CADE: The Missing Benchmark in Evaluating Dataset Requirements of AI-enabled Software

Hamed Barzamini and Mona Rahimi
Department of Computer Science
Northern Illinois University
DeKalb, IL
hbarzamini,mrahimi1@niu.edu

2022 IEEE 30th International Requirements Engineering Conference (RE) | 978-1-6654-7000-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/RE54965.2022.00013

Abstract—The inductive nature of artificial neural models makes dataset quality a key factor of their proper functionality. For this reason, multiple research studies proposed metrics to assess the quality of the models' datasets, such as dataset correctness, completeness, and consistency. However, these studies commonly lack a point of reference against which the proposed quality metrics could be assessed.

To this end, this paper proposes a generic process that extracts the necessary knowledge to build a reliable reference point for the purpose of explanation, assessment, and augmentation of the AI-software dataset. This process automatically builds a benchmark specific to the software operational domain, interprets the training and validation datasets of AI-enabled perception software systems, and evaluates the dataset semantic quality and completeness relative to the benchmark. We implemented this process within a framework called *Concept Augmentation and Dataset Evaluation (CADE)*, which leverages a series of novel natural language and image processing techniques to construct a semantic benchmark with respect to the domain specifications.

The application of CADE to three commonly-used autonomous driving datasets showed several common weaknesses present in the arbitrarily-collected datasets against the encoded domain specifications, demonstrating dataset divergence from the domain concepts and under-represented variances of the concepts in the data. The qualitative evaluation results showed an average of about 75% relevancy of CADE generated topics.

I. Introduction

William Deming (1900-1993), the leading management thinker in the field of quality, has a well-known saying, adopted by software engineers to emphasize the importance of following a systematic development process. The phrase states *"if you can not describe what you are doing as a process, you do not know what you are doing."* In the past decades, this phrase has been referred to in different phases of software development process, including requirements engineering [27]. With rapid prevalence of AI-enabled perception software systems (AIS) in a wide range of applications, this statement is barely applicable to the intelligent softwares. For instance, for developing software with visual perception capabilities data-driven neural models are often deployed. These models are known for behaving as black boxes since the intuitive process behind their predictions is not fully describable for the end-users.

Along with the application of data-driven AI and ML models in software, the domain of requirements engineering is altering as well. For instance, with the application of neural models to the visual perception tasks in a majority of autonomous vehicles domain, requirements specifications are gradually transforming from being explicitly articulated in the textual format or formal rules into being implicit within a set of training data, such as images and video frames.

During the conventional development processes, the software implements a set of pre-defined "agreed-upon" requirements specifications, gathered from stakeholders, domain experts, and customers [25], while software enabled with deep learning (DL) models instead learns and suggests the specifications from data. The model's learning ability is beneficial in developing perception capabilities for software, operating in domains containing concepts which are difficult to describe and therefore, are hard to program [28]. For instance, what is the exact specification for recognizing a cancerous tumor in computed tomography (CT) images in the medical domain?

The description of malignant lung tumors requires to cover the characteristics of any instance of cancerous tumors, but also exclude the characteristics of all cases of benign tumors. Defining such concepts whose instances vary over a wide range of features and values is a non-trivial task for humans, if not impossible. Therefore, the programmers lack sufficient instructions to program a diagnosis application for the domain concepts, which are difficult to describe, while several software applications are developed for diagnosis of malignant tumors, adopting DL models. The predictive models inductively learn the characteristics of both types of tumors from a large set of collected images, and further generalize their data-driven knowledge to unseen CT images of new patients. We refer to domain concepts which are challenging to specify due to their various instances with characteristics that are hard to predict (e.g., tumors differ from each other in terms of shape, size, and density) as *hard-to-specify* domain concepts, for which deriving a generic definition to include the entire vastly-deviated instances is difficult.

In the context of visual perception, we propose that dataset requirements relate to the specifications of hard-to-specify domain concepts, and therefore, need to be formulated with respect to these specifications. We believe here, the problem of dataset requirements and the problem of specifying hard-to-specify domain concepts both refer to the same concern. In order to enable AIS to visually perceive varying instances of a domain concept, the primary requirement is the quality

978-1-6654-7000-1/22/$31.00 ©2022 IEEE
DOI 10.1109/RE54965.2022.00013

of a dataset in representing the concept. While a high-quality dataset is not the only requirement for AIS visual perception, it surely is the necessary requirement. For instance, in the case of training a data-driven model, to recognize the malignant tumors, the necessary requirement is the dataset capability to provide a diverse, accurate, consistent and complete instances of both types tumors for the model.

In this regard, this paper aims to evaluate a dataset through identifying the gap between specification of *hard-to-specify* domain concepts and their visualization in the dataset. For this a generic process is proposed and is further implemented in a framework, called CADE. Adopting NLP algorithms and image processing methods, CADE systematically construct a benchmark for the specification of *hard-to-specify* domain concepts. This benchmark plays the role of a reference point for assessing and certifying the semantic completeness of a dataset with respect to one or multiple targeted concepts. Throughout this paper, we refer to a hard-to-specify domain concept whose recognition is an AIS objective as a *targeted concept* (e.g., tumor). Later, with a reference to the established benchmark, CADE identifies the missing and under-visualized dimensions of the concept in the dataset. Figure 1 represents a high-level design of CADE process.

Identifying the primary features of domain concepts, CADE additionally provides a map and guidance for next data collection processes. Moreover CADE contributes to the area of *explainable AI* by providing insight on the contained variations of a domain concept in a dataset.

Research Questions: We phrase our research questions as:

1) RQ1: In AIS perception tasks, how can we address the problem of specifying dataset requirements?
2) RQ2: Based on results from RQ1, how can we verify a dataset against the specifications?

Contributions: This paper's contributions include:

- Demonstrating that RE domain analysis can be adopted, adapted, and applied to the process of AIS engineering;
- Demonstrating that leveraging domain specifications, the quality of AIS datasets can be evaluated and analyzed;
- Implementing novel methods for deriving partial specifications to assess, improve, and represent the semantic completeness of AIS dataset.

All artifacts of this work are made public online[1].

II. The Problem Domain

This section explains the nature of the problem, as well as the primary root causes of the problem from our point of view.

A. Dataset Limitations

Nevertheless, the emergence of data requirements has posed new issues for the RE community. Sandkuhl [52] argued while data needed for AI projects are accessible from several companies, the available data lacks the structure and rules necessary to implement and train AIS. Whereas the existing requirements only address the data sampling rate and only

[1]https://anonymous.4open.science/r/CADE-9BE6/

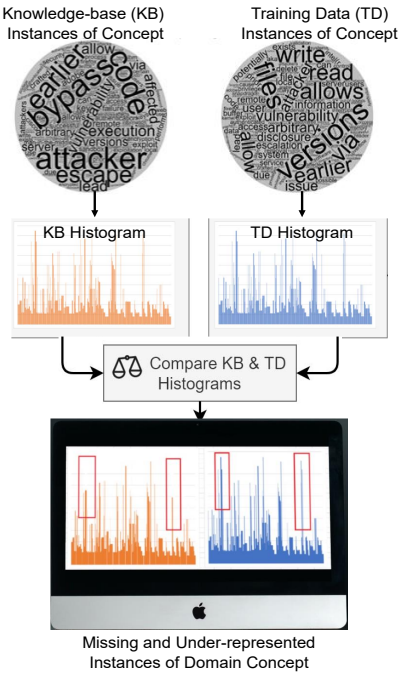

Fig. 1. The identification of dataset weakness points.

provide non-measurable characteristics of data quality [4]. Shin et al. argue that more samples in a dataset give a more diverse representation of domain [53], while ignoring the major problem of overfitting the models. Overfitting is a fundamental issue in supervised machine learning which prevents the model from being generalizable to unseen data during the operation [67]. The overfitted models, while performing well on the datasets they have been trained on, perform poorly on unseen data [19]. This is because the model is excessively tailored for *some* target dataset (training data), whose samples distribution is significantly different from the distribution of actual data during the operation. While most of the existing studies emphasize the importance of data requirements for AIS, there is limited empirical research available in this area.

