Overview of the EVALITA 2018 Spoken Utterances Guiding Chef's Assistant Robots (SUGAR) Task

Maria Di Maro
Università degli Studi
di Napoli 'Federico II'
Department of Humanities
maria.dimaro2@unina.it

Antonio Origlia
Università degli Studi
di Napoli 'Federico II'
URBAN/ECO Research Center
antonio.origlia@unina.it

Francesco Cutugno
Università degli Studi
di Napoli 'Federico II'
Department of Electrical
Engineering and Information
Technology
cutugno@unina.it

Abstract

English. The SUGAR task is intended to develop a baseline to train a voice-controlled robotic agent to act as a cooking assistant. The starting point will be therefore to provide authentic spoken data collected in a simulated natural context from which semantic predicates will be extracted to classify the actions to perform. Three different approaches were used by the two SUGAR participants to solve the task. The enlightening results show the different elements of criticality underlying the task itself.

Abstract

Italiano. *Con il task SUGAR si intende sviluppare una baseline per addestrare un aiuto-cuoco robotico controllato da comandi vocali. Il punto di partenza sarà, pertanto, quello di fornire materiale vocale autentico raccolto in un contesto naturale simulato da cui saranno estratti i predicati semantici al fine di classificare le azioni da eseguire. Tre 16 diversi approcci sono stati utilizzati dai due partecipanti per risolvere il task. I risultati mostrano i veri livelli di criticità che soggiaciono il task stesso.*

1 Introduction

In the last few years, Human-Machine interaction systems have been in the spotlight, as far as computer science and linguistics are concerned, resulting in many applications such as Virtual Assistants and Conversational Agents (Cassell et al., 2000; Cauell et al., 2000; Dzikovska et al., 2003; Allen et al., 2007). The possibility to use such Artificial Intelligence technologies in domestic environments is increasingly becoming a reality (Darby, 2018; Ziefle and Valdez, 2017). In order to ensure the future possibility of making such systems even more intelligent, further researches are needed. As it has been the case with Apple SIRI and Google Assistant technologies, recent approaches transformed the former dialogue systems in direct action actuators, removing or reducing, as much as possible, clarification requests that may arise in presence of ambiguous commands. In this view, Spoken Language Understanding (SLU) is nowadays one of the major challenge of the field. Making a system able to truly understand the intention of the speaker in different contexts and react correctly, even in presence of Automatic Speech Recognition (ASR) errors, is the ultimate purpose to pursue in the field. In this context, the application of various semantic annotation schemata and criteria of knowledge modelling are of particular interest. Among different techniques used to model the interpretation process we cite: (i) *semantic-frame parsing*, where the frame classification with the recognition of its attribute can improve the information retrieval process for a more precise domain specific answer (Wang, 2010); (ii) *semantic interpretation*, for which semantic-syntactic trees can be used to extract basic semantic units and their relationships (Miller et al., 1996); (iii) *intent classification*, for which structures comprising generic predicates working as semantic primitives (Wierzbicka, 1972) and domain-dependent arguments can be used to represent a specific intent (Tur and Deng, 2011; Serban et al., 2018). With this particular task, we propose a possible framework for semantic classification to be tested, recurring to state-of-the-art SLU systems participating to the EVALITA-SUGAR challenge (Caselli et al., 2018).

2 Corpus Collection and Description

In the SUGAR challenge, the underlying task is to train a voice-controlled robotic agent to act as

Figure 1: 3D Recontruction of Bastian in his Kitchen. On the wall, the television showing frames of video recipes, from which users could extract actions to utter as commands

a cooking assistant. For this purpose, a training corpus of annotated spoken commands was collected. To collect the corpus, we designed a 3D virtual environment reconstructing and simulating a real kitchen where users could interact with a robot (named Bastian) which received commands to be performed in order to accomplish some recipes. User's orders were inspired by silent cooking videos shown in the 3D scene, thus ensuring the naturalness of the spoken production. Videos were segmented into elementary portions (frames) and sequentially proposed to the speakers who uttered a single sentence after each seen frame. In this view, speakers watched at video portions and then gave instructions to the robot to emulate what seen in the frame (Figure 1). The collected corpus then consists of a set of spoken commands, whose meaning derives from the various combination of actions, items (i.e. ingredients), tools and different modifiers.

