Evaluation-as-a-Service: Overview and Outlook

ALLAN HANBURY, TU Wien
FRANK HOPFGARTNER, University of Glasgow
HENNING MÜLLER and IVAN EGGEL, University of Applied Sciences Western Switzerland
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. Also, crowdsourcing has changed the way that 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 overviewed, 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.

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.

1 INTRODUCTION

In areas of computer science such as Machine Learning and Information Retrieval, how well the developed algorithms function is measured by running the algorithms on data having associated ground truth showing the desired outcomes, and measuring the similarity of the algorithm outputs to the ground truth. Evaluations are often organised in a structured way in order to facilitate algorithm tests and make common data available on which algorithms can be evaluated. In the Information Retrieval area, there are regular cycles of 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]. In the area of machine learning, the series of PASCAL Challenges [Quiñonero-Candela et al. 2006] from 2005–2013 are well known. Similar initiatives exist in other domains such as computer vision and web science.

While the idea of running such evaluations is not new, the standard approach to running evaluations in the computational sciences involves distributing the data to the groups developing the systems so that they perform the computations locally and submit the results of the computations to the organisers for evaluation [Jones and van Rijsbergen 1975]. This is referred to as the *Data-to-Algorithms paradigm*. Participants submit the output of their software when run on a pre-published test dataset (a so-called "run"). This approach, however, has several shortcomings, including a complete lack of reproducibility of the shared task, and the necessity to publish test datasets prematurely, albeit sans ground truth. 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 [Hanbury and Müller 2010].

Evaluation-as-a-Service adopts the *Algorithms-to-Data paradigm*, in which the data are all stored on a (central) computational infrastructure, and participants can only access the data on this infrastructure [Hanbury et al. 2012]. This can also avoid any contact of the researchers with the test data in case of sensitive data as only the algorithms and not the developers need to see the data.

There are currently two main commercial platforms offering Data Science competitions: Kaggle[1] and TopCoder[2]. The Dream Challenges[3] offers competitions to solve biomedical challenges and have started migrating towards a cloud-based solution. All of these platforms currently use the Data-to-Algorithms paradigm, the current industry standard, by requiring the participants to download the data and submit result files. None of them offer an Algorithms-to-Data approach, which is currently the only approach that allows for scaling to Big Data and using sensitive data in competitions, such as medical data sets.

The Data-to-Algorithms approach of distributing data for challenges is in several cases 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. 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. The Twitter Terms of Service 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. Distribution of company documents for the evaluation of enterprise search are not permitted due to the commercial sensitivity of the data.
- 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 anonymised to allow distribution, as these results are too far removed from their operative requirements.

Even though these drawbacks are well known, all major organisers of evaluations currently adopt the Data-to-Algorithms approach. Kaggle, for example, specifically acknowledges this on their website by stating: "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."[4] Through the algorithms-to-data paradigm used by the Evaluation-as-a-Service, the necessity of moving data around is removed. The data safely remain on the servers of their owners or a trusted third party:

- The data does not need to be downloaded by the evaluation participants.
- It is not necessary for the data to be seen by the participants. In the case of private data, artificial data could be made available to the participants for training purposes, and the participant programs will only have access to the data when Virtual Machines (or other containers such as the lighter Docker containers) are submitted to the competition and the participants relinquish control of the Virtual Machines to the organisers.
- The data on the servers can be updated as regularly as needed, allowing experiments to be done on real-time data. Nevertheless, care must be taken in this case to make the competition fair by ensuring that entries are compared on the same data. As VMs including programs are submitted, the option exists to run the programs on multiple sets of data as the data evolves over time.

---

[1]http://www.kaggle.com/
[2]http://www.topcoder.com/
[3]http://dreamchallenges.org/
[4]https://www.kaggle.com/wiki/ANoteOnTorrents, visited on 28/08/2017

Fig. 1. Overview of important aspects of evaluation-as-a-service (EaaS). Aspects are grouped into five dimensions: technology, people, policy, research, and business. At the bottom of the graphic, the nine EaaS grassroot initiatives that were presented at the workshop are listed.

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]. Through preservation of data, programs, and results on a central infrastructure, EaaS should lead to improved reproducibility of research results.

A number of initiatives currently implement Evaluation-as-a-Service (EaaS), either making available APIs to access the data in a controlled way, or Virtual Machines (VMs) on which systems should be deployed. In order to organize these evaluation services, various aspects need to be considered. An overview of these aspects is given in Figure 1.

This paper is a revised version of the White Paper published as [Hanbury et al. 2015]. It begins by describing two success stories arising from evaluations that have used the EaaS paradigm. Then, an overview of existing EaaS initiatives is given in Section 3. The benefits of the EaaS paradigm and the shortcomings of EaaS in its current form are covered in Sections 4 and 5, respectively. The vision for a fully-developed EaaS approach is presented in Section 6. Finally, aspects to consider in the next steps toward achieving this vision are covered in Section 7.

2 EAAS SUCCESS STORIES

In this section, we detail two success stories arising from evaluations that have used the EaaS paradigm. In general the important impact of evaluations has been shown in [Rowe et al. 2010; Thornley et al. 2011; Tsikrika et al. 2013].

2.1 More efficient indexing at the National Library of Medicine

For the fifth year in a row, the National Library of Medicine (NLM) Medical Text Indexer (MTI) was used as one of the baselines for the international BioASQ Challenge. BioASQ is a series of challenges on biomedical semantic indexing and question answering with the aim of advancing the state of the art accessibility for researchers and clinicians to biomedical text (see Section 3.1.2 for more details).

The MTI indexing results are providing one of the baselines used in the "large-scale online biomedical semantic indexing" part of the challenge, which is designed to parallel the human indexing currently being done at NLM. The Medical Text Indexer (MTI) uses natural language processing technology and combines it with the expertise of indexers working at the NLM. The goal is to curate the biomedical literature with Medical Subject Headings (MeSH) more efficiently and consistently [Mork et al. 2014]. The BioASQ Challenge provided a continuous assessment of the indexing suggestions that are automatically generated by the MTI system used in support of the MEDLINE indexing process at the NLM.