The training datasets are collected in unsystematic and arbitrary manners. Hence, datasets used to train ML models are generally limited in the number and diversity of samples they comprise [24], [48]. For instance, the most recently established datasets in the context of autonomous driving, such as Caltech [20], are collected by a vehicle-mounted camera aimlessly navigating rural roads [24]. Unsystematic and unguided collection of datasets may result in an incomplete, unrepresentative, and undiversified dataset, leading to biased models. For instance, one prior research performed a simple cross-dataset evaluation to reveal that the majority of the state-of-the-art pedestrian detectors are biased and, therefore, are vulnerable to small domain shifts [24]. This is because the arbitrarily-collected datasets may not be a complete or fair representation of the actual concepts in the operational domain [9]. Without a high-quality and comprehensive dataset that includes a wide variety of the concept's instances, there will be a significant and inevitable misalignment between the specification of a domain's concept and what a collected

dataset represents as the targeted concept [28], [51].

B. The Necessity of Dataset Requirements

As discussed in Section I, the data-driven characteristic of AI and ML is particularly desirable for programming hard-to-specify concepts for which limited description exists to guide software programmers. However, the inductive nature of AI and ML makes the software's functionality highly reliant on the quality of the model's training dataset.

For this data-centered characteristic of AI, the primary requirement for AIS, as discussed in the literature, is to ensure the quality of training data is appropriate [4], [52], [60]. One study argues that the practices in RE mainly focus on requirements that are user-centric and do not pay enough attention to data requirements [5].

C. Reference Point of Dataset Requirements

There are a few studies in software engineering literature which focus on AIS dataset requirements. Among them a few propose several quality metrics as AIS dataset requirement. For instance, Challa et al. [13] list five essential characteristics contributing to the quality of data as *accuracy*, *completeness*, *consistency*, *credibility* and *correctness*. Vogelsang et al. argue that specifying data requirements includes information about the necessary *quantity* and *quality* of data [60]. Another work by A. Banks and R. Ashmore includes nine dataset quality metric including data *sufficiency*, self-*consistency* and *absence of bias*, to build confidence in training data [8]. Another research, identifies *representativeness*, *balancedness*, *timeliness* as the critical quality concerns in dataset [41]

The attention to dataset requirements is not limited to the software engineering domain. For instance, in the medical domain, Marc D. Kohli et al. proposed the specifications should be clearly defined for medical image data, covering all aspects of image management, including image metadata, pixel data, post-processing techniques, and image cataloging [30].

The common objective here is to improve the dataset's quality to lay a better foundation for training the AI model and ultimately improve the quality of the emerging AIS. However, the aforementioned works lack to provide a standard or point of reference against which the datasets can be evaluated. A research principle is to evaluate the outcomes and express preferences relative to an existing reference point. For example, for the automated pedestrian collision avoidance system, what are the specific criteria of data sufficiency for recognizing the concept "pedestrian"? The dataset characteristics, such as correctness, completeness, and quantity, should be assessed with respect to which benchmark? How much data is enough?

Without a reliable point of reference for assessment, the quality metrics sound rather abstract since once achieved, yet we will not know that the criteria are met. Several studies released large datasets as a benchmark against which the other collected data samples could be compared. However, the to-be-benchmark dataset themselves are arbitrarily collected without a systematic collection process or based on a reliable source according to which the comprehensiveness of the dataset could be measured. Furthermore, as discussed earlier, size is not a criterion of a dataset quality since overfitting in data-driven modeling methods is a common problem [67]. Overfitting the training data leads to the deterioration of the generalization properties of the model and results in its untrustworthy performance when deployed in the real world.

III. The Proposed Solution

In this section, we share the motivation and domain of our interest in this problem, discuss the adequacy of the solution we propose, and a generic process for the proposed solution in the form of a pseudo code.

A. Reference Point in Visual-Perception Tasks

The autonomous vehicles domain contain methods for acquiring, processing, analyzing, and understanding images and, in general, high dimensional data from the real world to produce numerical or symbolic information in the form of decisions. In this context, data-driven models are majorly adopted for visual perception of these systems' surroundings. Visual tasks are trained on the image and video frame datasets of the AIS targeted concept. For instance, a pedestrian detector is trained based on image and video dataset of numerous different-looking instances of pedestrians. An obstacle detector is trained on visual data of varying-looking obstacles.

However, depending on the context, the specification of targeted concepts may differ from domain to domain. For instance, potential obstacles of an autonomous boat contain a large number of static and dynamic obstacles such as shore, piers, boats, swimmers, debris, and buoys [71], while on-road obstacles of autonomous vehicles include pedestrians, cars and animals. Hence, the visual perception tasks are often domain-specific, and therefore neural models in each domain are particularly trained for the targeted concepts in that domain. Therefore, requirement specifications in perception tasks mirror the perception of the specifications of targeted concepts in a particular domain. For instance, the primary requirement of a pedestrian detector is to detect pedestrians with a minimum number of failures.

Specifying requirements for perception tasks is straightforward: to detect a *targeted concept* as accurately as possible. However, the problem here is demonstrated in the ambiguous term of targeted concept. Requirements specifications in AIS perception start with our very limited understanding of how hard-to-specify concepts should be defined even at the high level and how they should be represented in the training data. As such in perception tasks, the primary dataset requirement is to represent a comprehensive specification of hard-to-specify domain concepts properly since the dataset reflects the AIS domain concepts. To address the problem of dataset requirements, we need to address the problem of specifying hard-to-specify targeted domain concepts.

B. Domain Specification as a Reference Point

In conventional RE, to define requirements, the engineers are required to analyze the domain specifications to extract

and classify domain properties. In general, terms found in user scenarios, such as domain elements (e.g., account record), are organized in a central domain specification knowledge base to be re-used in the future implementation of other applications in the same or similar domain. For instance, in the medical domain, to develop a domain-specific software, such as a pacemaker application, typically first the user manual, description of functionalities, implantation and configuration instructions, and hazards documents are either automatically or manually parsed to extract the necessary domain information about pacemaker requirements. Once domain specifications become available, the requirements engineers typically continue creating a set of user scenarios and elaborating fully-fledged requirements specifications, based on which the software is later designed and implemented.

Domain knowledge is typically collected ad hoc and evolves over time until enough experience is accumulated that generic abstractions can be isolated and reused. In scenarios which the to-be-developed application is not the first attempt, the specifications of domain concepts are often available. For instance, concepts such as "account record" are possibly specified in domain documents available to developers to develop a banking application. In domains for which domain documents are not available, common practices still directly make such reuse, often through reusing terminologies from publicly available lexical databases, such as WordNet [40]. This, for instance, would involve adding concepts like "account record" to the specific application domain, either manually or through automated links. While the retrieval of domain specifications is not new in RE, the importance and application of domain specifications are neglected in RE of AIS.

Concerning the visual perception tasks, we propose to extract the specifications of a targeted domain and then assess the quality of a randomly-collected training dataset. In the following sections, we propose and implement a generic process for this purpose. The ultimate goal here is to make AIS better meet domain specifications by the automated creation of domain-specific benchmarks for AIS to be referred to evaluate their dataset requirements. In short, we tend to improve the inductive nature of AIS with domain analysis. The improvement occurs through incorporating domain knowledge into AIS training datasets, which in turn compensates for the missing variants of a targeted concept within the dataset, providing an augmented source of knowledge for neural models.

