Mastering language and computation: how well is teaching of language technology adapted to student learning?

Simon Dobnik

CLASP and FLOV, University of Gothenburg

simon.dobnik@gu.se

January 13, 2019

Abstract

In this paper we investigate self-reported knowledge and skills of students in the interdisciplinary masters programme in language technology where the majority of students have only studied one discipline at the previous level. We focus on the answers of students in Year 1 of the programme and those in Year 2 in order to examine how well are the current teaching strategies adapted to decrease this gap. Comparing the results with the individual reflections of the author of this study, who is a teacher in this programme, we make recommendations about improving the teaching strategies in the programme.

1 Introduction

Masters in Language Technology (MLT) at Gothenburg University is a one or a two years international masters programme in computational linguistics, natural language programming or language technology (LT), a field that is at the intersection of formal theoretical linguistics, psychology, cognitive science and computer science. The programme covers acquisition of knowledge and skills from several different and distinct disciplines where only a minority of students would have received training on all of them at the 1-cycle undergraduate level. This means that students starting the programme have different academic backgrounds and one the major challenges of the programme is to bring all the students to the same academic level in a cross-disciplinary field, particularly during the first academic semester.

There is some informal evidence that this factor has a high impact on student performance in the programme. For example, students with computer science background are believed to master formal theoretical linguistics faster than students with linguistic background master computer programming. However, this has never been systematically evaluated within the programme nor the programme has taken on systematic adaptations following results of any such recommendations, although several adaptations have been made based on the experience of the teachers.

This task of this paper is investigate to what the degree the current teaching in the MLT programme is adapted to student learning, in particular in the early semesters of the programme where students are put under considerable stress. The recommendations for the adaptation of teaching will be made on the basis of a self reported questionnaire (multiple-choice and free-answer) about knowledge and skills of students in the introductory courses in each field (linguistics and computer science): Introduction to formal linguistics (linguistics is an advantage), Introduction to programming in Python (computer science is an advantage), Natural Language Processing (both are an advantage), compared with the knowledge and skills of students

in the second year towards the end of the programme. The results of the investigation will be interpreted and reflected against the individual teaching experience of the author. Overall, the investigation will give recommendations for teaching and structuring the content of the programme that should lead to better student learning experience.

2 Previous work

(Trigwell, 2001) describes that "good teaching" at the university level is based on two pre-requisites: (i) conceptual competence of the teachers and (ii) orientation towards student learning which can be achieved by the adoption of *constructive alignment* (Biggs, 1999; Biggs and Tang, 2011). Constructive alignment is achieved when the content of the teaching (learning objectives), the nature of the teaching (teaching methods) and the assessment of the teaching are aligned with each other which gives a consistent experience to the students. By "constructive" it is meant that "meaning is not imposed or transmitted by direct instruction, but is created by student's learning activities" (Biggs, 1999, p.60). Therefore, in order to facilitate student learning, good teachers, according to Trigwell (2001), need to be able to (i) adapt to the context, (ii) encourage learning, understanding and development of skills, (iii) learn from their interactions with students and peers, (iv) adapt the content from their professional expertise to match the learning objectives and the needs of the students, and (v) apply appropriate teaching strategies to this end. As teaching and learning are intertwined under this view we will refer to the process as teaching/learning or teacher learning activities (TLAs) (Biggs, 1999).

In order to give students this kind of support it is therefore crucial that teachers understand differences between their students. For example, Felder and Brent (2013b) investigate three different categories of diversity among students that affect the teaching/learning context: (i) students have different *learning styles*, (ii) they adopt different *approaches to learning*, and (iii) they have different *intellectual levels*. They also argue strongly that a significant part of academic difficulties that some students have in particular programmes (in this case engineering) can be attributed to the fact that type of instruction is not in line with these three categories of student diversity. While it is difficult (if not impossible) to adapt teaching to every single student in terms of these categories, *constructive alignment* of learning should include a balanced approach to each of the factors contributing to (i)–(iii) as well as this will develop skills of students that will be later required by their careers. *Learning styles* relate to how students interact with the learning situation, for example do they focus on learning abstract knowledge in the forms of the theories or do they base their learning on observations from data. Following (Marton and Säljö, 1976), Felder and Brent (2013b) define three *approaches to learning*: a surface approach, a deep approach and a strategic approach. Finally, *intellectual levels* describe the intellectual development of students and their attitude towards knowledge. Felder and Brent (2013b) (also (Felder and Brent, 2013a)) discuss four different models and compare them according to levels (from lowest to highest development): absolute knowing, transitional knowing, independent knowing and contextual knowing. The current study will mainly focus on this aspect of differences between students, exploring how to face the challenge of bringing a balanced intellectual growth of students that are partially at different intellectual levels. It will also focus on differences in learning styles between students in as far as these are the reflections of the skills acquired during their previous academic training, e.g. programming, analysing data, writing a research paper, etc. To adapt to these differences between students, (Felder and Brent, 2013b, p.67–68) conclude that teaching should involve a variety of learning

tasks, promote inductive approaches to teaching/learning such as problem-based learning (PBL), explicitly communicate expectations and provide feedback, and show respect for students at all levels of development.

3 Hypotheses

H1 Students from different subject areas have different needs for acquiring knowledge (intellectual levels) when they start the programme but these differences decrease when they are close to the completion of the programme.

H2 Students from different subject areas have different skills when they start the programme but these differences decrease when they are close to the completion of the programme.

4 Method

4.1 Task and procedure

We created an online questionnaire to be answered by students at different stages of the programme (Year 1, Year 2 and Other) attached in Appendix A. The questionnaire consists of questions where a single answer can be chosen from a given list of options (for example Question 1), questions where longer textual answers can be entered (for example Question 3), and questions where a statement can be judged on a 7-point Likert scale from 1 (low) to 7 (high). Several questions that invite making a choice were accompanied by a text field where participants could give additional comments on the issues in the questions. As Questions 12 to 13 are intended to evaluate the opinions and experiences of students in the latter stages of the programme, they were hidden to those participants who have initially answered that they are Year 1.

