A User-centred Analysis of Explanations for a Multi-component Semantic Parser

Juliano Efson Sales[1,2(✉)], André Freitas[3], and Siegfried Handschuh[2]

[1] Department of Computer Science and Mathematics, University of Passau, Passau, Germany
[2] Institute of Computer Science, University of St. Gallen, St. Gallen, Switzerland
{juliano.sales,siegfried.handschuh}@unisg.ch
[3] School of Computer Science, The University of Manchester, Manchester, UK
andre.freitas@manchester.ac.uk

Abstract. This paper shows the preliminary results of an initial effort to analyse whether explanations associated with a semantic parser help users to generalise the system's mechanisms regardless of their technical background. With the support of a user-centred experiment with 66 participants, we evaluated the user's mental model by associating the linguistic features from a set of explanations to the system's behaviour.

Keywords: Explainable AI · User-centred analysis · Semantic parsing

1 Introduction

Archetypal natural language understanding (NLU) systems, such as question answering, natural language interfaces and semantic parsers, typically require the complex coordination of multiple natural language processing components, where each component can explore a large spectrum of resources and learning methods [3]. Offering end-user explanations for intelligent systems has becoming a strong requirement either to comply with legal requirements [5] or to increase the user confidence [11]. However, while delivering a human-interpretable explanation for a single component is challenging, the problem is aggravated in the context of multi-component systems [3,11].

Although the literature shows explanation models evaluated from an user-centred perspective, none of them targeted an NLU system [9,13,19,21]. As natural language gives vast possibilities of expression, explanations of NLU systems can allow the users to adapt their writing styles to favour the system comprehension according to the underline model.

This work analyses different types of explanations instantiated in a multi-component semantic parsing system for an end-user natural language programming task to analyse to what extent users, irrespective of their technical background, are able to improve their mental models by associating the linguistic features from the explanations to the system's behaviour.

E. Métais et al. (Eds.): NLDB 2020, LNCS 12089, pp. 37–44, 2020.
https://doi.org/10.1007/978-3-030-51310-8_4

2 Related Work

Lipton [11] defined a comprehensive taxonomy of explanations in the context of AI, highlighting various criteria of classification such as motivation (*trust*, *causality*, *transferability*, *informativeness* and *fairness & ethics*) and property (*transparency* and *post-hoc interpretability*).

Trust is by far the most common motivation presented in the literature, like Pazzani [13], and Biran & Cotton [2] whose results showed users demonstrate higher confidence when using a system they understand how it works. *Fairness & ethics* is also a strong driver as the well-known European General Data Protection Regulation [5] guarantees both rights *"for meaningful information about the logic involved"* and *"to non-discrimination"* to prevent bias and unfair behaviour.

Diversely, *post-hoc* explanations make use of interpretations to deliver meaningful information about the AI model. Instead of showing how the model works, it presents evidences of its rationale by making use of *(i)* textual descriptions [18], *(ii)* visualisations able to highlight image parts from which the decision was made [17], *(iii)* 2D-representation of high-dimensional spaces [12], or *(iv)* explanation by similarity [4].

3 Semantic Parsing of Natural Language Commands

The Problem The problem of *semantic parsing of natural language commands* consists of mapping a natural language command to a formal representation, called *function signature*, from a knowledge base (KB) of APIs.

We formalise the target problem as follows. Let F be a KB composed of a set of k function signatures $(f_1, f_2, \ldots, f_k)$. Let $f_i = (n_i, l_i, P_i)$ be an element of F, where n_i is the *function's name*, l_i is the *function's provider*, and P_i is the set of *function's parameters*. Let f_i' be a call of f_i, which also holds values for their parameters, totally or partially. Let c_j be a natural language command which semantically represents a target function call f_j'. The parser aims at building a ranking model which, given a set of function signatures F and a natural language command c, returns a list B of ordered function calls, satisfying the command intent.

The Semantic Parser Our end-user study is focused on an explanation model for a multi-component semantic parser proposed by Sales et al. [14], which is composed of a chain of components. Given the space restriction, the semantic parser is briefly summarised in this section.

