Industry vs Academia: Running a Course on Transformers in Two Setups

Irina Nikishina[1*], **Maria Tikhonova**[2,3*], **Viktoriia Chekalina**[4,5], **Alexey Zaytsev**[4], **Artem Vazhentsev**[4,5], **and Alexander Panchenko**[4,5]
[1]Universität Hamburg, [2]HSE University, [3]SaluteDevices, [4]Skoltech, [5]AIRI
irina.nikishina@uni-hamburg.de, a.panchenko@skol.tech

Abstract

This paper presents a course on neural networks based on the Transformer architecture targeted at diverse groups of people from academia and industry with experience in Python, Machine Learning, and Deep Learning but little or no experience with Transformers. The course covers a comprehensive overview of the Transformers NLP applications and their use for other data types. The course features 15 sessions, each consisting of a lecture and a practical part, and two homework assignments organized as CodaLab competitions. The first six sessions of the course are devoted to the Transformer and the variations of this architecture (e.g., encoders, decoders, encoder-decoders) as well as different techniques of model tuning. Subsequent sessions are devoted to multilingualism, multimodality (e.g., texts and images), efficiency, event sequences, and tabular data.

We ran the course for different audiences: academic students and people from industry. The first run was held in 2022. During the subsequent iterations until 2024, it was constantly updated and extended with recently emerged findings on GPT-4, LLMs, RLHF, etc. Overall, it has been ran six times (6 times in industry and 3 times in academia) and received positive feedback from academic and industry students.

1 Introduction

The Transformer (Vaswani et al., 2017) is a versatile neural network model that can be successfully used in various modalities, such as text, images, networks, or sequences of events. Transformer-based models have reached a pinnacle of popularity: they have established state-of-the-art performance in various text processing applications and come with user-friendly wrappers on numerous data science platforms. Therefore, many industrial applications rely on Transformer models.

Figure 1: A course instructor (Maria Tikhonova) gives a lecture for students.

Although current computer science students may study Transformers in their university courses, many machine learning engineers often lack a thorough understanding of the underlying mechanisms of these models. This gap in knowledge can hinder their ability to fully leverage the potential of Transformers, resulting in decreased efficiency and quality in their work.

In this paper, we present an overview of a course on transformer-based models (see Figure 1), which bridges the gap between academic training and industry needs thanks to the balanced program incorporating both theoretical knowledge and a large spectrum of practical use-cases which can be directly used for industrial needs. Therefore, it can be successfully taught in academic and corporate environments. The course seeks to concisely condense and present a vast amount of information on this topic, specifically targeting individuals with ML expertise but limited knowledge of NLP. It aims to provide a comprehensive understanding

[*] Equal contribution.

Proceedings of the Sixth Workshop on Teaching NLP, pages 7–22

of the Transformer architectures including the recently emerged topics connected with Large Language Models' (LLMs) theory and their application, enabling students to tackle the challenges that arise when working with them effectively. The course not only gives the theoretical knowledge of this model set but also provides studies with various practical scenarios and use cases they can encounter in industrial applications.

The course was developed and served first in July 2022 and substantially updated in the subsequent runs. Currently, the course has been held six times: 6 times in a corporate environment (data scientists and trained engineers from a large IT company) and 3 times in an academic institution.

The contributions of this paper are as follows:

- We present the syllabus of a modern course on transformer-based models, aimed at broad heterogeneous audiences both in academia and in the industry, which combines deep theoretical knowledge with modern practical applications of the Transformer models;
- We release the materials from the academic course run, which are available in our repo[1];
- We combine recent NLP trends and latest approaches with other best practices, such as fine-tuning of the pre-trained Transformers, multimodality, prompt-tuning, model compression, etc.;
- The course program includes a comprehensive set of Transformer applications, not only the NLP domain but also other modalities.

2 Related NLP Courses

Over the past two decades, dozens of classes on NLP emerged. With the "deep learning revolution" starting around 2017, almost every program in computer science (academic or industrial) features a class on NLP. Modern courses, such as CS224N[2], are focused on the use of deep learning models as the most efficient methodology allowing to obtain state-of-the-art results in a range of tasks. Most currently best-performing models for NLP are based on the Transformer architecture. Besides, Transformer architecture is widely adopted in other domains such as computer vision, tabular data processing, event, and sequence processing. Our course is therefore centered around the Transformer architecture, but in contrast to CS25[3], our course has more focus on NLP, Computer Vision, and applications to tabular and event data while not covering robotics and neuroscience.

The published works on teaching NLP consider different scales and scenarios. Some papers consider the design of extensive programs related to computational linguistics and NLP (Reiter et al., 2017). Other papers describe specific parts of courses, including competitions (Barteld and Flick, 2017; Bozhanov and Derzhanski, 2013). Generally, courses on NLP are either industry- or academia-oriented and target different audiences (Vajjala, 2021), can be held online (Artemova et al., 2021), offline, or in hybrid mode.

We consider a different course objective. Namely, we aim to provide one of the first courses focused on the specifics of Transformer architecture, how it can be applied to solve various problems in NLP, and how it can be used in other domains (e.g., for images, tabular data). The challenge here is how to embed various innovations related to this architecture into a single course. The course can be taken in two scenarios: as part of a computer science master's program or as an additional education course for an industrial audience.

3 Course Overview

The course comprises 15 sessions (two sessions per week) and assignments, which include two homeworks, a final quiz, and bonus lecture quizzes and practical tasks after each session. The course program, presented in Table 1, can be split into two main parts, which cover (i) basic Transformer architectures and models, (ii) the application of Transformers to different modalities, and efficient training procedures.

The course is based on the following prerequisites: (1) **advanced mathematics**: calculus, linear algebra, and statistics; (2) **data science**: classic machine learning methods, basic deep learning methods, and basic knowledge of natural language processing.

