TABLE 1: A summary of advantages and limitations of various deep learning-based table detection methods that are based on object detection frameworks.

Literature	Method	Highlights	Limitations
Gelani et al. [44]	Faster R-CNN (Section III-A1b). Images are transformed and then fed into the Faster R-CNN.	**a)** First deep learning based table detection approach on scanned document images, **b)** Transforming RGB pixels to distance metrics facilitates the object detection algorithm.	Extra pre-processing steps involved.
DeCNT [46]	Deformable convolutions implemented in the Faster R-CNN architecture (Section III-A1c).	The dynamic receptive field of deformable convolutional neural networks help in recognizing various tabular boundaries.	Deformable convolutions are computationally intensive as compared to traditional convolutions.
DeepDeSRT [45]	Faster R-CNN with transfer learning techniques (Section III-A1b)	Simple and effective end-to-end approach to detect tables and structures of the tables.	Not as accurate as compared to other states of the art approaches.
TableBank [62]	Faster R-CNN used as a baseline method for a novel dataset (Section III-A1b).	This approach presents that by leveraging a large dataset such as TableBank, a simple Faster R-CNN can produce impressive results.	Just a direct application of Faster R-CNN.
Sun et al. [63]	Faster R-CNN with locating corners (Section III-A1b).	**a)** Faster R-CNN is exploited to detect not only tables but the corners of the tabular boundaries as well, **b)** Novel method produces better results.	**a)** Computationally more extensive because of additional detections, **b)** Post-processing steps such as corners' refinement are required.
Huang et al. [47]	YOLO based table detection method (Section III-A1d).	Comparatively, faster and efficient approach.	The proposed method depends on the data driven post-processing techniques.
García et al. [72]	Employed Mask R-CNN, YOLO, SSD and RetinaNet to compare fine-tuning techniques (Section III-A1e).	Presented the benefits of leveraging a closer domain fine-tuning methods for table detection while employing object detection networks.	Still, closed domain fine-tuning is not enough to reach the state-of-the-art results.
CascadeTabNet [48]	Employed Cascade Mask R-CNN with an iterative transfer learning approach (Section III-A1f).	This work presents that transformed images with an iterative transfer learning can reduce the dependency of large-scale datasets.	Similar to [44], extra pre-processing steps are involved in this approach.
CDeC-Net [49]	Cascade Mask R-CNN with a deformable composite backbone (Section III-A1c).	**a)** Extensive evaluations on publicly available benchmark datasets for table detection. **b)** An end-to-end object detection-based framework leveraging composite backbone to produce state-of-the-art results.	Along with the deformable convolutions, a composite backbone is employed which makes the approach computationally intensive.
GTE [52]	Proposed a generic object detection approach (Section III-A1f).	**a)** An end-to-end technique that can operate on any object detection framework. **b)** This work proposed an additional piece-wise constraint loss that benefits the task of table detection.	Since the task of table detection is dependent on cell detections, annotations for cellular boundaries are required.

els like Mask R-CNN [73] or U-net [91]. After combining the GAN-based feature generator with Mask R-CNN, the approach is evaluated on the ICDAR2017 POD dataset [64]. Authors claim that this approach will facilitate other object detection and segmentation problems.

B. TABLE STRUCTURAL SEGMENTATION

Once, the boundary of the table is detected, the next step is to methodologies according to the architecture of deep neural networks. Table 3 summarizes these approaches by highlighting their advantages and limitations. Figure 6 illustrates the essential flow of table structural segmentation techniques that are discussed in this review paper.

