User opinion extraction model concerning consumer properties of products based on a recurrent neural network[1]

Yuri P. Yekhlakov
Professor, Department of Data Processing Automation
Tomsk State University of Control Systems and Radioelectronics
Address: 40, Prospect Lenina, Tomsk, 634050, Russia
E-mail: upe@tusur.ru

Egor I. Gribkov
Doctoral Student, Department of Data Processing Automation
Tomsk State University of Control Systems and Radioelectronics
Address: 40, Prospect Lenina, Tomsk, 634050, Russia
E-mail: drnemor@gmail.com

Abstract

This article offers a long short-term memory (LSTM) based structured prediction model taking into account existing approaches to sequence tagging tasks and allowing for extraction of user opinions from reviews. We propose a model configuration and state transition rules which allow us to use past predictions of the model alongside sentence features. We create a body of annotated user reviews about mobile phones from Amazon for model training and evaluation. The model trained on reviews corpus with recommended hyperparameter values. Experiment shows that the proposed model has a 4.51% increase in the F1 score for aspects detection and a 5.44% increase for aspect descriptions compared to the conditional random field (CRF) model with the use of LSTM when F1 spans are matched strictly.

The extraction of user opinions on mobile phones from reviews outside of the collected corpus was conducted as practical confirmation of the proposed model. In addition, opinions from other product categories like skin care products, TVs and tablets were extracted. The examples show that the model can successfully extract user opinions from different kinds of reviews. The results obtained can be useful for computational linguists and machine learning professionals, heads and managers of online stores for consumer preference determination, product recommendations and for providing rich catalog searching tools.

Key words: user feedback; deep learning; machine learning; natural language processing; opinion processing.

Citation: Yekhlakov Yu.P., Gribkov E.I. (2018) User opinion extraction model concerning consumer properties of products based on a recurrent neural network. *Business Informatics*, no. 4 (46), pp. 7–16. DOI: 10.17323/1998-0663.2018.4.7.16

[1] This study was conducted under government order of the Ministry of Education and Science of Russia, project No. 8.8184.2017/8.9

Introduction

Potential consumers face a challenging choice when looking for complicated technical devices (for example, a telephone, a refrigerator or a TV). The existence of a large number of manufacturers and model lines on the market, the variability of possible specifications of goods lead to a compound growth in the number of possible options for consumer properties of similar products. The consumer either randomly chooses a product relying on a brand name or makes a decision guided by advertising. With the rapid growth of the internet and social networks, most of pragmatic consumers are guided not only by advertisements, but also by user opinions of the consumer properties that become apparent over time.

Manufacturers are also interested in better understanding of their clients: which goods do they prefer, what pros and cons of product features do they notice. Based on this data, decision makers can create a product assortment, provide individual selection of goods and services, make special offers to clients and do other activities to raise loyalty of clients and increase competitiveness.

There are a great number of sources that can provide you with user opinions about goods. They can be thematic forums, review articles and videos and social network communities. Chain stores let their clients provide reviews of goods they bought on their sites. Aggregators like "Yandex.Market" can ease the search of this kind of information by collecting user comments in one place along with the ability to rate the usefulness of the information contained.

However, the majority of such platforms only solve a problem of gathering information in one place, without analyzing and generalizing information automatically. Users are forced to study and analyze the reviews on their own, which can be problematic considering the large amount of information. Natural language processing (NLP) methods based on machine learning can deliver a well-developed solution and practical representation of information for analysis of the user opinion extraction task concerning consumer properties of products.

Nowadays, most of the papers devoted to user review processing are based on sentiment analysis of the text. At the same time, sentiment is considered as an attribute of the whole text or its large parts (paragraphs and sentences) [1, 2], which is not sufficient for user opinion extraction. Research in the area of aspect-oriented sentiment analysis is devoted to the problems of searching for definite aspect mention of products (consumer properties and features) and determining the user attitude to them in general, by means of placing them into one of the categories: good, bad, neutral, unknown [3, 4]. The most developed problem statement of defining user attitude towards a product is a detailed sentiment analysis [5], where it is suggested to label aspects and opinions in review texts, where the user expresses sentiments about the given aspect. There are a number of works where conditional random fields [3, 6, 7] are used for detailed sentiment analysis. Syntactic analyzer results (part-of-speech tagging, dependency trees, immediate constituent trees etc.) are used as inputs for segmentation. Such analyzers are not available for a vast number of natural languages. In addition, the accuracy of these analyzers depends on the nature of training data. Moreover, the given model uses special glossaries: emotional, sentiment etc. Recent advances in deep learning based NLP statistic modeling allows us to avoid using extra features when training the models [8, 9]. Such models on their own learn necessary features for problem solving during training.

