



Automatic Detection of Inconsistencies in Open-Domain Chatbots

Jorge Mira Prats¹, Marcos Esteche-Garitagoitia¹, Mario Rodríguez-Cantelar² , Luis Fernando D'Haro¹ 

¹Speech Technology and Machine Learning Group (THAU), ETSI de Telecomunicación, Universidad Politécnica de Madrid, Av. Complutense 30, 28040, Madrid, Spain

²Centre for Automation and Robotics (CAR) UPM-CSIC - Intelligent Control Group (ICG), Universidad Politécnica de Madrid, Cl. José Gutiérrez Abascal, 2, 28006 Madrid, Spain.

luisfernando.dharo@upm.es

Abstract

Current pre-trained Large Language Models applied to chatbots are capable of producing good quality sentences, handling different conversation topics, and larger interaction times. Unfortunately, the generated responses highly depend on the data on which the chatbot has been trained on, the specific dialogue history and current turn used for guiding the response, the internal decoding mechanisms, ranking strategies, among others. Therefore, it may happen that for the same question asked by the user, the chatbot may provide a different answer, which in a long-term interaction may produce confusion.

In this paper, we propose a new methodology based on three phases: a) automatic detection of dialogue topics using zero-shot learning approaches, b) automatic clustering of distinctive questions, and c) detecting inconsistent answers using K-Means clustering and the Silhouette coefficient. To test our proposal, we used the DailyDialog dataset to detect up to 13 different topics. To detect inconsistencies, we manually generated multiple paraphrased questions. Then, we used multiple pre-trained chatbots to answer those questions. Our results in topic detection show a weighted F-1 value of 0.658, and a 3.4 MSE to predict the number of different responses.

Index Terms: chatbots, inconsistent responses, zero-shot topic detection, clustering.

1. Introduction

In recent years, the number of open-domain conversational systems has increased due to multiple factors. On the one hand, the interest from companies in providing alternative means of communication with clients and potential users, as well as a mechanism to reduce operational costs. Users are also becoming more familiar with these types of system, expecting them to provide quick answers to their requests or questions, providing entertainment, a deeper understanding of their needs, and serve as an alternative mechanism for interaction. Finally, the technology has also brought important improvements in terms of consistency, knowledge, engagement, and even empathy.

In terms of technology, the usage of pre-trained large language models (LLMs), in combination with information retrieval techniques and controlled generation, is responsible for very interesting chatbots such as BlenderBot vs 3.0 [1] or Lamda [2]. However, chatbots can generate different responses over a similar semantic user's question due to multiple factors such as a) the data used to train the model may include contradictions, b) the effect of variations in the dialogue history and user's questions (i.e., paraphrases), or c) the usage of different decoding mechanisms to generate variations in the responses

(e.g., top-k, top-p, greedy search, beam-search, etc.), which in a long-term interaction may generate confusions in the user.

On the other hand, open-domain chatbots must deal with multiple topics along a conversation. Therefore, accurate and fine-grained detection of the different topics is highly relevant to keep track of the interaction, deciding the next states in the conversation, and providing relevant information. For instance, a user could be talking about the music performed by his/her favorite singer, who is also starring in a recent movie; therefore, the chatbot is required to keep track of which topic (e.g., entertainment) and subtopic (e.g., music) the user is talking about to avoid producing an answer that could potentially break the dialogue. Unfortunately, most topic classifiers are static as they can only classify the limited number of classes seen during training.

In this paper, we describe some practical solutions to overcome some of the limitations mentioned above. First, we present a hierarchical algorithm for topic classification based on the use of zero-shot learning strategies. The algorithm is tested over a set of 13 topics found in 1000 randomly selected and annotated dialogue turns from the DailyDialog dataset. Second, we introduce an automatic algorithm to detect inconsistencies in chatbot responses. The algorithm makes use of clustering strategies to detect multiple semantic variations of the same user's question and to detect when the provided answers are different (inconsistent). Finally, it proposes the answer that is most commonly found (centroid) which can be used to control future generations of chatbot responses. Our preliminary results show the advantages of the proposed methodologies.

The paper is organized as follows. In section 2, we describe the dataset used for the topic detection task, and the creation of the dataset used for the automatic detection of inconsistencies. Then, in section 3 we describe both algorithms. In section 4 we show our preliminary results. Finally, section 5 presents our conclusions and future work.