The NLM has made an announcement on the significant benefits they have from their participation in the challenge. Specifically, as mentioned by the NLM[5], "the benefits of participating in this community-wide evaluation for MTI were twofold: firstly, MTI was rigorously compared to systems developed by a world-wide community of researchers and industrial teams all performing the same task; and secondly, the free exchange of the methods and ideas allowed the MTI team to incorporate the best practices explored by the participating teams. Incorporating some of these approaches into the MTI workflow in 2013–2014 improved the accuracy of MTI indexing suggestions by 4.5%".

2.2 Fostering Open Innovation through Living Labs

The key to the success of commercial information retrieval systems is their ability to efficiently assist users in satisfying their information need. Successful recommender systems have to suggest relevant items that might spark the users' interest. While setting up out-of-the-box retrieval or recommender systems is relatively easy, experience shows that adapting or fine-tuning algorithms to individual use cases requires significant resources.

More precisely, companies require talented developers, as well as knowledgeable engineers who have both a good overview of state-of-the-art techniques and the capability to develop innovative solutions to improve their systems' performance. This, however, can be rather challenging for companies, especially if they are small and do not have the required funds available. The idea of EaaS can serve here as a possibility to bridge this innovation blocker. In particular, it is a type of Open Innovation, defined as "Open Innovation is a paradigm that assumes that firms can and should use external ideas as well as internal ideas, and internal and external paths to market, as they look to advance their technology" [Chesbrough et al. 2006].

An example EaaS is implemented in NewsREEL, the first initiative that allows the evaluation of information access techniques in a living laboratory setting. NewsREEL addresses the challenge of news recommendation in real-time. The initiative was initially funded via a national knowledge transfer program that aims to foster collaboration between SME's and academia. Within the project, a commercial provider of news and ad recommendations developed a platform that allowed the

---

[5]https://www.nlm.nih.gov/news/indexer_challenge.html

academic project partner to develop recommendation algorithms and have them embedded in the live system of the provider. By providing statistics about the efficiency of the algorithms, the platform efficiently allowed the academic partner to perform A/B tests in a living lab. From the very beginning, this platform was open to anyone who wanted to develop and benchmark news recommendation algorithms. By organising NewsREEL as an academic challenge, participants were motivated to report their efforts at a scientific conference.

For the company that provided the infrastructure and API of NewsREEL, the advantages were manifold. First of all, it allowed them to raise awareness of their company and their use case. In fact, following the start of NewsREEL, the company received a number of job applications that directly referred to the evaluation campaign. Moreover, by interacting with the participants, they 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 academia, advantages are manifold as well. First of all, it allows researchers to gain experience in developing innovative techniques and have them tested under real live conditions. Moreover, it provides access to the infrastructure and large user base of a commercial service provider, hence reducing the gap between academic and industry research.

3 EXISTING EAAS INITIATIVES

This section begins by providing details on currently existing evaluation initiatives that use the Evaluation-as-a-Service paradigm. It then discusses some of the management systems used in facilitating Eaas. Finally, a comparison between the initiatives based on key characteristics of EaaS is provided.

3.1 Description of Initiatives

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

3.1.1 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 marks the fifth iteration of the track. For the past four years, the core task has been temporally-anchored ad hoc retrieval, where the putative user model is 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 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. That is, 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[6] and the Lucene search engine,[7] 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.
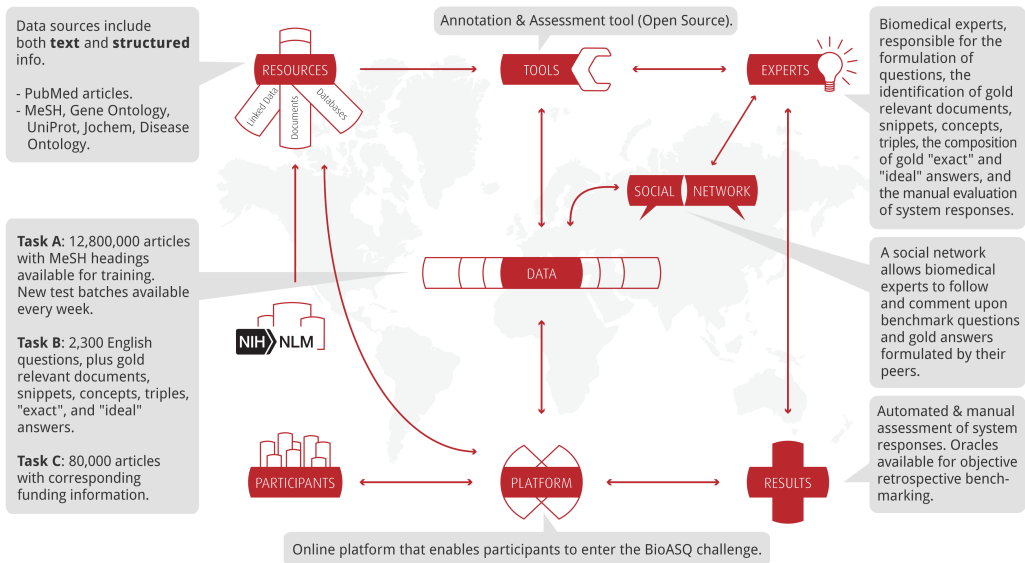
---

[6]http://thrift.apache.org/
[7]http://lucene.apache.org/

Data sources include both **text** and **structured** info.

- PubMed articles.
- MeSH, Gene Ontology, UniProt, Jochem, Disease Ontology.

Annotation & Assessment tool (Open Source).

Biomedical experts, responsible for the formulation of questions, the identification of gold relevant documents, snippets, concepts, triples, the composition of gold "exact" and "ideal" answers, and the manual evaluation of system responses.

Task A: 12,800,000 articles with MeSH headings available for training. New test batches available every week.