In this paper, we offer a long short-term memory (LSTM) model for user opinion

extraction which does not require the presence of such analyzers and relies on pre-trained vector representation). For training and quality testing, we use the labeled corpus (dataset) of mobile phone reviews marked by us.

1. Construction of the training sample for user opinions extraction from texts

There are few datasets for fine-grained sentiment analysis available at this moment. So, [10] describes the USAGE corpus – an annotated set of 800 Amazon user reviews from 8 categories about rather simple electronic devices like toasters and coffee machines. The proposed review texts annotation scheme consists of the following entities:

✦ **aspect** is an important product property or mention of it (with indication if aspect belongs to the product which is described in reviews);

✦ **description** is a text that contains the user's opinion about aspect (with sentiment);

✦ **coreference resolution** is used for cases where aspect refers to an entity in another sentence in the review;

✦ **aspect–description link** – for grouping together related aspects and descriptions.

Although the USAGE corpus is available, the authors decided to make an additional annotated corpus. This decision was motivated by two factors. First, the desire to evaluate the quality of the opinion extraction algorithms on the more complex and featured packed products (so we annotated reviews about mobile phones). Second, due to human resources limitations, we decided to use a simpler annotation scheme compared to the one used in USAGE (we dropped coreference resolution and annotated only full aspect–description pairs).

Annotation was done on the user reviews corpus from the Amazon online store presented in [11]. This corpus contains 143 million review texts about 25 product categories written during the period from May 1996 to July 2014 along with metadata about product title, identifier and product description, product category, brand, price, author identifier and user rating. The annotators were asked to mark aspect and description spans within review texts. In contrast to the USAGE dataset, we gave strict instructions to mark only full aspect-description pairs that together form user opinions. Below "opinion" will be referred as a pair

$$O = <A(a_{begin}, a_{end}), D(d_{begin}, d_{end})>,$$

where $A(a_{begin}, a_{end})$ – opinion's aspect starting with a word on position a_{begin} and ending with a word on position a_{end};

$D(d_{begin}, d_{end})$ – opinion's description starting with a word on position d_{begin} and ending with a word on position d_{end}.

Herein spans from different opinions should not intersect. Thus, an annotated sentence from a review text with opinion O_1 consisting of aspect with associated span $A(1, 2)$ and description with associated span $D(4, 5)$ can be presented this way:

$$\left[Battery_1\ life_2 \right]_{A-O_1}\ is_3 \left[quite_4\ impressive_5 \right]_{D-O_1}\ !_6 .$$

3,232 reviews were annotated in total. The annotated corpus contains 9,344 opinions, 1,994 unique aspects, 5,124 unique descriptions. The quantitative description of the annotated corpus is presented in *Table 1.*

2. User opinion extraction model

Tasks of the user opinion extraction model can be presented as a sequence tagging task where for each element of input sequence a class label (tag) should be determined. This requires spans of opinions to be reshaped as

Table 1.

Quantitative description of mobile phones user reviews corpus

Quantitative description	Value
Number of reviews	3,232
Number of opinions in corpus	9,344
Number of unique aspects	1,994
Number of unique descriptions	5,124

a tag sequence. One of the most popular ways to represent span as a sequence is an IOB format [12]. It uses three different tag types: **O** – absence of any particular entity; **B-X** – beginning of entity of type X; **I-X** – continuation of entity of type X. This paper proposes two types of entities – aspects and descriptions denoted with labels "Aspect" and "Description". Then the set of possible tags **Y** will contain the following elements: **O** – absence of any particular entity; **B-Aspect** – beginning of the aspect; **I-Aspect** – continuation of aspect; **B-Description** – beginning of the description; **I-Description** – continuation of description. This way we can uniquely associate a sentence containing a set of opinions with the sequence of tags as shown in the *Figure 1.*

As a formal technique for solving this problem, we proposed to use recurrent neural networks (RNNs). This kind of neural networks is widely used to solve a broad variety of machine learning tasks like natural language modeling [13], part-of-speech tagging [8], sequence classification [14], audio recognition [15], time series forecasting [16], etc.