Every session consists of a lecture and a practical seminar. Lectures are presented with slides, while practical sessions are real-time coding sessions. The instructor demonstrates code snippets in Jupyter Notebooks and explains them in detail. Both home assignments are started simultaneously

[1] https://github.com/s-nlp/transformers-course
[2] https://web.stanford.edu/class/cs224n
[3] https://web.stanford.edu/class/cs25

Session	Description
1	The Transformer: motivation, original architecture, and attention mechanism.
2	Transformer-based Encoders. Masked language models based on the Transformer architecture. BERT and related models.
3	Classification and sequence tagging with Transformers. Using encoders to generate feature representation for various NLU tasks.
4	Transformer-based Decoders. Generation of text using Transformers. GPT and related decoders.
5	Prompt and instruction tuning. Reinforcement Learning from Human Feedback (RLHF), ChatGPT, and related models.
6	Sequence to sequence tasks: machine translation, text detoxification, question answering, dialogue. Technical tricks for training and inference.
7	Multilingual language models based on the Transformer architecture.
8	Uncertainty estimation for Transformers and NLP.
9	Efficient Transformers.
10	Compression of Transformer models and low-rank approaches.
11	Network encoders with Transformers
12	Multimodal and Vision Transformers.
13	Transformers for event sequences.
14	Transformers for tabular data.
15	Deadline for both assignments. Final quiz.

Table 1: Course structure: each session, except the last one, dedicated to the final quiz, features lecture material and a seminar with code snippets.

from the beginning so that students may plan their time accordingly and try any transformer-based architecture they find applicable to any assignment.

The total score Total is calculated according to the following formula:

$$\mathrm{Total} = 0.4 \cdot A_1 + 0.4 \cdot A_2 + 0.2 \cdot Q + LQ + ST,$$

where A_i is the score for the i-th home assignment, Q is the score for the final quiz, LQ and ST are the extra points for bonus lecture quizzes and practical tasks, respectively. The total score is then uniformly mapped into the grading scale.

As was already mentioned, the course is suitable for both academic and industrial audiences. While the general course program and structure are similar in both environments, the presentation of the material differs, adapting to the audience's needs and objectives. In academic runs, we concentrate more on the theoretical material, giving more mathematical formulas and explanations. In contrast, during the industrial runs, we provide more practical examples, illustrating all methods with as many use cases and business applications as possible.

It is worth noting that the course can be conducted both online and offline. For industrial sessions, the course is delivered online, whereas at the university, we adopt a hybrid approach.

4 Syllabus

The following paragraphs describe each session in more details.

Session 1. The Transformer: motivation, original architecture, and attention mechanism. The first introductory **lecture** is devoted to the vanilla Transformer architecture (Vaswani et al., 2017), introduced for the Machine Translation (MT) task following the traditional approach.

First, we formulate the MT task and present a historical overview of the area. Next, we describe the idea of encoder-decoder or seq2seq architecture, starting with RNN-based models (Sutskever et al., 2014) and then introduce the concept of attention (Bahdanau et al., 2015). That brings us to the Transformer model, which we explain step by step. During the academic runs, we pay special attention to the theoretical background behind the Transformer architecture and its mathematical explanations.

We conclude the lecture with a short recap of the language modeling task and how the idea of attention can be transferred from MT to this field.

The **practical session** is based on Harvard NLP tutorial "Annotated Transformer"[4], that presents the PyTorch[5] implementation of the Transformer architecture. Thus, following the outline of the tutorial, we first go through the Transformer code step by step and then show how it works for WMT14 (Bojar et al., 2014) English-to-French translation task: we train and test the model and allow students to experiment with model training to achieve better scores.

Session 2. Transformer-based Encoders. Masked language models based on the Transformer architecture. BERT and related models. The **lecture** is devoted to the transformer-based Encoders. We begin with discussing most classical encoder-based models, such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), and discuss the peculiarities of their architecture and training. For academic students, we pay particular attention to analyzing the results of the original scientific papers, while for the industrial runs, we concentrate

[4] http://nlp.seas.harvard.edu/annotated-transformer
[5] https://pytorch.org

more on the use cases students can encounter in their practice.

The aim of the **practical session** is to teach students how to work with Transformer models using Transformers library[6] and how to utilize pre-trained models and other instruments from HuggingFace Hub[7]. Namely, we show students how to tokenize the data, visualize attention maps, and apply trained models on the example of the BERT model fine-tuned for the sentiment analysis task.

Session 3. Classification and sequence tagging with Transformers. Using encoders to generate feature representation for various NLU tasks. This session focuses on Natural Language Understanding (NLU) applications of Transformers, namely, tasks that need to extract implicit metadata from the text. In the **lecture**, we consider text classification for Sentiment Analysis and Natural Language Inference and the token classification for Named Entity Recognition and Extractive Question-Answering tasks. For the industrial runs, we elaborate more on the practical applications derived from these tasks and the use cases. We study various approaches to sentence encoders in detail and then delve into the realm of dialogue systems.

In the **practical session**, we fine-tune the transformer-based model for entity recognition in the Russian-language drug review corpus (Tutubalina et al., 2020) using a Russian-language compressed version of the BERT[8] for it.

Session 4. Transformer-based Decoders. Generation of text using Transformers. GPT and related decoders. This session is devoted to Generative Pre-trained Transformer (GPT) models based on the Transformer decoders and various text generation strategies.

In the first part of the **lecture**, we briefly recap various types of language models, emphasizing decoder-based Transformer models. Then, we carefully study GPT models (GPT-1,2,3, and 3.5), focusing on GPT-3 (Winata et al., 2021) and introducing the concept of few-shot learning.

Additionally, we explore the strategies for token sampling and text generation (e.g., BeamSearch, Sampling, Nucleus Sampling). For the academic students, we concentrate more on the theoretical aspects, while with the industrial students, we discuss what generation strategies in business applications from their work experience are preferred.