4.2 Participants

In mid-November 2018 the questionnaire was sent to 45 students registered on the programme. The questionnaire differentiates between three categories of studies: (i) active students who have started the programme in September 2018 and are in Year 1 of the programme, (ii) active students who have started the programme in September 2017 and are in Year 2 of the programme; (ii) active or non-active students who have been studying in the programme longer than 2 years. The survey was answered by 21 respondents (46.7% response rate) whereby 8 respondents (38.1%) are from Year 1, 7 (33.3%) respondents from Year 2 and 6 (28.6%) from other years. The response rate may appear small but it important to note that MLT programme is taught as small classes of active students. For example, the overall number of students in Year 1 is 18 and in 8 in Year 2.

Students start the programme with very diverse subject areas of their 3-year undergraduate studies. According to the the admission requirements to the programme[1] these can be language technology, computational linguistics, computer science or linguistics (with at least 30 hec in formal linguistics or programming) but also cognitive science, languages, philosophy, software engineering, information technology or mathematics (with at least 30 hec in programming or formal linguistics).

Students in Year 1 are half way through their first semester. In the first study period (September and October) they have attended two introductory courses that are intended to consolidate their existing knowledge and skills: (i) LT2001 Introduction to Programming for Natural Language Processing, and (ii) LT2002

[1] `https://flov.gu.se/english/education/masters-second-cycle/mlt/entrance+requirements`.

Introduction to Formal Linguistics.[2] These courses give a quick introduction to students in the areas that they may be new to them. For example, students who have studied computer science in the 1-cycle are new to formal representations in linguistics and students who have studies formal linguistics may be new to programming. In the past the courses were organised differently so that students only take the introductory course that they had little experience in. However, it was later decided on the current setup based (i) on the experience that students overall have varied academic backgrounds anyway and (ii) on the consideration that students would learn from in other in the courses where they would contribute with complementary knowledge. It was felt that separating the students into two groups would further contribute to the divide. The observational evidence shows that students find the course or the part of the course that they are new to very challenging and also that a background in linguistics (but less so for computer science) is not a pre-requisite to do well in the corresponding course since this discipline is taught very differently at 1-cycle and As a result students have a limited exposure to *formal linguistics* which builds mathematical models of language that can be directly implemented in computational applications. At the time Year 1 students have completed the questionnaire (November) they have been studying LT2003 Natural Language Processing course for two or three weeks which is the first introductory course in language technology and therefore requires a mixture of knowledge and skills from different subject areas. Hence, unless the students have studied language technology at an undergraduate level the topics will be new to most students in the course.

Students in Year 2 are in the second half of the 3rd semester. In their 2nd semester of Year 1 they have completed the obligatory courses on the core topics in language technology and the 3rd semester they have studied elective courses which allow a certain element of choice. Hence, by this time in the programme they are expected to have acquired both the core theoretical and applicable knowledge and skills of language technology with a high emphasis on practical implementation. The time of the questionnaire also coincided with them making a decision on their master thesis project which they will start in their 4th semester. Hence, at this time they are evaluating their knowledge and skills that will allow them to embark on an individual research projects.

Finally, the students that we group in the category Other are those who still have missing courses either from Year 1, Year 2 or both which are required to take out their degree. However, it is false to assume that students in this category have not completed their degree because of lack of skills and knowledge. Several have slowed down their studies because they have started their jobs, frequently related to language technology which is a sought-for skill among the employers in the local environment.

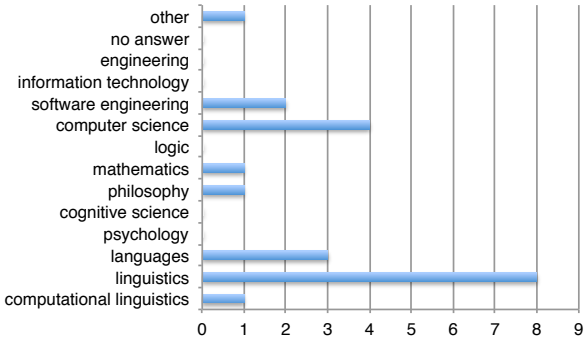
The MLT programme uses a wide-variety of forms of teaching: (i) lectures, (ii) seminars, (iii) classes with exercises, (iv) programming labs, and (v) individual supervision which are closely related to the diverse knowledge and skills required for language technology (see Section 5.1 and 5.2). Similarly diverse are forms of examination: (i) written exams, (ii) hand in exercises, (ii) programming labs, and (iii) larger course projects which required both programming and writing a project report.

5 Results

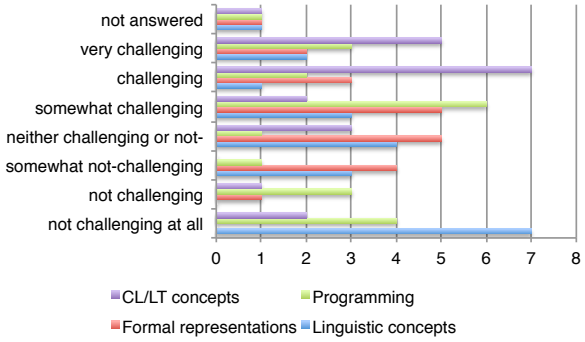
5.1 Academic background of students and their knowledge and skills starting the programme

As shown in Figure 1a the responses from the students (Question 2, $n = 21$ all responses) indicate that students do have highly variable academic background entering the programme encompassing several dif-

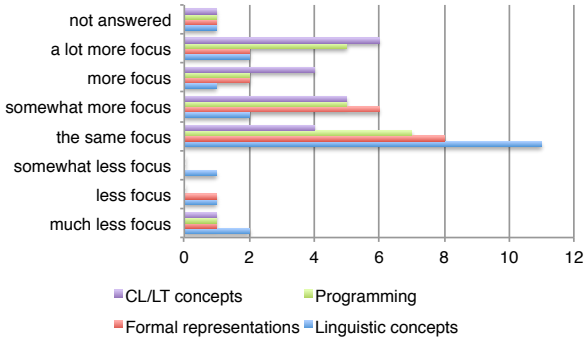
[2] `https://flov.gu.se/english/education/masters-second-cycle/mlt/programme-structure`



