Evaluating the MuMe Dialogue System with the IDIAL Protocol

Aureliano Porporato
Università degli Studi di Torino
aureliano.porporato@unito.it

Alessandro Mazzei
Università degli Studi di Torino
alessandro.mazzei@unito.it

Rosa Meo
Università degli Studi di Torino
rosa.meo@unito.it

Daniele P. Radicioni
Università degli Studi di Torino
daniele.radicioni@unito.it

Abstract

English. In this paper we describe the implementation of the MuMe dialogue system, a task-based dialogue system for a car sharing service, and its evaluation through the IDIAL protocol. Finally we report some comments on this novel dialogue system evaluation method.[1]

Italiano. *In questo lavoro descriviamo l'implementazione del sistema di dialogo MuMe, realizzato per un sistema di car sharing, e la sua valutazione attraverso il protocollo IDIAL. Infine, offriamo alcuni commenti su questo nuovo metodo per la valutazione di sistemi di dialogo.*

1 Introduction

The interest in dialogue systems is on the rise in the NLP community (McTear et al., 2016), under the strong demand for the introduction of a natural and effective user interaction in applications, like in the customer care domain (Hu et al., 2018). A related and central issue is the evaluation of such systems. In this setting, it is largely known that most evaluation metrics that come from machine translation and compare a model generated response to a single target response, exhibit a poor correlation with the human judgement (Liu et al., 2016).

In this paper we briefly illustrate a task-oriented dialogue system called MuMe (from "MUoversi MEglio", "travelling better" in English language), and examine how far the evaluation protocol IDIAL (Cutugno et al., 2018) is helpful in its assessment. IDIAL is composed by a usability evaluation (done by a group of users) and by an evaluation of the robustness of the dialog model based on the linguistic variations of the successful interactions with the users. The application being tested is a prototype dialogue system that we developed for the reservation of electric vehicles in the context of a car sharing service. A user must be able to interact with the system, to specify when and where s/he wants to leave and which sort of vehicle is needed. While there are some services and frameworks dedicated to the development of machine-learning-based dialogue systems, like Google Dialogflow[2] or the open source Rasa[3] frameworks, the lack of Italian dialogue corpora in the specific domain of car sharing reservations (see, e.g., Serban et al. (2018)) and the impossibility on our part to recruit a number of people large enough for the creation of such a corpus, forced us to choose a different solution: we developed a simpler and less data-reliant rule-based system, based on slot-filling semantics. Moreover, the decisions made by this kind of systems can be tracked throughout the computation, thereby resulting in the advantage of being quite explainable. This is a desirable feature, since it simplifies the debugging and the maintenance of the routines, and allows an easier extension of the system to meet additional requirements.

This paper is mostly concerned with the evaluation of the MuMe system. The structure of the paper is as follows. After surveying on related work (Section 2), we briefly introduce the overall architecture and the main components of the MuMe dialogue system (see Section 3); we evaluate MuMe by using the IDIAL protocol, and employ MuMe experimentation as a case study for giving feedback on the IDIAL protocol itself (Section 4); finally, in the final Section we briefly recap the main contributions of the paper, and point to ongoing and future work.

[2] https://dialogflow.com/

[3] https://rasa.com/

2 Related Work

The pioneering work of (Bobrow et al., 1977) proposed the frame-based architecture that most of task-based dialogue systems implement. The basic idea is to abandon the demanding goal to have a genuine logic representation of the dialog meaning and adopt a simpler slot-filling semantics. In some sense, the event-entities representation of the modern neural-based dialogue system frameworks can be seen as an ultimate evolution of that simplification idea. Aust et al. (1995) presented a rule-based system to some extents similar to ours in its purpose and structure, created for a train-seat reservation project. This system has to grasp the names of cities, train stations, dates and times, and it is able to perform quite sophisticated temporal information processing. Further rule-based systems are reviewed in the survey by (Abdul-Kader and Woods, 2015).

A different class of dialogue systems are based on neural networks. A survey on this class of systems can be found in (Mathur and Singh, 2018).

