Evaluation-as-a-Service for the Computational Sciences: Overview and Outlook

FRANK HOPFGARTNER, University of Sheffield
ALLAN HANBURY, TU Wien, Complexity Science Hub Vienna
HENNING MÜLLER and IVAN EGGEL, University of Applied Sciences Western Switzerland (HES-SO)
KRISZTIAN BALOG, University of Stavanger
TORBEN BRODT, plista GmbH
GORDON V. CORMACK and JIMMY LIN, University of Waterloo
JAYASHREE KALPATHY-CRAMER, Athinoula A. Martinos Center for Biomedical Imaging at Massachusetts General Hospital and Harvard Medical School
NORIKO KANDO, National Institute of Informatics
MAKOTO P. KATO, Kyoto University
ANASTASIA KRITHARA, National Center for Scientific Research "Demokritos"
TIM GOLLUB, Bauhaus-Universität Weimar
MARTIN POTTHAST, Leipzig University
EVELYNE VIEGAS, Microsoft Research
SIMON MERCER, Independent Consultant

Evaluation in empirical computer science is essential to show progress and assess technologies developed. Several research domains such as information retrieval have long relied on systematic evaluation to measure progress: here, the Cranfield paradigm of creating shared test collections, defining search tasks, and collecting ground truth for these tasks has persisted up until now. In recent years, however, several new challenges have emerged that do not fit this paradigm very well: extremely large data sets, confidential data sets as found in the medical domain, and rapidly changing data sets as often encountered in industry. Crowdsourcing has also changed the way in which industry approaches problem-solving with companies now organizing challenges and handing out monetary awards to incentivize people to work on their challenges, particularly in the field of machine learning.

This paper is based on discussions at a workshop on Evaluation-as-a-Service (EaaS). EaaS is the paradigm of not providing data sets to participants and have them work on the data locally, but keeping the data central and allowing access via Application Programming Interfaces (API), Virtual Machines (VM) or other possibilities to ship executables. The objectives of this paper are to summarize and compare the current approaches and consolidate the experiences of these approaches to outline the next steps of EaaS, particularly towards sustainable research infrastructures.

The paper summarizes several existing approaches to EaaS and analyzes their usage scenarios and also the advantages and disadvantages. The many factors influencing EaaS are summarized, and the environment in terms of motivations for the various stakeholders, from funding agencies to challenge organizers, researchers and participants, to industry interested in supplying real-world problems for which they require solutions.

We acknowledge financial support by the European Science Foundation via its Research Network Program "Evaluating Information Access Systems" (ELIAS) and by the European Commission via the FP7 project VISCERAL (318068).

1936-1955/2018/1-ART1 $15.00
https://doi.org/10.1145/3239570

EaaS solves many problems of the current research environment, where data sets are often not accessible to many researchers. Executables of published tools are equally often not available making the reproducibility of results impossible. EaaS on the other hand creates reusable/citable data sets as well as available executables. Many challenges remain but such a framework for research can also foster more collaboration between researchers, potentially increasing the speed of obtaining research results.

CCS Concepts: • **Information systems** → *Data management systems*; *Information systems applications*; *Information retrieval*;

Additional Key Words and Phrases: evaluation-as-a-service, benchmarking, information access systems

ACM Reference format:
Frank Hopfgartner, Allan Hanbury, Henning Müller, Ivan Eggel, Krisztian Balog, Torben Brodt, Gordon V. Cormack, Jimmy Lin, Jayashree Kalpathy-Cramer, Noriko Kando, Makoto P. Kato, Anastasia Krithara, Tim Gollub, Martin Potthast, Evelyne Viegas, and Simon Mercer. 2018. Evaluation-as-a-Service for the Computational Sciences: Overview and Outlook. *ACM J. Data Inform. Quality* 1, 1, Article 1 (January 2018), 32 pages.
https://doi.org/10.1145/3239570

1 INTRODUCTION

One of the most important aspects of scientific work is the evaluation of research questions or research hypotheses that allows us to assess the merits or limitations of new ideas. In areas of computer science such as Machine Learning (ML) and Information Retrieval (IR), evaluation often focuses around the question of how well a developed algorithm performs. Given the manifold scenarios in which algorithms can be employed, some of the main challenges are to perform evaluations that are not only transparent enough so that experiments can be reproduced by third parties, but also allow for a direct comparison with state-of-the-art algorithms or strong baselines. In the computational sciences, this challenge has been addressed by evaluation campaigns that release shared data sets consisting of data, tasks, associated ground truth, and other resources such as toolkits or additional metadata. Using these resources, interested researchers can develop algorithms or systems, perform their experiments locally and submit the results to the organisers of the campaigns who can then evaluate the performance using standardized evaluation metrics [Spärck Jones and van Rijsbergen 1975]. Given the requirement to share resources to enable such coordinated evaluation efforts, the tasks are also referred to as shared evaluation tasks.

In the IR research community, this structured evaluation methodology is sometimes referred to as Cranfield evaluation paradigm as it was first trialled in the 1960's by researchers of the Cranfield College of Aeronautics [Richmond 1963]. A more descriptive name for the concept is, however, *Data-to-Algorithms* (DtA) paradigm. The DtA paradigm requires participants to submit the output of their algorithms when run on a pre-published test data set (a so-called "run"). There are regular academic evaluation campaigns such as the Text REtrieval Conference (TREC) [Voorhees and Harman 2005], Conference and Labs of the Evaluation Forum (CLEF) [Ferro and Silvello 2014], and the NTCIR (NII Test Collection for IR Systems) [Kudo 2010] that employ the DtA paradigm for specific research challenges in the field of IR. In the area of machine learning and computer vision, the series of PASCAL Challenges [Quiñonero-Candela et al. 2006] from 2005–2013 are well known. Similar initiatives (see e.g., [Light and Börner 2013; Thomee et al. 2016]) exist in other domains such as multimedia and web science. Similarly, in a non-academic setting, various commercial platforms exist that rely on the DtA paradigm. Popular examples include the data science competition platforms Kaggle[1] and TopCoder[2] where participants have to download data

[1] http://www.kaggle.com/
[2] http://www.topcoder.com/

and submit the results of their computations. Another example is The Dream Challenges[3] which offers competitions to solve biomedical challenges. An open source solution machine learning challenge is crowdAI[4], maintained by the Digital Epidemiology Lab at EPFL in Switzerland.

Although the DtA paradigm is the de-facto standard evaluation approach in the computational sciences, it has several shortcomings, including a complete lack of reproducibility of the shared task, and the necessity to publish test data sets prematurely, albeit sans ground truth [Harman 1992]. Further, as Kaggle specifically acknowledges on their website: "While we are sympathetic to the fact that not everyone has access to a stellar broadband connection, the plumbing needed to move data is an unavoidable part of practicing data science."[5]. Notwithstanding these shortcomings, the organizers of shared tasks frequently employ run submission for its minimal organizational overhead. Criticisms of this paradigm have discussed the need for continuous evaluation and not only linked to a competition and also component evaluation, which is important to better understand the performance linked to the many components of a system in a complex way [Hanbury and Müller 2010]. We argue that in several cases, the DtA approach of distributing data is just not practical, because the data may be:

- Huge – For obtaining real world evaluation results, the evaluation needs to be performed on realistic amounts of data. In the case of web search, this could be Petabytes.[6] The currently most common approach of sending data on hard disks through the postal service or distribution via download has its limitations.
- Non-distributable – It is often not allowed to distribute data due to privacy, terms of service, or commercial sensitivity of the data. Privacy is an important concern for personal medical data. Even though the law often allows the distribution of anonymized medical data, large-scale anonymization can only be accomplished automatically and data owners usually do not trust it [Safran et al. 2007]. The Twitter Terms of Service[7] forbid redistribution of tweets, while query logs are not made available for researchers after the debacle surrounding the release of the AOL search logs in 2006 [Arrington 2006]. Distribution of company documents for the evaluation of enterprise search are not permitted due to the commercial sensitivity of the data [Kruschwitz and Hall 2017].
- Real-time – Companies working on real-time data, for example recommender systems, are often not interested in evaluation results obtained on static historical data, in particular if these data have to be anonymized to allow distribution, as these results are too far removed from their operative requirements that require real-time data access [Levy 2013].

In this paper, we explore an alternative evaluation methodology for the computational sciences based on the so-called *Algorithms-to-Data paradigm* (AtD) in which the data are all stored on a (central) computational infrastructure, and can only be accessed on this infrastructure [Hanbury et al. 2012; Müller et al. 2016]. We argue that evaluation campaigns that employ the AtD paradigm could require participants to either upload their algorithms, e.g., embedded in a Virtual Machine (VM) to a centralized infrastructure where they can be run on the provided data, or gain access to the data via an API. This does not only ease the burden of downloading vast amount of data but can also avoid access to sensitive data, as only the algorithms and not the developers need to see the data. Similar to the concept of shared evaluation campaigns, we refer to this type of evaluation as *Evaluation-as-a-Service* (EaaS) as the main emphasis is on offering a complete

[3] http://dreamchallenges.org/
[4] http://www.crowdai.org
[5] https://www.kaggle.com/wiki/ANoteOnTorrents, visited on 28/08/2017
[6] For example, the Common Crawl data set which is available at http://commoncrawl.org/ currently consists of petabytes of data.
[7] https://twitter.com/tos

computational infrastructure that serves as eco system for the evaluation rather than sharing data sets and evaluation protocols only. By tackling the above hurdles, we argue that EaaS can have a strong impact on empirical evaluation in the computational sciences in general, and further offers new opportunities for fruitful collaboration of academia and industry.

The main contributions of this paper are as follows:

- We define EaaS as a novel and impactful evaluation methodology and discuss its strengths and limitations.
- We discuss the main challenges that need to be addressed when offering EaaS.
- We provide a detailed survey on evaluation initiatives in the computational sciences that offer some form of EaaS.

The paper is structured as follows. We first motivate EaaS as a novel evaluation methodology in Section 2. In Section 3, we survey existing campaigns that already employ EaaS, or components thereof, for the evaluation of different tasks. This includes an introduction of the different use cases, followed by an overview of technologies used, and finally a discussion on various shortcomings. In Section 4 we then reflect on these initiatives and identify aspects to consider in the next steps toward implementing EaaS fully. Section 5 concludes this work.

2 MOTIVATION

As explained in the introduction, an established evaluation method in the computational sciences is to rely on shared resources (i.e., data sets or software) that are provided as part of an evaluation campaign. Such evaluation is based on the DtA paradigm which requires participants to download these resources and then run their experiments on their own IT infrastructure. As evidenced by the large numbers of papers that have been published in the past few years at top-tier conferences such as ACM SIGIR, ICML, or SIGKDD, this method has been the de-facto evaluation standard for research on computational sciences. However, despite its success, the approach does not come without caveats. As mentioned before, the main drawbacks include the challenges of sharing large-scale data sets, limitations due to the sensitive nature of data sets, and the emerging need to process streamed data in real-time. In the context of IR, Hawking [Hawking 2015] points out that these limitations already reduce the share of non-industrial research presented at the leading IR conference SIGIR. Further, as [Hanbury and Müller 2010] discuss, the DtA paradigm does not provide solutions for continuous evaluation, and it cannot easily be employed when ground truth data is not available. In the computational sciences in general, little focus has been directed toward the reproducibility of experimental results, raising questions about their reliability [Freire and Silva 2012]. There is currently work underway to counter this situation, ranging from presenting the case for open computer programs [Ince et al. 2012], through creating infrastructures to allow reproducible computational research [Freire and Silva 2012] to considerations about the legal licensing and copyright frameworks for computational research [Stodden 2009]. Given these limitations, parties with access to large IT infrastructure and large user bases have moved away from this paradigm and now increasingly perform online testing to evaluate their algorithms and systems. For a detailed introduction to online evaluation of information retrieval systems, we refer to [Hofmann et al. 2016]. While online evaluation is a viable option for entities whose services create large user traffic, who have access to appropriate resources, and can process data with manageable legal limitations, there are only limited chances for researchers who do not have access to these resources.

The EaaS paradigm in which evaluation campaigns require participants to upload their code rather than downloading data has the potential to address several of these limitations. It does not only help academia to get access to interesting data and challenges but also allows industry to get access to results that could improve the offerings to their clients. Furthermore, funding agencies

and science as a whole can get benefit from the EaaS paradigm as well, as all projects funded can become comparable and data are not limited to a small group that can use them but can be shared virtually for the analysis with a large number of research groups. This can lead to generally better science that is more reproducible and where more time can be taken for the large data creation as more people can work together instead of creating many small data sets.