IV. The Proposed Process

Domain specifications are typically retrieved during domain analysis, identifying the objects, operators, and relationships between what domain experts perceive to be important about a domain [42]. The experts' domain knowledge often becomes accessible to developers by analyzing the domain-specific documents. Several prior works have successfully extracted knowledge from the existing domain documents, either automatically or semi-automatically, for a large variety of domain applications [15], [23], [59]. Some studies, in addition, represented and stored the retrieved domain knowledge in the form of a semantic web (e.g., ontology) to formally capture metadata of the gathered knowledge about the domain and also to incrementally improve the web and re-use the information [18], [35]. Regardless of the presentation format, the objective is to remove any potential ambiguity from the terms specific to the domain. For instance, in a card-playing domain, such as poker applications, the domain specification models the *playing card* meaning of the word. In contrast, another specification in the computer domain may model the card as the *hardware memory or video card* meanings. These embedded semantics offer significant advantages, such as reasoning over data, operating with heterogeneous data sources, and dissolving ambiguities in the requirements. For instance, common metadata vocabularies (i.e., ontology) describe concepts as relationships between entities and categories of things. Each domain ontology typically models domain-specific definitions of terms. For example, the word *card* contains different meanings.

As discussed, hard-to-specify concepts refer to a domain element whose features and characteristics often vary from one instance to another, making it difficult to specify the concept for the software [50]. As such, no definite concept description exists to be passed to developers for implementation purposes, yet the software is expected to recognize varying instances of the concept during operation. To specify and verify dataset requirements with respect to variations of domain concepts, we propose the pseudocode below:

Algorithm 1 Concept Augmentation & Dataset Evaluation (CADE)

Require: C, the domain concepts to be evaluated.

$C \leftarrow c$. Domain Concept
$D \leftarrow d$. Data set
for each $c \in C$ do
  $F_c \leftarrow$Augment (c)
  $F_d \leftarrow$ Interpret (d)
end for
$report \leftarrow$ Evaluate (F_d, F_c)
return $report$
function AUGMENT($Concept\ c$)
  return F_c; . Primary Features
end function
function INTERPRET($Dataset\ d$)
  return F_d; . Primary Features
end function
function EVALUATE($array_1, array_2$)
  return $array_1 - array_2$;
end function

The algorithm receives hard-to-specify concepts common to a domain ($c \in C$), specifies each concept, and returns an assessment report for a given data set ($d \in D$).
For each concept, c, the auxiliary function, $Augment(c)$, will select and abstract common features which characterize variances of the input concept, F_c. Later, the $Interpret(d)$ function will select and abstract common features that characterize variances of the same concept in d a collected data set, F_d.

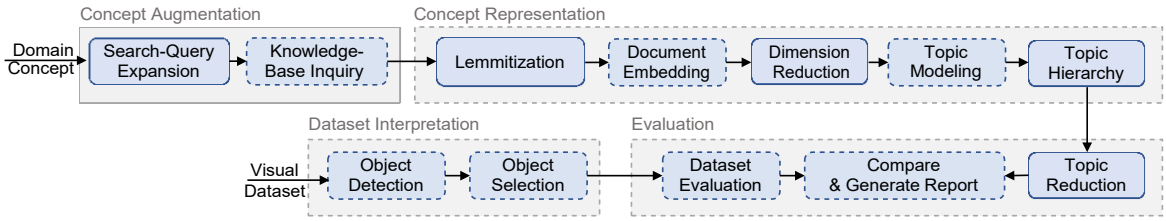

Fig. 2. The Automated Framework for Concept Augmentation and Dataset Evaluation (CADE).

The algorithm further, calls the $Evaluate(array_1, array_2)$ to evaluate the quality and quantity of data set representation of c with reference to F_c, $F_d - F_c$. We have created procedures to automate the above-mentioned operations.

V. The Functions Definition

This section provides a more detailed description of $Augment(concept\ c)$, $Interpret(dataset\ d)$ and $Evaluate(arrays\ arr1, arr2)$ functions, as well as possible implementations for each function.

A. Function $Augment(Concept) \rightarrow F_c$

Hard-to-specify concepts are inherently difficult to delineate, yet most humans have an intuition of what they refer to. In fact, their indescribable nature is the primary reason to adopt AI for their specification in the first place.

The initial challenge in gathering domain knowledge for the hard-to-specify concepts is the lack of knowledge sources. For instance, *pedestrian* is a socially constructed concept for which no relatively complete domain document exists. Although there are a few general domain semantic webs that include a limited specification of the term pedestrian, such as WordNet [40], they fail to adequately capture all varying instances of the concept in sufficient detail. For example, WordNet defines a pedestrian as a "person who travels by foot" and associates the word with the terms *walker* and *footer*. However, this definition is limited given that it excludes, for example, pedestrians riding a bike, roller-skating, or using a wheelchair. It also fails to describe a pedestrian's appearance in terms of attributes, such as clothing and posture.

To tackle the limited documents challenge for hard-to-specify domain concepts, we propose to extract domain knowledge from the online knowledge base, such as online books, articles, encyclopedia, dictionaries, semantic webs, legal documents, social media, news feeds, image and video repository. Due to different artifact types in the repositories, multiple processing methods can be applied. Regardless of the processing methods, the aim is to acquire domain knowledge through identifying a set of primary attributes associated with various instances of a concept as they appear in the knowledge-base. As such, we aim to define these concepts based on their sparse instantiating across a variety of sources.

In cases that domain knowledge is stored in a textual format, a wide range of Natural Language Processing (NLP) techniques [11], [14], [63], such as topic modeling [63], [69] and latent Dirichlet allocation (LDA) [11], [69], can be adapted to automatically mine textual sources of knowledge for *important* accompanying features of the concept. The importance of the features can be determined based on semantic, lexical, or syntactic characteristics, such as cosine similarity [47], frequency of co-occurrence [16], and grammatical importance [61], or based on a combination of multiple metrics.

Further to process visual sources of domain knowledge, such as available video and image sets, a variety of im-age processing techniques and convolutional neural networks (CNNs) [44], [56], [66], are available to be adopted. For instance, scene graph generation (SGG) techniques [56], [65], [68] translate the pixel-level visual data, such as video and image information, to natural language.

Later, to structure and re-use the extracted information, a series of machine learning (ML) techniques, such as classification and clustering [29], [62], can be applied to organize the information in a more readable format meaningfully.

B. Function $Interpret(Dataset) \rightarrow F_d$

In the computer vision domain, a wide range of Convolutional Neural Network models, such as Convolutional Neural Networks (R-CNNs) and Faster R-CNN [49], [49] are proposed and developed for visual perception tasks, such as object detection [56], [57], [65], [68]. A majority of these networks are previously trained on large-size data sets and can recognize various generic objects with high accuracy.

Yet, a pre-trained model can be re-trained for a specific domain to learn the specific hard-to-specify domain concepts and improve its perception of varying instances of the concept. For instance, several pre-trained models are particularly trained for pedestrian detection, car detection, and obstacle detection in a particular domain as an obstacle is described differently in the automotive domain than the naval domain. The application of the models on a data set will provide a label and a short description of the image scenes in natural language [33].

Training any similar model with the domain-specific concepts, the model is then applicable to a collected dataset to describe the images in a natural language containing the domain-specific terms (labels). This function also contributes to the explainable AI (XAI), understanding and interpreting the predictions made by the models [7].