The input of RNN at each time step t is the next element of an arbitrary sequence which is transformed into a sequence of outputs by recurrent relationships between the sequence of hidden states:

$$h_t = f\left(U x_t + W h_{t-1} + b\right),$$

where x_t – current input;

h_{t-1} – previous hidden state;

U and W – input and recurrent transformation matrices;

b – bias;

f – nonlinear activation function.

Recurrent connections between the hidden state allows for transferring contextual information about a sequence under processing and use of this information when predicting outputs h_t. This way we can see h_t as an intermediate representation of sequence that accumulates information about the preceding steps

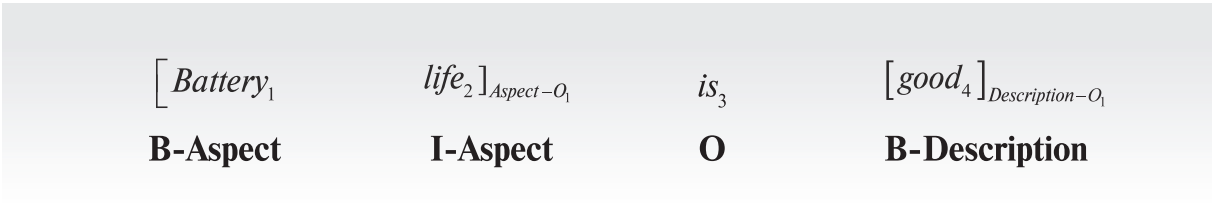

Fig. 1. Correspondence between opinion spans and IOB tag sequence

of processing. In this paper, we use the LSTM recurrent neural network [17] because it is less exposed to the vanishing gradient problem [18] compared to simple RNN:

$$\begin{aligned} f_t &= \sigma\left(W_f\, x_t + U_f\, h_{t-1} + b_f\right) \\ i_t &= \sigma\left(W_i\, x_t + U_i\, h_{t-1} + b_i\right) \\ o_t &= \sigma\left(W_o\, x_t + U_o\, h_{t-1} + b_o\right) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tanh\left(W_c\, x_t + U_c\, h_{t-1} + b_c\right) \\ h_t &= o_t \odot \tanh\left(c_t\right). \end{aligned} \quad (1)$$

A classifier for word tag determination is used in the following way: to each output value we apply linear transformation and the softmax function, resulting in a possible tag probability distribution:

$$P(\hat{y}_t \mid x_t)_i = \frac{\exp\left(W_{tag}\, h_t + b_{tag}\right)_i}{\sum_j \exp\left(W_{tag}\, h_t + b_{tag}\right)_j}. \quad (2)$$

From the general RNN definition, it follows that information propagates in left-to-right order within the network. In some cases, it can be useful to know the context of subsequent words for correct classification of the current word. In order to allow the use of information from both directions, in each step of the prediction a bidirectional neural network is used, one of which processes input sequence in left-to-right order and other processes it in right-to-left order, after which the hidden states corresponding to the same position are concatenated:

$$\begin{aligned} \overleftarrow{h}_t &= f\left(\overleftarrow{U} x_t + \overleftarrow{W}\overleftarrow{h}_{t-1} + \overleftarrow{b}\right), \\ h_t &= \left[\vec{h}_t ; \overleftarrow{h}_t\right]. \end{aligned}$$

It should be noted that in structured prediction tasks (which include the task of sequence tagging) there are dependencies between tags in the output sequence. Therefore, models that don't take these dependencies into account can produce ill-formed tag sequences. For example, when predicting tag sequence in the IOB format sequence of predictions the **O I-Aspect** is wrong because the tag **I-Aspect** can only follow after the corresponding **B-Aspect** tag.