The first component performs a semantic role labelling (SRL) classification of the command tokens, segmenting and identifying the *(i) function descriptor* and *(ii)* the set of *command objects*. The *function descriptor* is the minimal subset of tokens present in the command that allows identifying the target function signature in the function KB. A *command object* represents a potential descriptor or value of a parameter. It is implemented based on an explicit grammar defined by dependency relations and POS-tags.

The second component is the *Type Inferencer* which plays the role of a named entity recogniser. The *Inferencer*'s implementation combines heuristics with a gazetteer.

Based on the function descriptor and the list of command objects, the model generates potential function calls by combining the set of command object and the list of function signatures. For each function call, the *Relevance Classifier* generates a classification as (*i*) wrong frame (score 0); (*ii*) right frame with wrong parameters (score 1); (*iii*) right frame with partial right parameters (score 2); (*iv*) right frame with right parameters (score 3). The classification phase is implemented as a Random Forest model [6], which take as input the *semantic relatedness scores* and *densities* (described below) to identify jointly the most relevant function signature and the best configuration of parameters values.

Originally, the proposed semantic parser [14] defined an extra component responsible for reducing the search space. As the explanation model is evaluated in a setting with a restricted data set, we simplified the architecture by removing this component. Thus, the inference process can be described by Eq. 1, which defines the ranking score of a given function call for a natural language command, where $\delta(x)$ is a type inferencer that, given an expression in natural language x, it return its semantic type. For example, $\delta(\text{“}dollar\text{”}) = \texttt{CURRENCY}$ and $\delta(\text{“john@domain.com”}) = \texttt{EMAIL}$; $\boldsymbol{x}$ is a vector representation of x in a word embedding model; $\cos(\boldsymbol{x}, \boldsymbol{y})$ is a semantic similarity function, which score how similar the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are in the space. We use the cosine similarity for this purpose; $\bigodot_{i=1,j=1}^{n,k}(x_{ij})$ is a combinatorial optimiser that finds a maximum weight matching j to i. We use the Hungarian algorithm [8] for this purpose; and $den(p_i)$ is the set of the densities of the function parameters, which represents the inverse term-frequency in the function signatures vocabulary set.

$$cos(\boldsymbol{n}, \boldsymbol{d}) + \max_{j=1}^{k}(\cos(l, o_j)) + \sum \bigodot_{i=1,j=1}^{n,k}(cos(\boldsymbol{p_i}, \boldsymbol{\delta(o_j)})) + 1000 * \tau \qquad (1)$$

The equation defines the sum of (*i*) the semantic relatedness of the function descriptor from the command and the function name, (*ii*) the maximum semantic relatedness of the command objects and the function provider, (*iii*) the combinatorial optimisation of the command objects' types and the function's parameters, and (*iv*) the function signature class τ multiplied by a large weight.

4 Explanation of a Multi-component Semantic Parser

As heterogenity is an intrinsic characteristic of a multi-component AI system, demanding different explanation methods to different parts of the application, we organised the explanation in a hierarchical fashion motivating the construction of a model that is suitable for users with different levels of knowledge in machine

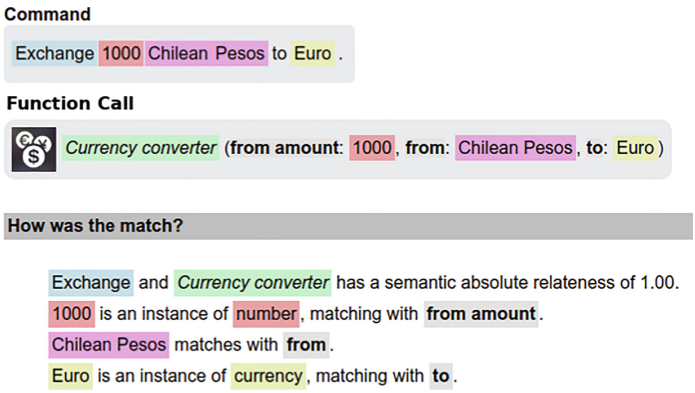

Fig. 1. Explanations of the *Semantic Role Labeler* and the *Type Inferencer*.

learning and linguistics. The explanation model, then, explores a hierarchical representation in an increasing degree of technical depth. In this context, it presents transparency-oriented explanations in the higher levels, going gradually to explanation that demand technical knowledge, and to the *post-hoc* ones.