In order to shed more light on the application of EaaS as novel evaluation methodology, we first outline in Section 2.1 two scenarios that depict a vision on how EaaS can be implemented. Further, in Section 2.2 we outline the benefits for all stakeholders involved to illustrate the potentials of EaaS. Our motivation is that it should be ensured that the benefits significantly outweigh the efforts for everyone involved as this will be an essential requirement for the successful implementation and operation of EaaS.

2.1 EaaS Vision

Below, two scenarios describe situations in which EaaS is employed for the evaluation of different algorithms. The two scenarios are derived from two evaluation campaigns that have implemented the EaaS paradigm. A more detailed overview of these, and more, campaigns is provided in Section 3.

> **Scenario 1:** *Company X* provides a blog entry recommendation service based on a user profile and a user click history. *Company X* wishes to improve its recommendation algorithms, and decides to make this challenge open to all through the EaaS paradigm, thereby increasing the size of the pool of highly-qualified people from which the solution can come. Participants submit their proposed recommender systems as executables installed on VMs, where the parameters such as maximum response time are strictly specified. Thanks to the use of a standardised VM exchange format, the participation overhead is reduced, as the same VM images can be submitted to participate in any EaaS campaign. Upon submission of the VMs, standardized tests are automatically run to ensure that the systems satisfy the specified parameters, and any shortcomings are reported back to the participant. Once a system satisfies all parameters, it is randomly assigned requests for recommendation and is evaluated based on the clicks by end users of the *Company X* recommendation service on the links returned. A well-designed experimental protocol ensures that links suggested by the submitted recommender systems are shown often enough to obtain statistically significant results for all participants, and participants are sent results in the form of performance metric values linked to a permanent identifier that are ready to be inserted directly into a publication. *Company X* gets information on how well the performance of their recommender system compares to the state of the art, and can contact the teams having the best performance to discuss potential technology transfer.
>
> **Scenario 2:** *Company Y*, a pharmaceutical company, wishes to make drug development for *Disease Z* more efficient and cost-effective. It identifies that there are two main components to doing this, better extraction of key information from the biomedical literature, and better prediction of the outcomes of combining various ingredients. *Company Y* wishes to get the solution from the largest possible pool of experts, and therefore opens the challenge as EaaS. The two parts of solving the challenge are heavily interdependent (as prediction is influenced by the available facts), but require different skill sets on the part of the solution providers, so participants can select to participate in either extraction or prediction. A huge

collection of biomedical literature and of facts that are already known to *Company Y* are placed in protected form on an EaaS infrastructure, and some examples of these data are made public. People participate by submitting VMs containing their executable software to the EaaS infrastructure, where the VMs are sandboxed and the executables are run on the data. The standardised VM exchange format reduces the overhead for participants. Once the VM has run and produced results, it is destroyed (to ensure privacy of the data), while the outputs of the executables remain available for *Company Y* and the evaluation metric values are returned to the participant who submitted the VM. Participants can choose to make these metric values public, and receive a Digital Object Identifier allowing these results to be referred to in a publication. Visual analytics software on the EaaS infrastructure allows participants a detailed analysis of the performance of their algorithms and a comparison to other publicly available results. *Company Y* can examine combinations of the outputs of the submitted software of the two parts of the challenge to find the optimal combination, have experts examine newly extracted facts or predictions to evaluate their relevance (increasing the size of the ground truth), and contact the participants having the best performing submissions to negotiate terms for the further use and development of their techniques.

2.2 Benefits

After getting an introduction to the EaaS paradigm in the previous section, in this section we now highlight how participants and other stakeholders, such as campaign organizers or companies who employ the EaaS paradigm can benefit from this. The remainder of this section is structured as follows. We first highlight the advantages of these stakeholders in Section 2.2.1. Section 2.2.2 then sheds some light on the simplicity and cost-effectiveness of EaaS. Benefits of reproducibility and cooperation are discussed in Sections 2.2.3 and 2.2.4.

2.2.1 Advantages. Looking at academia and industry as stakeholders, one of the main advantages to both sides is that the EaaS paradigm should bridge the gap between the two sides, allowing more straightforward cooperation. This would in particular counter the commonly perceived view that scientists working in industry have access to larger amounts of more interesting data than those working in academia [Huberman 2012; Markoff 2012]. The EaaS paradigm benefits researchers in academia because it gives them access to industry data and to quickly publish results, but caters to industry because there is no necessity to release the data in any uncontrolled way, as the data can remain behind a firewall on company servers. By employing EaaS on sensitive data such as medical data or other material that can not easily be shared (e.g., copyright material, enterprise search data, etc.), new opportunities for wider research on such sensitive topics can be supported.

At the same time, EaaS allows access to data sets that would be too big to be shared and that a single research group or a small group could not easily assemble and treat. Thinking of the evaluation of approaches using real users, the EaaS paradigm also allows researchers in academia to evaluate methods using real-time data and under realistic conditions as common in industry [Kohavi et al. 2012], which again brings benefits to companies owning the data, as they get results on dynamic data currently of interest to them, rather than on static data from months or years back (the sort of data that could be considered less commercially interesting and hence more suitable to release to researchers in the traditional way).

This is a direct benefit for companies involved, as by making available data and associated challenges via the EaaS paradigm, they get exposure to the latest processing and analysis approaches from academia. So instead of working on solutions in-house as larger search engines currently do

with their log files, they get the possibility to access the best performing techniques, to get a clear idea of techniques, their performance and their stability based on the work of participants, which can also lead to new ideas or projects based on real data. At the same time, they can position themselves as leaders in the field, shape the task of many scientists by influencing research directions that tailor their needs, increase their visibility in the community, and even attract talented professionals who are interested in working on their use cases. As example, we point to one of the EaaS initiatives, presented in Section 3.2.6, in which a commercial provider of news and ad recommendations offered an EaaS platform that allowed for benchmarking of recommendation algorithms in the live system of the provider. Following the start of the initiative, the company received a number of job applications that directly referred to the evaluation campaign. Moreover, by interacting with the participants, the company could learn of innovative ideas on how to address their recommendation task. Finally, by allowing participants to benchmark the performance of their ideas in a live system, the company could save expensive development time.

For researchers in academia, beyond the obvious benefits of access to large amounts of interesting data linked to challenges of commercial relevance, there are also potential benefits in terms of increased reputation for organizers and participants. This would be particularly true for those EaaS instances that become accepted as benchmarks in a scientific area rather than one-off competitions. In a fully developed EaaS infrastructure, participants could also have access to technical benefits, such as an interface for performing visual analytics of the experimental results and potentially even a service to semi-automatically write the experimental section of a scientific paper. Potentially, submissions could also be automatically encapsulated as services to be made available on a demo webpage, thereby also increasing the visibility of the participants' work.

Both academia and industry can take advantage of the capability inherent in submitting VMs containing functioning services, namely the running of automated ensemble approaches. It is well known that ensembles of classifiers can often perform better than a single classifier [Dietterich 2000] — the EaaS paradigm makes it straightforward to test multiple combinations of classifiers in various ensemble approaches to obtain new scientific insight as well as better performing classifiers. An example study exploiting an EaaS infrastructure for ensemble learning is provided by [Lommatzsch 2014].

Finally, EaaS instances can be used as part of university courses on Data Science. As studies (e.g., [Ortiz-Repiso et al. 2018]) indicate, there is a growing need to give students experience with working on real challenges on huge amounts of data, instead of the "toy problems" that are generally part of their course work. This brings a benefit to industry by ensuring that university graduates entering the Data Science job market are better qualified for the work that they will have to do. Such EaaS instances could, for example, be combined with a Massive Open Online Course (MOOC). First efforts to establish EaaS campaigns as a resource for learning and teaching are outlined by [Hopfgartner et al. 2016].

2.2.2 Simplicity and Cost-Effectiveness. The benefits of EaaS should significantly outweigh the effort required from all stakeholders. While the previous section concentrated on making the benefits explicit, this section focuses on how the effort can be reduced.

To reduce the effort needed from organizers of EaaS instances, they should not be forced to set up the full infrastructure necessary for EaaS each time they organize a competition or benchmark, as this would be an unacceptable overhead. Optimally, EaaS infrastructures should be available that can be used for reasonable costs. These infrastructures should also be easily scalable in terms of the number of participants, so that the effort for a small and large number of participants is similar. Furthermore, there should be effective support in carrying out all the steps required in setting up

the challenge, including steps such as designing an effective evaluation protocol and selecting the most suitable metrics for the task.

The effort required for participants must also be reduced. An effective way of doing this is to use a common VM format that can be executed on all EaaS infrastructures. This means that participants could have prepared VMs containing their algorithms, and can submit them easily to a benchmark or competition on any EaaS infrastructure, avoiding the need to spend time in adapting the code to multiple cloud architectures. Where possible, standardisation of other aspects of a submission, such as data formats, could also lower participation overheads.

2.2.3 Reproducibility. The drawbacks of publishing papers containing the results of experiments done only on proprietary data that is not available to other researchers to ensure reproducibility of results has been widely discussed (e.g., in [Callan and Moffat 2012]). Fuhr [Fuhr 2017] even argues that despite the use of shared data sets that eliminates some of these drawbacks, reproducibility is still one of the main challenges in the field of IR evaluation. As he points out, methods have become so complex that a detailed description of all components and settings is sheer impossible. He therefore argues for sharing implementations of algorithms, e.g., on publicly accessible repositories.

The EaaS paradigm can contribute to increasing reproducibility of results through making available not only data and associated tasks in the long term but also a library of executable algorithms that have been applied to solving the tasks on the data, and the results that have been produced by these algorithms.

The availability of these resources should contribute to addressing an observed practice in computer science of comparing new algorithms to weak baselines [Armstrong et al. 2009]. It should also ensure access to results using a large palette of metrics, so that all aspects of the performance of an approach can be examined. This can include execution time info (efficiency), as every approach is evaluated on the same infrastructure.

Further contributions toward reproducibility can also be expected. As the data set is stored centrally, mechanisms can be put in place to collect additional ground truth for a task over time. Crowd-sourcing, either among competitors or among a wider group of people, could be used to obtain this additional ground truth [Foncubierta-Rodríguez and Müller 2012]. Whenever the ground truth available has expanded significantly, all approaches already submitted for a task could be re-evaluated automatically using the expanded ground truth, which then becomes the standard for future submissions. Through the use of a publishing approach such as executable papers, it can be ensured that the latest results for a published approach are always available.

The AtD paradigm provides us with the means to distribute data in the form of a constant data stream. In fact, real-time data analysis is emerging as the new "holy grail" of (big data) analytics [Marz and Warren 2015], as an increasing number of use cases emerge that deal with streamed data. Given the fluctuating nature of streamed data, evaluation of computational methods that analyse such data can lack reproducibility [Ferro et al. 2016]. EaaS initiatives that incorporate streamed data can address this by introducing version management controls that allow to reconstruct data at a particular time and version.

The EaaS setup also allows research that is not easily possible today for academic researchers. An example is the development of new performance metrics better adapted to modelling specific tasks. With such studies currently being dominated by industry-based researchers (see e.g., Lalmas et al.'s work on user engagement metrics at Yahoo! [Lalmas and Hong 2018]), EaaS allows researchers developing new metrics to experiment at a large scale on how the use of a new performance metric affects the ranking of submitted approaches in comparison to existing metrics.

2.2.4 Cooperation. EaaS could also lead to new ways to encourage cooperation, either between multi-disciplinary teams, or also between academia and industry. For tasks requiring a collection of

complementary skills to solve them, the insight provided by EaaS results into how different types of approaches perform could assist in effective team formation. This also means that participating teams have the possibility to concentrate on the aspect of the task for which they feel most qualified, and collaborate with other participants (either explicitly or through re-use of their approaches) to cover those aspects of the task for which they have lower expertise.

The demand for further cooperation between academia and industry is increasingly being pushed by major research funding agencies. For example, the European Commission's most recent research funding framework[8], Horizon 2020, emphasizes the need to support innovation, and to deliver solutions to end users. Similar funding schemes also exist on national levels, e.g., the Knowledge Transfer Partnership[9] (KTP) scheme in the United Kingdom or the ZIM[10] scheme in Germany.