To model correlations between different predictions within the same sequence, it is proposed to use the conditional random field (CRF) model that was proposed in [19]:

$$s\left(W, y\right) = \sum_{i=0}^{n} A_{y_i,\, y_{i+1}} + \sum_{i=1}^{n} P_{i,\, y_i}\,, \quad (3)$$

where A – matrix with probability of transition from tag y_i to tag y_{i+1};

$P_{i,\, y_i}$ – probability of tag on position i;

$W\{w_1, w_2, ..., w_n\}$ – input sequence;

$y\{y_1, ..., y_n\}$ – predicted sequence.

Then the probability of the sequence of predictions y is evaluated as follows:

$$p(y \mid W) = \frac{\exp\left(s\left(W, y\right)\right)}{\sum_{\dot{y}} \exp\left(s\left(W, \dot{y}\right)\right)}, \quad (4)$$

where summation in the denominator happens by all possible sequences y.

During training of the conditional random filed model, the log-probability of the true tag sequence is maximized:

$$\log(p(y \mid W)) = s\left(W, y\right) - \log\left(\sum_{\dot{y}} exp\left(s\left(W, \dot{y}\right)\right)\right). \quad (5)$$

An optimal sequence of predictions can be computed using dynamic programming. In doing so, the optimal sequence of predictions should correspond to the maximum of expression $p(y \mid W)$ as follows:

$$\hat{y} = \arg\max_y p(y \mid W)\,.$$

Using the model of conditional random field allows us to predict a globally optimal structure only in case the linear structure is considered and only local features are used for each node of prediction. This limitation led to the devel-

opment of structured prediction methods that can handle more complex structures (trees, for example) and use non-local features (like word classification results from previous steps) [20, 21]. Accordingly, for word tag prediction the following expression is proposed:

$$P(\hat{y}_t \mid \phi(c_t))_i = \frac{\exp\,(W_{tag}\phi(c_t) + b_{tag})_i}{\sum_j \exp\,(W_{tag}\phi(c_t) + b_{tag})_j}, \quad (6)$$

where c_t – model configuration at the moment t;

$\phi(c_t)$ – function for mapping model configuration c_t to feature set.

Applying the expression (6) for the user opinion extraction task in practice requires that we define the form of configuration c and state-to-features mapping function $\phi(c_t)$. In our work, we use inspiration from the dependency parsing model presented in [22]. Configuration is defined as 4-tuple $c = (S, B, l, Y)$ consisting of the buffer B that holds unprocessed elements of input sequence, the stack S that holds words from all currently found entities, the tag of last found entity l (Aspect or Description) and partially constructed output sequence Y. Feature vector $\phi(c_t)$ is formed at each step t which is used to determine tag $\hat{y}_t$ of current word and change model configuration according to this tag. The model configuration transition rules presented in *Table 2*. The semicolon symbol in the table denotes sequence concatenation.

The following form of $\phi(c_t)$ is proposed based on model configuration c_t and configuration transition rules:

$$\phi(c_t) = \left[B_t, ..., B_{t+M_B}, S_1^{t-1}, ..., S_{M_S}^{t-1}, Y_{t-1}^{t-1}, Y_{t-2}^{t-1}\, E_{l^t} \right],$$

where B_i – i-th element of buffer B;

S_j^t – j-th element of stack S on the step t;

E_k – k-th row of matrix E;

Y_n^t – n-th element of predicted tags sequence on the step t.

We use hidden states from the last layer of multilayer bidirectional LSTM as the elements of the buffer. In this regard, every element of the buffer will contain information not only about the word at the corresponding position, but also about the preceding and subsequent context. The authors assume that information about words in a stack of found entities will contribute to the accuracy of a starting position detection for subsequent entities. For example, the "battery life" aspect found may give a hint to the model that the next words "is perfect" are the description. In addition, an extra hint is the tag of the previously found entity l^{t+1}. Rows of matrix $E \in \mathbb{R}^{2\times d}$ serve as features for l. Input for LSTM are pretrained vector representations of words obtained with the use of the FastText model [23]. This work uses vectors[2].