In the second part, we consider examples of controllable text generation where we aim to generate text with specific desired properties at the model level. Then, starting with an additional steering layer (Dathathri et al., 2019), we come to the concept of a Generative Adversarial Network in the model GeDi (Krause et al., 2021).

In the **practical session**, we provide guidance on introductory text generation and sampling strategies and experiment with various methods. Experiments are set on the encoder-based Transformer model sourced from the HuggingFace library (e.g., GPT-2[9]) and aim to analyze the impact of generation hyperparameters on text quality and styling. Upon request, we can develop a straightforward chatbot utilizing the model and explore its practical relevance in industry and startup contexts.

Session 5. Prompt and instruction tuning. Reinforcement Learning from Human Feedback (RLHF), ChatGPT, and related models. This section continues the exploration of advanced text generation models, specifically focusing on the concept of Reinforcement Learning from Human Feedback (RLHF). RLFH involves incorporating human feedback into the learning process to establish an optimal starting point for the further model's training for the given task. As an example, we consider the task of summarization with human feedback (Stiennon et al., 2020) and analyze the concept (Ouyang et al., 2022) of the modern LLM training which underlies the power of such models as ChatGPT[10] or GPT-4[11] models. For the academic runs, we pay special attention to the theory behind the RLHF method and its potential development. In the industrial runs, we discuss the practical difficulties (e.g., data collection, computational resources, cost) of RLHF LLM training.

The next part is devoted to prompt-tuning methods (Li and Liang, 2021; Lester et al., 2021; Liu et al., 2021; Konodyuk and Tikhonova, 2021), for automatically learning language model prompts.

We conclude the lecture by discussing the emerging variety of LLMs (LLaMA-2 (Touvron et al.,

[6] https://pypi.org/project/transformers
[7] https://huggingface.co
[8] https://huggingface.co/cointegrated/rubert-tiny
[9] https://huggingface.co/gpt2
[10] https://openai.com/blog/chatgpt
[11] https://openai.com/gpt-4

2023)[12], Mistral[13], Mixtral[14], etc., their industrial application scenarios, possible downsides connected with their usage, and the overall impact. This list is updated each run with newly released models.

The **practical session** is devoted to prompt-tuning methods. We allow students to experiment with ruPrompts[15], a convenient library for fast language model tuning via automatic prompt search, and use it to solve the Russe Detoxification task (Dementieva et al., 2022).

Session 6. Sequence to sequence tasks: machine translation, text detoxification, question answering, dialogue. Technical tricks for training and inference: infrastructure and performance. During the **lecture**, students' attention is drawn to the models with the standard Transformer Encoder-Decoder architectures, which are aimed at solving sequence-to-sequence tasks such as machine translation, summarization, question answering, etc. In the first part of the lecture, we discuss the existing sequence-to-sequence models such as BART (Lewis et al., 2019), T5 (Raffel et al., 2020a) and PEGASUS (Zhang et al., 2019). With the industrial students, we discuss possible applications of these models in their business practice.

The lecture's second part is devoted to optimizing the Transformers' training process. We discuss such optimization techniques as gradient accumulation, training of only some layers, Adafactor optimizer (Shazeer and Stern, 2018), quantization (Hawks et al., 2021) and mixed precision (Micikevicius et al., 2018), gradient checkpointing (Chen et al., 2016), optimized padding, and ONNX runtime (developers, 2021).

The **practical session** for the sequence-to-sequence Transformers aims to solve the Hypernym Prediction task using the T5 (Raffel et al., 2020b) model. Students are expected to experiment with the zero-shot and few-shot setups and compare them with fine-tuning.

Session 7. Multilingual language models based on the Transformer architecture. This session is devoted to multilingual language modeling. We begin the **lecture** by discussing the specifics of this phenomenon, a short overview of the MT approaches, and methods for parallel corpora creation. Then, we switch to the multilingual transformer models and discuss such models as mBERT[16], XGLM (Newson, 2016), BLOOM (Scao et al., 2022), and mGPT (Shliazhko et al., 2022). In addition, we discuss possible ways to add a new language to the Transformer model (e.g., mBERT).

In the **practical session** we fine-tune XLM-R (Zhuang et al., 2021) for multilingual and cross-lingual word-in-context disambiguation (MCL-WiC), proposed for the SemEval2021 competition (task2) (Martelli et al., 2021).

Session 8. Uncertainty estimation for Transformers and NLP. This session aims to provide a general introduction to the uncertainty estimation (UE) field and methods and their application to NLP, especially transformer-based models.

The **lecture** begins with highlighting the importance of uncertainty estimation and introducing standard and well-established methods, such as Softmax Response (Geifman and El-Yaniv, 2017) and Monte-Carlo (MC) Dropout (Gal, 2016). We also cover various regularization techniques (Xin et al., 2021), density-based methods (Lee et al., 2018), and also state-of-the-art UE methods for the classification task (Yoo et al., 2022). Next, we show the importance of uncertainty estimation for LLMs, e.g., to avoid hallucinations. We discuss the most advanced techniques, including both white-box methods (Kuhn et al., 2023), applicable for any open-sourced model, and black-box methods (Lin et al., 2023), which are useful for closed-sourced models available via API. We conclude the lecture by discussing the practical application of uncertainty estimation for active learning (Settles, 2009), out-of-distribution (OOD) detection, etc.

In the **practical session**, we implement several UE methods, such as Mahalanobis distance (Lee et al., 2018), MC Dropout, and HUQ (Vazhentsev et al., 2023), and apply them for the selective classification and OOD detection tasks. In addition, we study two baseline methods (Sequence Probability and Lexical Similarity) (Fomicheva et al., 2020) for UE of LLMs and use them to detect factual errors in the summarization task.