Regarding the evaluation of dialogue systems, the work by (Bohlin et al., 1999) proposes the Trindi Tick-list, a wish list of the desired dialogue behaviour and features specified as a checklist of ”yes-no” questions. As regards this approach, Braunger and Maier (2017) argue that standardised evaluation models do not enable a complete evaluation of a dialogue system. Rather, they suggest that such evaluation must take into account the *natural flow* of the interaction between the user and the system itself; such measure involves many language- and user-dependant factors, such as the length of the user utterances. Such principles were tested in human-computer vocal interactions occurring on board of vehicles. Further information on dialogue systems evaluation methods can be found in the survey by Deriu et al. (2019).

3 The MuMe system architecture

In Figure 1 we depicted the basic architecture of the MuME dialogue system. The information flow starts from a sentence typed by the user: this sentence is handled by the OpenDial system (see Section 3.1) which plays both the role of the dialogue manager and of the system orchestrator. So, the sentence is syntactically parsed and semantically analyzed by an IE module (see Section 3.2). At this point, the result of the processing is converted into a slot-filling form. When control returns to OpenDial, it generates an answer and returns it to the user on the basis of a dialogue control strategy (see Section 3.3).

3.1 The OpenDial Dialogue Manager

The main component of our software architecture is the OpenDial open source framework for dialogue management (Lison, 2015). The system, that was designed for speech interaction, adopts the *information state* approach for modelling the state of the dialogue (Traum and Larsson, 2003), that is a collection of variables representing the actual state of the system. The transition between states, i.e. the change of the variables values, is governed by the activation of a set of ”if-then-else” rules on input values as well as on the variation of some variables. Indeed, OpenDial uses these rules when it models the sub-tasks of user utterance understanding, the dialogue management and the response generation. Moreover the integration of the system with external tools is simple. We exploited this capability in MuMe since for language understanding we used a module based on an external parser (see below). Additionally, the OpenDial framework implements some statistical-based techniques to deal with uncertainty. This is a way to learn interaction models from existing dialogues. This feature is particularly important for speech based dialogue systems where uncertain information arises from automatic speech recognition. However, at this stage of the MuMe project, we did not use this feature since we were working on written texts only.

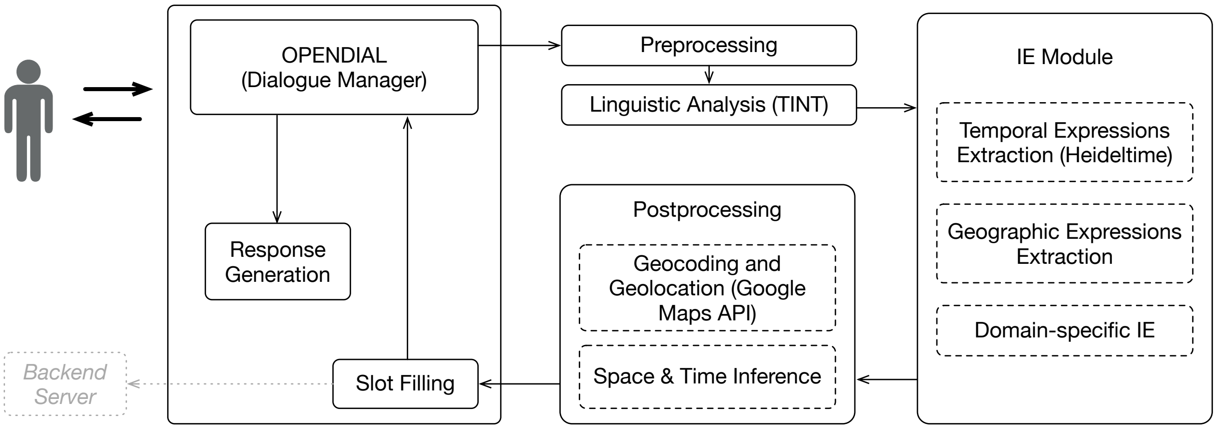
3.2 Parsing and Information Extraction

In order to assign semantic roles to the entities in the dialogues, we decided to use a syntactic parser on the text inserted by the user.