Task B: 2,300 English questions, plus gold relevant documents, snippets, concepts, triples, "exact", and "ideal" answers.

Task C: 80,000 articles with corresponding funding information.

A social network allows biomedical experts to follow and comment upon benchmark questions and gold answers formulated by their peers.

Automated & manual assessment of system responses. Oracles available for objective retrospective bench-marking.

RESOURCES · Linked Data · Documents · Databases

TOOLS

EXPERTS

SOCIAL NETWORK

DATA

NIH NLM

PARTICIPANTS

PLATFORM

RESULTS

Online platform that enables participants to enter the BioASQ challenge.

Fig. 2. Overview of the BioASQ ecosystem.

3.1.2 BioASQ. The FP7 BioASQ project[8] aims to push research towards highly precise biomedical information access systems by establishing a series of challenges in which systems from teams around the world compete [Tsatsaronis et al. 2015]. BioASQ provides data, software and the evaluation infrastructure for the challenge. By these means, the project 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. BioASQ comprises two main tasks. In Task A systems are required to automatically assign MeSH (Medical Subject Headings) terms to biomedical articles, thus assisting the indexing of biomedical literature. This task uses the EaaS approach to include participating systems directly in the indexing process of the National Library of Medicine (NLM) — Systems participating in the task are given newly published MEDLINE articles, before the NLM curators have assigned MeSH terms to them. The systems assign MeSH terms to the documents, which are then compared against the terms assigned by the NLM curators. Task B focuses on obtaining precise and comprehensible answers to biomedical questions. The systems that participate in Task B are given English questions written by biomedical experts that reflect real-life information needs. For each question, the systems are 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]. Since last year, a new task has been introduced, namely the "Funding Information Extraction From Biomedical Literature", where the participants are asked to extract grant information of new PubMed documents, from full text available in PubMed Central. They have to respond to each test article with grant ids and grant agencies mentioned in the article's full text. Annotations from PubMed are used to evaluate the information extraction performance of participating systems.
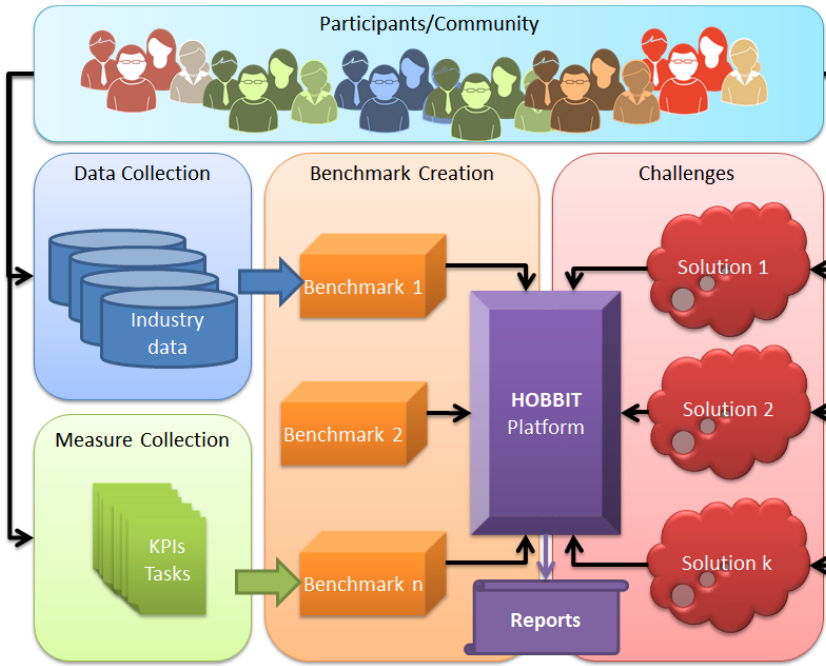
---

[8]http://bioasq.org/

Fig. 3. Overview of the HOBBIT architecture.

3.1.3 HOBBIT. Big Data is one of the key assets of the future. However, the cost and effort required for introducing Big Data technology in a value chain is significant. Mastering the creation of value from Big Data will enhance European competitiveness, will result in economic growth and jobs and will deliver societal benefit. The H2020 project HOBBIT[9] aims at abolishing the barriers in the adoption and deployment of Big Linked Data by European companies, by providing European companies with open benchmarking reports that allow them to assess the fitness of existing solutions for their purposes. 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 means of a strong team composed of leading research institutes, large industry customers and innovative small and medium-sized enterprises. In particular, the consortium aims 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.

3.1.4 VISCERAL. The FP7 project VISCERAL[10] organized a series of benchmarks on the processing of large-scale 3D radiology data [Langs et al. 2012]. The tasks include the segmentation of organs in the volumes, the detection of lesions and the retrieval of similar cases including images

---

[9]http://project-hobbit.eu/
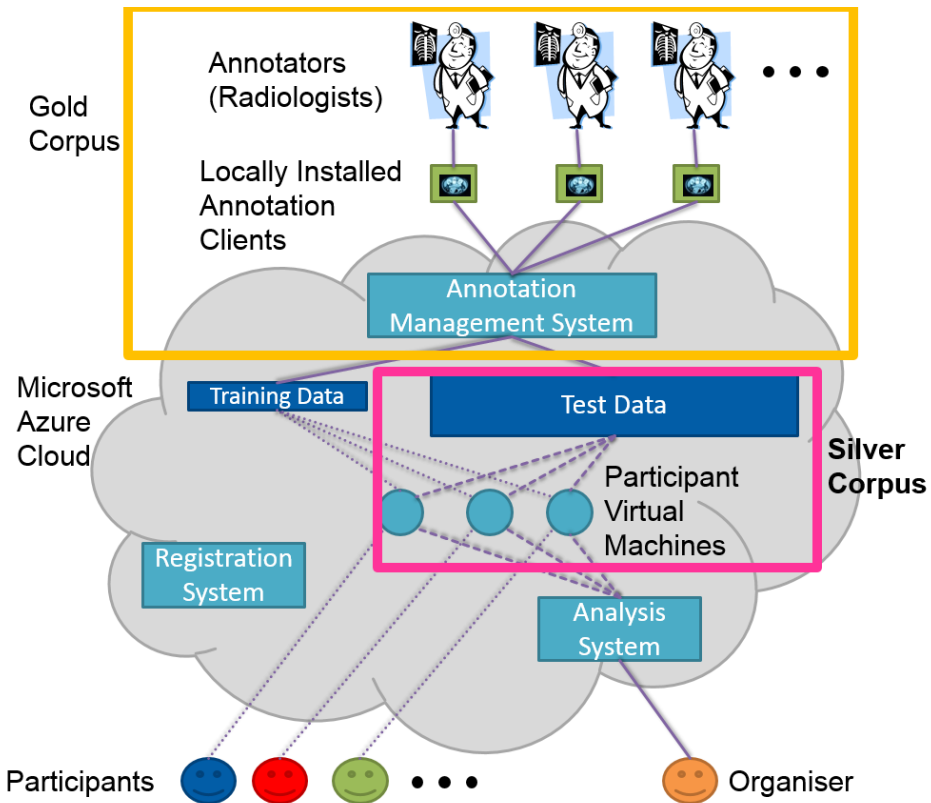[10]http://visceral.eu/

