Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 190 (2021) 735–744

Procedia Computer Science

www.elsevier.com/locate/procedia

2020 Annual International Conference on Brain-Inspired Cognitive Architectures for Artificial Intelligence: Eleventh Annual Meeting of the BICA Society

Machine learning algorithms for teaching AI chat bots

Evgeny Tebenkov[1], Igor Prokhorov[1]*

National Research Nuclear University MEPhI
(Moscow Engineering Physics Institute), Moscow, Russia
tebenkov.ev@gmail.com, ivprokhorov@mephi.ru

Abstract

Machine learning is a method of data analysis, which allows the analytical system to learn in the course of solving many similar problems. Machine learning is based on the idea that analytical systems can learn how to identify patterns and make decisions with minimal human involvement. The history of already completed dialogues between users is used to train chat bots for automated communication with interlocutors. There are many machine learning algorithms, and this article describes the most popular of them and their use for teaching chat bots.

Peer-review under responsibility of the scientific committee of the 2020 Annual International Conference on Brain-Inspired Cognitive Architectures for Artificial Intelligence: Eleventh Annual Meeting of the BICA Society

Keywords: Machine learning, artificial intelligence, chat bots, algorithms of learning chat bots, learning algorithm, machine learning algorithm.

1. Introduction

Nowadays, chatbots have become widespread in messengers and social networks and due to the proliferation of chatbot platforms have become easy to create and use, while chatbots with artificial intelligence are still rare and not so common. The most common reasons why such chatbots are not popular are the complex and lengthy learning process, as well as imperfect algorithms for processing human requests, which reduces the effectiveness of chatbots and users' trust in them. Today, there are many machine learning algorithms that are used to create chatbots with artificial intelligence, all of which have their own strengths and weaknesses and are adapted to solve various problems. Within the framework of this article, a comparison of machine learning methods from a theoretical point

* Corresponding author. Tel.: +7-495-788-56-99; fax: +7-495-788-56-99.
E-mail address: ivprokhorov@mephi.ru

1877-0509 © 2021 The Authors. Published by Elsevier B.V.

Peer-review under responsibility of the scientific committee of the 2020 Annual International Conference on Brain-Inspired Cognitive Architectures for Artificial Intelligence: Eleventh Annual Meeting of the BICA Society
10.1016/j.procs.2021.06.086

of view is given and an example of the implementation of a chatbot platform using NLU is given, as well as a methodology for training a chatbot with users using the example of the Bayesian algorithm is described and the results of its testing are given.

2. Statement of problem

As noted earlier, chatbots with artificial intelligence are not popular due to the complexity of their implementation and imperfect machine learning algorithms, which lead to the fact that chatbots cannot fully communicate in human language. In order for the chatbot to efficiently perform its tasks, it is necessary that the platform provides a quick and accurate response to the user's request, this can only be achieved if a microservice architecture is used and the speed of message processing and preparation of responses by the chatbot will not change depending on the load on the server and the number of incoming messages. If these conditions are met, then it will be possible to test the effectiveness of chatbots with NLU and get constructive feedback from chatbot users.

3. Machine learning in NLP

NLP (Natural Language Processing) and machine learning are computer science fields related to AI (Artificial Intelligence). Machine learning can be used in many different fields. NLP takes care of "understanding" the natural language of the person with whom a program (e.g. chatboot) is trying to communicate. This understanding allows a program (e.g. chat-bot) to both interpret input data and produce it in human language. The machine "learns" and uses its algorithms through controlled and uncontrolled learning [1]. Supervised learning means teaching a machine to convert input data into a desired output value. In other words, it assigns the output function to the data so that newer data examples give the same output result for this "learned" interpretation. Uncontrolled learning means finding new patterns in the data without prior information and learning. The machine assigns the output function to the data itself through careful analysis and extrapolation of patterns from raw data. The levels are designed for hierarchical data analysis. With the help of hidden layers, the function can be retrieved by training under supervision or without supervision. Hidden layers are part of the data processing levels in the neural network [2]. Machine learning is used in various tools for solving applied tasks, such tools may include chat bots, RPA, video and audio analytics tools [4].