As our main parsing module we used Tint (The Italian NLP Tool) (Palmero Aprosio and Moretti, 2016), a framework modeled on Stanford CoreNLP (Manning et al., 2014). Tint performs some fundamental processing of user utterances, such as dependency parsing, Named Entities Recognition and the extraction of Temporal Expressions. In particular, the tasks are executed by interfacing with external tools.

For the recognition of temporal expressions (such as dates and times), Tint integrates the services provided by HeidelTime (Strötgen and Gertz, 2013). HeidelTime allows the extraction

Figure 1: The schematic architecture of the MuMe dialogue system.



of various sorts of temporal expressions in various languages, including the Italian language, and represents them in the standard TIMEX3 format.

For the treatment of geographic expressions, Tint is interfacing with the Nominatim wrapper.[4] However, this (free and open source) service performs poorly in geocoding (i.e., in searching the GPS coordinates of a given address). As a consequence we decided to use the Google Maps API[5], which provides for better performances. Indeed, Maps offers an API for address autocomplete, once this information piece has been isolated from the rest of the sentence, and for geolocation (i.e., searching the coordinates of the user), too.

3.3 Dialogue Control Strategy

The simple control strategy implemented, that governs the *moves* of the dialogue, is based on the fulfillment of a number of mandatory slots in the domain-specific slot-filling semantics adopted for the car reservation domain.

In particular, the mandatory slots are the *start date*, the *start time* and *start stall* (which encodes the start position). Indeed, the simplest reservation in MuMe needs only of these pieces of information: a person reserves a standard car, starting at a specific time of a specific day from a specific stall, and will return the car in the same stall without the need to specify the return date and time.

However, more complex reservations need more information, that are encoded in the non-mandatory slots of *end date*, *end time*, *end stall* and *vehicle type*. For example, the user can choose between three types of vehicles, but if the kind of vehicle is not specified, the system assigns a default 'economy car' to the *vehicle type* slot.

The MuMe system adopts a mixed initiative for dialogue handling. Although the dialogue is overall system-driven, the user starts the conversation by possibly providing some initial information. A richer initial information is expected to result in a shorter dialogue interaction. Indeed, a design goal of the MuMe system is to produce a dialogue as short as possible. For this reason, also in the subsequent interactions, if the user gives various pieces of information in a single utterance, the system can extract all such information and is able to assign each filler to the corresponding slot, thus avoiding further unnecessary questions.

When the user begins the interaction with the MuMe system, the system replies with a welcome message, and with a general question aiming at encouraging the user to start the interaction in the most natural way.

In order to give more details on the control strategy, we consider now the following running example and its processing in MuMe (see Figure 1):

(*it*) "**User:** Ho bisogno di un'auto domani per andare in via Pessinetto"
(*en*) "**User:** I need a car tomorrow to go in Pessinetto street"[6]

The Information Extraction phase detects a date (through HeidelTime) and an address (extracted through a basic set of custom rules) in the user

[4] `http://nominatim.org/`.

[5] `https://cloud.google.com/maps-platform/`.

[6] The English version of the user and system sentences are given for clarity. The system is available in Italian language only.

sentence. By means of other rules that check the shape of the dependency tree (obtained through Tint), date and address are labelled as *start date* and *end address*. Particularly relevant in this case is the verb "andare" ("to go"), that signals that the following address is where the user wants to arrive and not a starting point. In the post-processing phase some additional information can be inferred, like the value of the *start address*, left unspecified by the user: it can be selected by retrieving the GPS coordinates of the address by means of the Google Maps API. Once the user's current location has been identified, the nearest stall is selected as the *start stall*.

At the end of this processing, the system successfully filled the *start address*, *start stall*, *end address*, *end stall* and *start date* slots. Some mandatory slots are still left unfilled, such as the *start time*, so that the system will ask the user to provide the missing information. As a consequence, the response of the system will be a question selected from a fixed list based on unfilled slots: in this specific example, the system will continue asking for the departure time.

