Grounding end-to-end Architectures for Semantic Role Labeling in Human Robot Interaction

Claudiu Daniel **Hromei**[1,2], Danilo **Croce**[1] and Roberto **Basili**[1]

[1]*University of Roma Tor Vergata, Rome, Italy*

[2]*Università Campus Bio-Medico di Roma, Rome, Italy*

Abstract

Natural language interactions between humans and robots are intended to be situated in the sense that both user and robot can access and refer to the shared environment. Contextual knowledge plays a key role in resolving the ambiguities inherent in interpretation tasks. In addition, we expect the interpretation produced to be well-founded, e.g., that all mentions of entities in the environment (as perceived by the robot) are correctly grounded. In this paper, we propose the application of a transformer-based architecture that combines the input utterance with a linguistic description of the environment to produce interpretations and references to the environment in an end-to-end fashion. Experimental results demonstrate the robustness of the proposed methodology, overcoming previous approaches in which linguistic interpretation and grounding are composed of possible complex processing chains.

Keywords
Grounded Semantic Role Labeling, Human Robot Interaction, End to End Sequence to Sequence Architectures, Robotics and Perception

1. Introduction

In a world that is moving towards a widespread use of virtual assistants (at home, for work, or as a hobby) and robotic platforms that perform difficult or risky tasks for humans, making sure these technologies understand human language becomes increasingly important. Virtual assistants are designed to satisfy a user need that is often informational or merely entertaining, like in the daily requests made to search for translation of individual words or the requests for popular songs. Understanding a command or the title of a song turns out to be crucial to satisfying these needs in a natural manner. The quality of such interpretation processes is important, especially in critical scenarios, e.g. involving robotic platforms that perform sensitive, medical or critical tasks usually carried out under speech-based control.

The use of natural language to control these platforms or teach them the movements or actions needed to perform a task could be a key factor in the not-too-distant future. Today, there are countless domestic robots whose job is to perform domotic tasks, such as automatic cleaning or even cooking. As suggested in [2], domestic robots have to contend with complex problems such as (1) self-localization and navigation in complex environments, (2) precise recognition of objects and people, (3) manipulation of physical objects, and (4) meaningful interaction with humans to satisfy their needs, being them physical or abstract (Human Robot Interaction).

It is necessary to make these home automation assistants aware of their surroundings and the elements in them. In order to correctly interpret a sentence such as

$$\text{``Take the volume on the table near the window''} \tag{1}$$

it is necessary to support capability for entity association able to retrieve objects mentioned in the command (such as *volume*, *table* and *window*) and possibly disambiguate between entities of the same type. For example, in the case where several *tables* exist in the environment, it is necessary to explicitly guide the interpretation towards the one that is next to the *window*.

Several works, such as [3] proposed specific methods for a Grounded language interpretation of robotic commands. We investigate here the approach recently proposed in [4], namely *Grounded language Understanding via Transformers* (GrUT): this method suggests adopting a Transformer-based architecture (e.g., BART presented in [5]) that can produce the linguistic interpretation of an utterance by taking in input *i*) the transcription of the input command, *ii*) a linguistic description of the entities from the map involved in the command and *iii*) a linguistic description of the robot's capabilities.

GrUT is appealing as it drastically reduces the need for task-specific engineering of the model (such as [3]): it only requires a way to linguistically describe a map, so that the Transformer generates interpretations consistent both with the utterance *and* the map. In other words, the same command can generate different linguistic interpretations when coupled with different map descriptions. Let us consider the utterance in example 1. Whenever the *volume* is actually next to the *table*, the input utterance is extended with the additional synthetic text "v_1 *is a volume, t_1 is a table and v_1 is near t_1*" so that GrUT produces the following interpretation: TAKING(THEME("*the volume on the table near the window*")). On the contrary, GrUT generates "BRINGING(THEME("*the volume*"),GOAL("*on the table near the window*")) whenever these objects are far from each other, that can be expressed by "v_1 *is a volume, t_1 is a table and v_1 is far from t_1*". The final interpretation is consistent with Frame Semantics [6]: in the first case, the robot is expected to move towards the table (i.e., $t1$) and take the volume (i.e., v_1), while in the second case it is expected to take the volume that is far away from the table and bring it over there. However, while the actual state of the environment affects the interpretation process, GrUT only produces just an approximate linguistic effect, expressed by texts fragments as fillers of the output argument predicate structure. A further step is still required to *ground* each fragment to the intended entity triggered by the predicate.