C. Function $Evaluate(F_d, F_c) \rightarrow report$

The primary goal of the two previous functions of $Augment(Concept)$ and $Interpret(Dataset)$ is to facilitate mapping the artifacts of different natures to each other. Given the converted artifacts into the same human-understandable type (i.e., natural language), the presence of each defined variance of the concept in F_c is then searched within the data set descriptions of the same nature, F_d. However, a pragmatic

TABLE I
TOP FIVE TOPICS OF DOMAIN CONCEPTS *car* AND *pedestrian*.

Id	Car Topic	Top Term Examples
351	car manufacturer	volkswagen, jaguar, beetle, mini, bmw, vehicle
131	automobile	ford, chrysler, motor, automotive, engine, chevrolet
805	subcompact cars	subcompact, chevrolet, ford,volkswagen, sedan, toyota, mercedesbenz
67	v8 cars	buick, v8, chevrolet, oldsmobile, sedan, coupe, v6
793	automobile company	vehicle, motor, brand, guangzhou,company, truck, factory
Id	**Pedestrian Topic**	**Top Term Examples**
196	walking areas	pedestrian, crossing, crosswalk, traffic, signal, stripe
412	walkability	walk, walking, walkable,pedestrian, arrested
172	traffic	traffic, road, vehicles, streets, transportation, cars, lanes, pedestrian
363	victim	victim, stalking, bystander, intervene, witnesses, assault
198	parking	parking, garages, vehicles, cars, parkjockey, towing, park

solution is to apply the existing automated traceability methods [43] to trace the derived features of the concept to the data set labels. This verification process, in turn, characterizes the extent to which a data set contains or lacks features that are important to learning the concept.

This process enables a systematic and semantic-based assessment of a visual dataset according to the domain specifications. In addition, the same process is applicable to generate an initial map and guidance for data set collection in contrast to ad hoc collection manners. This process will improve the dataset quality and thus create a more representative data set, which trains a more reliable AIS.

VI. EVALUATION

To implement the proposed process, we developed an automated framework for *Concept Augmentation and Dataset Evaluation (CADE)*. Figure 2 represents the primary phases of CADE process. This framework receives any number of domain-specific terms as input, automatically creates and visualizes domain benchmarks specific to the visual perception tasks, then evaluates a given visual dataset relative to the benchmarks, and finally generates a report of the dataset weaknesses for future improvements.

A. Concept Augmentation

In the initial step of the process, a large set of knowledge-base is searched for any term contextually related to the input, creating an initial domain-specific search query. For example, recognizing a person as opposed to a non-person in autonomous vehicles domain is a common use case for perceiving hard-to-specify domain concepts. Hence, we selected *cars* and *pedestrians* as two contextually related concepts to be augmented by CADE automated process in this domain. Each process is computationally expensive since a large set of knowledge bases, such as Google n-gram [17] Onelook [2] and, RelatedWords [3] are thoroughly searched for each query.

Google n-gram is an online search engine that provides a search for 155 billion words from American English and 34 billion words from British English and provides high-frequency terms associated with a given term as a search query. The RelatedWords is an open-source project that runs several algorithms, such as word embedding, to convert words into multidimensional real-valued vectors representing their meanings. The generated vectors of the words are then mapped in a space of pre-computed vectors according to a set of existing corpora. The similarity of the vectors is then specified according to their distance in the space. RelatedWords also uses ConceptNet [55] to retrieve words that have meaningful relationships to our query. Onelook indexes over a thousand online dictionaries and encyclopedias to return the words related to a search query. In addition to dictionaries and encyclopedias, Onelook internally works on Datamuse API to search various data sources.

We implemented a two-phase process to retrieve the most related terms to our initial seeds. First, the Google n-gram knowledge base is searched for accompanying and co-occurring terms with each concept. The database will return all terms that more frequently occurred within a given short distance (up to four terms before and after) of the initial term. Yet to identify the related terms that did not appear within our identified range, the RelatedWords and OneLook are searched for semantically related terms to the input. This process resulted in retrieving 1,052 and 412 terms as car-related and pedestrian-related, respectively.

We then applied lemmatization to the retrieved words, resulting in 957 and 358 related terms, respectively, for car and pedestrian [10]. We decided to use lemmatization rather than stemming since both reduce the inflectional forms of terms, while lemmatization preserves the derivationally related words, such as those starting or ending with un-, dis-, mis-, -ness, -ish, -ism, -ful, and -less. This is accomplished by specifying the words' part-of-speech tags (grammatical roles).

Given the expanded list of domain-specific terms, to further improve the quality of the search, each term was automatically searched in additional sources, possibly including detailed specifications of the concept-related term, such as online dictionaries and documents. For this purpose, two different online encyclopedias, namely Britannica and Wikipedia, were first searched for each term in the extended list. The Google search engine was secondly utilized for each *term*, being replaced in a search query as "What is the *term*?". The documents related to the first 100 returned links were retrieved for each query. We performed level one web scraping for each document, meaning that we only extracted the textual information and not the additional links within each page. This phase retrieved a large set of documents related to each augmented term.

As we used publicly available services, we faced Google search engine rate limit of 5 requests per 20 minutes and 20-30 requests per minute on Wikipedia and Britannica. Given the 957 terms related to the car and the rate limit, this process took about 127.6 hours in total for using the Google search engine to find the related links on Britannica and Wikipedia which

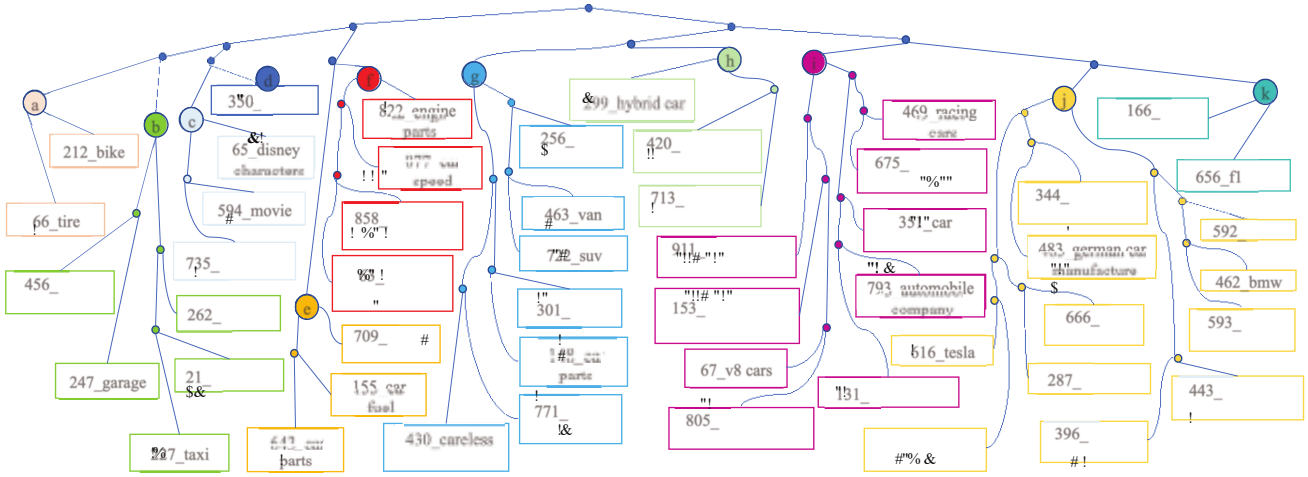

Fig. 3. Topic Hierarchy Relevant to the Domain Concept Car.

led to retrieving a total of 130,964 documents in 72.7 hours. The document length average for the retrieved terms related to cars was about 73 lines. Given 358 terms related to pedestrians, search process took approximately 47.7 hours in total and led to retrieving 51,963 documents in 28.8 hours. The document length average for retrieved terms related to pedestrians was about 52 lines. Table I represents the five most similar topics of each hard-to-specify domain concept, car, and pedestrian. Due to space limitations, the rest of topics are available in our repository.

