# Empowering the Future with Multilinguality and Language Diversity

En-Shiun Annie Lee University of Toronto, Ontario Tech University Ontario, Canada annie.lee@ontariotechu.ca

Syed Mekael Wasti Ontario Tech University Ontario, Canada syedmekael.wasti@ontariotechu.net

### Abstract

The rapid advancements and the widespread transformation of Large Language Models, have made it necessary to incorporate these cutting-edge techniques into the educational curricula of Natural Language Processing (NLP) with limited computing resources. This paper presents an applied NLP course designed for upper-year computer science undergraduate students on state-of-the-art techniques with an emphasis on multilinguality and language diversity. We hope to empower learners in advancing their language community.

## **1** Introduction to Pedagogical Approach

We present a newly designed Natural Language Processing (NLP) course for upper-year computer science students at a primarily undergraduate teaching institution of a diverse multicultural audience. The rapid advancement in the field poses a significant challenge for educators to adequately cover both traditional linguistic techniques in addition to the latest neural techniques and large language model (LLM) developments (Goldberg, 2016; Santra et al., 2023). Therefore, to address these challenges, our course emphasizes selfdirected learning by incorporating hands-on labs, assignments<sup>1</sup>, and two exams, all designed to promote in-depth and robust life-long learning.

**Target Audience:** Demographically, the local region is known for multiculturalism where the institution is composed of a high proportion of first-generation immigrants who speak their native tongue. Turning challenges into opportunities, this course covers multilinguality and language diversity, which is ideal for empowering the local student population. Furtheremore, this course is designed for senior undergraduate students in computer science, with the prerequisites of linear algebra, calculus, probabilities, and most importantly

Kosei Uemura

University of Toronto Ontario, Canada k.uemura@mail.utoronto.ca

Mason Shipton Ontario Tech University Ontario, Canada mason.shipton@ontariotechu.net

machine learning (introductory). Some junior graduate students may also join the course if they have an interest in NLP research.

**Learning Outcomes** The learning outcomes of the course are: 1) Understanding and knowing the NLP concepts, terminology, tasks, methods, and techniques; 2) Modifying and debugging NLP code with comfort and proficiency; 3) Applying advanced NLP techniques in building and extending LLMs; 4) Connecting personal cultural and personal experiences to the latest work in multilinguality and language diversity.

## 2 Course Structure and Content

The course is 12 weeks long with 9 weeks of lectures and 3 weeks for invited speakers who are working in multilinguality and language diversity (Table 1). There are weekly in-person labs with optional TA assistance, where students run Python notebooks and answer short quiz questions related to the code. Then there are 3 assignments, each designed to complement the materials covered in the lecture. The lectures are used to cover the problems and methods of the assignments at a high level and then allow learners to have hands-on implementation of the material. The course has a midterm exam and a final exam to ensure learners grasp the foundations of the materials.

Assignment 1: A Journey through Language Modelling. This assignment introduces students to the foundations of language modelling applied to low-resource languages. The goal is to introduce a series of language models that are increasing in complexity (Gaddy et al., 2021). Students will gain experience loading datasets of low-resource languages, which they will process, tokenize, and use to construct custom vocabularies (Schmidt et al., 2024). First, students begin by implementing a basic statistical n-gram model (Brown et al., 1992), followed by a feed-forward neural n-gram model,

<sup>&</sup>lt;sup>1</sup>Public Access of the Course Assignments can be found at https://github.com/Kosei1227/OTU-LLM-Course

Week	Lecture Topics	Lab Notebooks	Assignments
1	Course Introduction	Python and Regex	
2	Corpus Statistics and n-Gram Language Model	N-Gram Language Modelling	
3	Entropy Decisions	PyTorch Introduction	A1
4	Machine Learning and Feature Classification	Naive Bayes and Text Classification	
5	Neural Language Models	Word Embeddings and Vector Semantics	
6	MIDTERM EXAM	RNN	MIDTERM
SB	STUDY BREAK		
8	Attention and Transformers	Pytorch and Attention	A2
9	Large Language Models	Transformer (Illustrated and Annotated)	
10	Multilinguality and Language Diversity	HuggingFace1	
11	Multilinguality and Language Diversity	HuggingFace2	A3
12	Multilinguality and Language Diversity	Transfer Learning	
	FINAL EXAM	-	FINAL

Table 1: Contents of the weekly lectures and corresponding lab notebooks.

and finally, the transformer language model (Vig and Belinkov, 2019). After the series of implementations, students conduct an open-ended exploration, attempting to improve results beyond the given exercise, either from a list of provided ideas or based on their intuition. <sup>2</sup>

**Assignment 2: Neural Machine Translation with** Custom Vocabulary Building & Transformer. This assignment covers the foundational principles of neural machine translation (NMT), through the integration of a custom transformer architecture. After the hands-on processing of low-resource languages, students implement a custom transformer (Vaswani et al., 2023) class through the use of Py-Torch modules and layers (Radford et al., 2023), exposing students to the architecture's inner workings. The model's construction will conclude by integrating forward and masking methods with Py-Torch classes. Following the creation of their models, students craft a custom training loop, where they gain hands-on experience working with gradient descent optimization (Robbins, 1951), backpropagation (Rumelhart et al., 1986), and loss functions (LeCun et al., 2015). Finally, the assignment will conclude by allowing students to evaluate the translation performance of their hand-crafted models, through the use of industry-standard metrics, such as the BLEU score (Papineni et al., 2002). Upon completion of this assignment, students will have developed a holistic understanding of core NLP principles along with a strengthened machine learning foundation.

