Working with Ambiguous Case Representations

Joseph Kendall-Morwick[1]

[1]Washburn University, 1700 SW College Ave, Topeka, KS 66621, USA

Abstract

Case-based reasoning systems used in complex tasks (such as case-based design) often rely on complex case representations. The challenges posed by these cases have been mitigated in a variety of ways by careful attention to the implementation of the traditional "4R" cycle (Retrieval, Reuse, Revision, and Retention). This paper proposes a different but compatible perspective on managing the complexity of these tasks by recognizing and embracing the ambiguity within complex case representations (ambiguity meaning that records of complex problem solving experiences may be interpreted in a number of ways and may not fit the traditional constraints of a singular case). Specifically, this paper will propose an abstract framework for retaining ambiguous experiential knowledge from which cases can be extracted at retrieval time. This framework is compared to and aligned with related approaches to managing complex cases and explained through the specification of a work in progress CBR system for form understanding.

Keywords

Case Representation, Process-Oriented Case-Based Reasoning, Case-Based Design, Form Understanding

1. Introduction

Consider the following scenario:

> Simon is preparing dinner for his friends and considering his options. He wants to go with chili but knows there will be several vegetarians at the dinner. Looking through recipes for vegetarian chili, he chooses one that features beans and smoked peppers. He is now unsure what desert would go well with his choice of main. He recalls the last dinner party he had been to with a smoky, earthy main dish (burritos, in this case) had a desert, churros, that worked well. Finally, he notes that his chili recipe contains coriander – an ingredient at least one of his guests has an aversion to. Searching for advice, he finds a blog post for a curry recipe that suggests substituting cumin for coriander.

Computer aided cooking has been a popular domain with the CBR community, perhaps not because we are all such bad cooks that we need a lot of computer assistance, but more so the opposite: it serves as a salient example for applying the CBR cycle to problems with complex case representations because we are mostly all familiar with the reasoning process. In this example, we can see the complexity and inherent structure of the cases: dinner plans involve

ICCBR BEAR'23: Workshop on Beyond Attribute-Value Case Representation at ICCBR2023, July 17 – 20, 2023, Aberdeen, Scotland

Joseph.Kendall-Morwick@washburn.edu (J. Kendall-Morwick)

individual recipes which involve individual ingredients and cooking techniques. However, in Simon's reasoning process, we also see how traditional notions of CBR may be challenged.

The prior experience that Simon seeks to leverage comes in a number of formats. He recalls a memory of an overall dinner plan (a case encompassing several dishes). He reads recipes for single dishes, some he reuses entirely where the 'solution' of the case appears to be the recipe itself, and others he extracts just a small element from; the 'solution' for his coriander problem is simply to replace one ingredient. Simon is able to evaluate all of this experiential knowledge and distill from it the relevant case details he needs to develop his dinner plans, but the structure and boundaries of the 'cases' he leverages are not just complex (cannot be represented with a simple feature vector), they are ambiguous (able to be interpreted in multiple ways). Why was cumin of interest in the curry recipe and not coconut milk? Why wasn't the curry a solution to all of his constraints with the main dish – it also can be prepared vegetarian. These cases could be reused in multiple ways with different problem descriptions to consider and different solutions offered, but these case details aren't apparent until Simon views them through the lens of a particular sub-task within his dinner plan that he is working on.

Although this example is perhaps most relevant to case-based design, the ambiguity involved in the scenario is apparent in many CBR domains involving complex cases, especially where tasks are complex and involve multiple sub-tasks, or when multiple problem solving episodes are related and directly influenced by solutions provided in prior episodes. This paper, for example, will examine the role of ambiguity in case representations in the domain of form understanding. However, even if this phenomenon was unique to case-based design, this paper makes the argument that the benefit of loosening our restrictions on what it means to be a case is beneficial. Put simply, for many problem domains involving complex cases, it is better to take a more abstract view of the case base as a knowledge container holding experiences whose interpretation as cases may be dependent on details of the problem to be solved.

This paper isn't introducing ambiguous cases as a brand new idea. There are many examples of what would qualify as ambiguous cases in existing CBR literature. Furthermore, this paper isn't intended to be an authoritative review of ambiguous cases in prior CBR projects. However, this paper makes several contributions:

- Several related projects are referenced to provide a broad view of ambiguous cases. Readers will be better versed in determining ambiguity present in their own CBR projects and how to connect this ambiguity with the complexity of the task their system performs.
- This view is applied in an abstract framework intended to be compatible with and improve existing approaches and frameworks dealing with ambiguous cases. Overall, this framework presents a new perspective in CBR that encourages CBR researchers using complex cases to consider how an *ambiguous* case representation could broaden the applicability of their work and improve their problem solving capabilities.
- The abstract framework is illustrated through its application to the domain of document understanding within a work in progress digitization project.