B. Concept Representation

Each set of documents is then organized into a meaningful hierarchy of topics. Although topic models such as LDA [11] and NMF [34] have shown promises for topic modeling, tuning their hyper-parameters is often challenging. For this reason, to identify dominant topics of relevance to the domain concept, we adopted a transformer-based topic modeling technique [22] shown to produce highly cohesive clusters [58].

TABLE II
HIGH-LEVEL TOPICS OF *car* AND *pedestrian* DOMAIN CONCEPTS.

Id	Car Topic	Pedestrian Topic
a	wheels	transportation modes
b	car lots	locomotive
c	car-related movies	road features
d	jaguar cars	car accidents
e	engine features	road types
f	engine combustion system	racial protest
g	larger cars	pedestrians in customs
h	smaller more efficient cars	visual perception difficulties
i	car makers	pedestrians with moving disabilities
j	car types	safety
k	car racing	children
l	-	car communication system
m	-	background context
n	-	careless pedestrians
o	-	campers

This approach first converts any sentence of each document to an embedding vector (numerical values). It maps each vector in a multi-dimensional space so that vectors of contextually similar sentences are placed closer to each other. The model we selected was pre-trained, containing the embedding representations of generic words and sentences. However, due to the domain-specificity characteristics of the concept, we decided to retrain the model with the collection of domain-specific documents we earlier retrieved. To facilitate the arrangement of the embedding vectors, later a dimension reduction technique was selected, namely UMAP [39], shown to well preserve a significant portion of the structure of high-dimensional data. Later a hierarchical density-based algorithm, namely HDBSCAN [38] was applied to optimally cluster the documents embedding according to the clusters cohesion.

As the hierarchy is meant to display the concept, it is essential to provide meaningful names for each topic. Our approach selects the top four important words to represent the topic of each cluster. The importance of the word is computed according to *class-based TF-IDF metric (c-score)*, which is the same as the standard TF-IDF only regularizes the frequent words in each cluster instead of each document. Hence, the scores are a proxy of information density relevant to each cluster. Looking at the top words we manually selected a general topic for each cluster. The topic selection took less that five minutes for the both domain concepts.

A few default parameters in the topic model which we adopted [22] could be tuned to improve the model performance. We chose not to adjust the parameters to minimize the manual interference with our automated process. For instance, the default parameter retrieves the ten most similar words per topic, but the top_n_words parameter could modify this number. The topic representations can be controlled through the n_gram_range variable, specifying the number of most similar words selected as the topic. In addition, the number of the clusters can be adjusted, while setting the parameter too low for a large set of documents leads to a large number of microclusters which a higher parameter will merge to reduce their number. There are other variables related to the UMAP for controlling the number of neighboring, reducing the embedding dimensionality and distance metric.

This process lead to identifying 1,699 and 843 topics for cars and pedestrians, respectively. Due to the nature of HDBSCAN, we cannot specify the number of clusters in advance, but we can reduce the number of topics that have been created

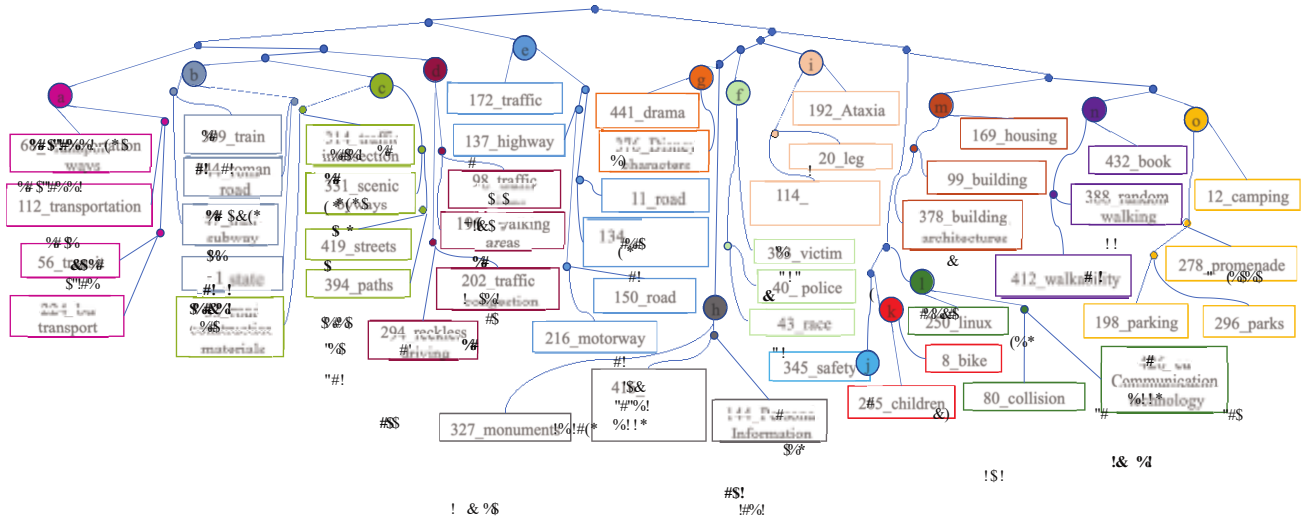

Fig. 4. Topic Hierarchy Relevant to the Domain Concept Pedestrian.

by merging the most similar topics automatically. As such, the model will reduce the number of topics, starting from the least frequent topic, as long as the two topics' similarity exceeds a minimum similarity of 0.915. To use this option, we set nr topics to auto. After the automatic topics reduction, 920 and 452 topics for cars and pedestrians were made, respectively.

To facilitate interpreting the hierarchy of topics, we manually assigned a higher-level topic to a set of topics in the same car hierarchy branch. These super-topics are represented in Table II and each corresponding id is displayed in the hierarchy in Figure 3. For instance, both clusters under node *k* contain terms closely relevant to car racing. For instance, cluster 166 includes terms such as NASCAR (National Association for Stock Car Auto Racing), Wallace (Bubba Wallace), Daytona (500-mile-long NASCAR Cup Series), motorsports (the website of racing results, news, and driver information), lap, race, speedway, Busch (Kyle Busch) and cluster 656 has words such as F1, IndyCar, McLaren, Andretti, race, Ferrari. As another example node *j* consists of clusters with car manufacturer topics, such as Benz, Tesla, Alfa Romeo, BMW, etc. Node *h* primarily contains small and fuel-efficient cars and related terms while node *g* includes terms relevant to large-size vehicles, such as van, SUV, and truck, under the right branch, and safety-related terms within the left-side clusters, such as NHSTA (National Highway Traffic Safety Administration), pedestrian, car, and collision in cluster 771, as well as device, texting, vehicle, crash, and driver in cluster 430.

Similarly, the last column of Table II, shows the high-level topics we selected for a subset of clusters within a hierarchical branch for the pedestrian benchmark in Figure 4. For instance, the clusters under node *i* talk about pedestrians with moving disabilities as cluster 192 contain Ataxia, Diplegia, Palsy, Cerebellum (a part in the brain that controls walking balance), gait, disorder, syndrome, and cerebral. Cluster 20 contains leg-related terms such as leg, ankle, foot, toe, dance, and crossing. Cluster 114 consists of terms related to people with disabilities, such as a wheelchair, Paralympic, disability, and Paralympics. As another example, the node *l* directly refers to issues relevant to in-car communication systems which lead to car collisions as cluster 250 contains terms such as Linux, driver, vehicle, umdf (User Mode Driver Framework), and kmdf (Kernel-Mode Driver Framework), and cluster 426 consists of communication technology terms, such as vanet (Kernel-Mode Driver Framework), v2x, v2v, cv2x (communication technologies), vehicle, vehicular, and 80211p (IEEE 802.11 standard to add wireless access in vehicular environments (WAVE), a vehicular communication system). On the other hand, cluster 80 within the same branch contains the collision-related terms, including brake, fatality, NHSTA, collision, accidents, velocity, drive, and momentum.

