Knowledge Retention Through Observation of Instant Messaging Systems

João Costa
jmsfc.costa@estgv.ipv.pt
Polytechnic Institute of Viseu
Viseu, Portugal

Carlos Cunha
cacunha@estgv.ipv.pt
Polytechnic Institute of Viseu
Viseu, Portugal

Rui P. Duarte
pduarte@estgv.ipv.pt
Polytechnic Institute of Viseu
Viseu, Portugal

João Menoita
joaohenriques@estgv.ipv.pt
Polytechnic Institute of Viseu
Viseu, Portugal

ABSTRACT

Knowledge is the most valuable asset in today's organizations. Since it offers an unbeatable competitive advantage, valuable knowledge demands strict management principles to avoid being lost. Instant messengers provide an opportunity to gather knowledge passed through individuals in the organization. By modeling that knowledge using machine learning techniques, it becomes possible to retain and make it ubiquitous throughout the organization. This paper presents a solution for gathering, modeling, and retrieving knowledge associated with the technical support in organizations, using machine learning algorithms. The solution comprises the architecture, data preparation techniques and machine learning algorithms. The experimental evaluation exhibits the algorithms with better performance for this class of problems.

CCS CONCEPTS

• **Computing methodologies → Information extraction**; **Multi-agent systems**; **Supervised learning by classification**; • **Applied computing → Service-oriented architectures**.

KEYWORDS

knowledge management, bots, machine learning

ACM Reference Format:
João Costa, Carlos Cunha, Rui P. Duarte, and João Menoita. 2019. Knowledge Retention Through Observation of Instant Messaging Systems. In *23rd Pan-Hellenic Conference on Informatics (PCI '19), November 28–30, 2019, Nicosia, Cyprus.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3368640.3368656

1 INTRODUCTION

Knowledge management represents a big concern in organizations of all sizes [1][2][3]. Notwithstanding the available documentation created and made possible by information systems of today's companies, most of the knowledge is transmitted through the direct interaction between collaborators. This process may not only be inefficient and error-prone but can also lead to migration of expertise towards the border of the company along with employees, when they leave the organization [4].

Capturing the knowledge kept by employees is desirable to avoid losing it. Commonly, that knowledge is transferred informally between employees, which difficult its acquisition and storage. The emergence of new communication channels like instant messaging systems between organization collaborators has created new opportunities for the retrieval of that knowledge. On top of that, if we combine the information transmitted through these channels with machine learning techniques, we obtain a new world of solutions for the problem mentioned above.

Conversation Agents (CA) represent a solution for replacing human knowledge owners in organizations. It integrates knowledge query technology into the instant messaging environment [5][6]. Recent evolution of natural language [7] and artificial intelligence [8] has leveraged the potential of CA [9], enabling its application to several distinct domains, such as psychological counseling [10], customer service support [11] and even sexual teenagers' counseling [12]. CA has several underneath motivations, such the increase in productivity, entertainment, and enhancement of social experiences [13]. Therefore, human-human collaboration has much to gain with the inclusion of CA in the collaborative communication process.

This paper presents a CA solution for acquisition and exploitation of enterprise knowledge, more precisely, technical support knowledge. It is responsible for gathering, processing, and modeling knowledge encoded in messages transmitted by employees of the technical support of one of the biggest Portuguese companies. The solution includes:

- an instant messaging plugin that can be incorporated into a broad number of applications developed in the company;
- a web service implementing the machine learning activities, as data gathering, data cleaning, model training, and classification.

The CA implemented in our solution promotes the learning of questions and answers based on the feedback of technical support staff.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
PCI '19, November 28–30, 2019, Nicosia, Cyprus
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-7292-3/19/11…$15.00
https://doi.org/10.1145/3368640.3368656

This paper presents the design and methodology implemented by our CA solution and addresses the following research questions:

- Which machine learning algorithms perform better choosing the adequate answers to questions provided by technical staff?
- Which data preparation techniques reduce the error of classification models?

By answering the previous research questions, we validate the choice of algorithms and the data preparation methodology.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work. Section 3 identifies the problem to be solved in this work. Section 4 describes the supervised machine learning algorithms studied in the solution presented in this paper. Section 5 details the architecture, methodology and tools adopted. In Section 6 experimental validation of our work is carried out. Section 7 concludes the paper, providing some hints to future work.