3 EXISTING EAAS INITIATIVES

After discussing the benefits of the EaaS paradigm for the evaluation in the computational sciences in the previous section, this section provides further details on currently existing initiatives that have employed this paradigm. As it will become apparent, these initiatives have been established independently of each other and address different use cases. In order to ease comparison of these campaigns, Section 3.1 first highlights key features of these initiatives. A detailed introduction of the use cases of these initiatives is then provided in Section 3.2. Section 3.3 discusses some of the technologies used to facilitate EaaS for these initiatives. Finally, shortcomings are discussed in Section 3.4.

3.1 Key Features

Given the different demands and requirements posed by the individual EaaS use cases, it comes with no surprise that organizers of these initiatives have opted for different solutions to run their campaigns. Consequently, the EaaS initiatives that are introduced in the next section differ significantly from each other. Despite their differences, each initiative had to come up with solutions for the following core characteristics of EaaS. In the remainder of this section, we briefly introduce these key aspects. A more detailed discussion of these challenges is provided in Section 4.1.

First of all, all initiatives require a Management System that serves as the technical backbone of the individual campaigns. The range goes from fully open sourced systems to proprietary closed systems. Moreover, different levels of implementation efforts are required for participants to join the individual EaaS initiatives. In some cases, participants have complete freedom of choice on what programming language or platform they want to use. Other initiatives provide plugins in different languages that participants can use.

Different methods on how participants can get access to the data are employed. Possibilities include downloading the data using the traditional DtA approach, or methods following the AtD paradigm that requires users interacting with an API, or accessing data stored on the cloud via a VM or a Docker[11] container on the cloud. The type of access provided also has a direct impact on where the participants' solutions are executed. While some initiatives allow participants to run their algorithms on their local machines, others require them to run them in virtualized environments, or let the organizers execute it. The execution type also has a direct impact on how participants have to submit their solutions or results. Possibilities are by uploading result files in a specified format, by interacting with an API, or by submitting code installed on a VM.

[8] http://ec.europa.eu/research/
[9] http://ktp.innovateuk.org/
[10] https://www.zim-bmwi.de/
[11] Docker is an open source tool that can package software and its dependencies in a virtual container. For further details, we refer to [Matthias 2015].

A key question is also how the results are evaluated. Some initiatives allow for continuous evaluation with results submitted at any time. Others, however, define a fixed deadline for result submission. Related to this, all EaaS initiatives provide some sort of feedback that allows their participants to interact with the evaluation results, e.g. visual analytics functionalities for result comparison.

Finally, one of the main requirements for the operation of successful software initiatives is the provision of technical supports. The EaaS initiatives provide technical support to different degrees.

3.2 Initiative Use Cases

In what follows, more information on the existing initiatives is given.

3.2.1 TREC Microblog Task. The TREC Microblog tracks began in 2011 to explore search tasks and evaluation methodologies for information seeking behaviors in microblogging environments such as Twitter. TREC 2015 marked the fifth iteration of the track. For the last four years, the core task had been temporally-anchored ad hoc retrieval, where the putative user model was as follows: "At time T, give me the most relevant tweets about an information need expressed as query Q." Since its inception, the track has had to contend with challenges related to data distribution, since Twitter's terms of service prohibit redistribution of tweets. For TREC 2011 and 2012 [Ounis et al. 2011], the track organizers devised a solution following the DtA paradigm whereby the *ids* of the tweets were distributed, rather than the tweets themselves. Given these ids and a downloader program (also developed by the track organizers), a participant could "recreate" the collection [McCreadie et al. 2012]. This approach adequately addressed the no-redistribution issue, but was not scalable as participants in the end had to recreate the collection locally. TREC 2013 [Lin and Efron 2013] implemented an entirely different solution, which was to provide an API through which participants could complete the evaluation task. Employing the AtD paradigm, the organizers gathered a collection of tweets centrally, but all access to the collection was mediated through the API, such that the participants could not directly interact with the raw collection. The search API itself was built using Thrift[12] and the Lucene search engine[13], which are both widely-adopted open-source tools. A nice side-effect of the API approach is that common infrastructure promotes reproducibility [Rao et al. 2015] and sharing of open-source software components. An overview of the underlying technology, the TREC Total Recall Management System, is given in Section 3.3.5.

3.2.2 BioASQ. The FP7 BioASQ project[14] aimed to push research towards highly precise biomedical information access systems by establishing a series of evaluation campaigns in which systems from teams around the world compete [Tsatsaronis et al. 2015]. One of the main motivations for these campaigns was to ensures that the biomedical experts of the future can rely on software tools to identify, process and present the fragments of the huge space of biomedical resources that address their personal questions. Two main tasks were offered by BioASQ. In Task A systems were required to automatically assign MeSH (Medical Subject Headings) terms to biomedical articles, thus assisting the indexing of biomedical literature. This task used the EaaS paradigm to include participating systems directly in the indexing process of the National Library of Medicine (NLM) — Systems participating in the task were given newly published MEDLINE articles, before the NLM curators had assigned MeSH terms to them. The systems assigned MeSH terms to the documents, which were then compared against the terms assigned by the NLM curators. Task B focused on

[12] http://thrift.apache.org/
[13] http://lucene.apache.org/
[14] http://bioasq.org/

obtaining precise and comprehensible answers to biomedical questions. The systems that participated in Task B were given English questions written by biomedical experts that reflected real-life information needs. For each question, the systems were required to return relevant articles, snippets of the articles, concepts from designated ontologies, RDF triples from Linked Life Data, an 'exact' answer (e.g., a disease or symptom), and a paragraph-sized summary answer [Tsatsaronis et al. 2015]. In 2016, a new task was introduced, namely the "Funding Information Extraction From Biomedical Literature", where the participants were asked to extract grant information of new PubMed documents, from full text available in PubMed Central. They had to respond to each test article with grant ids and grant agencies mentioned in the article's full text. Annotations from PubMed were used to evaluate the information extraction performance of participating systems. The data provided can be downloaded by the participants, who can run their softwares and upload their results. The evaluation in the first task is dynamic, as soon as the NLM curators publish annotated articles in PubMed.

3.2.3 HOBBIT. The H2020 project HOBBIT[15] aims at abolishing the barriers in the adoption and deployment of Big Linked Data, by providing companies with open benchmarking reports that allow them to assess the fitness of existing solutions for their purposes. Big Data is one of the key assets of the future [Manyika et al. 2011]. However, the cost and effort required for introducing Big Data technology in a value chain is significant. Achieving this goal demands: 1) the deployment of benchmarks on data that reflects reality within realistic settings, 2) the provision of corresponding industry-relevant key performance indicators and 3) the computation of comparable results on standardized hardware. HOBBIT aims to address these tasks by employing the EaaS paradigm to achieve the following goals: 1) define benchmarks for domains of industrial relevance in Europe that make use of Big Linked Data, 2) determine the key performance indicators for processing Big Linked Data by collaborating with stakeholders from industry and research, 3) create an open benchmarking platform to evaluate the performance of state-of-the-art systems on standardized hardware and 4) organize yearly evaluation campaigns, using the platform and the industry-defined KPIs. The participants are granted access to a VM where they can implement their approach. The evaluation is done by the challenge organizers. The HOBBIT management system is described further in Section 3.3.6.

3.2.4 VISCERAL. The FP7 project VISCERAL[16] organized a series of benchmarks on the processing of large-scale 3D radiology image data [Langs et al. 2012]. The tasks included the segmentation of organs in the images (VISCERAL Anatomy), the detection of lesions (VISCERAL Detection) and the retrieval of similar cases including images and semantic terms as queries (VISCERAL Retrieval). Implementing the AtD approach, VISCERAL Anatomy used a cloud-based evaluation approach, where all data are stored in the cloud. Upon registration, participants were assigned VMs on which to install their software and access to a storage container containing the training data. For the test phase, access to the VMs was blocked for the participants while the organizers ran the executables in the VMs on a different storage container containing the test data. The use of the cloud also facilitates the creation of ground truth. A *Gold Corpus* of manually segmented organs was created by radiologists, with the process managed by an Annotation Management System. This system sends tickets to radiologists hired as annotators with instructions on the organ to segment in a specific volume, and tracks the annotation progress, allowing the process to run efficiently. It also manages a quality control process by which the manual segmentations are controlled by other radiologists and corrections can be requested. The executables submitted by participants are also

[15] http://project-hobbit.eu/
[16] http://visceral.eu/

used in collaboration with the participants to run the algorithms on additional non-annotated data sets with the goal to use label fusion and create more ground truth for training by fusing the output of all participant approaches. The ground truth created in this way is called the *Silver Corpus* [Krenn et al. 2015]. It is not of the same quality as the Gold Corpus, but was shown to be better than the best participant submission.

3.2.5 C-BIBOP. The Cloud-based Image Biomarker Optimization Platform (C-BIBOP)[17] is being developed as a technical resource for the cancer research community in the United States to support the development and assessment of quantitative imaging biomarkers. Lesion segmentation is a critical step in the development and also the use of imaging biomarkers in cancer and the two are organized as challenges in the system. Another task that is organized as part of C-BIBOP requires the analysis of Magnetic Resonance Imaging (MRI) to identify biomarkers that best correlate with clinical outcomes. C-BIBOP is being developed and used to support reproducible science by allowing researchers to compare the performance of their image analysis algorithms that are co-located with large medical imaging data sets. The size of the data sets as well as the concerns about the sensitive nature of the data has highlighted the need for a cloud-based solution to run the data analysis. EaaS using Docker containers for the code allows the challenge organizers to customize the evaluation methods for the clinical questions being addressed. C-BIBOP is run using the CodaLab platform[18] that was adapted for its needs and to run Docker containers directly on the Azure cloud. CodaLab is introduced in Section 3.3.3.

3.2.6 NewsREEL. The News REcommendation Evaluation Lab (NewsREEL)[19] was a campaign-style evaluation lab that was first organized as a news recommender challenge held in conjunction with ACM RecSys 2013, and then from 2014 – 2017 as a campaign-style evaluation lab at CLEF. NewsREEL focused on the application scenario of plista GmbH[20], a company that provides a recommendation service for online publishers. Whenever a user requests an article from one of their customers' web portals, plista recommends in real-time similar articles that the user might be interested in. NewsREEL consists of two separate tasks that follow different evaluation paradigms.

Implementing the more traditional DtA paradigm, a static data set [Kille et al. 2013] was made available for download that consists of historic transactions between plista and their customers, i.e., the users' requests for articles on publishers' web portals, a list of provided recommendations, users' clicks on these recommendations, as well as content updates on these portals. Besides, the open source tool Idomaar [Scriminaci et al. 2016] was made available which allowed participants to "replay" this historic data set, hence creating an artificial data stream. Participating teams were expected to present their results at the annual CLEF conference.

Implementing the AtD paradigm, participants were asked to provide recommendations in real-time for actual users. For this, participants' solutions were embedded in plista's operational system. Whenever a user requests an article from one of their customers' web portals, plista recommends similar articles that the user might be interested in. In NewsREEL, plista outsourced this recommendation task for a selected subset of their customers to participants. The Open Recommendation Platform (ORP) [Brodt and Hopfgartner 2014], further introduced in Section 3.3.2, handled communication between the participants and plista by providing an API that allowed participants to gain access to the dynamic data stream. ORP was also used to measure the performance of participants' algorithms. The selected performance metric was the click-through rate, i.e., the ratio between the number of requested recommendations and the number of recommendations that users clicked

[17] http://cbibop.org/
[18] https://competitions.codalab.org/
[19] http://www.newsreelchallenge.org/
[20] http://www.plista.com/

on. While ORP allows for continuous evaluation, in the context of CLEF, various fixed evaluation periods, of several weeks duration each, were defined during which the performance of participants' algorithms were measured and compared to a baseline run [Hopfgartner et al. 2014].

3.2.7 CLEF Living Labs for Information Retrieval. Living Labs for Information Retrieval (LL4IR)[21] was an effort similar to NewsREEL. Organized as campaign-style evaluation lab at CLEF 2015, the activity focused on retrieval as opposed to recommendation. LL4IR provided a benchmarking platform where researchers could gain access to privileged commercial data (click and query logs) and could evaluate their ranking systems in a live setting, with real users, in their natural task environments. The lab focused on three specific use-cases that employ EaaS: product search (on an e-commerce site), local domain search (on a university website), and web search (through a major commercial web search engine). A key idea to removing the harsh requirement of providing rankings in real-time for query requests was to focus on head queries [Balog et al. 2014]. Participants could produce rankings for each query offline and upload these to the commercial provider. The commercial provider then interleaved a given participant's ranked list with their own ranking, and presented the user with the interleaved result list. Finally, feedback was made available to participants to facilitate improved offline ranking generation. Data exchange between live systems and participants was orchestrated by a web-based API.