Table 2.

Model configuration transition rules

$\hat{y}_t$	S^{t+1}	l^{t+1}	Y^{t+1}	Precondition
$B-y$	$b_t; S^t$	y	$Y^t; \hat{y}_t$	—
$I-y$	$b_t; S^t$	l^t	$Y^t; \hat{y}_t$	$\hat{y}_{t-1} \in \{B-y, I-y\}$
O	S^t	l^t	$Y^t; \hat{y}_t$	—

[2] Source: https://github.com/plasticityai/magnitude

When calculating tag probabilities $P(\hat{y}_t \mid \phi(c_t))_i$ only tags that obey preconditions from *Table 2* are considered. This makes it possible to avoid an ill-formed prediction sequence.

3. Sequence tagging algorithm

Based on the foregoing, the sequence tagging algorithm can be framed as follows.

Let the input text be given as a sequence of words $W\{w_1, w_2, ..., w_n\}$. We should determine the output sequence of word tags $Y\{y_1, y_2 ..., y_n\}$. The Adam optimization method [24] with parameters $lr = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and gradient clipping at 3.0 is used to train the model.

Step 1. Initialize model state as $B = \varnothing$, $C = \varnothing$, $l = 0$, $Y = \varnothing$, $t = 0$.

Step 2. Fill buffer with hidden states from input sequence processed by LSTM network: $\forall t \;\; B_t = LSTM\left(w_t, B_{t-1}\right)$.

Step 3. If $t < n$, then go to step 4, otherwise go to Step 6.

Step 4. Make the feature vector $\phi(c_t)$ and determine the tag for current position in buffer: $\hat{y}_t = \arg\max P(\hat{y}_t \mid \phi(t))$.

Step 4. Change S, i, Y according to the rules from *Table 2* depending on the tag $\hat{y}_t$.

Step 5. $t = t + 1$, go to Step 3.

Step 6. End.

The results obtained from processing mobile phone reviews from the Amazon online store by proposed model and algorithm are described in *Table 3*. For example, from the sentence "The screen is fantastically large while the overall dimensions of the phone are manageable for those without giant hands" opinions "screen is fantastically large" and "dimensions of the phone are manageable" were extracted.

Furthermore based on the annotated dataset, user opinions on other product categories were processed (*Table 4*). The results obtained suggest that the model has shown good results both for extracting opinions on mobile phones and on products of other categories.

4. Experimental evaluation of the model

Experimental evaluation of the proposed model was done in comparison with the bidirectional CRF–LSTM model without character features from [22]. Models were trained with the backpropagation method. Parameters were optimized by Adam [24] with the following parameters: $lr = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.99$ and gradient clipping at 3.0.

Table 3.

Opinions extracted from mobile phone reviews

Product	Opinions
Sony Xperia XA	Phone is awesome; phone is easy to use; phone is perfect for those who need extra storage; battery life is mediocre; battery life is absolutely terrible compared; no great sound
Apple iPhone 6S	It didn't work properly from the beginning; it's a decent; bad charger; worry free product
Huawei P20	Phone is a flagship performer; phone stopped receiving phone calls; phone is absolutely amazing for the price; the screen is fantastically large; camera is simply amazing; fantastic camera; camera produces great photos

Table 4.

Opinions extracted from reviews of other product types

Product	Opinions
EltaMD PM Therapy Facial Moisturizer	is a great night cream product; product highly emollient without being greasy; product recommended by my dermatologist; very moisturized skin feel
Samsung UN55MU6500 Curved 55–Inch 4K Ultra HD Smart LED TV	Easy setup TV; TV is not worth the money or aggravation; wonderful picture; picture is so clear; remote is easy to use; remote is ergonomic and a breeze to use; color is unbelieveable
Fire 7 Tablet with Alexa	Small tablet; amazing little tablet; screen does not react well to water; screen is freezing; battery doesn't hold a charge; battery dies to quickly; charging port its weakness

The dataset described in section 1 was used for user opinions extraction. Because of the small sample size, estimation was done by 5-fold cross-validation. To exclude the influence of the lexicon of train sample during testing, we split data at the documents level.