2 RELATED WORK

Conversation agents have been gaining popularity in the last years in several domains. Briggs et al. [14] proposed a six-layer model of collaborative work practices in the design and use of CAs. Bittner et al. [15] improved on their work and presented a taxonomy for the design of CA facilitators, peers, and experts. To infer the importance of human-chatbot communication, the authors of [16] carried out a comparative study between human-human online conversations and human-chatbot conversations. They compared 100 instant messaging conversations to 100 exchanges with the Cleverbot chatbot. Results show that they communicated with the chatbot for longer time-periods than they did with another human. They stated that this difference is due to the attempt of humans to adapt to chatbot conversations. In [5] the authors demonstrated the importance of multi-domain knowledge bases on building conversation bots. They propose several techniques to retrieve and dynamically construct web knowledge from semi-structured data. The experimental work is based on the MSN Messenger application running on top of knowledge bases.

An increasing number of companies are trying to integrate conversation agents for the automation of specific processes related to education, information retrieval, business, e-commerce, and amusement [17]. In the enterprise domain, bots enable new interaction forms within companies and between companies and external agents. Xu et al. [18] explored Long Short-Term Memory (LSTM) networks to generate responses for customer-service requests. They trained their system with nearly one million Twitter conversations between users and agents from over 60 brands. Their results show that over 40% of the requests are emotional, and the system is as emotional as human agents and shows empathy to help users with emotional situations. Based on collaborative discourse theory [19], Rich et al. [20] used reusable behavioral components to implement COLLAGEN, a collaborative interface agent between two participants. Agent responses are determined from collaborative states that are updated after the interpretation of new events. They illustrated the use of *plug-ins* to produce responses in a complex tutoring system. COBOTS [21] is a framework to automate the process of guided troubleshooting. From the question, it uses IBM Watson's parsing and semantic analysis for extracting the symptom, intent, and entity. Entities are components of concern that roots their Knowledge Graph on their specific domains. By crossing bots operating in different domains, the framework was able to increase the accuracy obtained when bots are used independently.

Conversation agents are commonly seen in the context of human-agent communication. However, communication between agents may add value to CA solutions. Lieberman [22] focused on the importance of having agents that are simultaneously interface agents and autonomous agents. Allen et al. [23] proposed a collaborative problem-solving model between agents and, in particular, between human and software agents. Agents collaborate between themselves to select recipes to reach predefined objectives. They applied their model to implemented systems [24] by including as many elements as possible from formal models of collaboration.

There are other related implementations of CA agents. Uthus and Aha [25] stressed the importance of learning the design of CAs that can participate in chat conversations between video game players. Letizia [26] is a CA used for assisting the user browsing the web in real-time. As the user navigates the page, Letizia analyses the page and displays recommendations for the user. In his work, he also established design principles for autonomous interface agents.

Notwithstanding the diversity of studies addressing the implementation of CA agents, the choice of algorithms, methods and architectures is dependent on the problem and its respective knowledge base domain. This paper presents the architecture, the data preparation methods and the selection of machine learning algorithms to build a technical support agent that retains knowledge and answers questions related to its domain.

3 PROBLEM STATEMENT

This paper addresses the problem of learning the association of technical issues written textually in a messaging system of a company with the most appropriate answers. The model resulting from the learning process represents the knowledge used to answer similar questions in the future.

The technical knowledge is represented through the mapping of questions to answers $k : Q \longrightarrow A$, being Q and A word sets. As usually one answer $a \in A$ can be given to several questions $O \subset Q$, it is expected $|Q| \gg |A|$.

During the learning process, the machine learning model M is trained to associate the combination of words W of $q \in Q$ to the correct answer a.

For classification of new questions, the model M provides the most likely answer a for each question q. Data preparation may be required to normalize words in terms of meaning (i.e., create a single representation for words with the same meaning) and representation.

We assume that words are written without grammatical errors. This assumption is justified by the inclusion of a grammar assistant into the messaging application. As well, the language used internally by employees in their professional context avoids slang and nonstandard expressions.

4 MACHINE LEARNING ALGORITHMS

Machine learning classification algorithms can be classified into supervised learning and unsupervised learning. Supervised learning

algorithms require an oracle to provide the classification outcome during the learning process. On the other side, unsupervised learning algorithms do not assume that the classification outcome is known.

The answer to each problem is known at the beginning of the learning process. Thus, supervised learning algorithms are adequate for our problem. The choice of the algorithms was also based on the characteristics of the available training data in organizations. In particular, the amount of data available for this type of problems may not be enough to train deep learning models. Deep learning methods require large amounts of data that cannot be provided by instant messengers when used for technical support, due to the small number of cases associated with each problem. For that reason, we excluded deep learning methods from our analysis.