3.2.8 PAN Shared Task Series on Digital Text Forensics. PAN is a network for the digital text forensics [Stamatatos et al. 2015].[22] It offers researchers and practitioners a forum to study technologies tailored to the analysis of text originality, authorship, and trustworthiness. Applications of these technologies are found within law enforcement, intelligence, but also marketing and potentially within information retrieval [Potthast et al. 2016a]. Given the sensitive nature of such technologies, and their ethical implications, it is important to study them as transparently as possible, so that the general public has a chance of following up on, and discussing their capabilities in order to make informed decisions about them. It must be conceded, though, that all of these technologies are still in their infancy, and despite the fact that they are used in practice, their fitness has recently been called into question by successful attempts at attacking them, flipping on average 50% of true positive decisions to false negative ones [Potthast et al. 2016b]. Therefore, PAN's main goal is to foster progress in this area by organizing evaluation tasks and creating benchmarks for selected tasks from this domain. An important goal of PAN over the past years has been to establish evaluation campaigns that are reproducible, so that future evaluations within and without PAN can be done in comparison to the state-of-the-art. To attain true reproducibility in a shared task competition, however, it is necessary to allow for exchanging all of its building blocks, including software, data, and performance measures at any time. Neither of them can be assumed fixed forever, so that once someone proposes, for instance, a new data set, it should be possible to re-evaluate all existing software on this new data set. This insight informed PAN's move to adopt the EaaS paradigm for all of its shared tasks since 2012. For a management system, PAN employs the open source TIRA[23] platform (see Section 3.3.1), where software, data sets, and performance measures can be deployed in the cloud, and where the software solving a given task can be remotely executed. Participants are asked to implement software and to deploy binaries within VMs hosted by TIRA, which are then submitted for virtualized execution and evaluation. The software binaries within the VMs are remote-controled by TIRA, feeding them the test data in a way so that participants do not gain

[21] http://living-labs.net/

[22] http://pan.webis.de/, "PAN" used to be an acronym for "Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection", but now it is just a name.

[23] http://www.tira.io/; "TIRA" is short for TIRA Integrated Research Architecture; it is available open source at http://github.com/tira-io/.

direct access, ensuring a blind evaluation. Participants only have access to training data which can be downloaded for local model formation. Evaluations are conducted within phases with strict deadlines, whereas results are returned in moderated form via TIRA's web front-ends.

3.2.9 CoNLL Shared Task. The Conference on Natural Language Learning (CoNLL)[24] organized by the ACL Special Interest Group on Natural Language Learning (SIGNLL)[25] has been an early adopter of shared task evaluations in the natural language processing community. Since 1999, the conference has organized annual shared tasks on various important problems of natural language learning as a regular part of the conference program. Some of the evaluation resources that have been developed for these shared tasks are widely used today [van Erp et al. 2016]. However, the software that has been developed for these shared tasks throughout the years has not been collected by the shared task organizers but remains with their participants. In time, the chances of being able to obtain certain pieces of software decrease rapidly, since the researchers responsible move on in their careers and may no longer be available. This has been recognized as a major limitation to the reproducibility of CoNLL's shared tasks, so that SIGNLL has decided to adopt the EaaS paradigm as of 2015, offering it to the organizers as means to improve reproducibility, which has been picked up ever since [Xue et al. 2015, 2016; Zeman et al. 2017].[26] For a management system, CoNLL employs the TIRA experimentation platform (see Section 3.3.1), where the evaluation data sets and performance measures for the shared task have been deployed, and participants have been invited to deploy their software into TIRA's VMs. The evaluation procedures are the same as for the aforementioned PAN initiative, so that we omit a detailed discussion here for brevity. At the time of writing, more than 100 teams have submitted software to the shared tasks, demonstrating the transition of CoNLL's shared task to the EaaS paradigm as implemented by TIRA did not cause participation rates to decrease. Moreover, it also showed that TIRA can scale as the latest shared task involved data sets in more than 40 languages and hardware requirements for individual participants up to 190 GB of RAM, 18 CPUs, and more than a terabyte of disk space [Zeman et al. 2017].

3.2.10 TREC Total Recall Track. The principal purpose of the Total Recall Track 2015 was to evaluate, through a controlled simulation, methods to achieve very high recall – as close as practicable to 100% – with a human assessor in the loop. Motivating application domains include legal eDiscovery [Grossman and Cormack 2014], systematic reviews for meta-analysis in evidence-based medicine [Lefebvre et al. 2008], and the creation of fully labeled test collections for information retrieval evaluation [Cormack and Lynam 2005]. A secondary – but important – purpose was to develop a sandboxed virtual test environment within which information retrieval systems may be tested while preventing the disclosure of sensitive test data to participants. At the same time, this EaaS test environment operates as a black box, affording the participants' confidence that their proprietary systems cannot easily be reverse engineered.

The task to be solved was:

> Given a topic description (like those used for ad-hoc and web tasks), identify the documents in a corpus, one at a time, such that, as nearly as possible, all relevant documents are identified before all non-relevant documents. Immediately after each document is identified, its ground-truth relevance or non-relevance is disclosed.

Data sets, topics, and automated relevance assessments were all provided by a Web server supplied by the Total Recall Track. Participants were required to implement either a fully automated or semi-automated ("manual") process to download the data sets and topics, and submit documents for

[24] http://ifarm.nl/signll/conll/
[25] http://ifarm.nl/signll/
[26] http://www.cs.brandeis.edu/~clp/conll15st/

assessment to the Web server, which rendered a relevance assessment for each submitted document in real time. Thus, participants were tasked with identifying documents for review, while the Web server simulated the role of a human-in-the-loop assessor. Rank-based and set-based evaluation measures were calculated based on the order in which documents were presented to the Web server for assessment, as well as the set of documents that had been presented to the Web server at the time the participant declared that a "reasonable" result had been achieved. Particular emphasis was placed on achieving high recall while reviewing the minimum possible number of documents.

3.2.11 MIREX. The Music Information Retrieval Evaluation eXchange (MIREX) is a community-based framework for the formal evaluation of Music Information Retrieval (MIR) systems and algorithms [Hu et al. 2015], which has been running annually since 2005. This evaluation campaign has the difficulty that distributing the music recordings on which the evaluation tasks are run is not permitted due to copyright. To compensate for this limitation, the AtD and EaaS paradigms are employed as participants are required to upload executable files that carry out the tasks that are evaluated, and these executables are run on a single repository of music files. An online submission system supports the submission process. Running the executables on the repository of music files is done manually by the principal organizer of MIREX — to facilitate this, participants must adhere to a specification for calling the executable, and must provide details on software/architecture dependencies and other configuration details.

3.2.12 NTCIR OpenLiveQ Task. The NTCIR-13 OpenLiveQ task is one of the core tasks organized within the NTCIR project, and aims to provide an open live test environment of Yahoo Japan Corporation's community question-answering service (*Yahoo! Chiebukuro*[27]) for question retrieval systems [Kato et al. 2017]. The task was simply defined as follows: given a query and a set of questions with their answers, return a ranked list of questions. Submitted runs were evaluated in two ways: offline evaluation using the DtA paradigm computed evaluation metrics for ad-hoc retrieval (e.g. normalized discounted cummulative gain and expected reciprocal rank) based on relevance judgment on questions, while online evaluation employing the AtD paradigm estimated the quality of each ranked list of questions based on clicks from real users. Participants of the OpenLiveQ task could submit multiple runs and obtain the offline evaluation result right after the submission. Only the best run from each participating team could proceed to the online evaluation. Interleaving was used in the OpenLiveQ online evaluation, in which the OpenLiveQ organizers merged multiple ranked lists from participating teams into a single ranked list, and recorded clicks on the interleaved result to compute the score of each run.

3.3 EaaS Technologies

This section contains descriptions of a selection of EaaS tools that are either available as a running service and have been used by multiple evaluation campaigns or challenges, or for which the source code has been made available.

3.3.1 TIRA. The TIRA web service implements a management system to support shared task organizers with accepting software submissions [Gollub et al. 2012]. TIRA automates this process so that it imposes as little overhead as possible on organizers and participants. The system has been in active use from the start: since 2012, TIRA is employed for the PAN shared task series on digital text forensics [Potthast et al. 2014], and as of 2015, TIRA hosts the annual shared task of the CoNLL conference [Xue et al. 2015, 2016; Zeman et al. 2017]. TIRA's technology stack relies primarily on a combination of low-level (LXC, Docker) and high-level (hypervisor) virtualization technology, server-side control software, and a Web front-end that allows for the remote management of shared

[27] https://chiebukuro.yahoo.co.jp/

tasks. TIRA distributes VMs across several TIRA hosts, which are remote-controlled by a master server. Every VM is accessible from the outside by participants via SSH and remote desktop, and both Linux and Windows are supported as guest operating systems. This allows for a variety of development environments, so that participants in a shared task can directly work as they usually would. TIRA employs AtD, i.e., it hosts the data sets used in a shared task, split into training data sets and test data sets. The former are publicly visible to participants, including ground truth data, whereas the latter are accessible only to participant software in a secure execution environment that protects the test data sets from leaking to participants. Before executing the software on a test data set, TIRA clones its VM into the secure execution environment, where Internet access is disabled. After the software successfully executed on the test data set, its output is copied, whereas the cloned virtual machine is deleted to prevent any potentially private files on its virtual hard disk from exiting the execution environment. In this way, participants in a shared task can execute their software on the shared task's test data sets, whereas its organizers need not worry about the data leaking. TIRA also enables the use of proprietary and sensitive data as evaluation data. Finally, TIRA hosts a special purpose virtual machine for each shared task, where the organizer deploys software for performance measurement. The output of participant software that was executed on a training data set or a test data set is fed directly into the performance measurement software at the click of a button. The results are displayed on a dedicated web page for the shared task on TIRA's web front-end.

3.3.2 Open Recommendation Platform (ORP) and Idomaar. As described in Section 3.2.6, NewsREEL allows for the evaluation of news recommender algorithms by either "replaying" a news recommendation situation using a static data set, or by providing access to a live data stream of user requests for news articles [Kille et al. 2017]. While the "replay" task can be seen as a traditional shared evaluation task that implements the DtA paradigm, the latter task implements the EaaS paradigm.

The key component for enabling EaaS is the proprietary Open Recommendation Platform (ORP) [Brodt and Hopfgartner 2014] that participants can reach via a Web front-end and an API. After registering an account on the platform, participants need to provide a server address (and port number) and activate their account. ORP then starts broadcasting item updates, event notifications and recommendation requests, hence sending a constant data stream [Kille et al. 2013] to participants for processing. Event notifications are the actual user interactions, i.e., user visits, referred to as impressions, to one of the news portals that rely on the plista service, or clicks to one of the recommended articles. The item updates include information about the creation of new pages on the content providers' server and it allows participants to provide content-based recommendations. Expected responses to the recommendation requests are related news articles from the same content provider, which are then provided as recommendations to the visitors of the page.

ORP is set up as a Web Service with a specified API that sends requests in the form of HTTP POST requests and uses JSON for data encoding. This allows participants to implement their algorithms using their preferred programming language and execute it locally using their own hardware. In order to simplify initial setup for participants, basic plugins in different programming languages were provided that allowed participants to get started quickly. Since the main requirement of this scenario is that recommendations have to be provided in near real-time, network latency caused by transferring data between plista's data center and the participants becomes an issue. Dependent on this latency, the amount of time remaining to compute recommendations can be reduced significantly. In order to avoid this time loss, participants could also deploy a VM in plista's data center, hence executing their code in a virtualized environment.

ORP provides a Web front-end that consists of five different tabs, namely dashboard, statistics, debugging, leaderboard, and documentation. The dashboard allows users to set up their server and activate individual algorithms. The statistics page visualizes the performance of the registered algorithms, hence providing visual feedback of the EaaS. The online leaderboard page shows the overall performance of all teams that were currently participating in the challenge. This required full automation of the evaluation procedure.