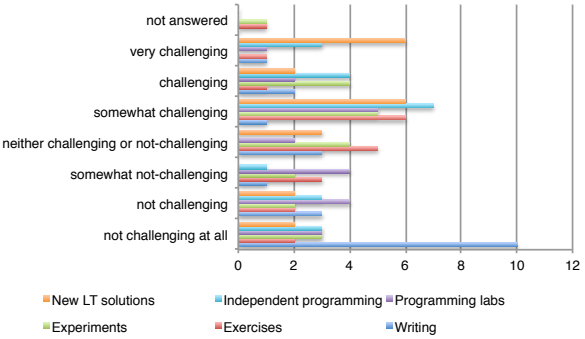
(a) Subject of the undergraduate degree



(b) Knowledge at the start



(c) Desired focus on knowledge



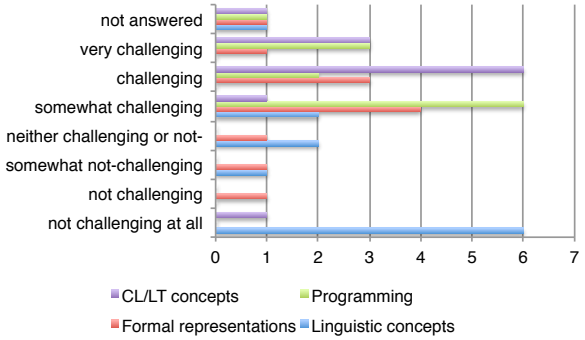
(d) Skills at the start

Figure 1: Academic background of students starting the programme ($n = 21$, Year 1, Year 2 and Other)

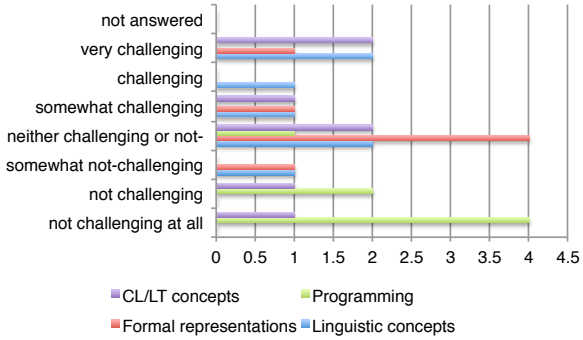
ference disciplines. The disciplines are ranked according to their relatedness (except computational linguistics which is a category related to all). The counts show a bi-modal distribution. The dominant mode is linguistics while a secondary mode is computer science. Students have therefore background in either dominant disciplines for language technology, with only one student having background in both (computational linguistics or language technology).

With the intention to shed light on their skills when entering the programme Question 6 asked the students how challenging they found the introductory courses in the first semester of the programme (i) Introduction to formal linguistics (ii) Introduction to programming, and (iii) Natural language processing in terms of the topics and methods that they introduce. These included: (i) *linguistic concepts*, (ii) *formal representations from linguistics, mathematics, logic and statistics*, (iii) concepts and methods related to *computer programming*, and (iv) formal theories and algorithms in *computational linguistics*. The participants indicated their answers by choosing an option on a Likert scale where is 1 not challenging at all and 7 extremely challenging. By challenging we mean that the background of the topic was new to them and that they had to do extra background reading in order to be able to understand the lectures or they needed extra help to complete the labs. By not challenging we mean that they have already covered these topics to the same extent in their previous Bachelor/1-cycle courses.

The results *from all respondents* are shown in Figure 1b. The chart indicates that there is quite a variation in the topics and methods that students found challenging when starting the programme and that there is no clear separation in terms of missing knowledge when they are considered as a group. More closely, there appears to be a bias for linguistic concepts to be considered less challenging (but for a few students they have been very challenging) and there is also a bias towards concepts related to computational linguistics/language technology (CL/LT) to be considered challenging (but for a few they have been not



(a) Linguistics related background ($n = 12$)



(b) CS related background ($n = 7$)

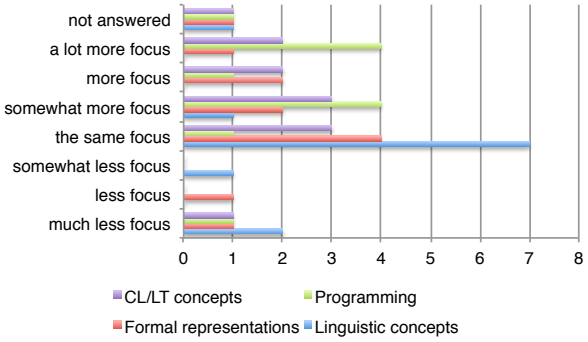
Figure 2: Knowledge at the start of the programme (Year 1, Year 2 and Other)

challenging at all). On the other hand, programming and formal representations are more spread across all response categories and they centre around neither challenging or not challenging or somewhat challenging.

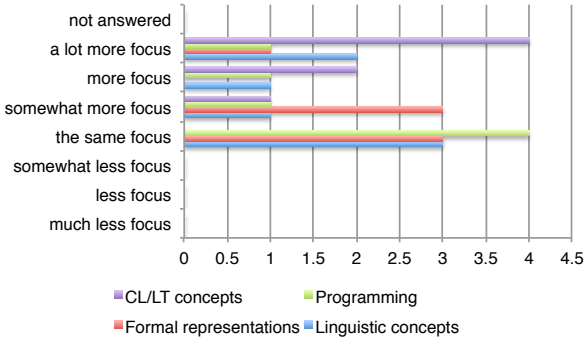
Question 7 focuses on what knowledge students believe should be covered in more detail in the three introductory courses. As shown in Figure 1c most students would prefer the same or more focus on all of the topics. Most believe the amount of linguistic topics receives sufficient focus in the courses and they would like somewhat more focus on formal representations, but a lot more focus on programming and concepts in language technology. As in the previous case, the answers are quite spread between different degrees of positive focus.

Question 8 focuses on the skills of students required for language technology when they have entered the programme. Figure 1d shows that (a) writing longer textual answers in English is not challenging for most students and that (f) independently finding a new way how to solve a particular language technology problem with a computer programme is on the very challenging side. This is expected: students entering the programme should have good English communication skills, while topics in language technology are those that most of them have not studied before. There is also wide distribution of other categories ranked roughly in the order of the challenge they represent: (b) solving practical exercises on paper, (c) performing experimental work including data collection and evaluation, (d) implementing code in a programming language within a structured lab (where some code is already given), and (e) implementing code independently, including pre-processing of data. This demonstrates that students start the programme with very different skills which consequently also affects their learning styles.