In this paper, we propose an end-to-end grounded interpretation process[1], so that GrUT is expected to produce indexed representations in the robot KB, such as TAKING(THEME(v_1)) or BRINGING(THEME(v_1), GOAL(t_1)). It is worth noting that the connection between words in a spoken command and the entities (here referred to as *linguistic grounding*) is not a trivial task. In general, as in [3], it is assumed that each entity in a map is denoted by one or more labels to enable such a grounding: for example, one or more linguistic references, such as *volume* or *book*, correspond to the object v_1. These are used in [3] and [4] to retrieve all entities involved by a command from the map. The simplest approach here is to select all and only those entities whose denotation coincides with one of the command words, as applied in [4]. Unfortunately, this assumption is quite unrealistic, as for the role of synonyms or paraphrases used to refer to

---

[1]We released an extended version of GrUT at https://github.com/crux82/grut

objects in user commands: "*take the handbook . . .*" or "*take the tome . . .*" can be equivalently used in natural language interaction. To overcome this limitation, a more complex retrieval function is explored in this paper. It makes use of more expressive associations between several linguistic labels including words that are highly similar according to a neural semantic similarity function. To the best of our knowledge, this expands recent research like [7] and [8] as it is the first end-to-end technique for Fully Grounded Linguistic Interpretation, made dependent on an explicit (logical) description of the environment.

Results indicate that the adoption of expressive functions for linguistic grounding enables an end-to-end process that is even more robust than an *a-posteriori* application of the linguistic grounding process to the interpretations produced by GrUT.

In the rest of this paper, section 2 summarizes the related work, section 3 presents the proposed extension of GrUT, section 4 reports the experimental evaluation, while section 5 derives some conclusions.

2. Related Work

The semantic interpretation of texts or spoken utterances is generally modeled as a Semantic Role Labeling (SRL) task, which consists of identifying all the expressed linguistic predicates (such as BRINGING vs. TAKING evoked by the verb *"to take"*) and their corresponding semantic arguments (such as *"the volume"* or *"on the table near the window"*) in order to perform a deep semantic interpretation of a human-generated utterance [9]. Data-driven approaches for SRL have gotten a lot of attention since the pioneering work of Gildea and Jurafsky [10], leading to multiple benchmarking initiatives [11, 12, 13]. As in [14, 15, 16], the majority of methods divide the processing tasks into at least two steps: first, the target predicates are identified and clarified; second, for each predicate, the relevant arguments are located and organized according to their roles in the corresponding predicate. The latter often focuses solely on semantic role labeling while ignoring previous predicate identification and disambiguation techniques. This decomposition generally holds even when transformer-based models have been increasingly used in SRL, since the seminal works of [17]: in [18, 19] or [20] a pre-trained architecture, such as BERT [21], RoBERTa [22], BART [5] or T5 [23] are succesfully applied, but always according to the above task decomposition.

The authors of [19] demonstrate how BERT can be applied to semantic role categorization without relying on syntactic features and yet produce cutting-edge results. Instead, [18] uses a graph neural network stacked on BERT encodings to demonstrate the value of incorporating dependency structures within the transformers: first, the output of the transformers is fed into the graph neural network, and then semantic structures are imposed within the attention layers. Both methods produce outcomes that are on par with the current state-of-the-art. The importance of predicate disambiguation for the overall process is demonstrated in [20] by modeling the two tasks of argument labeling and predicate disambiguation using RoBERTa and the PropBank corpus [24], showing how predicate disambiguation is helpful for the overall process. The initial proposals for an end-to-end architecture, which accepts plain text as input and uses T5 and BART to simultaneously identify predicates and arguments, are found in [7] and [8], respectively. In essence, T5 and BART take a simple sentence as input and create

an artificial text that allows all predicates and roles to be derived. In the same way that the argument is recognized by specifying its position within the phrase, BART is used to identify the predicate by signaling to the GSRL model the token that inspires it. The model was evaluated on CoNLL2012 [13].