Session 9. Efficient Transformers. LLMs impose high memory requirements, and this, consequently, leads to substantial energy costs and noteworthy CO2 emissions (Rae et al., 2021) through-

[12] https://ai.meta.com/research/publications/llama-2-open-foundation-and-fine-tuned-chat-models

[13] https://mistral.ai/news/announcing-mistral-7b

[14] https://ollama.com/library/mixtral

[15] https://github.com/ai-forever/ru-prompts

[16] https://huggingface.co/bert-base-multilingual-cased

out the training and inference process.

In the **lecture**, we explore various approaches to reduce the model size without compromising quality. We delve into pruning (Sanh et al., 2020), (Lagunas et al., 2021), quantization (Hawks et al., 2021), (Wang et al., 2022b), and distillation (Hinton et al., 2015). In the academic runs, we spend more time on the theory behind these methods, while in the industrial runs, we concentrate more on the practical applications of these methods.

We also examine methods that aim to reduce the computational complexity of the attention layer (Tay et al., 2020). It includes approaches that simplify the calculation procedure, such as the kernel method, techniques that decrease the input sequence size, and techniques for selecting a subset of tokens for attention computation (learnable or fixed patterns).

Finally, we overview two approaches to parallelism during training: model- and data-level parallelism. Using the examples of the Megarton (Shoeybi et al., 2019) and Varuna (Athlur et al., 2021) pipelines, we explore options for distributed computing across multiple GPU cards and nodes and the concomitant challenges. We touch upon the topic of the impact of model training processes on the environment and methods for its evaluation, which is relevant to the industry.

During the **practical session**, students will construct their own layer, based on *torch.nn.Linear* incorporating a quantization mechanism. The objective is to minimize the total memory footprint of the model by representing several layers in a compressed bit format.

Session 10. Compression of Transformer models and low-rank approaches. In this session, we explore methods for decreasing the number of parameters by representing layer weights in a more compressed way. Focusing on the representation by SVD, Kronecker decomposition, and Tensor Train Matrix (TTM) (Oseledets, 2011) decomposition; we study the peculiarities of the structure of such layers and the propagation of signals within them.

For the **practical session**, students are offered a layer implementation based on SVD and TTM decomposition. The objective is to assess the performance of a compressed model achieved by substituting fully connected layers with algebraic structures based on the compression rank.

Session 11. Network encoders with Transformers This **lecture** starts with a short recap of graph theory. Then we introduce Graph Convolutional Networks (GCNs) (Kipf and Welling, 2016) and analyze those that are related to Transformers: GAT (Yun et al., 2022), Graph BERT (Zhang et al., 2020), and GreaseLM (Zhang et al., 2022). We also discuss how such models could be applied and for which NLP tasks.

In the **practical session**, we discuss the Taxonomy Enrichment task using Graph Transformers (GCN, GAT, and Graph-BERT). We also revise the code of GAT-v2 (Brody et al., 2022) and make a quick overview of the OpenHGNN library[17] with the implementation of Graph-based models.

Session 12. Multimodal and vision Transformers. Multimodal Transformer architectures are significant as they generate representations of language concepts by leveraging textual data and information from diverse sources such as images, videos, and knowledge bases.

We start the **lecture** with the CLIP (Radford et al., 2021) model architecture analysis, which provides a joint embedding for words and pictures representing this word. Then go through all the most important and relevant multimodal models (DALLE, DALLE-2 (Ramesh et al., 2022), VQ-VAE (van den Oord et al., 2017), Rudolph (AIRI, 2022), Fromage (Koh et al., 2023), Flamingo (Alayrac et al., 2022), OFA (Wang et al., 2022a), Kandinsky (Razzhigaev et al., 2023), ImageBind (Girdhar et al., 2023)). We discuss the possible use cases of these models with the industrial students in their work practice.

During the **practical session**, we made a zero-shot classifier with CLIP and implemented a visual saliency map. We also studied how Kandindky[18] works for generating images by text.

Session 13. Transformers for event sequences. Another essential data modality in modern applications is event sequences. We consider a sequence of events with features or marks provided for each event. The model can be used end-to-end or as an encoder to get embeddings of a sequence (Zhuzhel et al., 2021; Babaev et al., 2022). We also note that in this topic, we describe a connection between these models and temporal point random processes (Shchur et al., 2021). During the **lecture**, we study the adaptation of the Transformer architecture to this problem and compare it to other

[17] https://github.com/BUPT-GAMMA/OpenHGNN
[18] https://huggingface.co/ai-forever/Kandinsky3.1

approaches. During the **practical session**, we train from scratch a Transformer model (Zuo et al., 2020) for processing open-sourced financial transactions data (Fursov et al., 2021), a modality that is widely used in major banks (Babaev et al., 2022).

Session 14. Transformers for tabular data. In this session, we step aside from classical Transformer applications and discuss their use for tabular data. It is a new but quite promising area.

The **lecture** is based on three papers devoted to this subject. We start with (Huang et al., 2020), which proposes the TabTrasformer model, applying the attention mechanism for categorical feature embeddings. Then, we walk through (Gorishniy et al., 2021), which extends the idea of the attention mechanism to numerical features by embedding them via linear transformation and subsequently applying the Transformer block. We also study various embedding types for numerical features proposed in (Gorishniy et al., 2022) and how they can be combined with the Transformer block. Finally, we discuss practical applications and how the architectures can be adapted to industrial needs (we pay special attention to this part of the lecture during industrial runs).

In the **practical session**, we study the described transformer-based tabular models implemented in PytorchTabular[19] Python library and apply them for one of the classical tabular datasets (e. g., Bank Marketing Data Set[20]).