4. Algorithms grouped by learning style

There are various ways that an algorithm can simulate a problem based on its interaction with experience or the environment, or on what we want to call the input data. In machine learning and artificial intelligence tutorials, it is popular to first consider the learning styles that an algorithm can adopt. There are only a few basic learning styles or models that an algorithm can use, and we will look at them here with a few examples of algorithms and the types of problems they suit. This taxonomy, or the way machine learning algorithms are organized, is useful because it helps you to define the input roles and, while preparing the model, select the one that best suits your task to get the best result [3] (Figs. 1-3).

4.1. Supervised learning Algorithms

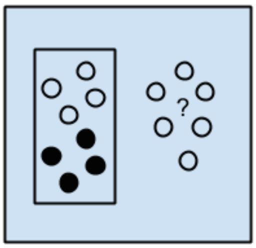

Fig. 1. Supervised learning Algorithms.

Input data on controlled learning algorithms is called learning data and has a known label or result such as spam/not spam or one day's stock price. The model is prepared in the learning process where the forecasts need to be made and adjusted when the forecasts are erroneous. The learning process continues until the model reaches the desired level of accuracy of the learning data. Examples of such tasks are classification and regression. Examples of algorithms include: Logistic regression and neural network reverse propagation [5].

4.2. Unsupervised learning Algorithms

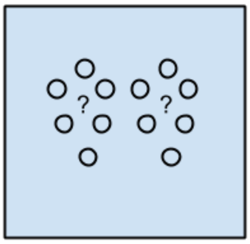

Fig. 2. Unsupervised learning Algorithms.

Input data are not marked and have no known result.The model is prepared by subtracting the structures present in the input data. This can be a general rule extraction. It may be through a mathematical process to systematically reduce redundancy, or it may be to organize the data in a similar way. Examples of problems are clustering, dimensional reduction and the study of association rules. Examples of algorithms include the A priori and K-Mins algorithms [5].

4.3. Semi-supervised learning Algorithms

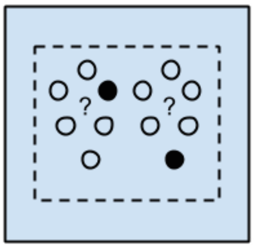

Fig. 3. Semi-supervised learning Algorithms.

Input data are a mixture of marked and unmarked examples. There is a desired problem of forecasting, but the model must examine the structures for organizing the data as well as make forecasts. Examples of problems are classification and regression. Examples of algorithms are extensions to other flexible methods that make assumptions about how to model marked and unmarked data [5].

5. Machine learning algorithms

5.1. Neural networks

Neural networks are one of the learning algorithms used in machine learning. They consist of different layers to analyze and study data.

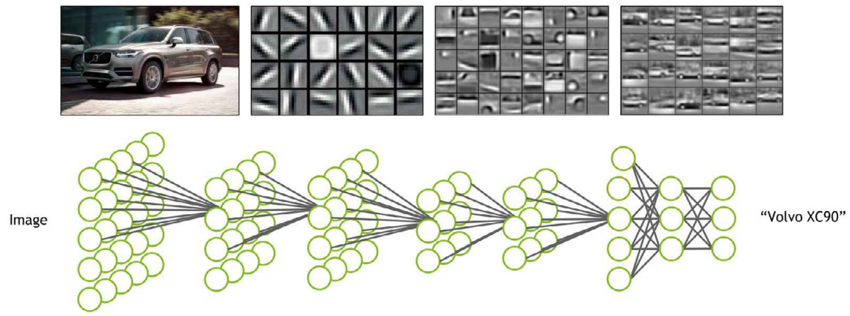

Fig. 4. Algorithm of neural networks operation.

Each hidden layer tries to detect patterns in the image. When the pattern is detected, the next hidden layer is activated, and so on. The Volvo XC90 image above perfectly illustrates this. The first layer detects the edges. Then the next layers combine the other edges found in the data and eventually the specified layer tries to detect the wheel pattern or window pattern. Depending on the number of layers, it will or will not be able to determine what is in the picture, in this case the car. The more layers in the neural network, the more it will know and the more accurate the detection of the picture will be. Neural networks study and assign weights to connections between different neurons each time the network processes data. This means that the next time it encounters such a picture it will know that this particular part of the picture is probably related to, for example, a bus or door [6].

5.2. Decision tree algorithms

In this algorithm, the solution tree is used to display solutions and their possible consequences, including chances, costs and utilities. This method allows you to approach the problem logically and step by step to come to the right conclusion. An important algorithm developed based on this algorithm is the Random tree algorithm. This algorithm uses several trees to avoid the overflow that often occurs when using solution trees.

