

Task-oriented Conversational Agent Self-learning Based on Sentiment Analysis

Serena Leggeri¹, Andrea Esposito², and Luca Iocchi³

¹ Sapienza University of Rome, Italy
leggeri.1228424@studenti.uniroma1.it

² BadgeBox srl, Roma, Italy

andrea.esposito@badgebox.com

³ Sapienza University of Rome, Italy
iocchi@diag.uniroma1.it

Abstract. One of the biggest issues in creating a task-oriented conversational agent with natural language processing based on machine learning comes from size and correctness of the training dataset. It could take months or even years of data collection and the resulting static resource may get soon out of date thus requiring a significant amount of work to supervise it. To overcome these difficulties, we implemented an algorithm with the ability of improving learning efficiency based on the emotions and reactions arising from the conversation between a user and the bot, automatically and in real time. To this end, we have studied an error function that, as in any closed loop control system, corrects the input to improve the output. The proposed method is based on both calibrating the interpretation given to the initial dataset and expanding the dictionary with new terms. Thanks to this innovative approach, the satisfaction of the interlocutors is higher if compared to algorithms with a static dataset or with semi-automatic self-learning rules.

Keywords: task-oriented conversational agent · self-learning · semi-supervised learning

1 Introduction

Task-oriented conversational agents are software components based on artificial intelligence that are able to simulate an intelligent conversation with the user on a chat and offer a functional support service through the main messaging platforms such as Slack, Telegram and Facebook Messenger. Conversational agents are created for various purposes: from customer care, to the dissemination of news, offers, promotions and as support for the activation of a service. The strength of these solutions is in being autonomous, available 24 hours a day to offer help to the user who requests it.

When a developer designs a task-oriented conversational agent, its main purpose is to make sure that it fulfills all the user requests based on the specific topic for which it was designed, trying to find the most relevant answer to the

question that was sent, without the intervention of human operators. Agents based on machine learning techniques make use of a training dataset. Initially, the dataset contains a finite number of contexts that describe the topic for which the bot was created and for each context there is a finite number of sentences describing the user’s intention. As the dataset is created to satisfy certain types of requests, there may be limited ways in which the user can ask a question and limited types of answers. In some cases, the answer may not be adequate to the question and it is important to keep improving the efficacy of the agent during the operation. However, this requires manual operations in labelling new samples for refining the learning process.

The idea of the method described in this paper is to exploit the analysis of user satisfaction, to improve the effectiveness of the learning process of the agent. More specifically, we aim at on-line and automatic generation of new labelled samples to be included in the training dataset to refine the agent learned model.

Starting from this idea, we have developed a method that allows to increase the dataset automatically and in real time inserting new terms and recalibrating those already present in the dataset thus improving the recognition of the user’s intentions. To do this, we analyzed the emotionality generated in the users by the bot answers.

The proposed approach has been deployed and validated on a real use case, coming from a commercial application of a chatbot acting as customer care helping on timesheet and employee management in a company [1]. The evaluation process also contains a comparison with other techniques. The results show that, when compared with other techniques not using such analysis, the proposed method can automatically increase the dataset in real time and improve the quality of the chatbot’s answers. The proposed method is also faster in recognizing the contexts compared with other techniques.

Although the deployment and experimental evaluation have been focussed on a particular real use case, the proposed method has no domain specific components or assumptions and thus we believe that it can be properly applied to other domains as well.

2 Related work

Over the course of time, numerous chatbots have been created to provide information, help making decisions, allow services or simply for entertainment [2].

Initially, the development of a bot was based on two fundamental components [3]:

- Natural Language Understanding module, used by the Dialogue Manager, that processes the user input to search for keywords through which to understand the action to be taken.
- Natural Language Generation module that generates answers from the information gathered by the Dialogue Manager.

Over time, we have faced a real evolution in the development of task-oriented conversational agents thanks to the availability of deep learning techniques [4] [5].

These agents are typically trained for several years with the use of human intervention supervising them and verifying the correctness of the answers. In this way, it was able to carry out operations that the other assistants could not perform such as making payments or booking a trip.

The most difficult challenge for a task-oriented conversational agent is to be able to incorporate the linguistic context (all that is said during the conversation and that gives meaning to each taking of the turn in the dialogue) and the physical context (for example user information, place, date and time of the conversation, etc.). A good exploitation of the linguistic context is possible only when the bot is trained on a good training dataset supervised by human experts in order to minimize errors.

