Chatbol, a chatbot for the Spanish "La Liga"

Carlos Segura, Àlex Palau, Jordi Luque, Marta R. Costa-Jussà and Rafael E Banchs

Abstract This work describes the development of a social chatbot for the football domain. The chatbot, named *chatbol*, aims at answering a wide variety of questions related to the Spanish football league "La Liga". Chatbol is deployed as a Slack client for text-based input interaction with users. One of the main Chatbol's components, a NLU block, is trained to extract the intents and associated entities related to user's questions about football players, teams, trainers and fixtures. The information for the entities is obtained by making sparql queries to Wikidata site in real time. Then, the retrieved data is used to update the specific chatbot responses. As a fallback strategy, a retrieval-based conversational engine is incorporated to the chatbot system. It allows for a wider variety and freedom of responses, still football oriented, for the case when the NLU module was unable to reply with high confidence to the user. The retrieval-based response database is composed of real conversations collected both from a IRC football channel and from football-related excerpts picked up across movie captions, extracted from the OpenSubtitles database.

Key words: Language understanding, Natural language processing, Spoken dialogue systems, Question answering system, Retrieval based chat oriented dialogue, Chat oriented dialogue system

Carlos Segura
Telefónica Research, Barcelona e-mail: carlos.seguraperales@telefonica.com

Àlex Palau
Telefónica Research, Barcelona e-mail: alex.palau.domenech@gmail.com

Jordi Luque
Telefónica Research, Barcelona e-mail: jordi.luque@telefonica.com

Marta R. Costa-Jussà
TALP Research Center - Universitat Politècnica de Catalunya, Barcelona e-mail: marta.ruiz@upc.edu

Rafael E Banchs
Institute for Infocomm Research, Singapore e-mail: rembanchs@i2r.a-star.edu.sg

1 Introduction

Recent advances in Machine Learning and Natural Language Understanding are making possible the implementation of human-machine interaction systems that are driven by means of natural language conversations. These systems, commonly referred to as chatbots, have been recently gaining popularity in different mobile and online platforms. Their main objective is to serve as ”virtual assistants” to support informational and transactional user requirements in a wide variety of domains and sectors. The recent boom on conversational interfaces is capturing the attention of both the research community and tech companies as the possible next big technological revolution: *”Until now, we have been forced to learn the language of computers. But in the technological revolution currently underway, computers are finally learning to understand ours.”*[1]

Regardless the actual impact and adoption of numerous applications in the market, the state-of-the-art of these technologies is still quite limited and their development requires a high cost of adaptation to the specific domain. Handcrafted applications with a limited amount of automation and a high level of human intervention is typically the common scenario. Nevertheless, successful applications, which are able to collect and model huge amount of interactions, are the best candidates to leverage on machine learning techniques to significantly reduce the level of human intervention.

Some examples of existing service chatbots include systems in the health domain, such as currently described chatbot service to provide Health information about HIV/AIDs [4], primary care [14] or mental Health counseling [15]; the educational domain, such as [21], a chatbot to improve interactions in online courses or MOOCBuddy chatbot [11] to provide personalized course recommendations; the entertainment domain, such a Stark Trek system [12] or Humorist Bot that can generate and recognize humorous sentences [1] ; Human Resources chatbot [5] or Customer Service agents [29], just to cite a few.

In this work we describe CHATBOL, a chatbot for providing information about the Spanish Football League ”La Liga”. Different from similar systems that offers news and other factual information about the Spanish Football League, the presented system combines two different conversational functions into a single agent: a question answering system that search over Wikidata entities to generate responses related to the topic of interest and a retrieval-based chat-oriented dialogue system that serves as fallback mechanism to reply to either out-of-domain user queries or in-domain queries that cannot be served by the question answering system.

The rest of the paper is structured as follows. First, in section 2, an overall description of our system is presented. In section 3, some related work is briefly described. In Section 4, a comprehensive description of the system architecture and its main components is provided. In section 5, the experimental framework is described

[1] https://medium.com/conversational-interfaces/conversational-interfaces-the-next-big-technological-revolution-4efea1d97606

along with the conducted evaluations and results. Finally, in section 6, the main conclusions of our work are presented along with the future work to be conducted.

