Shared Tasks as Tutorials: A Methodical Approach

Theresa Elstner[1], Frank Loebe[1], Yamen Ajjour[2], Christopher Akiki[1], Alexander Bondarenko[3], Maik Fröbe[3], Lukas Gienapp[1], Nikolay Kolyada[2], Janis Mohr[4], Stephan Sandfuchs[4], Matti Wiegmann[2], Jörg Frochte[4], Nicola Ferro[5], Sven Hofmann[1], Benno Stein[2], Matthias Hagen[3], Martin Potthast[1]

[1]Leipzig University, Germany
[2]Bauhaus-Universität Weimar, Germany
[3]Martin-Luther-Universität Halle-Wittenberg, Germany
[4]Bochum University of Applied Sciences, Germany
[5]University of Padua, Italy

Abstract

In this paper, we discuss the benefits and challenges of shared tasks as a teaching method. A shared task is a scientific event and a friendly competition to solve a research problem, the task. In terms of linking research and teaching, shared-task-based tutorials fulfill several faculty desires: they leverage students' interdisciplinary and heterogeneous skills, foster teamwork, and engage them in creative work that has the potential to produce original research contributions. Based on ten information retrieval (IR) courses at two universities since 2019 with shared tasks as tutorials, we derive a domain-neutral process model to capture the respective tutorial structure. Meanwhile, our teaching method has been adopted by other universities in IR courses, but also in other areas of AI such as natural language processing and robotics.

Introduction

In computer science, a shared task is a friendly research competition in which solutions to a given challenging research problem, formulated as a computational task, are developed by several independent teams and then comparatively evaluated. Typical results of such a "shared experiment" are an overview of the effectiveness and efficiency of state-of-the-art approaches to solve the task, but also standardized benchmarks, often adopted by the respective community. Participants in shared tasks are usually asked to describe their approaches in a paper. The organizers then publish a technical report on the benchmark, the experimental setup, the participants' solutions and their performance in solving the task. In this paper, we show that shared tasks are a promising and effective way to better link research and teaching (Healey 2005).

The use of shared tasks as part of required coursework is particularly appropriate for research disciplines in which shared tasks are already commonly organized. This requirement is met in artificial intelligence (AI), where there are a variety of competitions, such as the RoboCup in robotics (Alami et al. 2021; Nardi et al. 2014); the CASC (Sutcliffe 2021, 2016), SAT (Balyo et al. 2021; Järvisalo et al. 2012), and ASP (Gebser, Maratea, and Ricca 2020) competitions in reasoning;[1] the International Planning Competition (Coles et al. 2012); Chess (Krabbenbos, van den Herik, and Haworth 2019) and general gaming competitions (Genesereth and Björnsson 2013); and hundreds of machine learning classification benchmarks (Bischl et al. 2021). Such diversity is also found in applied AI fields, e.g., natural language processing (NLP) and information retrieval (IR). Across all of these fields, shared tasks generally prioritize scientific aspects over competitive aspects.

In our experience of organizing shared tasks over the last 15 years,[2] the proportion of student participants has steadily increased. To our knowledge, however, the use of shared tasks in teaching has so far been limited to individual efforts by students and/or faculty, and there is a lack of methodical application and shared experiences, which hinders the dissemination of best practices.

We have developed the teaching method proposed in this paper in courses on IR, a predominantly empirical research area on search engine construction and design that increasingly borrows methods from other AI fields such as machine learning or NLP. The inaugural Text REtrieval Conference (TREC) in 1992 (Harman 1992) at the National Institute of Standards and Technology (NIST) established a systematic evaluation scheme for shared tasks in IR. TREC has run them annually since then, and more recent sister conferences such as NTCIR, CLEF, and FIRE have emerged. Today, shared tasks are one of the pillars of IR evaluation.

In this paper, we present a didactically grounded integration of shared tasks in tutorials as a novel teaching method, discuss the implications for teaching, and share best practices. Our contributions are: (1) a semi-formal process model developed from our practical experience for structuring tutorials that methodically integrate a shared task, (2) an overview of shared task-based teaching from a didactic perspective, taking into account its relations to well-known approaches, and (3) a practical report based on ten IR courses at two universities with tutorials based on our method, together with the results of student evaluations.

[1] CADE ATP (Automated Theorem Proving) System Competition (CASC) of the Conference on Automated Deduction (CADE), Satisfiability (SAT), Answer-Set Programming (ASP)

[2] E.g. https://pan.webis.de/, https://touche.webis.de/ at CLEF

Background and Related Work

After discussing the concept and use of shared tasks in computer science in general, we review the literature on teaching and learning information retrieval in particular, and on linking research and teaching.

Shared Tasks in Computer Science and AI

We delineate two complementary concepts referred to by the term "shared task." The first, which we call a "shared task event," refers to a scientific event centered around an experiment. It invites scientists to work on solving a particular problem, independently implementing their best ideas, and then sharing their approach and findings. The second concept, which we call "shared task experiment," refers to the event's experimental setup. Besides a clear description of the task to be solved, a shared task experiment is basically defined by (1) an input data set consisting of problem instances and their solutions from the problem domain, and (2) a choice of effectiveness and/or efficiency measures as optimization criteria that determine how well the solutions generated by the participants solve the problem. Participants develop software that processes the input data to solve the task, whose well-formed outputs are called runs. These runs, or the software that generates them, are submitted to the shared task organizers for evaluation.

Our focus is on scientific shared task events organized at conferences and workshops (e.g., NeurIPS, TREC, CLEF and many more)[3] whose results are published transparently in proceedings. In contrast, shared tasks organized by industry (e.g., offered at Kaggle)[4] typically do not result in publications but attract data scientists to compete for substantial financial rewards. Shared task events are organized in great variety in computer science. It is not known how many shared tasks there are in total. However, considering the fact that up to dozens of shared tasks are held at conferences each year, and that in addition to independently organized events, many benchmarks and leaderboards are maintained only online, it seems reasonable to assume that there are thousands of shared tasks in the subfields of AI alone. Thus, an overview of shared tasks in their many manifestations is beyond the scope of this paper.