1) Semantic Image Segmentation

Along with table detection, *TableNet* segments the structure

A Description Logic of Change*

Alessandro Artale University of Bolzano, Italy artale@inf.unibz.it
Carsten Lutz University of Dresden, Germany clu@tcs.inf.tu-dresden.de
David Toman University of Waterloo, Canada david@uwaterloo.ca

Abstract

We combine the modal logic S5 with the description logic (DL) $\mathcal{ALCQI}$. The resulting multi-dimensional DL $\mathsf{S5}_{\mathcal{ALCQI}}$ supports reasoning about change by allowing to express that concepts and roles change over time. It cannot, however, discriminate between changes in the past and in the future. Our main technical result is that satisfiability of $\mathsf{S5}_{\mathcal{ALCQI}}$ concepts with respect to general TBoxes (including GCIs) is decidable and 2-EXPTIME-complete. In contrast, reasoning in temporal DLs that *are* able to discriminate between past and future is inherently undecidable. We argue that our logic is sufficient for reasoning about temporal conceptual models with time-stamping constraints.

1 Introduction

An important application of Temporal Description Logics (TDLs) is the representation of and reasoning about temporal conceptual models [Artale, 2004; Artale *et al.*, 2003; 2002]. Knowledge captured by such models is translated into a TDL TBox and reasoning algorithms for TDL are then used to detect inconsistencies and implicit IS-A relations in the temporal model [Artale and Franconi, 1999; Artale *et al.*, 2002; Calvanese *et al.*, 1998]. A serious obstacle for putting this general idea to work is posed by the fact that for many natural temporal conceptual formalisms and their associated TDLs, reasoning turns out to be undecidable.

The most prominent witnesses of this problem are the various temporal entity-relationship (TER) models used to design temporal databases [Chomicki and Toman, 2005]. TERs are classical ER data models extended with two additional classes of constraints that model the temporal evolution of data in an application domain [Spaccapietra *et al.*, 1998]. First, *timestamping constraints* are used to distinguish temporal and atemporal components of a TER model. Timestamping is usually implemented by marking entities (i.e., classes), relationships and attributes as *snapshot*, *temporary*, or *unrestricted*. The idea behind such a classification is to express that object membership in entities, relationships, and attribute values cannot or must change in time; this is achieved by snapshot and temporary marks in the diagram, respectively. Second, *evolution constraints* govern object migration between entities and can state, for example, that every instance of the entity Child will eventually become an instance of the entity Adult.

TER models with both timestamping and evolution constraints can be translated into the TDL $\mathsf{DLR}_{\mathcal{US}}$ [Artale *et al.*, 2002]. Unfortunately, reasoning in this logic is undecidable. Moreover, the computational problems are not due to the translation to TDLs: even direct reasoning in the generally less powerful TER models is undecidable [Artale, 2004]. There are two principal ways around this problem. The first approach restricts the application of timestamping: it allows arbitrary timestamping of entities, but gives up timestamping of relationships and attributes (i.e., all relationships and attributes are unrestricted). This re-establishes decidability of TER models with restricted timestamping and evolution constraints [Artale *et al.*, 2002]. The second approach to regaining decidability allows for full use of timestamping, but prohibits the use of evolution constraints.

This second alternative is pursued in the current paper. We devise a multi-dimensional description logic $\mathsf{S5}_{\mathcal{ALCQI}}$ that is obtained by combining the modal logic S5 with the standard DL $\mathcal{ALCQI}$. The S5 modality can be applied to both concepts and roles; axioms in the TBox are, however, interpreted globally. This logic can be viewed as a *description logic of change*: it can express that concept and role memberships change in time, but does not permit discriminating between changes in the past or future. We show that TER models with full timestamping (i.e., timestamping on entities, relationships, and attributes) but without evolution constraints can be captured by $\mathsf{S5}_{\mathcal{ALCQI}}$ TBoxes.

The main contribution of this paper is to show that reasoning in $\mathsf{S5}_{\mathcal{ALCQI}}$ is decidable. We also pinpoint the exact computational complexity by showing 2-EXPTIME completeness. Thus, adding the S5 *change modality* pushes the complexity of $\mathcal{ALCQI}$, which is EXPTIME-complete, by one exponential. Our upper bound can be viewed as an extension of the decidability result for a simpler multi-dimensional DL, $\mathsf{S5}_{\mathcal{ALC}}$, [Gabbay *et al.*, 2003] which is not capable of capturing TER models. However, we had to develop completely new proof techniques as the decidability proof for $\mathsf{S5}_{\mathcal{ALC}}$ re-

*The research was supported by EU IST-2005-7603 FET Project TONES and by NSERC.