Audio files were captured in a real acoustic environment, with a microphone posed at about 1 mt of distance from the speakers. The resulting corpus contains audio files for each speaker. These files were then segmented into sentences representing isolated commands. Orthographic transcriptions of the audio files were not be provided. Consequently, participants could use whichever ASR they prefer, whose performance was not under assessment. Nevertheless, the developed systems were expected to be strongly efficient despite the possible ASR deficiencies. Each resulting audio file was paired to a textual one containing the corresponding action annotation.

Training set Actions are represented as a finite set of generic predicates accepting an open set of parameters. For example, the action of *putting* may refer to a pot being placed on the fire

$$put(pot, fire)$$

or to an egg being put in a bowl

$$put(egg, bowl)$$

The annotation process resulted in determining the optimal action predicate corresponding to each command.

The training set consists of audio files and predicate description pairs, where the predicate serves as an interpretation of the intention to be performed by the robot. For these scenarios, the audio files are always mapped on a single interpretative predicate. The training set consists of 1721 utterances (and therefore 1721 audio files) produced by 36 different speakers annotated by two linguistic experts. The action templates, which have been inferentially defined through the video collection, are shown in Table 1, where [ ] indicates a list of ingredients, / the alternative among possible arguments, *quantity* and *modality* are not mandatory arguments, and * is used when the argument is recoverable from the context (i.e. previous instantiated arguments, which are not uttered, not even by means of clitics or other pronouns) or from the semantics of the verb. For instance,

friggere (fiori)[1]

is represented as

*aggiungere(fiori, *olio*)*[2]

because *olio* (En. *oil*) is implicitly expressed in the semantics of the verb *friggere* (En. *to fry*) as an instrument to accomplish the action. Among other phenomena, it is worth mentioning the presence of actions paired with templates, even when the syntactic structure needs a reconstruction, as in

coprire(ciotola, pellicola)[3]

which is annotated with the generic template as

mettere(pellicola, ciotola)[4].

[1] *fry(flowers)*
[2] *add(flowers, *oil*)*
[3] *cover(bowl, wrap)*
[4] *put(wrap, bowl)*

Predicate	Arguments
prendere	quantità, [ingredienti]/recipiente
aprire	quantità, [ingredienti], recipiente
mettere	quantità, utensile/[ingredienti], elettrodomestico, modalità
sbucciare	quantità, [ingredienti], utensile
schiacciare	[ingredienti, utensile
passare	[ingredienti], utensile
grattare	[ingredienti], utensile
girare	[ingredienti], utensile
togliere	utensile/prodotto, elettrodomestico
aggiungere	quantità, [ingredienti], utensile/recipiente/ elettrodomestico/[ingredienti], modalità
mescolare	[ingredienti], utensile, modalità
impastare	[ingredienti]
separare	parte/[ingredienti],ingrediente/utensile
coprire	recipiente/[ingredienti], strumento
scoprire	recipiente/[ingredienti]
controllare	temperatura, ingrediente
cuocere	quantità, [ingredienti], utensile, modalità

Table 1: Italian Action templates

In other cases, the uttered action represents the consequence of the action reported in the template, as in

separare(parte, fiori)[5]

and

pulire(fiori)[6],

or

mescolare([lievito, acqua])[7]

and

sciogliere(lievito, acqua)[8].

The argument order does not reflect the one in the audio files, but the following:

azione(quantità[9], oggetto, complemento, modalità)[10]

The modality arguments are of different types and the order is *adverb*, *cooking modality*, *temperature* and *time*.

Test set The test set consists of about 572 audio files containing uttered commands without annotations. Task participants were asked to provide, for each target command, the correct action predicate following the above-described format. Although single actions are of the same kind of the ones found in the training set and in the template file, the objects, on which such actions may be applied to, vary (i.e. different recipes, ingredients, tools...). Participants have been evaluated on the basis of correctly interpreted commands, represented in the form of predicates.

The task could be carried out either by using only the provided linguistic information of the training set or by means of other external linguistic tools, such as ontologies, specialised lexicons, and external reasoners.

3 Evaluation Protocol

The evaluation protocol covered the following possibilities:

- The proposed system correctly detects the requested action and all its parameters;
- The proposed system asks for repetition;
- The proposed system correctly detects the requested action but it assigns wrong parameters;
- The proposed system misses the action.