Although student participation in shared tasks seems to be common, there is not yet much support for it. In terms of data analytics training, Baba et al. (2018) argue that dedicated competition platforms should be developed for training purposes rather than relying on existing platforms. Support for students participating in shared tasks is rarely offered. One example of organized student participation is the mentoring track offered by the organizers of the eHealth task since at least 2017.[5] Although mentoring by shared task organizers can be helpful, a local support of students within an educational program can focus more on their individual needs and provides an additional motivation for our work.

Teaching and Learning Information Retrieval (IR)

In what follows, we review specifically the literature on IR teaching. At a general level, Efthimiadis et al.'s (2011) collection of papers on "Teaching and Learning in Information Retrieval" outlines three main issues: (1) What is the theory to be taught? (2) What practical problems should students solve? (3) What is the target audience for a course? Thornley (2011) states: "IR is [...] both a solution to a problem" (of how to access information) "and a problem of how to improve the solution." The target audience affects whether IR is taught as a solution or as a problem within a course. For example, MacFarlane (2011), Mothe and Sahut (2011), and Halttunen (2011) assume the former for library and information science students, while Mizzaro (2011) and Fox et al. (2011) assume the latter for computer science students. Likewise, our focus is on courses that teach IR as a problem.

Nevertheless, many shared tasks are interdisciplinary and involve other AI and computer science disciplines, allowing for blended courses that address both groups of students. The difficulties of teaching such interdisciplinary courses are considered theoretically by Blank et al. (2011). Eickhoff et al. (2017) argue in their special issue on the subject of search and learning that expertise in information retrieval can also benefit the effectiveness of learning in general; the articles in this issue focus mainly on increasing users' search skills. Should the use of shared tasks enable instruction with mixed groups of students, it could have an overall positive impact on the learning abilities of all students.

Without mentioning shared tasks, Fernández-Luna et al. (2009) provide an overview of course content and pedagogical aspects related to teaching and learning methods (i.e., e-learning, face-to-face instruction, and online approaches). Underlying these methods are a variety of philosophies, ranging from traditional lectures to personalized learning approaches. Halttunen and Sormunen (2000) propose the use of shared task data to analyze different search scenarios.

Tool-supported Teaching. A number of papers present tools that provide hands-on experience with information retrieval. Some of these tools index shared task data, such as JASSjr and VIRLab (Trotman and Lilly 2020; Fang et al. 2014). Although not designed to encourage students to participate in shared tasks, they demonstrate the utility of shared task data for instruction. In addition, PyTerrier (Macdonald et al. 2021) and Pyserini (Lin et al. 2021) simplify hands-on exercises by enabling researchers and students alike to implement AI-based retrieval pipelines in Python and run experiments directly on Google Colab.

In recent years, shared task support tools have been developed to reduce the workload of organizers and participants and to make results reproducible (Yadav et al. 2019; Breuer et al. 2019; Vanschoren et al. 2013; Jagerman, Balog, and de Rijke 2018; Tsatsaronis et al. 2015; Hopfgartner et al. 2015; Potthast et al. 2019). There are currently four platforms in productive use that are of particular interest for our purposes: CodaLab, EvalAI, STELLA, and TIRA.[6] They

[3] https://neurips.cc, https://trec.nist.gov, http://clef-initiative.eu

[4] https://www.kaggle.com

[5] https://archive.ph/nmzgb

[6] https://codalab.org, https://eval.ai, https://stella-project.org, https://tira.io

implement the so-called evaluation-as-a-service paradigm in the form of cloud-based web services for evaluations (Hopfgartner et al. 2018). Of these four systems, only STELLA and TIRA are hosted within a university, while CodaLab and EvalAI are based on Microsoft Azure and Amazon S3, respectively. Shared task support tools can be integrated into existing learning platforms to support the organization of shared task tutorials, even when staff is limited.

Linking Research and Teaching

The desire to link research and teaching leads to a wealth of further related work on the so-called research–teaching nexus, a multifaceted matter (Obwegeser 2016) and the subject of much debate (see Barnett (2005); McGill, Hobbs, and Pigott (2020)). McGill, Hobbs, and Pigott (2020) take a detailed look at the state of the art in IT education and empirically investigate how students perceive the associated benefits. Overall, linking research and teaching is a driving force for many methodical approaches (Healey 2005; Obwegeser 2016), including the one presented in this paper. Several authors have already applied appropriate didactic methods and techniques in their fields, most recently Raschka (2021) in courses on machine learning. Examples in IR education include the work of Thornley (2011) and Jones (2009). However, the use of shared tasks for teaching has not yet been studied or formally analyzed.

Didactics of Shared Tasks

After a brief overview of selected didactic approaches from the field of active learning (Bonwell and Eison 1991), we present our shared task teaching method.

Related Didactic Methods

Project-based Learning (PBL). Bender (2012) characterizes PBL as "[...] an instructional model based on having students confront real-world issues and problems that they find meaningful, determine how to address them, and then act in a collaborative fashion to create problem solutions." Many other variants of PBL exist. Frey (2012) describes one as a scheme of five main project phases called initiative, outline, plan, implementation, and completion. Each project phase can be completed with milestones or meta-interactions such as interim discussions. Collaborative and autonomous action contributes to the education of students.

Other Learning Methods. Competition-based learning is an extension of PBL in which competitive aspects are added to learning. Burguillo (2010) implements a competition-based approach with games where scoring influences student learning outcomes. He also discusses collaborative learning (maximizing collaboration for greater motivation and mutual reinforcement) and problem-based learning (based on open-ended problems, where teachers moderate the solution attempts); see also Barrett (2005) and Thornley (2011). Both are relevant to our approach, as is inquiry-based learning, which combines elements of problem-based learning and small-scale research (Kahn and O'Rourke 2005). Shared tasks lead to a method that integrates aspects of the aforementioned approaches.