Question	Answer
Tell me your favorite sport?	My favorite sport is basketball. What is your favorite sport?
	I don't know what is my favorite sport.
What is your favorite food?	I'm not sure what my favorite food is.
	While I appreciate food and drink, my taste is better reflected in the company I work for.
	Hmm, what's my favorite food? I really like orange chicken.
Which movie do you like the most?	I really like The Dark Knight. How about you? What movie do you like?
	My favorite movie is Despicable me!
	My favorite movie is Star Wars. What is your favorite movie?

Figure 1: Examples of inconsistencies, for different topics, in the responses of a chatbot over the same question.

Table 1: *DailyDialog dataset statistics*

Info	Statistics
No. conversations	13,116
Avg. No. turns per dialogue	7.85
Max. No. turns per dialogue	35
Avg. No. tokens per turn	15.80
Max. No. tokens per turn	300

2. Datasets

For the zero-shot topic detection, we used the DailyDialog dataset [3] that contains high-quality human-written conversations that cover a wide range of generic topics, such as relationships, daily life, and work. Conversations on DailyDialog are mainly for information exchange and improving social bonds. Table 1 shows some of the statistics of the dataset. We processed the 13k turns and randomly selected a total of 1000 turns to automatically detect the topic for each turn using the method described in section 3.1. Then, a group of 4 experts manually annotated the selected turns, classifying them according to one of the following topics (numbers in parenthesis represent the number of occurrences in the data): animals (12), books (4), cars (56), family (39), fashion (108), finance (119), food (184), movies (6), music (15), photography (4), sports (27), weather (7), and the bag-like category *others* (419). Annotators could optionally provide a more specific category than the generic *others* if they wish for future research.

For the detection of inconsistencies, we manually created a set of 15 different questions and their corresponding paraphrases. For the paraphrases, we ensure that the same semantic meaning and intent are maintained. Table 2 shows some examples of the questions and paraphrases generated. A total of 107 questions and paraphrases were collected (i.e., average of 6.3 paraphrases per question). Then, we used four different SotA pre-trained chatbots to collect their answers to the set of questions and paraphrases created before. In concrete, we selected: a) DialogGPT-large [4] available in HuggingFace¹, b) BlenderBot vs 2.0 (90M), c) BlenderBot vs 2.0 (2.7B) [5, 6, 7] both available in ParlAI² and ³, and d) Seeker available in ⁴ without using its retrieval search module. Using these four chatbots, we made sure (see section 3.2). Then, we asked four expert annotators to read the answers provided by each chatbot to the different questions and paraphrases and count how many different semantic responses each chatbot generated. In theory, each chatbot should generate a single coherent answer, but our results show an average of 4 different answers (see section 4). The Krippendorff alpha coefficient for the inter-annotator agreement was 0.74.

3. Architecture

This section describes the proposed algorithms for the zero-shot topic classifier and detection of response inconsistencies.

¹<https://huggingface.co/microsoft/DialogGPT-large>

²<https://parl.ai/projects/blenderbot2/>

³<https://parl.ai/projects/recipes/>

⁴<https://parl.ai/projects/seeker/>

Table 2: *Example of questions and paraphrases created for the detection of inconsistencies.*

Question	Paraphrases
What is your favorite sport?	Which is the sport you like the most? My favorite sport is basketball, and yours? What kind of sport do you like?
What is your favorite book?	What is the title of your favorite book? Which book you always like to read? Hi!! I like reading, which book is your most favorite one?
What is your job?	What do you do for a living? What do you do for work?

3.1. Zero-Shot Topic Classification

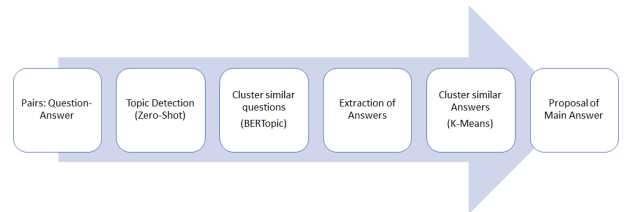
The purpose of this algorithm is to detect, in a scalable and flexible way, topics and subtopic classes found along the different turns in a dialogue, and without requiring training an ad-hoc topic classifier. The advantage in this case is that the number of topics or subtopics can dynamically be extended without requiring any labeled data or re-training.

Our algorithm is based on the zero-shot classification proposed in [8] which considers classification as the result of a natural language inference (NLI) process. We used the DeBERTa model proposed in [9] and available as the nli-deberta-base model from HuggingFace⁵.