The model presents seven explanations grouped by three layers focused on the components' behaviour and their input features. As expected in a heterogeneous architecture, each component operates under a different method and have different types of inputs.

SRL Rules and Syntactic Tree Layers. The first level describes the *Semantic Role Labeler* and the *Type Inferencer*. The explanations show the rules activated *(i)* to identify the command objects, *(ii)* to generated multi-word objects, and *(iii)* to identify the semantic types, highlighting the tokens and features involved in the process as shown in Fig. 1. The second level depicts the features on which the rules operate, namely the syntactic tree and the part of speech (POS) of each token. Figure 2 shows a natural language command and both the set of POS-tags and the dependency tree associated with its tokens. These layers aim at showing the connection between the linguistic features and main concepts of the parsing system, whose interpretability is dependent on the understanding of the role of linguistic features in the classification.

Word Embedding Layer. The matching process relies on the semantic relatedness scores, which represent the degree of semantic similarity the function descriptor and command objects have in relation to the function signature [16]. The semantic relatedness is calculated from a word embedding model, which represent terms as vectors in a high-dimensional space. The explanation provides a cluster-based visualisation using t-SNE [12], where it plots the semantic elements that plays a role in the matching process from both the command and the

function signature as shown in Fig. 3. The cosine between the points represents the degree of semantic relatedness in a typical *post-hoc* explanation fashion.

The Ranking and Classification Layer. The lower level is devoted to the most technical explanations which shows the mathematical expression that defines the final ranking score of the function signature along with the features used in both the expression itself and in the function relevance classifier. To simplify the model to non-technical users, we reduced Eq. 1 to $\sum_{i=0}^{n}(z_i) + 1000 * \tau$, where all elements in the expression is represented by z, the vector of all features. Additionally, this level also presents the trained random forest classifier, showing the relevance of each feature in the final classification using the visualisation proposed by Welling et al. [20], called *Random Floor*.

5 Evaluation

We asked the participants to simulate the use of a semantic parser, in which the user inputs the natural language commands, and the system suggests a list of function calls as depicted in Fig. 4. We showed twelve pre-configured natural language commands and their corresponding list of 3 to 5 potential function calls as a result of the execution of the parser. The pre-configured commands as well as the function signatures came from the data set defined in the Task 11 of the SemEval 2017 [15], which presents a broader set of functions and describes commands closer to the daily routine of end users.

Mental Models. A mental model is a cognitive representation of the external world to support the human reasoning process [7]. In our task, the "external world" is represented by the semantic parsing system, and we evaluate the user's mental model by assessing whether the presented explanations help the user to generalise the system's mechanisms. So, we designed a set of questions to measure whether the user realised the correct influence of linguistic features in the overall performance of the parser in both SRL and classification phases. Given a contextual command, the participants were asked to judge affirmative sentences in a Likert 7-point scale [10]. We evaluated three aspects of the SRL: *(i)* the role of proper nouns, *(ii)* the importance of the correct spelling and use of grammar and *(iii)* the verb mood (indicative *vs.* imperative).

Proper nouns are generally written in capitalised letter in English. As proper nouns define a command object, we want to identify to what extend users identify the impact of this feature in the system's performance. After given a contextual command, we asked the participants to judge the veracity of sentences like *"Writing 'Swiss Francs' with capital letter increases the system comprehension"*.