The "replay" task that implemented the DtA paradigm relied on the open source tool Idomaar[28] as management system. Idomaar is an evaluation framework that simulates live data streams, hence bringing offline evaluation closer to online A/B testing. In other words, It further supports measuring performances of stream-based and set-based recommendation algorithms with respect to precision-related and technical aspects [Scriminaci et al. 2016]. In the context of CLEF, Idomaar allowed to mimic the work flow of the EaaS scenario introduced above. In order to join the "replay" task, participants were first asked to download a historic data set of transactions as created by ORP and then employ Idomaar locally to "replay" this data stream. Given that the management system only took over the role of creating a simulated data stream, participants had the freedom to develop solutions using their preferred programming language.

3.3.3 CodaLab. The CodaLab platform[29] is an ongoing open source development project[30] with the goal of minimizing duplication of effort between research groups and making research truly reproducible. Sharing data sets and evaluation scripts reduces time spent on setting up experiments, and CodaLab also encourages the unrestricted sharing of algorithms between researchers, streamlining access to new tools and techniques. CodaLab is a cloud-based platform powered by Microsoft Azure, allowing researchers to write executable programs and make them available in CodaLab Worksheets[31] and via an online community created around sharing and execution of versioned components. The availability of such components enables the community to re-evaluate data and algorithms in future scenarios.

In addition to worksheets CodaLab also supports EaaS Competitions[32], in which a community of researchers evaluate a common data set using different algorithms. Competitions require each algorithm to submit output for evaluation – but at the discretion of the competitors, CodaLab may also be used to make the algorithms themselves available to other researchers and these may be made available for use in Worksheets. Competitions may therefore increase the range of algorithms available to other users of CodaLab.

Extensive documentation on how to participate in a competition or creating a competition can be found in the GitHub repository.[33] While the medical image analysis community were early adopters of CodaLab Competitions, the platform has been developed and embraced by the broader scientific community in machine learning, computer vision, human language technologies, to name a few, as can be seen on CodaLab Competitions.[34]

3.3.4 OpenML. OpenML[35] is a platform that allows machine learning researchers to share data, code and results (e.g., models, predictions, and evaluations) [Vanschoren et al. 2013]. The types of objects that OpenML currently handles are data, tasks, flows and runs. *Data* can be uploaded

[28] https://github.com/crowdrec/idomaar
[29] http://www.codalab.org/
[30] https://github.com/codalab/codalab-competitions/wiki/Project_About_CodaLab
[31] https://worksheets.codalab.org/
[32] https://competitions.codalab.org/
[33] https://github.com/codalab/codalab/wiki/
[34] https://competitions.codalab.org/competitions/
[35] http://www.openml.org/

to the platform or linked to by a URL. *Tasks* describe what should be done with a data set, and include additional information such as training/test splits and what needs to be returned. Tasks can be of various types such as machine learning, clustering and regression. *Flows* are algorithms, workflows, or scripts for solving tasks, and *Runs* are applications of flows on tasks. Runs contain all information necessary to make the experiment reproducible, including data, flows, and parameter settings. All objects are searchable on the OpenML platform.

3.3.5 TREC Total Recall Management System. The TREC 2015 Total Recall Track used three modes of participation: "Practice" participation, "At Home" participation, and "Sandbox" participation. Practice and At Home participations were done using the open Web: participants ran their own systems and connected to the Web server at a public address. Employing the DtA paradigm, the Practice collections were available for several weeks prior to the At Home collections; the At Home collections were available for official runs throughout July and August 2015 (and continue to be available for unofficial runs).

Sandbox runs were conducted entirely on a Web-isolated platform hosting the data collection, hence implementing the AtD paradigm. To participate in the Sandbox task, participants were required to encapsulate – as a VM – a fully autonomous solution that contacts the Web server and conducts the task without human intervention. The only feedback available to participants as part of EaaS consisted of summary evaluation measures showing the number of relevant documents identified, as a function of the total number of documents identified to the Web server for review.

To aid participants in the Practice, At Home, and Sandbox tasks, as well as to provide a baseline for comparison, a Baseline Model Implementation (BMI) was made available to participants.[36] BMI was run on all the collections, and summary results were supplied to participants for their own runs, as well as the BMI runs.

3.3.6 HOBBIT Platform. The HOBBIT evaluation platform is a distributed FAIR benchmarking platform for the Linked Data lifecycle. This means that the platform was designed to provide means to benchmark any step of the linked data lifecycle. This includes generation and acquisition, analytics and processing, storage and curation as well as visualization and services. Moreover, it is developed to ensure that benchmarking results can be found, accessed, integrated and reused easily (FAIR principles). It is the first distributed benchmarking platform for Linked big data. It is an open source evaluation platform that can be downloaded[37] and executed locally. In addition, an online instance of the platform[38] is provided for a) running public challenges and b) making sure that even people without the required infrastructure are able to run the benchmarks they are interested in. The HOBBIT benchmarking platform ensures that:

- The benchmarks are easy to use.
- New benchmarks can be easily created and added to the platform by third parties.
- The evaluation can be scaled up to large data sets and on distributed architectures.
- The publishing and analysis of the results of different systems can be carried out in a uniform manner across the different benchmarks.

The first version of the platform was released in February 2017. It offered the main features of an evaluation platform and has been further enhanced over time, leading to the second version that was released in February 2018. The latter focuses on the usability of the platform as well as its support for additional features that can be used by the benchmark implementations (e.g.,

[36] http://plg.uwaterloo.ca/~gvcormac/trecvm/
[37] https://github.com/hobbit-project/
[38] http://master.project-hobbit.eu/

shared volumes or hardware statistics). The execution is virtualized and the evaluation is performed partially automatically. The results are announced in the online leaderboard of the challenge.

3.4 Shortcomings

Besides all the advantages mentioned before, there are also a few entry barriers and problems for participants in competitions that employ the EaaS paradigm. In particular, we identified the following shortcomings:

- Participants need to reinstall the full software stack if VMs are used, so this is harder than running tools locally; even though using Docker. could reduce this problem, as software can be moved more easily between local machines and the evaluation infrastructure;
- If participants are of big and well known groups in the field then poor performance can hurt their reputation and for this reason some groups only participate in challenges that they feel have a very high chance to have very good results;
- Some software tools frequently used for research such as MatLab usually require access to license servers so if the evaluations are run in a totally closed environment this can mean that the software might not run properly or additional adaptations are necessary;
- Participants in VISCERAL mentioned to feel a loss of control if they do not feel the data (or have it locally) as this is what most people are used to and it allows to quickly check the data visually and the first results; this obviously does not scale to big data;
- Errors on the test data could be different from errors on the training data and this might only become visible after the end of a competition run and limit performance for potentially good techniques, which is also related to the control loss;
- Sustainability is a problem as well for cloud or local infrastructure and installing everything for one single run of a competition might not be worth the effort but if it remains reusable then the effort might be worthwhile. In fact, as a survey amongst participants of the NewsREEL campaign revealed, the initial efforts required to set-up participation in the campaign were identified as a burden, when compared to traditional shared evaluation tasks that follow the DtA paradigm. Moreover, updates to the API and the release of a new data set caused some frustration amongst participants since it required changes to their own source code [Lommatzsch et al. 2017]. Nevertheless, as the survey revealed, the majority of participants appreciated the advantages of NewsREEL as EaaS, since it allowed them to learn new skillsets. Concluding from this feedback, we argue that significant efforts should be made into setting up stable and consistent EaaS to guarantee a sustainable service.
- VMs may not be sufficiently powerful, or do not have specific hardware such as GPUs available that some participants need for their algorithms to run well. For very large data sets the software tools are increasingly adapted to specific hardware for efficiency reasons – this becomes a problem if standard hardware in the cloud is being used and emulating specific hardware does not always work well.

Also the campaign organizers have to deal with potential problems:

- When work is done on confidential data the organizers need to make all security provisions and they may be held responsible for any shortcoming in the security infrastructure;
- Manual feedback for participants is costly (mails etc.), but it can be necessary if problems occur that could not be visible on the training data to make sure that no participant feels disadvantaged in the evaluation, particularly if price money is involved;
- Legal questions can arise that are different from standard benchmarks, for example, when reusing the code of participants for other tasks, and it can mean to take risks for participants;

if companies want to make sure that their proprietary code is safe, then it is the organizer who needs to assure this;
- Participants in existing cloud-based campaigns left VMs running without doing anything in them, which causes costs and is difficult to prevent entirely.
- As funding often ends after projects it is necessary to keep data sets alive. Organizers of shared evaluation campaigns who use static data sets can hand over the costs of hosting the data to third parties. For example, in the multimedia domain, the ACM SIGMM community hosts datasets that were presented in the Open Dataset & Software track of the annual the ACM Multimedia Systems conference[39]. This cost-sharing model can not easily be employed for EaaS campaigns that use cloud computing facilities or dynamic datasets. Therefore, it is necessary to think about other financial models than project funding but something long term and sustainable that might likely be in the way of a public-private partnership;

There are risks that companies and funding agencies need to take into account:

- Losing reputation for companies if results are not good is a real risk and for this reason some possibilities to remove runs and the partners may be needed;
- If a funding agency supports an evaluation campaign, it is important that validity is assured, for example, by really taking the best and meaningful performance measures; also statistical power needs to be checked as otherwise there is a risk that the results will in the end not mean much and the best and worst groups are extremely close together. We acknowledge that this is the case for all evaluation campaigns, but argue that the risk is higher for campaigns implementing the EaaS paradigm since the organizers are in full control of the data sets and other infrastructure. The reason for this is that by keeping full control over the EaaS infrastructure, the chances for third parties to develop alternative evaluation schemes using the same data sets are very limited, if not even impossible.

4 ADVANCING EAAS

After having introduced evaluation initiatives that have implemented some aspects of EaaS in the previous section, this section now presents the next steps in advancing EaaS as mature evaluation paradigm for the computational sciences. Based on the analysis of existing initiatives, we first identify key technical aspects of EaaS in Section 4.1. Section 4.2 then reflects on efforts required to improve acceptance of EaaS. Section 4.3 then addresses regulatory aspects.

4.1 Technical Aspects

As shown in the previous section, there is more than one way to implement an EaaS platform and it is as of yet unclear which way is the best one or which of the ways that have been pursued so far will prevail. However, as the summary of all initiatives has shown, despite all these differences, the initiatives also share several common design choices, which we refer to as key technical aspects, that have impacted the implementation and operation of EaaS. As shown in Table 1, the differences can be categorized as follows: Management System, Implementation, Submission, Data Access, Execution, Evaluation, Result INteraction, and Automation. In the remainder of this section, we clarify these differentials further.

4.1.1 Management System. One of the main requirements for the operation of EaaS is the development of a software platform that supports the AtD and EaaS paradigms. The following three options have been used:

Open Source: Some EaaS systems use open source EaaS management systems.

[39] http://sigmm.org/

Table 1. Comparison of selected EaaS initiatives

Initiative	Management System	Implementation	Submission	Data Access	Execution	Evaluation	Result Interaction	Automation
TREC Microblog 2013–2014	Proprietary	Software	Run Output	API	Managed	Fixed Deadline	None	Little
BioASQ	Dedicated	Software	Run Output	Download	Local	Fixed Deadline	Online Leaderboard	Part
Hobbit	Dedicated	Modules	VM	Management System	Virtualized	Fixed Deadline & Continuous	Online Leaderboard	Part
VISCERAL Anatomy 1/2	Dedicated	Software	VM	Management Sytem	Virtualized	Fixed Deadline	None	Little
VISCERAL Anatomy 3	Dedicated	Software	VM	Management System	Virtualized	Continuous	Online Leaderboard	Part
C-BIBOP	Open Source	Software	Run Output, Source Code, VM (planned)	Download, API (planned)	Local	Fixed Deadline & Continuous	Online Leaderboard	Full
NewsREEL	Proprietary & Open Source	Services	API	Download & API	Local, Managed & Virtualized	Fixed Deadline & Continuous	Online Leaderboard & Web Front-End	Full
CLEF LL4IR	Open Source	Services	API	API	Managed	Fixed Deadline	None	Part
PAN	Open Source	Software	VM	Managment System	Virtualized	Fixed Deadline	Web Front-End	Full
CoNLL Shared Tas	Open Source	Software	VM	Management System	Virtualized	Fixed Deadline	Web Front-End	Full
TREC Total Recall	Open Source	Plugins	VM & Source Code	API & Download	Managed	Fixed Deadline	None	Part
MIREX	Dedicated	Software	Compiled Binary	Management System	Virtualized	Fixed Deadline	None	Little
NTCIR OpenLiveQ Task	Dedicated	Software	Run Output	Download	Local	Fixed Deadline	None	Little

Dedicated: Other EaaS initiatives have developed their own dedicated management systems which were developed for the particular EaaS task.