In addition to task-oriented conversational agents realized assuming a large amount of training data and the availability of human experts during all the training phases, it is interesting to study development of agents with minimal requirements in terms of availability of training data and human supervision. This goal is motivated by the need of developing chatbots by small companies for which high amount of data and human supervision have a too high costs compared to the commercial benefit of the chatbot product.

Therefore, our goal is to study the definition of a task-oriented conversational agent to understand user intents and perform actions starting from a limited dataset and improving its performance over time in a semi-automatic way.

Some approaches for machine learning classification exploiting both labeled and unlabeled data include:

- *Co-training* [6] [7], where a labeled dataset is used to train two classifiers A and B, while the unlabeled dataset is divided in two subsets: each subset is classified through one classifier (e.g., A) and the confident values are used to train the other classifier (e.g., B).
- *Re-weighting* [8] and *Common Components Using EM* [9] [10] aim at finding a function using the labeled dataset to redefine the unlabeled dataset.

Although these approaches provide for interesting ways of exploiting unlabeled data, they are not appropriate to our problem of creating agents from a limited dataset and with minimal human supervision. In the first case, we do not have a function to estimate the error, thus a scheduled human intervention would be required to check for it. In the other two cases, our labeled set is too small to define a real and effective function to apply at the unlabeled set.

The method proposed in this paper is inspired by the co-training approach combined with an estimation of error by sentiment analysis for the bot to learn what is right and what is wrong. Moreover, we want to rely on the user's satisfaction without asking for explicit feedback, but analyzing user answers through a sentiment analysis.

A novel contribution of this paper is thus the use of sentiment analysis to drive the on-line automatic learning of an agent.

3 Proposed method

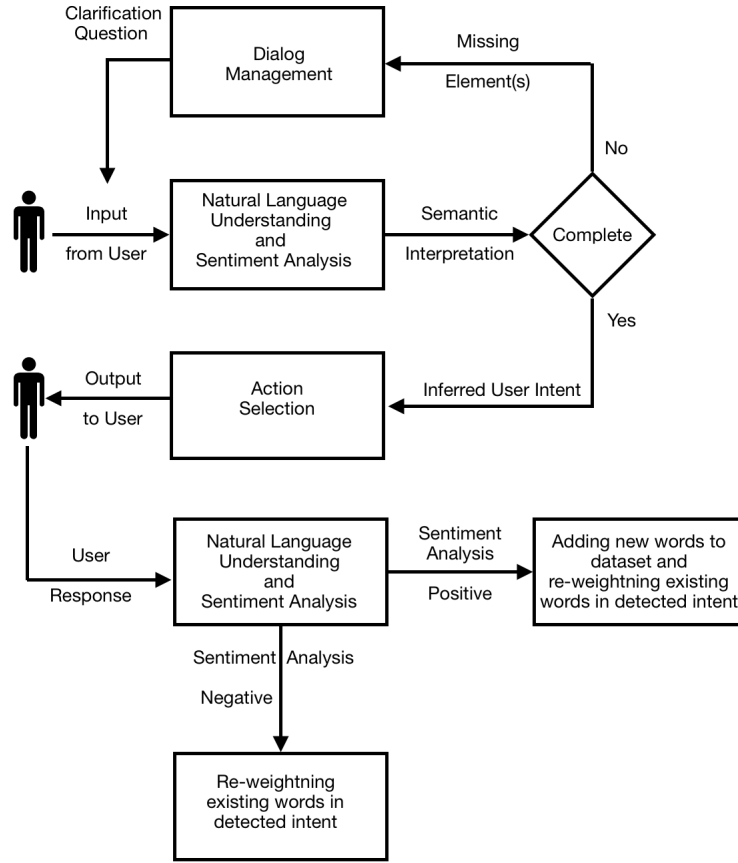


Fig. 1. Architecture of proposed method.

Figure 1 shows the architecture of the proposed method. A first (small) training set is available to train the system off-line before deployment. Then on-line learning, is applied during operation by using sentiment analysis to provide for automatic labeling of new instances.

The method described in this paper is based on a continuous learning process in which each sentence is analyzed against two classification sub-systems: one for identifying the class of the answers, one for assessing the sentiment of the sentence. At the end of the processing of each sentence the learning model is updated according to the detected sentiment. This is based on a data structure formed by **intents**.