Incomplete sentences might introduce errors in the POS-tagger and grammar tree parser, which on the other hand leads to wrong interpretation about the objects. In this task, we present grammatically incomplete commands (keyword-search style) to support users in the identification of the importance of grammatically correct sentences instead of keywords, such as traditional information retrieval systems. we asked the participants to judge the veracity of sentences

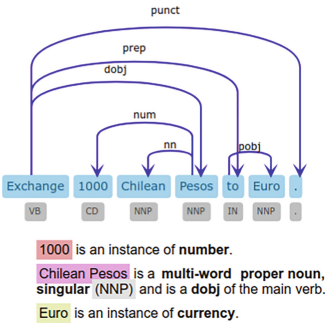

Fig. 2. Grammar tree.

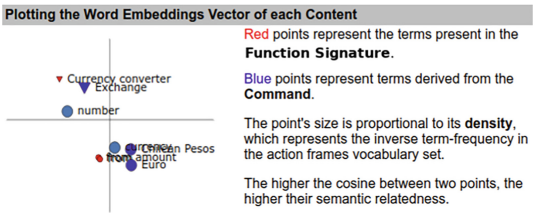

Fig. 3. Plot of the elements from the command and function signature in which the cosine between the points represents the semantic relatedness.

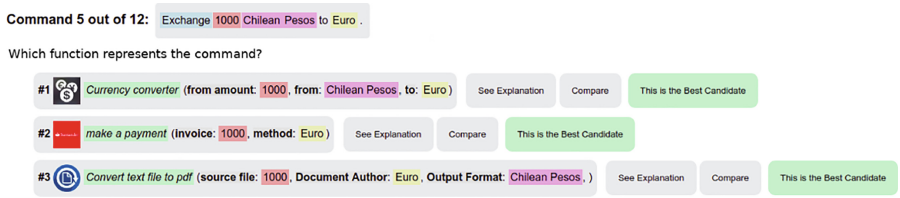

Fig. 4. A natural language command and a list of potential function signatures representing its intent.

like *"Writing a set of keywords for the command has the same result as grammatically correct sentences"*.

Regarding verb moods, we presented to the participants commands written as questions and in the indicative form. After given a contextual command, we asked the participants to judge the veracity of sentences like *"Starting by 'I would like' increases the system comprehension"*.

Participants. We recruited 66 adult participants from the authors' professional networks whose unique requirement was to be fluent in English. The set of participants is composed of 26 females and 40 males, whose age vary from 20 to 49. They reported their level of knowledge in *machine learning* (ML) and *English grammar* (EG) according to the same scale suggested by Azaria et al. [1], to which we attributed a score from 1 to 6 respectively *none*; *very little*; *some background from high school*; *some background from university*; *significant knowledge, but mostly from other sources*; *bachelor with a major or minor in the related topic*.

The participants were divided randomly into the *control group* composed of 34 participants, which have access to the system without the explanation, and the *treatment group*, composed of 32 participants, with access to the explanation. The random division longed for balancing the number of participants with and without ML knowledge in each group.

We introduced the experiment to the participants by exposing its main goals and the expected procedures in the task. We highlighted that the idea behind the parser is to allow a user to find suitable functions and their parameters from her/his commands expressed in natural language, regardless of her/his technical knowledge. We asked them to select the correct function call for each pre-configured command, while examining the tool to infer how it works. For the users that participated in the treatment group, we encouraged them to see the explanations, which shows how the system maps commands to the function calls.

Table 1. The results regarding the mental model assessment.

Metrics	Treatment group		Control group	
Average	1.13		0.73	
r (ML)	0.54		0.45	
r (EG)	0.45		0.46	
	Acquainted	Non-	Acquainted	Non-
Avg. (ML)	1.62	0.63	1.07	0.54
Avg. (EG)	1.42	0.62	1.11	0.34

6 Results and Discussion

We associated the answer in the Likert 7-point scale to the interval -3 to 3, where 0 is the neutral answer and 3 represents *strongly agree* when the question reflects a true statement, and *strongly disagree* when it represents a false statement. We also analysed the statistical significance of the results using the *t-test*, which is represented by p.