C. Dataset Interpretation

To evaluate the benchmarks, we selected three datasets specific to the autonomous driving domain: the CityPerson [70], EuroCity [12], and Caltech [20] datasets, respectively including 2,975, 23,892, and 4,285 image frames.

To interpret the datasets with respect to the classes of *pedestrian* and *car*, we used an *anchor-based object detection* technique related to computer vision and image processing [21]. The process of object detection typically happens through two levels: one involving image classification and the other object localization. While image classification assigns an object to one or multiple existing classes, object localization identifies the location of a potential object by drawing an imaginary surrounding *bounding box* around its extent. To localize an object, the anchor-based object detection algorithm first predicts an object's position in an image by creating predefined *anchor boxes* in the image. Anchor boxes are referred to as candidate boxes a model initially predicts to identify an object's location, size, and shape. For each anchor box later, the detector calculates a probability according to the Intersection over Union (IoU), implying the overlapping areas of the finally selected anchor box and the ground truth [49].

We adopted a Faster R-CNN model equipped with ResNeXt-101-FPN backbone [37], [64], a batch size of 8, and an initial learning rate of 8×10^3. We trained the model on the training set of Visual Genome, a large and dense general

dataset containing 108,077 images with a detailed description of each image [32]. The dataset includes 75,729 unique objects (labels), and each image has an average of 35 objects, 26 attributes, and 21 pairwise relationships between objects [32]. The model detected 81, 116, and 96 distinct objects in the CityPerson, EuroCity, and Caltech datasets out of the 141 of the visual genome distinct objects present in the three datasets.

D. Dataset Evaluation

We then automatically searched the detected objects within both car and pedestrian benchmarks for each dataset. Figures 5 and 6 represents the topics with respect to each benchmark. The nodes marked in orange represent the missed topics by each dataset which is marked with a circle on top of the node. The blue circle illustrates the missing topics of CityPerson, while green and yellow are representative of EuroCity and CalTech Datasets.

As shown, 60% and 48% of topics are missed by the three datasets in car and pedestrian benchmarks, respectively. With respect to the car variances, as identified in Table II, the racing cars, racing-relevant concepts, and a majority of car makes such as Benz, Cadillac, and luxurious cars (e.g., Bentley and Rolls-Royce in topic 675), smaller cars (coup and hatchback cars), and engine-related or external car parts (e.g., exhaust and thrust) are not recognized in any of the datasets. With respect to the pedestrian benchmark as displayed in Figure 6 the branch of pedestrians with walking disabilities, as well as node *n* related to unexpected pedestrians containing terms, such as stochastic, walk, step, Brownian (movement), Markov, diffusion, and random in cluster 388 (randomly walking pedestrians); book, magazine, Pulitzer, novel, walk and reading in cluster 432 (distracted walking pedestrians); and PERS (Pedestrian Environment Review System), preps, arrest, walk, walkability, pedestrian, and walkable (pedestrians walking in unexpected areas due to the walkability issues of the environment), are not covered by any of the datasets.

E. Qualitative Evaluation

We manually tested the terms in the most similar 50 topics of both car and pedestrian domain terms to verify the relevance of the terms to the given topic, as well as to the domain. Yet to additionally seek an external opinion, we designed an evaluation process. For obvious reasons, the generated topics and terms in this domain (i.e., driving) do not necessarily require domain experts and rare domain expertise; instead, they can be evaluated by the common sense knowledge of regular drivers. Therefore, for evaluation purposes, we created an online multiple-choice survey. Each question contained a topic and the 10 top words of the topic. We sent the survey to five computer science PhD students who were not involved in the research topic. We asked the participants to mark as many terms below a topic as they find relevant to the given topic.

The surveys indicated an average of 72% and 78% participants' agreement with the relevance of the extracted terms and the suggested topics in the pedestrians and cars topic models, respectively. The surveys are available in our repository.

VII. Related Work

RE for AI: Since engineering software systems with AI components highly depend on data with limited or no insight on how to map the data to the system performance, RE faces new challenges in this domain [4]. One main focus of RE for AI research area is on data requirement [4]. For instance, multiple work emphasized on differences of development process in conventional software and software systems with machine-learned components. In such systems, part of development is derived by data [26], [28], [46].

To this end, the Google PAIR guideline emphasizes bias prevention by ensuring that data is comprehensive [1]. Another work highlights that during the RE analysis phase, requirements engineers need first to understand the prescriptive data lineage to facilitate the discussion of data quality and preparation between customers and the data scientists [60].

In [53], the authors investigated the requirements of data quality, applying machine learning algorithms in the energy desegregation domain. The findings point out that the performance of algorithms has a highly positive correlation with the data sampling rate. Furthermore, [13] states that researchers should not solely rely on static methods to verify the qualitative characteristics of data. Therefore, they listed five essential characteristics for data quality, including accuracy, completeness, consistency, credibility, and correctness.

Domain analysis: Extracting and making use of domain knowledge has played a significant role in improving the quality of software products and the efficacy of software development processes [6], [42], [45]. For this reason, several studies in the SE domain have previously sought to extract domain knowledge for various concepts from the existing domain documents [15], [23], [59]. Several studies have gone further by capturing the retrieved information in the form of a semantic web or ontology [18], [35].

Visual perception: Many of the advanced automated driving functionalities require software components to perceive the environment in the automotive domain. The majority of perception-related functionalities may not be completely specifiable due to the presence of hard-to-specify concepts in the environment, such as pedestrian and automobile [54]. In this regard, Kondermann [31] argues that the RE practices are not yet properly applied to AI, specifically in tasks relevant to AI tasks relevant to visual perception (e.g., pedestrian recognition). He later emphasizes the need to investigate further applications of RE to include data selection techniques [31]. For example, while one study provides evidence for overfitting the state-of-the-art pedestrian detectors [24], another prior inspection of a commonly used pedestrian dataset revealed the lack of images of pedestrians in wheelchairs [46]. The accessible domain semantic webs include a short and incomplete specification of the term pedestrian while failing to capture the details of varying instances of the concept sufficiently. For example, WordNet defines a pedestrian as a "person who travels by foot" and associates the word with the terms *walker* and *footer*. However, this definition is limited given that it

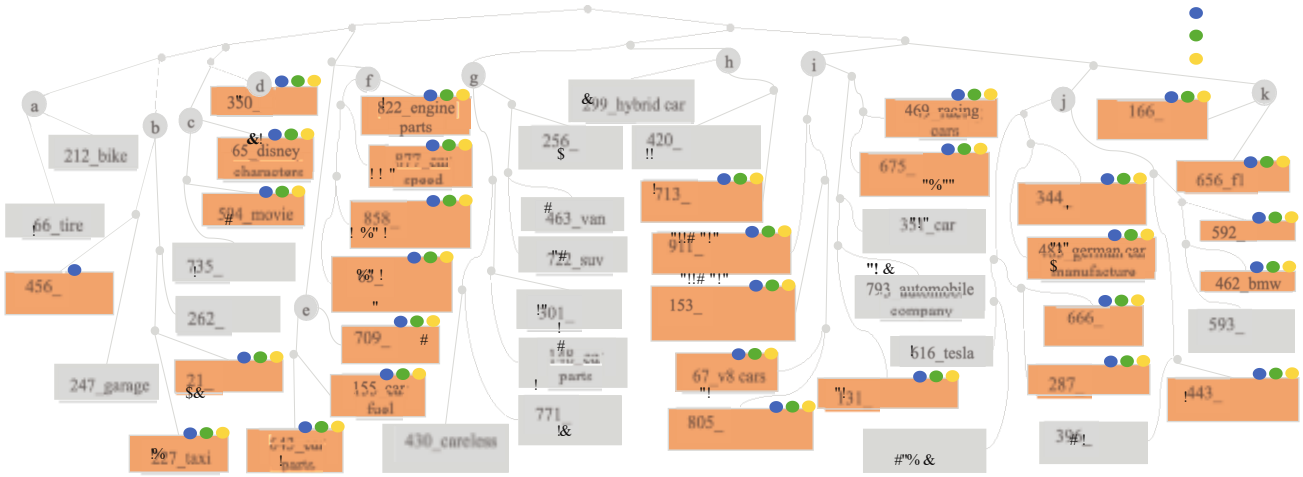

Fig. 5. The Missed topics(Orange nodes) of Car Hierarchical Topics in the Three Autonomous Driving Benchmark Datasets.