The following section reviews other prior work that recommends modifications to traditional CBR frameworks to address complex case structures and distinguishes these approaches from our proposed perspective. In the third section, a unifying view over ambiguous cases is developed through examination of the ambiguous qualities of complex cases used in other prior work. The fourth section develops an abstract framework that provides a focus for implementations or CBR frameworks looking to adopt this perspective. The fifth section introduces form understanding as a problem domain for CBR and details how the proposed perspective will be applied through an ongoing CBR research project implementing a case-based form understanding system. The last section outlines conclusions and directions for future work.

2. Prior Work Adapting CBR to Complex Case Structures

It is not uncommon for CBR researchers, particularly those using complex case representations, to consider frameworks with significant alterations to the traditional "4 R" case-based reasoning cycle (Retrieval, Reuse, Revision, and Retention) [1]. Working with complex cases can bring unique challenges for comparison of cases. For example, determining subgraph isomorphism (common when comparing graph-based complex cases) is NP-Complete [2]. The difficulty of comparing complex cases, in turn, significantly impacts case retrieval, adaptation, and retention. Because of this, CBR researchers have mitigated the impact of complex cases with CBR frameworks that take an alternative view to the 4R cycle. Such frameworks can make it easier to relieve storage and retrieval burdens associated with complex cases or to maximize the benefit of the knowledge these cases contain.

Eisenstadt et al. introduced FLEA-CBR to address a number of applications, but notably case-based design tasks (in particular floor plan design with MetisCBR) [3]. The framework uses 4 phases (find, learn, explain, and adapt) that can be ordered in several ways. Eisenstadt et al. developed this framework to address what they saw as the stifling sequential constraint of the 4R cycle, particularly for creative tasks such as case-based design. While this paper also amends the traditional definition of CBR, the primary focus is on how knowledge is structured and interpreted within the system. This does carry implications for the CBR cycle (specifically retrieval and retention), but does not necessarily shift away from the traditional 4R cycle as FLEA-CBR does. Instead, there is more of a focus on the definition of the knowledge containers utilized by a CBR system. These changes can be adopted instead of, or in addition to, any of the proposed changes to the 4R cycle in the referenced literature.

Such a focus on the knowledge containers of a CBR system to accommodate complex cases is also not unprecedented. Cordier et al. implement an approach to acquiring adaptation knowledge both from users and from data mining [4]. Klein et al. mitigate the challenges of complex cases in the POCBR domain by maintaining retrieval knowledge augmenting the case base [5]. The proposed perspective differs mainly from these by focusing primarily on the case knowledge container itself, how it is structured, and how it is utilized within the system. That is, the focus is on how experiential knowledge is represented by the CBR system. However, such extensions to traditional representations such as those suggested by this paper are also not new and have been studied by CBR researchers. A non exhaustive selection of such work is referenced in the following section while introducing a unifying perspective of ambiguity in cases along with specific advice towards developing CBR systems that work with complex case representations.

3. Recognizing Ambiguous Case Structures

For the purposes of this paper, the concept of complexity in case representations will be considered broadly to mean any significant divergence from a feature vector representation in which a case is composed of a fixed-size vector of atomic values. In this sense, any of the structured or semi-structured categories of representations recognized by Bergmann et al. would be considered "complex" [6]. An ambiguous case will be defined as a case having multiple interpretations for the purpose of problem solving. Although complexity in case representations often leads to ambiguity, they are not the same concept. This section will distinguish two principal sources of ambiguity frequently found in complex case representations: conceptual and compositional.

Conceptually ambiguous cases contain syntactically atomic abstractions that make the case ambiguous – that is, they contain symbols that cannot be decomposed but can be interpreted in multiple ways, semantically. For example, in the workflow domain, Bergmann and Gil developed a framework used in several workflow management CBR applications that provides for a conceptual hierarchy of labels applied to graph elements [7]. Such cases exist in many domains and are given many different names throughout the CBR literature but a common term and perspective taken is a 'generalized case'. Maximini et al. formally define a generalized case as a representation of a subset of the case-space [8]. Bichindaritz considers various ways in which these generalized cases are developed [9]. A generalized case can be a means of case base maintenance and case revision by retaining important qualities of a number of frequently occurring and mostly redundant cases, either mined from the case base itself, or developed by an expert. Although generalized cases represent more than one distinct reasoning episode and may contain abstract components, they are still mostly recognized as a coherent case, readily available for retrieval, and with distinct problem and solution descriptions.