A Method Based on Shared Tasks

Building on our experience in teaching IR, we developed the process model in Figure 1 in the form of a UML activity diagram (Rumbaugh, Jacobson, and Booch 2005) to formalize the course structure for a shared task tutorial. It shows the shared task event (top) and its interaction with an associated shared task tutorial (bottom). It is focused on the macro level and spans an entire term. The shared task event may take place independently of the tutorial. The diagram shows the logical flow of activities and indicates time overlaps within the shared task event and tutorial. The size of the activities does not reflect the amount of work required to complete an activity, which depends on student ability.

Model Description. The tutorial begins with instructions explaining the concept of a shared task and detailing the specific tasks students will be working on in small groups.

Student groups meet regularly with faculty in individual tutoring sessions to ask questions and receive advice and feedback on their progress. They are guided to first search for related work for the shared task and to familiarize themselves with academic writing and literature review. In parallel, they select tools, libraries, and APIs for their implementation. With the release of the datasets from the shared task event and based on their own data analyses, students develop a methodological approach and implement a vertical prototype in the first coding phase. In the next step, students refine the prototype to achieve better results. At this point in the tutorial, it is advisable to hold interim presentations where student groups report on their status and share ideas.

Then begins a second phase of individual tutoring, report writing, and coding. To determine the effectiveness of their systems, the groups conduct evaluations based on real data. Thus, in addition to receiving qualitative feedback from instructors, students also receive quantitative results and learn the basics of systematic evaluation of AI systems. By the end of the second learning phase, students have created their final source code and the software runs. After the final presentation, they receive feedback on their work through plenary discussions and comments from the instructors.

At the end of the term (right in Figure 1), students revise their final report based on the feedback they received. The report is submitted together with the source code (both shown in red) of their system, e.g., as a Git repository. Grading is based on the final group presentations and the report.

Students are asked to submit their report in the form of a scientific paper to practice scientific writing and lower the bar for a possible participation in the official shared task event. To participate as an extracurricular activity, they sign up and submit the required artifacts (represented as a submit signal). The organizers of the shared task event decide on the contribution based on peer review and its evaluation. It is important that official participation should be on a voluntary basis and independent of the tutorial grade. In addition, teachers should communicate in advance with the shared task organizers about potential participants.

Discussion. We revisit the process model in Figure 1 to place shared-task tutorials in the context of the related didactic methods reviewed above. We see the shared task teaching

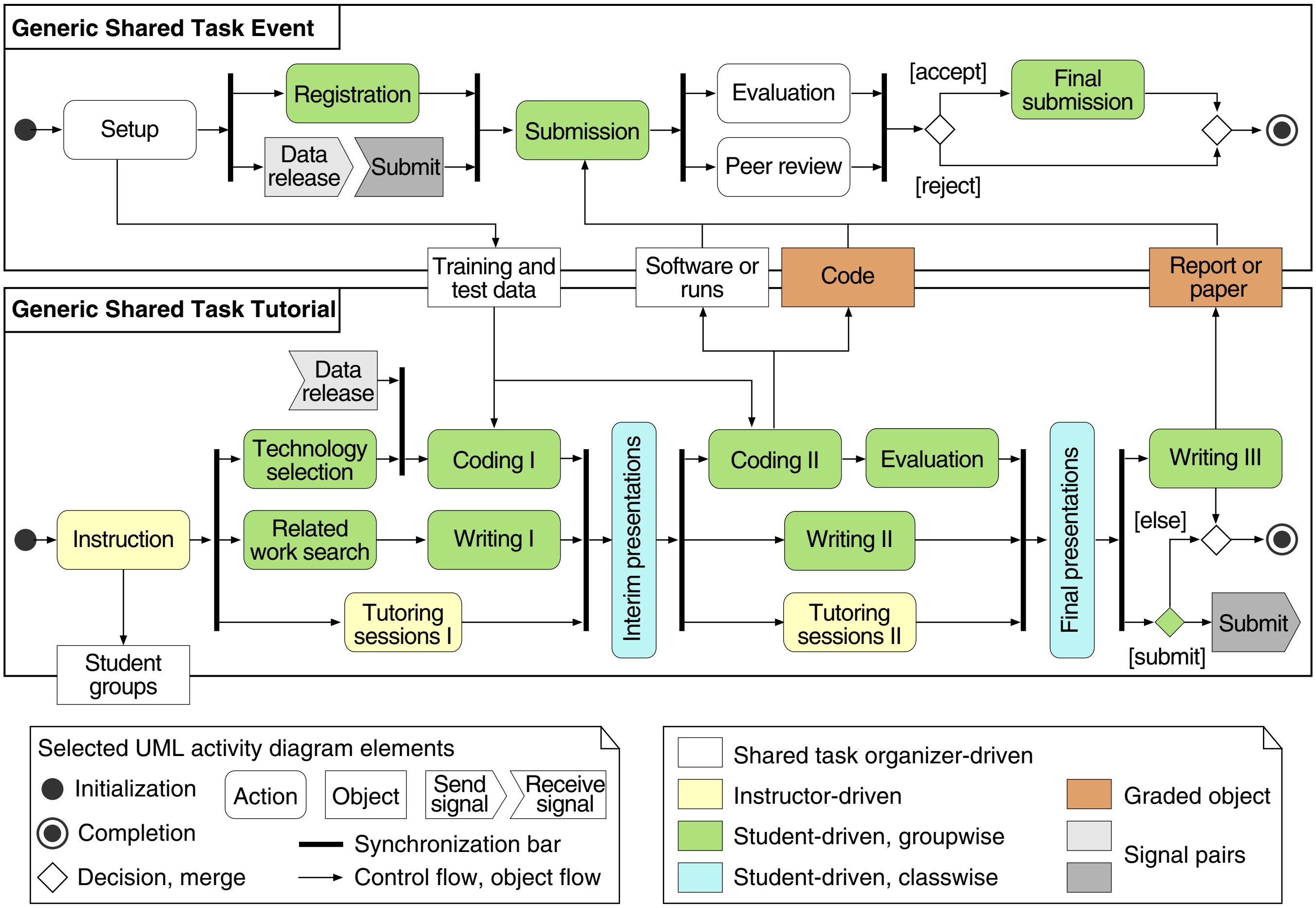

Figure 1: UML 2 activity diagram of the process model of a shared task event (top) and an associated shared task tutorial (bottom). Note: Concurrency is not a requirement. The amount of work per action varies.

method as a specialization of project-based learning (PBL) as described by Bender (2012) and Frey (2012). Several PBL elements, such as initiative and plan, become possible only at the beginning of vertical prototyping (see activity Coding I in Figure 1), since students are not free to choose their projects, but are restricted to the shared task offered for a given event. Nevertheless, our approach is specialized to PBL, as relevant elements are present in all of its student-driven actions, e.g., in the selection of technologies and tools or in the elaboration and planning of own solutions.