5.3. Bayesian algorithms

Applies the Bayesian theorem for regression and classification problems related to probability. It tries to show the probabilistic relationship between different variables and determine, taking into account the variables, which category it most likely belongs to.

5.4. Regression algorithms

Well suited for statistical machine learning, regressions seek to simulate the relationships between variables. By observing these relationships, you seek to establish a function that more or less simulates these relationships. This means that when you observe more variables, you can tell with some confidence and with some margin of error where they may be along the function.

5.5. Support Vector

The vector support algorithm is used at grouping of points on a dimensional plane. Grouping is performed by creating a hyperplane that separates groups with the widest possible field. This helps at classification and is used, for example, in advertising or splicing of human RNA.

5.6. Methods of grouping

Ensemble methods combine various weaker controlled learning algorithms. Combination of very different models usually gives better results. By combining different methods, you can process displacements with specific models, reduce dispersion, and reduce re-fitting by averaging more.

5.7. Clustering algorithms

The main purpose of this algorithm is to combine available data into groups where data points in such a group are more similar to each other than data points in other groups. The most important clustering methods are hierarchical, centroid, distribution and density.

5.8. Algorithms of learning the rules of association

These are the rules that can be set between sets of elements and transactions for these elements and sets of elements. The relationship between X and Y, which means that when you get X, you also get Y. This rule is located in the database, watching the element sets and the elements contained within them. This is the most common algorithm for learning chat bots with artificial intelligence.

5.9. Artificial neural network algorithms

The algorithms of artificial neural networks are inspired by the human brain. Artificial neurons are interconnected and communicate with each other. Each connection is weighted by previous learning events and more learning occurs with each new data input. There are many different algorithms associated with artificial neural networks, and one of the most important is deep learning. An example of deep learning can be seen in Fig. 4 above. This is especially true for building much larger complex neural networks.

5.10. Algorithms of dimensional analysis

Dimension is the number of variables in data and the size to which they belong. This type of analysis is aimed at reducing the number of dimensions with corresponding variables while preserving the same information. In other words, it strives to remove less important data and at the same time to provide the same end result.

6. Chat Bot Platform Architecture

Architecture of the chat-bot platform due to its specifics should have the ability to quickly scale. The interface used by chat bots is based on the structure of human conversation resulting from the processing of natural language [7]. Therefore the load and number of requests to the platform are not evenly distributed and are not regulated in any way, and the performance of the platform depends on the performance of the chat bots. To solve this problem, the platform uses a microservice architecture with containers for chat bots, which allows working in parallel with an unlimited number of chat bots without creating queues from messages being processed and thus not slowing down their work.