Fig. 4. Overview of the VISCERAL cloud-based approach.

and semantic terms as queries. VISCERAL used an innovative cloud-based evaluation approach, illustrated in Figure 4, where all data are stored in the cloud. Participants in the tasks obtain Virtual Machines (VMs) to install their software and access to training data via the cloud. For the test phase the virtual machines are blocked for the participants and the organizers take over the VMs and run the executables connecting the VM to a different storage with 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 directly in the cloud. This system sends tickets to radiologists hired as annotators with instructions on the organ to segment, 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. The executables submitted by participants are also 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] and it was shown to be better than the best participating algorithms.

3.1.5   C-BIBOP. The Cloud-based Image Biomarker Optimization Platform (C-BIBOP)[11] is being developed as a technical resource for the cancer research community in the United States to support
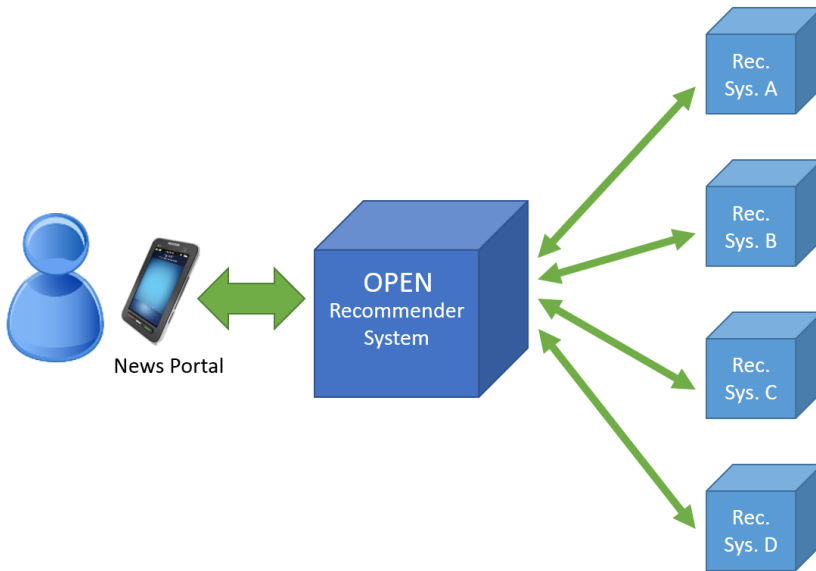
---

[11]http://cbibop.org/

Fig. 5. Overview of the CLEF NewsREEL approach.

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 datasets. The size of the datasets 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. Evaluation-as-a-Service 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[12] that was adapted for its needs.

3.1.6   NewsREEL. The News REcommendation Evaluation Lab (NewsREEL)[13] is 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. Apart from providing tools [Scriminaci et al. 2016] and data sets [Kille et al. 2013] to evaluate news recommendation in an offline scenario, it also implements the idea of *living laboratories* where researchers gain access to the resources of a company to evaluate different information access techniques using A/B testing [Hopfgartner et al. 2014]. The infrastructure is provided by plista GmbH, 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 similar articles that the user might be interested in. In NewsREEL, plista outsourced this recommendation task for a selected subset of their customers to interested researchers: Participants are asked to provide recommendations in real-time for actual users, i.e., the list of related articles is not determined by

---

[12]https://competitions.codalab.org/

[13]http://clef-newsreel.org/

plista, but by the participating research teams. The communication between the participants and plista, as well as the monitoring and evaluation is handled by the Open Recommendation Platform (ORP) [Brodt and Hopfgartner 2014] (see Section 3.2.3 for further details). ORP serves as a Web Service that is constantly sending users' requests for articles, as well as informs about new articles being added by the publisher, or existing articles being updated. The platform allows participants to register various recommendation algorithms in parallel and benchmark their performance over a longer period of time, as illustrated in Figure 5. In the context of CLEF, various evaluation periods, of several weeks duration each, were defined during which the performance of participants' algorithms were measured and compared to a baseline run. 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 on.

3.1.7   CLEF Living Labs for Information Retrieval. Living Labs for Information Retrieval (LL4IR)[14] is 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: 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.

3.1.8   PAN Shared Task Series on Digital Text Forensics. PAN is a network for the digital text forensics [Stamatatos et al. 2015].[15]. 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 shared tasks and creating benchmarks for selected tasks from this domain. An important goal of PAN over the past years has been to establish shared task competitions 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 dataset, it should be possible to re-evaluate all existing software on this new dataset. This insight informed PAN's move to adopt the Evaluation-as-a-Service

---

[14]http://living-labs.net/
[15]http://pan.webis.de/

paradigm for all of its shared tasks since 2012. PAN employs the TIRA experimentation platform (see Section 3.2.1), where software, datasets, and performance measures can be deployed in the cloud, and where the software solving a given task can be remotely executed. This way, PAN has assembled the largest collection to date of more than 150 pieces of digital text forensics software from researchers all over the world.