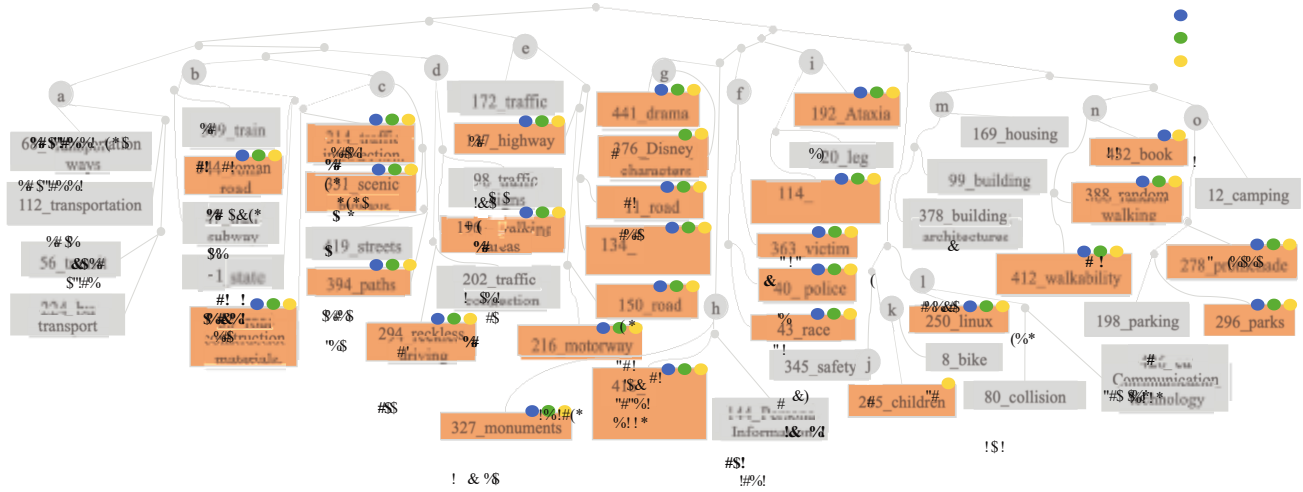

Fig. 6. The Missed topics(Orange nodes) of Pedestrian Hierarchical Topics in the Three Autonomous Driving Benchmark Datasets.

excludes, for example, pedestrians riding a bike, roller-skating, or using a wheelchair. It also fails to describe a pedestrian's appearance in terms of attributes, such as clothing and posture.

VIII. Threats

Although we used multiple sources, such as online encyclopedias and Britannica, to create the reference points for automotive-related datasets, the benchmark completeness can be questioned. We tried to at least estimate the topics completeness relative to domain-specific dataset benchmarks. As discussed, several primary and sensitive topics within the benchmarks were not covered in any of the datasets. A threat to the construct validity may arise from the evaluations with a limited number of datasets and object detectors. For instance, the missing topics in the dataset benchmarks may be due to poor training of the object detector since they are trained on a set of limited objects. We minimized this threat by selecting well-performing detectors and most commonly used state-of-the-art datasets. For our future work, we intend to use more advanced image processing techniques, such as scene graph generation and region captioning [36], to extract more information from the visual datasets to be compared against our automotive domain benchmarks. A threat to the external validity is carrying out the experiments only in one domain (automotive) and two domain concepts (pedestrian and car). We designed a generalizable process, and implemented a general framework, and referred to general knowledge sources. Therefore, no limitation is foreseen to extend the application domain. However, due to the expensive computations, we limited the application to two concepts of the automotive domain for which the accuracy of visual perception tasks is particularly important for the reliability of the functions.

IX. Conclusion

This paper presented a generic process for evaluating dataset requirements of AI-enabled perception software. We emphasized on extracting the necessary knowledge to build an inclusive reference point of domain concepts against which the relative completeness of a dataset can be assessed. To build a reference point, we implemented an automated approach to collect domain-specific knowledge from online sources such as encyclopedias and dictionaries. We evaluated our framework, called CADE, against three autonomous driving dataset benchmarks and identified that several critical topics

related to the concepts were missing in the widely-used dataset benchmarks. We then qualitatively evaluated the generated hierarchies of topics with independent researchers resulting in 75% participants agreement.

X. Acknowledgments

This research is partially funded by NSF: CCF: 124606 and partially used resources of ddiLab Laboratory. Laboratory.

References

[1] Google research. (2019) the people + ai guidebook. Available at pair.withgoogle.com/guidebook. Accessed: 2022-01-12.

[2] Onelook dictionary search. Available at https://www.onelook.com/. Accessed: 2022-01-08.

[3] Related words. Available at https://www.relatedwords.org/. Accessed: 2022-01-08.

[4] K. Ahmad, M. Bano, M. Abdelrazek, C. Arora, and J. Grundy. What's up with requirements engineering for artificial intelligence systems? In *2021 IEEE 29th International Requirements Engineering Conference (RE)*, pages 1–12. IEEE, 2021.

[5] H. H. Altarturi, K.-Y. Ng, M. I. H. Ninggal, A. S. A. Nazri, and A. A. Abd Ghani. A requirement engineering model for big data software. In *2017 IEEE Conference on Big Data and Analytics (ICBDA)*, pages 111–117. IEEE, 2017.

[6] S. S. Anand, D. A. Bell, and J. G. Hughes. The role of domain knowledge in data mining. In *Proceedings of the fourth international conference on Information and knowledge management*, pages 37–43, 1995.

[7] A. B. Arrieta, N. Dıáz-Rodrıǵuez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcıa,́ S. Gil-Lopẽz, D. Molina, R. Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

[8] A. Banks and R. Ashmore. Requirements assurance in machine learning. In *Workshop on Artificial Intelligence Safety*, 2019.

[9] H. Barzamini, M. Shahzad, H. Alhoori, and M. Rahimi. A multi-level semantic web for hard-to-specify domain concept, pedestrian, in ml-based software. *Requirements Engineering*, pages 1–22, 2022.

[10] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[11] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[12] M. Braun, S. Krebs, F. Flohr, and D. Gavrila. EuroCity persons: A novel benchmark for person detection in traffic scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, Feb. 2019.

[13] H. Challa, N. Niu, and R. Johnson. Faulty requirements made valuable: on the role of data quality in deep learning. In *2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, pages 61–69. IEEE, 2020.

[14] G. G. Chowdhury. Natural language processing. *Annual review of information science and technology*, 37(1):51–89, 2003.

[15] J. Cleland-Huang. Mining domain knowledge [requirements]. *IEEE Software*, 32(3):16–19, 2015.

