

Demonstrating the MYCBR Rest API

Kerstin Bach, Bjørn Magnus Mathisen, and Amar Jaiswal

Norwegian University of Science and Technology (NTNU)
Department of Computer Science & Norwegian Open AI Lab
Trondheim, Norway
{kerstin.bach,bjornmm,amar.jaiswal}@ntnu.no

Abstract. Case-based reasoning (CBR) tools are important to reduce the effort of developing CBR systems. MYCBR has been a tool for researchers and practitioners over the last ten years providing CBR system building blocks and functionality through the MYCBR-SDK and means to develop CBR models in the MYCBR-workbench. In this paper we present the MYCBR Rest API which exposes the functionality of both MYCBR-SDK and MYCBR-workbench through a RESTful API. It includes the MYCBR-SDK functionality to enable researchers the fast development and experimentation of CBR applications from not only Java, but from the programming language of the developers choice. Most of the MYCBR-workbench functionality has also been exposed in the same fashion enabling users to programmatically create, modify and delete CBR models and case-bases, so that the Rest API also allows MYCBR to act as a service, and be accessed by the client software through HTTP.

Keywords: Case-based Reasoning · CBR tools · MYCBR

1 Introduction

CBR tools have been developed since the very beginning of the CBR research activities. The most general CBR tools developed and provided as bundled or open source software are COLIBRIStudio (and their predecessors COLIBRI, jCOLIBRI) [5], CBRworks [16] and its successor MYCBR[18]. Furthermore there are more specific CBR tools targeting certain domains or case representations. For process-oriented CBR, the Collaborative Agent-based Knowledge Engine (CAKE) [3] has been introduced, while CREEK [1] is a tool for knowledge-intensive CBR and (B)EAR [9, 10] focuses on the adaptation in CBR systems. In addition there are also application specific tools such as eXiT*CBR for medical diagnosis.

MYCBR¹, which is the basis of this work, was developed by German Research Center for Artificial Intelligence (DFKI) and has been introduced as rapid prototyping tool for research and industrial applications. Initially, MYCBR was introduced as an add-on to the Protégé ontology tool [13], but later re-implemented

¹ <http://mycbr-project.org>

as a Java standalone tool and software development kit (SDK) [2]. Up until today, using MYCBR still required the CBR system to be integrated in a Java environment.

Allowing users to model a CBR system using MYCBR’s workbench and then deploying the application as a web service would make it easier to build, test, compare and deploy CBR applications.

Much of the current success of Machine Learning can be linked to the availability of machine learning models in tools like Scikit-learn [14] and Keras [4]. Such tools allows researchers and developers to use the methods more easily and provide reproducible results [15]. Currently successful Machine Learning APIs are DialogFlow², TensorFlow.js³ or the Microsoft Cognitive Toolkit⁴. They all have in common that users interact with them via web pages or web services. Users of such services take responsibility for the provision of data, configuration of each method’s parameters, and validation and verification of the results. The services provide the functionality of the core methods.

RESTful (REpresentational State Transfer) Application Programming Interface (API) [7] is the current state-of-the-art to provide web services. Restful APIs, or Rest APIs, have been developed and their services have been deployed in industry over many years. However, in research they have not received a lot of attention. We believe, however, that implementing research prototypes using a Rest API can be a way forward to ease the development of systems in general, and CBR systems in particular.

2 MYCBR Rest API

MYCBR applications are typically built as a Java application on top of the MYCBR-SDK Java library. MYCBR-REST is designed to expose MYCBR’s modeling functionality (e.g. creating concepts and similarity functions) and MYCBR’s runtime functionality through a HTTP REST API. This enables the user to programatically access the features of what was previously exposed through two different tools (MYCBR workbench and MYCBR-SDK) into an API. This API is also conveniently accessible from all programming languages that support accessing HTTP REST APIs. Figure 1 shows the architecture of this design.