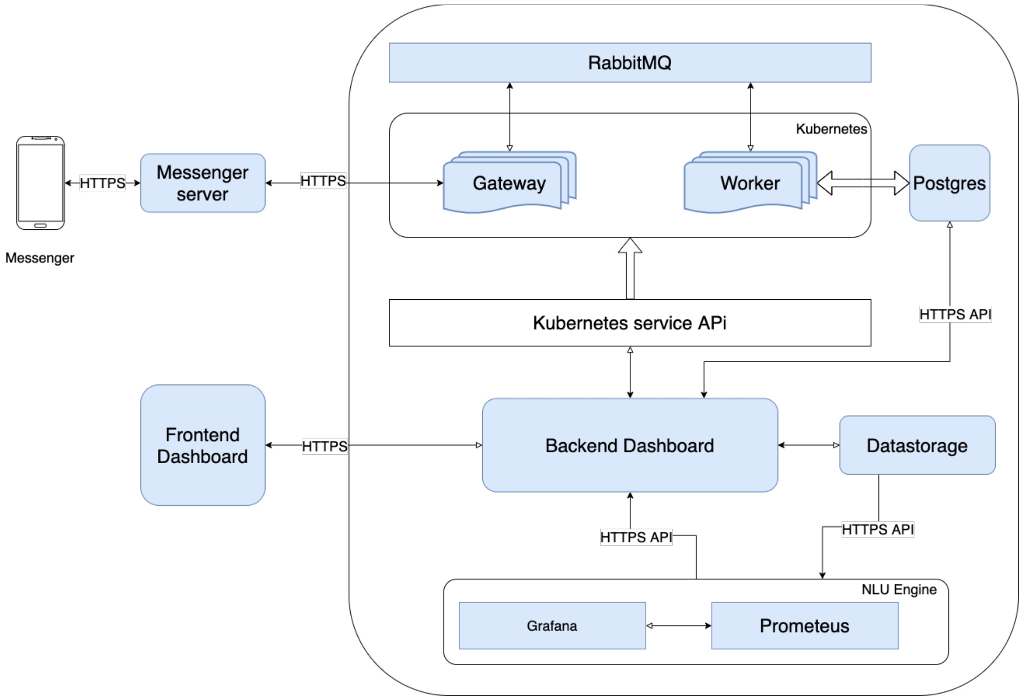

Fig. 5. IT architecture of the chat-bot platform.

Description of elements of the chat-bot platform (Fig. 5):

- RabbitMQ is a software message broker based on the AMQP standard - replicable binding software, focused on message processing;
- Messenger - a messenger with which a user interacts, such as Facebook or Telegram;
- Messenger Server - a server of the platform of the used messenger, which processes WebHooks between messenger and chat-bot platform;
- Kubernetis service API - API that is used for message routing and defines logical container grouping that allows to define container pools and load balancing between them;
- Kubernetes - open software for automation of deployment, scaling and management of containerized applications. Each container stores the code for the chat bot, Worker and Gateway. The peculiarity is that when deploying another chat bot, a separate container is created for it, which allows you to easily create an unlimited number of chat bots without complicating the chat platform and run them in parallel;
- Worker - processes incoming messages from users and sends a message to the RabbitMQ queue;
- Gateway - by analogy with Worker, it routes outgoing chat-bot messages;
- Data storage - A storage of data from chat bots that is used to store dialogs and media files that chat bots deployed on the platform use to generate their responses;
- Backend Dashboard - Server part of the personal cabinet and administrative panel of the platform's chat bots;
- Frontend Dashboard - web part of the personal cabinet and the administrative panel of the chat-bot platform;
- NLU Engine - is the main component that interprets what users say at any time, and converts the language into structured intuitions that the system can process in the future. The NLP Engine contains advanced machine learning algorithms to determine the user's intentions and then compare them with a list of available intentions supported by the chat bot;
- Postgres – Database management system.

7. Machine learning algorithms in NLP Engine

One way to extract meaning from natural language is to determine the function of the text/sentence (e.g. is this a question, suggestion, offer, or command); this is called dialogue act recognition. In dialogue act recognition systems, a corpus of sentences (training data) is labeled with the function of the sentence, and a statistical machine learning model is built which takes in a sentence and outputs its function. The model uses a number of different features to classify the sentences including: (i) words and phrases such as "please" (function=request) and "are you" (function=yes/no question), and (ii) syntactic and semantic information [8].

Table 1. Dialogue Act Recognition Example.

Speaker	Dialogue Act	English
A	Conventional-opening	Hallo?
B	Conventional-opening	Hi Peter!
B	Statement	It's me, Michael.
B	Question	How are you?
A	Conventional-opening	Hello Michael!
A	Statement	Very well.
A	Question	And you?
B	Statement	I'm well too.

The first-step in creating a dialogue act recognition system, is defining the relevant functions or the DA tag-set. This involves choosing labels that are general enough to be re-used in multiple tasks, specific enough to remain relevant for the target task, and clear/separable enough that there is little confusion for humans in labeling the functions of sentences in the training set. A number of tag-sets have gained prominence and are the most frequently used in chatbots: Dialogue Act Markup in Several Layers (DAMSL), Switchboard SWBD-DAMSL, Meeting Recorder, VERBMOBIL, and Map-Task [9].

The DAMSL annotation scheme labels a sentence in four dimensions: communicative status, information level, forward-looking function, and backward-looking function. Communicative status labels a sentence as: (i) uninterpretable, (ii) abandoned, or (iii) self-talk. Information level labels a sentence as: (i) task, (ii) task management, (iii) communication-management, or other. Forward-looking functions encode any information that will affect future conversation and label sentences along eight sub-dimensions: (i) statement – assert, reassert, or other, (ii) influencing- addressee-future-action – open-option or action directive, (iii) info-request, (iv) committing-speaker-future-action – offer or commitment, (v) conventional opening-closing, (vi) explicit performative, (vii) exclamation, or (viii) other. Backward-looking functions encode the relationship between current and previous speech, such as (i) agreement, (ii) understanding, (iii) answer, or (iv) information relation [10].