3.1.9   CoNLL Shared Task. The Conference on Natural Language Learning (CoNLL)[16] organized by the ACL Special Interest Group on Natural Language Learning (SIGNLL)[17] 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 have become standard benchmarks and are widely used today. 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 emerging Evaluation-as-a-Service 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].[18] CoNLL employs the TIRA experimentation platform (see Section 3.2.1), where the evaluation datasets and performance measures for the shared task have been deployed, and participants have been invited to deploy their software into TIRA's virtual machines. More than 100 teams have submitted software to the shared tasks, demonstrating the transition of CoNLL's shared task to the Evaluation-as-a-Service 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 datasets 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].

3.1.10   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, the test environment operates as a black box, affording 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.

---

[16]http://ifarm.nl/signll/conll/
[17]http://ifarm.nl/signll/
[18]http://www.cs.brandeis.edu/~clp/conll15st/

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 datasets and topics, and submit documents for 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.

3.1.11   MIREX. The Music Information Retrieval Evaluation eXchange (MIREX) is the community-based framework for the formal evaluation of Music Information Retrieval (MIR) systems and algorithms [Downie 2008], which has been running annualy 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, 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 organiser 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.

3.1.12   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*[19]) 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 computed evaluation metrics for ad-hoc retrieval (e.g. nDCG and ERR) based on relevance judgment on questions, while online evaluation 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.

3.2   EaaS Management Systems

This section contains descriptions of a selection of EaaS management software 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.

3.2.1   TIRA. The TIRA web service implements and experimentation platform to support shared task organizers with accepting software submissions [Gollub et al. 2012]. TIRA[20] automates this process so that it imposes as little overhead as possible on organizers and participants. TIRA 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

---

[19]https://chiebukuro.yahoo.co.jp/
[20]http://www.tira.io/

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 allow for the remote management of shared tasks. TIRA distributes virtual machines across a number of TIRA hosts, which are remote-controlled by a master server. Every virtual machine 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 further hosts the datasets used in a shared task, split into training datasets and test datasets. 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 datasets from leaking to participants. Before executing the software on a test dataset, TIRA clones its virtual machine into the secure execution environment, where Internet access is disabled. After the software successfully executed on the test dataset, 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 run their software on the shared task's test datasets, 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 dataset or a test dataset 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.

3.2.2   *VISCERAL Registration System.* The VISCERAL Registration System is the software used to manage the VISCERAL benchmarks. The code is available.[21] The system is currently closely tied to the Azure cloud. It supports participant registration, provision of benchmark documentation and files to participants, management of VM assignment, and almost fully automated VM submission and result evaluation for image segmentation benchmarks. The metrics calculated for each submission are available to the participant that submitted, and a participant can choose to make the results publicly visible on a leaderboard.

3.2.3   *Open Recommendation Platform (ORP).* 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]. The key component for the online setting that is introduced in Section 3.1.6 is the Open Recommender Platform (ORP) [Brodt and Hopfgartner 2014]. 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. 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. Requests are sent in the form of HTTP POST requests. JSON is used for data encoding.

One of the main requirements of this scenario is that recommendations have to be provided in almost real-time. Considering that a constant data stream [Kille et al. 2013] is exchanged between ORP and the participants' server, network latency becomes an actual issue since it reduces the

---

[21]https://github.com/Visceral-Project/registration-system

amount of time remaining to compute recommendations. In order to avoid this time loss, plista allows participants to run their algorithms on VMs in their data center.

ORP provides a Web interface 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. The leaderboard page shows the overall performance of all teams that are currently participating in the challenge.

3.2.4   CodaLab. The CodaLab platform[22] is an ongoing open source development project[23] with the goal of minimizing duplication of effort between research groups and making research truly reproducible. Sharing datasets 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 papers and make them available in CodaLab Worksheets[24] 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 Competitions[25], in which a community of researchers evaluate a common dataset 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 thmselves 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.[26] 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.[27]

3.2.5   OpenML. OpenML[28] 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 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.

3.2.6   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 participation was done using the open Web: participants ran their own systems and connected to the Web server at a public address. 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 availalbe for unofficial runs).

Sandbox runs were conducted entirely on a Web-isolated platform hosting the data collection. To participate in the Sandbox task, participants were required to encapsulate – as a VirtualBox virtual machine – a fully autonomous solution that would contact the Web server and conduct the task without human intervention. The only feedback available to participants 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.[29] BMI was run on all the collections, and summary results were supplied to participants for their own runs, as well as the BMI runs.

3.2.7 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, including generation and acquisition, analytics and processing, storage and curation as well as visualization and services, ensure that benchmarking results can be found, accessed, integrated and reused easily (FAIR principles), benchmark Big Data platforms by being the first distributed benchmarking platform for Linked data. It is an open source evaluation platform that can be downloaded[30] and executed locally. In addition, an online instance of the platform[31] 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 out to large datasets 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 has been released in February 2017. It offers the main features of an evaluation platform and has been further enhanced over time. The release of the second version is planned for February 2018. The latter will focus on the usability of the platform as well as its support for additional features that can be used by the benchmark implementations (e.g., shared volumes or hardware statistics).

3.3 Comparison between Initiatives

The EaaS initiatives are compared using the following characteristics (see Table 1 for an overview):

Software: If one of the available EaaS management systems is used, or if dedicated software was written for the initiative.

Data: If the data is a static collection or if the evaluation is run on dynamic (real-time) data.

Data Access: How the participants get access to the data. Possibilities include downloading the data, interacting with an API, or accessing data stored on the cloud via a Virtual Machine (VM) on the cloud.

Submission: How the results are submitted. Possibilities are by uploading result files in a specified format, by interacting with an API, or by submitting code installed on a VM.