Competitive learning (Burguillo 2010) differs from PBL in terms of how students are assessed. Students' grades in the course cannot be based on the effectiveness of their solutions in the shared tasks because there is no clear evidence that this has an impact on collaborative learning (ter Vrugte et al. 2015; Goodman and Crouch 1978). However, when shared task tutorials emphasize research aspects by supporting publications in the shared task event, they are a form of inquiry-based learning (Kahn and O'Rourke 2005).

Overall, the variety of shared tasks available in IR allows students from different disciplines to work together. They become familiar with related areas and learn that one-size-fits-all solutions are rarely enough. The heterogeneity among students, especially in terms of their skills, facilitates mutual exchange and potential research contributions, thus advancing IR itself. The use of shared tasks in teaching prompts students to actively participate in research.

Integrating Shared Tasks

We discuss 13 IR courses, ten with shared task tutorials and three without, their target audience, structure, content, student feedback, and voluntary participation in shared task events. Course materials are in the public domain.[7]

Teaching IR with Shared Tasks

We employed shared task tutorials in information retrieval courses at the Bachelor's and Master's level over five terms (2019 to 2022) at two German universities, namely Leipzig University (University A), and Martin-Luther-Universität Halle-Wittenberg (University B).[8] The shared task was run in association with the Touché lab at CLEF.

[7] https://webis.de/lecturenotes.html#information-retrieval, https://webis.de/lecturenotes.html#part-scientific-toolbox

[8] Winter and summer term 2019/20 and 2020: https://archive.ph/t5yLd; winter and summer term 2020/21 and 2021: https://archive.ph/efCQZ; and winter term 2021/22: https://archive.ph/d5nAg.

(a) Registrations and group submissions from University A

Term	Level	Comp. Sci.		Data Sci.		Dig. Hum.		Other		Total	Sub.
		Abs.	Rel.	Abs.	Rel.	Abs.	Rel.	Abs.	Rel.		
4	Master	32	1.00	0	0.00	0	0.00	0	0.00	32	6/10
5	Bachelor	53	0.83	0	0.00	5	0.08	6	0.09	64	
6	Master	19	0.61	9	0.29	3	0.10	0	0.00	31	3/9
7	Bachelor	52	0.81	0	0.00	9	0.14	3	0.05	64	
8	Master	9	0.30	20	0.67	1	0.03	0	0.00	30	6/7
Total		165	0.75	29	0.13	18	0.08	9	0.04	221	15/26

(b) Registrations and group submissions from University B

Term	Level	Comp. Sci.		Bio. Inf.		Other		Total	Sub.
		Abs.	Rel.	Abs.	Rel.	Abs.	Rel.		
4	Bachelor	7	1.00	0	0.00	0	0.00	7	1/4
6	Bachelor	11	0.69	4	0.25	1	0.06	16	5/6
6	Master	3	0.75	1	0.25	0	0.00	4	3/3†
8	Bachelor	4	1.00	0	0.00	0	0.00	4	1/2
8	Master	4	1.00	0	0.00	0	0.00	4	1/1
		29	0.83	5	0.14	1	0.03	35	11/16

(c) Course evaluation from University A

Item	Scale	Term:	1*	2*	3*	4	5	6	7	1-3*	4-7
		Submitted Forms:	26	14	30	13	30	9**	7**	Avg.	Avg.
Pace	too slow (1) - too fast (5)		3.2 (0.4)	3.1 (0.3)	3.4 (0.6)	2.9 (0.5)	3.1 (0.6)	3.1 (0.3)	3.1 (0.4)	3.2	**3.1**
Scope	too narrow (1) - too broad (5)		3.5 (0.5)	3.2 (0.6)	3.5 (0.6)	3.2 (0.7)	3.3 (0.6)	2.9 (0.9)	3.3 (0.5)	3.4	**3.2**
Effort	too little (1) - too much (5)		3.3 (0.6)	3.4 (0.6)	3.3 (0.5)	3.3 (0.5)	3.1 (0.6)	3.3 (0.7)	3.4 (0.5)	**3.3**	**3.3**
Structure	disagree (1) - agree (5)		4.5 (0.7)	4.4 (0.6)	4.1 (1.0)	4.7 (0.5)	4.2 (0.7)	4.3 (1.0)	4.3 (0.8)	4.3	**4.4**
Consistency	disagree (1) - agree (5)		4.4 (0.9)	4.8 (0.4)	4.1 (0.9)	4.4 (0.7)	4.0 (1.0)	4.1 (1.3)	4.4 (0.5)	**4.4**	4.2
Transparency	disagree (1) - agree (5)		4.5 (0.6)	4.7 (0.5)	4.2 (1.0)	4.7 (0.5)	4.1 (1.0)	4.1 (1.3)	—	**4.5**	4.3
Satisfaction	low (1) - high (5)		4.4 (0.6)	4.4 (0.5)	4.0 (0.9)	4.5 (0.5)	4.3 (0.7)	4.4 (0.9)	4.1 (0.7)	4.27	**4.33**

† Two groups among these three were mixed teams containing Bachelor students and one Master student each.

Table 1: Overview of (a,b) number and ratio of enrolled students per term, course level and degree program, number of student groups submitting reports to the shared task event; (c) course evaluation as given by students with item, scale, and mean (std.) score per term and as average (better one in bold font) over all terms with and without (*) using shared tasks; (a,b) per university, (c) exclusively from University A. (**) During the pandemic, fewer students returned the online evaluation forms; enrollment and successful completion did not deviate from previous terms.