4.1 Support Vector Machines

Support vector machines (SVMs) [27] are a set of supervised learning methods effective in high dimensional spaces. These methods see data points as p-dimensional vectors, while classes are separated by $(p-1)$-dimensional hyperplanes.

SVMs select a small number of critical boundary samples called *support vectors* from each class that separate classes as widely as possible. The maximum margin hyperplane that separates two classes is represented as in (1), where i reference each support vector, a the test instance, $a(i)$ the support vectors, and finally b and α_i are parameters that determine the hyperplane that separates two classes.

$$x = b + \sum_i \alpha_i Y_i a(i) \cdot a \tag{1}$$

Examples of SVMs applied to text classification can be found in [28] and [29].

4.2 Multinomial Naïve Bayes

Multinomial Naïve Bayes (MNB) is one of the most popular applications of machine learning for the analysis of text data. In this algorithm, feature vectors represent the frequencies with which certain events have been generated by a multinomial distribution.

$$p(\mathbf{x} \mid C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i {p_{ki}}^{x_i} \tag{2}$$

A feature vector $\mathbf{x} = (x_1, \ldots, x_n)$ is then a histogram, with x_i counting the number of times event i was observed in an instance. When applied to text classification, an event represents the occurrence of a word in the text. Examples of the application of MNB to text classification can be found in [30], [31] and [32].

4.3 Logistic Regression

Logistic Regression has been used to model the probability of a particular class represented by a categorical dependent variable.

Equation 3 presents the linear relationship between the predictor variables, being b the base of the logarithm, x_i the predictors, and β_i the parameters of the model.

$$\ell = \log_b \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \tag{3}$$

4.4 Ridge Classifier

Ridge regression is an extension for linear regression. It is a regularized linear regression model that uses a parameter γ for the regularization of each coefficient, with the purpose of avoiding overfitting. This method works well with high dimensional problems because they are likely to be linearly separable and so with text classification problems.

4.5 Decision Tree Classifier

A Decision Tree provides a simple tree representation for classifying examples that is interpretable and can be visualised. A node in a decision tree represents an instance, the outcomes of the test are represented by branch, and the leaf node represents the class label. Decision trees algorithms create trees by continuously splitting data according to a particular parameter.

4.6 Random Forest Classifier

Ensembles group several models that are generated and combined to solve a particular problem. By including several models into the classification process, it is expected improvement over the performance provided by a single model and a reduction of the likelihood of an unfortunate selection of a poor one.

The Random Forest classifier consists of an ensemble of a large number of individual decision trees. Each tree in the random forest spits out a class prediction. The model's prediction is the class with the most votes. Ensemble predictions are more accurate than individual decision trees. The reason is that the trees protect each other from their errors.

4.7 Gradient Boosting Classifier

Gradient Boosting is another prediction model commonly used to build an ensemble of decision trees. This classifier uses gradients in the loss function:

$$y = ax + b + error \tag{4}$$

The loss function indicates how good the model's coefficients are at fitting the underlying data.

5 APPROACH

This section presents the architecture, methodology and tools adopted by our approach.

5.1 Architecture

Figure 1 presents the solution architecture, comprising the: (1) development of a messaging plugin for being incorporated in all applications developed in the company to be used by senders and receivers of messages; and (2) design and implementation of services for learning and classification of answers.

Users interact with the messaging plugin in two different ways. They first issue questions that are answered by the answer prediction service. Further, they provide feedback about the correctness of the answer.

Questions with incorrect answers are assigned to the person responsible for providing technical support, in order to be answered. The manual answer is further interpreted as the correct class for the

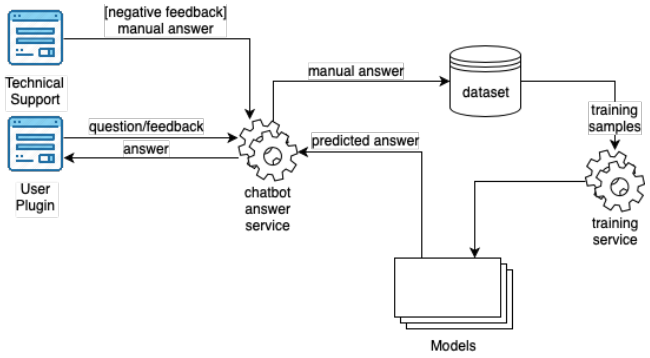

Figure 1: Architecture of the solution.

question issued by the user, in case the user give proper feedback to it. Otherwise, the technical support person should provide another answer until a positive feedback condition occurs.