All of the aforementioned methods, however, simply consider linguistic evidence. According to the concept of Grounded Semantic Role Labeling (G-SRL) in [25], the proper interpretation of an utterance in a given context depends on the language's grounding concerning the environment itself, such as the real objects the speaker refers to. The interpretation is to be dependent on data derived from the analysis of photographs depicting the environment, and a probabilistic model is proposed in the same paper. This concept is further emphasized in [3], which explains how a domestic robot's perception of commands depends on proofs of properties the robot can carry out over a logical map of the environment. This latter is used to describe the surrounding area, the objects located in specific positions and other relevant relationships.

It's interesting to note that texts are annotated using the Frame Semantics theory [6], which [3] suggests can be exploited by the robot to directly derive the necessary plan and action primitives. However, [3] adopts the traditional SRL processing chain. Additionally, their output still only exists at the linguistic level: labeling only defines roles associated with words, not with actual objects, whereas interpretation depends on correlations between words and objects in the environment. As suggested in [3, 25], contemporary approaches to the interpretation of robotic spoken commands must be harmonized across several semantic dimensions, at the very least: (1) spatial awareness, or knowledge about the physical environment in which the robot acts; (2) self-awareness, or knowledge of its proper capabilities and limitations as a robotic platform; and (3) linguistic competence needed to understand user's utterance and produce meaningful statements in response to stimuli or needs.

Recently, in [4] _Grounded language Understanding via Transformers_ (GrUT) is proposed as a sequence-to-sequence (seq2seq) approach for GSRL, sensitive to the map information in form of linguistic descriptions and capable of directly perform Grounding during interpretation, effectively linking entities in the map with the Arguments predicted. In this paper, we extend GrUT to make it end-to-end, thus allowing richer interpretations that are also grounded in the environment. Moreover, different and more expressive policies to retrieve entities from the map are investigated.

3. End-to-end Grounded Semantic Role Labeling

As discussed in the previous section, the semantic interpretation of spoken commands (and in general texts) strongly benefits from the application of Transformer-based architectures such as BART [5]. In a nutshell, a Transformer is applied to interpretation processes by taking in input a text expressed in natural language, and "translating" it into an artificial text reflecting the underlying linguistic predicate. In order to extend the application of Transformer to Grounded SRL tasks, the idea behind GrUT [4] is to use a natural language description of the map and add it to the input sentence. If we want to make the interpretation sensitive to the entities, their proprieties, position and relational information (such as proximity or distance), we first need a way to refer to them. In our context, one entity is known through its (English) noun (possibly

its most commonly used lexical reference, e.g. the word *volume*) as well as its conceptual type. The association with the environment (i.e. the grounding) is realized through its identifier (Existence Constraint, EC) that is linked to the position of the corresponding physical object in the environment. For example, the map to be paired with the command in (1) can give rise to the following description:

EC: "b_1, also known as volume or book, is an instance of the class BOOK, t_1, also known as table, is an instance of the class TABLE and w_1, also known as window, is an instance of the class WINDOW.".

Moreover, if book b_1 and the table t_1 are close to each other in the environment, a further declaration of a Proximity Constraint (PC) acting over them will be added:

PC: "b_1 is near t_1 and t_1 is near w_1"

Finally, for each selected entity, a description of whether the property containing other objects is true (Containability Constraint, CC) will be added. For illustrative purposes, imagine the existence of a hypothetical cup c_1:

CC: "c_1 can contain other objects"

The entire description is a micro-story[2] useful for the SRL model to disambiguate between the different situations. Notice that only when the spatial constraint PC is true, the correct interpretation for the ambiguous situation (1) finally corresponds to the role labeled logical form:

$$\text{TAKING}(\text{THEME}("\textit{the volume on the table near the window}")). \tag{2}$$

Since the book b_1, referenced through the noun *volume*, is close to t_1, it is interpreted thus as the THEME of the TAKING predicate. In this work, only these 3 constraints are defined and used, but in the future a wider list of properties will be explored. It is worth noticing that the linguistic description of the map enables the use of highly accurate transformers (such as BART [5] and T5 [23]). These are pre-trained on large natural language corpora and may take advantage of linguistic features, relationships and cross-dependencies to properly carry out SRL on the overall textual examples made by the informative pairs in GRUT. The extraction algorithm acting over a map, given a command c is reported in Algorithm 1.