Overall, the results indicate that students entering the MLT programme have very diverse knowledge and skills and that they feel that the courses should give more emphasis to several diverse areas which is a challenge for teachers of these courses. An interesting and relevant question to plan teaching in these courses is whether this diversity is general or could it be linked to the academic background of students. If the latter is true then systematic changes to the programme should be made. The disciplines in Figure 1a can be split into two groups: (i) those related to linguistics (Ling) which include linguistics, languages, psychology, cognitive science and philosophy), and (ii) those related to computer science (CS) and include mathematics, logic, computer science, software engineering, information technology, and engineering. Grouping the students into these two classes creates two subsets of respondents: (i) those with linguistics related background ($n = 12$) and (ii) those with computer science related background ($n = 7$). Note that these subsets do not include two students: a student that has studied language technology and a student that has studied another subject at the undergraduate 1st-cycle level.

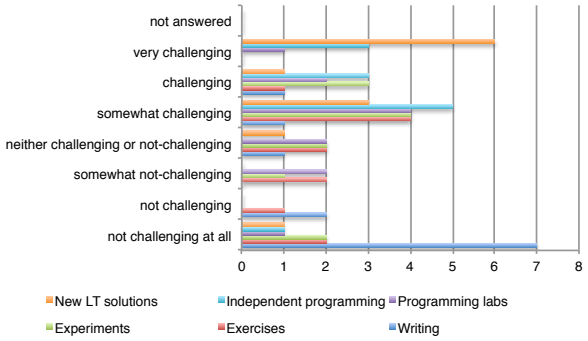


(a) Linguistics related background ($n = 12$)

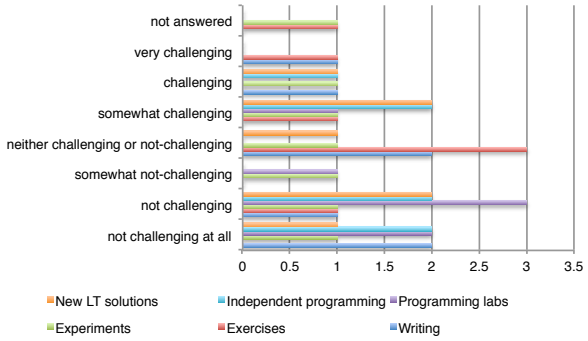


(b) CS related background ($n = 7$)

Figure 3: Desired focus on knowledge at the start of the programme (Year 1, Year 2 and Other)



(a) Linguistics related background ($n = 12$)



(a) CS related background ($n = 7$)

Figure 4: Skills at the start of the programme (Year 1, Year 2 and Other)

Figure 2 shows the self-assessed student knowledge between these two groups. There are clear differences: the Ling group does not find linguistics concepts very challenging, formal representations are more challenging but most challenging are programming and concepts from language technology. On the hand, the CS group does not find programming very challenging and can get by in formal representations but concepts from linguistics and language technology are challenging to them to a varying degree, including very challenging. Therefore, linguistics appears to be a less challenging topic overall in Figure 1b because there is a bias to students with the Ling background (12 vs 7).

In Figure 3 we examine the desired knowledge of respondents from the two broad subject areas after starting the programme. It turns out that Ling would like to keep the same focus on linguistic topics and CS would like to keep the same focus on computer programming. Both in the Ling and the CS group formal representations should receive more focus but the degree to which this should be done diverges much more with the Ling group than the CS group. A possible explanation for the diversity in the Ling group is that students of linguistics and related disciplines receive quite a varied undergraduate education in linguistics, sometimes strong on formal methods and representations and sometimes not. It is also interesting that the Ling group shows quite a variation in the desired focus on CL/LT concepts in the introductory courses while the CS group would like a lot more focus. We would expect this to be the case for both groups equally as the subject is equally new to both. This might be explained by different expectations of students what language technology is. At this introductory level a lot of concepts and methods from language technology are closely related to formal representations and therefore the Ling group may consider some of them as formal representations and the CS group may consider some formal representations as CL/LT concepts.

Finally, Figure 4 shows the self-assessed skills of students from the Ling and the CS groups when entering the programme. Writing is considered as not challenging at all by the Ling group (although there are some who find it challenging), while finding new language technology solutions is most challenging, followed roughly by independent programming, programming labs, experiments and exercises. On the other hand, the CS group shows more variation: programming labs is considered as less challenging, writing, independent programming and language technology solutions are spread between the challenging and non-challenging categories. Experiments are placed in every category and exercises are mainly considered as neither challenging or non-challenging. The variation in the CS group shows that their skills are less coherent than those in the Ling group, perhaps because this represents several related disciplines that are linked to language technology in different ways. The Ling group appears more coherent in terms of skills because these disciplines are more closely related, they are about the study of language from different perspectives.

Overall, Figures 2, 3 and 4 indicate that there is a perceived difference between the Ling and CS groups in terms of knowledge and skills when entering the programme. A distinction that emerges is that students in the Ling group are good at studying language but they lack computational technical knowledge and skills, in particular programming while the students in the CS group are good at this kind of knowledge and skills but are frequently not sure about knowledge and skills related to computational modelling of language. In a way this result is expected as this is the focus of disciplines that have been previously studying but the preceding discussion also points out finer differences in knowledge and skills of students from both areas and how they interact with the requirements of language technology that are relevant for the future planning of the courses.

5.2 Academic background of students and their knowledge and skills towards the end of the programme

The purpose of Question 12 is to establish if there are differences in student knowledge and skills required for language technology in the second or later year of the programme. Note that the question focuses only on knowledge skills related to language technology and not to knowledge and skills in linguistics, computer science and language technology as previously. In order to limit their scope of the expectations we explicitly refer to the knowledge and skills required to complete their thesis (rather than, e.g. future career). The tasks of respondents is to evaluate several statements on the Likert scale from 1 to 7 where 1 is not true, 4 is neither true or false, and 7 is completely true. The statements are as follows: (A) The preceding courses gave a me a good and sufficient background to find and solve an interesting language technology problem for my thesis; (B) I am able to do data processing and programming for my thesis on my own without a difficulty; (C) I am able to analyse and critically evaluate my results; (D) I am able to present and discuss my work in a scientific manner; (E) I can follow a seminar on language technology and I can have a discussion with a colleague in the field; (F) I can follow a seminar in formal linguistics or have a discussion with a linguist; (G) I can follow a seminar in computer science on machine learning (not necessarily on natural language processing) and have a discussion with a computer scientist.