Proprietary: Some initiatives embedded propriety software solutions such as third party APIs in their EaaS management system.

While the use of *Open Source software* with all its advantages (e.g., transparency, potentially larger group of contributors) seems like the preferred option for initiatives that aim for transparency and impact, it cannot always be used as the basis of the management system. Sometimes, *dedicated* or *proprietary software* has to be used, e.g., when EaaS is offered in collaboration with a company that wants to guarantee that its internal procedures are kept confidential and therefore cannot release software.

4.1.2 Implementation. The implementation aspect refers to the programming efforts that are expected from participants. Basically, four alternatives can be distinguished:

Software: Participants are asked to implement their entire software themselves without any restrictions regarding programming languages used, its architecture, its components, or its interface. The only restriction is given by the format of the input data, and the expected format of its output as defined by the organizers. This is the default modus operandi of almost all shared task competitions to date, so that it serves as baseline for this aspect.

Plugins: Participants are presented with a fully-fledged piece of software that features a plugin architecture, where a plugin is supposed to solve the problem underlying the shared task. In this case, the programming language and the interfaces are pre-defined and cannot be chosen by participants. A plugin architecture also usually prescribes by which approaches a given problem can be solved, or at least limits the space of possible approaches at solving the shared task's underlying problem.

Modules: Participants are asked to implement a software module that is integrated in a given processing pipeline. The organizers of a shared task competition have to specify the software architecture of the pipeline up front, specifying the interfaces of all modules. Moreover, organizers need to provide baseline implementations of all modules up front. Participants may then choose which of the modules they wish to replace with their own implementations, but are at liberty to resort to the baseline implementations. Restrictions to programming languages may be avoided in this case if the module interface is, for example, a POSIX command line interface with pre-specified input and output formats for each module.

Services: Participants are asked to implement their software as a web service with a pre-specified API. In this case, no restrictions apply with regard to how the service is implemented internally, whereas implementing and hosting a web service creates an overhead for participants.

When considering *plugins*, *modules*, and *services* as alternatives to the default *software*, each one brings about its own pros and cons. In general, it can be said that all of these alternatives have negative side effects on participants and organizers alike, but choosing them may improve the long-term effects of a shared task competition significantly.

4.1.3 Submission. The submission aspect refers to what piece of data participants are supposed to submit to a shared task competition. Basically, five alternatives can be distinguished:

Run Output: Participants are asked to submit the output of their software when it is executed on a test data set supplied by the organizers. The output's format has to comply with a pre-specified format supplied by them. This is the default modus operandi of almost all shared task competitions to date, so that it serves as baseline for this aspect.

Source Code: Participants are asked to submit the source code of their software, whereas the software must comply with the input and output formats specified by the organizers. Furthermore, participants are asked to supply instructions as to how to get the source code compiled and running.

Compiled Binary: Participants are asked to submit compiled binaries of their software in the form of libraries or executables. Moreover, participants must supply instructions about dependent software as well as how the binary can be executed.

VM: Participants are asked to submit VMs or Docker instances. Moreover, participants must supply instructions on how to execute their program or follow instructions given by the organizers on how to execute the software.

API: Participants are asked to submit the output of their algorithms via an API. This is the preferred method when dynamic data sets are used and when continuous evaluation is implemented.

When considering the submission of *source code*, *compiled binaries*, and *VMs* as alternative to the default, submitting *run output*, the former obviously requires much more effort from organizers (e.g., ongoing, infrastructure, and person hours), whereas participants have little to no overhead besides providing documentation. *APIs*, on the other hand, might require least efforts from organizers, assuming that the EaaS infrastructure is running stable. However, it is just as obvious that submitting *source code* or at least *compiled binaries* has a significant benefit in improving the long-term effects of shared tasks in terms of repeatability, reproducibility, and sustainability.

4.1.4 Data Access. The data access aspect refers to how participants are granted access to the data sets. Three different options are employed:

Download: Allowing participants to download (parts of) the data, hence employing the DtA paradigm. This method is often employed to allow participants to train their models on historic data.

Management System: Participants are only granted access to the data via the EaaS management system. This is the preferred method when the data is of sensitive nature and hence cannot easily be shared, when the data is copyright protected, or when it is too large to be made available for download.

API: Participants can gain access to the data via a provided API. This allows to transfer data in form of a constant stream. Dependent on the aim of the EaaS task, streamed data might allow for continuous evaluation.

Having different means of providing access to data is the main advantage of EaaS. Allowing participants to *download* data has been the default approach for shared evaluation tasks and in fact, some EaaS initiatives still follow this method. It does come with limitations though as discussed before. Providing access to data through the *EaaS Management System* only can guarantee that data is protected. This solution requires submission of source code, compiled binaries, or VMs. It also requires *managed* or *virtualized execution* (see below). *APIs* provide the advantage of supporting different use cases such as data stream processing.

4.1.5 Execution. The execution aspect refers to where the participant software is executed in order to obtain its output for evaluation. Basically, three alternatives can be distinguished:

Local: Participants are asked to execute the software locally using their own hardware. For this purpose, organizers must provide test data sets to participants, typically without revealing the ground truth. This is the default modus operandi of almost all shared task competitions to date, so that it serves as baseline for this aspect.

Managed: Organizers execute the participant software on their own hardware. This presumes that participants submit either source code or a compiled binary. This way, organizers need not release test data sets to participants.

Virtualized: Participants are asked to deploy and execute their software in a VM provided by the organizers. In this case, it depends on whether the software execution can be handled

remotely by participants to decide whether the test data sets need be directly accessible to participants to execute their software, or not.

When considering the *managed* or *virtualized* execution of participant software as alternative to the default, *local* execution, there are negative side effects for both participants and organizers. Organizers face the problem of executing untrusted software, and they need to provide the infrastructure for a timely execution of submissions. Participants may prefer a *virtualized* execution over *managed* execution, since the former gives them more control over their software, and whether it works as advertized, whereas the latter has a high turn-around time for them. Again, a significant improvement for the repeatability, reproducibility, sustainability, and even efficiency may be attained when choosing one of the alternatives to *local* execution.

4.1.6 Evaluation. Most EaaS campaigns have been organized as track or lab of one of the major evaluation campaigns TREC, CLEF, or NTCIR where participants are expected to present their efforts at an annual conference. The approach of evaluating the performance of algorithms and methods is influenced by this procedure. The following two evaluation approaches have been employed:

Fixed Deadline: Participants are expected to submit their solutions or outputs at a fixed deadline. The overall performance will then be determined by the organizers. This is the preferred method for shared evaluation tasks.

Continuous: The performance of participants' solutions is continuously evaluated. This is used in particular when dealing with streamed data. By providing participants means to interact with the results, they can improve their approaches over time. Continuous evaluation usually requires a high degree of automation (see below).

In the context of shared evaluation tasks, *fixed deadlines* have been used for many years. The advantage is that it allows organizers to define a clear cut that marks the end of an evaluation period. On the downside, *fixed deadlines* can hinder research, especially when researchers are running out of time and either give up, or spend less time on improving their methods. *Continuous evaluation* can counter this negative effect as researchers can work following their own pace.

4.1.7 Result Interaction. In traditional shared evaluation tasks, participants are requested to submit the outputs of their methods to the organizers for evaluation who will then determine the overall performance of each run submitted. EaaS campaigns, on the other hand, can offer a richer experience for participants who would like to interact with the results of their approaches. Dependent on the level of automation, type of evaluation, and EaaS infrastructure, the following three methods have been deployed:

None: Some initiatives do not provide any direct facilities that allow participants to inspect the performance of their approaches. Following the traditional method of shared evaluation campaigns, participants are informed about their performance via email and overall results are presented at conferences (such as CLEF, TREC, or NTCIR).

Web Front-End: Results are presented to participants via the Web front-end of the EaaS management system. This might require a higher level of automation as results might be made available automatically once participants submitted their results, or shortly after the end of a fixed submission deadline.

Online Leaderboard: Participants can compare the performance of their solutions in relation to other participants of the EaaS campaign. If provided at the end of the evaluation period, the main purpose is to display the best performing approach. However, if the EaaS campaign supports continuous evaluation and displays results using an online leaderboard, this feedback mechanism can trigger participants' motivation to improve their algorithms.

Providing the possibility of interacting with the results via a *Web front-end* or *Online leaderboards* requires a higher level of automation, as results need to be automatically analyzed and returned to the participants within a short time period. If the EaaS inititative is organized as part of a larger conference (e.g., CLEF, TREC, or NTCIR), a negative impact of this increased interaction is that participants might lose the incentive to present their work at the conference.

4.1.8 Automation. To make any form of EaaS viable, one must depart from the default approaches to Implementation, Submission, and Execution. However, doing so incurs significant risks for organizers both in terms of time spent as well as driving away participants. The aforementioned initiatives have still taken these risks, and managed them by some means of automation. In general, assisting participants and organizers with automating as many of the aforementioned technical aspects as possible will increase the acceptance of EaaS, ideally reducing its overhead to a point at which going with the default options is attractive. Some aspects, however, cannot be automated, these only concern the organizers of a shared task competition: for example, a *plugin architecture* has to be built on a task-by-task basis, and only certain components of such an architecture may be shared across tasks. Nevertheless, the successes of the aforementioned initiatives in prototyping EaaS platforms suggest that, in time, the development of robust automated services to assist participants and organizers in adopting this paradigm will become available.

4.2 Acceptance Aspects

Besides mastering the technical challenges, one further key step for advancing EaaS is to communicate the benefits of the paradigm to the research community, funding agencies and companies and to overcome expressed or experienced concerns.

On the one hand this can be achieved by providing compelling incentives for participation. In fact, as the study of [Jenkins Jr. et al. 1998] suggests, incentives can have a positive effect on people's willingness to participate in research activities. Lowering the entry barriers would be an important first step, so something simpler than VMs or running code, but something easy to replicate from a local installation. Docker containers might be a solution but they need to become available for several platforms (Windows and MacOS potentially as well) and also in security models suitable to run them in a variety of contexts. Standardization across tools and data is another point that can lower entrance barriers [Acemoglu et al. 2012] and this could be a top-down problem as the diversity in research is high and bottom-up standardization might be harder to achieve. Standardization can be related to data formats, interfaces of tools and components and portability of the software containers. Further, to motivate participation it is also important to look at which type of participants a challenge aims at, for example those interested in prize money, publications or free T-shirts and then address the desired community very clearly.

On the other hand, acceptance can be achieved by countering the fears the various stakeholders of EaaS campaigns might have. Fears could, for example, center around issues related to increased costs, e.g., due to the need for long-term sustainability of infrastructures including data availability and computing power. As [Kitchin 2014] points out, Data science in particular has many challenges in terms of managing the data and keeping research data available. Addressing this will require public-private technology partnerships that need to be based both on academic and industrial needs and aim at the long term and not only a short period to be sure to benefit from the main advantages. For a discussion on benefits and challenges of public-private technology partnerships, we refer to [Stiglitz and Wallsten 1999]. Government and funding agencies can help by engaging in campaigns and infrastructures as they can have high benefit but need to motivate funded groups to not only make data available but engage in serious performance comparison and collaboration with other groups. Such support would be in line with already existing activities such as the US National

Institute of Standards and Technology's (NIST) organisation of the TREC conference for shared evaluation task of IR tasks. An overview of NIST's role in this process is provided by [Voorhees and Harman 2005]. In the long run such EaaS has to be integrated into the entire research process and similar to current initiatives for data sharing by funding agencies, clear financial incentives in the entire research process can help everyone involved. Such integration needs to be done on an international level beyond current national or regional funding bodies if possible.