The Rest API has been implemented using the Spring Boot Framework and configured to expose its documentation via a tool called swagger. From swagger a developer can use the interactive documentation for testing requests and developing applications.

During the demo, we will show how use the Rest API to obtain information about the case representation, carry out the retrieval as well as evaluating the results. We will provide examples using Python 3, however the Rest API can be used with all programming languages languages that support Rest API and parsing JSON objects.

² <https://dialogflow.com/>

³ <https://www.tensorflow.org/js/>

⁴ <https://docs.microsoft.com/en-us/cognitive-toolkit/>

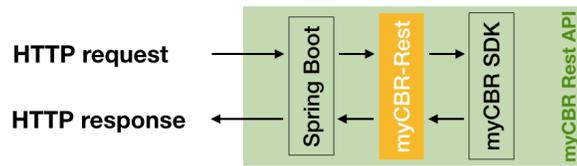


Fig. 1: Components of the myCBR-REST architecture.

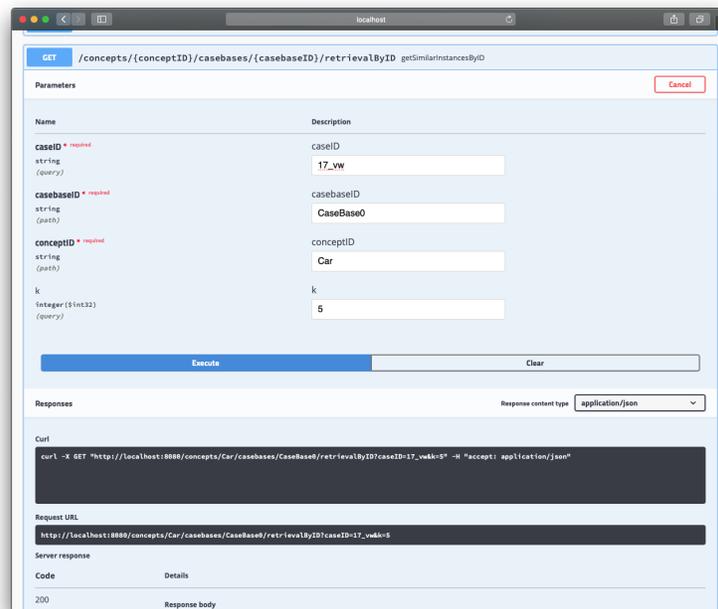


Fig. 2: Example of the Swagger tool that shows how a Rest call can be tested against a CBR application

2.1 Retrieval

Figure 3 shows how the retrieval against the Rest API can be implemented. The JSON data returned by the API can be directly included in data frames and further evaluated from there.

3 Experiments and Applications

The myCBR Rest API described in this demo has been used to build and experiment with CBR systems over the last years. In the following, we will describe the different areas of application.

```

In [2]: server = 'localhost'

port = '8080'
url = 'http://' + server + ':' + port + '/'

In [3]: def retrieve_k_sim_byID_content(concept, casebase, amalFct, queryID, k):
raw = pd.DataFrame(requests.get(url + 'concepts/' + concept + '/casebases/'
+ casebase + '/retrievalByIDWithContent?amalgamation%20function='+amalFct+'&caseID='
+ queryID + '&k=' + k).json())
results = raw.apply(pd.to_numeric, errors='coerce').fillna(raw).sort_values(by='similarity', ascending=False)
returned_df = results.apply(pd.to_numeric, errors='coerce').fillna(results)
return returned_df

In [4]: # Find most similar cases and get the full case content for them
retrieve_k_sim_byID_content('Car', 'CaseBase0', 'CarFunc', '17_vw', '5')

Out[4]:
```