2 Chatbol Description

3 Related Work

Chatbots have a long history since first prototype appeared in mid 60s [27]. The objective of this section is to make the paper self-contained, for complete survey the reader should refer to [18]. During all these years, chatbots have been approached from different perspectives which mainly include rule-based [26] or data-driven which are retrieval/statistical-based [8, 19, 2, 6] or generative-based [25, 28] just to name a few examples. Within these data-driven approaches, the main challenge nowadays is to take advantage of large datasets. For tackling previous challenge, our proposed model uses a mixture of rule and retrieval-based approach and is fed by different public resources such as Wikidata and the OpenSubtitles database. Note that both data resources have already been employed to develop chatbot engines, previously in [22, 25].

More specifically, in this paper, we are aiming at training a task-oriented chatbot. In task-oriented, the objective of the tool is to get information from the user in order to accomplish a particular task. Tools of these kind that have become very popular in real life are Siri or Alexa, among others. In our case, we focus the chatbot in the domain of football, similarly to tools like Toni from Facebook Messenger [2] or Tokabot [3], but for the particular case of the Spanish league.

4 Architecture

The architecture of Chatbol is composed of a Natural Language Understanding module based on RASA NLU [3], a Wikidata entity search module to generate responses and Slack plugin. Additionally a retrieval-based conversational engine is included as a fallback option, when the first block is unable to provide a response due to the NLU block is not recognizing intents or entities with a high likelihood score or when the entities are not found in Wikidata. A diagram of the architecture is depicted in Fig. 1.

---

[2] https://toni.football/

[3] https://botlist.co/bots/tokabot-football

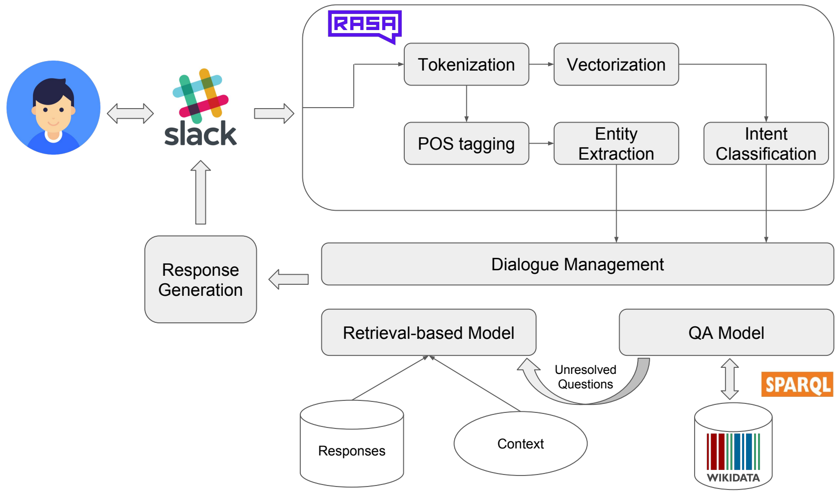

Fig. 1 Chatbol architecture: Rasa NLU is used to recognize intents and entities that are passed to the QA model that maps user input into a query to Wikidata to generate the answer. As a fallback strategy, for unresolved questions, a response is selected from the retrieval-based model.

4.1 Natural Language Understanding

Rasa NLU is an open source library for intent classification and entity extraction. It is a high level library similar to online services that parses user messages into dialog acts composed of intents and a set of corresponding entities. Rasa NLU follows a modular design, relying on existing NLP and Machine learning libraries, like spaCy [10], scikit-learn [16], sklearn-crfsuite [13].

The Rasa NLU pipeline consists of a tokenization process followed by a parts of speech (POS) tagging by means of spaCy NLP library. Then, the GloVe vectors [17] extracted from each token are concatenated forming a feature vector of the whole sentence. This sentence representation is used to train a multiclass support vector machine used to recognize user intents. Finally, a conditional random field classifier is trained on the sentence tokens and POS tags to extract entities. The intent classifier and entity extractor are trained using scikit-learn and sklearn-crfsuite, respectively.