The proposed model and CRF-based one both use 2-layer bidirectional LSTM with 100 dimensional hidden state; the input vector size is 100. We used the following hyperparameters for mapping $\phi(t)$: $C_{BT} = 2$, $C_{ST} = 4$, $e_{label} \in \mathbb{R}^{20}$.

During testing, we tracked a set of criteria common for such kinds of tasks: precision, recall and F1-measure which determines the overall quality of the model. Values of criteria were calculated in two ways: strict – when the match counted only if the found span is the same as true span; soft – when match counted if found and true spans have at least one common word.

Analysis of *Tables 5* and 6 reveals that the proposed model shows better performance than the Bi-LSTM-CRF model for both ways of criteria calculation. For strict matching, absolute improvement in the aspect span detection is 4.51%; in the description span detection – 5.44%. For soft matching, absolute improvement in the aspect span detection is 3.77%; in the description span detection – 3.52%.

Table 5.

Results of extracting opinions from mobile phones dataset (strict matching)

Model	Aspects			Description		
	R	P	F1	R	P	F1
Bi–LSTM + CRF	39.20	50.58	44.17	41.30	54.03	46.82
Proposed model	47.87	49.51	48.68	49.70	52.93	52.26

Table 5.

Results of extracting opinions from mobile phones dataset(strict matching)

Model	Aspects			Description		
	R	P	F1	R	P	F1
Bi–LSTM + CRF	53.93	63.05	57.83	56.09	64.93	60.19
Proposed model	61.08	62.74	61.9	62.49	64.98	52.26

Conclusion

The suggested approach as a combination of the annotated dataset and an LSTM-based structured prediction model allows us to extract from review texts opinions on consumer properties of both the product as a whole and its features. The developed RNN-based structured prediction model is capable of using non-local features for entities prediction and does not require additional syntactic features.

The model trained on dataset has shown better results compared to the CRF-based model: F1 for aspect extraction is higher by 4.51%, F1 for description extraction is higher by 5.44%. The experiments carried out have revealed that extra features in the base model have positive effect on the results.

The results obtained can be useful for NLP and computational linguist specialists, for the business community when selling goods, providing services and developing their consumer properties.

In follow-on papers we will discuss questions on the prediction of links between aspects and descriptions for better opinion extraction and also incorporating information about opinion sentiment, as well as indication if the aspect corresponds to the product under review and coreference resolution. ■

References

1. Sadegh M., Ibrahim R., Othman Z.A. (2012 Opinion mining and sentiment analysis: A survey. *International Journal of Computers & Technology*, vol. 2, no. 3, pp. 171–178.
2. Zhang L., Wang S., Liu B. (2018) Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, pp. 1942–4787.
3. Pontiki M., Galanis D., Papageorgiou H., Androutsopoulos I., Manandhar S., Mohammad Al-S., Al-Ayyoub M., Zhao Y., Qin B., De Clercq O. (2016) SemEval-2016 task 5: Aspect based sentiment analysis. Proceedings of the *10th International Workshop on Semantic Evaluation (SemEval–2016). San Diego, CA, USA, 16–17 June 2016*, pp. 19–30.
4. Jo Y., Oh A.H. (2011) Aspect and sentiment unification model for online review analysis. Proceedings of the *Fourth ACM International Conference on Web Search and Data Mining (WSDM 2011). Hong Kong, China, 9–12 February 2011*, pp. 815–824.
5. Zirn C., Niepert M., Stuckenschmidt H., Strube M. (2011) Fine-grained sentiment analysis with structural features. Proceedings of *5th International Joint Conference on Natural Language Processing (IJCNLP 2011). Chiang Mai, Thailand, 8–13 November 2011*, pp. 336–344.
6. Yang B., Cardie C. (2012) Extracting opinion expressions with semi-Markov conditional random fields. Proceedings of the *2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP–CoNLL 2012). Jeju Island, Korea, 12–14 July 2012*, pp. 1335–1345.