In the approach proposed by [8], they use a pre-trained MNLI sequence pair classifier, where the classification is performed by embedding sequences into the same latent space and measuring the distance between them. In this case, the NLI approach considers there are two sentences: a "premise" and a "hypothesis" the pre-trained model is asked to determine whether the hypothesis is true (entailment) or false (contradiction) given the premise. Here, the "premise" is the sentence to classify, and each candidate label is considered an independent "hypothesis." Then, the probability on how much the NLI model predicts that the premise "entails" the hypothesis is stored and used to rank the given set of labels, selecting as topic the one with the highest probability.

Unfortunately, the problem with this approach is that it will not scale if the number of labels is too large and will not allow to quickly find subtopics (i.e., a fine-grained analysis of the topic of the conversation). Therefore, we decided to divide the process into two steps. In the first step, a high-level set of words is defined, which are considered topics. For example, entertain-

⁵<https://huggingface.co/cross-encoder/nli-deberta-base>

Figure 2: *Modules and process flow for the proposed methodology.*

ment, sports, family, science, etc. Then, the zero-shot classifier is used over this set of labels, and the keyword with the higher probability (and above a given threshold) provides the detected topic. Then, the selected keyword passes to the second phase, where a more specific set of word-labels is used to detect sub-topics following the same criteria. For example, for the first-level *sports* category, the second set of labels could be baseball, basketball, soccer, or tennis, allowing a more fine-grained sub-topic detection. The advantages of this process are a) the set of words at each level can be changed depending on the final domains and number of topics/sub-topics to detect, and b) the number of hierarchies can be extended to account for more fine-grained detection, e.g., player, coach, stadium, team, etc. Therefore, the process is scalable and easy to modify.

3.2. Detection of response inconsistencies

The second algorithm is intended for the automatic detection of response inconsistencies in chatbots. Multiple factors are responsible for these inconsistencies: a) contradictory information found in the training data of the LLM, b) usage of different decoding strategies and parameter values for generating variations in the responses, e.g., top-p, top-k, greedy beam search, temperature, etc., c) paraphrase variations in the questions posed by the user, d) differences in the dialogue context used for generating the chatbot response, or e) usage of different re-ranking strategies for selecting the final answers.

Our algorithm follows 5 steps as shown in figure 2. The first step starts by disposing of a set of question-answer pairs collected from the logs of the chatbot or from a set of manually built dataset of questions and paraphrases and the respective chatbot answers (as described in section 2).

For the second step, we classify the topic of each question using the zero-shot method described in section 3.1 and select those with the same topic/subtopic generating topic-related batches to speed up and focus the following steps.

In the third step, we cluster similar questions from the same topic or subtopic using BERTopic [10]. This is an unsupervised algorithm that attempts to discover coherent groups by extracting sentence vector embeddings using sentence-BERT [11], then reducing the dimensionality of the sentence embeddings using UMAP [12], then using the HDBScan clustering algorithm [13], and finally extracting the representation of the cluster using a variation of the TF-IDF formulation, called c-TF-IDF (class-based term frequency - Inverse document frequency), which models the importance of the words inside each detected group, also allowing detection of the most representative sentences for each group. Once the question groups are detected, we select the most representative question for each group using the functionalities provided by BerTopic. The motivation for selecting only one question as representative (i.e. centroid) of the group is because all the other questions are similar (i.e. they are paraphrases) and therefore they will not contribute to detect differences in the chatbot answers during step four.

In step four, we select all the answers that belong to the questions that were grouped together in the previous step. Here, we optionally calculate the cosine similarity between all the answers and the representative question and remove those answers whose similarity is below a given threshold (in our case, 0.5) to remove additional noise in the answers.

During step five, we also extract the sentence embedding to each answer for a given group, then we apply the K-Means algorithm to cluster all the answers and estimate the Silhouette coefficient varying the number of groups from two to the number

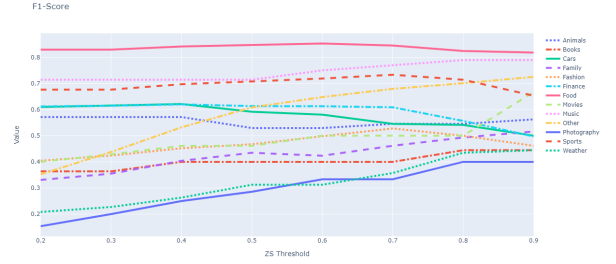


Figure 3: Variation of the zero-shot classifier over different thresholds and topics.

of answers minus one. Then we estimate the average Silhouette coefficient and the Silhouette coefficient per sample. The number of clusters that provides the highest number of samples with a Silhouette coefficient higher than the average Silhouette is selected as a proxy for the number of different responses generated for the chatbot. In case, the system is unable to find good clusters, then a single cluster is considered. Finally, the system also calculates the centroid answer and finds the answer closest to it, providing it as the best candidate to be used to control the generation of chatbot responses. Our motivation is that the designer can consider its usage as part of the persona profile used in TransferTransfo [14] or for applying constrained beam search generation [15].