In this paper, we extend the above method in two directions. First, we make GRUT an end-to-end architecture. To this end, the transformer is trained to generate a predicate that expresses both the semantic information and the object involved. In other words, while the input is kept consistent with that used by GRUT, the output is:

$$\text{TAKING}\big(\text{THEME}(b_1)\big) \tag{3}$$

and it is expected that the transformer autonomously learns this transformation. This schema produces a different input in the case that book b_1 is far from table t_1. Part of the map description will be *PC: "b_1 is far from t_1 and t_1 is near w_1"*, while the output is expected to be significantly different, i.e.,

$$\text{BRINGING}\big(\text{THEME}(b_1), \text{GOAL}(t_1)\big) \tag{4}$$

---

[2]All the constraints are appended to the input, each of which is divided by a "#" delimiter character.

Algorithm 1 GRUT compilation Algorithm

1: **procedure** CONSTRUCT_INPUT(*Sentence* $s = (w_1, ..., w_{|s|})$, *LexSim. ls*, *Threshold* τ_{ls})
2:      $Entities \leftarrow \emptyset$
3:      **for** $i = 1, ..., |s|$ **do** ▷ All entities that can be potentially referred to by a word in the command are collected to be considered in the map description. The implementation of GET_CANDIDATE_ENTITIES is in Algorithm 2
4:          **if** $PosTag(w_i) ==$ NOUN **then**
5:              $Entities \leftarrow Entities \cup$ GET_CANDIDATE_ENTITIES(w_i, ls, τ_{ls})
6:      $ec \leftarrow ""$ ▷ Existence Constraints
7:      **for** $e \in Entities$ **do**
8:          $ec \leftarrow ec +$ " # "$+ get\_ref(e) +$" also known as "$+ get\_lexical\_ref(e) +$
9:                                   " is an instance of class "$+ get\_class(e)$
10:     $cc \leftarrow ""$ ▷ Containability Constraints
11:     **for** $e \in Entities$ **do**
12:         **if** $containability(e)$ **then**
13:            $cc \leftarrow cc +$ " # "$+ get\_ref(e) +$" can contain other objects"
14:     $pc \leftarrow ""$ ▷ Proximity Constraints
15:     **for** $e_1 \in Entities$ **do**
16:         **for** $e_2 \in Entities$ **do**
17:            **if** $e_1 \neq e_2 \wedge distance(e_1, e_2) < \tau$ **then**
18:               $pc \leftarrow pc +$ " # "$+ get\_ref(e_1) +$" is near "$+ get\_ref(e_2)$
19:         $Entities \leftarrow Entities - \{e_1\}$
20:     **return** $s + ec + pc + cc$

In addition, this paper extends the grounding process to improve the robustness and applicability of GRUT. In fact, GRUT assumes that each entity e in the environment is enriched with a set of lexical references $LR(e) = \{w_1^e, \ldots, w_l^e\}$, used to link words $(w_1, \ldots, w_{|s|})$ in the sentence s. As an example let us consider the volume v_1 and its corresponding lexical reference $LR(v_1) = \{$*volume, book*$\}$. A robust linguistic grounding function is essential for GRUT (and our extended counterpart) to build the map description. In this sense, the algorithm 2 expresses the policy adopted to retrieve the entities involved in the utterance. In a nutshell, we propose to retrieve all entities that have significant lexical similarity with each noun in s. We will experimentally evaluate three *LexicalSimilarity* functions, characterized by incremental levels of expressiveness:

1. **Exact Match**: the simplest function corresponds to the naive exact match between two input strings. It produces a Boolean result, i.e., 1 if the two strings are equal, and 0 otherwise. It allows retrieving all entities that have at least one lexical reference perfectly matching one word in the command; it is the function used in GRUT and, while it is very precise, it fails when the user refers to entities using synonyms, e.g., referring to v_1 with *handbook, tome* or *manual*. Despite its simplicity, it assumes significant effort in map construction: in many practical scenarios, we cannot assume that all possible lexical references are defined for all entities.