[16] I. Dagan, L. Lee, and F. C. Pereira. Similarity-based models of word cooccurrence probabilities. *Machine learning*, 34(1):43–69, 1999.

[17] M. Davies. Google books corpus. (based on google books n-grams). Available at https://www.english-corpora.org/googlebooks. Accessed: 2022-01-5.

[18] D. Dermeval, J. Vilela, I. I. Bittencourt, J. Castro, S. Isotani, P. Brito, and A. Silva. Applications of ontologies in requirements engineering: a systematic review of the literature. *Requirements Engineering*, 21(4):405–437, 2016.

[19] T. Dietterich. Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, 27(3):326–327, 1995.

[20] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 304–311. ieeexplore.ieee.org, June 2009.

[21] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[22] M. Grootendorst. Bertopic: Leveraging bert and c-tf-idf to create easily interpretable topics., 2020.

[23] J. Guo, M. Gibiec, and J. Cleland-Huang. Tackling the term-mismatch problem in automated trace retrieval. *Empirical Software Engineering*, 22(3):1103–1142, 2017.

[24] I. Hasan, S. Liao, J. Li, S. U. Akram, and L. Shao. Generalizable pedestrian detection: The elephant in the room. Mar. 2020.

[25] B. C. Hu, R. Salay, K. Czarnecki, M. Rahimi, G. Selim, and M. Chechik. Towards requirements specification for machine-learned perception based on human performance. In *2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, pages 48–51, 2020.

[26] B. C. Hu, R. Salay, K. Czarnecki, M. Rahimi, G. Selim, and M. Chechik. Towards requirements specification for machine-learned perception based on human performance. In *2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, pages 48–51. IEEE, 2020.

[27] E. Hull, K. Jackson, and J. Dick. A generic process for requirements engineering. In *Requirements Engineering*, pages 21–40. Springer, 2005.

[28] C. Kaestner. Machine learning is requirements engineering — on the role of bugs, verification, and validation in machine learning. https://medium.com/analytics-vidhya/machine-learning-is-requirements-engineering-8957aee55ef4, 2020.

[29] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE transactions on pattern analysis and machine intelligence*, 24(7):881–892, 2002.

[30] M. D. Kohli, R. M. Summers, and J. R. Geis. Medical image data and datasets in the era of machine learning—whitepaper from the 2016 c-mimi meeting dataset session. *Journal of Digital Imaging*, 30(4):392–399, 2017.

[31] D. Kondermann. Ground truth design principles: an overview. In *Proceedings of the International Workshop on Video and Image Ground Truth in Computer Vision Applications*, pages 1–4, 2013.

[32] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.*, 123(1):32–73, May 2017.

[33] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2891–2903, 2013.

[34] D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13, 2000.

[35] Y. Li and J. Cleland-Huang. Ontology-based trace retrieval. In *2013 7th International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE)*, pages 30–36. IEEE, 2013.

[36] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang. Scene graph generation from objects, phrases and region captions. In *Proceedings of the IEEE international conference on computer vision*, pages 1261–1270, 2017.

[37] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[38] L. McInnes, J. Healy, and S. Astels. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017.

[39] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[40] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[41] K. Nakamichi, K. Ohashi, I. Namba, R. Yamamoto, M. Aoyama, L. Joeckel, J. Siebert, and J. Heidrich. Requirements-driven method to determine quality characteristics and measurements for machine learning software and its evaluation. In *2020 IEEE 28th International Requirements Engineering Conference (RE)*, pages 260–270. IEEE, 2020.

[42] J. M. Neighbors. *Software construction using components*. University of California, Irvine, 1980.

[43] R. Oliveto, M. Gethers, D. Poshyvanyk, and A. De Lucia. On the equivalence of information retrieval methods for automated traceability link recovery. In *2010 IEEE 18th International Conference on Program Comprehension*, pages 68–71. IEEE, 2010.

[44] M. M. Petrou and C. Petrou. *Image processing: the fundamentals*. John Wiley & Sons, 2010.

[45] R. Prieto-Diaz. Domain analysis: An introduction. *ACM SIGSOFT Software Engineering Notes*, 15(2):47–54, 1990.

[46] M. Rahimi, J. L. Guo, S. Kokaly, and M. Chechik. Toward requirements specification for machine-learned components. In *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, pages 241–244. IEEE, 2019.

[47] F. Rahutomo, T. Kitasuka, and M. Aritsugi. Semantic cosine similarity. In *The 7th International Student Conference on Advanced Science and Technology ICAST*, volume 4, page 1, 2012.

[48] A. Rasouli, I. Kotseruba, and J. K. Tsotsos. It's not all about size: On the role of data properties in pedestrian detection. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[49] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

[50] R. Salay and K. Czarnecki. Using Machine Learning Safely in Automotive Software: An Assessment and Adaption of Software Process Requirements in ISO 26262. *ArXiv*, abs/1808.01614, 2018.

[51] R. Salay and C. Krzysztof. Using machine learning safely in automotive software: An assessment and adaption of software process requirements in iso 26262. *arXiv preprint arXiv:1808.01614*, 2018.

[52] K. Sandkuhl. Putting ai into context-method support for the introduction of artificial intelligence into organizations. In *2019 IEEE 21st Conference on Business Informatics (CBI)*, volume 1, pages 157–164. IEEE, 2019.

[53] C. Shin, S. Rho, H. Lee, and W. Rhee. Data requirements for applying machine learning to energy disaggregation. *Energies*, 12(9):1696, 2019.

[54] B. Spanfelner, D. Richter, S. Ebel, U. Wilhelm, W. Branz, and C. Patz. Challenges in applying the iso 26262 for driver assistance systems. *Tagung Fahrerassistenz, München*, 15(16):2012, 2012.

[55] R. Speer, J. Chin, and C. Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge, 2017.

[56] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3716–3725. openaccess.thecvf.com, 2020.

[57] K. Tang, H. Zhang, B. Wu, W. Luo, and W. Liu. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6619–6628, 2019.

[58] L. Thompson and D. Mimno. Topic modeling with contextualized word representation clusters. *arXiv preprint arXiv:2010.12626*, 2020.

[59] K. Verma and A. Kass. Requirements analysis tool: A tool for automatically analyzing software requirements documents. In *International semantic web conference*, pages 751–763. Springer, 2008.

[60] A. Vogelsang and M. Borg. Requirements engineering for machine learning: Perspectives from data scientists. In *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, pages 245–251. IEEE, 2019.

[61] A. Voutilainen. Part-of-speech tagging. *The Oxford handbook of computational linguistics*, pages 219–232, 2003.

[62] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al. Constrained k-means clustering with background knowledge. In *Icml*, volume 1, pages 577–584, 2001.

[63] H. M. Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977– 984, 2006.

[64] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[65] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5419, 2017.

[66] P. Xu, X. Chang, L. Guo, P.-Y. Huang, X. Chen, and A. G. Hauptmann. A survey of scene graph: Generation and application. *IEEE Trans. Neural Netw. Learn. Syst*, 2020.

[67] X. Ying. An overview of overfitting and its solutions. In *Journal of Physics: Conference Series*, volume 1168, page 022022. IOP Publishing, 2019.

[68] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5831–5840, 2018.

[69] K. Zhai, J. Boyd-Graber, N. Asadi, and M. L. Alkhouja. Mr. lda: A flexible large scale topic modeling package using variational inference in mapreduce. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 879–888, New York, NY, USA, 2012. ACM.

[70] S. Zhang, R. Benenson, and B. Schiele. Citypersons: A diverse dataset for pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3221. openaccess.thecvf.com, 2017.

[71] L. Žust and M. Kristan. Learning maritime obstacle detection from weak annotations by scaffolding. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 955–964, 2022.