Audience Table 1a and b summarize course statistics. At University A, students come mainly from three degree programs: computer science (∼76% of students), data science (∼12%), and digital humanities (∼8%). Others (∼4%) are in programs such as business informatics and mathematics. A total of 219 students completed the courses at University A during the observation period, with class sizes ranging from approximately 30 participants (Master's level) to 64 (Bachelor's level). At University B, the courses are smaller (35 students total), with participation ranging from 4 students (Master's level) to 16 (Bachelor's level) per term. Most participants study computer science (∼83%), the remainder bioinformatics (∼14%) and other programs (∼2%).

Structure. The course is structured in the same way at both universities. It is divided into weekly 90-minute lectures and tutorial sessions. The lecture recapitulates the fundamentals and introduces advanced concepts and methods of IR. The aim of the tutorial is to transfer theoretical content into practical training. In addition, tutorial sessions are held in an open Q&A format to allow students to interact with each other and with faculty. The course duration is 14 weeks of lectures/tutorial sessions and an additional 6 weeks to complete final reports without coursework. University A courses are offered in alternating terms at Bachelor and Master levels, while University B courses are offered at both levels in the same term.

Content. The content of the lecture considers IR in its entire breadth and is divided into four blocks: (1) Introduction that gives an overview of the architecture of a search engine and examples of retrieval problems. (2) Indexing, with lectures on the origins and role of indexes for retrieval, on text preprocessing and morphological analysis (both topics in NLP), and on inverted indexes. (3) Evaluation, which addresses common practices in laboratory experiments (linking to shared tasks) in IR, measures of effectiveness for evaluating retrieval performance, and methods for comparing the effectiveness of different retrieval systems. And finally, Part (4) reviews groups of retrieval models, addresses the concepts of empirical, probabilistic, and generative models, and covers some machine learning methods and various applications of IR.

In the tutorial, students develop their own retrieval system, in terms of the shared tasks we have selected. Following the notation in the process model (see Figure 1), Instruction lasts exactly one session, we allow Student groups of 2 to 4 students, and hold Tutoring sessions every week. We expect students to implement basic retrieval systems as a vertical prototype (Coding I) using open source search engines on datasets according to the shared task. Advanced students can reproduce the systems from the previous Related work search. A Bachelor's level course is supported by programming sessions in which the instructor demonstrates the development of a non-competitive baseline solution during the term. Assuming better programming skills, this is dropped at the Master's level in favor of more extensive tutoring. Coding II leads to refined prototypes that draw on students' original ideas to develop new approaches that have not yet been used in the shared task context. In the assessment, students compare the retrieval performance of their vertical prototype and their refined prototype.

In the last tutorial session, students give a 15 minute presentation followed by a discussion of their approach. At the end of the term, each group submits a report in the form of a scientific paper, their source code (a Git repository with history), and a working implementation of their system on the TIRA platform (Potthast et al. 2019). Students are graded on the basis of their group's talk (25% of the grade) and their report (75%). After their final presentation, they decide whether or not to officially submit their approach to the shared task event, independent of the grade.

Official Shared Task Participation. Our data refers to eight terms, 1-3 before using shared tasks, 4-8 involving them. Official participation is voluntary and welcome from Bachelor and Master level. The timelines of shared task event and course conflict in odd terms. Thus official submissions from University A stem exclusively from Master students but University B has mixed contributions (even terms). This is summarized in the last columns of Table 1(a) and (b): At University A, 6 of 10 groups chose to submit in Term 4, decreasing to 3 of 9 in Term 6. Students' feedback suggests that the strenuous learning situation during the COVID-19 pandemic caused this drop. In Term 8 (online teaching only in the second half) group submissions increased to 6 of 7. The pandemic did not seem to lower student submissions at University B, where 1 of 4 groups submitted in the fourth, 8 of 9 in the sixth, and 2 of 3 in the eighth term. The higher portion of Bachelor level groups at University B did not decrease the number of submissions, indicating shared task tutorials as an adequate teaching approach for both levels.

Student Evaluation. At the end of term, a questionnaire-based course evaluation is conducted by the students at University A,[9] whereas the course sizes in University B are too small for reliable and anonymous evaluation results. Table 1c lists seven questionnaire items that are particularly relevant to measuring the success of teaching IR with shared-task tutorials. A total of 129 students (59 with and 70 without shared tasks) provided feedback.

For the Pace item, students were asked to answer whether "The pace of this course is ..." too slow (1) to too fast (5) on a corresponding scale. Since the course was designed to fit the schedule of the shared task tutorial and, in part, the shared task event, the overall pace of the content is critical. The score indicates that this has been successfully implemented, with a mean score across all terms of 3 and a low standard deviation. In addition, no discrepancies were found between Bachelor's and Master's degree programs. The scope item, framed as "The scope of this course is ...", had a range of 1 (too narrow) to 5 (too wide). The requirement is to balance basic IR methods with those specifically applicable to the shared task. In addition, the the practical exercises must be consistent with the theoretical content of the lectures. Again, scores around 3 with little variation and no systematic differences between course levels indicate a successful outcome.

The third item measures the Effort as "The amount of work required for this course is ...", rated from 1 (too low) to 5 (too high). A total workload of 300 hours (10 ECTS credits) for lectures and tutorials was targeted for the course. The mean score of 3.3 across all terms with little variation suggests that this goal was well met, but the slight increase of 0.3 from the perfect score suggests that faculty should pay attention to student workload in a shared task environment.

The next three items assess Structure ("The course is clearly structured."), Consistency ("All parts of the course are well coordinated."), and Transparency ("The goals of the course are transparent.") on a scale of 1 (strongly disagree) to 5 (strongly agree). Here, the rating was consistently above 4, indicating that shared task tutorials did not affect students' ability to follow the objective and structure of the course. Overall student Satisfaction is also high, exceeding 4 in all terms. On the whole, the students evaluations indicate that teaching IR with shared task tutorials is well received and does not have a detrimental effect on their workload, success, and satisfaction. Note also that all Terms 5 to 7 were conducted online due to the COVID-19 pandemic. The consistency of results between online and face-to-face instruction is a testament to the flexibility and adaptability of the shared task method.