For training Rasa NLU, a list of questions annotated with intents and entities is prepared using the human readable json format. In this work, the following 7 intents are included in the chatbot:

- What team is [player] playing for ?
- List of players of [team]
- What is the stadium of [team] ?
- What team is [stadium] of ?
- Who is the coach of [team] ?
- What team is coached by [coach] ?
- Get fixtures of [team]

The list of questions is created by writing around 10 ad hoc phrases for every single intent and afterwards expanded using a list of football players, teams, stadiums and coach gathered using the Wikidata API [4]. This process helps Rasa NLU training process to correctly identify not only the intents but also the entities the chatbol is interested in.

4.2 Response generation

Chatbol bot is connected to Slack using slackclient API [5]; therefore, all interaction with the bot is through this instant messaging software. Once the bot receives a message, Rasa NLU analyses the sentence and returns a human readable json package with information on:

- The likelihood of the sentence belonging to each one of the trained intents.
- Every entity that has been identified.

Using this information, the chatbot has been designed to try the 3 (number obtained via trial and error) most likely intents in order. For each intent and entity, Chatbol sends a query to either Wikidata or an ad hoc structured database with information on the following matches of every team obtain via the scrapping of the site futbolportv.com [6].

In case the combination of intent and detected entity results in an answer, this answer is sent via the Slack API. It is also possible that none of the 3 intents and entities results in a response; in that case, a retrieval-based response block described in subsection 4.4 will be the generator of the answer.

4.3 Database query

In order to have always the most up-to-date information, Chatbol obtains the answer to the detected intent on the fly using the Wikidata API or web scrapping. In case of Wikidata, it first obtains the Qnumber (reference number used in Wikidata to point at an element of information) of the entity via wikidata, by making a POST to the Wikidata API service with the name of the entity. Then, once the number is found, we generate a query to Wikidata using SPARQLWrapper [7], a SPARQL endpoint interface to Python. The answer from Wikidata is analyzed, reformatted to natural language and sent as a response to the user. The sample Python code depicted in Fig. 2 shows how to get the list of players of a team, given the team qnumber:

There are two ways this process may go wrong. First, the Qnumber may not be found, either because the detected entity does not exist or because the name is not

[4] `https://www.mediawiki.org/wiki/API`

[5] `https://api.slack.com/`

[6] `https://futbolportv.com/`

```
def get_playeres_team(qnumber):
    sparql = SPARQLWrapper("https://query.wikidata.org/sparql")
    sparql.setQuery("""SELECT DISTINCT ?playerLabel WHERE {
        ?player wdt:P106 wd:Q937857.
        ?player p:P54 ?teamMember.
        ?teamMember ps:P54 wd:""" + qnumber + """.
        OPTIONAL {?player p:P1449 ?Nickname.}
        ?player wdt:P21 wd:Q6581097.
        SERVICE wikibase:label { bd:serviceParam  \
                wikibase:language "[AUTO_LANGUAGE],en, \
            es, ca, de". }
        FILTER(NOT EXISTS { ?teamMember pq:P582 ?end. })
        FILTER(EXISTS { ?teamMember pq:P580 ?start. })
        FILTER(NOT EXISTS { ?player wdt:P570 ?death. })
        OPTIONAL { ?player wdt:P569 ?birth. }
        FILTER(year(?birth) > 1979).
        }
        ORDER BY ?playerLabel""")
    sparql.setReturnFormat(JSON)
    return sparql.query().convert()
```

Fig. 2 Automatic Python + SPARQL code to retrieve the list of players of a football team specified by its Qnumber

exactly the one Wikidata has for that entity. For instance, the player Iniesta is not recognized as Andrés Iniesta by Wikidata. The big number of alias and ways of calling the different players and also the objective of making the chatbot as most user-friendly as possible made this a critical problem.

Therefore, a solution based on Fuzzywuzzy Python package [9] has been applied. This solution allows the comparison of the similarities between a word and an array of words and the selection of the most similar one from the array while also returning a value from 0 to 100 of the similarity between both words. In the chatbot case, the entity detected in the user input is compared with the array of words formed by the names of the players, coaches, teams or stadiums (depending on the intent). If the similarity is big enough (70 in our case), the most similar word is used to find the Qnumber in Wikidata. If no word is similar enough, the intent is supposed to be erroneous and the algorithm jumps to the next most likely intent.