When a manual answer receives positive feedback from the user, it is stored in the dataset used to train classification models. This event triggers training of new models with the newly added sample.

5.2 Learning and Classification Methodology

Figure 2 presents the methodology followed by answer prediction and training activities. Each question issued by the user starts by being pre-processed into several stages:

(1) **Tokenization** of question's text, to obtain a list of words belonging to it;
(2) **Noise filtering** the list of words, to remove words with low discriminatory power (e.g., *the* and *a*) obtained from an existing list;
(3) **Stemming** words to reduce words from the same family to a common root (e.g., *cats* to *cat*).

After the pre-processing stage, the list of words are coded into a two-stage sequence:

(1) **Vectorization** of the list of words to obtain a matrix representation in the vector space model [33]. Machine learning algorithms use the resulting structure for training and classification.
(2) **Feature weighting** providing higher weights to features (i.e., words) associated with rare events. That means that uncommon words specific to a document or small set of documents receive higher weights than words appearing in most documents.

Model training resorts to the combination of encoded-words and respective TFIDF (Term Frequency Times Inverse Document Frequency) [34] weights with the class associated with the correct answer.

5.3 Tools

We adopted several tools in our solution:

- the implementation of machine learning algorithms is provided by the popular Scikit-learn Python library [35];

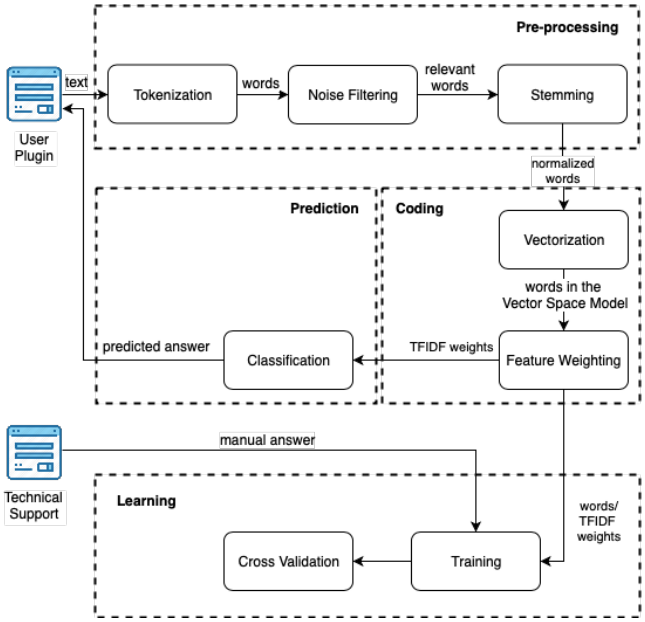

Figure 2: Methodology followed by the classification and learning activities.

- the Natural Language Toolkit [36] contains the functionality required for natural language processing. It includes text tokenization, noise filtering, and word stemming;
- the Pandas library [37] supports the conversion between dataset sources and data frames structures required for analysis;
- the Joblib tool [38] accommodates persistence of models created by machine learning algorithms;
- the Flask framework [39] is a lightweight web application that provides transparent communication between the interface and services by encapsulating RESTful dispatching of requests.

The client plugin was implemented using the Bootstrap framework and the JQuery language. The plugin communicates with services through HTTP Requests;

6 EXPERIMENTAL WORK

Experimental validation of our work unveils the ability of machine learning models to retain the technical support's knowledge in instant messaging systems.

We decided to use several publicly available datasets to evaluate the models' performance, namely:

- **DBPedia Ontology Classification** [40] is an encyclopedic dataset of structured information holding a multi-domain ontology which has been derived from Wikipedia. A subset of this dataset with 35 thousand abstracts out of the 38 million present in the original dataset was chosen for validation. This subset contains 14 different classes.
- **20 Newsgroup Text Dataset** [41] contains 18000 newsgroups posts on 20 topics (classes) split in two subsets: one for training and another for testing.

The choice of datasets is justified by their independence from any specific technical knowledge database. Due to their higher entropy relatively to common questions, we expect worst performance results with those datasets than with organic datasets gathered using real technical knowledge. Entropy is defined as the percentage of words with low discriminative power, increasing the dimensionality of samples. The higher the entropy, the larger the set of words that can compromise the semantic separation of questions between classes.