TABLE 6: Table Detection Performance Comparison. The double horizontal line partitions the results obtained on various datasets. Outstanding results in all the respective datasets are highlighted. For the ICDAR-2019 dataset [80], all of the three approaches are not directly comparable to each other because they report F-Measure on different IOU thresholds. Hence, results on ICDAR-2019 dataset are not highlighted.

Literature	Year	Dataset	IOU	Precision	Recall	F-Measure	Method
Gelani et al. [44]	2017	UNLV	0.9	**82.3**	**90.6**	**86.3**	Faster R-CNN (Section III-A1b)
García et al. [72]	2019	UNLV	0.9	48.0	49.0	49.0	YOLO (Section III-A1d)
DeCNT [46]	2018	UNLV	0.5	78.6	74.9	76.7	Deformable Convolutions (Section III-A1c)
DeepDeSRT [45]	2017	ICDAR-2013	0.5	97.4	96.1	96.7	Faster R-CNN (Section III-A1b)
Kavasidis et al. [81]	2018	ICDAR-2013	0.5	97.5	98.1	97.8	Semantic Image Segmentation (Section III-A2)
DeCNT [46]	2018	ICDAR-2013	0.5	99.6	99.6	99.6	Deformable Convolutions (Section III-A1c)
Huang et al. [47]	2015	ICDAR-2013	0.5	100	94.9	97.3	YOLO (Section III-A1d)
TableBank [62]	2019	ICDAR-2013	0.5	96.2	96.2	96.2	Faster R-CNN (Section III-A1b)
TableNet [84]	2019	ICDAR-2013	0.5	96.3	96.9	96.6	Fully Convolutional Networks (Section III-A2a)
CascadeTabNet [48]	2020	ICDAR-2013	0.5	**100**	**100**	**100**	Cascade Mask R-CNN (Section III-A1f)
GTE [52]	2021	ICDAR-2013	0.5	-	-	95.7	Object Detection (Section III-A1f)
CDeC-Net [49]	2020	ICDAR-2013	0.5	94.2	99.3	96.8	Cascade Mask R-CNN (Section III-A1f)
García et al. [72]	2019	ICDAR-2013	0.6	70.0	97.0	81.0	Mask R-CNN (Section III-A1e)
Li et al. [88]	2019	ICDAR-2017	0.6	94.4	94.4	94.4	GANs (Section III-A4)
DeCNT [46]	2018	ICDAR-2017	0.6	97.1	96.5	96.8	Deformable Convolutions (Section III-A1c)
Huang et al. [47]	2015	ICDAR-2017	0.6	**97.8**	**97.2**	**97.5**	YOLO (Section III-A1d)
Sun et al. [63]	2019	ICDAR-2017	0.6	94.3	95.6	94.5	Faster R-CNN (Section III-A1b)
García et al. [72]	2019	ICDAR-2017	0.6	92.0	87.0	89.0	Retina Net (Section III-A1e)
CDeC-Net [49]	2020	ICDAR-2017	0.6	89.9	96.9	93.4	Cascade Mask R-CNN (Section III-A1f)
CascadeTabNet [48]	2020	ICDAR-2019	0.6	-	-	94.3	Cascade Mask R-CNN (Section III-A1f)
CDeC-Net [49]	2020	ICDAR-2019	0.6	93.9	98.0	95.9	Cascade Mask R-CNN (Section III-A1f)
GTE [52]	2021	ICDAR-2019	0.8	96.0	95.0	95.5	Object Detection (Section III-A1f)
Riba et al. [87]	2019	RVL-CDIP	0.5	15.2	36.5	21.5	Graph Neural Network (Section III-A3)