The second way the Wikidata query may go wrong is because the query generated returned no valid information (it may be asking Wikidata for the stadium of a coach, for instance). In this case the intent is also categorized as erroneous and the same process as in the case before is followed.

4.4 Chichat fallback component

Chatbol includes a fallback mechanism to produce responses when the main system component is unable to do it. This can happen when the confidence score of the intent detection falls below a threshold, or when the information requested is not found in Wikidata. This mechanism is based on a retrieval-based conversational engine, which follows the Informal Response Interactive System (IRIS) approach [2]. In this type of systems, semantic similarity scores at the turn and dialogue history levels are combined to retrieve the most suitable matches from a database of dialogues. The retrieved candidate responses are then ranked by using a recall-oriented re-ranking algorithm. Favoring recall (instead of precision) helps producing more varied responses, making the chatbot more interesting

Spanish movie data from OpenSubtitles [23] was used as dialogue database. As we wanted the system to be able to converse in the topic of interest, only those files containing the word ”fútbol” (Spanish for soccer) more than ten times were extracted. In this way, from a total of 191,987 files contained in the Spanish OpenSubtitles data collection, only 94 were useful for our purpose. In order increase the amount of data supporting the Chitchat component, we also included chat sessions from the #futbol channel at IRC-HISPANO[7]. A total of 3,638 sessions comprising 148,893 utterances were collected and included as part of the dialogue database.

The implementation was done in Python, using the scikit-learn library [16]. Standard word tokenization was used and spanish accents removed, mapping all characters to ASCII. Separated indexes were constructed for turns and dialogue sessions in the data collection. The definition of a dialogue session in the OpenSubtitles dataset is somehow ambiguous as no explicit information on when a new dialogue session starts is available. To this end, a breaktime threshold of 3 seconds was used for deciding on dialogue boundaries.

The chat-oriented dialogue system is inspired on the vector space model framework [20, 24]. The vectorization of the user input and database utterances is completed by projecting them into their Term frequency - inverse document frequency (Tf-idf) representation. The vector similarity among user’s input and stored sentences are computed for retrieving best matches from the dialogue databases. For doing so, three different vector space models (VSM) are computed. It aims at describing the dialogue database Tf-idf vector distributions with respect the current state of the dialogue. That is, to account for the utterance-level and the dialogue-level information. Cosine similarity is employed to compute scores among both previous and current state vectors and their respective VSMs. Similarly, the history score is also obtained by using cosine similarity through the current dialogue history, which includes all utterances interchanged by the current user and Chatbol. It is worth to note that the history vector is updated with each new sentence in the dialogue yielding to a collection of stacked Tf-idf vectors at each turn level. The history score is computed between this vector and the vector representations for each full dialogue stored in the dialogue database. In summary, for each new user input in a

[7] https://chathispano.com/

given conversation, three cosine similarity scores are computed (current utterance, previous utterance and dialogue history) and combined into an overall score to retrieve the candidate responses. The overall cosine similarity score is computed as in the following equation and updated at each dialogue-turn step,

$$Score = 0.6 * u_c + 0.1 * u_p + 0.3 * h \,,$$

where the utterance-level scores (u_c and u_p) and the dialogue-level h score are combined for drawing a final rank of candidate responses. The following weighting factors: 0.6, 0.1 and 0.3, were set after some manual experimentation.

5 Experimental Framework and Evaluation Results

The experimental framework is based on Slack, where users can interact with direct messages to the Chatbol app. This makes it possible to maintain a separate conversation context for each user and is very convenient for evaluating the chatbot responses.

The evaluation protocol used follows the current WOCHAT Shared Task evaluation framework[8]. In this sense, two different evaluations were conducted, one a the session level and the other at the turn level. For the session level evaluation, we used the available User Satisfaction Survey, in which each user has to respond to eight closed questions and three open questions after interacting with the chatbot. These questions are mainly focused on the user experience aspect of the interaction, as well as on the overall performance of the system. Results of the user satisfaction survey are depicted in Fig. 3.

