Teaching Computational Aspects in the Digital Humanities Program at University of Stuttgart – Intentions and Experiences

Nils Reiter[1], **Sarah Schulz**[1], **Gerhard Kremer**[1],
Roman Klinger[1], **Gabriel Viehhauser**[2] and **Jonas Kuhn**[1]

[1]Institut für Maschinelle Sprachverarbeitung
[2]Institut für Literaturwissenschaft
Universität Stuttgart
viehhauser@ilw.uni-stuttgart.de
firstname.lastname@ims.uni-stuttgart.de

Abstract

The structure of the Digital Humanities master's program at University of Stuttgart is characterized by a big proportion of classes related to natural language processing. In this paper, we discuss the motivation for this design and associated challenges students and teachers are faced with. To provide background information, we also sum up our underlying perspective on Digital Humanities. Our discussion is driven by a qualitative analysis of a survey handed to the students of the program.

1 Introduction

The importance of computer-assisted methods is increasing in various research fields, for instance in Biology (*Bioinformatics* and *Computational Biology*), Media Sciences (*Mediainformatics*), or Geography (*Geoinformatics*). More recently, the broad fields of Humanities and Social Sciences adopted the use of computational methods, which are often referred to as *Digital Humanities* (Jannidis et al., 2017). However, in contrast to preceding research domains and sciences, the use of quantitative and statistical methods in this area is less popular, which poses additional challenges to the introduction of formal methods to the field.

The University of Stuttgart introduced a master's program for Digital Humanities (DH) in 2015. While other universities have been offering DH programs in various forms, one key characteristics of the DH program in Stuttgart is the strong influence of Computational Linguistics (CL) on the program, both on the design and planning of the program and on the actual courses.

The program consists of three main areas: (1) A specific discipline in Humanities, in which each student deepens their knowledge in the field they studied in a previous undergraduate program, (2) Digital Humanities, and (3) Computer Sciences (CS). While different computer science institutes are offering courses in this program, a majority of the courses are offered by the *Institut für Maschinelle Sprachverarbeitung* (Institute for Natural Language Processing), both electable and compulsory courses.

In this paper, we present the intentions behind the study program and report on results of a survey conducted among the first two cohorts of students.

2 Digital Humanities and Computational Linguistics

Digital Humanities is a new and diverse field, and pinpointing and defining its actual novelty has been a hot topic in the past years (Presner and Johanson, 2009; Berry, 2011; Gibbs, 2011; Svensson, 2012; Kuhn and Reiter, 2015; Dunst, 2017; Thaller, 2017). While differing views are plausible and valid, we believe that *formalization* is one key aspect of the field's novelty, applied to both the research questions and to the analysis objects. The formal definition of – in principle – quantifiable properties is a fundamental step when switching the focus from particular, incomparable pieces of art to comparing, counting and categorizing objects. Only properly formalized concepts can be reliably applied on different objects of interest, and only then can these objects be compared or viewed quantitatively in the first place (for instance, the comparison of syntactic profiles for different authors relies on the proper formalization of syntax).

Formalization, in this view, does not necessarily imply the implementation of such approaches in a computer. There are formalized approaches to Humanities research questions or objects that are non- or pre-digital, e. g., John Snow's map of a