Session 15. Deadline for both assignments. Final quiz. During the final session, students are expected to share their feedback on the course and discuss their solutions for the home assignments. We first discuss each task's strengths, weaknesses, and difficulties and the time spent developing the method that outperforms the baseline. Afterward, we split students into groups to discuss the developed methods and to share their experiences. Each group is asked to present one method for each task to share with other groups.

5 Assessment

5.1 Home assignments

We provide two home assignments for the course described in sub-subsections 5.1.1 and 5.1.2.

For both tasks, students are expected to provide a technical report (max 10 points) and code (max 10 points) and submit the results of the best-performing model to the CodaLab competition leaderboard (max 15 points) (see Appendix A for the detailed grading criteria). They should also write a technical report in the provided IPynb template describing the method used in their solution and the analysis of the obtained results. In the code section, students are expected to develop a solution and provide a reproducible code in the provided template. Then, the (best) model output is expected to be submitted to the CodaLab platform[21] with the name of the user for evaluation.

There is no formal difference in assignments or grading criteria for the academic and industrial runs. However, during the academic runs, we stimulate students to concentrate more on the theoretical analysis of their results and the scientific conclusions they can draw from them, while in industrial runs, we ask students to elaborate more on practical effects that can be inferred from the results they obtained in the assignments.

We chose these assignments since they cover the two most widespread tasks in NLP: classification and generation. In the classification task, we ask to perform sequence tagging which is a token classification task using any Encoder Transformer model, while text detoxification assignment is one of the text generation tasks to be solved with Decoder or Encoder-decoder models.

Each assignment involves a deep dive into the task, offering the opportunity to try several different approaches to solving it, carefully considering and discussing their advantages, disadvantages, and possible modifications. Such an assignment closely models the process of solving real-world problems and takes at least a month to complete. Therefore, the number of assignments is selected given the course's total length and ability to cover the basic use cases for Transformers.

5.1.1 Assignment 1: Semantic Role Labelling

The first assignment is to perform semantic role labelling for comparisons. Task is formulated as a classical sequence tagging organized in the form of CodaLab competition[22]. The main goal is identifying objects, aspects, and predicates given an input sentence. For instance: [Python=OBJECT] is [better=PREDICATE] than [Matlab=OBJECT] for [Deep Learning=ASPECT]. Such kind of semantic role labelling is usually applied for comparative argument

[19] https://github.com/manujosephv/pytorch_tabular
[20] https://archive.ics.uci.edu/ml/datasets/bank+marketing
[21] https://codalab.lisn.upsaclay.fr
[22] https://codalab.lisn.upsaclay.fr/competitions/531

mining (Schildwächter et al., 2019).

Students are required to train a sequence labeling model on a provided labeled dataset. For this task, they can use any transformer-based model and experiment with different types of embedding initialization and the fine-tuning procedure. The provided data files are in CoNLL-U format. Each line contains one word, and its label is in BIO format for predicting "Objects", "Aspects" and "Predicates in the sentence".

In the latest edition of the course, this task was replaced by the KGQA task which is a binary classification, so, in principle the new assignment can nicely complement the sequence tagging task.[23]

5.1.2 Assignment 2: Text Detoxification

For the second assignment, students participate in the competition of automatic text detoxification (Dementieva et al., 2022). This task is seq2seq style transfer task: its required to paraphrase a sentence from the toxic (i.e. rude) to the non-toxic (ie. neutral) style while preserving its meaning. Such textual style transfer can be used to process toxic content on social media.

In the assignment context, students need to train a model and submit its output to the CodaLab competition[24]. They are free to use any methods and/or models for style transfer or pre-trained models for text generation (GPT (Radford et al., 2019), T5 (Raffel et al., 2020a), etc.). The competition provides baselines that may be improved. Otherwise, the students are allowed to rely on them when composing their own solutions.

In the last edition of the course, the students were asked to participate in the multilingual text detoxification shared task at CLEF 2024 (Bevendorff et al., 2024) where 9 languages to be supported instead of a single one.[25]

5.2 Final Quiz

The final session is followed by a comprehensive quiz covering all topics studied. It consists of 26 multiple-choice questions (1 point for each question). Each topic covered in the course is presented in the quiz with one or two questions. We keep the list of questions closed to avoid revealing them to the current running of the course.

[23] https://codalab.lisn.upsaclay.fr/competitions/18214
[24] https://codalab.lisn.upsaclay.fr/competitions/642
[25] https://codalab.lisn.upsaclay.fr/competitions/18243

5.3 Bonus Lecture Quizzes and Practical Tasks

After each session, students are given a lecture quiz consisting of ten multiple-choice questions on the topic and a short practical task, which usually includes several simple experiments. Such activities allow students to revise the material and gain small extra points (1 point maximum for each lecture quiz or practical task).[26]

6 Expected Outcomes

First, we expect the students to acquire a comprehensive understanding of the transformer-based models and the underlying mechanisms and to get acquainted with a diverse set of Transformer architectures. Second, we expect them to learn how to train and apply Transformers to multiple NLP tasks and how to adapt them to other domains. Third, we anticipate that the students will be able to use pre-trained models from the HuggingFace library and to employ other tools or datasets from the HuggingFace project.

7 Formal Course Evaluation

At the end of each running of the course, we collect feedback by asking students to complete a short survey. Figure 2 presents the aggregated results for all industrial runs. We can see that most students are satisfied with the course and find it quite engaging. The average rating is 8.5 out of 10, and the highest grade of 10 accounts for 44 percent of all ratings. The feedback from the academic run is also strongly positive (See Appendix B); all students note clear objectives and explanations, challenging enough content, and grading criteria. Both industry (94%) and academia (100%) respondents note the usefulness of practical skills and the quality of the course organization and teaching (87% for industry and 89% for academia).

8 Academic vs Industrial Students

We summarize qualitative differences in expectations of the course between students from industry and academia below (based on obtained feedback and evaluation comments):

- Students from industry demand more practical programming materials and sessions, being

[26] We provide free access to quizzes and tasks upon a request from an academic email to professors, lecturers, and teachers.