As seen from the figure, most of the users found the system responses to be appropriate either *sometimes* or *often*. In the open questions, users commented on the strengths and problems of the systems. Among the strengths, it was mentioned that the system was able to provide "*accurate information about Spanish soccer teams*" and "*capable to understand incorrect words and very flexible with team names variations*". Overall the system was perceived as a "*fast way to get football data without searching Internet*. Among the main problems reported, evaluators agreed that "*sometimes the question intents are not properly captured by the system*", even "*misunderstands basic questions*" and "*despite small changes in the sentence*.

For the turn level evaluation, we followed the provided guidelines for appropriateness score, in which each turn of the dialogue session is annotated with one of the following three scores: *valid*, *acceptable* or *invalid*, according to its perceived appropriateness with respect to the current utterance and dialogue context. A total of 6 users, tested and evaluated Chatbol. The results of our evaluation are summarized in table 1, showing the cumulative number of scores for both conversational engines. Defining an appropriate answer as the response scored with *valid*, or *acceptable*, the QA system is able to generate appropriate responses 72% of the time, whereas the

[8] http://workshop.colips.org/wochat/main_sharedtask.html

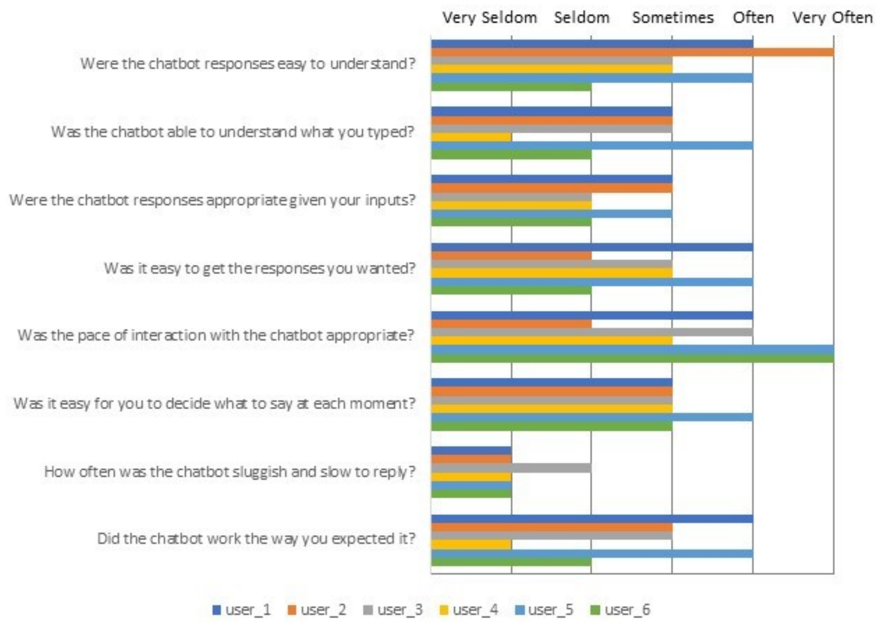

Fig. 3 Results of the User Satisfaction Survey

Responses / System	Valid	Acceptable	Invalid
QA Model	34 (53%)	12 (19%)	18 (28%)
Retrieval Model	50 (43%)	28 (25%)	37 (32%)

Table 1 Evaluation of the responses generated by Chatbol for each conversational model.

chitchat system obtains a 68% of appropriate answers. Below is a sample of a few selected interactions with the the chatbot for both conversational engines. Samples were originally in Spanish and have been translated into English:

Chitchat engine:
user: how’s life ?
chatbol: I am a bit injured, I had an ankle sprain last Tuesday... How are you?
user: Which team do you support ?
chatbol: I support the rival for your team
user: I am an atletic fan
chatbol: atletic is a great team, good players, coacher and supporters, they are huge!
user: is the football match on TV now ?
chatbol: it just started

user: how is the match going ?
chatbol: leading 1-0
user: the game ended in a tie
chatbol: was the goal during the penalty shoot-out?
user: yes, in the very last moment
chatbol: what were the scores?
user: 1-1
chatbol: cool, I won the sweepstakes! I think no one else bet for 1-1
user: bye
chatbol: take care :d

