Natural Language Processing Advice for Linguistics Tripos Students

There are several benefits of completing the Linguistics Tripos before going into NLP, particularly when linguistic insight is so severely lacking in cutting-edge NLP technology. Perhaps the most useful and transferable skill from the Tripos is comparing different theories and formalisms which gives you a useful perspective on the limitations of the current Machine Learning paradigm that forms the foundation of contemporary NLP systems. In the Tripos, beyond Li18, you can do your dissertation either with someone in the Language Technology Lab (Prof Nigel Collier or Prof Anna Korhonen) or Dr Andrew Caines/Prof Paula Buttery in the Natural Language & Information Processing (NLIP) Group, and potentially some other academics as well. However, there is a steep practical and mathematical learning Li18 which is not currently formally accommodated in the Linguistics Tripos, so you do have to be a little outgoing/resourceful. Here are my suggestions:

- **Practical Skills & Resources:** The Python component of Li18 is not nearly enough for Part III/ACS (as lecturers often mention when introducing this part of the course) – the quickest way of improving is by working on an NLP research project and taking a look at open-source codebases released on the ACL Anthology (https://aclanthology.org/) or https://paperswithcode.com/ on topics of interest.
    - First, read some introductory Machine Learning tutorials/textbooks – as a starting point, you could work through Kozzy Voudouris' ML4Linguists course: https://github.com/kozzy97/ML4Linguists. Then, read through the Computer Science lecture notes for MLRD taught by the NLP Group: https://www.cl.cam.ac.uk/teaching/2324/MLRD/materials.html.
        - You should able to answer the IA examination questions on this, as this is all quite fundamental material.
    - Next, use the lecture notes relevant courses from Computer Science are Data Science (Part IB, https://www.cl.cam.ac.uk/teaching/2425/DataSci/) and ML & Bayesian Inference (Part II, https://www.cl.cam.ac.uk/teaching/2324/MLBayInfer/materials.html, though this is *very* theoretical). Most of Paula Buttery's course on Formal Model of Languages should be mostly familiar from Li18, but the exposition is a bit more mathematical: https://www.cl.cam.ac.uk/teaching/2324/ForModLang/.
    - Useful Machine Learning concepts to familiarise yourselves with:
        - **Basic Concepts** – use ML@CL Deep Neural Networks resources (especially the Colab notebooks) and the above Tripos courses to learn about the following:
            - Deep Neural Networks – Backpropagation, Stochastic Gradient Descent, Regularisation (L1, L2), Optimisers (e.g., the Adam optimiser, AdaGrad).
            - Know the main properties of architectures, from the perspective of a ML practitioner: for example, what are the practical limitations of Recurrent Neural Networks (RNNs)? Also, be aware of the main properties of Support Vector Machines, and unsupervised learning algorithms (k-nearest neighbours) and unsupervised clustering algorithms (the k-nearest neighbours algorithm).
        - **LLMs:** Transformers, Vision-Language Models, Reinforcement Learning with Human Feedback (RLHF), etc – read white papers, go through the latest ACL proceedings, some tutorials, and more importantly go through code.
        - Choose ACL papers of interest on some of the following: Active Learning and Tasks like Fact Verification, Natural Language Generation, Natural Language