---

[29]http://plg.uwaterloo.ca/~gvcormac/trecvm/
[30]https://github.com/hobbit-project/
[31]http://master.project-hobbit.eu/

Continuous: If the system allows for continuous evaluation with results submitted at any time, or if there is a fixed deadline for result submission.

Automation: The degree of automation of the result processing, where result processing includes collecting participant submissions, analysing them, and making the results available to participants. Possibilities are *little*, meaning that the organisers conduct the result processing almost completely manually, *partly*, meaning that significant parts of the result processing are automated, and *fully*, meaning that only very minor interactions are needed from the organisers (on the order of interacting with the system by clicking buttons to start processes).

Result Interaction: The type of interaction that the EaaS system allows with the submitted results, e.g. visual analytics functionalities for result comparison.

Technical Support: If and how technical support is provided.

4 BENEFITS OF EAAS

The goal of this section is to highlight what participants and other stakeholders, such as campaign organizers or companies crowdsourcing the technology development by proposing tasks, get as benefits and why they participate in these types of events. Funding agencies and science as a whole can get benefit from the EaaS paradigm 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. 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 datasets. Below we consider the benefits for participants, companies proposing tasks, organisers and science as a whole.

The participant's benefits usually relate to the following points:

- access to annotated data sets and a clear evaluation scenario, making it quick and easy to publish if the results are good;
- access to data sets that would be too big to be shared and that a single research group or a small group could not assemble and treat;
- access to sensitive data such as medical data but also in other domains (copyrighted music, enterprise search data). Without EaaS the companies would likely not share the data but maybe work on it in-house only, such as large search engine companies currently do with their log files;
- get a comparison to strong baselines, so other techniques and algorithms do not need to be reimplemented and then optimized; this has the reverse risk that it can make one's own results look less positive than comparing to a low baseline [Armstrong et al. 2009];
- get impact via publications, mainly by reusing the data after the end of competitions for further publications;
- if sharing of components is done, then this could also give more visibility, citations and reputation but this is currently not very often the case;
- advertisement via demos that are dissemination channels of own techniques;
- workshops to discuss with people working on the same data to get ideas on new approaches and avoid mistakes others have done but not published, as publications of negative results are rare;
- access to a broader range of challenges and testing own tools on the data best adapted for them;
- potentially better contacts to business partners if the challenges are proposed by a company for example, this could also lead to job offers for graduate students.

Table 1. Comparison of selected EaaS initiatives

Initiative	Software	Data	Data Access	Submission	Continuous	Automation	Result Interaction	Technical Support
TREC Microblog 2013-2014	Twitter Tools	Static	API	Result file upload	Fixed deadline	Little	None	Online forum
BioASQ	Dedicated	Static / Dynamic	Download	Result file upload	Fixed deadline	Part	Online leaderboard	Online forum
HOBBIT	Dedicated	Static / Dynamic	VM	VM	Fixed deadline and continuous	Part	Online leaderboard	Mailing list
VISCERAL Anatomy1/2	VISCERAL Registration System	Static	VM	VM	Fixed deadline	Little	None	Mailing list
VISCERAL Anatomy3	VISCERAL Registration System	Static	VM	VM	Continuous	Full	Online leaderboard	Mailing list
NewsREEL	Open Recommendation Platform	Static / Dynamic	API / Download	API & Report	Fixed deadline	Part	Online leaderboard	Tutorials & Mailing list
CLEF LL4IR	Living Labs API	Static	API / Download	API / Upload	Fixed deadline	Part	None	Mailing list
C-BIBOP	Codalab	Static	Download, API planned	Result file upload, code upload, Docker planned	Fixed deadline and continuous	Full	Online leaderboard	Online forum
PAN Evaluation Lab	TIRA	Static	VM	VM	Fixed deadline	Full	Web front end	Mailing list
CoNLL Shared Task 2015	TIRA	Static	VM	VM	Fixed deadline	Full	Web front end	Mailing list
TREC Total Recall Track	Baseline Model Implementation	Static	API / Download	VM / Script	Fixed deadline	Part	None	Online forum
MIREX	MIREX submission system	Static	None	Compiled Code	Fixed deadline	Little	None	Mailing List
NTCIR OpenLiveQ Task	Dedicated	Static	Download	Result file upload	Fixed deadline	Little	Online leaderboard	Mailing List

It is clear that the motivation of researchers can be intrinsic, so a student looking for a good scientific reputation through using good techniques. It can be extrinsic as well, for example winning prize money at a competition. For senior researchers, winning a competition can also lead to an

easier access to public funding for the techniques or also industry funding if winning an important competition.

The potential advantages of EaaS for a company that organizes tasks to crowdsource part of their innovation processes are:

- tapping into the skills of external data analytics experts to solve a data analytics problem for the company with a limited amount of funding available;
- obtaining a solution that is potentially much better than would have been produced internally as many more tools and algorithms can be tested and compared;
- identifying and getting access to new talent for hiring and already knowing about their skills and qualities beforehand;
- if a Virtual Machine containing software or a Docker container is submitted, the company can also test the software on other data (as long as this is specified in the participation agreement) and can test generalizability.

Organizers of evaluation campaigns also have several benefits:

- getting a possibility to shape the task of many scientists, so shaping research directions and getting influence in this;
- getting potentially many citations if data sets are reused and could become the standard for a specific sub field;
- getting the possibility to position themselves as leaders in the field if people use the data and scenarios provided;
- getting access to the best performing techniques and 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;

Science as a whole can benefit from such an approach in the following ways:

- efficiency of research will increase as data sets are reused. By working together larger data sets with more statistical power can potentially be obtained and work on creating small and potentially meaningless data sets is no longer necessary;
- community building around a problem can help everyone to get in contact with people working on similar problems and data and thus foster collaboration and collaborative problem solving, potentially also combining code and reusing existing tools that work well;
- reproducibility of papers is extremely high if data and evaluation scenarios are available in addition to executable code and all can be cited and rerun if necessary, also on new data or extended data sets;
- entry barriers to these domains for new persons such as PhD students are much lower if tools and data can be reused and all energy is spent on improving tools and new approaches based on strong baselines and not getting an evaluation scenario organized.