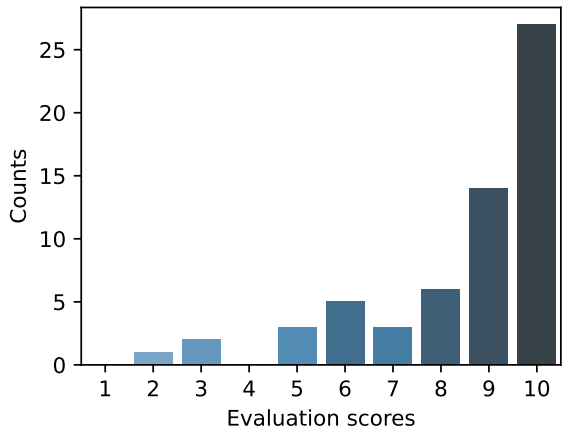

Figure 2: Students feedback for the course in the industrial setup. 10 is the maximum, 1 is the minimum score.

less happy with dense lectures than academic students who are used to such format.

- Professionals asked for a translation of terms and materials to their native language, while academics didn't mind English.

- Industrial students were asking more questions during lectures and chat discussions.

- In competition results (e.g., for the shared task on text detoxification), the leaderboard was a mixture of industrial and academic students with no apparent leader.

- Attendance in percentage was more significant for industrial students (while industrial sessions were in the working days afternoon, 18-21 time slot while for academic students were during the day, usually 16-19 time slot), indicating overall greater motivation/commitment of professionals.

- Industrial students often are determined and specifically seeking for application of a particular task (e.g., motivated by the job project). In contrast, academic students do not have such "extra" learning goals, with a few exceptions where it is required for their research.

- For both types of students, not as much the topic matter so much its presentation. Even "hottest" lecture on how ChatGPT works may get very variable levels of student involvement, depending on the instructor.

9 Conclusion

This paper describes a course on Transformer models, initially designed in early 2022 and updated in the subsequent runs. During lectures and practical sessions, we present a comprehensive overview of transformer-related concepts and a variety of Transformer applications, including practical industrial use cases, covering both NLP and language modeling, as well as other domains, such as computer vision and processing of event sequences. The course was run several times for both academic and industrial audiences.

The theoretical outcome of the course for the students is a deep understanding of the Transformer architecture, the attention mechanism, and knowledge of a diverse set of transformer-based models and their adaptations for various domains. As a practical outcome, the students acquire diverse skills in working with all types of transformer-based models and using Transformers for other modalities and domains. The feedback about the course from the students from both industry and academia was generally positive yet different in various aspects, such as the desired balance between theory and practice (with industrial learners being more proactive and demanding hands-on skills).

In the future, we plan to add lectures related to newer Transformer models and more applications to other modalities and domains.

10 Acknowledgements

First, we thank David Dale, who prepared and delivered about half of the lectures and seminars in the first run of the industrial course (partially based on materials from the Skoltech course on Neural NLP, among other sources). Indeed, without David, this course would not seen the light.

Second, we would like to acknowledge Anton Razzhigaev's essential contribution to the lecture and seminar related to multimodal and vision Transformers.

Finally, we acknowledge contributions of Ilyar Alimova and Vladislav Zhuzhel, who were instructors of several sessions in some runs of the industrial part of the course. Finally, we would like to acknowledge the TAs of the academic part of the course: Mikhail Salnikov, Maria Lysuyk, Sergey Petrakov, Daniil Moscovskiy, and Daniil Larionov.

References

AIRI. 2022. Rudolph: One hyper-tasking transformer can be creative as dall-e and gpt-3 and smart as clip. https://github.com/ai-forever/ru-dolph.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. 2022. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Ekaterina Artemova, Murat Apishev, Veronika Sarkisyan, Sergey Aksenov, Denis Kirjanov, and Oleg Serikov. 2021. Teaching a massive open online course on natural language processing. *CoRR*, abs/2104.12846.

Sanjith Athlur, Nitika Saran, Muthian Sivathanu, Ramachandran Ramjee, and Nipun Kwatra. 2021. Varuna: Scalable, low-cost training of massive deep learning models. *CoRR*, abs/2111.04007.

Dmitrii Babaev, Nikita Ovsov, Ivan Kireev, Mariya Ivanova, Gleb Gusev, Ivan Nazarov, and Alexander Tuzhilin. 2022. Coles: Contrastive learning for event sequences with self-supervision. In *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*, pages 1190–1199. ACM.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Fabian Barteld and Johanna Flick. 2017. LEA - linguistic exercises with annotation tools. In *Proceedings of the Workshop on Teaching NLP for Digital Humanities (Teach4DH) 2017, Berlin, Germany, September 12, 2017*, volume 1918 of *CEUR Workshop Proceedings*, pages 11–16. CEUR-WS.org.

Janek Bevendorff, Xavier Bonet Casals, Berta Chulvi, Daryna Dementieva, Ashaf Elnagar, Dayne Freitag, Maik Fröbe, Damir Korenčić, Maximilian Mayerl, Animesh Mukherjee, Alexander Panchenko, Martin Potthast, Francisco Rangel, Paolo Rosso, Alisa Smirnova, Efstathios Stamatatos, Benno Stein, Mariona Taulé, Dmitry Ustalov, Matti Wiegmann, and Eva Zangerle. 2024. Overview of pan 2024: Multi-author writing style analysis, multilingual text detoxification, oppositional thinking analysis, and generative ai authorship verification. In *Advances in Information Retrieval*, pages 3–10, Cham. Springer Nature Switzerland.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Ales Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*, pages 12–58. The Association for Computer Linguistics.