The last two columns summarize the scores of each teaching method (with and without shared tasks). The differences are very small, but the average scores of the terms with shared tasks (4 – 7) are slightly better in the categories of Pace, Scope, Structure, and overall Satisfaction, but with a tie in Effort. In contrast, the terms without shared tasks (1 – 3) scored better on Consistency and Transparency. Student feedback suggests that this is due to problems with synchronizing the timeline and accessing data at the beginning.

Lessons Learned from Teaching Practice

In this section we discuss benefits and challenges as lessons learned, reflecting on our experience on the past years of teaching with shared tasks at University A and University B.

Timeline Synchronization. A few terms started about four weeks before the shared tasks' data sets were released. We addressed this issue by using the time to instruct students to start working on the related work of their papers.

Heterogeneous Knowledge and Skills. Students' programming skills were on different levels. We expected a lower level among the Bachelor students and provided hands-on programming sessions at the expense of teaching more advanced retrieval methods. The Master students were expected to catch up on their own. We advised the students to team up with at least one member with sufficient skills. While this solution does not maximize an individual group's efficiency, it fosters inclusion of all students.

[9] Unfortunately, Term 8 could not be evaluated due to a change in the questionnaire that made it incomparable with the others.

Teaching Format. Due to the COVID-19 pandemic, most weekly tutoring sessions had to be held online. Instead of live teaching, for lectures we provided videos. Communication took place virtually via video conference tools (weekly Q&A sessions), Discord (for quick communications between tutoring sessions), and email (for important announcements). We found the online format integrates very well with the shared task concept because students engaged actively in the course. We think this transfers to hybrid teaching formats and allows for locally distributed classes.

Research Experience. We have seen that shared tasks allow students to experience scientific practice, which may inspire them to consider a career in research after their studies.

Interdisciplinarity. Course participation of students outside of computer science was limited mostly to students of other computational disciplines. Contributing solutions to multi-domain shared tasks could benefit from interdisciplinary student teams also from other fields of study.

Platforms and Tools. Existing shared task platforms and tools could be extended to assist teaching, e.g. by coordinating shared task timelines and term deadlines. This could ease the teaching effort and therefore might increase the allowed number of participating students in the future.

Conclusion

In this paper, we have suggested to use shared tasks as a teaching method that combines research with higher education: students get in touch and possibly even shape state-of-the-art research. Shared tasks have the potential to provide an open-minded and inclusive educational environment. We discussed challenges and benefits of using shared tasks as a teaching method as per the experiences in ten courses at two universities. In the future, one could rethink the role of evaluation campaigns and analyze whether and how they can systematically support the use of shared tasks for teaching in an international context. While this would be challenging due to differences between curricula, universities, and countries, it would also provide a great opportunity for educating new generations of students, researchers, and developers.

Perspectives in Other Courses

Shared tasks following our methodology are currently also used in teaching within and beyond IR at universities other than the one's focused on in this paper. As an outlook, we highlight three additional experiences.

Within IR. University of Padua in Italy realized a Master's level IR course in computer engineering using shared tasks in 2020. Students participated in groups in the shared task Touché[10] at CLEF. The task targeted advanced methods but it was possible to address the task also with ad-hoc baseline systems. Students reported an above average satisfaction with the teaching approach and considered the workload adequate. Some students found the course a bit demanding in terms of coding skills, though. From the lecturer's point of view, teaching with shared tasks required continuous interaction with students and was therefore more time consuming than a teaching approach based on lectures and a final exam. However, this course was explicitly seeking interaction with students and shared tasks offered an opportunity to drive this process in a uniform way. Therefore, rather than as an additional load, this lecturer sees shared tasks as a pillar for interactive and project-based teaching.

Beyond IR: Shared Tasks for Teaching Robotics. In 2022, a shared-task-like exercise in path planning for dynamic environments was developed at University of Applied Sciences Bochum in Germany and used as an exercise in an online robotics course for Bachelor's students. The task was not an official shared task within a scientific event but was specifically designed for the course to support students in understanding the lecture's content in a motivating environment of friendly competition with their classmates. The students developed and implemented a path-planning algorithm to navigate a robot through a simulated maze. Overall, they reacted positively to the shared task-like setup and emphasized the increased learning progress compared to other exercises, especially due to creative thinking and hands-on programming. However, the students struggled with the setup of the simulation environment. We believe, running on a shared task platform as reviewed in Section "Background and Related Work" with the simulation environment already integrated would be an interesting future use case of these tools. From the lecturer's point of view on the teaching effort, the shared task tutorial was about equally time consuming as teaching the class without a shared task. The lecturer will continue to use the developed task in their later courses.

Beyond IR: Shared Tasks for Teaching NLP. Several shared tasks were developed at Bauhaus-Universität Weimar in Germany as exercises in two online NLP courses for Master's students in computer science. In a first course in 2020, students worked in groups on one of 17 established shared tasks. All student groups surpassed their respective tasks' baseline from the official event, although with limited novelty. The students' feedback was very positive and they emphasized that they enjoyed doing "real" research.

In a second course in 2021, the students solved four smaller, shared task-like exercises individually. Similar to the above mentioned robotics course, the exercises in the second NLP course were aligned to topics of the course and explicitly designed to help the students understand them. All passing students' results improved on the provided baselines, their feedback was positive and emphasized the freedom in exploring their own solutions.

From the lecturer's point of view, the shared task tutorial was less time consuming than teaching the class without shared tasks but with bi-weekly assignments instead. The main reason for the lower teaching effort using shared tasks is the system performance-based approach to evaluate if student's passed or not, which was done mostly automatically using TIRA. This approach, however, does not produce grades and therefore calls for an additional solution to assess students' learning success on a more fine grained level.

