SUGAR - Spoken Utterances Guiding chef's Assistant Robots

Francesco Cutugno, Maria Di Maro, Antonio Origlia (Università degli Studi di Napoli 'Federico II')
Claudia Tortora (Università degli Studi di Napoli 'L'Orientale')

Brief Task Description

The task is intended to train a voice-controlled robotic agent to act as a cooking assistant.

Motivation and Background

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. The possibility to use such Artificial Intelligence technologies in domestic environments is increasingly becoming a reality [1, 2]. 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 ASR errors, reproducing the behavioural stimulus-response model, is one of the first steps before starting structuring a real computational knowledge system [3]. In this context, the application of various semantic annotation schemata and criteria of knowledge modelling are of particular interest. For this reason, in this task, we are going to propose a possible framework to be tested, recurring to state-of-the-art SLU systems participating to the EVALITA-SUGAR challenge.

Task description

In the SUGAR challenge, the underlying task is to train a voice-controlled robotic agent to act as a cooking assistant. For this purpose, a training corpus of spoken commands is collected and annotated. To collect the corpus, we designed a 3D virtual environment reconstructing and simulating a real kitchen where users can interact with a robot which receives commands to be performed in order to accomplish some recipes. User's orders are inspired by silent cooking videos shown in the 3D scene, thus ensuring the naturalness of the spoken production. Videos are segmented into elementary portions (frames) and sequentially proposed to the speakers who will utter a single sentence after each seen frame. In this view, speakers watch at video portions and then give instructions to the robot to emulate what seen in the frame. The collected corpus then consists of a set of commands, whose meaning derives from the various combination of actions, items (i.e. ingredients), tools and different modifiers. Audio files will be 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 of which speakers' voice is segmented into sentences representing isolated commands. Orthographic transcriptions of the audio files will not be provided. Consequently, participants can use whichever ASR they prefer, whose performance will not be under assessment. Nevertheless, the developed systems are expected to be strongly efficient despite the possible ASR deficiencies. Each audio file will be paired to a further one containing the corresponding action annotation.

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 results in determining the optimal action predicate corresponding to each command. The training set consists of audio file 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 file will be always mapped on a single interpretative predicate. The training set consists of 1721 utterances produced by 36 different speakers annotated by experts. The action templates which have been defined through the data collection are

prendere(quantità, [ingredienti]/recipiente)

aprire(quantità, [ingredienti], recipiente)

mettere(utensile/[ingredienti], elettrodomestico, modalità, quantità)

sbucciare(quantità, [ingredienti], utensile)

schiacciare([ingredienti, utensile)

passare([ingredienti], utensile)

grattare([ingredienti], utensile)

girare([ingredienti], utensile)

togliere(utensile/prodotto, elettrodomestico)

aggiungere(quantità, [ingr.], utensile/recipiente/elettrodomestico/[ingr.], modalità)

mescolare([ingredienti], utensile, modalità)

impastare([ingredienti])

separare(parte/[ingredienti], ingrediente/utensile)

coprire(recipiente/[ingredienti], strumento)

scoprire(recipiente/[ingredienti])

controllare(temperatura, ingrediente)

cuocere([ingredienti], utensile, modalità, quantità)

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 (previous instantiated arguments, which are not uttered, not even by means of clitics or other pronouns) or from the semantics of the verb (i.e. *friggere (fiori)* is represented as *aggiungere(fiori, *olio*)* because *oil* is implicitly expressed in the semantics of the verb *to fry*). 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)* which is annotated with the generic template *mettere(pellicola, ciotola).* In other cases, the uttered action represents the consequence of the action reported in the template, as in *separare(parte, fiori)* and *pulire(fiori)*, or *mescolare([lievito, acqua])* and *sciogliere(lievito, acqua).* The argument order does not reflect the one in the audio files, but the following:

action(quantity[1], object, complement, modality)

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

The test set consists of about 572 audio files containing uttered commands without annotations. Task participants should provide, for each target command, the correct action predicate following the above-described format. Although single actions are the same 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 will be evaluated on the basis of correctly interpreted commands, represented in the form of predicates.

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

[1] The quantity always precedes the noun it is referred to. Therefore, it can also come before the complement

Evaluation protocol

The evaluation protocol will cover 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 will assign a weaker penalisation than the one considered for missing the parameters or the action. The collected corpus will not, however, contain situations in which the system asks for repetitions.

The designed evaluation procedure outputs 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 are not recognised, even the argument number is set on 0;
3. the number of expected arguments as indicated in the reference annotation file;
4. the distance between the output file and the reference file computed by means of the Levenshtein distance [4]; the higher the computed distance in the output is, the more mistakes the system has detected;
5. 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)]

and the recognition procedure outputs

1; prendere(500 g, panna)

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 will output

1_1

(1, 2, 1, 0)

1_2

(0, 0, 0, 0)[2]

The output format must follow the one provided for the training data. For instance, asterisks indicating the implicitness of the arguments must 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 is recognised, the resulting output will not contain recognition errors. On the contrary, when the system reports both alternatives in the output file, the Levenshtein distance will increase. In the reference files, alternatives can also occur as implicit arguments, when an utterance can be completed by more than one possible argument.

References

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

[2] Martina Ziefle, André Calero Valdez. Domestic Robots for Homecare: A Technology Acceptance Perspective, International Conference on Human Aspects of IT for the Aged Population, Springer, 57-74 (2017).

[3] Burrhus F. Skinner. The generic nature of the concepts of stimulus and response, The Journal of General Psychology, 12, 1, 40-65, Taylor & Francis (1935).

[4] Levenshtein, Vladimir I. Binary codes capable of correcting deletions, insertions, and reversals, In Soviet physics doklady, Vol. 10, No. 8, pp. 707-710 (1966).

Contact

Maria Di Maro: mdimaro17@gmail.com

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