At the end of the filling-phase of the mandatory-slots, the systems gives the user the possibility to modify the request and to correct possible errors and misunderstandings. The slot-filling values will be sent to a dedicated server for the finalization of the reservation.

4 Evaluation

In order to have a first preliminary evaluation of the MuMe system, we applied the Trindi Tick-list protocols, that is a set of "yes-no" questions concerning specific capabilities of the developed system (Bohlin et al., 1999). While this simple questionnaire is helpful in the development phase, since it is able to give a measure of the system limits, it is not suitable to completely evaluate the actual experience of the user. At this stage of development, the MuMe system has a Trindi score of six over twelve with respect to the (original) list. Among the six features not yet implemented, there are complex tasks, such as the management of the *help* and *non-help* sub-dialogues, dealing with negative information, and dealing with noisy input.

In the rest of the Section, we report the results obtained by applying the IDIAL evaluation protocol to the current version of the MuMe system, which is split in a questionnaire concerning the user experience (Section 4.1), and a number of *stress tests* concerning the linguistic robustness of the system (Section 4.2).

4.1 IDIAL User Evaluation

A group of 5 subjects (3 males, 2 females, 19, 22, 25, 26 and 61 years old) were recruited for the evaluation task by personal invitation and without rewards. After a brief oral description of the domain and of the basic mechanisms of interaction with the system, each user was asked to generate 7 complete dialogues with the system in a controlled environment. We asked the users to simulate the process of reserving a car without other specific constraints.

In Table 1 we report the ten questions of the IDIAL user test with the average score, obtained by using a Likert scale based on five points.[7] Note that the questions 3, 4, 7 and 10 have been designed to evaluate the effectiveness of the dialogue system, while questions 1 and 2 regard the system efficiency.[8]

4.2 IDIAL Stress Tests

The second evaluation stage in the IDIAL protocol consists in a set of linguistic stress tests. We selected 5 dialogues (one for each user) among those successfully completed[9] during the user evaluation stage. Following the IDIAL protocol, we modified one sentence in each dialogue, once for each test, as illustrated in (Cutugno et al., 2018), and repeated the dialogue with the modified sentence. The results are reported in Table 2.

Note that we could not perform three stress tests for distinct reasons. We could not perform the ST-8 test, regarding active-passive alternation, because the users almost always used intransitive verbs (like "andare" ["to go"] and "partire" ["to depart"]). We could not perform the ST-9 test, concerning adjective-noun alternation, since the users used a very few adjectives (like vehicle types modifiers "lussuosa" ["luxurious"]), and no adjectives have been used in a successful dialogue. Fi-

[7] We used the Italian version of the questionnaire, found in the Appendix A of `https://tinyurl.com/yxngqkx4`, but for sake of readability in Table 1 we report the English version.

[8] The answers of each subjects are available at `https://tinyurl.com/y6nruwon`

[9] We considered an interaction as 'successfully completed' if the system recognized and processed correctly all the data given by the user.

N	Sentence	Evaluation
1	The system was efficient in accomplishing the task.	3.2 (0.45)
2	The system quickly provided all the information that the user needed.	3.6 (0.55)
3	The system is easy to use.	3.6 (1.52)
4	The system is awkward when the user interacts with a non-standard or unexpected input.	2.8 (0.84)
5	The user is satisfied by his/her experience.	3.0 (0.00)
6	The user would recommend the system.	3.2 (0.84)
7	The system has a fluent dialogue.	2.8 (0.84)
8	The system is charming.	3.4 (0.90)
9	The user enjoyed the time s/he spent using the software.	3.8 (0.84)
10	The system is flexible to the user's needs.	3.6 (0.55)

Table 1: IDIAL user ratings of their experience: the average scores are provided on a 1-5 Likert scale with standard deviation, in parentheses.

nally, we could not perform the ST-10 test, concerning anaphora resolution, since at the actual stage of development the system never asks the user to pick an answer from a set of options.