Bozhidar Bozhanov and Ivan Derzhanski. 2013. Rosetta stone linguistic problems. In *Proceedings of the Fourth Workshop on Teaching NLP and CL*, pages 1–8, Sofia, Bulgaria. Association for Computational Linguistics.

Shaked Brody, Uri Alon, and Eran Yahav. 2022. How attentive are graph attention networks?

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. 2016. Training deep nets with sublinear memory cost. *CoRR*, abs/1604.06174.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *CoRR*, abs/1912.02164.

Daryna Dementieva, Irina Nikishina, Varvara Logacheva, Alena Fenogenova, David Dale, Irina Krotova, Nikita Semenov, Tatiana Shavrina, and Alexander Panchenko. 2022. Russe-2022: Findings of the first russian detoxification task based on parallel corpora. In *Computational Linguistics and Intellectual Technologies*, pages 114–131.

ONNX Runtime developers. 2021. Onnx runtime. https://onnxruntime.ai/. Version: x.y.z.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Ivan Fursov, Matvey Morozov, Nina Kaploukhaya, Elizaveta Kovtun, Rodrigo Rivera-Castro, Gleb Gusev, Dmitry Babaev, Ivan Kireev, Alexey Zaytsev, and Evgeny Burnaev. 2021. Adversarial attacks on deep

models for financial transaction records. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 2868–2878. ACM.

Yarin Gal. 2016. *Uncertainty in Deep Learning*. Ph.D. thesis, University of Cambridge.

Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NeurIPS 2017, page 4885–4894, Red Hook, NY, USA. Curran Associates Inc.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023. Imagebind one embedding space to bind them all. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 15180–15190. IEEE.

Yury Gorishniy, Ivan Rubachev, and Artem Babenko. 2022. On embeddings for numerical features in tabular deep learning. In *NeurIPS*.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2021. Revisiting deep learning models for tabular data. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 18932–18943.

Benjamin Hawks, Javier M. Duarte, Nicholas J. Fraser, Alessandro Pappalardo, Nhan Tran, and Yaman Umuroglu. 2021. Ps and qs: Quantization-aware pruning for efficient low latency neural network inference. *Frontiers Artif. Intell.*, 4:676564.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar S. Karnin. 2020. Tabtransformer: Tabular data modeling using contextual embeddings. *CoRR*, abs/2012.06678.

Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907.

Jing Yu Koh, Ruslan Salakhutdinov, and Daniel Fried. 2023. Grounding language models to images for multimodal inputs and outputs. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 17283–17300. PMLR.

Nikita Konodyuk and Maria Tikhonova. 2021. Continuous prompt tuning for russian: How to learn prompts efficiently with rugpt3? In *Recent Trends in Analysis of Images, Social Networks and Texts - 10th International Conference, AIST 2021, Tbilisi, Georgia, December 16-18, 2021, Revised Supplementary Proceedings*, volume 1573 of *Communications in Computer and Information Science*, pages 30–40. Springer.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

François Lagunas, Ella Charlaix, Victor Sanh, and Alexander M. Rush. 2021. Block pruning for faster transformers.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, volume 31, pages 7167–7177.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *CoRR*, abs/2305.19187.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT understands, too. *CoRR*, abs/2103.10385.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Federico Martelli, Najla Kalach, Gabriele Tola, Roberto Navigli, et al. 2021. Semeval-2021 task 2: Multilingual and cross-lingual word-in-context disambiguation (mcl-wic). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 24–36.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory F. Diamos, Erich Elsen, David García, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed precision training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Roger Newson. 2016. Xglm: Stata module to extend glm.

Ivan V. Oseledets. 2011. Tensor-train decomposition. *SIAM J. Sci. Comput.*, 33(5):2295–2317.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, H. Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *CoRR*, abs/2112.11446.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020a. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020b. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125.

Anton Razzhigaev, Arseniy Shakhmatov, Anastasia Maltseva, Vladimir Arkhipkin, Igor Pavlov, Ilya Ryabov, Angelina Kuts, Alexander Panchenko, Andrey Kuznetsov, and Denis Dimitrov. 2023. Kandinsky: An improved text-to-image synthesis with image prior and latent diffusion. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 286–295, Singapore. Association for Computational Linguistics.

Nils Reiter, Sarah Schulz, Gerhard Kremer, Roman Klinger, Gabriel Viehhauser, and Jonas Kuhn. 2017. Teaching computational aspects in the digital humanities program at university of stuttgart - intentions and experiences. In *Proceedings of the Workshop on Teaching NLP for Digital Humanities (Teach4DH) 2017, Berlin, Germany, September 12, 2017*, volume 1918 of *CEUR Workshop Proceedings*, pages 43–48. CEUR-WS.org.

Victor Sanh, Thomas Wolf, and Alexander M. Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100.

Matthias Schildwächter, Alexander Bondarenko, Julian Zenker, Matthias Hagen, Chris Biemann, and Alexander Panchenko. 2019. Answering comparative questions: Better than ten-blue-links? In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, pages 361–365.

Burr Settles. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4603–4611. PMLR.

Oleksandr Shchur, Ali Caner Türkmen, Tim Januschowski, and Stephan Günnemann. 2021. Neural temporal point processes: A review. In

Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, pages 4585–4593. ijcai.org.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mgpt: Few-shot learners go multilingual. *CoRR*, abs/2204.07580.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *CoRR*, abs/1909.08053.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize from human feedback. *CoRR*, abs/2009.01325.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020. Efficient transformers: A survey. *CoRR*, abs/2009.06732.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Elena Tutubalina, Ilseyar Alimova, Zulfat Miftahutdinov, Andrey Sakhovskiy, Valentin Malykh, and Sergey I. Nikolenko. 2020. The russian drug reaction corpus and neural models for drug reactions and effectiveness detection in user reviews. *CoRR*, abs/2004.03659.