Switchboard is an adaptation of DAMSL for automated telephone conversations [11]. Meeting Recorder DA (MRDA) is similar to Switchboard: its training Corpus, however, is labeled recordings of 72 hours of conversations from meetings. MRDA handles intricacies well given the complications that often occur during meetings such as speaker overlap, frequency of abandoned comments, and complicated turn-taking interactions [12]. Finally, Map Task is a hierarchy of levels: the first level labels transactions, the second is conversational games, which labels patterns like question and answer pairs, and the third includes 19 conversational moves [13].

8. Bayesian Approaches to DA Models

The idea behind using a Bayesian approach to DA models is to find the probability of every possible sequence of dialogue acts DA that could represent a sentence or utterance (U), and find the dialogue act sequence DAmax with the highest probability of occurring [8].

$$DA_{max} = argmax_{DA}P(DA|U) = argmax_{C}\frac{P(DA)P(U\}DA)}{P(U)} \quad (1)$$

$$DA_{max} = argmax_{DA}P(DA) * P(U|DA) \quad (2)$$

Assuming the N individual words of the utterance are known, and the Naïve Bayes assumptions of independence of subsequent words holds, this leads to:

$$DA_{max} = argmax_{DA}P(DA) * \prod_{i=1..N} P(W_i|DA) \quad (3)$$

This is the unigram model, where P(DA) and $P(W_i|DA)$ can be quantified from empirical data. Using this Naïve Bayes classifier, Reithinger et. al. finds a 74% recognition/accuracy rate for classifying the correct dialogue act from a given sentence [14]. N-gram models are frequently used to include dialogue history in the model. These models estimate P (DA | N historical DAs) rather than P(DA). Assuming N = 3, we would estimate $P(\mathrm{DA}|DA_{n-1}, DA_{n-2}, DA_{n-3})$ [15]. Hidden Markov Models can also be used to model dialogue history, where each state represents a dialogue act in the conversation history of the chatbot [16]. Likewise, neural network classifiers can be trained as well. A combination of HMMs and neural networks has achieved 76% accuracy [17].

9. Testing chat bots with users

To test the effectiveness of using the methods of machine learning in a chat-bot with users, a chat-bot (Fig. 6) was developed and trained to communicate on the topic of volunteerism in Russian.

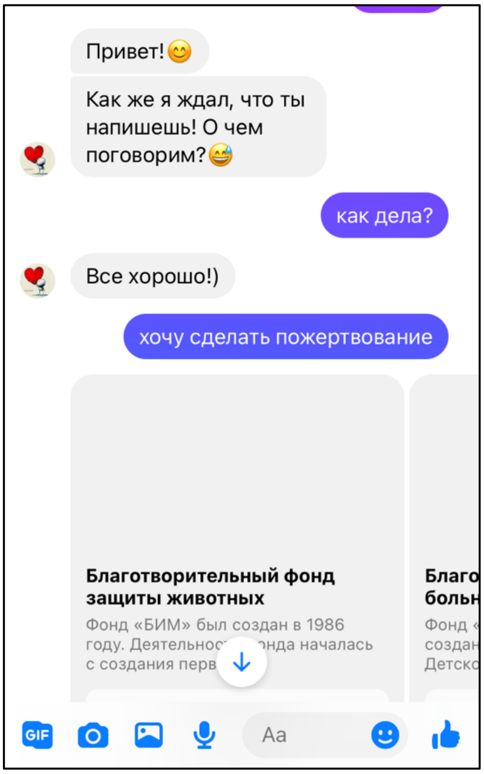

Fig. 6. Example of a chat-bot dialog with a user in Facebook messenger.

It's task was to share information about volunteer projects with the interlocutor in the form of a free dialogue and invite the interlocutor to take part in one of them. The chat-bot's task was considered completed if the user agreed to take part in the event and left his contact information, while the users communicating with the chat-bot did not know any information that the chat-bot owned and therefore could not arrange communication with the chat-bot in such a way as to simplify or otherwise complicate his task.

A focus group of 20 people was gathered to test the effectiveness of the chat-bot. As a result of communicating with the chat-bot, 12 (60%) of them were satisfied with the answers of the chat-bot and agreed to participate in the activities, 5 (25%) of users were satisfied with communicating with the chat-bot, but refused to participate in volunteer activities, and another 3 (15%) of users refused to continue communicating with the chat-bot after he was unable to answer their questions.