Another strong part of acceptance is the creation of a community feeling that involves all partners from data and problem providers to participants and that takes their comments into account. As research (e.g., [Kim 2000]) suggests, communication and transparency is key to successful online community building. In addition, we argue that lowering entry barriers and increasing the collaboration can benefit all participants. Community-building also depends on trust that all challenges are objective and that no cheating is possible and each participant has the same chances. Also, for company participants that would like their code protected or data providers such as hospitals that need to assure that no data leaves an infrastructure, trust is an essential part. Such trust can likely only be built over time and with longer term experiences and this is what the systems should be optimized for.

In the end, for all partners the personal benefits are an important criterion and these need to be made visible and measured to increase motivation. This can be the case for both participants and providers of tasks and challenges. There should be local optimization on the researcher level but particularly much more global optimization.

4.3 Regulatory Aspects

This section discusses the regulatory aspects of EaaS from two viewpoints: legal considerations for effectively running EaaS competitions, and potential steps toward running EaaS sustainably.

4.3.1 Legal Considerations. Four groups of stakeholders were identified in the organisation of an EaaS competition:

Data Owner: The organisation that owns and provides the data to be used in the EaaS competition. This could be one organisation or a group of organisations for a more complex task, which can also include distributed data storage and execution.

Competition Organizer: The organisation or group of organisations that define the tasks to be solved on the data for the competition, specify the evaluation criteria and administer the EaaS competition.

Infrastructure Owner: The organisation or group of organisations providing the infrastructure for running the EaaS competition. There could be more than one organisation involved if, e.g., one organisation provides the infrastructure on which the EaaS competition is run, while another provides the software to administrate the EaaS competition.

Participant: The organisations or individual people participating in the EaaS competitions.

The following three levels of necessary legal regulation were identified for EaaS. For each of these levels, the stakeholders involved are mentioned.

Data: This includes aspects such as regulating the appropriate use of the data, ensuring consistency in the terms of data release, and certification of an infrastructure to host a specific data type. The *Data Owner* and *Infrastructure Owner* stakeholder groups are involved in this agreement.

Participation: This includes the rules for participation in a specific EaaS competition, regulating, e.g., withdrawal from participation and permitted channels of result publication (such as no use of results in advertisements). The *Competition organizer* and *Participant* stakeholder groups are involved in this agreement.

Coordinators: This regulates what the coordinators may and may not do, including the re-use of programs submitted by participants on further data. The *Competition organizer* and *Infrastructure Owner* stakeholder groups are involved in this agreement.

In order to facilitate the organisation of an EaaS competition, standardized templates for these three agreements would be useful. Optimally, it would be possible to automatically generate the agreements based on options selected on a website, although this is made more complex as different agreements would be needed for different jurisdictions. These agreements should also take into account various specific requirements by organisations, such as the possibility for a participating company to embargo results, specific data requirements of some government research laboratories, and foreseeing the use of Non-Disclosure Agreements in some cases. Even if these agreements do exist, there is the complication of the enforceability of participant agreements signed in other countries. A clear chain of liabilities will also have to be defined. Further, due to the 2018 EU General Data Protection Regulation (GDPR)[40] that grants users more control over their data, the need for transparent agreements increases even further when personal data is used [Chassang 2017].

In order to make the organisation of EaaS competitions as straightforward as possible and avoid extremely complex legal agreements, a set of guidelines covering the best case of organisation should be released. This would include suggestions such as the following:

- data needs to be released under conditions that allow it to be as broadly usable as possible;
- non-anonymous data should only be used on a secure infrastructrue, but this still involves some risk;
- the algorithm creator should agree to the broadest possible terms, in the best case an open source release (or at least making the code available), and allowing use of the submitted algorithms on further data at the discretion of the organizers.

4.3.2 Sustainability. EaaS has an additional cost beyond standard evaluation campaigns and competitions in that it needs an infrastructure on which to run the EaaS. It therefore needs to provide a clear return on investment for a company to organize such a competition. Two potential sources of return on investment are identified here:

Open Innovation: Through making challenging tasks available as competitions, companies can receive potential solutions to their challenges from a significantly larger number of experts than would be available within the company. For this to work, participants have to agree to conditions for the company to continue to use their work (e.g., in the participation agreement).

Access to Talent: Companies could hire the people providing the best solutions to the challenges, therefore getting access to the best matches in terms of skill. This could also be used by venture capitalists to identify talent to fast-track to a new incubator.

Due to the impact that EaaS can have on innovation, it would also make economic sense for an EaaS infrastructure to be supported by public funds, at least in an initial stage until a business model for running competitions on behalf of companies and other organisations can be put in place as a public-private partnership. Supporting such initiative is of particular importance for entities such as higher education institutions in the United Kingdom who are required to showcase societal impact of their work, e.g., by delivering research impact case studies for the Research Excellence Framework[41] assessment [Watermeyer 2016].

[40] https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en
[41] http://impact.ref.ac.uk/

5 CONCLUSIONS

Evaluation campaigns have advanced many scientific areas and fields and focused research also in economic areas via platforms such as Kaggle that proposes machine learning challenges. Several companies have managed to make a business out of these challenges and crowdsourcing. The machine learning development can bring benefit to many areas to obtain and use optimized solutions. The impact is important and has advantages for organizers of such challenges but also to participants and companies who can propose their research challenges.

Evaluation-as-a-Service was created due to problems with the typical challenge of distributing large test data sets, working with confidential data that cannot be shared, and real time data that cannot be packaged. Several approaches have been created over the past few years to respond to the shortcomings, and different solutions were developed that are compared in this paper. The paper was started at a workshop on EaaS [Hopfgartner et al. 2015] in March 2015 in Sierre, Switzerland, has evolved into a white paper [Hanbury et al. 2015], and now has become much more concrete with many aspects being detailed based on experiences.

EaaS has the potential to change the way scientific challenges are run and to integrate with other initiatives such as clouds in the scientific sector to create more efficient and effective research infrastructures in the future. Motivations are manifold, both for funding agencies, organizations proposing data and tasks for challenges but also challenge organizers and participants in terms of impact and best use of available funding. Funding agencies push towards reproducibility, open data and open source code, as this can make the scientific process better.

It is foreseen that EaaS will, once it is further developed, ensure the full reproducibility from citable data to executable papers and the possibility to run existing tools on new data directly to create strong baselines automatically and assess the best techniques. It will automate routine tasks and concentrate real effort on novelty and improving existing techniques. Common platforms should also foster experience sharing and comparison of components, something that has not always been successful in past challenges. It can be much easier with central data and all tools accessing this data on the same platform, as has been done in some very specific domains, for example with NITRC[42]. Docker containers have made such sharing of executables also much easier and more lightweight.

Big data and data science need new approaches to create a sustainable research infrastructure and we expect EaaS to be a central part of such an infrastructure. Particularly the ever-increasing amount of data created and analysed creates challenges that are not easy to resolve. This can also help to address current machine learning challenges around data bias, explainability of results and also the ontrol of algorithms.

Many challenges still need to be addressed but much experience has already been gained via the existing approaches and this creates a solid foundation for the next steps to build better data science infrastructures.

REFERENCES

Daron Acemoglu, Gino Gancia, and Fabrizio Zilibotti. Competing engines of growth: Innovation and standardization. *Journal of Economic Theory*, 147(2):570 – 601.e3, 2012.

Timothy G. Armstrong, Alistair Moffat, William Webber, and Justin Zobel. Improvements that don't add up: ad-hoc retrieval results since 1998. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 601–610. ACM, 2009. ISBN 978-1-60558-512-3. doi: http://doi.acm.org/10.1145/1645953.1646031.

Michael Arrington. AOL Proudly Releases Massive Amounts of Private Data, 2006. URL https://techcrunch.com/2006/08/06/aol-proudly-releases-massive-amounts-of-user-search-data/.

[42] https://www.nitrc.org/

Krisztian Balog, Liadh Kelly, and Anne Schuth. Head First: Living Labs for Ad-hoc Search Evaluation. In *CIKM'14: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1815–1818, 2014.

Torben Brodt and Frank Hopfgartner. Shedding Light on a Living Lab: The CLEF NEWSREEL Open Recommendation Platform. In *IIiX'14: Proceedings of Information Interaction in Context Conference*, pages 223–226. ACM, 08 2014.

Jamie Callan and Alistair Moffat. Panel on use of proprietary data. *SIGIR Forum*, 46(2), 2012.

Gauthier Chassang. The impact of the EU general data protection regulation on scientific research. *Ecancermedicalscience*, 11(709), January 2017.

Gordon V. Cormack and Thomas R. Lynam. Spam corpus creation for TREC. In *CEAS 2005 – The Second Conference on Email and Anti-Spam*, 2005.

Thomas G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, volume 1857 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2000.

Nicola Ferro and Gianmaria Silvello. CLEF 15th Birthday: What Can We Learn From Ad Hoc Retrieval? In Evangelos Kanoulas, Mihai Lupu, Paul Clough, Mark Sanderson, Mark Hall, Allan Hanbury, and Elaine Toms, editors, *Information Access Evaluation. Multilinguality, Multimodality, and Interaction*, volume 8685 of *Lecture Notes in Computer Science*, pages 31–43. Springer, 2014.

Nicola Ferro, Norbert Fuhr, Kalervo Järvelin, Noriko Kando, Matthias Lippold, and Justin Zobel. Increasing Reproducibility in IR: Findings from the Dagstuhl Seminar on "Reproducibility of Data-Oriented Experiments in e-Science". *SIGIR Forum*, 50(1):68–82, June 2016. ISSN 0163-5840. doi: 10.1145/2964797.2964808. URL http://doi.acm.org/10.1145/2964797.2964808.

Antonio Foncubierta-Rodríguez and Henning Müller. Ground Truth Generation in Medical Imaging: A Crowdsourcing Based Iterative Approach. In *Workshop on Crowdsourcing for Multimedia, ACM Multimedia*, oct 2012.

Juliana Freire and Claudio T. Silva. Making computations and publications reproducible with VisTrails. *Computing in Science & Engineering*, 14(4):18 –25, August 2012. ISSN 1521-9615.

Norbert Fuhr. Some Common Mistakes in IR Evaluation, And How They Can Be Avoided. *SIGIR Forum*, 51(3):32–41, 2017.

Tim Gollub, Benno Stein, and Steven Burrows. Ousting Ivory Tower Research: Towards a Web Framework for Providing Experiments as a Service. In *SIGIR'12*, pages 1125–1126. ACM, 2012. ISBN 978-1-4503-1472-5.

Maura R. Grossman and Gordon V. Cormack. Comments on "The Implications of Rule 26 (g) on the Use of Technology-Assisted Review". *Fed. Cts. L. Rev.*, 2014:285–285, 2014.

Allan Hanbury and Henning Müller. Automated component–level evaluation: Present and future. In *International Conference of the Cross–Language Evaluation Forum (CLEF)*, volume 6360 of *Lecture Notes in Computer Science (LNCS)*, pages 124–135. Springer, September 2010.

Allan Hanbury, Henning Müller, Georg Langs, Marc André Weber, Bjoern H. Menze, and Tomas Salas Fernandez. Bringing the algorithms to the data: cloud-based benchmarking for medical image analysis. In *CLEF'12: Proceedings of the 3rd International Conference of the CLEF Initiative*, pages 24–29. Springer Verlag, 2012.

Allan Hanbury, Henning Müller, Krisztian Balog, Torben Brodt, Gordon V. Cormack, Ivan Eggel, Tim Gollub, Frank Hopfgartner, Jayashree Kalpathy-Cramer, Noriko Kando, Anastasia Krithara, Jimmy J. Lin, Simon Mercer, and Martin Potthast. Evaluation-as-a-service: Overview and outlook. *CoRR*, abs/1512.07454, 2015. URL http://arxiv.org/abs/1512.07454.

Donna K. Harman. Evaluation issues in information retrieval. *Inf. Process. Manage.*, 28:439–440, 1992.

David Hawking. If SIGIR Had an Academic Track, What Would Be In It? In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 1077–1077, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3621-5. doi: 10.1145/2766462.2776784. URL http://doi.acm.org/10.1145/2766462.2776784.

Katja Hofmann, Lihong Li, and Filip Radlinski. Online evaluation for information retrieval. *Found. Trends Inf. Retr.*, 10(1):1–117, June 2016. ISSN 1554-0669.

Frank Hopfgartner, Benjamin Kille, Andreas Lommatzsch, Torben Brodt, and Tobias Heintz. Benchmarking News Recommendations in a Living Lab. In *CLEF'14: Proceedings of the 5th International Conference of the CLEF Initiative*, pages 250–267. Springer Verlag, 09 2014.