2. **Levenshtein similarity**: a "soft" string matching that sometimes captures also morphological relatedness between input word pairs. We defined a Levenshtein Similarity

---

Algorithm 2 Entity Retrieval Algorithm

1: **procedure** GET_CANDIDATE_ENTITIES(*WORD w*, *LEXICALSIMILARITY ls*, *THRESHOLD* τ_{ls})
2:   $Candidate\_Entities \leftarrow \emptyset$
3:   **for** $e \in KB\_Entities$ **do**
4:    **for** $lex\_ref \in LR(e)$ **do** $\triangleright$ For each lexical reference
5:     **if** $ls(w, lex\_ref) > \tau_{ls}$ **then**
6:      $Candidate\_Entities \leftarrow Candidate\_Entities \cup e$
7:   **return** $Candidate\_Entities$

($LevSim$) in the form

$$LevSim(w_i, w_j) = 1 - \frac{LevDist(w_{i,j})}{|longest(w_i, w_j)|} \tag{5}$$

which is based on the well-known Levenshtein distance ($LevDist$) between input strings w_i and w_j. While Levenshtein distance produces values ranging from 0 to the length of the longest word (between the two), $LevSim$ again ranges between 0 (totally different strings) and 1 (the same string). This similarity function is more robust in linking slightly different input strings (e.g., *book* vs *handbook*) and it may capture some sort of morphological analogy between words, but it fails when the user refers to entities using synonyms.

3. **Neural Semantic similarity**: it corresponds to the cosine similarity[3] between embeddings representing both words [26], providing a more robust connection between words involved in paradigmatic relationships, such as quasi-synonymy. In this work, we adopted a neural representation based on Word Embeddings, using the well-known Word2vec formulation [27] derived from the analysis of the English version of Wikipedia.

To prevent smoothed measures (such as the one based on word embeddings) from causing an excessive number of entities to be retrieved (e.g., from a map containing dozens of objects), we applied a threshold τ: as a result, only entities with significant similarity to one of the command words are retrieved. Our method should be robust even in cases where multiple entities are retrieved for the same word. Although the input text may contain redundant entities in the map description, the arguments in the output should contain only relevant entities. As a result, the transformer is expected to select only valid entities (using the attention mechanism of the encoder) to produce the correct output.

4. Experimental Evaluation

The goal of the following evaluation is to demonstrate the effectiveness of the proposed extension of GrUT and the different impacts of the lexical similarity functions we considered.

---

[3]The cosine similarity function ranges between [-1, 1]. To maintain a consistency with the other policies, we replaced it with the $max\big(0, cosim(w_i, w_j)\big)$.

4.1. Experimental Setup

The proposed approach is evaluated here in a home automation scenario, where a robot is supposed to receive spoken commands and is required to interpret commands in order to perform the expected actions. Examples are picking up a book on a table, taking out the rubbish, or looking for the keys. The evaluation is applied to the HuRIC[4] dataset, which consists of 656 voice commands in English coupled with interpretations, in terms of predicates and arguments, to which the Grounding processes of the previous chapter were applied, i.e. linking them with identifiers of entities in the surrounding environment. In HuRIC, predicates are defined according to a subset of the semantic frames of FrameNet [6] and the corresponding arguments are selected. On average, each entity is represented by 1.37 lexical references. Inspired by the previous evaluation in [4] we adopted a 10-fold cross-validation scheme with 80/10/10 data split between training/validation/test and the following aspects of the extended version of GRUT are evaluated:

- *Frame Prediction (FP)*, i.e. the ability of the models to correctly generate the names of frames evoked by the voice command; it is measured as the F1-measure, where Precision and Recall reflect the capability of GRUT to recover the correct frame(s) expressed in the spoken command.
- *Argument Identification and Classification (AIC)* as an *Exact Match (AIC-ExM)* evaluation in which the ability of the systems to correctly generate the names of the Arguments evoked by the command and to associate them all with the entities that evoke that Argument is evaluated; AIC-ExM is measured as the F1-Measure of produced arguments that perfectly corresponds to the gold-standard in their complete form, including frame, type of argument and corresponding grounded entities.
- *Argument Identification and Classification (AIC)* as a more relaxed evaluation, in which the models are required to be able to associate with each Argument at least the correct *Entity Head (AIC-Enty)*; it is measured as the F1-measure, where Precision and Recall reflect the capability of GRUT to recover the correct arguments, i.e., having the same argument type and grounded entity of the ones in the spoken command.