	Body	CCM	Car Code	Color	Doors	Gas	Manufacturer	Miles	Model	Power	Price	Speed	Year	ZIP	caseID	similarity
0	sedan	1900	17	dark_blue	4	diesel	vw	71433	passat	90	22099	183	1995	6	17_vw	1
1	sedan	1900	891	blue	4	diesel	vw	68993	passat	110	23599	183	1995	2	891_vw	0.658371
2	sedan	1900	532	dark_blue	4	diesel	vw	13701	passat	90	33199	183	1996	3	532_vw	0.658114
3	sedan	1900	602	blue	4	gasoline	vw	80748	passat	193	26499	183	1996	1	602_vw	0.632703
4	sedan	1900	467	dark_blue	4	gasoline	vw	91911	passat	100	16699	183	1995	8	467_vw	0.632574

Fig. 3: Example of the Python Code carrying out a retrieval using the Rest API

3.1 Application in Research Projects

Up to today the integration of a CBR component as a web service required a Java-based framework. With the presented work, a CBR engine can be developed and easily deployed. One of the currently running CBR applications developed in the SELFBACK project [12] uses this infrastructure. Here the MYCBR-driven engine is the core of the decision support system to compare patient profiles and generate self-management plans for low back pain patients. Moreover the extended RESTful services are used to monitor and evaluate the patient's progress and provide responses to clinicians.

Furthermore, research prototypes have been developed in the last three years. Skjold et al. [17] presented an application focusing on adapting sandwich recipes using CBR⁵. In this application the Rest API has been used as the backend to carry out the similarity-based retrieval and building ephemeral case bases for the adaptation processes. In Engin et al. [6], Verma et al. [19] and Jaiswal et al. [8] the Rest API was used in combination with Python to prototype and evaluate different CBR methods.

In Mathisen et al. [11] Rest API is used to evaluate the performance of different neural network architectures used as a amalgamation function. The neural amalgamation function can be used to add different pre-trained neural networks to be used a global similarity function.

3.2 Application in Education

During the last two years while the API has been developed, we introduced it to students who then implemented CBR systems during courses at Master's and PhD level. Especially web-based applications such as IntelliMeal ([17]) were created. In the beginning of the courses we gave an introduction explaining on

⁵ The prototype of IntelliMeal is available at <http://hv-6151.idi.ntnu.no>

how to create a MYCBR project using the MYCBR tutorial⁶, followed by the introduction to the MYCBR Rest API similar to the examples given in the aforementioned iPython Notebook. Until now, about ten MYCBR applications have been developed and experimented with at NTNU using the introduced approach while four more are currently under development.

Compared to the Java-only approach we see that the time until the first prototype is up and running is dramatically reduced. Also, for a basic CBR system, no Java knowledge is needed and students can focus on case representations, similarity measure development or adaptation strategies. The students found it convenient to interact through the REST interface, so that the integration of a CBR component in a more comprehensive data processing pipeline is feasible.

4 Conclusion and Outlook

This demo presents a further development of the MYCBR tool that allows developers and researchers easier prototyping, integration and deployment of CBR systems. The MYCBR Rest API creates CBR services that can be deployed and used anywhere as well as it allows a systematic evaluation of the CBR systems. All content provided in this paper is made available on GitHub under LGPL to be shared with the community⁷. In conclusion, the MYCBR Rest API provides a flexible framework for creating CBR systems and developing new components. Moreover, it lowers the entry bar to experiment with Case-Based Reasoning and compare it with other AI methods.

Acknowledgement

This work has been supported by the Norwegian Open AI Lab as well as NTNU's rector's funds for multidisciplinary research. Further, parts have been conducted within the SELFBACK research project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 689043 and the Exposed Aquaculture centre for research-based innovation funded by the centre partners and the Research Council of Norway.