Definition 1 (Intents). *An intent is a semantic label representing an intention of the end-user.*

For each intent, we have defined a set of sentences that represent it. Each sentence that describes an intent contains **entities** that are attributes specific to the given intent.

Definition 2 (Entities). *Entities are the parameters of the intent that help in defining the specific user request.*

An example taken from the dataset is shown below.

Example 1. Request: "I need days off from tomorrow to the day after tomorrow".

Intent: *LEAVE_REQUEST*

Entities:

- *start date*: tomorrow.
- *end date*: day after tomorrow.

In this example, the scenario of an employee requesting holidays is represented by the *LEAVE_REQUEST* intent and by the start date and end date entities.

An example of how the proposed method works in case of positive sentiment is shown below.

Example 2. Time off request

Request: "Hi" {*intent: [HELLO] detected: [Hi] sentiment: [neutral]*}

Bot: "Hi, Dave"

Request: "I'm stuck in traffic, I'll be there soon" {*intent: [TIMEOFF_REQUEST] detected: [there,soon] new words: [stuck,traffic] sentiment: [neutral]*}

Bot: "Ok, do you want to create a time off request?"

Request: "Yes, thank you!" {*intent: [CONFIRM] detected: [Yes, thank, you] sentiment: [positive]*}

In this example the agent detects the correct intent by the words *there* and *soon* and enriches the dictionary with *stuck* and *traffic*. In the future, if these words are often used for a time off request, they will become characteristic for this intent. (i.g. Dave in the future he will can write "I'm stuck in traffic" or "There is traffic" to request a time off).

An example of how the proposed method works in case of negative sentiment is shown below.

Example 3. Time off request

Request: "Hi" {*intent: [HELLO] detected: [Hi] sentiment: [neutral]*}

Bot: "Hi, Dave"

Request: "Tomorrow I'll be busy" {*intent*: [LEAVE_REQUEST] *detected*: [tomorrow,busy] *sentiment*: [neutral]}

Bot: "Ok, do you want to create a leave request?"

Request: "No, that's not what I want!" {*intent*: [NOT_CONFIRM] *detected*: [No, that's not, what, I, want] *sentiment*: [negative]}

In this example the agent detects the uncorrect intent by the words *tomorrow* and *busy*. In the future, if the bot will always receive a negative response to the request that he proposes then the words found will no longer be characteristics of the intent found and can be totally eliminated.

Our approach is frame-based, conversational agent extracts from the text the main informations to fill the user's request and if them aren't enough it can directly ask the missing information.

We have also defined a dictionary that the bot uses to translate the type of some words. For example, the terms "tomorrow" and "day after tomorrow" are assigned to the type *date*.

Notice that the concepts of *intents* and *entities* are domain independent, while of course the values associated to them must be provided by an expert of the chatbot application domain.

3.1 Classification of intents

The classification problem considered in our method is determining the intent and the entities associated to a given user sentence.

User sentences are represented with **bag of words** [18], without considering the order of the words. To improve classification accuracy, we also use a vocabulary of n-words with an N-gram model [20].

The classification algorithm is based on *Naive Bayes Text Classifier* [21], a statistical technique able to estimate the probability of an element belonging to a certain class. The Naive Bayes technique estimates the conditional probabilities of each word given the classification category by associating every word that convey the same meaning in the intents, a numerical value that we will consider as a weight. The words that characterize an intent will have greater weight because they will only be found within that intent, so their occurrence is limited compared to non-characterizing words that we find in numerous intents.

More formally, let Z_1, \dots, Z_n be the words that form the user input, the classification process aims at retrieving the intent I_n such that $P\{I_n|Z_1, \dots, Z_n\}$ is maximum.

Example 4. Given an intent *Leave* representing requests from a user regarding leaves, we would like sentences such as "I want go to holidays", "I'm tired, I need to rests", "I want holidays for this month", etc. to be classified as *Leave*.

3.2 Self-learning based on sentiment analysis

The main idea developed in this paper is to provide the agent with the ability to automatically collect feedback about its answers in order to improve its knowledge base. To this end, we experimented the use of sentiment analysis. At the beginning, the agent acts according to a model derived from the initial training set, but during the use the model is updated according to the self-learning process explained in this section. In particular, we have defined an error function for the agent exploiting the sentiment analysis derived from the dialogue between the user and the bot.