Assignment 3: Adapting Languages with Fine-Tuning. This assignment guides students through the process of adapting existing language models to a low-resource language, providing hands-on experience with modern neural machine translation techniques and transfer learning strategies. Students will begin by selecting a low-resource language, leveraging datasets from prior NMT research, and exploring various fine-tuning methods, such as full parameter fine-tuning, LoRA (Hu et al., 2021), and prompt tuning (Lester et al., 2021). The rationale for the chosen strategy must be discussed, emphasizing efficiency and effectiveness given the constraints of limited computational resources. Using appropriate software and repositories, students will fine-tune the language models and develop custom benchmarks to evaluate performance. The assignment will culminate in a comprehensive evaluation of the adapted models against baseline models, enabling students to critically analyze and understand the impact of their modifications. This assignment builds upon students' knowledge of neural architectures, such as Transformers (Vaswani et al., 2023), and equips them with the skills to undertake research in NLP by focusing on real-world applications and interactive learning methods.

### 3 Conclusion

A recent challenge in NLP pedagogy is the rapid advancement of LLMs and the consequent surge in computational requirements(Kaplan et al., 2020). In light of those challenges, our course is designed to ensure learners remain abreast of the latest techniques emphasizing multilinguality and language diversity to empower students and their communities of the institutional and local demographic. We hope to cultivate essential lifelong learning skills that empower them to adapt to the ever-evolving landscape.

<sup>&</sup>lt;sup>2</sup>This assignment is from UC Berkeley's Computer Science graduate NLP course (cs288) Interactive Assignments for Teaching Structured Neural NLP, Project 1: Language Modeling https://sites.google.com/view/nlp-assignments

#### References

- Peter F Brown, Vincent J Della Pietra, Peter V Desouza, Jennifer C Lai, and Robert L Mercer. 1992. Classbased n-gram models of natural language. *Computational linguistics*, 18(4):467–480.
- David Gaddy, Daniel Fried, Nikita Kitaev, Mitchell Stern, Rodolfo Corona, John DeNero, and Dan Klein. 2021. Interactive assignments for teaching structured neural nlp. In *Proceedings of the Fifth Workshop on Teaching NLP*, pages 104–107.
- Yoav Goldberg. 2016. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.
- Yann LeCun, Y. Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521:436–44.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *Preprint*, arXiv:2104.08691.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR.
- Herbert E. Robbins. 1951. A stochastic approximation method. Annals of Mathematical Statistics, 22:400– 407.
- David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by backpropagating errors. *Nature*, 323:533–536.
- Payel Santra, Madhusudan Ghosh, Shrimon Mukherjee, Debasis Ganguly, Partha Basuchowdhuri, and Sudip Kumar Naskar. 2023. Unleashing the power of large language models: A hands-on tutorial. In *Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation*, pages 149–152.
- Craig W. Schmidt, Varshini Reddy, Haoran Zhang, Alec Alameddine, Omri Uzan, Yuval Pinter, and Chris Tanner. 2024. Tokenization is more than compression. *Preprint*, arXiv:2402.18376.

- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.
- Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, page 63. Association for Computational Linguistics.

### **A** Appendix

**Technical Stack** All technical and teaching materials are integrated into a cloud environment accessible by internal students. The libraries and packages used for the environment are: 1) Debian OS - TensorFlow Jupyter Notebook with CUDA driver installed so that it can utilize the server GPU hardware; 2) Python libraries installed are: NLTK, Gensim, SciPy, PyTorch, Portalocker, tqdm, and scikit-learn. <sup>3</sup> 3) Regarding the NLTK and Gensim libraries, it has datasets/models already baked into the image. The students/users won't need to download them, it saves up storage space.

The current jupyter environment load all the lab contents the moment a user logins into the hubdey. Student can login now and test it. There are some behaviours applied to for the sake of the student and ease-ability of updating the lab content: The lab content files in the student's home directory will always get updated to match that from the Docker image. If in the middle of the semester you want to change the lab content, it will be easier to just update the docker image and all students will receive the updated lab in their home directory (as long as they log out / restart their notebook). The lab content (.ipynb, .csv, and .txt files) are all "read-only" files. Students have to save-as a new .ipynb file for saving their lab progress. Jupyter will alert them to save-as. This is to avoid cases where a student accidentally deleted or modified the original lab content. So that there will not be the case where a student is messaging the prof in the middle of the night just because they made a mistake and asks for a copy of the original file. If a student deleted the lab content file (they really have to work out of their way to do this), they will just need to restart their notebook and the lab content will be there again.

<sup>&</sup>lt;sup>3</sup>More libraries or packages can be added if the assignments/labs are updated.

## Acknowledgments

We would like to thank the following people for testing and setting up the labs and assignments: Kevin Chandra, Mason Shipton, Ethan Randle-Bragg, and Mehdi Benallegue. Special thanks to Professor Ken Pu from OntarioTech University for helpful feedback on the teaching compute, and Professor Laura Burdick from the University of Michigan for sharing her teaching materials. Most importantly, we thank the teaching NLP community and slack group for providing resources, and a platform for sharing pedagogical thoughts and ideas that made this process an enriching journey.