Compositionally ambiguous cases do not necessarily contain such abstractions, but the boundaries of these cases may not neatly line up with a problem/solution pair. A compositionally ambiguous case may contain multiple problems with one solution, multiple solutions with one problem, or multiple cases entirely. They are distinct from conceptually ambiguous cases in that the case knowledge is represented concretely and not abstractly and the source of ambiguity is the boundary of the case rather than the concepts within it. One example is the Phala project (Kendall-Morwick and Leake) which did not have a strict delineation between problem and solution components of a case and instead extracted these components during the reuse phase [10].

Some case representations contain elements of both types of ambiguity. For example, semantic traces of stroke management episodes used by Montani et al. contained frequent series of actions that were abstracted in to a single "macro-action", making the case ambiguous both in terms of its content and concepts [11]. Zeyen et al. developed a system which extracted subworkflows (workflow streams) from workflow cases and utilized them as adaptation knowledge [12]. These cases also use the same semantic workflow framework previously mentioned to exhibit conceptual ambiguity [7].

Cases involving semi-structured data can typically be both conceptually ambiguous (using more abstract terms and concepts in its descriptions) or compositionally ambiguous (conveying numerous or otherwise structured concepts within the text). Berg et al. developed the FEATURE-

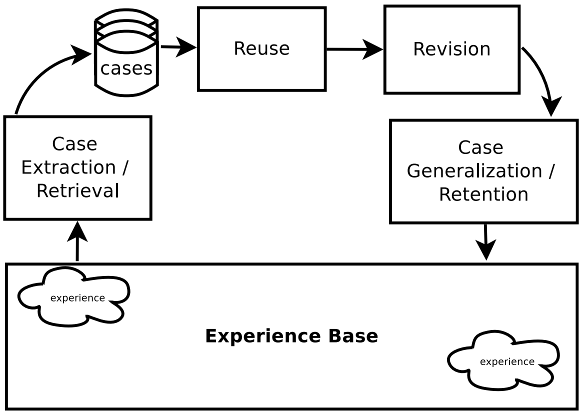


Figure 1: An Abstract Framework for Incorporating Ambiguous Cases

TAK framework to automate knowledge acquisition in the aviation domain [13]. Cases in FEATURE-TAK are essentially mined from text sources but the process illustrates both the ambiguity of their original source and how the retrieval process can be developed to extract a leveragable case representation from a more ambiguous source.

4. An Abstract Framework for Working with Ambiguous Cases

Distinct from prior work, this paper suggests a framework that views the case base as a more generalized knowledge container than a collection of discrete cases with distinct problem and solution components. From this perspective we could instead view this knowledge container as the *experience base*. This experience base (pictured in Figure 1) is the source of case knowledge leveraged in latter phases of the 4R (or modified) cycle. This name "experience base" perhaps implies the presence of discrete 'experiences'; however, this framework does not *require* that the content of the container necessarily be segmented in this way, nor does it require that elements of this container also have distinct problem and solution components. However it also does not forbid such structure. That is to say, a traditional case base still fits this framework. The perspective simply requires that discrete cases with distinct problem and solution components can be retrieved by some means from the experience base. The framework is considered abstract in the sense that fewer constraints are placed on representations of experience than for a traditional case representation, however this change does carry important implications for retrieval and retention as described below.

Retrieval within a system using a more generalized experience base involves not only finding the case knowledge necessary to present cases to the reuse phase, but also assembling it in to coherent cases. Commonly this will involve extracting cases from more ambiguous experiences, similar to the approach taken by Kendall-Morwick and Leake [10]. Thus, the retrieval knowledge of such a system would include not only traditional indexing strategies but also a view over the experiential knowledge of the system that elicits discrete case structures ready for reuse. In this way, the retrieval phase must be augmented to consider also "extraction" of cases from existing experiential knowledge. The extraction component of retrieval is tied to the specifics of the problem the system is attempting to solve, but this also means that the same experiential knowledge can be applied to more varied problems within a system utilizing rich retrieval knowledge that can provide multiple views over the experience base. The shared experience base distinguishes this approach from simply developing several small CBR systems for each sub-problem specification. Traditional case retrieval still falls within this interpretation of CBR in which we would consider extraction to be an identity function between experiences and cases.