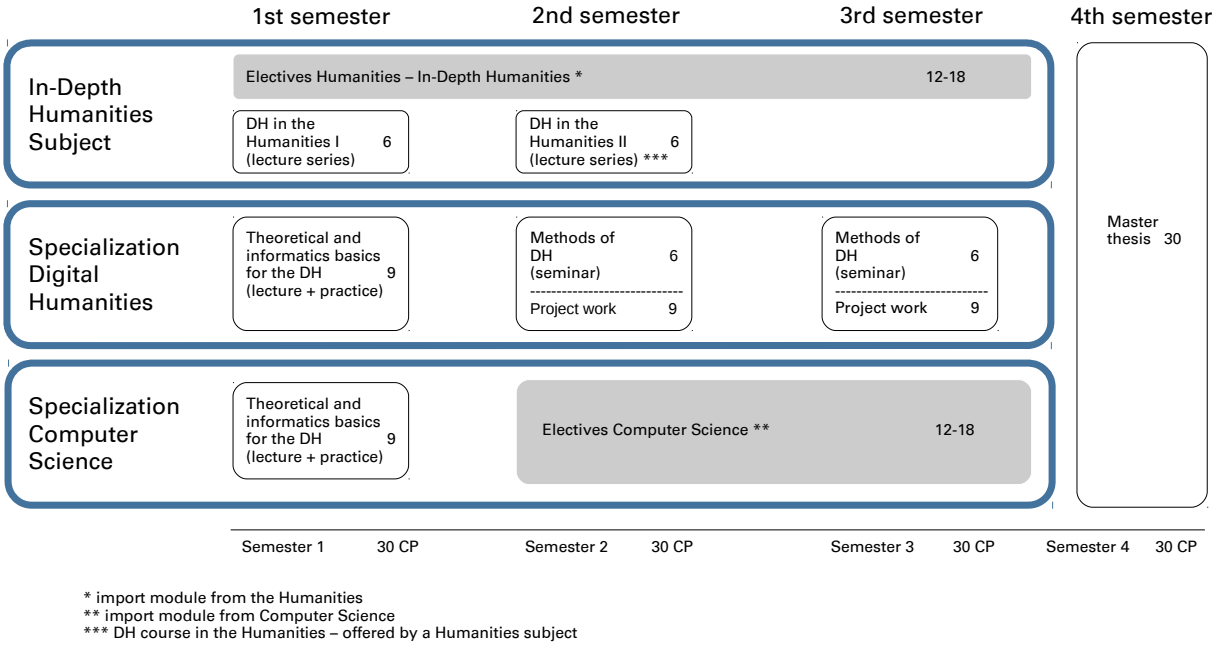

Figure 1: Structure of the master's program Digital Humanities at University of Stuttgart

London cholera outbreak in 1854 (which enabled a visual detection of the outbreak center), or the configuration analysis of 19th century traveling theaters (which enabled a quick overview of the required number of actors to perform a play). As the examples show, formalized approaches do not imply 'big data' or large-scale analyses.

Applications that have been popular in the Digital Humanities (e. g., network analysis or stylometry) are all built on this formalization: Independent of the visualization, a network *is* a formal model, in which data properties are represented by nodes and edges between them. Stylometric analysis, e. g., implies a formalized notion of what tokens are, and how they are counted and compared.

Given the fact that text is a frequently used medium in many Humanities disciplines (on the object and/or meta-level), Computational Linguistics plays a crucial role in two – complementary – ways: (i) On the operationalization level, formalizations of, for instance, literary concepts can be built upon linguistic structures (for which operationalizations do exist). In many cases, this requires tested and proven annotation guidelines as well as implementations of tools for the automatic discovery of such structures – *(computational) linguistic structures* can therefore form the basis of more complex and abstract formalizations (e. g., narrative categories defined on the phrase-level). (ii) On the methodological level, CL has established a number of *best practices for creating such formalizations*, which can be put to use on non-linguistic phenomena. The most prominent example is the annotation workflow, including measuring inter-annotator agreement as a metric for annotation guideline quality (Hovy and Lavid, 2010) or the use of shared tasks to foster tool or corpus creation (Reiter et al., 2017).

3 Structure of the DH Master's Program at University of Stuttgart

Given the above, the DH master's program in Stuttgart aims at both teaching conceptual understanding and practical experience, while at the same time deepening students' Humanities backgrounds and interdisciplinary skills. This is achieved through a combination of theoretical lectures and practical exercises, programming courses, and group projects.

The program is open for undergraduate students of a Humanities discipline that is also taught in Stuttgart (e. g., Literary Studies, History, Philosophy, or Art History). Interested undergraduates may apply once a year and start each year in October. The program is designed to be completed within four semesters. Courses are split into three categories, although not all classes can be clearly assigned: Humanities, Digital Humanities, and Computer Sciences. The structure of the program is illustrated in Figure 1.

In the set of *Humanities courses*, students take

classes in the discipline of their undergraduate program, where they are joined by their non-DH fellow students (e. g., master students of German studies).

In contrast, *Digital Humanities* classes are specific to the DH students, only. After a compulsory introductory lecture (6 hours/week, lecture and exercises), students take part in a group project in the second semester, where 'real-world' research tasks of delimited scope are tackled. Emphasis is put on teamwork and on the independent development of research strategies, two competences we regard as crucial and also characteristic for research in the DH. Thus, students learn to split up a research problem in smaller parts and establish data models that serve as the base for the application of formalized computational methods. Those who choose a CL-oriented project are advised by teachers from Computational Linguistics. Other courses in the DH area are seminars to familiarize students with the most recent research in preparation of their master's theses.

The third area covers *Computer Sciences* and includes the Computational Linguistics courses. In total, these courses cover roughly one third of the credit points each student has to achieve (excluding the master's thesis). Two courses from this area are compulsory, both in the first semester: *Computational Linguistics Methods for Digital Humanities* (6 hours/week; lecture and exercises) and *Programming* (2 hours/week; lecture and exercises). These compulsory courses are designed for and offered specifically to the DH students and are only taken by DH students. Content-wise, *Computational Linguistics Methods for Digital Humanities* resembles introductory courses for students in the computational linguistics programs. In addition, the use of Natural Language Processing (NLP) tools and/or workflows for addressing non-linguistic research questions is covered. In *Programming*, no foreknowledge at all is assumed, treating every student as a first-time programmer. Some emphasis is on the fact that many programming concepts exist in many programming languages, although we use Python (version 3) throughout as our programming language in teaching. The main reason for this is that Python is widely used in the DH community. Many exercises in the programming course cover algorithms and ideas that have been discussed in the NLP-methods course (e. g., to implement functions that measure precision and recall). In general, we aim at performing exercises that students perceive as being related to (Digital) Humanities. Apart from these two compulsory courses, students are free to choose from a selection of courses that are offered in the Computational Linguistics and Computer Science bachelor's and master's programs, in which they share courses with the students from the respective programs (e. g., data visualization).