Table 1 presents the results of the mental model assessment. On average, participants in the treatment group give scores 55% higher than those in the control group (1.13 *vs.* 0.73, $p < 0.05$). The results also demonstrate that knowledge in machine learning and English grammar have significant positive relationship with the mental model scores in both treatment group ($r = 0.46$ for ML, $r = 0.40$ for EG) and control group ($r = 0.45$ for ML, $r = 0.41$ for EG). The invariance of the correlation coefficients among the groups and the mental model scores strongly suggest the explanation model helps users to build better mental models. To explicitly present this conclusion, we divided both treatment and control groups into four subgroups according to their knowledge in ML. We considered acquainted with ML those users that declared having *significant knowledge, but mostly from other sources* or a *bachelor with a major or minor in the topic*. In average, the score of the users acquainted with ML in the target group is 1.62, while 1.07 in the control group ($p < 0.05$). Although not being the focus of our study, the results concerning EG knowledge present a similar tendency as shown in Table 1.

7 Conclusion

Our experiments showed evidences explanations are an effective method to build mental models, regardless of the users' technical background. The experiment also suggests technical knowledge is boosted when accompanied by explanations, given its high correlations with mental model scores.

References

1. Azaria, A., et al.: Instructable intelligent personal. In: AAAI 2016 (2016)
2. Biran, O., Cotton, C.: Explanation and justification in machine learning: a survey. In: Workshop on Explainable AI (XAI), IJCAI 2017, p. 8 (2017)
3. Burgess, A.: AI prototyping. In: Burgess, A. (ed.) The Executive Guide to Artificial Intelligence, pp. 117–127. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-63820-1_7
4. Caruana, R., Kangarloo, H., Dionisio, J.D., Sinha, U., Johnson, D.: Case-based explanation of non-case-based learning methods. In: Proceedings of AMIA Symposium (1999)
5. Goodman, B., Flaxman, S.: European union regulations on algorithmic decision-making and a "right to explanation". AI Mag. **38**, 50–57 (2017)
6. Ho, T.K.: Random decision forests. In: Proceedings of the Third International Conference on Document Analysis and Recognition (1995)
7. Jones, N., Ross, H., Lynam, T., Perez, P., Leitch, A.: Mental models: an interdisciplinary synthesis of theory and methods (2011)
8. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Res. Logist. Q. **2**(1–2), 83–97 (1955)
9. Kulesza, T., Burnett, M., Wong, W.K., Stumpf, S.: Principles of explanatory debugging to personalize interactive machine learning. In: IUI 2015 (2015)
10. Likert, R.: A technique for the measurement of attitudes (1932)
11. Lipton, Z.C.: The mythos of model interpretability. In: Proceedings of the ICML 2016 Workshop on Human Interpretability in Machine Learning (2016)
12. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**, 2579–2605 (2008)
13. Pazzani, M.J.: Representation of electronic mail filtering profiles. In: IUI 2000 (2000)
14. Sales, J.E., Freitas, A., Handschuh, S.: An open vocabulary semantic parser for end-user programming using natural language. In: 12th IEEE ICSC (2018)
15. Sales, J.E., Handschuh, S., Freitas, A.: Semeval-2017 task 11: end-user development using natural language. In: SemEval-2017 (2017)
16. Sales, J.E., Souza, L., Barzegar, S., Davis, B., Freitas, A., Handschuh, S.: Indra: a word embedding and semantic relatedness server. In: 11th LREC (2018)
17. Selvaraju, R.R., et al.: Grad-CAM: why did you say that? Visual explanations from deep networks via gradient-based localization (2016)
18. Silva, V.D.S., Handschuh, S., Freitas, A.: Recognizing and justifying text entailment through distributional navigation on definition graphs. In: AAAI 2018 (2018)
19. Stumpf, S., et al.: Toward harnessing user feedback for machine learning. In: IUI 2007 (2007)
20. Welling, S.H., Refsgaard, H.H.F., Brockhoff, P.B., Clemmensen, L.K.H.: Forest floor visualizations of random forests (2016)
21. Zhou, J., et al.: End-user development for interactive data analytics: uncertainty, correlation and user confidence. IEEE Trans. Affect. Comput. **9**, 383–395 (2018)