To detect the sentiment from user sentences, we have defined another classification problem from user input to three classes: Positive, Negative and Neutral and use again a Naive Bayes approach to train this classifier on a specific dataset [22]. For any user’s sentence, we keep track of local and global sentiment score, local score is about the last sentence, global score is an average value across the dialogue. Furthermore, to improve the idea, we can define some particular intents A_n that act as modifiers. For example, when the user corrects the bot with phrases like "I’m sorry I did not mean this", this is considered as a negative feedback, while phrases containing specific thanks, such as "Thank you! I was trying to do exactly this!" provide for positive feedback.

Based on the result of the sentiment analysis, we can recalibrate the calculated weights of words w_{ij} for the Intent classifier and, if necessary, add new terms to the dataset. We have implemented a low-pass filter [28] for smoothing the high frequency in update function in order to mitigate possible errors.

The main algorithm for self-learning is summarized below.

Let:

1. I_i be the i -th detected intent in the user input U
 2. w_{ij} be the value of the weight of the j -th word in U
 3. c be a value between 0 and 1 that represents the sentiment for the user during the dialog and it’s computed by the results of negative or positive nearest of the words
 4. m_v and M_v be constant values (in our experimental sessions we set $m_v = 0.1$ and $M_v = 0.3$)
 5. v be a variable set to m_v by default, and set to M_v if $I_i \in A$ (where A is a know positive or negative intent)
 6. k be a constant (in our experimental session we set $k = 0.4$).
- If the sentiment analysis is positive, a new word is added to the vocabulary (if it does not exist) and the weight for every word in the user input (if it is already present) is recalculated, according to this formula:

$$w_{ij} = w_{ij}(1 - v) + n_{ij}v \tag{1}$$

where $n_{ij} = w_{ij} + ck$.

- If the sentiment analysis is neutral, no changes are made.

- If the sentiment analysis is negative than the weight for every word in the user input (if it is present) is recalculated according to the formula (1) where $n_{ij} = w_{ij} - ck$. When w_{ij} becomes negative, the word is removed from the detected intent.

4 Experimental results

The task-oriented conversational agent described in this paper was created to be used as a virtual assistant to help people understand the use of BadgeBox [1] and to help them to perform actions directly and easily. The experiments reported in this section have been performed during normal operations of the chatbot in a real operational setting. More specifically, to verify the effectiveness of self-learning through the use of sentiment analysis, four groups of six people were recruited to interact with the chatbot for 28 days. We have selected users between 20 and 50 years old, with an almost equal gender distribution (13 males and 11 females). The groups were randomic but with a fair distribution of genders and ages. We have created 4 instances of the bot and invited them to use the correspondent instance. Each day, at 18 o'clock, we used to send them a survey about the use of the system and how it was useful for the scope. The survey asks an evaluation from -5 to 5 about how the bot was responsive for their requests. With these evaluations we have been able to elaborate their satisfaction. None of the chosen people has worked on the project development, so none of the users were not aware of the phrases present in the dataset. The four variants of the chatbot tested were:

1. a baseline without self-learning
2. a method using a random self-learning
3. supervision of a human who updates the dataset manually
4. the proposed self-learning method based on sentiment analysis

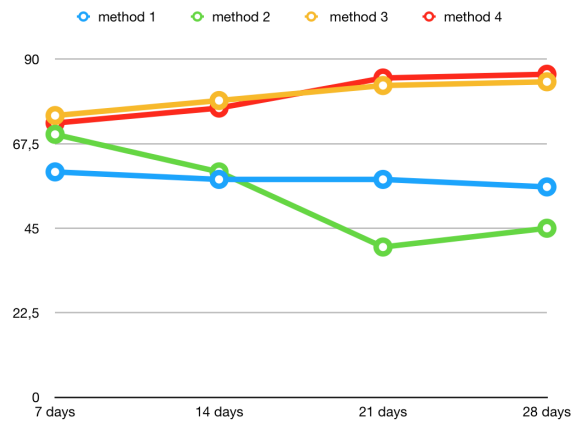


Fig. 2. Intents detected

Day	Method	Interactions	Satisfaction	Identified intents	Variance
7	1	1230	-0.5	60%	6.06
	2	1150	0.6	70%	8.44
	3	1101	0	75%	1.76
	4	1180	0.2	73%	1.84
14	1	2250	-1.2	58%	7.46
	2	2200	-0.2	60%	8.48
	3	2050	0.6	79%	1.52
	4	2120	0.5	77%	1.67
21	1	3020	-1.6	58%	7.13
	2	3100	-0.8	40%	6.83
	3	3270	0.9	83%	1.25
	4	3120	1.1	85%	1.21
28	1	4530	-1.7	56%	8.68
	2	4332	-0.6	45%	7.65
	3	4157	1.0	84%	1.18
	4	4423	1.18	86%	1.15

Table 1. The experimental results obtained during the experiment.