The number of participants that answered this question is 13 (7 from Year 2 and 6 from Other). Figure 5a shows the combined answers for all respondents. Answers A to E are skewed towards agree and therefore the overall conclusion is that generally students have acquired knowledge and skills required to complete the programme in language technology: identifying problems (A), programming and data processing (B), analysing and evaluating results (C), scientifically presenting work (D) and engaging in scientific discourse

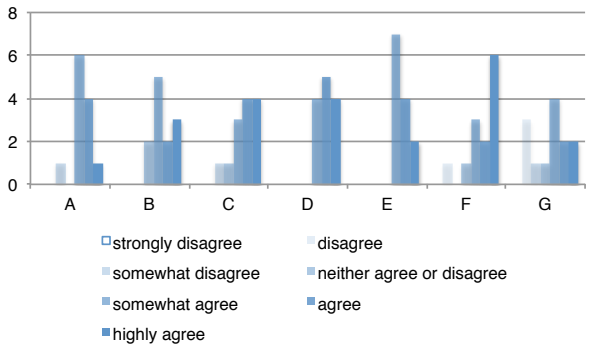
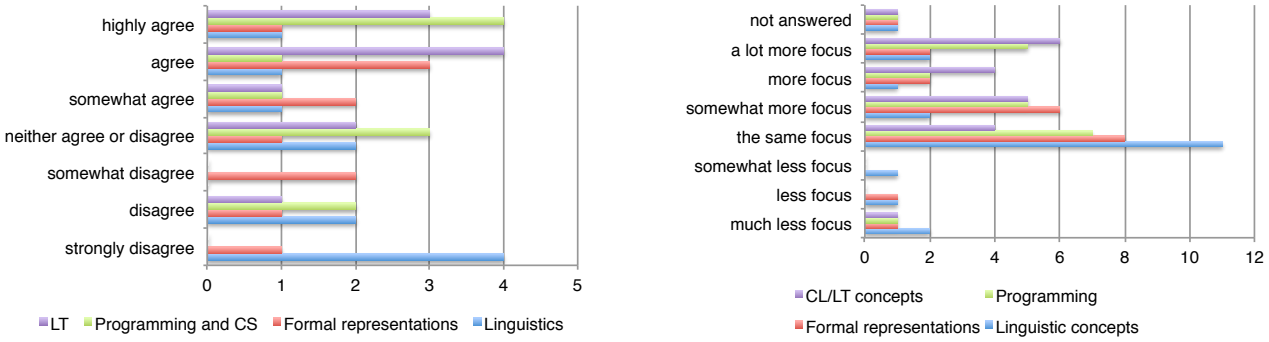


Figure 5: Final knowledge and skills ($n = 13$, Year 2 and Other)



(a) End ($n = 13$)

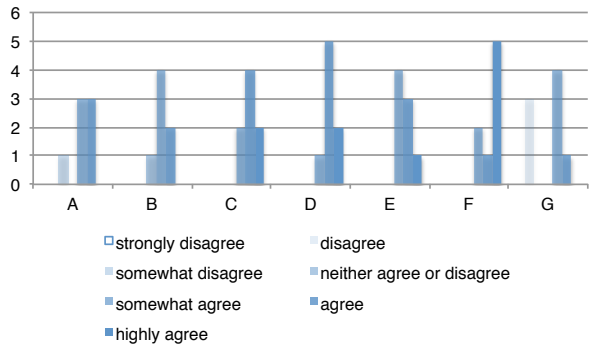
(b) Beginning ($n = 21$) from Figure 1c

Figure 6: Comparison of desired knowledge required to complete the thesis with the desired knowledge at the beginning of the programme

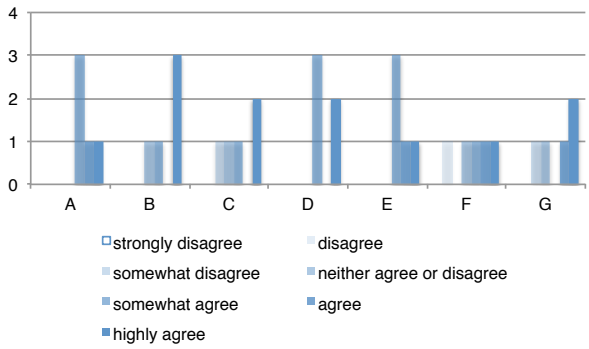
(E). One respondent somewhat disagreed with statement A and one with statement C. Statements F and G ask to what degree they could engage in a scientific discourse in individual fields from which they have started from or towards the other complementary discipline (linguistics = F, and computer science = G) where similar results emerge as in the previous section but the lowest score is only "somewhat disagree". Overall, it appears the students feel more confident about language technology as well as the individual related disciplines.

Question 13 asks about the knowledge and skills students wish they had better before starting the thesis formulated as a series of statements: (i) Better understanding of facts about language; (ii) Better understanding about formal representations used; (iii) Better understanding of programming and computer science; (iii) Better understanding of language technology. The answers are shown in Figure 6. This shows that most students are comfortable about knowing about the linguistic facts but they wished they had improved understanding of all 3 other categories: language technology, programming and computer science, and formal representations. There is also considerable variation. Recall that in Question 7 (Figure 1c) we asked participants about what knowledge they believe should be covered more in the three introductory courses which we replicate here in Figure 6b. Although the focus of questions is slightly different we can see that there are surprising similarities between the answers. This therefore suggests that although students have more knowledge and skills than at the start of the programme, the differences in real or perceived lack of knowledge and skills is still present at the end of the programme.