It is a deliberate choice that DH students take courses that are also offered in the CS and CL programs. This way, students are exposed to different disciplinary styles and cultures, reflecting the 'in between worlds'-nature of DH in general. In addition, many of the courses that feature exercises foster group exercises in order to strengthen team-skills (which are crucial when working across disciplines).

Strong interdisciplinary ties are also present among the teachers involved in the program, who all are experienced in working in mixed teams with members from different disciplines.

4 Evaluation of DH Students' Appraisal

4.1 Methodology

To get an impression of how the conceptional course design decisions are reflected and perceived by the students, we created an online questionnaire and distributed this survey among both cohorts currently enrolled, first and second year students, by the end of the teaching term. Since there were slight adjustments to the courses after the first year such as an emphasis on independent learning and changes to the programming course which was adjusted to the needs of Digital Humanities students, we analyze their feedback separately.

The questionnaire covers topics with respect to the students' overall satisfaction with their choice of study, the differences they perceive between their humanities discipline and the Digital Humanities context, but especially the integration of NLP courses in their curriculum. We inquired their personal attitude towards the practical courses, their assessment of the difficulty of the offered courses, and their opinion about the necessity of the acquisition of NLP-related knowledge and skills for their understanding of Digital Humanities. Appendix A contains the complete questionnaire content.

We distributed 34 questionnaires out of which 15 were returned completely filled out. Since the entire study program has a small number of students and a return of 15 does not allow for a reliable statistical analysis, we rather catch the mood of

approval of the program's structure rather than a full-fledged evaluation.

The questionnaire comprised a few free-text answers, but mostly, participants were asked to mark their personal view of adequateness for given statements on a 6-point Likert scale.

4.2 Results

Firstly, students stated to enjoy their DH studies and both cohorts presume that their future career will profit from their education in Digital Humanities, whereas the second year cohort commits stronger to both statements.

Since all of the students hold a bachelor's degree in a humanities discipline, they emphasize the shift to a more practical, computational training as a clear difference from what they were used to. However, in the first cohort most students stress the addition of CS as a difference, whereas in the second cohort the focus on practical courses/sessions accompanying a theoretical course is mainly mentioned as a difference to earlier studies. A few students point out that sometimes basic knowledge is taken for granted, which leads to excessive demands. These experiences highlight the difficult balance of overload and underload resulting from a very heterogeneous group of students.

This major shift from 'theory' (or more abstract humanist approaches) towards 'practice' is also reflected in the students' expectations of how CL-methods should be taught: Both cohorts agree that practical exercises are a very important aspect (for the second cohort it has highest importance for everyone answering the questionnaire) and some of the students even wish to have more practical training.

However, it seems that the practical exercises should be based on a solid theoretical ground: Students in both cohorts tend to prefer a teaching approach in which theoretical knowledge serves as the basis to these practical sessions rather than an approach in which one is introduced to a topic in a practical manner and later on provided with the theoretical background.

Regarding self-perception and acceptance of CL-skills, our results seem to indicate a characteristic difference between the two cohorts: In both groups, students feel capable of coping with the DH courses in general. But, regarding CL courses, most students in the first cohort feel overwhelmed, whereas the majority of the second cohort does not. Instead, in general they feel equally well as in the DH courses. Students who feel overwhelmed often emphasize the newness of the methods as a reason. Among the students in the first cohort there are also more persons who are not very confident in their ability of familiarizing themselves with a new topic on their own. At the same time, divergent opinions also exist with respect to the question whether they attach importance to a deeper understanding of NLP tools. Even though students seem to agree that an understanding contributes to their abilities in DH, the second cohort in particular tends to find it more essential. The same trend can be observed in their appraisal of the necessity to possess programming skills. The second cohort clearly agrees that programming should be part of their skill set as Digital Humanists, whereas the first cohort has more divided views.

Thus, it seems that a higher confidence in CL-skills also fosters the acceptance of these methods. But, admittedly, it might alternatively just show the inherent difference between the two cohorts.

Being asked about suggestions for improvements for the program, the students wish for even more practical exercises, concrete preparation for their professional life and more diversity with respect to application examples.