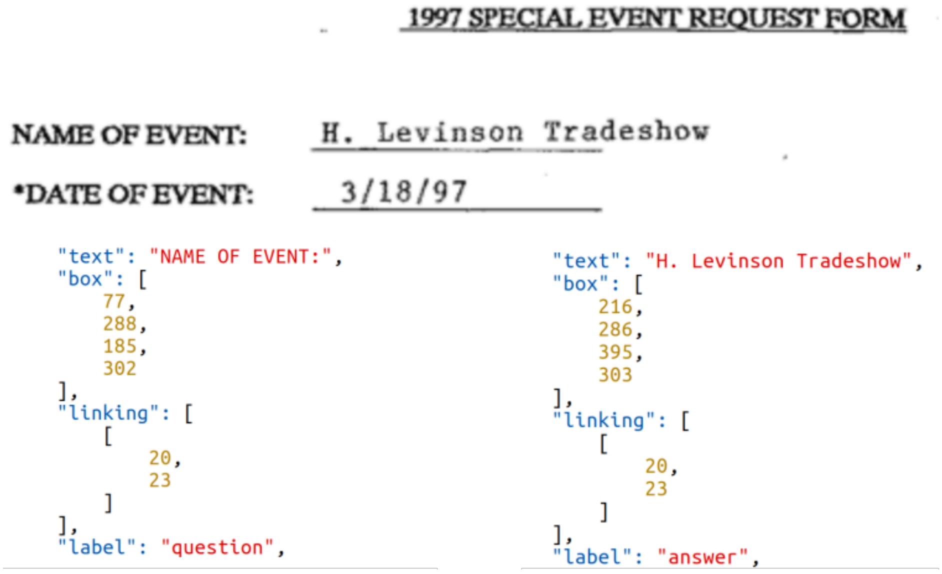
Retention is also complicated by this perspective. The case to be retained may not comprise a new experience in the experience base and may instead involve an alteration of existing experiences within this knowledge container. For example, a conceptually ambiguous case that subsumes the most recent reasoning episode (as well as several prior reasoning episodes) may be updated to better reflect a world in which the most recent case occurred (perhaps a concept in the case becomes further abstracted). As another example, a compositionally ambiguous case may be updated to include a portion or all of the most recent case. Or perhaps retention is deferred to a time in which multiple cases have been resolved and are ready to be retained collectively as a new experience in the experience base. This component of the retention phase is referred to as "generalization", and again, traditional case retention could fit this perspective by simply performing no generalization on revised cases.

5. Exploring Ambiguous Case Structure in Case-Based Form Understanding

This perspective on CBR is explored through early work to develop a case-based form understanding system. This system is being developed as part of a project to digitize four decades of physical documents used by the Computer Information Sciences department at Washburn University and organize the information contained in these documents in a database. Forms are the focus of this effort due to the structural nature of these documents that can be exploited by CBR techniques.

It's worth noting that a project of this nature undertaken by an organization such as the CIS department will benefit from the frequent occurrence of common document formats (for example, internal course enrollment permission forms), allowing for carefully developed domain-specific solutions and limiting the need for a sophisticated case-based approach. However, the department's records are primarily a test case for a project intended for personal use by the general public. Individuals in their own personal lives collect a large number of disparate forms containing important information for that individual that can be difficult to retrieve in situations where that information can be important (medical issues, tax audits, etc), making for a more compelling problem to which CBR can be brought to bear. Furthermore, unlike a computer science department, most individuals do not have access to professional programmers that can tailor software solutions to digitizing their personal files, thus highlighting the public benefit of such an open source software project.

This project is a work in progress. As of the time of this writing, scanning is currently ongoing and source code for the form understanding system is being developed. The abstract



1997 SPECIAL EVENT REQUEST FORM

NAME OF EVENT: H. Levinson Tradeshow

•DATE OF EVENT: 3/18/97

```
"text": "NAME OF EVENT:",
"box": [
    77,
    288,
    185,
    302
],
"linking": [
    [
        20,
        23
    ]
],
"label": "question",
```

```
"text": "H. Levinson Tradeshow",
"box": [
    216,
    286,
    395,
    303
],
"linking": [
    [
        20,
        23
    ]
],
"label": "answer",
```

Figure 2: Example of a partial scan and annotations from the FUNSD data set

framework introduced in this paper, however, has been implemented through the development of sub-problem specifications described in later subsections.