Many of these advantages are also strong incentives for funding organizations that can favor projects participating in such tasks or even organizing tasks or providing data. In addition to performance comparisons of algorithms, run time of algorithms can be measured if evaluated on the same infrastructure, so effectiveness and efficiency are used for ranking approaches.

Some of these points are definitely valid for all types of evaluation campaigns and data sharing but some are closely linked to the paradigm of EaaS, as the code is available and can be reused, for example, for creating a silver corpus on new data [Krenn et al. 2015]. Having running code in addition to data and published papers can also help to speed up the direct transfer of technology to industry.

5 SHORTCOMINGS OF EXISTING EAAS INITIATIVES

Besides all the advantages mentioned just before, there are also a few entry barriers and problems for participants in competitions, particularly if EaaS is used, so no simple download of data but the requirement to provide executables:

- participants need to reinstall the full software if virtual machines 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;
- 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 organisers 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 but is necessary to keep data sets alive and allow for evaluation if cloud computing is used it is also 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 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;
- in general, the concentration on competitions can limit the number of new techniques, as there is a higher incentive to use small modifications of existing techniques than to develop something really new with maybe a higher potential but also a higher risk that in the beginning the performance is limited.

6 EAAS VISION

Once it is fully developed, the EaaS paradigm should provide advantages for both academia and industry, with academia getting access to interesting data and challenges, and industry getting access to results that could improve the offerings to their clients. To illustrate the advantages obtained by academia and industry, we first present two scenarios illustrating how EaaS could be used once it is fully developed:

> **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 Virtual Machines, where the parameters such as maximum response time are strictly specified. Due 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. Upon submission of the VMs, standardised 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 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 that 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.

We now outline what we see as the main contributions of the full EaaS paradigm, in terms of benefits, simplicity, reproducibility and cooperation. We also consider the points of view of both academia and industry.

6.1 Benefits

In designing and implementing the EaaS paradigm, it should be ensured that the benefits significantly outweigh the effort invested both from academia and industry. One of the main benefits to both sides is that the EaaS paradigm should bridge the gap between academia and industry, 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 allows them access to industry data, 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. Furthermore, through making available data and associated challenges through the EaaS paradigm, companies get exposure to the latest processing and analysis approaches from academia. The EaaS paradigm also makes A/B testing [Kohavi et al. 2012] on real-time data available to researchers in academia, 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).

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 organisers 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 to 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.

Finally, EaaS instances can be used as part of university courses on Data Science 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 Coursera course. First efforts to establish the evaluation campaign NewsREEL as a resource for learning and teaching are outlined by Hopfgartner et al. [Hopfgartner et al. 2016].

6.2 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 organisers of EaaS instances, they should not be forced to set up the full infrastructure necessary for EaaS each time they organise 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.

6.3 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 [Callan and Moffat 2012]. The EaaS paradigm can contribute to increasing reproducibility of results through making available not only data and associated tasks over a 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, as every approach is evaluated on the same infrastructure.

Further contributions toward reproducibility can also be expected. As the dataset 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 EaaS setup would also allow research that is not easily possible today. An example is the development of new performance metrics better adapted to modelling specific tasks. Researchers developing new metrics would be able 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.

6.4 Cooperation

EaaS could also lead to new ways to encourage cooperation and form multi-disciplinary teams. 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.

7 ADVANCING EAAS

This section presents the next steps in advancing EaaS in terms of the technical, acceptance and regulatory aspects, which were identified as the key aspects of EaaS.

7.1 Technical Aspects

There is more than one way to implement an Evaluation-as-a-Service 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. We review design choices for Evaluation-as-a-Service platforms from the perspective of organizers and participants in a shared task competition, as well as with regard to reproducibility properties of the evaluation results obtained; both along the three technical aspects "Implementation," "Submission," and "Execution." Table 2 gives an overview.

7.1.1 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 (see Table 2).

The flexibility of participants is much worse with Plugins, whereas Modules, and particularly Services conserve some flexibility. Setting up Services require much more effort from participant than developing software in isolation, whereas Plugins and especially Modules require less effort, respectively. From the organizer point of view, up front effort and ongoing effort are worst when asking for Plugins, because the entire software has to be provided in a working condition to participants, who will argue a lot about details. That is less so, when asking for Modules or Services, since here only some up front effort is required, mostly specifying interfaces. Little to no extra infrastructure is required with either solution, disregarding software provided by organizers, but Plugins require significant person hours from organizers, whereas Modules less so, and Services even less.

Regarding evaluation results, their repeatability, reproducibility, sustainability, and efficiency will be much better when employing one of the alternatives to Software. There is one exception, though, namely Services: one cannot rely on participants of a shared task competition to host and maintain their services for a long time after the competition has passed. Therefore, this alternative is much less sustainable and its repeatability and reproducibility depends on whether a participant's service is still available. The service can be employed on a central infrastructure though, as well.

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

Run Output: Participants are asked to submit the output of their software when it is executed on a test dataset 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.

When considering the submission of Source Code and Compiled Binaries as alternative to the default, submitting Run Output, the former obviously requires much more effort from organizers (e.g., ongoing, infrastructure, and man hours), whereas participants have little to no overhead besides providing documentation (see Table 2). 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.

Table 2. Comparison of alternative implementation approaches for the Evaluation as a Service paradigm. We distinguish implementation from submission and execution of participants software. The traditional approach to organizing shared task competitions serves as a baseline in each case (top row of each category). The alternative approaches are judged whether they perform much worse (– – –), worse (– –), a little worse (–), similar (○), a little better (+), better (+ +), or much better (+ + +) compared to their respective baseline with respect to 10 criteria (columns). For some criteria, an approach performs better or worse (+ –), dependent on the circumstances.