References

1. Aamodt, A.: Knowledge-intensive case-based reasoning in CREEK. In: Advances in Case-Based Reasoning, 7th European Conference, ECCBR 2004, Madrid, Spain, August 30 - September 2, 2004, Proceedings. pp. 1–15 (2004)
2. Bach, K., Althoff, K.D.: Developing case-based reasoning applications using mycbr 3. In: Agudo, B.D., Watson, I. (eds.) Case-Based Reasoning Research and Development. pp. 17–31. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)

⁶ <http://mycbr-project.org/tutorials.html>

⁷ <https://github.com/orgs/ntnu-ai-lab/teams/mycbr/repositories>

3. Bergmann, R., Gessinger, S., Görg, S., Müller, G.: The collaborative agile knowledge engine CAKE. In: Proceedings of the 18th International Conference on Supporting Group Work. pp. 281–284. GROUP '14, ACM, New York, NY, USA (2014)
4. Chollet, F., et al.: Keras. <https://keras.io> (2015)
5. Díaz-Agudo, B., González-Calero, P.A., Recio-García, J.A., Sánchez-Ruiz-Granados, A.A.: Building cbr systems with jcolibri. *Science of Computer Programming* 69(1), 68 – 75 (2007), special issue on Experimental Software and Toolkits
6. Engin, H.K., Nadim, F., Carotenuto, P., Bach, K.: Estimation of pile capacities using case-based reasoning (cbr) method. In: Proceedings of 4th International Symposium on Computational Geomechanics (2018)
7. Fielding, R.T.: Architectural Styles and the Design of Network-based Software Architectures. Ph.D. thesis, University of California, Irvine (2000), aAI9980887
8. Jaiswal, A., Bach, K., Meisingset, I., Vasseljen, O.: Case representation and similarity modeling for non-specific musculoskeletal disorders - a case-based reasoning approach. In: FLAIRS-32 Conference. AAAI Press (2019)
9. Jalali, V., Leake, D.: Cbr meets big data: A case study of large-scale adaptation rule generation. In: Hüllermeier, E., Minor, M. (eds.) *Case-Based Reasoning Research and Development*. pp. 181–196. Springer International Publishing, Cham (2015)
10. Jalali, V., Leake, D.: Enhancing case-based regression with automatically-generated ensembles of adaptations. *J. Intell. Inf. Syst.* 46(2), 237–258 (Apr 2016)
11. Mathisen, B.M., Aamodt, A., Bach, K., Langseth, H.: Learning similarity measures from data. Submitted to *Progress in Artificial Intelligence* (2019)
12. Mork, P.J., Bach, K.: A decision support system to enhance self-management of low back pain: Protocol for the selfback project. *JMIR Res Protoc* 7(7), e167 (Jul 2018)
13. Musen, M.A.: The protégé project: A look back and a look forward. *AI Matters* 1(4), 4–12 (Jun 2015), <http://doi.acm.org/10.1145/2757001.2757003>
14. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.* 12, 2825–2830 (Nov 2011)
15. Recio-García, J.A., Díaz-Agudo, B., González-Calero, P.A.: The COLIBRI open platform for the reproducibility of CBR applications. In: *Case-Based Reasoning Research and Development - 21st International Conference, ICCBR 2013, Saratoga Springs, NY, USA, July 8-11, 2013. Proceedings*. pp. 255–269 (2013)
16. Schulz, S.: Cbr-works - a state-of-the-art shell for case-based application building. In: *Proceedings of the 7th German Workshop on Case-Based Reasoning, GWCBR'99, Wrzburg*. pp. 3–5. Springer-Verlag (1999)
17. Skjold, K., Øynes, M., Bach, K., Aamodt, A.: Intellimeal - enhancing creativity by reusing domain knowledge in the adaptation process. In: *ICCBR 2017 Workshops Proceedings, Trondheim, Norway, June 26-28, 2017*. pp. 277–284 (2017)
18. Stahl, A., Roth-Berghofer, T.R.: Rapid prototyping of cbr applications with the open source tool mycbr. In: *European conference on case-based reasoning*. pp. 615–629. Springer (2008)
19. Verma, D., Bach, K., Mork, P.J.: Modelling similarity for comparing physical activity profiles - a data-driven approach. In: Cox, M.T., Funk, P., Begum, S. (eds.) *Case-Based Reasoning Research and Development*. pp. 415–430. Springer International Publishing, Cham (2018)