While document scanning at Washburn University is ongoing, initial work in this project is being performed with the FUNSD data set developed by Jaume et al. [14]. This data set contains 199 scanned documents containing form data from a variety of domains (ex. marketing, advertising, scientific). Each scanned document is paired with ground truth JSON encoded data containing a list of semantic entities (text and other data visually grouped together in the form). Entities are labeled as headers, questions, answers, or other depending on their role within the form. An example from this data set is pictured in Figure 2.

Jaume et al. additionally recognize three primary sub-tasks for which the data set provides training and testing data: text detection, text recognition, and form understanding, the last of which is split in to word grouping, semantic entity labeling, and entity linking. While various ML techniques are also applicable to these tasks, the opportunities for CBR, particularly utilizing the framework presented in this paper, are identified for two components of the form understanding problem along with two additional sub-tasks (document grouping and document classification).

Each sub-task will be performed either entirely manually (for evaluation purposes, the training data of the FUNSD data set is considered manually developed cases), interactively (the system provides a confidence score for a particular annotation and a user confirms or rejects it), or automatically (annotations with high enough confidence scores are automatically accepted). Results from reasoning episodes of each sub-task are retained within the experience base through an ambiguous experience record (an ordered collection of annotated scans, henceforth refereed to as a "document") and reused in potentially any of the other sub-task scenarios. Tasks can be performed in any order as the decisions recorded from any task can aid in the execution of any other task. The diversity of sub-tasks coupled with the universal reliance on the same experiential knowledge structure through each sub-task makes this form understanding domain a good candidate for the implementation of the framework proposed in this paper.

5.1. Document Reconstruction

Documents within the FUNSD data set do well to capture most of the types of noise one would encounter in a digitization project, however current progress through the Washburn CIS digitization project has uncovered an additional sources of noise: multi page forms that are split across several scans, forms that are duplicated in multiple scans, and multi-page scans that contain several different forms. Therefore "document reconstruction" was added as an additional task in which the page structure of the original documents is recovered from the scans of those documents.

For efficiency reasons, large stacks of physical documents are often scanned together in duplex. After scanning, all PDFs are split in to single page files (from this point forward a "scan" will refer to a one page file), but metadata conserving the order and grouping of the scans is retained. A problem solving episode consists of comparing two scans not known to be related and considering whether they are the same page, sequential pages in the same document, or unrelated.

Duplicate detection is fairly straightforward and can be performed successfully by comparing the raw data in the scans. However, determining whether two scans are part of the same physical document and the order they come in is more subtle and can benefit from the metadata recorded from scanning but also annotations retained from this and other form understanding sub-tasks described below. Cases are pairs of documents and are derived from the experience base by splitting existing documents in to two. Queries can be generated by considering any pair of existing documents in the experience base.

5.2. Text Detection, Recognition, and Grouping

The current plan is to implement these tasks with deep learning baselines evaluated through the FUNSD project (such as Tesseract or the Google Vision API). The baselines perform well for text detection and recognition. Word grouping involves determining what words recognized in the scan should be considered as part of a group. FUNSD baselines consider this a clustering problem but perform poorly against the ground truth assessment. Although there are not current plans for a case-based approach to this task, a method improving on the baselines by incorporating layout data and semantic annotations from other sub-tasks would be warranted in future work.

5.3. Semantic Entity Labeling and Linking

This task is to assign semantic labels to known groups of words (question, answer, etc) and determine which questions and answers are associated with each other. The first task is a straightforward classification task. Given correctly annotated word groupings, cases are derived from the experience base by extracting labeled word groupings along with contextual details of the containing document. Queries can be generated from any unlabeled word grouping in any document in the experience base.

Linking labeled semantic entities involves generating cases by extracting existing linked pairs of question-labeled and answer-labeled semantic entities as positive cases in a binary classification task. Unlinked pairs will not be considered negative cases since, initially, most needed labels will not be present. Confidence in linking two question and answer semantic entities as a query will be determined from the similarity of the retrieved cases.

5.4. Document Classification

Classifications for documents associate them with known form types to facilitate information extraction. Cases and queries for this task are straightforward: classified and unclassified documents. Once word groupings are semantically labeled and linked in an unclassified document, they can collectively be used as a basis of comparison to other classified documents for case retrieval and reuse.