In the next part we examine the answers to Question 12 and Question 13 from the Ling and CS subgroups of students in relation to the two major sub-disciplines of the preceding undergraduate education. Figure 7



(a) Linguistics related background ($n = 8$)



(a) CS related background ($n = 5$)

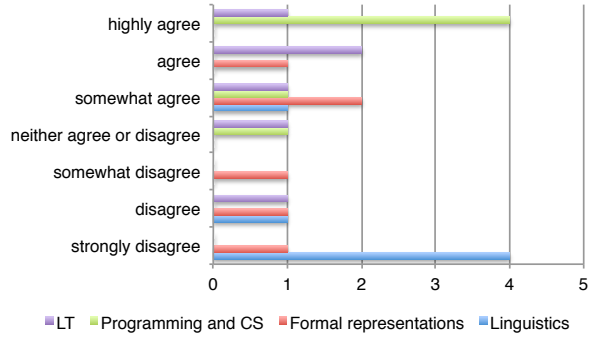
Figure 7: Final knowledge and skills (Year 2 and Other)

compares their answers to Question 12. Both groups somewhat agree/agree that the preceding courses gave them sufficient background knowledge to solve a language technology task for their thesis, the answer being slightly weaker with the Ling group (A). The CS group agrees more strongly that they are able to do data processing and programming for their thesis independently (B). On the other hand, the Ling group more coherently agrees that they are able to analyse and critically evaluate their results (C) and that they are able to present and discuss their work scientifically (D). They both more or less coherently agree that they are able to follow a language technology seminar or discussion (E). The CS groups shows a great diversity in terms of understanding and engaging in conversation in linguistic seminars (F) and computer science/machine learning seminars (G), whereas the Ling group is confident in the linguistics scientific discourse (F) but not computer science (F). Therefore, although overall students feel confident about their language technology knowledge and skills there are still individual differences between students of the two overarching groups depending on their previous undergraduate education.

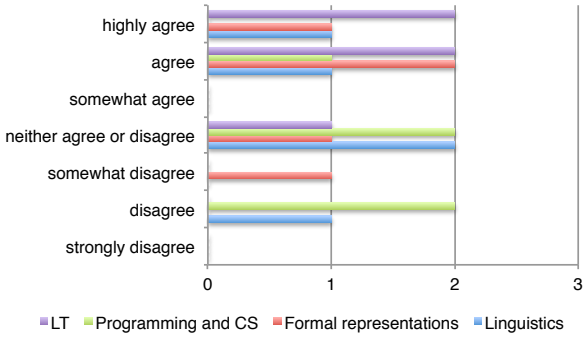
The results for Question 13, on the knowledge and skills the students wish they had when starting their thesis, in Figure 6 are expanded per two undergraduate subject groups in Figure 8a and b. There are clear differences between the two groups. The Ling group is confident in their knowledge and skills of facts about the language but they wish they had better knowledge and skills of programming and computer science and of language technology. On the other hand, the CS group is satisfied with their knowledge and skills of programming and computer science and also facts about language but they would desire more knowledge on formal representations and language technology. Figure 8c and d repeat the results from Figure 3a and b which show the beliefs of participants what topics should be covered more in introductory courses. The results indicate that the similarities identified in Figure 6 between desired knowledge and skills at the beginning of the programme and those at end of the programme are also persistent at the level of the individual subgroups.

6 Discussion

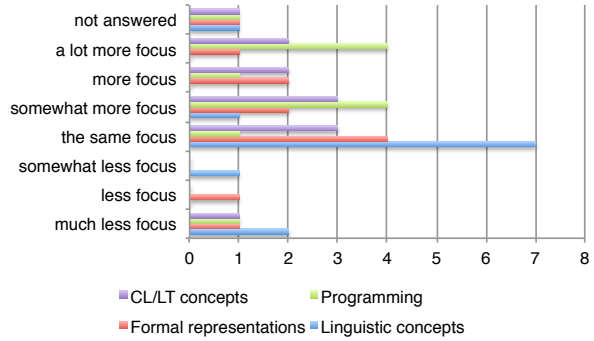
Our hypotheses in Section 3 were as follows: (H1) students from different subject areas have different needs for acquiring knowledge (intellectual levels) when they start the programme but these differences decrease when they are close to the completion of the programme; and (H2) students from different subject areas have different skills (which will be reflected in different learning styles) when they start the programme but these differences decrease when they are close to the completion of the programme.



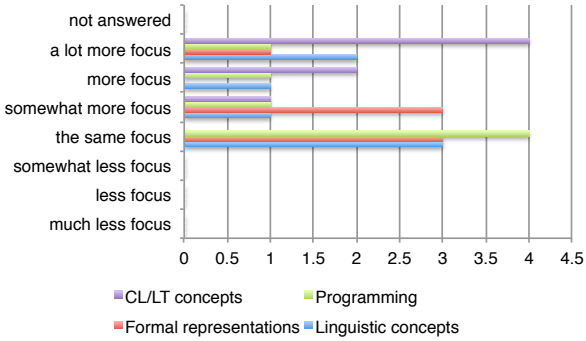
(a) Ling, end ($n = 8$)



(b) CS, end ($n = 5$)



(c) Ling, beginning ($n = 12$) from Figure 3a



(d) CS, beginning ($n = 7$) from Figure 3b

Figure 8: Comparison of desired knowledge required to complete the thesis with the desired knowledge at the beginning of the programme per subject group

The results and discussion in the previous section (Section 5) indicate that these cannot be confirmed. The effect of a different academic background is reflected in differences in self reported knowledge and skills at the beginning of the programme as well as towards the end of the programme. Although these differences in self reported knowledge and skills exist it is important to note that students with different background can cope and successfully finish the programme which requires acquisition of both of these, Section 5.1 and 5.2) and get employment in the same field. In terms of (Felder and Brent, 2013b) they must all achieve *contextual knowing* in respect of their intellectual development. This means, however, that students have different focus in terms of acquiring knowledge and learning styles (as reflected in the acquisition of skills) through their studies and also different learning challenges. The results also indicate that training at undergraduate first-cycle studies has a strong effect on how students will acquire knowledge and skills in their later studies. In order to achieve a *constructive alignment* between teaching and learning (Biggs, 1999; Biggs and Tang, 2011) these should be taken account also in terms of teaching as this will facilitate the learning experiences of individual students and achieve a higher quality of education.