10. Conclusion

The article considered 10 machine learning algorithms that can be described for training chat bots, disclosed the IT architecture of a chat bot platform using the NLU Engine learning Bayesian method and the results of testing this chatbot with real users are given. Each of the algorithms described in the article has its advantages and disadvantages, but if used correctly, they allow you to complete any task assigned to the chatbot. The development of a chatbot platform with a microservice architecture and the use of NLU allows you to test various algorithms and test their effectiveness when interacting with people. 75% of users were satisfied with chatbot communication with positive feedback. As part of the continuation of the research and refinement on the platform of choosing the learning algorithm, the effectiveness of the bot, depending on the problem being solved, which will increase the efficiency of interaction with the user and his confidence in him.

References

[1] Word2vec: // URL: https://habr.com/ru/post/446530/ (Date of the application: 15.09.2020)

[2] Top 12 Benefits of Chatbots: Comprehensive Guide [2020 update] // URL: https://blog.aimultiple.com/chatbot-benefits/ (Date of the application: 13.09.2020)

[3] How chatbots can help reduce customer service costs by 30% // URL: https://www.ibm.com/blogs/watson/2017/10/how-chatbots-reduce-customer-service-costs-by-30-percent/ (Date of the application: 13.09.2020)

[4] Kanakov Fedor, Prokhorov Igor, Research and development of software robots for automating business processes of a commercial bank, // Postproceedings of the 10th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2019 (Tenth Annual Meeting of the BICA Society), Procedia Computer Science,2020

[5] A Tour of Machine Learning Algorithms // URL: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/ (Date of the application: 15.09.2020)

[6] «12 Algorithms Every Data Scientist Should Know» // URL: https://datafloq.com/read/12-algorithms-every-data-scientist-should-know/2024#datascience (Date of the application: 16.09.2020)

[7] William Villegas-Ch, Adrián Arias-Navarrete,Xavier Palacios-Pacheco, Proposal of an Architecture for the Integration of a Chatbot with Artificial Intelligence in a Smart Campus for the Improvement of Learning, // Sustainability 2020, 12(4), 1500. https://doi.org/10.3390/su12041500

[8] McTear, M., Callejas, Z., & Griol, D. (2016). The conversational interface: talking to smart devices. Cham: Springer.

[9] Kral, P., & Cerisara, C. (2010). Dialogue Act Recognition Approaches. Computing and Informatics, 29, 227–250. http://doi.org/10.1109/mlsp.2008.4685529

[10] Draft of DAMSL: Dialog Act Markup in Several Layers. Retrieved from http://www.cs.rochester.edu/research/cisd/resources/damsl/RevisedManual/ (Date of the application: 20.09.2020)

[11] WS-97 Switchboard DAMSL Coders Manual. Retrieved from https://web.stanford.edu/~jurafsky/ws97/manual.august1.html

[12] Dhillon, R., Bhagat, S., Carvey, H., & Shriberg, E. (2004, February 9). Meeting Recorder Project: Dialog Act Labeling Guide. Retrieved from http://www1.icsi.berkeley.edu/ftp/pub/speech/papers/MRDA-manual.pdf

[13] Carletta, J., Isard, S., Doherty-Sneddon, G., Isard, A., Kowtko, J. C., & Anderson, A. H. (1997). The Reliability of a Dialogue Structure Coding Scheme . Association for Computational Linguistics, 23(1), 13–31. Retrieved from http://www.aclweb.org/anthology/J97-1002

[14] Reithinger, N., & Klesen, M. (1997). Dialogue Act Classification Using Language Models. In Proceedings of EuroSpeech, 2235–2238.

[15] Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., ... Meteer, M. (2000). Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. Computational Linguistics, 26(3), 339–373. http://doi.org/10.1162/089120100561737

[16] Stolcke, A., & Shriberg, E. (1998). Dialog Act Modeling for Conversational Speech. AAAI Technical Report SS-98-01, 98–105. Retrieved from https://pdfs.semanticscholar.org/d7de/4acec556524a8ab2b00aa4414768bc1617e9.pdf

[17] Ries, K. (1999). HMM and neural network based speech act detection. 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258). http://doi.org/10.1109/icassp.1999.758171