ribeiro and Sameer Singh and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, (2016).
9. Volodymyr Kovenko. GitHub of the project. URL: https://github.com/HikkaV/Ukrainian-Reviews-Estimation.

10. TF–IDF. In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8_832 (2011).
11. Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of Statistics, 1189–1232.
12. Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297.
13. Cox, D. R. (1958). The regression analysis of binary sequences. Journal of the Royal Statistical Society: Series B (Methodological), 20(2), 215–232.
14. Tian, Zhenya & Xiao, Jialiang & Feng, Haonan & Wei, Yutian. (2020). Credit Risk Assessment based on Gradient Boosting Decision Tree. Procedia Computer Science. 174. 150-160. 10.1016/j.procs.2020.06.070.
15. Bijoyan Das, Sarit Chakraborty. An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation. URL: https://arxiv.org/abs/1806.06407
16. K. S. Kalaivani, R. Felicia Grace, M. Aarthi, M. Boobeash. Classification of Sentiment Reviews using POS based Machine Learning Approach. URL: https://www.ijert.org/research/classification-of-sentiment-reviews-using-pos-based-machine-learning-approach-IJERTCONV6IS04061.pdf.
17. Lithgow-Serrano O, Cornelius J, Kanjirangat V, Méndez-Cruz CF, Rinaldi F. Improving classification of low-resource COVID-19 literature by using Named Entity Recognition. Genomics Inform. 2021 Sep;19(3):e22. doi: 10.5808/gi.21018. Epub 2021 Sep 30. PMID: 34638169; PMCID: PMC8510872.
18. Subbaraju Pericherla and E Ilavarasan 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1085 012008.
19. Yoon Kim. Convolutional Neural Networks for Sentence Classification. URL: https://arxiv.org/abs/1408.5882.
20. Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space URL: https://arxiv.org/abs/1301.3781.
21. Devendra Singh Sachan, Manzil Zaheer, Ruslan Salakhutdinov. Revisiting LSTM Networks for Semi-Supervised Text Classification via Mixed Objective Function. URL: https://arxiv.org/abs/2009.04007.

22. Chunting Zhou, Chonglin Sun, Zhiyuan Liu, Francis C.M. Lau. A C-LSTM Neural Network for Text Classification. URL: https://arxiv.org/abs/1511.08630.
23. Peng Zhou , Zhenyu Qi , Suncong Zheng, Jiaming Xu, Hongyun Bao, Bo Xu. Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling. URL: https://arxiv.org/pdf/1611.06639.
24. Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for Aspect-level Sentiment Classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 606–615, Austin, Texas. Association for Computational Linguistics.
25. Sheikh Rabiul Islam, William Eberle, Sheikh Khaled Ghafoor, Mohiuddin Ahmed. Explainable Artificial Intelligence Approaches: A Survey. URL: https://arxiv.org/pdf/2101.09429.
26. Marco Ancona, Enea Ceolini, Cengiz Öztireli, Markus Gross. Towards better understanding of gradient-based attribution methods for Deep Neural Networks. URL: https://arxiv.org/abs/1711.06104.
27. Junlin Wang, Jens Tuyls, Eric Wallace, Sameer Singh. Gradient-based Analysis of NLP Models is Manipulable. URL: https://aclanthology.org/2020.findings-emnlp.24.pdf.
28. Jianxing Yu, Zheng-Jun Zha, Meng Wang, Tat-Seng Chua. Aspect Ranking: Identifying Important Product Aspects from Online Consumer Reviews. URL: https://aclanthology.org/P11-1150.pdf.
29. Minqing Hu and Bing Liu. 2004. Mining opinion features in customer reviews. In Proceedings of the 19th national conference on Artifical intelligence (AAAI'04). AAAI Press, 755–760.
30. Wu, Yuanbin and Zhang, Qi and Huang, Xuanjing and Wu, Lide 2009. Phrase dependency parsing for opinion mining Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3
31. Aitor Garc´ıa-Pablos, Montse Cuadros, German Rigau. V3: Unsupervised Aspect Based Sentiment Analysis for SemEval-2015 Task 12. URL: https://aclanthology.org/S15-2121.pdf.

32. Hercig, Tomás Brychcín, Tomás Svoboda, Lukás Konkol, Michal Steinberger, Josef. Unsupervised Methods to Improve Aspect-Based Sentiment Analysis in Czech. URL: https://www.redalyc.org/articulo.oa?id=61547469006.
33. Babenko, Dmytro. Determining sentiment and important properties of Ukrainian-language user reviews : Master Thesis : manuscript rights / Dmytro Babenko ; Supervisor Vsevolod Dyomkin ; Ukrainian Catholic University, Department of Computer Sciences. – Lviv : 2020. – 35 p.
34. Microsoft Text Translation. URL: https://www.microsoft.com/en-us/translator/business/translator-api/
35. Armand Joulin, Edouard Grave, Piotr Bojanowski, Tomas Mikolov. Bag of Tricks for Efficient Text Classification. URL: https://arxiv.org/abs/1607.01759
36. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939785
37. V. Nguyen, "Bayesian Optimization for Accelerating Hyper-Parameter Tuning," 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), Sardinia, Italy, 2019, pp. 302-305, doi: 10.1109/AIKE.2019.00060.
38. Vilém Zouhar, Clara Meister, Juan Gastaldi, Li Du, Tim Vieira, Mrinmaya Sachan, and Ryan Cotterell. 2023. A Formal Perspective on Byte-Pair Encoding. In Findings of the Association for Computational Linguistics: ACL 2023, pages 598–614, Toronto, Canada. Association for Computational Linguistics.
39. Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15, 1929-1958.
40. Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, .
41. Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo,Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay

Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens,Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

42. Chollet, F., & others. (2015). Keras. GitHub. URL: https://github.com/fchollet/keras
43. Lee, S., & Lee, C. (2020). Revisiting spatial dropout for regularizing convolutional neural networks. Multimedia Tools and Applications, 79(45-46), 34195-34207. https://doi.org/10.1007/s11042-020-09054-7
44. Ba, J. L., Kiros, J. R. & Hinton, G. E. (2016). Layer Normalization (cite arxiv:1607.06450)
45. Ioffe, S. & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.. CoRR, abs/1502.03167.
46. Rafael Müller, Simon Kornblith, Geoffrey Hinton. When Does Label Smoothing Help? URL: https://arxiv.org/abs/1906.02629 Author, F.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010). LNCS Homepage, http://www.springer.com/lncs, last accessed 2016/11/21.