According to (Trigwell, 2001) good teaching is adaptable to learning contexts (Section 2). However, in this case the adaptation to different contexts is somewhat conflicting. One solution to overcome the differences in student experience is to separate students in the two main groups (Ling and CS) for the introductory courses and teach them only one of the introductory courses: either formal linguistics or computer programming but for twice the size. Indeed, this strategy has already been tried and in the early years of the programme but it was later abandoned as students did not like that a small group was further separated, especially at the beginning of the programme where interaction between students is crucial. This suggests that a better solution is to allow students to catch up on the knowledge and skills that are new

to them at the beginning of the programme *before* they start the programme and allow them a smoother transition. For example, the programme could publish a list of suggested readings or for programming a list of online courses that students can follow in their own time before starting the programme. Indeed, some students already contact the lecturers or current students out of their own initiative to request such materials. This also means that more teaching resources should be directed towards individual tutoring where students are assigned individual advisers or tutors which is in line with the recommendations of (Felder and Brent, 2013b) discussed in Section 2. The results indicate the knowledge and skills of students from different backgrounds are really in complementary distribution. The programme could use this to organise learning activities so that students with complementary knowledge and skills are paired together and learn from each other. This is also a recommendation of (Trigwell, 2001) discussed in Section 2. One immediate solution is to organise student mentorship where students from the second year with complementary knowledge and skills mentor students in the first year. Another solution is to ensure a close collaboration between students from different subject areas from the start of the course. However, this means that significant changes to the organisation of the initial courses should be made since for example, learning facts about language and programming should be intertwined. A good example of such pedagogical practice is (Bird et al., 2009) which is not only a textbook about programming but also introduces linguistics structures and language technology solutions in parallel. However, it is not used in our programme in this way. The same approach could be extended to all other courses we offer so that the acquisition of knowledge and skills from different strands would be parallel but also continuous throughout the two years of the programme. An important pre-condition for this is that we know more about the knowledge and skills of students when they start the programme as well that we monitor them throughout. To our knowledge this study is the first attempt to evaluate them for the MLT programme and the results at a detailed level are useful for fine-tuning and the development of the programme.

7 Conclusions and future work

In this paper we have investigated self-reported knowledge and skills of students in the MLT programme in relation to their previous subject background with the purpose to examine whether the differences between the subject backgrounds diminish as the programme progresses. The results indicate that although the students successfully master language technology skills towards the end of the programme, their desired knowledge and skills at the end of the programme still align with those at the beginning of the programme. As these are in complementary distribution, a suggested improvement is to re-structure the courses in a way that knowledge and skills from different areas are learned in parallel in closer collaboration between students.

We focused on self-self reported *intellectual level* of students and partially on *learning styles* insofar these are reflected in skills. However, the questionnaire also included specific questions about *learning styles* (Questions 9, 10 and 11) and *learning motivation* (Questions 3 and 14), the discussion of which we leave for future work. Finally, we emphasise that the results presented here are of a descriptive nature and based on the answers of 46.7% of students on one particular occasion. To have firmer results that could be supported with statistical testing we would need a repeated longitudinal study. This may be possible if such an investigation is integrated with the regular course procedures.

References

Biggs, J. (1999). What the student does: teaching for enhanced learning. *Higher Education Research & Development 18*(1), 57–75.

Biggs, J. B. and C. Tang (2011). *Teaching For Quality Learning At University*. Maidenhead, UK: McGraw-Hill Education.

Bird, S., E. Klein, and E. Loper (2009). *Natural language processing with Python*. O'Reilly.

Felder, R. M. and R. Brent (2013a). The intellectual development of science and engineering students. part 1: Models and challenges. *Journal of Engineering Education 93*(4), 269–277.

Felder, R. M. and R. Brent (2013b). Understanding student differences. *Journal of Engineering Education 94*(1), 57–72.

Marton, F. and R. Säljö (1976). On qualitative differences in learning: I – outcome and process. *British Journal of Educational Psychology 46*(1), 4–11.

Trigwell, K. (2001). Judging university teaching. *International Journal for Academic Development 6*(1), 65–73.

Appendix A: Questionnaire

[Last name], [First name]
[Street address]
[Zip code] [City]

Dear [First name],

Many thanks for participating in a survey about teaching and learning in the Masters in Language Technology Programme (MLT) at Gothenburg University.

The survey focuses on your experiences and thoughts as a student at different stages of the programme. It contains structured questions where you can either pre-select an answer or evaluate the validity of statements on a scale from 1 (low) to 7 (high). In addition, several questions can also be answered by free textual answers where you can explain your answers in greater detail or raise additional points.

The author of this study is Simon Dobnik (simon.dobnik@gu.se). The data collected in this questionnaire will be used in two ways: (i) as a basis for a scientific report/publication; (ii) to identify future policies about the programme. The answers will be collected and aggregated in a way so that it will not be possible to identify identify you as a person and will be handled according to the data protection legislation in Sweden.

Thank you in advance for your time and feedback.

Simon Dobnik

Associate Professor in Computational Linguistics, University of Gothenburg

---

How to fill in the paper survey
Below you can see how you mark an answer option in the check boxes, and how you change a selection.

 The answer option has been marked correctly

 The answer option has been marked incorrectly, the cross must be in the middle of the box

 The answer option has been marked incorrectly, the cross is too strong

 Changed selection, the answer option will <u>not</u> be counted as being marked

1. What year of the programme are you in?

- [ ] First
- [ ] Second
- [ ] Other

2. What is the main subject of your first degree, before MLT?

- [ ] computational linguistics / language technology
- [ ] linguistics
- [ ] languages
- [ ] psychology
- [ ] cognitive science
- [ ] philosophy
- [ ] mathematics
- [ ] logic
- [ ] computer science
- [ ] software engineering
- [ ] information technology
- [ ] engineering
- [ ]
- [ ] If other, please specify

Comment

3. Why did you decide to study language technology?

4. Where did you hear about our programme?

- ☐ I attended ”Visiting Day” at University of Gothenburg and learned about the programme then
- ☐ I visited the departments ”Open house” event in March and learned about the programme then
- ☐ A student counsellor informed me about the programme
- ☐ I googled
- ☐ I found information about the programme on Facebook
- ☐ I saw an advert about the programme
- ☐ I heard about the programme from a friend
- ☐ I heard about the programme from a sibling (or other relative)
- ☐ I looked for information about the programme at www.antagning.se
- ☐ I looked for information about the programme at University of Gothenburg’s website
- ☐ If other, please specify