We run algorithms over both datasets and performed ten-fold cross-validation [42] to evaluate algorithms' performance. Figure 3 presents the rate of correct predictions of each algorithm used to train models. It is noticeable the higher performance of support vector machines (LSVC) in both datasets. The overfitting protection intrinsic to these classifiers, allowing them to handle large feature spaces, constitutes a common explanation for their good performance in text classification [27].

Algorithms	DBPedia	Newsgroups
Multinomial Naive Bayes *(MNB)*	0.942	0.851
Logistic Regression *(LR)*	0.958	0.895
Rigged classifier *(RC)*	0.962	0.769
SGD classifier *(SGDC)*	0.920	0.891
Decision Tree Classifier *(DTC)*	0.883	0.640
Random Forest classifier *(RFC)*	0.900	0.658
Gradient Boosting classifier *(GBC)*	0.936	0.833
Linear SVC *(LSVC)*	0.966	0.929

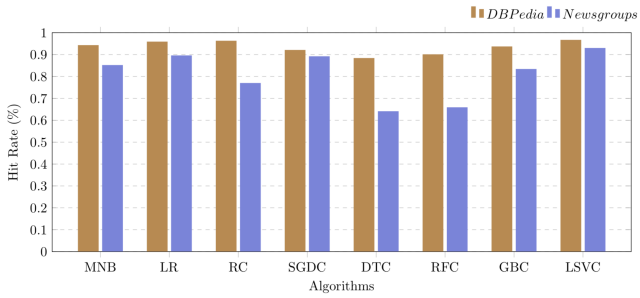

Figure 3: Rate of correct predictions of each algorithm.

The Newsgroups dataset exhibits lower performance than the DBPedia dataset. Since the latter contains abstracts with less irrelevant words for the context, it shows better performance than the former. From the experimental results, we can determine that machine learning algorithms can be valuable assets to retain technical knowledge in organizations. Intuitively, the main expected limitation of these algorithms is the discriminating power of words (features) used to formulate questions.

7 CONCLUSION

Information loss is an important problem in organizations. Despite the effort spent to keep the most important knowledge documented inside organizations, a significant part of this important asset is still in the employees' individual context. That knowledge is habitually transferred through direct communication between employees. The widespread adoption of instant messengers created an opportunity to capture that knowledge and use it later on by questioning an employee's surrogate implemented by a conversation agent.

In this paper, we present the architecture, data preparation methods, and machine learning algorithms required for the implementation of a conversation agent to be used in one of the biggest Portuguese companies. The choice of the algorithm and data preparation techniques was validated experimentally using two large datasets available on the internet.

As further research, we plan to extend the knowledge base created to other communication services with large text entropy (e.g., email). These services present new challenges evidenced by the Newsgroups dataset, mainly caused by the higher levels of entropy in the text classification.

ACKNOWLEDGMENTS

This work is financed by National Funds through the Research Centre in Education, Technology and Heath (CI&DETS) and the Research Centre in Digital Services (CISeD) of the Polytechnic Institute of Viseu.