6. Conclusions and Future Work

This paper has outlined the subtle difference between complexity and ambiguity in case representations and presented an abstract framework augmenting case representation, retrieval, and retention that is designed to take advantage of this ambiguity in problem domains involving complex and varied sub-tasks. The framework is explained through its relationship to prior work and its integration in an ongoing form understanding project. As this framework only involves a shift in how experiential knowledge is stored and retrieved, it is also compatible with most existing alternative frameworks to the traditional 4R CBR cycle that are popular in domains involving complex case representations. This framework is expected to enhance the flexibility of CBR systems working in these problem domains and provide inspiration for developers of full-featured frameworks used for design and other complex tasks.

As the form understanding project is ongoing, future work will involve an analysis of the results of the tasks described in section 5, comparing case-based approaches to machine learning baselines. Additional work on this project is expected to further automate the process of information extraction (for example, detecting hierarchical elements of forms and object detection). Other future work could include a wider literature review of ambiguity in case representations. As this framework is intended for CBR tasks with complex problem descriptions, additional comparison of the extraction phase to problem elaboration methods as well as conversational retrieval methods is warranted. Additionally, given that this framework involves frequent updates to existing experiential data, rather than simply the addition and/or deletion of cases, a comparison against studies of case base maintenance is important, as well as consideration of the implications of managing the provenance of experiential data under this perspective. Finally, although the form understanding project along with prior work in POCBR represents significant diversity of application of this framework, application to new domains (potentially creative applications, health CBR, or other design domains not mentioned) would be beneficial.

References

[1] V. Eisenstadt, K.-D. Althoff, Overview of 4r cbr cycle modifications (extended version), in: Proceedings of the Conference on "Lernen, Wissen, Daten, Analysen", Berlin, Germany, September 30 - October 2, 2019, volume 2454 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 230–240.

[2] S. A. Cook, The complexity of theorem-proving procedures, in: Proceedings of the Third Annual ACM Symposium on Theory of Computing, STOC '71, Association for Computing Machinery, New York, NY, USA, 1971, p. 151–158.

[3] V. Eisenstadt, C. Langenhan, K.-D. Althoff, Flea-cbr – a flexible alternative to the classic 4r cycle of case-based reasoning, in: Case-Based Reasoning Research and Development, Springer International Publishing, Cham, 2019, pp. 49–63.

[4] A. Cordier, E. Gaillard, E. Nauer, Man-machine collaboration to acquire cooking adaptation knowledge for the taaable case-based reasoning system, in: Proceedings of the 21st International Conference on World Wide Web, WWW '12 Companion, Association for Computing Machinery, New York, NY, USA, 2012, p. 1113–1120.

[5] P. Klein, L. Malburg, R. Bergmann, Learning workflow embeddings to improve the performance of similarity-based retrieval for process-oriented case-based reasoning, in: Case-Based Reasoning Research and Development, Springer International Publishing, Cham, 2019, pp. 188–203.

[6] R. Bergmann, J. Kolodner, E. Plaza, Representation in case-based reasoning, The Knowledge Engineering Review 20 (2005) 209–213.

[7] R. Bergmann, Y. Gil, Similarity assessment and efficient retrieval of semantic workflows, Information Systems 40 (2014) 115–127.

[8] K. Maximini, R. Maximini, R. Bergmann, An investigation of generalized cases, in: Case-Based Reasoning Research and Development, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 261–275.

[9] I. Bichindaritz, The case for case based learning, in: Case-Based Reasoning Research and Development, Springer International Publishing, Cham, 2018, pp. 45–61.

[10] J. Kendall-Morwick, D. Leake, Facilitating representation and retrieval of structured cases: Principles and toolkit, Information Systems 40 (2014) 106–114.

[11] S. Montani, M. Striani, S. Quaglini, A. Cavallini, G. Leonardi, Semantic trace comparison at multiple levels of abstraction, in: Case-Based Reasoning Research and Development, Springer International Publishing, Cham, 2017, pp. 212–226.

[12] C. Zeyen, L. Malburg, R. Bergmann, Adaptation of scientific workflows by means of process-oriented case-based reasoning, in: Case-Based Reasoning Research and Development, Springer International Publishing, Cham, 2019, pp. 388–403.

[13] O. Berg, P. Reuss, R. Stram, K.-D. Althoff, Comparing similarity learning with taxonomies and one-mode projection in context of the feature-tak framework, in: Case-Based Reasoning Research and Development, Springer International Publishing, Cham, 2019, pp. 1–16.

[14] G. Jaume, H. Kemal Ekenel, J.-P. Thiran, Funsd: A dataset for form understanding in noisy scanned documents, in: 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), volume 2, 2019, pp. 1–6.