QA engine:
user: What team does Messi play for?
chatbol: Lionel Messi plays for the FC Barcelona
user: What are the next matches for Barça ?
chatbol: Next Thursday 11th January at 21:30h Barcelona - Celta in Copa del Rey leage. You can watch the match in: TBD
user: who is Real Madrid coach now ?
chatbol: Ziniedine Zidane is the coach of Real Madrid
user: Who is the coarch of sevilla ?
chatbol: Vincenzo Montella is the coach of Sevilla Fútbol Club
user: What stadium does Osasuna play in ?
chatbol: El Sadar Stadium is is the home stadium of Club Atlético Osasuna
user: When does Geta play?
chatbol: Next Friday 12th January at 21:00h Getafe - Málaga in LaLiga Santander leage. You can watch de match in: LaLiga TV, beIN LaLiga.
user: What is the current squad of Barça ?
chatbol: The players on the squad of Fútbol Club Barcelona are: André Gomes, Andrés Chávez, Andrés Iniesta, Arda Turan, Gerard Piqué, Ivan Rakitić, Jasper Cillessen, Javier Mascherano, Jorge Alba Jose Manuel, Arnáiz Díaz, Lionel Messi, Lucas Digne Marc-André ter Stegen Munir El Haddadi Nélson Semedo Ousmane Dembélé Paulinho Philippe Coutinho, Samuel Umtiti, Sergi Roberto, Sergi Samper, Sergio Busquets, Thomas Vermaelen, Wilfrid Kaptoum

6 Conclusions

This paper described the development of a chatbot for providing information about the Spanish Football League ”La Liga”. The proposed system combines two different conversational engines into a single agent: a QA engine that maps natural language utterances of the user into SPARQL queries to retrieve data from Wikidata to generate responses and a retrieval-based chat-oriented dialogue engine that serves as fallback mechanism to reply to either out-of-domain user queries or in-domain queries that cannot be served by the question answering system. The evaluation of the chatbot at turn level show that the users found the system responses to be appropriate 72% of the time for the QA engine and 68% for the chitchat engine. From the session level evaluation, users found that current implementation is robust against word errors and able to recognize variations of player and team names, but sometimes failed to recognize the intent correctly, even for simple questions. For future work we plan to improve the intent classification block and to extend the QA model by incorporating to the system the most popular questions found in user inputs. We are also currently implementing interactive message buttons after each chatbot response to be able to get instant feedback from users.

Acknowledgements

This study has been funded by the Spanish Ministerio de Economía y Competitividad, the European Regional Development Fund and the Agencia Estatal de Investigación, through the postdoctoral senior grant Ramón y Cajal, the contract TEC2015-69266-P (MINECO/FEDER,EU) and the contract PCIN-2017-079 (AEI/MINECO).

References

1. Agnese Augello, Gaetano Saccone, Salvatore Gaglio, and Giovanni Pilato. Humorist bot: Bringing computational humour in a chat-bot system. In *Complex, Intelligent and Software Intensive Systems, 2008. CISIS 2008. International Conference on*, pages 703–708. IEEE, 2008.
2. Rafael E. Banchs and Haizhou Li. IRIS: a chat-oriented dialogue system based on the vector spac e model. In *The 50th Annual Meeting of the Association for Computational Lin guistics, Proceedings of the System Demonstrations, July 10, 2012, Jeju Is land, Korea*, pages 37–42, 2012.
3. Tom Bocklisch, Joey Faulker, Nick Pawlowski, and Alan Nichol. Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181*, 2017.
4. Jacqueline Brixey, Rens Hoegen, Wei Lan, Joshua Rusow, Karan Singla, Xusen Yin, Ron Artstein, and Anton Leuski. Shihbot: A facebook chatbot for sexual health information on hiv/aids. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 370–373, 2017.
5. Praveen Chandar, Yasaman Khazaeni, Matthew Davis, Michael Muller, Marco Crasso, Q. Vera Liao, N. Sadat Shami, and Werner Geyer. Leveraging conversational systems to assists new hires during onboarding. In Regina Bernhaupt, Girish Dalvi, Anirudha Joshi, Devanuj K. Balkrishan, Jacki O'Neill, and Marco Winckler, editors, *Human-Computer Interaction - INTERACT 2017*, pages 381–391, Cham, 2017. Springer International Publishing.
6. Mehdi Fatemi, Layla El Asri, Hannes Schulz, Jing He, and Kaheer Suleman. Policy networks with two-stage training for dialogue systems. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 101–110, Los Angeles, September 2016. Association for Computational Linguistics.
7. Sergio Fernández and Alexey Zakhlestin. Sparql endpoint interface to python. `https://github.com/RDFLib/sparqlwrapper`, 2017.
8. M. Gašić, F. Jurčíček, S. Keizer, F. Mairesse, B. Thomson, K. Yu, and S. Young. Gaussian processes for fast policy optimisation of pomdp-based dialogue managers. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '10, pages 201–204, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