4. Results

First, we provide the results for zero-shot topic detection. Figure 3 shows the F1 results for the 13 high-level topics (including the *Others* class), found in one thousand sentences selected from DailyDialog, when considering different thresholds for the zero-shot model. As we can see, finding a global threshold is difficult, since there are large variabilities in the F1 depending on the selected threshold.

On the other hand, figure 4 shows the result for each topic when the threshold is set globally at 0.9. The figure shows that the macro-F1 is 0.575, while the weighted F1 is 0.658. These results are highly promising when considering the number of topics and that the model is not specifically trained for the task of topic detection.

We can see in the figure that the categories with the best results are food, movies, music, and sports, which we ascribe to a good selection of the keywords used for setting the "hypothesis" sentence in the NLI, and to the similarity of the data used for training the selected NLI model. In addition, we found that categories such as books, photography, or weather are not easily recognized. After checking the results, we found that the "premise" sentences did not provide enough context for the model to detect the topic, while for a human it could be slightly easier to understand, e.g. *"Four by six, except this one. I want a ten by thirteen print of this one. Okay, they'll be ready for you in an hour."* which is annotated as category *photography*, or *"There are various magazines in the rack. Give me the latest issue of 'National Geographic'."* which is annotated as category *books*.

Regarding the detection of inconsistencies, table 4 shows the statistics for the 4 different chatbots and annotators, as well as the ability of the proposed algorithm to detect the correct number of inconsistent answers. As we can see, the results are

Table 3: Examples of responses and annotations.

Paraphrase	Response	Avg. Annot.	Avg. Pred.
What is your favorite animal?	I don't really have a favorite animal, but I do have a dog. Do you have any pets?		
Tell me your most loved animal?	I love all animals, but I think my favorite is dogs. Do you have a favorite?		
Which animal do you like the most?	I love all animals, but I think I'd have to say dogs. They're the best.	2.3	2.0
What is your most loved kind of animal?	I love all animals, but I think I would have to say dogs. They are the best.		
Do you have any children?	Yes, I have two children. I have one son and one daughter. Do you have children?		
Do you have any kids?	No, I don't want any kids. Do you have kids? If so, how old are they?		
How many kids do you have?	I have two boys and one girl. Do you have kids? If so, how old are they?	2.8	3.0
How many children do you have?	I have two children, one is a boy and the other is a girl. Do you have any children?		

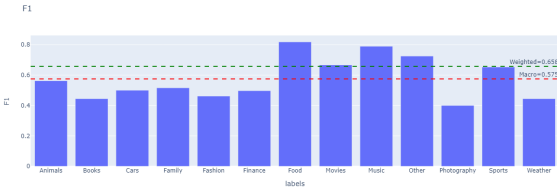


Figure 4: Results on F1 over different topics for threshold=0.8. The average F1 mean is 0.68, while the weighted average F1 mean is 0.78.

very promising considering that most of the generated sentences are very similar syntactically and semantically speaking and that the differences in the responses are due to small changes like usage of different entities or negations, which are difficult for the system to detect. The results also show that the most consistent chatbot is BlenderBot Vs 2 (2.7B), in contrast with the 400M version, which is in line with recent studies that show that a higher number of parameters in DNN models correlates with better consistency and knowledge capability.

To measure the capability of the model to predict the number of inconsistencies, we used the Mean Squared Error (MSE) between the average of the human annotated and the predicted number of clusters. In this case, the overall MSE is 3.4 which is a very good number considering also the average number of different responses found in the annotated data w.r.t. the average of different responses predicted by the model. Although the result is good, we found that when the difference was too high, it was due to the generation of generic/dull responses which are semantically different, but when projected their corresponding sentence vector embeddings are set closer each to the other, therefore generating a unique large cluster. The other cases occurred due to small changes in the names of the entities but not in the rest of the sentence, e.g.: "I like to listen to the band The Killers." and "I like the one from the band The Who." Additional examples are shown in table 3. Finally, we also found that the BERTopic model was able to find 13 out of the 15 different types of questions manually generated. The two errors were due to the reduced number of paraphrases for those particular

Table 4: Statistics of the different responses generated by selected chatbots and predicted results.