[10] https://archive.ph/t5yLd

References

Alami, R.; Biswas, J.; Cakmak, M.; and Obst, O., eds. 2021. *RoboCup 2021: Robot World Cup XXIV*. Springer.

Baba, Y.; Takase, T.; Atarashi, K.; Oyama, S.; and Kashima, H. 2018. Data Analysis Competition Platform for Educational Purposes: Lessons Learned and Future Challenges. In McIlraith, S. A.; and Weinberger, K. Q., eds., *The 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 7887–7892. AAAI Press.

Balyo, T.; Froleyks, N.; Heule, M.; Iser, M., Markus amd Järvisalo; and Suda, M. 2021. Proceedings of SAT Competition 2021 : Solver and Benchmark Descriptions. Technical Report B-2021-1, University of Helsinki. http://hdl.handle.net/10138/333647.

Barnett, R., ed. 2005. *Reshaping the University: New Relationships between Research, Scholarship and Teaching*. SRHE and Open University Press.

Barrett, T. 2005. Understanding Problem-based Learning (PBL). In *Handbook of Enquiry and Problem-based Learning*, chapter 2. AISHE and CELT, National University of Ireland.

Bender, W. N. 2012. *Project-Based Learning: Differentiating Instruction for the 21st Century*. Corwin.

Bischl, B.; Casalicchio, G.; Feurer, M.; Gijsbers, P.; Hutter, F.; Lang, M.; Mantovani, R. G.; van Rijn, J.; and Vanschoren, J. 2021. OpenML Benchmarking Suites. In Vanschoren, J.; and Yeung, S., eds., *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Blank, D.; Fuhr, N.; Henrich, A.; Mandl, T.; Rölleke, T.; Schütze, H.; and Stein, B. 2011. Teaching IR: Curricular Considerations. In (Efthimiadis et al. 2011), chapter 3, 31–46.

Bonwell, C. C.; and Eison, J. A. 1991. *Active Learning: Creating Excitement in the Classroom*. Number 1 in 1991 AEHE-ERIC Higher Education Report. George Washington University.

Breuer, T.; Schaer, P.; Tavakolpoursaleh, N.; Schaible, J.; Wolff, B.; and Müller, B. 2019. STELLA: Towards a Framework for the Reproducibility of Online Search Experiments. In Clancy, R.; Ferro, N.; Hauff, C.; Lin, J.; Sakai, T.; and Wu, Z. Z., eds., *Proceedings of the Open-Source IR Replicability Challenge co-located with 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, OSIRRC@SIGIR 2019, Paris, France, July 25, 2019*, volume 2409 of *CEUR Workshop Proceedings*, 8–11. CEUR-WS.org.

Burguillo, J. C. 2010. Using game theory and competition-based learning to stimulate student motivation and performance. *Computers & Education*, 55(2): 566–575.

Coles, A.; Coles, A.; Olaya, A. G.; Jiménez, S.; López, C. L.; Sanner, S.; and Yoon, S. 2012. A Survey of the Seventh International Planning Competition. *AI Magazine*, 33(1): 83–88.

Efthimiadis, E.; Fernández-Luna, J. M.; Huete, J. F.; and MacFarlane, A. 2011. *Teaching and Learning in Information Retrieval*, volume 31 of *The Information Retrieval Series*. Springer Science & Business Media.

Eickhoff, C.; Gwizdka, J.; Hauff, C.; and He, J. 2017. Introduction to the special issue on search as learning. *Information Retrieval Journal*, 20(5): 399–402.

Fang, H.; Wu, H.; Yang, P.; and Zhai, C. 2014. VIRLab: a web-based virtual lab for learning and studying information retrieval models. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR 2014), Gold Coast, Queensland, Australia, Jul 6-11*, SIGIR '14, 1249–1250. Association for Computing Machinery.

Fernández-Luna, J. M.; Huete, J. F.; MacFarlane, A.; and Efthimiadis, E. N. 2009. Teaching and learning in information retrieval. *Information Retrieval*, 12(2): 201–226.

Fox, E.; Murthy, U.; Yang, S.; Torres, R. d. S.; Velasco-Martin, J.; and Marchionini, G. 2011. Pedagogical Enhancements for Information Retrieval Courses. In (Efthimiadis et al. 2011), chapter 4, 47–60.

Frey, K. 2012. *Die Projektmethode: Der Weg zum bildenden Tun*. Beltz, 12 edition. In German.

Gebser, M.; Maratea, M.; and Ricca, F. 2020. The Seventh Answer Set Programming Competition: Design and Results. *Theory and Practice of Logic Programming*, 20(2): 176–204.

Genesereth, M.; and Björnsson, Y. 2013. The International General Game Playing Competition. *AI Magazine*, 34(2): 107–111.

Goodman, D. A.; and Crouch, J. 1978. Effects of competition on learning. *Improving College and University Teaching*, 26(2): 130–133.

Halttunen, K. 2011. Pedagogical Design and Evaluation of Interactive Information Retrieval Learning Environment. In (Efthimiadis et al. 2011), chapter 5, 61–73.

Halttunen, K.; and Sormunen, E. 2000. Learning information retrieval through an educational game. Is gaming sufficient for learning? *Education for Information*, 18(4): 289–311.

Harman, D. 1992. Overview of the First Text REtrieval Conference (TREC-1). In Harman, D., ed., *Proceedings of The First Text REtrieval Conference, TREC 1992, Gaithersburg, Maryland, USA, November 4-6, 1992*, volume 500-207 of *NIST Special Publication*, 1–20. National Institute of Standards and Technology (NIST).

Healey, M. 2005. Linking Research and Teaching: Exploring Disciplinary Spaces and the Role of Inquiry-based Learning. In (Barnett 2005), chapter 5, 67–78.

Hopfgartner, F.; Brodt, T.; Seiler, J.; Kille, B.; Lommatzsch, A.; Larson, M. A.; Turrin, R.; and Serény, A. 2015. Benchmarking news recommendations: The CLEF NewsREEL use case. *SIGIR Forum*, 49(2): 129–136.