Inference, Word Sense Disambiguation, and Argument Structure, alongside Li18 “applications” (Machine Translation, Chatbots, etc). – more importantly than reading widely, is understanding algorithms (and having an assessment of their utility) and implementing code and trying out some of your own extensions.
        - Machine Learning tutorials and resources (assumed background for CST *Machine Learning for NLP*, L101): Textbooks are largely dated, so I’d suggest relying on lecture notes and Colab notebook tutorials, particularly Stanford’s NLP courses: https://web.stanford.edu/class/cs224n/index.html#coursework. Yoav Goldberg’s 2015 Primer on Neural Network Language Modelling is very comprehensive, as is Neural Network Methods in NLP (Goldberg & Hirst 2017).
    - Note, that there are standard mathematical prerequisites for any serious Machine Learning work. First year Computer Scientists complete NST Maths and an additional course on probability theory. Beyond A Level Further Maths, you should try to familiarise yourself with the following: multivariate calculus, basic linear algebra (take a look at the course schedules on the Maths Faculty for NST Maths and IA Probability).
- **How to apply your linguistic background for cutting-edge NLP research:**
    - **Li2:** Even the most linguistically-keen NLP researchers are unfamiliar with content from Li9 or Li10, but your Li2 knowledge will be useful in very specific ways. However, to do this, you have to understand that Computer Scientists operate very differently to linguists, so here are some starting points to help you get readjusted:
        - **Parsing Formalisms:** As a Computer Scientist, we shift formalisms a lot (e.g, Universal Dependencies, Combinatory Categorial Grammars, older Attribute-Value and feature-based grammars). Once you have familiarised yourself with these and more recent neural dependency and constituency parsers which combine ideas from Machine Learning with classical ideas about parsing, you might have ideas for how these systems might be improved.
        - **Computational Semantics:** Logic for Linguists is a highly useful prerequisite for Computational Semantics. You might look into Semantic Parsing or Computational Pragmatics models (e.g., Rational Speech Acts) – take a look at some of Dr Guy Emerson’s work, amongst others.
        - To get a flavour of this, you can take a look at Chen & Manning (2014)’s Biaffine Syntactic Parser (https://aclanthology.org/D14-1082.pdf) and/or a state-of-the-art semantic dependency parser (Dozat & Manning 2017).
    - **Li17:** Your knowledge of typology will be invaluable for NLP – take a look at Machine Learning approaches for computational typology and low-resource languages, particularly meta-learning or Transfer learning.
    - Your other linguistic interests will, of course, guide your interests in various ways.
- **Research Projects:** If you are interested in applying for the MPhil Advanced Computer Science, working with someone in the group before you apply is always going to be favourable to your application.
    - If Paula is supervising you for Li18, definitely approach her and/or email Andrew Caines to see if they have any research opportunities over the summer (even if this informally working on a project). Equally, you could contact other members of the NLP group – Dr Weiwei Sun works on computational syntax and meaning representations (and is a Generativist), Dr Fermin Moscoso del Prado Martin has functionalist leanings, but both do some form of computational cognitive modelling. Even if they are unable to take you on, it might be a good idea to make yourself known to them! The other Faculty

members, Prof Andreas Vlachos and Prof Simone Teufel, are technically on research sabbatical this year.
    - Nigel also maintains a list of student projects: https://ltl.mmll.cam.ac.uk/student_projects/. I approached him after an Li18 lecture and we ended up working on one of these in the summer before my third year – I had an ACL paper draft by the end of the project, which was very useful for my master's applications.To give you a sense of the timeline, we started on this in Lent with initial meetings, took a break for exam term, and then worked on this nearly full-time for two months in the summer.
    - **An informal point about funding/logistics:** there is lots of variation about funding – in my personal experience, my UROP in the NLP group was well paid and I learnt a lot, but my work with Nigel was more *ad hoc*, although I got college funding to subsidise costs like accommodation, etc. Even if you do not secure funding, please do try to ask for college support, and you can still work remotely on computational projects from home over the summer, which minimises costs.
- **Seminars & Reading Groups:** I'd highly suggest attending the LTL (Thursday) and NLIP seminars (Friday) regularly. There is also an NLP Reading Group in the Computer Lab on Tuesdays fortnightly at 1:15pm – most people attend these in-person, but you can also join online. I'm happy to invite you to these if you can make it.

The Part III/ACS at Cambridge have different admissions processes, but you can apply for both. The Computer Science DoS at your college would assess your Part III application internally and they will likely not be working in NLP, whereas you will be interviewed by someone in the NLP group for an ACS application. You might have a preference about which process would work best for you.

You can also work in the LTL through the Linguistics MPhil, and they have a PhD in Computation, Cognition & Language that you can straightforwardly progress onto. Beyond Cambridge, Edinburgh has a great NLP Faculty, and their courses are a bit more accommodating to interdisciplinary backgrounds compared to switching Triposes.

Suchir Salhan
November 2024