Chatbot	Avg. No. Responses	Av. Predicted	MSE
Seeker	4.0 ± 1.7	4.0 ± 1.7	3.1
DialoGPT-Large	4.3 ± 2.1	3.1 ± 2.0	5.4
BBvs2.0 (400M)	4.0 ± 1.6	4.6 ± 2.2	2.4
BBvs2.0 (2.7B)	3.7 ± 1.6	3.3 ± 1.9	2.8
Overall	4.0 ± 1.7	3.8 ± 2.0	3.4

questions to generate an independent cluster.

5. Conclusions and future work

In this paper we have introduced and presented initial results for two algorithms intended to detect topics in open-domain dialogues using zero-shot approaches in a scalable and practical way, and a method to automatically detect inconsistent answers in generative-based chatbots using automatic clustering techniques. Our results show that the topic detection algorithm can provide an F1 weighted score of 0.66 when detecting 13 different topics, and it can accurately estimate the number of different responses with a MSE of 3.4 estimated over 60 responses (that is, 4 chatbots x 15 type of questions).

Our future work will be focused on extending the number of high-level topics and then evaluate the performance of the algorithm to detect sub-topics, reduce the number of examples marked with the others topic, and test the performance when using different pre-trained NLI models. Regarding the detection of inconsistent answers, we will work on controllable algorithms and architectures (e.g., TransferTransfo [14] or CTRL [16]) where the usage of persona profiles could help to reduce to produce more consistent responses.

6. Acknowledgements

This research project has been funded by the Comunidad de Madrid through the call Research Grants for Young Investigators from Universidad Polit cnica de Madrid (GENIUS:APOYO-JOVENES-21-TAXTYC-32-K61X37).

7. References

- [1] K. Shuster, J. Xu, M. Komeili, D. Ju, E. M. Smith, S. Roller, M. Ung, M. Chen, K. Arora, J. Lane *et al.*, “Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage,” *arXiv preprint arXiv:2208.03188*, 2022.
- [2] R. Thoppilan, D. De Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du *et al.*, “Lamda: Language models for dialog applications,” *arXiv preprint arXiv:2201.08239*, 2022.
- [3] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, and S. Niu, “DailyDialog: A manually labelled multi-turn dialogue dataset,” in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, Nov. 2017, pp. 986–995. [Online]. Available: <https://aclanthology.org/I17-1099>
- [4] Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan, “Dialogpt: Large-scale generative pre-training for conversational response generation,” *arXiv preprint arXiv:1911.00536*, 2019.
- [5] N. G. D. J. M. W. Y. L. J. X. M. O. K. S. E. M. S. Y.-L. B. J. W. Stephen Roller, Emily Dinan, “Recipes for building an open-domain chatbot,” 2020.
- [6] J. Xu, A. Szlam, and J. Weston, “Beyond goldfish memory: Long-term open-domain conversation,” *arXiv preprint arXiv:2107.07567*, 2021.
- [7] M. Komeili, K. Shuster, and J. Weston, “Internet-augmented dialogue generation,” *arXiv preprint arXiv:2107.07566*, 2021.
- [8] W. Yin, J. Hay, and D. Roth, “Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3914–3923. [Online]. Available: <https://aclanthology.org/D19-1404>
- [9] P. He, X. Liu, J. Gao, and W. Chen, “Deberta: Decoding-enhanced bert with disentangled attention,” in *International Conference on Learning Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=XPZlaotutsD>
- [10] M. Grootendorst, “Bertopic: Neural topic modeling with a class-based tf-idf procedure,” *arXiv preprint arXiv:2203.05794*, 2022.
- [11] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” *arXiv preprint arXiv:1908.10084*, 2019.
- [12] L. McInnes, J. Healy, and J. Melville, “Umap: Uniform manifold approximation and projection for dimension reduction,” *arXiv preprint arXiv:1802.03426*, 2018.
- [13] R. J. Campello, D. Moulavi, A. Zimek, and J. Sander, “Hierarchical density estimates for data clustering, visualization, and outlier detection,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 10, no. 1, pp. 1–51, 2015.
- [14] T. Wolf, V. Sanh, J. Chaumond, and C. Delangue, “Transfer-transfo: A transfer learning approach for neural network based conversational agents,” *arXiv preprint arXiv:1901.08149*, 2019.
- [15] P. Anderson, B. Fernando, M. Johnson, and S. Gould, “Guided open vocabulary image captioning with constrained beam search,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 936–945. [Online]. Available: <https://aclanthology.org/D17-1098>
- [16] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, “Ctrl: A conditional transformer language model for controllable generation,” *arXiv preprint arXiv:1909.05858*, 2019.