4.3 Discussion

With respect to the user evaluation test, a number of considerations arise from scores. The main issue pointed out by the users during the evaluation phase is the difficulty in grasping when and why the system misunderstood (or lost) some pieces of information, thereby resulting in a relatively poor evaluation score for the fluency of the system (average score of 2.8). The lack of feedback due to the too simple way we used to generate system responses has even worsened this problem, leading the user to repeat the same mistake more than once. The standard deviation of the evaluations given to question 3 shows the high subjectivity of the user experiences with the system, and points out the necessity to equip the system with some form of *user model* to account for the expectation of different kinds of users. It is worth noting that 4 out of 5 users explicitly stated (in private conversations after the evaluation phase) that they expected longer interactions. Also, they expected to receive more questions by the system, challenging our assumption on the length of dialogues. However, two of the same users added that 7 interactions are enough to evaluate the system.

Stress Test		Passed
Spelling Substitutions		
ST-1	Confused words	60%
ST-2	Misspelled words	40%
ST-3	Character replacement	80%
ST-4	Character swapping	60%
Lexical Substitutions		
ST-5	Less frequent synonyms	60%
ST-6	Change of register	40%
ST-7	Coreference	100%
Syntactic Substitutions		
ST-8	Active-Passive alternation	–
ST-9	Nouns-adjectives inversion	–
ST-10	Anaphora resolution	0%
ST-11	Verbal-modifier inversion	80%

Table 2: IDIAL stress test results.

With respect to the evaluation of the stress tests, we can say that the sentences provided by the users during the interaction with the system, were often very short and scarcely usable from the viewpoint of the IDIAL stress tests (especially those concerned with lexical and syntactic aspects). Another source of problems are *typos*, in particular in expressions regarding time and addresses. While our system seems quite robust to this kind of errors (see the first 4 rows of Table 2), it is difficult to automatically deal with them without some domain specific knowledge on their occurrence and some correction strategies.

As a final note, we want to report some comments given by the users about the questionnaire. Two users expressed some doubts on the interpretation of question 8 and in general all of them found difficult to assign a meaningful evaluation to it. For example, some of the users interpreted the question as regarding the lack of a GUI, absent in our prototype. We think that the ambiguity of the sentence explains the slightly higher standard deviation for that question in respect to others. Other comments include the lack of diversity between some sentences (like questions 1 and 5, often judged as redundant), and the inadequacy of this Likert scale to evaluate some questions, like 5 and 9: they consider a more subjective scale ("poco" ["few"] - "molto" ["a lot"]) more appropriate, perceiving the whole process as a single experience.

While the linguistic stress test can be a valuable tool for the improvement of the system, the questionnaire concerning the user experience should be revised for addressing some critics that we collected. In particular, the questionnaire should be augmented with more specific questions.

5 Conclusion and Future Work

We presented the MuMe system, a prototype of a rule-based dialogue system and its evaluation through the IDIAL method.

Since the MuMe project is still in development, there is much room for improvement. The most pressing problem to be addressed in future development is the generation of a response more meaningful to the user. The application of a natural language generation pipeline for Italian (e.g. (Mazzei et al., 2016; Mazzei, 2016; Conte et al., 2017; Ghezzi et al., 2018)) could help to these ends.

Acknowledgments

This project has been partially supported by the MuMe Project (Muoversi Meglio), funded by the Piedmont Region and EU in the frame of the F.E.S.R. 2014/2020.

References

[Abdul-Kader and Woods2015] Sameera A Abdul-Kader and JC Woods. 2015. Survey on chatbot design techniques in speech conversation systems. International Journal of Advanced Computer Science and Applications, 6(7).

[Aust et al.1995] Harald Aust, Martin Oerder, Frank Seide, and Volker Steinbiss. 1995. The philips automatic train timetable information system. Speech Communication, 17(3-4):249–262.

[Bobrow et al.1977] Daniel G. Bobrow, Ronald M. Kaplan, Martin Kay, Donald A. Norman, Henry Thompson, and Terry Winograd. 1977. Gus, a frame-driven dialog system. Artif. Intell., 8(2):155–173, April.