REFERENCES

[1] Audrey S Bollinger and Robert D Smith. Managing organizational knowledge as a strategic asset. *Journal of knowledge management*, 5(1):8–18, 2001.
[2] Hsinchun Chen, Roger HL Chiang, and Veda C Storey. Business intelligence and analytics: From big data to big impact. *MIS quarterly*, 36(4), 2012.
[3] Kimiz Dalkir. *Knowledge management in theory and practice*. Routledge, 2013.
[4] Hana Urbancová and Lucie Linhartová. Staff turnover as a possible threat to knowledge loss. *Journal of competitiveness*, 3(3), 2011.
[5] Ong Sing Goh, Chun Che Fung, and Arnold Depickere. Domain knowledge query conversation bots in instant messaging (im). *Knowledge-Based Systems*, 21(7):681–691, 2008.
[6] Robert Dale. The return of the chatbots. *Natural Language Engineering*, 22(5):811–817, 2016.
[7] Erik Cambria and Bebo White. Jumping nlp curves: A review of natural language processing research. *IEEE Computational intelligence magazine*, 9(2):48–57, 2014.
[8] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.
[9] Ana Paula Chaves and Marco Aurelio Gerosa. Single or multiple conversational agents?: An interactional coherence comparison. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 191. ACM, 2018.
[10] Yuanchao Liu, Ming Liu, Zhimao Lu, and Mingkai Song. Extracting knowledge from on-line forums for non-obstructive psychological counseling q&a system. *International Journal of Intelligence Science*, 2(02):40, 2012.
[11] Shinhee Hwang, Beomjun Kim, and Keeheon Lee. A data-driven design framework for customer service chatbot. In *International Conference on Human-Computer Interaction*, pages 222–236. Springer, 2019.
[12] Rik Crutzen, Gjalt-Jorn Y Peters, Sarah Dias Portugal, Erwin M Fisser, and Jorne J Grolleman. An artificially intelligent chat agent that answers adolescents' questions related to sex, drugs, and alcohol: an exploratory study. *Journal of Adolescent Health*, 48(5):514–519, 2011.
[13] Petter Bae Brandtzaeg and Asbjørn Følstad. Why people use chatbots. In *International Conference on Internet Science*, pages 377–392. Springer, 2017.
[14] Robert O Briggs, Gwendolyn L Kolfschoten, Gert-Jan de Vreede, Conan Albrecht, Stephan Lukosch, and Douglas L Dean. A six-layer model of collaboration. In *Collaboration Systems*, pages 225–242. Routledge, 2015.
[15] Eva Bittner, Sarah Oeste-Reiß, and Jan Marco Leimeister. Where is the bot in our team? toward a taxonomy of design option combinations for conversational agents in collaborative work. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*, pages 284–293, 2019.
[16] Jennifer Hill, W Randolph Ford, and Ingrid G Farreras. Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations. *Computers in Human Behavior*, 49:245–250, 2015.
[17] Abu Shawar and ES Atwell. Chatbots: are they really useful? *Journal for Language Technology and Computational Linguistics*, 22(1):29–49, 2007.
[18] Anbang Xu, Zhe Liu, Yufan Guo, Vibha Sinha, and Rama Akkiraju. A new chatbot for customer service on social media. In *Proceedings of the 2017 CHI Conference*

on Human Factors in Computing Systems, pages 3506–3510. ACM, 2017.

[19] Karen E Lochbaum. A collaborative planning model of intentional structure. *Computational linguistics*, 24(4):525–572, 1998.

[20] Charles Rich, Neal Lesh, Andrew Garland, and Jeff Rickel. A plug-in architecture for generating collaborative agent responses. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2*, pages 782–789. ACM, 2002.

[21] Sethuramalingam Subramaniam, Pooja Aggarwal, Gargi B. Dasgupta, and Amit Paradkar. Cobots - a cognitive multi-bot conversational framework for technical support. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pages 597–604, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

[22] Henry Lieberman. Autonomous interface agents. In *CHI*, volume 97, pages 67–74. Citeseer, 1997.

[23] James Allen, Nate Blaylock, and George Ferguson. A problem solving model for collaborative agents. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2*, pages 774–781. ACM, 2002.

[24] George Ferguson, James Allen, Nathan Blaylock, Donna Byron, Nate Chambers, Myroslava Dzikovska, Lucian Galescu, Xipeng Shen, Robert Swier, and Mary Swift. The medication advisor project: Preliminary report. 2002.

[25] David C Uthus and David W Aha. Multiparticipant chat analysis: A survey. *Artificial Intelligence*, 199:106–121, 2013.

[26] Henry Lieberman et al. Letizia: An agent that assists web browsing. *IJCAI (1)*, 1995:924–929, 1995.

[27] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

[28] Aixin Sun, Ee-Peng Lim, and Ying Liu. On strategies for imbalanced text classification using svm: A comparative study. *Decision Support Systems*, 48(1):191–201, 2009.

[29] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[30] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

[31] Sang-Bum Kim, Kyoung-Soo Han, Hae-Chang Rim, and Sung Hyon Myaeng. Some effective techniques for naive bayes text classification. *IEEE transactions on knowledge and data engineering*, 18(11):1457–1466, 2006.

[32] Shasha Wang, Liangxiao Jiang, and Chaoqun Li. Adapting naive bayes tree for text classification. *Knowledge and Information Systems*, 44(1):77–89, 2015.

[33] Dik L Lee, Huei Chuang, and Kent Seamons. Document ranking and the vector-space model. *IEEE software*, 14(2):67–75, 1997.

[34] Akiko Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65, 2003.

[35] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[36] Edward Loper and Steven Bird. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*, 2002.

[37] Wes McKinney. pandas: a python data analysis library. *see http://pandas. pydata. org*, 2015.

[38] Joblib. Joblib tool. https://joblib.readthedocs.io/en/latest/.

[39] Armin Ronacher. Flask: A simple framework for building complex web applications. https://www.palletsprojects.com/p/flask/.

[40] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[41] The 20 newsgroups data set. http://qwone.com/~jason/20Newsgroups/. Accessed: 2019-09-17.

[42] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.