7. Yang B., Cardie C. (2013) Joint inference for fine-grained opinion extraction. Proceedings of the *51st Annual Meeting of the Association for Computational Linguistics. Sofia, Bulgaria, 4–9 August 2013*. Vol. 1, pp. 1640–1649.
8. Collobert R., Weston J., Bottou L., Karlen M., Kavukcuoglu K., Kuksa P. (2011) Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, no. 12, pp. 2493–2537.
9. Zhai F., Potdar S., Xiang B., Zhou B. (2017) Neural models for sequence chunking. Proceedings of the *Thirty-First Conference on Artificial Intelligence (AAAI-17). San Francisco, CA, USA, 4–9 February 2017*, pp. 3365–3371.
10. Klinger R., Cimiano P. (2014) The USAGE review corpus for fine-grained, multi-lingual opinion analysis. Proceedings of the *Language Resources and Evaluation Conference (LREC 2014). Reykjavik, Iceland, 26–31 May 2014*, pp. 2211–2218.
11. He R., McAuley J. (2016) Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. Proceedings of the *25th International Conference on World Wide Web (WWW 2016). Montreal, Canada, 11–15 April 2016*, pp. 507–517.
12. Sang E.F., Veenstra J. (1999) Representing text chunks. Proceedings of the *Ninth Conference on European Chapter of the Association for Computational Linguistics (EACL 1999). Bergen, Norway, 8–12 June 1999*, pp. 173–179.
13. Mikolov T., Karafiat M., Burget L., Cernocky J., Khudanpur S. (2010) Recurrent neural network based language model. Proceedings of the *Eleventh Annual Conference of the International Speech Communication Association (INTERSPEECH 2010). Makuhari, Chiba, Japan, 26–30 September 2010*, vol. 2, pp. 1045–1048.
14. Ghosh M., Sanyal G. (2018) Document modeling with hierarchical deep learning approach for sentiment classification. Proceedings of the *2nd International Conference on Digital Signal Processing (ICDSP 2018). Tokyo, Japan, 25–27 February 2018*, pp. 181–185.
15. Graves A., Jaitly N., Mohamed A. (2013) Hybrid speech recognition with deep bidirectional LSTM. Proceedings of the *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2013). Olomouc, Czech Republic, 8–12 December 2013*, pp. 273–278.
16. Guo T., Xu Z., Yao X., Chen H., Aberer K., Funaya K. (2016) Robust online time series prediction with recurrent neural networks. Proceedings of the *IEEE International Conference on Data Science and Advanced Analytics (DSAA 2016). Montreal, Canada, 17–19 October 2016*, pp. 816–825.
17. Hochreiter S., Schmidhuber J. (1997) Long short-term memory. *Neural Computation*, vol. 9, no. 8, pp. 1735–1780.
18. Bengio Y., Simard P., Frasconi P. (1994) Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166.
19. Lafferty J., McCallum A., Pereira F.C.N. (2001) Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proceedings of the *18th International Conference on Machine Learning (ICML 2001). Williamstown, MA, USA, 28 June – 1 July 2001*, pp. 282–289.
20. Chen D., Manning C.D. (2014) A fast and accurate dependency parser using neural networks. Proceedings of the *19th Conference on Empirical Methods in Natural Language Processing (EMNLP 2014). Doha, Qatar, 25–29 October 2014*, pp. 740–750.
21. Dyer C., Ballesteros M., Ling W., Matthews A., Smith N.A. (2015) Transition-based dependency parsing with stack long short-term memory. Proceedings of the *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. Beijing, China, 27–31 July 2015*, vol. 1, pp. 334–343.
22. Lample G., Ballesteros M., Subramanian S., Kawakami K., Dyer C. (2016) Neural architectures for named entity recognition. Proceedings of the *15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HTL 2016). San Diego, CA, USA, 12–17 June 2016*, pp. 260–270.
23. Bojanowski P., Grave E., Joulin A., Mikolov T. (2017) Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, no. 5, pp. 135–146.
24. Kingma D.P., Ba J.L. (2017) *Adam: A method for stochastic optimization*. arXiv:1412.6980v9 [cs. LG]. Available at: https://arxiv.org/pdf/1412.6980.pdf (accessed 10 October 2018).