Frank Hopfgartner, Allan Hanbury, Henning Mueller, Noriko Kando, Simon Mercer, Jayashree Kalpathy-Cramer, Martin Potthast, Tim Gollub, Anastasia Krithara, Jimmy Lin, Krisztian Balog, and Ivan Eggel. Report on the Evaluation-as-a-Service (EaaS) Expert Workshop. *SIGIR Forum*, 49(1):57–65, 2015.

Frank Hopfgartner, Andreas Lommatzsch, Benjamin Kille, Martha Larson, Torben Brodt, Paolo Cremonesi, and Alexandros Karatzoglou. The potentials of recommender systems challenges for student learning. In *Proceedings of CiML'16: Challenges in Machine Learning: Gaming and Education*, 10 2016.

Xiao Hu, Jin Ha Lee, David Bainbridge, Kahyun Choi, Peter Organisciak, and J. Stephen Downie. The MIREX grand challenge: A framework of holistic user-experience evaluation in music information retrieval. *Journal of the Association for Information Science and Technology*, 68(1), 2015.

Bernardo Huberman. Big data deserve a bigger audience. *Nature*, 482, 2012.

Darrel C. Ince, Leslie Hatton, and John Graham-Cumming. The case for open computer programs. *Nature*, 482(7386): 485–488, February 2012.

G. Douglas Jenkins Jr., Atul Mitra, Nina Gupta, and Jason D. Shaw. Are financial incentives related to performance? A meta-analytic review of empirical research. *Journal of Applied Psychology*, 83(5):777–787, 1998.

Makoto P. Kato, Takehiro Yamamoto, Tomohiro Manabe, Akiomi Nishida, and Sumio Fujita. Overview of the NTCIR-13 OpenLiveQ Task. In *The 13th NTCIR Conference*, 2017.

Benjamin Kille, Frank Hopfgartner, Torben Brodt, and Tobias Heintz. The plista dataset. In *NRS'13: Proceedings of the International Workshop and Challenge on News Recommender Systems*, pages 14–21. ACM, 10 2013.

Benjamin Kille, Andreas Lommatzsch, Frank Hopfgartner, Martha Larson, and Arjen P. de Vries. A stream-based resource for multi-dimensional evaluation of recommender algorithms. In *SIGIR 2017*, pages 1257–1260, 2017.

Amy Jo Kim. *Community Building on the Web: Secret Strategies for Successful Online Communities*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 2000. ISBN 0201874849.

Rob Kitchin. *The Data Revolution: Big Data, Open Data, Data Infrastructures and Their Consequences*. Sage, 2014.

Ron Kohavi, Alex Deng, Brian Frasca, Roger Longbotham, Toby Walker, and Ya Xu. Trustworthy online controlled experiments: Five puzzling outcomes explained. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 786–794. ACM, 2012.

Markus Krenn, Matthias Dorfer, Oscar Alfonso Jimenez del Toro, Henning Müller, Bjoern Menze, Marc-Andre Weber, Allan Hanbury, and Georg Langs. Creating a large-scale silver corpus from multiple algorithmic segmentations. In *Medical Computer vision workshop 2015 at MICCAI*, volume 9059 of *LNCS*. Springer, Munich, Germany, 2015.

Udo Kruschwitz and Charlie Hall. Searching the enterprise. *Foundations and Trends in Information Retrieval*, 11(1), 2017.

Takuya Kudo. Creating an age where anyone can find the information they truly need: NTCIR's information retrieval ideal. *NII Today*, (34):4–7, 2010.

Mounia Lalmas and Liangjie Hong. Tutorial on metrics of user engagement: Applications to news, search and e-commerce. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, pages 781–782, 2018.

Georg Langs, Henning Müller, Bjoern H. Menze, and Allan Hanbury. VISCERAL: Towards Large Data in Medical Imaging — Challenges and Directions. In *MCBR–CDS'12: Proceedings of the Third MICCAI International Workshop*, pages 92–98. Springer, 2012.

Carol Lefebvre, Eric Manheimer, and Julie Glanville. Searching for studies. *Cochrane handbook for systematic reviews of interventions*, pages 95–150, 2008.

Mark Levy. Offline evaluation of recommender systems: All pain and no gain? In *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation*, RepSys '13. ACM, 2013.

David E. Polley Light, Robert P. and Katy Börner. Open data and open code for big science of science studies. In *Proceedings of International Society of Scientometrics and Informetrics Conference*, pages 1342–1356, 2013.

Jimmy Lin and Miles Efron. Overview of the TREC-2013 Microblog Track. In *TREC'13: Proceedings of the 22nd Text REtrieval Conference*, Gaithersburg, Maryland, 2013.

Andreas Lommatzsch. Real-time news recommendation using context-aware ensembles. In *Advances in Information Retrieval - 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings*, pages 51–62, 2014.

Andreas Lommatzsch, Benjamin Kille, Frank Hopfgartner, Martha Larson, Torben Brodt, Jonas Seiler, and Özlem Özgöbek. CLEF 2017 NewsREEL Overview: A Stream-Based Recommender Task for Evaluation and Education. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 239–254, Cham, 2017. Springer International Publishing.

James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Hung Byres. Big Data: The next frontier for innovation, competition, and productivity. Technical report, 2011.

John Markoff. Troves of personal data, forbidden to researchers. *The New York Times*, 21 May 2012.

Nathan Marz and James Warren. *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*. Manning Publications Co., Greenwich, CT, USA, 1st edition, 2015. ISBN 1617290343, 9781617290343.

Karl Matthias. *Docker: Up and Running*. O'Reilly, 2015.

Richard McCreadie, Ian Soboroff, Jimmy Lin, Craig Macdonald, Iadh Ounis, and Dean McCullough. On Building a Reusable Twitter Corpus. In *SIGIR'12: Proceedings of the 35th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1113–1114, Portland, Oregon, 2012.

Henning Müller, Jayashree Kalpathy-Cramer, Allan Hanbury, Keyvan Farahani, Rinat Sergeev, Jin H. Paik, Arno Klein, Antonio Criminisi, Andrew Trister, Thea Norman, David Kennedy, Ganapati Srinivasa, Artem Mamonov, and Nina Preuss. Report on the cloud–based evaluation approaches workshop 2015. *ACM SIGIR Forum*, 51(1):35–41, 2016.

Virginia Ortiz-Repiso, Jane Greenberg, and Javier Calzada-Prado. A cross-institutional analysis of data-related curricula in information science programmes: A focused look at the ischools. *Journal of Information Science*, 2018. doi: 10.1177/0165551517748149.

Iadh Ounis, Craig Macdonald, Jimmy Lin, and Ian Soboroff. Overview of the TREC-2011 Microblog Track. In *TREC'11: Proceedings of the 20th Text REtrieval Conference*, Gaithersburg, Maryland, 2011.

Martin Potthast, Tim Gollub, Francisco Rangel, Paolo Rosso, Efstathios Stamatatos, and Benno Stein. Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In *CLEF'14: Proceedings of the 5th Int. Conference of the CLEF Initiative*, pages 268–299. Springer Verlag, 2014. ISBN 978-3-319-11381-4.

Martin Potthast, Sarah Braun, Tolga Buz, Fabian Duffhauss, Florian Friedrich, Jörg Marvin Gülzow, Jakob Köhler, Winfried Lötzsch, Fabian Müller, Maike Elisa Müller, Robert Paßmann, Bernhard Reinke, Lucas Rettenmeier, Thomas Rometsch, Timo Sommer, Michael Träger, Sebastian Wilhelm, Benno Stein, Efstathios Stamatatos, and Matthias Hagen. Who Wrote the Web? Revisiting Influential Author Identification Research Applicable to Information Retrieval. In *ECIR'16*, volume 9626, pages 393–407, Berlin Heidelberg New York, March 2016a. Springer.

Martin Potthast, Matthias Hagen, and Benno Stein. Author Obfuscation: Attacking the State of the Art in Authorship Verification. In *Working Notes Papers of the CLEF 2016 Evaluation Labs*, volume 1609 of *CEUR Workshop Proceedings*. CLEF and CEUR-WS.org, September 2016b.

Joaquin Quiñonero-Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché Buc, editors. *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment.* Number 3944 in LNAI. Springer, 2006.

Jinfeng Rao, Jimmy Lin, and Miles Efron. Reproducible Experiments on Lexical and Temporal Feedback for Tweet Search. In *ECIR'15: Proceedings of the 37th European Conference on Information Retrieval*, pages 755–767, Vienna, Austria, 2015.

Phyllis A. Richmond. Review of the Cranfield project. *American Documentation*, 14(4):307–311, 1963.

Charles Safran, Meryl Bloomrosen, W. Edward Hammond, Steven Labkoff, Suzanne Markel-Fox, Paul C. Tang, and Don E. Detmer. Toward a national framework for the secondary use of health data: An american medical informatics association white paper. *Journal of the American Medical Informatics Association*, 14(1):1–9, 2007.

Mario Scriminaci, Andreas Lommatzsch, Benjamin Kille, Frank Hopfgartner, Martha Larson, Davide Malagoli, András Serény, and Till Plumbaum. Idomaar: A framework for multi-dimensional benchmarking of recommender algorithms. In *Proceedings of the Poster Track of the 10th ACM Conference on Recommender Systems (RecSys 2016), Boston, USA, September 17, 2016.*, 2016.

Karen Spärck Jones and Cornelius Joost van Rijsbergen. Report on the need for and provision of an ideal information retrieval test collection. British Library Research and Development Report 5266, Computer Laboratory, University of Cambridge, 1975.

Efstathios Stamatatos, Martin Potthast, Francisco Rangel, Paolo Rosso, and Benno Stein. Overview of the PAN/CLEF 2015 Evaluation Lab. In Josiane Mothe, Jacques Savoy, Jaap Kamps, Karen Pinel-Sauvagnat, Gareth J.F. Jones, Eric SanJuan, Linda Cappellato, and Nicola Ferro, editors, *CLEF 2015*, pages 518–538, Berlin Heidelberg New York, September 2015. Springer. ISBN 978-3-319-24026-8. doi: http://dx.doi.org/10.1007/978-3-319-24027-5_49.

Joseph E. Stiglitz and Scott J. Wallsten. Public-Private Technology Partnerships: Promises and Pitfalls. *American Behavioral Scientist*, 43(1):52–73, 1999. doi: 10.1177/00027649921955155.

Victoria Stodden. The legal framework for reproducible scientific research: Licensing and copyright. *Computing in Science & Engineering*, 11(1):35 –40, February 2009. ISSN 1521-9615. doi: 10.1109/MCSE.2009.19.

Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Commun. ACM*, 59(2):64–73, January 2016. ISSN 0001-0782. doi: 10.1145/2812802.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):138, 2015.

Marieke van Erp, Pablo Mendes, Heiko Paulheim, Filip Ilievski, Julien Plu, Giuseppe Rizzo, and Joerg Waitelonis. Evaluating entity linking: An analysis of current benchmark datasets and a roadmap for doing a better job. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA). ISBN 978-2-9517408-9-1.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. OpenML: networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013.

Ellen M. Voorhees and Donna K. Harman, editors. *TREC: Experiment and Evaluation in Information Retrieval.* MIT Press, 2005.

Richard Watermeyer. Impact in the REF: issues and obstacles. *Studies in Higher Education*, 41(2):199–214, 2016. doi: 10.1080/03075079.2014.915303.

Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Rashmi Prasad, Christopher Bryant, and Attapol Rutherford. The CoNLL-2015 Shared Task on Shallow Discourse Parsing. In *Proceedings of the Nineteenth Conference on Computational Natural*

Language Learning - Shared Task, pages 1–16, Beijing, China, July 2015. Association for Computational Linguistics.

Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Attapol Rutherford, Bonnie Webber, Chuan Wang, and Hongmin Wang. CoNLL 2016 Shared Task on Multilingual Shallow Discourse Parsing. In *Proceedings of the CoNLL-16 shared task*, pages 1–19. Association for Computational Linguistics, 2016. doi: 10.18653/v1/K16-2001.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinkova, Jan Hajic jr., Jaroslava Hlavacova, Václava Kettnerová, Zdenka Uresova, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonca, Tatiana Lando, Rattima Nitisaroj, and Josie Li. CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19. Association for Computational Linguistics, August 2017. doi: http://doi.org/10.18653/v1/K17-3001.