The possibility of asking for repetitions is left to participants to avoid forcing them to provide an answer in uncertain conditions. In this case, the evaluation protocol would assign a weaker penalisation than the one considered for missing the arguments or the action. The collected corpus did not, however, contain situations in which the system asks for repetitions.

The designed evaluation procedure outputted the following pieces of information:

1. an id comprising the listing number of the recognised predicate and the number of actions, in case of pluri-action predicates (1_1, 1_2, 2_1, etc);
2. a Boolean value (1: True, 0: False) indicating if the predicate has been recognised; when the predicates were not recognised, even the argument number is set on 0;
3. the number of expected arguments as indicated in the reference annotation files[11];

[5] *separate(part, flowers)*
[6] *clean(flowers)*
[7] *stir([yeast, water])*
[8] *melt(yeast, water)*
[9] The quantity always precedes the noun it is referred to. Therefore, it can also come before the complement
[10] action(quantity, object, complement, modality)
[11] The reference annotation files were annotation files created for the test set although not being made available

4. the distance between the participating systems' output file and the reference file computed by means of the Levenshtein distance (Levenshtein, 1966); the higher the computed distance in the output was, the more mistakes the system had detected;

5. the number of arguments for which the system asked for repetition.

Suppose the action in reference file is annotated as

1; [prendere(500 g, latte), aggiungere(latte, pentola)][12]

and the recognition procedure outputs

1; prendere(500 g, panna)[13]

instead of returning the following result, indicating a correct recognition

1_1 *(first predicate)*
(1, 2, 0, 0)

1_2 *(second predicate)*
(1, 2, 0, 0)

the evaluation outputs

1_1
(1, 2, 1, 0)

1_2
(0, 0, 0, 0)[14]

where the first predicate is recognised despite one mistaken argument, whereas the second predicate is not recognised at all.

The output format had to follow the one provided for the training data. For instance, asterisks indicating the implicitness of the arguments had to be included in the output file. As a matter of fact, retrieving the implicit function of a reconstructed argument serves to catch the degree of understanding of the system, along with making use of the processing of this information for the improvement of fine-grained action detection tasks. On the other hand, the choice between alternative arguments (separated by a slash in the reference files) do not invalidate the results. In fact, to execute an action, only one of the uttered alternatives must be chosen. Therefore, when one of the alternatives was recognised, the resulting output did not contain recognition errors. On the contrary, when the system reports both alternatives in the output file, the Levenshtein distance increased. In the reference files, alternatives were also occurring as implicit arguments, when an utterance can be completed by more than one possible argument.

[12] 1; [take(500 g, milk), add(milk, pot)]

[13] 1; take(500 g, cream)

[14] The first action was recognised; two arguments were expected but one of them was wrong. The second action was not recognised at all.

4 Participating Systems

In this section, we will report the results collected from testing the two participants' systems: the first (Section 4.1) have been developed at Fondazione Bruno Kessler (FBK), while the second by an Italian company which has decided to remain anonymous (Section 4.2). In table 2, results are summarised, showing that FBK had better performances in terms of correct predicate and arguments recognition for the intent classification, as far as the second system is concerned (Figure 2). On the other hand, the first one outputted worse results, despite the introduction of the argument repetition request. In this phase, the argument repetition percentage was not weighted in the accuracy rate of the system, which would have resulted in a slight increase of the accuracy itself, but we reported it as an additional performance of the participating system. For the anonymous system the action recognition is slightly beyond the 50%, but the argument recognition shows some issues (Figure 2) concerned with an over-fitting problem (see Section 4.2). For all three systems, recognition errors seemed to be random and not justifiable as semantically-related word selections.

4.1 FBK-HLT-NLP

To solve the proposed task, two different approaches were introduced. The first system was similar to the architecture proposed in (Madotto et al., 2018) and was based on an encoder-decoder approach. The encoder consisted of a MemNN network (Sukhbaatar et al., 2015) that stored each previous sentences in memory, from which relevant information was retrieved for the current sentence. The decoder was a combination of i) a MemNN to decode the input to an instruction containing tokens from output vocabulary and ii) a Pointer network (Vinyals et al., 2015) that chose which token from the input was to be copied to

	Correct Actions	Correct Arguments	Incorrect Actions	Incorrect Arguments	Argument Repetition
FBK System 1 [a]	50,16	28,31	49,83	71,68	**4,11**
FBK System 2	**66,36**	**46,22**	33,64	53,78	0
Anonymous System	53,89	17,46	46,11	82,54	0

[a] One user is missing.