[Bohlin et al.1999] Peter Bohlin, Johan Bos, Staffan Larsson, Ian Lewin, Colin Matheson, and David Milward. 1999. Survey of existing interactive systems. Deliverable D1, 3:1–23.

[Braunger and Maier2017] Patricia Braunger and Wolfgang Maier. 2017. Natural language input for in-car spoken dialog systems: How natural is natural? In Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, pages 137–146.

[Conte et al.2017] Giorgia Conte, Cristina Bosco, and Alessandro Mazzei. 2017. Dealing with italian adjectives in noun phrase: a study oriented to natural language generation. In Proceedings of the Fourth Italian Conference on Computational Linguistics (CLiC-it 2017), Rome, Italy, December 11-13, 2017., December.

[Cutugno et al.2018] Francesco Cutugno, Maria Di Maro, Sara Falcone, Marco Guerini, Bernardo Magnini, and Antonio Origlia. 2018. Overview of the evalita 2018 evaluation of italian dialogue systems (idial) task. In EVALITA@ CLiC-it.

[Deriu et al.2019] Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2019. Survey on evaluation methods for dialogue systems. arXiv preprint arXiv:1905.04071.

[Ghezzi et al.2018] Ilaria Ghezzi, Cristina Bosco, and Alessandro Mazzei. 2018. Auxiliary selection in italian intransitive verbs: A computational investigation based on annotated corpora. In Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), pages 1–6, Berlin. CEUR.

[Hu et al.2018] Tianran Hu, Anbang Xu, Zhe Liu, Quanzeng You, Yufan Guo, Vibha Sinha, Jiebo Luo, and Rama Akkiraju. 2018. Touch your heart: A tone-aware chatbot for customer care on social media. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, page 415. ACM.

[Lison2015] Pierre Lison. 2015. A hybrid approach to dialogue management based on probabilistic rules. Computer Speech & Language, 34(1):232 – 255.

[Liu et al.2016] Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. arXiv preprint arXiv:1603.08023.

[Manning et al.2014] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, pages 55–60.

[Mathur and Singh2018] Vinayak Mathur and Arpit Singh. 2018. The rapidly changing landscape of conversational agents. arXiv preprint arXiv:1803.08419.

[Mazzei et al.2016] Alessandro Mazzei, Cristina Battaglino, and Cristina Bosco. 2016. Simplenlg-it: adapting simplenlg to italian. In Proceedings of the 9th International Natural Language Generation conference, pages 184–192, Edinburgh, UK, September 5-8. Association for Computational Linguistics.

[Mazzei2016] Alessandro Mazzei. 2016. Building a computational lexicon by using SQL. In Pierpaolo Basile, Anna Corazza, Francesco Cutugno, Simonetta Montemagni, Malvina Nissim, Viviana Patti, Giovanni Semeraro, and Rachele Sprugnoli, editors, Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016), Napoli, Italy, December 5-7, 2016., volume 1749, pages 1–5. CEUR-WS.org, December.

[McTear et al.2016] Michael McTear, Zoraida Callejas, and David Griol. 2016. The Conversational Interface: Talking to Smart Devices. Springer Publishing Company, Incorporated, 1st edition.

[Palmero Aprosio and Moretti2016] A. Palmero Aprosio and G. Moretti. 2016. Italy goes to Stanford: a collection of CoreNLP modules for Italian. ArXiv e-prints, September.

[Serban et al.2018] Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. 2018. A survey of available corpora for building data-driven dialogue systems: The journal version. Dialogue & Discourse, 9(1):1–49.

[Strötgen and Gertz2013] Jannik Strötgen and Michael Gertz. 2013. Multilingual and cross-domain temporal tagging. Language Resources and Evaluation, 47(2):269–298.

[Traum and Larsson2003] David Traum and Staffan Larsson. 2003. The Information State Approach to Dialogue Management. In Current and New Directions in Discourse and Dialogue, pages 325–353. Springer.