We compared the impact of the three proposed Lexical Similarity functions involved in the entity retrieval step. For each function, a specific threshold τ was estimated on the validation set by maximizing the F1-measure of the entity retrieval step. In this subtask, Precision is measured as the average percentage of entities that are correctly retrieved when constructing the map description, while Recall is measured as the average percentage of entities that were expected to be retrieved as mentioned in the command. In particular, under the Exact Match policy a definition of $\tau_{\mathrm{EM}} = 0.50$ was applied, while under the policy based on Levenshtein Similarity and the Word Embedding one, $\tau_{\mathrm{Lev}} = 0.80$ and $\tau_{\mathrm{WE}} = 0.55$ are respectively used. In Table 1, the values of the main parameters used to train the Transformer are reported as the configuration maximizing the AIC-ExM on the development set, on average.

---

[4]https://github.com/crux82/huric

Table 1
Summarization of parameters of the BART based Transformer

Param Name	Value
Optimizer	AdamW
Learning Rate	5e-5
Early_stopping_delta	1e-4
Early_stopping_metric	eval_loss
Batch_size	16
Gradient_accumulation_steps	2
Early_stopping_patience	3
Scheduler	linear_schedule_with_warmup
Warmup Ratio	0.1
Max_length	256
Epochs	50(max)

4.2. Results and Discussion

The experimental results are reported in Table 2. In the first rows, GRUT + *ExPostGrounding* represents our strong baseline and it is based on GRUT which, in [4] was demonstrated competitive with state-of-the-art models for Grounded Semantic Role Labeling, such as the one proposed in [3]. GRUT generates logic forms expressing the interpretation of commands at a linguistic level and in [4] it was reported to achieve 92.28% of F1 in the FP task, 88.41% in the Argument Identification and Classification Exact match (AIC-ExM) and 93.29% as the score in recovering the Semantic Head of the individual arguments.

To implement our baseline, we re-used the predictions of GRUT and applied the linguistic grounding *a-posteriori*: for each argument in the produced logic form, such as GOAL("*on the table near the window*"), the first noun is selected (here the *table*) and the entity maximizing the lexical similarity is selected to replace the argument. If no entity is retrieved (or the threshold is not exceeded) the related argument is removed. We considered the three Lexical Similarity functions, i.e. based on Exact Match (EM), Levenshtein Similarity (LS) and semantic similarity estimated over Word Embeddings (WE).

Table 2
Comparative Evaluation on the Frame Prediction FP, Argument Identification and Classification AIC tasks of the different G-SRL models: Exact Match (*ExM*) and Head Match (*AIC-Enty*) are the different metrics for AIC.

Model	Retrieval Policy	FP	AIC-ExM	AIC-Enty
GRUT + *ExPostGrounding*	Exact Match		78.62%	80.00%
	Levenshtein Sim.	92.18%	79.80%	81.37%
	Word Embeddings		80.91%	82.22%
GRUT *End-to-End*	Exact Match	90.38%	83.16%	84.79%
	Levenshtein Sim.	**92.40%**	84.24%	85.66%
	Word Embeddings	91.90%	**90.03%**	**91.46%**

The application of the grounding function causes a significant performance drop: as an

example, AIC-ExM drops from 88.41% to 80.91%. This is mainly due to the 10% of entities in HuRIC whose lexical reference does not match any of the words used in a candidate argument.

The values raise from 78.62% and 80.00% for *AIC* tasks at argument level (indeed simpler tasks), when using the String Matching Retrieval method, to 80.91% and 82.22% for the W2V Retrieval method, showing that the word embeddings and the vector similarity function are useful for recovering some connections. The small difference between a grounding function based on the Exact Match and the one based on WEs suggests that lexical references in HuRIC are generally the same words used in the commands. The different entity retrieval policies do not affect the Frame Prediction subtask.

When applying our proposed model, namely GrUT-*End-to-End*, the Transformer-based model outperforms the baselines: +50% of error reduction on *AIC* tasks between *ExPostGrounding*$_{W2V}$ at 80.91% and *End-to-End*$_{W2V}$ at 90.03%. The lexical grounding based on neural representations seems indeed robust in retrieving entities, while the attention mechanism within the transformer effectively grounds the correct interpretation. The transformer effectively learns how to map entities in the descriptions of the map associated with the input command to the correct entities in the produced interpretations. It also seems to improve the results of GrUT[5]. The differences between the EM, LS and WE are here more evident. When the Exact Match fails, no entity is retrieved and the argument cannot be included in the final command. On the contrary, smoother measures like the one based on WEs may introduce a super-set of the correct entities, thus introducing some noise; however, the transformer seems effective in pruning those entities not involved in the command. The different retrieval policies slightly affect the FP tasks: the Exact Match generally ignores several entities not retrieved in the input, and it negatively affects the capability of frame disambiguation. The effect of LS allows improving the quality of frame disambiguation also against the WE: we speculate that if "extra" entities not mentioned in the command do help the overall grounding process from one point of view, from the other these do not help the frame prediction substep.