The experimental results are summarized in Table 1, where the columns have the following meaning:

- *Day*: day on which the experimental data are collected.
- *Method*: variant of the chatbot tested.
- *Interactions*: the number of requests made to the bot during the testing period.
- *Satisfaction*: the average satisfaction score of the user surveys.
- *Identified intents*: correctness of the bot answers compared to the user requests. We have checked them by analyzing the operations run on the application during the test.
- *Variance*: satisfaction variance.

Table 1 shows the results obtained after 7, 14, 21 and 28 days of tests. We can see how the system proposed in this paper (Method 4) improves the level of satisfaction compared to the baseline methods (Methods 1 and 2) and has similar or superior recognition performances compared to the manually supervised one (Method 3). It is interesting to notice also that the self-learning method retains the advantage over the manually supervised system, probably because the manual supervision was updated not in real time but on a weekly basis. This result further confirms the benefits of on-line self-learning.

At the moment, we are running the test in production environment and we are constantly monitoring how the novel approach is performing. After a month of testing, with about 1.500 users, we have noticed an increment of 10% of correctness of the answers of the bot with respect to the right intent (from 65% to 75,4%).

From the results obtained during these experiments, we can conclude that the agent is actually able to self-learn through sentiment analysis and achieves

performance that are similar to the agents that learn through human supervision, but with a significant less effort in human resources necessary for training purposes.

5 Conclusions and future works

In this paper, we have presented a self-learning method for a chatbot, using the analysis of the emotionality of the responses sent by the user as an error function and as a self-learning balancer.

One of the difficulties overcome and solved was precisely the nature of the discursive data source that can not be defined before, leaving the domain of the unlabeled set unknown until it is written by the user.

With our implementation and our experimental results, we have shown that the sentiment analysis approach is able to improve the initial dataset in real time and automatically, choosing on the basis of the answers what to learn continuously and in a collaborative way among all the users who interact with the bot.

The results also show that our approach is better, in some cases, or similar to the one supervised by a human expert, where answers and learning are carried out thanks to the intervention of a person who corrects and improves the dataset manually.

Consequently, the proposed method, in addition to significantly reducing maintenance costs, paves the way for many applications aimed at customer satisfaction and make the software closer and more similar to the people who have to use it.

Learning slang, new specific words and consequently improving and expanding the training set, makes its service to users more effective, including more complex vocabularies and adapting to the target audience.

After the various tests performed, we have implemented the bot in a production application by tracing the quantity of actions performed and the degree of satisfaction. The system is still on-line and is able to guarantee the use of the software via chat in a natural and increasingly comprehensive way compared to the requests.

Among future applications, we can think of working on the parameters and give personality to the bot, make it more surly or more helpful, or to adapt to the user. We can also increase its learning ability by giving it the opportunity to learn new intents through the use of certain actions that expand its skills. Finally, we can also use this approach to evaluate and measure the skills of bots without self-learning.

References

1. BadgeBox, <https://www.badgebox.com/en/>.
2. A. Shawar, E. Atwell: Chatbots: are they really useful?. *Journal for Language Technology and Computational Linguistics*, vol. 22, no. 1. GSCL German Society for Computational Linguistics, 2007, pp. 29-49.