Sowmya Vajjala. 2021. Teaching NLP outside linguistics and computer science classrooms: Some challenges and some opportunities. *CoRR*, abs/2105.00895.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6306–6315.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Artem Vazhentsev, Gleb Kuzmin, Akim Tsvigun, Alexander Panchenko, Maxim Panov, Mikhail Burtsev, and Artem Shelmanov. 2023. Hybrid uncertainty quantification for selective text classification in ambiguous tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11659–11681, Toronto, Canada. Association for Computational Linguistics.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022a. OFA: unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 23318–23340. PMLR.

Zheng Wang, Juncheng B. Li, Shuhui Qu, Florian Metze, and Emma Strubell. 2022b. Squat: Sharpness- and quantization-aware training for BERT. *CoRR*, arXiv:2210.07171.

Genta Indra Winata, Andrea Madotto, Zhaojiang Lin, Rosanne Liu, Jason Yosinski, and Pascale Fung. 2021. Language models are few-shot multilingual learners. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 1–15, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. 2021. The art of abstention: Selective prediction and error regularization for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1040–1051, Online. Association for Computational Linguistics.

KiYoon Yoo, Jangho Kim, Jiho Jang, and Nojun Kwak. 2022. Detection of adversarial examples in text classification: Benchmark and baseline via robust density estimation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3656–3672, Dublin, Ireland. Association for Computational Linguistics.

Seongjun Yun, Minbyul Jeong, Sungdong Yoo, Seunghun Lee, Sean S. Yi, Raehyun Kim, Jaewoo Kang, and Hyunwoo J. Kim. 2022. Graph transformer networks: Learning meta-path graphs to improve gnns. *Neural Networks*, 153:104–119.

Jiawei Zhang, Haopeng Zhang, Congying Xia, and Li Sun. 2020. Graph-bert: Only attention is needed for learning graph representations. *CoRR*, abs/2001.05140.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *Preprint*, arXiv:1912.08777.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D. Manning, and Jure Leskovec. 2022. Greaselm: Graph reasoning enhanced language models for question answering. *CoRR*, abs/2201.08860.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized bert pre-training approach with post-training. In *Proceedings of the 20th chinese national conference on computational linguistics*, pages 1218–1227.

Vladislav Zhuzhel, Rodrigo Rivera-Castro, Nina Kaploukhaya, Liliya Mironova, Alexey Zaytsev, and Evgeny Burnaev. 2021. COHORTNEY: deep clustering for heterogeneous event sequences. *CoRR*, abs/2104.01440.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. 2020. Transformer hawkes process. *CoRR*, abs/2002.09291.

A Assessment criteria

A.1 Technical Report Grading Criteria

The technical report is evaluated via the two criteria:

- Methodology (5 points): description of all methods they try and the best method. Here, students can include some tricks with pre-processing, a description of the models and motivation of their usage, and details of the training process (train-test split, cross-validation, etc.).
- Discussion of results (5 points): here, we expect the final comparison table. Even if some methods did not bring students to the top of the leaderboard, they should nevertheless indicate this result and a discussion of why, in their opinion, some approaches work while others fail.

A.2 Code Grading Criteria

The code of the students is graded according to the following criteria:

- Readability (5 points): code should be well-structured, preferably with indicated parts of your approach (Pre-processing, Model training, Evaluation, etc.).
- Reproducibility (5 points): code should be reproduced without any mistakes with the "Run all" mode (obtaining experimental part).

A.3 CodaLab Competition Grading Criteria

Students get points for participating in the corresponding CodaLab competition. For example, a student receives 5 points for outperforming the baseline, an additional 5 points for being in the top 20% on the leaderboard, or an additional 10 points for being top-1. As a result, students may get 0-15 points depending on their performance.

B Feedback

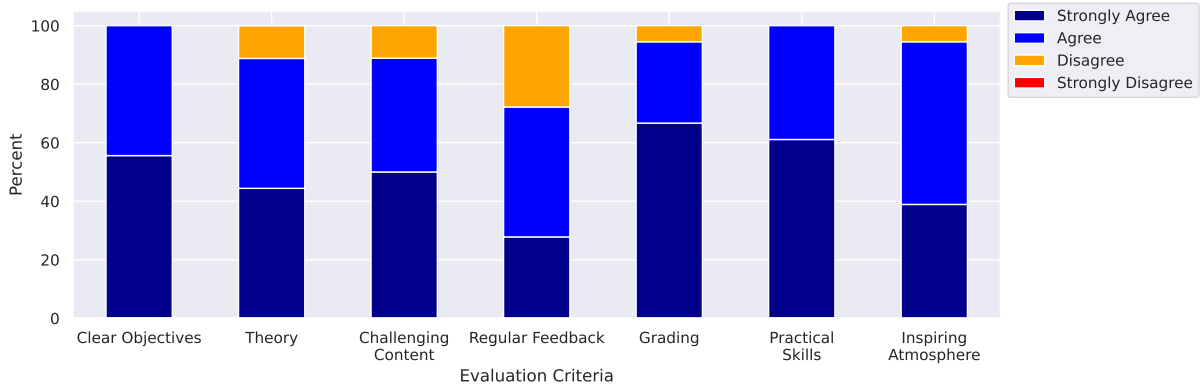

Figure 3: Students feedback for the course in academic setup.

After the academic running of the course (2023), students were asked to provide feedback. Figure 3 demonstrates the statistics of students' feedback for each question, which are listed below:

1. Course objectives were clear to me.
2. Key concepts and theories were well explained by the Course instructor(s).
3. Course content was difficult enough to be challenging.
4. I regularly received feedback on my performance.

5. Grading criteria were well explained, and I understood what action was required to achieve each of the performance levels.
6. The course was useful in developing practical skills.
7. The Course atmosphere was inspiring for active learning.