Table 3
Error analysis of the GrUT - *End-to-End* model (and different retrieval policies) applied to the command *"take the book that is in the kitchen"*.

Retrieval Policy	Output	Correct
Exact Match	BRINGING(THEME(*'the book'*), GOAL(*'the bedroom'*))	NO
Levenshtein Similarity	BRINGING(THEME(*'the book'*), GOAL(s_6))	NO
Word Embeddings	BRINGING(THEME(w_6)) GOAL(s_6))	YES

The error analysis summarized by the example in 3 confirms that most of the misinterpretations are due to errors in the linguistic grounding phase. This example refers to a simple map with two entities, w_6 that is instance of the BOOK class, with the only lexical reference *volume* and s_6 that is instance of the ROOM class, with the only lexical reference *guest room*. Given the command *"take the book to the bed room"*, the GrUT-*End-to-End* model adopting the Exact Match is not able to retrieve any entity, and, even though it is able to infer the correct predicate and

---

[5]The evaluation of this paper follows a slightly different policy than [4] : here the "simpler" measure just requires that since an interpretation include the correct frame, arguments and mentioned entities; in [4] evaluation is correct only when also all words in the text are correctly mapped to their corresponding frames and arguments.

arguments, it simply rewrites the input text. When using Levenshtein Similarity, the "guest room" is correctly linked so that the second argument is correctly grounded. The adoption of the cosine similarity applied in the Neural Word Embedding space allows generating the input *"take the book to the bed room # w_6 also known as volume is an instance of class BOOK & s_6 also known as guest room is an instance of class BEDROOM & w_6 is far from s_6"*, that leads to the correct interpretation. Most of the errors involve entities whose lexical references are exactly one and cannot be retrieved by the linguistic grounding function because they are uncommon nouns, e.g. the lexical reference *volume* for the Book type entity.

5. Conclusions

This paper presents an End-to-End sequence-to-sequence process for Grounded Semantic Role Labeling. The proposed approach suggests providing the input text, which expresses the user's utterance enriched by a description of the surrounding environment expressed in natural language, as the input of a Transformer-based architecture. Correspondingly, the desired output is a logical form in which the entities of the environment are correctly associated with the command and grounded as well. Several policies have been applied as strategies for retrieving the entities from the map that are involved by the input utterance.

The experimental results confirm the robustness of the presented methods, especially when compared with traditional architectures chaining the linguistic interpretation of the utterance and then the linguistic grounding of the involved entity. This result is widely applicable as it does not require a costly adaptation to a particular scenario or domain.

Future work will extend the proposed methodology to consider additional properties of the environment (e.g., object properties crucial to disambiguate multiple instances of the same class, such as multiple books in a map), the user profile, or information extracted during the previous interactions between the user and the robot (exploiting the dialogue history).

References

[1] D. Nozza, L. Passaro, M. Polignano, Preface to the Sixth Workshop on Natural Language for Artificial Intelligence (NL4AI), in: D. Nozza, L. C. Passaro, M. Polignano (Eds.), Proceedings of the Sixth Workshop on Natural Language for Artificial Intelligence (NL4AI 2022) co-located with 21th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2022), November 30, 2022, CEUR-WS.org, 2022.

[2] M. E. Foster, Natural language generation for social robotics: opportunities and challenges, Philosophical Transactions of the Royal Society B: Biological Sciences 374 (2019) 20180027. doi:10.1098/rstb.2018.0027.

[3] A. Vanzo, D. Croce, E. Bastianelli, R. Basili, D. Nardi, Grounded language interpretation of robotic commands through structured learning, Artif. Intell. 278 (2020). doi:10.1016/j.artint.2019.103181.

[4] C. D. Hromei, C. Lorenzo, D. Croce, R. Basili, Embedding contextual information in seq2seq models for grounded semantic role labeling, in: (*submitted at the*) AIxIA 2022 - Advances in Artificial Intelligence - 21th International Conference of the Italian Association for Artificial Intelligence, 2022.

[5] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, ArXiv abs/1910.13461 (2020).