9. J Gonzalez, P Rodrigues, and A Cohen. Fuzzywuzzy: Fuzzy string matching in python. https://github.com/seatgeek/fuzzywuzzy, 2017.
10. Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. https://github.com/explosion/spaCy, 2017.
11. Adrian Iftene and Jean Vanderdonckt. Moocbuddy: a chatbot for personalized learning with moocs. In *RoCHI–International Conference on Human-Computer Interaction*, volume 91, 2016.
12. Grishma Jena, Mansi Vashisht, Abheek Basu, Lyle Ungar, and João Sedoc. Enterprise to computer: Star trek chatbot. *arXiv preprint arXiv:1708.00818*, 2017.
13. Mikhail Korobov. sklearn-crfsuite. https://github.com/TeamHG-Memex/sklearn-crfsuite, 2017.
14. Lin Ni, Chenhao Lu, Niu Liu, and Jiamou Liu. Mandy: Towards a smart primary care chatbot application. In Jian Chen, Thanaruk Theeramunkong, Thepachai Supnithi, and Xijin Tang, editors, *Knowledge and Systems Sciences*, pages 38–52, Singapore, 2017. Springer Singapore.
15. Kyo-Joong Oh, Dongkun Lee, Byungsoo Ko, and Ho-Jin Choi. A chatbot for psychiatric counseling in mental healthcare service based on emotional dialogue analysis and sentence generation. In *Mobile Data Management (MDM), 2017 18th IEEE International Conference on*, pages 371–375. IEEE, 2017.
16. 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.
17. Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
18. Kiran Ramesh, Surya Ravishankaran, Abhishek Joshi, and K. Chandrasekaran. *A Survey of Design Techniques for Conversational Agents*, pages 336–350. Springer Singapore, Singapore, 2017.
19. Alan Ritter, Colin Cherry, and William B. Dolan. Data-driven response generation in social media. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 583–593, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
20. G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975.
21. D. Song, E. Y. Oh, and M. Rice. Interacting with a conversational agent system for educational purposes in online courses. In *2017 10th International Conference on Human System Interactions (HSI)*, pages 78–82, July 2017.
22. Alessandro Suglia, Claudio Greco, Pierpaolo Basile, Giovanni Semeraro, and Annalina Caputo. An automatic procedure for generating datasets for conversational recommender systems. *CLEF*, 2017.
23. Jörg Tiedemann. News from opus-a collection of multilingual parallel corpora with tools and interfaces. In *Recent advances in natural language processing*, volume 5, pages 237–248, 2009.
24. C. J. 'Keith' van Rijsbergen. A probabilistic logic for information retrieval. In David E. Losada and Juan M. Fernández-Luna, editors, *Advances in Information Retrieval*, pages 1–6, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
25. Oriol Vinyals and Quoc V. Le. A neural conversational model. *CoRR*, abs/1506.05869, 2015.
26. Richard Wallace. The elements of aiml style. 2003.
27. Joseph Weizenbaum. Eliza: a computer program for the study of natural language communica tion between man and machine. *Commun. ACM*, 9(1):36–45, January 1966.
28. Yu Wu, Wei Wu, Zhoujun Li, and Ming Zhou. Topic augmented neural network for short text conversation. *CoRR*, abs/1605.00090, 2016.
29. 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.