Table 2: Percentages of accuracy and error rate for each tested system

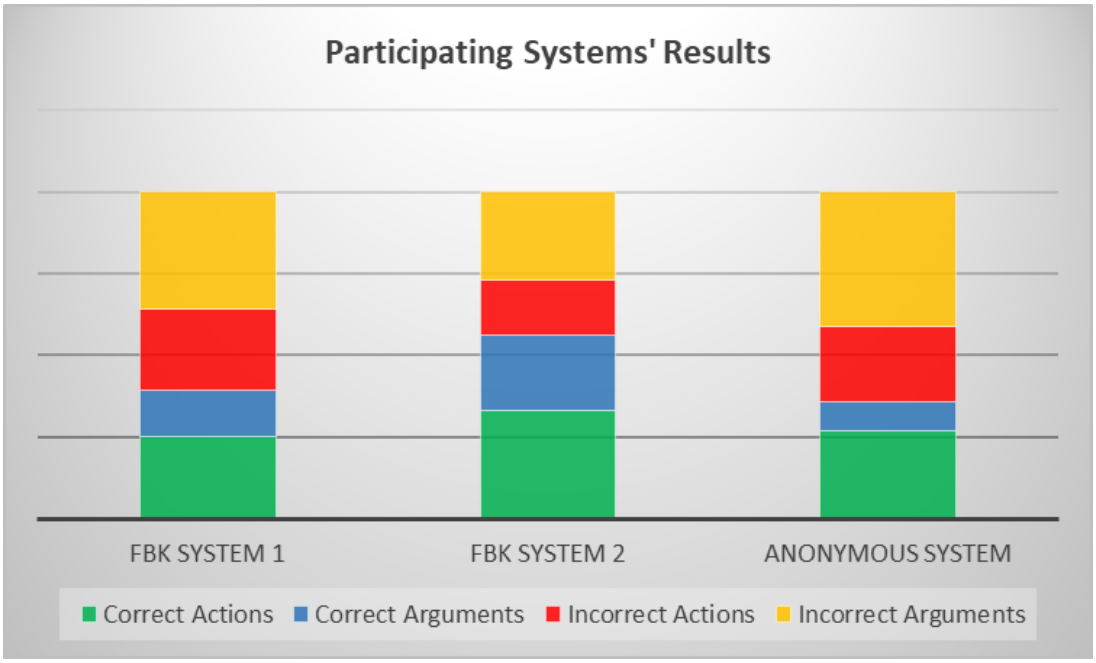

Figure 2: Results of the FBK first system

the output instruction. This system was used to classify the SUGAR corpus intents after an ASR transcription (System 1).

The second approach consisted of modeling the task as a *sequence to sequence* problem. Rather than implementing a new system, *Fairseq* (Gehring et al., 2017) - a fully convolutional architecture for sequence to sequence modeling - was used. Instead of relying on Recurrent Neural Networks (RNNs) to compute intermediate encoder states z and decoder states h convolutional neural networks (CNN) were adopted. Since the amount of training data was not big enough to train the model with such a system, written synthetic data were generated. To generate new data two main methodologies were adopted: on one hand random words were substituted with similar words based on similarity mechanisms, such as word embeddings; on the other hand, training sentences were generated by replacing verbs and names with synonyms extracted from an online vocabulary (System 2).

4.2 Deep neural network for SUGAR

The anonymous participant built a deep neural network system to tackle this task[15]. First of all, to convert the spoken utterances into text the Google Speech API was used. The neural network used a word embeddings lexicon trained on a corpus of recipes crawled on the web (4.5 million words) as features. The word embeddings, with vectors having 100 dimensions, were trained with the skip-gram algorithm of fastText[16] (Bojanowski et al., 2016).

As a preliminary step an autoencoder to embed the predicates in a vector was built. The encoder was made of a two Bi-LSTM layers. The first one was in charge of processing the token sequences for each predicate. The second layer processed the sequence of predicates and embeds them into a vector called predicates embedding. This vector was then split into n-parts where n was the

[15] The following report is a result of a conversation with the involved participant, whose report was not officially submitted to EVALITA 2018 in order to remain anonymous.