5. Why did you choose MLT at GU?

6. In the first semester of the MLT programme you had courses Introduction to formal linguistics, Introduction to programming and Natural language processing. Evaluate how challenging were the following tasks for you in these courses. By challenging we mean that the background of the topic was completely new to you and that you had to do extra background reading in order to be able to understand the lectures or you needed extra help to complete the labs. By not challenging we mean that you have already covered these topics to the same extent in your previous bachelor courses.

	not challen-ging at all	not challen-ging	somew-hat not-cha-llengin-g	neither challen-ging or not-cha-llengin-g	somew-hat challen-ging	challen-ging	very challen-ging
Understanding linguistic concepts, e.g. phones and phonemes, morphs and morphemes, parts of speech, syntactic trees, entailment, presupposition, implicature, etc.	☐	☐	☐	☐	☐	☐	☐
Understanding formal representations, e.g. waveforms, maths and statistics, parts of speech tags, finite state machines, context free grammars, features and unification, logical forms	☐	☐	☐	☐	☐	☐	☐
Understanding concepts and methods related to computer programming	☐	☐	☐	☐	☐	☐	☐
Understanding formal theories and algorithms in computational linguistics: e.g. unification, tagging, parsing, Bayesian probabilistic models, machine learning and classification, theorem proving for inference	☐	☐	☐	☐	☐	☐	☐

Comment

7. Which of the topics do you think should be covered in more detail?

	much less focus	less focus	somew-hat less focus	the same focus	somew-hat more focus	more focus	a lot more focus
Understanding linguistic concepts, e.g. phones and phonemes, morphs and morphemes, parts of speech, syntactic trees, entailment, presupposition, implicature, etc.	☐	☐	☐	☐	☐	☐	☐
Understanding formal representations, e.g. waveforms, maths and statistics, parts of speech tags, finite state machines, context free grammars, features and unification, logical forms	☐	☐	☐	☐	☐	☐	☐
Understanding concepts and methods related to computer programming	☐	☐	☐	☐	☐	☐	☐
Understanding formal theories and algorithms in computational linguistics: e.g. unification, tagging, parsing, Bayesian probabilistic models, machine learning and classification, theorem proving for inference	☐	☐	☐	☐	☐	☐	☐

Comment

8. How challenging did you find the following tasks?

	not challen-ging at all	not challen-ging	somew-hat not-cha-llengin-g	neither challen-ging or not-cha-llengin-g	somew-hat challen-ging	challen-ging	very challen-ging
Writing longer textual answers in English	☐	☐	☐	☐	☐	☐	☐
Solving practical exercises on paper	☐	☐	☐	☐	☐	☐	☐
Performing experimental work including data collection and evaluation	☐	☐	☐	☐	☐	☐	☐
Implementing code in a programming language within a structured lab (where some code is already given)	☐	☐	☐	☐	☐	☐	☐
Implementing code independently, including pre-processing of data	☐	☐	☐	☐	☐	☐	☐
Independently finding a new way how to solve a particular language technology problem with a computer programme	☐	☐	☐	☐	☐	☐	☐

Comment

9. How useful do you think were the following types of instruction to learn new things?

	not useful at all	not useful	somew-hat not useful	neither useful or not useful	somew-hat useful	useful	very useful
lectures	☐	☐	☐	☐	☐	☐	☐
hands-on tutorials	☐	☐	☐	☐	☐	☐	☐
exercises on paper	☐	☐	☐	☐	☐	☐	☐
practical labs with a presence of labs assistants	☐	☐	☐	☐	☐	☐	☐
discussion seminars	☐	☐	☐	☐	☐	☐	☐
individual reading and practical assignments	☐	☐	☐	☐	☐	☐	☐

Comment

10. What do you think it should be done with the courses to improve your learning?

	strongly disagree	disagree	somew-hat disagree	neither agree or disagree	somew-hat agree	agree	highly agree
The courses should have more lecture hours	☐	☐	☐	☐	☐	☐	☐
Extra help should be given to those who need it, for example in the form of individual tutoring by teachers	☐	☐	☐	☐	☐	☐	☐
Extra help can be received from fellow students	☐	☐	☐	☐	☐	☐	☐
Students can catch up individually by reading the relevant literature	☐	☐	☐	☐	☐	☐	☐

Comment

11. If you were responsible for the programme, how would you organise teaching?

12. How true are the following statements?

	strongly disagree	disagree	somew-hat disagree	neither agree or disagree	somew-hat agree	agree	highly agree
The preceding courses gave a me a good and sufficient background to find and solve an interesting language technology problem for my thesis.	☐	☐	☐	☐	☐	☐	☐
I am able to do data processing and programming for my thesis on my own without a difficulty.	☐	☐	☐	☐	☐	☐	☐
I am able to analyse and critically evaluate my results.	☐	☐	☐	☐	☐	☐	☐
I am able to present and discuss my work in a scientific manner.	☐	☐	☐	☐	☐	☐	☐
I can follow a seminar on language technology and I can have a discussion with a colleague in the field.	☐	☐	☐	☐	☐	☐	☐
I can follow a seminar in formal linguistics or have a discussion with a linguist.	☐	☐	☐	☐	☐	☐	☐
I can follow a seminar in computer science on machine learning (not necessarily on natural language processing) and have a discussion with a computer scientist.	☐	☐	☐	☐	☐	☐	☐

Comment

13. Looking back, what knowledge and skills you wish you had more before starting your thesis?

	strongly disagree	disagree	somew-hat disagree	neither agree or disagree	somew-hat agree	agree	highly agree
Better understanding of facts about language	☐	☐	☐	☐	☐	☐	☐
Better understanding about formal representations used	☐	☐	☐	☐	☐	☐	☐
Better understanding of programming and computer science	☐	☐	☐	☐	☐	☐	☐
Better understanding of language technology methods and approaches	☐	☐	☐	☐	☐	☐	☐

Comment

14. What is your goal after finishing the MLT programme?

☐ Apply for a PhD in language technology

☐ Work in a commercial company on development of language technology products

☐ Work in academia or industry in a related technical field

☐ Work in a different field

15. Please include here any comments about the topics of the questions or survey in general.