3. D. Jurafsky, H. James: *Speech and language processing an introduction to natural language processing, computational linguistics, and speech*. Pearson Education, 2000.
4. R. Collobert, J. Weston: A unified architecture for natural language processing: Deep neural networks with multitask learning, in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 160-167.
5. Y. LeCun, Y. Bengio, G. E. Hinton: Deep learning. in *Nature: International weekly journal of science*, vol. 521, no. 7553. Macmillan, 2015, pp. 436-444.
6. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In *Proceedings of the Workshop on Computational Learning Theory*, pp. 92-100, Madison, WI, 1998.
7. Nigam, K. Ghani, R.: Analyzing the effectiveness and applicability of co-training. In *Proceedings of Ninth International Conference on Information and Knowledge Management*, 2000, pp. 86-93.
8. Crook, J. Banasik, J.: Sample selection bias in credit scoring models. *International Conference on Credit Risk Modeling and Decisioning*, Philadelphia, PA, 2002.
9. Ghahramani, Z., Jordan, M.I.: *Learning from incomplete data*. Technical Report 108, MIT Center for Biological and Computational Learning, 1994.
10. Miller, D. Uyar, S.: A mixture of experts classifier with learning based on both labeled and unlabeled data. *Advances in Neural Information Processing Systems 9*, 1997, pp. 571-578, MIT Press.
11. Zhang, T., Oles, F.J.: A probability analysis on the value of unlabeled data for classification problems. In *Proceedings of Seventeenth International conference on Machine Learning*, 2000, pp 1191-1198, Stanford, CA.
12. Seeger, M.: *Learning with labeled and unlabeled data*. Technical report, Institute for ANC, Edinburgh, UK, 2000. <http://www.dai.ed.ac.uk/seeger/papers.html>.
13. Kremer, S., Stacey, D.. *NIPS 2001 Workshop and Competition on unlabeled data for supervised learning*, 2001. <http://q.cis.guelph.ca/skremer/NIPS2001/>.
14. Karakoulas, G., Salakhutdinov, R.: Semi-supervised Mixture of Experts Classification. In *Proceedings of the Fourth IEEE International Conference on Data Mining*, 2003, pp. 138-145, Brighton UK.
15. Goldman, S., Zhou, Y.: Enhancing supervised learning with unlabeled data. In *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, pp.327-334, San Francisco, CA.
16. Provost, F., Fawcett, T.: Robust classification for imprecise environments. *Machine Learning*, 42, 2001, 203-231.
17. Tomoya Sakai, Marthinus Christoffel du Plessis, Gang Niu, Masashi Sugiyama: *Semi-Supervised Classification Based on Classification from Positive and Unlabeled Data*.
18. S. George, S. Joseph: Text Classification by Augmenting Bag of Words (BOW) Representation with Co-occurrence Feature. *IOSR Journal of Computer Engineering (IOSR-JCE)* e-ISSN: 2278-0661, p- ISSN: 2278-8727 Volume 16, Issue 1, Ver. V (Jan. 2014), PP 34-38 www.iosrjournals.org
19. Goldberg Yoav: *Neural Network Methods in Natural Language Processing*. 1st edn. Morgan and Claypool publishers, 2017.
20. Daniel Jurafsky, James H. Martin: *Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 2nd edn. Pearson, 2008.
21. H. Zhang, D. Li: Naive Bayes Text Classifier. *Granular Computing*, 2007. GRC 2007. *IEEE International Conference on (2007)*.

22. Russo, Irene; Frontini, Francesca and Quochi, Valeria, 2016, OpeNER Sentiment Lexicon Italian - LMF, ILC-CNR for CLARIN-IT repository hosted at Institute for Computational Linguistics "A. Zampolli", National Research Council, in Pisa, <http://hdl.handle.net/20.500.11752/ILC-73>.
23. Paul Prasse, Christoph Sawade, Niels Landwehr, Tobias Scheffer: Learning to Identify Concise Regular Expressions that Describe Email Campaigns. *Journal of Machine Learning Research* 16 (2015) 3687-3720
24. Jugal Kalita, Marc Moreno Lopez: Deep Learning applied to NLP. [arXiv:1703.03091v1](https://arxiv.org/abs/1703.03091v1) [cs.CL] 9 Mar 2017.
25. Zhang, Y., Jin, R., Zhou, Z. H. (2010). Understanding bag-of-words model: A statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4), 43-52. DOI: 10.1007/s13042-010-0001-0
26. Jonathan J. Webster, Chunyu Kit: Tokenization as the initial phase in nlp. *Proceeding COLING '92 Proceedings of the 14th conference on Computational linguistics - Volume 4*, Pages 1106-1110.
27. Wilson, T. Wiebe, J. Hoffmann, P.: Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis, *Proceeding HLT '05 Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Pages 347-354.
28. Low pass filter, https://en.wikipedia.org/wiki/Low-pass_filter.