[16] https://fasttext.cc/

maximum number of predicates. The decoder was made of two Bi-LSTM layers, where the first layer was in charge of decoding the sequence of predicates and the second layer was in charge of decoding the sequence of token for each predicate. To test the autoencoder, a development test set was extracted from the training test. The autoencoder was able to encode and decode with no changes the 96.73% of the predicates in the development test set.

The different possible actions have been represented as classes in a hot-encode vector, and for each action a binary flag has been used to represent whether the action was implicit or not. The predicates have been encoded into a vector, using the aforementioned encoder, and for each predicate a flag was used to represent their alleged implicitness.

A multitask neural network was used to classify the actions, to detect whether they were implicit and to predict the predicates. The network took in input a recipe as a list of commands, each of whom was encoded by a Bi-LSTM layer. A second Bi-LSTM layer processed the command sequence and outputted a list of command embeddings. Each embeddings was split into n-parts which identified the actions included in the command. Each of these actions was passed to 4 dense layers that predicted the action class, the implicitness of the action, and the predicates embedding. Finally, the above-described decoder translated the predicates embedding into actual predicates.

5 Conclusions

With this task we proposed a field of application for spoken language understanding research concerned with intents classification of a domain-dependent system using a limited amount of training data. The results show that further analysis should be carried out to solve such semantic recognition problems, starting with an analysis of the errors occurred in the participating systems, an enlargement of the reference corpus, up to finding a suitable pipeline for data processing, including a rule-based module to model issues such as the argument implicitness, both in anaphoric- or semantic-dependent situations. This task is therefore intended to be a first reflection, whose next developments would include the creation of a corpus for the English language and the introduction of multimodality. As a matter of fact, pointing gestures or mimed actions and movements, on the basis of which the interlocutor should be capable of re-performing them with actual tools and ingredients, are multimodal activities that are of interest for this field of application as for any other spoken understanding task where a shared context of interaction is expected.

Acknowledgments

We thank the EVALITA 2018 organisers and the SUGAR participants for the interest expressed. A special thank also goes to Claudia Tortora, who helped us collect recipes and annotate our training set, and, last but not least, to the numerous testers who had fun talking with our dear Bastian.

This work is funded by the Italian PRIN project *Cultural Heritage Resources Orienting Multimodal Experience* (CHROME) #B52F15000450001.

References

James Allen, Mehdi Manshadi, Myroslava Dzikovska, and Mary Swift. 2007. Deep linguistic processing for spoken dialogue systems. In *Proceedings of the Workshop on Deep Linguistic Processing*, pages 49–56. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Tommaso Caselli, Nicole Novielli, Viviana Patti, and Paolo Rosso. 2018. Evalita 2018: Overview of the 6th evaluation campaign of natural language processing and speech tools for italian. In Tommaso Caselli, Nicole Novielli, Viviana Patti, and Paolo Rosso, editors, *Proceedings of Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018)*, Turin, Italy. CEUR.org.

Justine Cassell, Joseph Sullivan, Elizabeth Churchill, and Scott Prevost. 2000. *Embodied conversational agents*. MIT press.

Justine Cauell, Tim Bickmore, Lee Campbell, and Hannes Vilhjálmsson. 2000. Designing embodied conversational agents. *Embodied conversational agents*, pages 29–63.

Sarah J Darby. 2018. Smart technology in the home: time for more clarity. *Building Research & Information*, 46(1):140–147.

Myroslava O Dzikovska, James F Allen, and Mary D Swift. 2003. Integrating linguistic and domain knowledge for spoken dialogue systems in multiple

domains. In *Proc. of IJCAI-03 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. *arXiv preprint arXiv:1804.08217*.

Scott Miller, David Stallard, Robert Bobrow, and Richard Schwartz. 1996. A fully statistical approach to natural language interfaces. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 55–61. Association for Computational Linguistics.

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.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Gokhan Tur and Li Deng. 2011. Intent determination and spoken utterance classification. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, pages 93–118.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.

Ye-Yi Wang. 2010. Strategies for statistical spoken language understanding with small amount of data-an empirical study. In *Eleventh Annual Conference of the International Speech Communication Association*.

Anna Wierzbicka. 1972. Semantic primitives.

Martina Ziefle and André Calero Valdez. 2017. Domestic robots for homecare: A technology acceptance perspective. In *International Conference on Human Aspects of IT for the Aged Population*, pages 57–74. Springer.