In summary, we attribute the differences between both cohorts to the changes we made after the feedback at the end of the program's first year. As an overall reflection of the affinity towards programming, independent learning, and a preference for practical courses, the second cohort has a higher self-perception of skills and also feels more confident to autonomously carry out a project with a topic in Digital Humanities. We interpret this as a sign that our program structure with a focus on practical sessions prospers.

5 Conclusion and Discussion

An often discussed problem of interdisciplinary collaborations between humanists and computer scientists are communication difficulties that can lead to all kinds of misunderstandings, loss of valuable time and frustration on both sides. These issues root in the differences of research traditions and the often opposed way of tackling research objectives. By familiarizing students with both fields and making them aware of these differences, we aim at opening doors to even more fruitful collaborations in the future.

In this study, we recognize a general difficulty in estimating specific needs and issues of Digital Humanities students. The survey that was designed to develop an understanding of particularities of this group revealed that, partially, its characteristics are not different from what one might expect from other discipline switches – we presume that a student changing from a Humanities program to an engineering field would feel similar aspects to be eye-catching (for instance, the combination of lectures and exercises). This might indicate that the difficulties lie not necessarily in the program itself, but in the special combination of Humanities with a formal and more technical research area.

In comparing our teaching experiences in Computer Science/Computational Linguistics and Digital Humanities, another aspect surfaces: CS/CL students are typically confronted with problems new to them (and the accompanying solutions), which is a straightforward way in teaching (from the teacher's perspective). In contrast, students of DH have a background in a Humanities discipline and thus already have been confronted with a number of research questions and possible solution methods. Naturally, they are expecting relatively concrete new solution methods to these diverse, pre-existing questions. This makes DH a more application-oriented subject than many CS disciplines.

References

David Berry. 2011. The computational turn: Thinking about the Digital Humanities. *Culture Machine*.

Alexander Dunst. 2017. Digital American studies: An introduction and rationale. *Amerikastudien*, 61(3):381 – 395.

Fred Gibbs. 2011. Critical discourse in Digital Humanities. *Journal of Digital Humanities*, 1(1).

Eduard Hovy and Julia Lavid. 2010. Towards a 'science' of corpus annotation: A new methodological challenge for Corpus Linguistics. *International Journal of Translation Studies*, 22(1), January.

Fotis Jannidis, Hubertus Kohle, and Malte Rehbein, editors. 2017. *Digital Humanities – Eine Einführung*. J.B. Metzler.

Jonas Kuhn and Nils Reiter. 2015. A plea for a method-driven agenda in the Digital Humanities. In *Proceedings of Digital Humanities 2015*, Sydney, Australia, June.

Todd Presner and Chris Johanson. 2009. The promise of Digital Humanities. Available as white paper online from `http://humanitiesblast.com/Promise of Digital Humanities.pdf`.

Nils Reiter, Evelyn Gius, Jannik Strötgen, and Marcus Willand. 2017. A shared task for a shared goal – Systematic annotation of literary texts. In *Digital Humanities 2017: Conference Abstracts*, Montreal, Canada.

Patrik Svensson. 2012. Envisioning the Digital Humanities. *Digital Humanities Quarterly*, 6(1).

Manfred Thaller. 2017. Digital Humanities als Wissenschaft. In Jannidis et al. (Jannidis et al., 2017).

Appendix A Questionnaire

Below we provide the English translation of the student survey questionnaire. Horizontal rules designate the space for free-text answers. In most cases, students were asked to mark the appropriateness of every statement as shown here:

I disagree					*I agree*
☐	☐	☐	☐	☐	☐

1. I like studying Digital Humanities.

2. My Humanities area: _____

3. The Digital Humanities study program will be helpful for my deliberate professional future (if assessable).

4. This DH study program differs from my bachelor's study program.
   If you (rather) agree, please explain how it differs: _____

5. I feel overwhelmed in DH courses.
   If you (rather) agree, please explain why:
   - ☐ I'm lacking basic knowledge.
   - ☐ Pace of the course is too fast.
   - ☐ The structure of the course is not intuitive for me.
   - ☐ Other reasons: _____

6. I feel overwhelmed in CL courses.
   If you (rather) agree, please explain why:
   - ☐ I'm lacking basic knowledge.
   - ☐ Pace of the course is too fast.
   - ☐ The structure of the course is not intuitive for me.
   - ☐ Other reasons: _____

7. I can contribute with my skills during CL courses.
8. Programming skills are important to me.
9. It is important to me to understand the internal functional principle of computational linguistics tools.
10. Practical modules (like exercises offered additionally to lectures) are important to me.
11. I am confident successfully conducting a hands-on DH project with my skills.
12. I prefer learning the theoretical background before applying it.
13. I prefer learning about hands-on applications before addressing the theoretical background.
14. I can perfectly familiarize myself with a topic on my own.
15. My suggestions to improve the DH study program: _____