Hopfgartner, F.; Hanbury, A.; Müller, H.; Eggel, I.; Balog, K.; Brodt, T.; Cormack, G.; Lin, J.; Kalpathy-Cramer, J.; Kando, N.; Kato, M.; Krithara, A.; Gollub, T.; Potthast, M.;

Viegas, E.; and Mercer, S. 2018. Evaluation-as-a-service for the computational sciences: Overview and outlook. *Journal of Data and Information Quality*, 10(4): 15:1–15:32.

Jagerman, R.; Balog, K.; and de Rijke, M. 2018. OpenSearch: Lessons learned from an online evaluation campaign. *Journal of Data and Information Quality*, 10(3): 13:1–13:15.

Jones, G. 2009. An inquiry-based learning approach to teaching information retrieval. *Information Retrieval*, 12: 148–161.

Järvisalo, M.; Le Berre, D.; Roussel, O.; and Simon, L. 2012. The International SAT Solver Competitions. *AI Magazine*, 33(1): 89–92.

Kahn, P.; and O'Rourke, K. 2005. Understanding Problem-based Learning (PBL). In *Handbook of Enquiry and Problem-based Learning*, chapter 1. AISHE and CELT, National University of Ireland.

Krabbenbos, J.; van den Herik, J.; and Haworth, G. 2019. WCCC 2019: The 25th World Computer Chess Championship. *ICGA Journal*, 41(4): 206–221.

Lin, J.; Ma, X.; Lin, S.; Yang, J.; Pradeep, R.; and Nogueira, R. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In Diaz, F.; Shah, C.; Suel, T.; Castells, P.; Jones, R.; and Sakai, T., eds., *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, 2356–2362. ACM.

Macdonald, C.; Tonellotto, N.; MacAvaney, S.; and Ounis, I. 2021. PyTerrier: Declarative Experimentation in Python from BM25 to Dense Retrieval. In Demartini, G.; Zuccon, G.; Culpepper, J. S.; Huang, Z.; and Tong, H., eds., *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, 4526–4533. ACM.

MacFarlane, A. 2011. Using Multiple Choice Questions to Assist Learning for Information Retrieval. In (Efthimiadis et al. 2011), chapter 8, 107–121.

McGill, T. J.; Hobbs, V.; and Pigott, D. 2020. Integrating Research into Information Technology Education. In Luxton-Reilly, A.; and Szabo, C., eds., *ACE 2020, Proceedings of the Twenty-Second Australasian Computing Education Conference, Melbourne, VIC, Australia, February 4-6, 2020*, 1–10. ACM.

Mizzaro, S. 2011. Teaching Web Information Retrieval to Computer Science Students: Concrete Approach and Its Analysis. In (Efthimiadis et al. 2011), chapter 10, 137–151.

Mothe, J.; and Sahut, G. 2011. Is a Relevant Piece of Information a Valid One? Teaching Critical Evaluation of Online Information. In (Efthimiadis et al. 2011), chapter 11, 153–168.

Nardi, D.; Noda, I.; Ribeiro, F.; Stone, P.; von Stryk, O.; and Veloso, M. 2014. RoboCup Soccer Leagues. *AI Magazine*, 35(3): 77–85.

Obwegeser, N. 2016. Integrating Research and Teaching in the IS Classroom: Benefits for Teachers and Students. *Journal of Information Systems Education*, 27(4): 249–258.

Potthast, M.; Gollub, T.; Wiegmann, M.; and Stein, B. 2019. TIRA Integrated Research Architecture. In Ferro, N.; and Peters, C., eds., *Information Retrieval Evaluation in a Changing World: Lessons Learned from 20 Years of CLEF*, volume 41 of *The Information Retrieval Series*, 123–160. Springer.

Raschka, S. 2021. Deeper Learning By Doing: Integrating Hands-On Research Projects Into A Machine Learning Course. *Proceedings of Machine Learning Research*, 170: 46–50.

Rumbaugh, J.; Jacobson, I.; and Booch, G. 2005. *The Unified Modeling Language Reference Manual*. Addison Wesley, 2. edition.

Sutcliffe, G. 2016. The CADE ATP System Competition – CASC. *AI Magazine*, 37(2): 99–101.

Sutcliffe, G. 2021. Proceedings of CASC-28 – the CADE-28 ATP System Competition. Technical report, tptp.org; University of Miami. https://tptp.org/CASC/28/Proceedings.pdf.

ter Vrugte, J.; de Jong, T.; Vandercruysse, S.; Wouters, P.; van Oostendorp, H.; and Elen, J. 2015. How competition and heterogeneous collaboration interact in prevocational game-based mathematics education. *Computers & Education*, 89: 42–52.

Thornley, C. 2011. Teaching Information Retrieval Through Problem-Based Learning. In (Efthimiadis et al. 2011), chapter 13, 183–198.

Trotman, A.; and Lilly, K. 2020. JASSjr: The Minimalistic BM25 Search Engine for Teaching and Learning Information Retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, China, Jul 25-30*, SIGIR '20, 2185–2188. ACM.

Tsatsaronis, G.; Balikas, G.; Malakasiotis, P.; Partalas, I.; Zschunke, M.; Alvers, M. R.; Weissenborn, D.; Krithara, A.; Petridis, S.; Polychronopoulos, D.; Almirantis, Y.; Pavlopoulos, J.; Baskiotis, N.; Gallinari, P.; Artières, T.; Ngomo, A. N.; Heino, N.; Gaussier, É.; Barrio-Alvers, L.; Schroeder, M.; Androutsopoulos, I.; and Paliouras, G. 2015. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16: 138:1–138:28.

Vanschoren, J.; van Rijn, J. N.; Bischl, B.; and Torgo, L. 2013. OpenML: Networked science in machine learning. *SIGKDD Explor.*, 15(2): 49–60.

Yadav, D.; Jain, R.; Agrawal, H.; Chattopadhyay, P.; Singh, T.; Jain, A.; Singh, S.; Lee, S.; and Batra, D. 2019. EvalAI: Towards better evaluation systems for AI agents. *CoRR*, abs/1902.03570.