[6] C. J. Fillmore, Frames and the semantics of understanding, Quaderni di Semantica 6 (1985) 222–254.

[7] A. Kalyanpur, O. Biran, T. Breloff, J. Chu-Carroll, A. Diertani, O. Rambow, M. Sammons, Open-domain frame semantic parsing using transformers, 2020. URL: https://arxiv.org/abs/2010.10998. doi:10.48550/ARXIV.2010.10998.

[8] R. Blloshmi, S. Conia, R. Tripodi, R. Navigli, Generating senses and roles: An end-to-end model for dependency- and span-based semantic role labeling, in: Z. Zhou (Ed.), Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, ijcai.org, 2021, pp. 3786–3793. URL: https://doi.org/10.24963/ijcai.2021/521. doi:10.24963/ijcai.2021/521.

[9] M. Palmer, D. Gildea, N. Xue, Semantic Role Labeling, Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers, 2010. URL: https://doi.org/10.2200/S00239ED1V01Y200912HLT006. doi:10.2200/S00239ED1V01Y200912HLT006.

[10] D. Gildea, D. Jurafsky, Automatic Labeling of Semantic Roles, Computational Linguistics 28 (2002) 245–288. URL: https://doi.org/10.1162/089120102760275983. doi:10.1162/089120102760275983.

[11] C. Baker, M. Ellsworth, K. Erk, SemEval-2007 task 19: Frame semantic structure extraction, in: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 99–104. URL: https://aclanthology.org/S07-1018.

[12] X. Carreras, L. Màrquez, Introduction to the CoNLL-2005 shared task: Semantic role labeling, in: Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005), Association for Computational Linguistics, Ann Arbor, Michigan, 2005, pp. 152–164. URL: https://aclanthology.org/W05-0620.

[13] S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, Y. Zhang, Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes, in: Joint Conference on EMNLP and CoNLL-Shared Task, 2012, pp. 1–40.

[14] D. Das, D. Chen, A. F. T. Martins, N. Schneider, N. A. Smith, Frame-Semantic Parsing, Computational Linguistics 40 (2014) 9–56. URL: https://doi.org/10.1162/COLI_a_00163. doi:10.1162/COLI_a_00163.

[15] D. Marcheggiani, I. Titov, Encoding sentences with graph convolutional networks for

semantic role labeling, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 1506–1515. URL: https://aclanthology.org/D17-1159. doi:10.18653/v1/D17-1159.

[16] J. Zhou, W. Xu, End-to-end learning of semantic role labeling using recurrent neural networks, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Beijing, China, 2015, pp. 1127–1137. URL: https://aclanthology.org/P15-1109. doi:10.3115/v1/P15-1109.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, 2017, pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[18] D. S. Sachan, Y. Zhang, P. Qi, W. Hamilton, Do syntax trees help pre-trained transformers extract information?, arXiv preprint arXiv:2008.09084 (2020).

[19] P. Shi, J. Lin, Simple bert models for relation extraction and semantic role labeling, arXiv preprint arXiv:1904.05255 (2019).

[20] N. Wang, J. Li, Y. Meng, X. Sun, J. He, An mrc framework for semantic role labeling, arXiv preprint arXiv:2109.06660 (2021).

[21] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[22] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized BERT pretraining approach, CoRR abs/1907.11692 (2019). URL: http://arxiv.org/abs/1907.11692. arXiv:1907.11692.

[23] C. Raffel, N. M. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, ArXiv abs/1910.10683 (2020).

[24] M. Palmer, D. Gildea, P. Kingsbury, The Proposition Bank: An annotated corpus of semantic roles, Computational Linguistics 31 (2005) 71–106. URL: https://aclanthology.org/J05-1004. doi:10.1162/0891201053630264.

[25] S. Yang, Q. Gao, C. Liu, C. Xiong, S.-C. Zhu, J. Y. Chai, Grounded semantic role labeling, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016, pp. 149–159. URL: https://aclanthology.org/N16-1019. doi:10.18653/v1/N16-1019.

[26] M. Sahlgren, The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-Dimensional Vector Spaces, Ph.D. thesis, Stockholm University, Stockholm, Sweden, 2006.

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: C. J. C. Burges, L. Bottou, Z. Ghahramani, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Sys-

tems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, 2013, pp. 3111–3119. URL: https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html.