EaaS aspect	Participant		Organizational effort				Evaluation result			
	flexibility	effort	up front	ongoing	infrastructure	man hours	repetition	reproduction	sustainment	efficiency
Implementation										
Software	○	○	○	○	○	○	○	○	○	○
Plugins	– – –	– –	– – –	– –	–	– – –	+ + +	+ + +	+ + +	+ + +
Modules	– –	–	– –	–	– –	– –	+ + +	+ +	+ + +	+ +
Services	–	– – –	– –	–	○	–	+ –	+ –	– – –	+
Submission										
Run output	○	○	○	○	○	○	○	○	○	○
Source code	○	– –	○	– – –	– – –	– – –	+ + +	+ + +	+ + +	○
Compiled bin.	○	–	○	– –	– – –	– –	+ + +	+ +	+ +	○
Execution										
Local	○	○	○	○	○	○	○	○	○	○
Managed	– – –	– –	–	– – –	– – –	– – –	+ + +	+ +	+ + +	+ + +
Virtualized	–	–	– – –	– –	– – –	– – –	+ + +	+ + +	+ + +	+ +

7.1.3 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 datasets 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 datasets to participants.

Virtualized: Participants are asked to deploy and execute their software in a virtual machine 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 datasets 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 advertised, 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.

7.1.4 Automation Avoids Negative Side Effects. To make any form of Evaluation-as-a-Service 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 Evaluation-as-a-Service, 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 Evaluation-as-a-Service platforms suggest that, in time, the development of robust automated services to assist participants and organizers in adopting this paradigm will become available.

7.2 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, on the other hand by countering the fears the various stakeholders of EaaS campaigns might have. Lowering the entry barriers would be an important first step, so something simpler than virtual machines 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.

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.

Sustainability of infrastructures including data availability and computing needs to be assured and not only linked to short-term projects. Likely this will require public-private 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.

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. Data science has many challenges in terms of managing the data and keeping research data available. This should likely rather highlight a collaboration aspect and not a pure competition that often does not offer incentives for collaborations.

Trust needs to be built as well 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.

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. Also the communication between participants can be fostered in this way, lowering entry barriers and increasing the collaboration that can benefit all participants.

Standardization across tools and data is another point that can help acceptance 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.

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.

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.

7.3 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.

7.3.1 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 Organiser: 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, and a diagram showing the relation between the stakeholders and the agreements is shown in Figure 6.

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 (e.g., HIPAA). 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 Organiser* 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 Organiser* and *Infrastructure Owner* stakeholder groups are involved in this agreement.

In order to facilitate the organisation of an EaaS competition, standardised 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
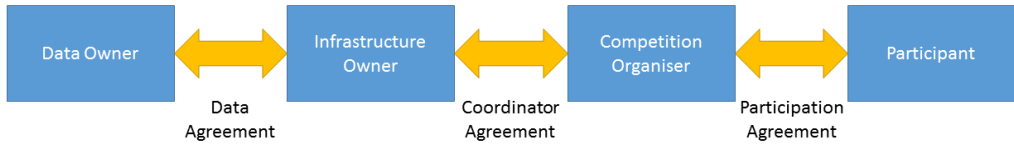
Fig. 6. The stakeholders and potential agreements necessary to organise an EaaS competition.

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.

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 organisers.

7.3.2 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 organise 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.

Examples of successes with the EaaS paradigm are given in Section 2 of this white paper. 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.

8 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 and crowdsourcing part of the machine learning development can benefit in 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 white paper.

The white paper was started at a workshop on EaaS in March 2015 in Sierre, Switzerland, but has evolved since then and become much more concrete with many aspects being detailed based on experiences.

EaaS has the potential to change the way 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.

It is foreseen that EaaS will, once it is further developed, ensure 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 make routine tasks automatic 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.[32]

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. 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.

REFERENCES

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.

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.

Henry Chesbrough, Wim Vanhaverbeke, and Joel West. *Open Innovation: Researching a New Paradigm*. Oxford University Press, 2006.

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.

J. Stephen Downie. The music information retrieval evaluation exchange (2005-2007): A window into music information retrieval research. *Acoustical Science and Technology*, 29(4):247–255, 2008.

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.

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.

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.

---

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

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.

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, 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.

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.

K. Sparck Jones and C.J. 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.

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.

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.

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.

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. New York: Wiley*, pages 95–150, 2008.

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.

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

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.

James Mork, Dina Demner-Fushman, Susan Schmidt, and Alan Aronson. Recent enhancements to the nlm medical text indexer. In *Working Notes for CLEF 2014 Conference*, volume 1180 of *CEUR*, pages 1328–1336, 2014.

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 Nicola Ferro, Fabio Crestani, Marie-Francine Moens, Josiane Mothe, Fabrizio Silvestri, Giorgio Maria Di Nunzio, Claudia Hauff, and Gianmaria Silvello, editors, *Advances in Information Retrieval. 38th European Conference on IR Research (ECIR 16)*, volume 9626 of *Lecture Notes in Computer Science*, pages 393–407, Berlin Heidelberg New York, March 2016a. Springer. doi: http://dx.doi.org/10.1007/978-3-319-30671-1_29.

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. URL http://ceur-ws.org/Vol-1609/.

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.

Brent R. Rowe, Dallas W. Wood, Albert N. Link, and Diglio A. Simoni. Economic impact assessment of NIST's text retrieval conference (TREC) program. Technical report project number 0211875, National Institute of Standars and Technology, 2010.

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.

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.

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.

Clare V. Thornley, Andrea C. Johnson, Alan F. Smeaton, and Hyowon Lee. The scholarly impact of TRECVid (2003–2009). *Journal of the American Society for Information Science and Technology*, 62(4):613–627, 2011.

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.

Theodora Tsikrika, Birger Larsen, Henning Müller, Stefan Endrullis, and Erhard Rahm. The scholarly impact of CLEF (2000–2009). In *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, pages 1–12. Springer, 2013.

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.

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. URL http://aclanthology.coli.uni-saarland.de/pdf/K/K16/K16-2001.pdf.

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. URL http://www.aclweb.org/anthology/K17-3001.