– IN5550 –
Neural Methods in Natural Language Processing
Home Exam: Task Overview and Kick-Off

Andrey Kutuzov, Vladislav Mikhailov, Roxana Pop, David Samuel,

& Sondre Wold

University of Oslo

April 29th, 2024



All home exam info is on the semester page



Final exam

Obligatory assign-ments

slides

videos

IN5550 - Nevrale metoder i språkprosessering

Final exam

Between Monday April 29 and Monday June 3, there will be a home exam in the form of a small research project. To pass the exam, students need to submit and have accepted a scientific paper to the 6th IN5550 Workshop on Neural Natural Language Processing (WNNLP 2024).

Background

Final examination in this class takes the form of a 'home exam', i.e. a project that students can work on over a period of three weeks. Like for the exam-qualifying obligatory assignments, group work (in teams of up to three students) is encouraged (but **PhD-students** enrolled for **IN9550** must complete the home exam **individually**). Team composition needs to be declared at the start of the exam period and, cannot be changed after May 1st, 2024. Also, each team needs to decide beforehand which of the available topics they want to research; these 'tracks' of the exam will be introduced in the lecture on April 29 (examples from previous years have included named entity recognition, definition generation, and sentiment analysis). Further background on the tracks and supporting data and code will be available through the course GitHub repo. Please announce your team composition and choice of track no later than May 1st by emailing the course contact address `in5550-help@ifi.uio.no`.

The main exam period will be Monday, April 29 to Monday, May 20, 2024. Once a

Home Exam

General Idea

- Use as guiding metaphor: Preparing a scientific paper for publication.

Home Exam

General Idea

- Use as guiding metaphor: Preparing a scientific paper for publication.

Sixth IN5550 Teaching Workshop on Neural NLP (WNNLP 2024)

Home Exam

General Idea

- Use as guiding metaphor: Preparing a scientific paper for publication.

Sixth IN5550 Teaching Workshop on Neural NLP (WNNLP 2024)

Standard Process

(0) Problem Statement

(1) Experimentation

(2) Analysis

(3) Paper Submission

(4) Reviewing

(5) Camera-Ready Manuscript

(6) Presentation

WNNLP 2024: Call for Papers

General Constraints

- Five specialized tracks:
  - Pretraining tiny language models
  - Scandinavian language identification
  - Fact-checking with graph evidence
  - Artificial text detection
  - Predicting a prompt from a generated text
- Long papers: up to 8 pages (minimally 5), excluding references, in ACL Rolling Review style.
- Submitted papers must be anonymous:
  - peer reviewing is double-blind.
- Replicability: Submission backed by code repository (track chairs only).

WNNLP 2024: Important Dates

Schedule

By May 1st	Declare choice of track (and team composition)
Week 19 or 20	Track specific joint mentoring meetings
May 20	(Strict) Submission deadline for scientific papers
May 21–24	Reviewing period: Each student reviews two papers
May 27	Track Chairs announce acceptance decisions
May 31	Camera-ready papers due, with requested revisions
June 3	Oral presentations and awards at the workshop

All dates AOE (anywhere on Earth).

WNNLP 2024: Programme Committee

General Chair

- Andrey Kutuzov

Track Chairs

1. Pretraining tiny language models: Lucas Georges Gabriel Charpentier
2. Scandinavian language identification: David Samuel and Petter Mæhlum
3. Fact-checking for English with knowledge graphs: Roxana Pop and Sondre Wold
4. Artificial text detection: Vladislav Mikhailov
5. Predicting a prompt from a generated text: Andrey Kutuzov and Louis Give

Peer Reviewers

- All students who have submitted a scientific paper

Contents

1. Track 1: Pretraining tiny language models
2. Track 2: Scandinavian language identification
3. Track 3: Fact-checking with graph-evidence
4. Track 4: Artificial text detection
5. Track 5: Predicting a prompt from a generated text

Track 1: Pretraining tiny language models

Track chair: Lucas Georges Gabriel Charpentier

See more in the detailed task description:

`https://github.uio.no/in5550/2024/tree/main/exam/tiny_lms`

Track 1: Pretraining tiny language models

Goal

- Pretrain a language with the architecture of your choice.
- Evaluate on the two evaluation tasks provided.

Restrictions

- Your models have to have 7M parameters or less.
- The training time should not exceed 5 GPU hours (i.e. 5 hours on a single GPU).

Track 1: Pretraining tiny language models

Provided Data

- The BabyLM 2024 challenge data for the small-strict track (10M words).
- You can use other data if you think it is useful.

Evaluation

- Both tasks are zero-shot and can be run on your own computer.
- BLiMP: Uses minimal pair differences to evaluate the grammatical knowledge of a model [Warstadt et al., 2020].
- LAMBADA: Verifies the ability of the model to solve long context dependencies [Paperno et al., 2016]

Track 1: Pretraining tiny language models

Source	Weight	Word Count	Domain
BNC	8%	800K	Dialogue
CHILDES [MacWhinney, 2000]	29%	2.9M	Dialogue, Child Directed
Project Gutenberg [Gerlach and Font-Clos, 2020]	26%	2.6M	Fiction, Nonfiction
OpenSubtitles [Lison and Tiedemann, 2016]	20%	2M	Dialogue, Scripted
Simple English Wikipedia	15%	1.5M	Nonfiction
Switchboard [Godfrey et al., 1992, Stolcke et al., 2000]	1%	100K	Dialogue

Table: Number of sentences across the datasets and their splits.

Track 1: Pretraining tiny language models

Task	Random Decoder	Random Encoder	Random Encoder–Decoder
LAMBADA			
Accuracy	0.00	0.00	0.00
Perplexity	3578.65	3611.27	3567.16
BLiMP			
Linguistic Term			
Ellipsis	40.10	54.80	44.30
Binding	45.86	61.84	47.91
Quantifiers	36.50	55.93	34.85
Argument Structure	54.50	51.26	52.31
Subject Verb Agreement	48.23	50.00	48.45
Anaphor Agreement	60.30	64.25	65.60
Filler Gap Dependency	57.40	52.63	48.80
Determiner Noun Agreement	49.44	50.08	49.99
NPI Licensing	55.96	43.86	37.09
Island Effects	53.24	75.83	48.15
Control Raising	56.30	55.96	53.56
Irreglar Forms	55.30	44.65	51.95
Field			
Syntax	53.78	59.46	49.47
Syntax/Semantics	49.65	60.66	48.48
Semantics	49.46	43.80	35.94
Morphology	50.89	51.02	51.43
Average	51.56	55.34	47.97

Table: Results on evaluation metrics of randomly initialized decoder, encoder, and encoder–decoder models.

Track 1: Pretraining tiny language models

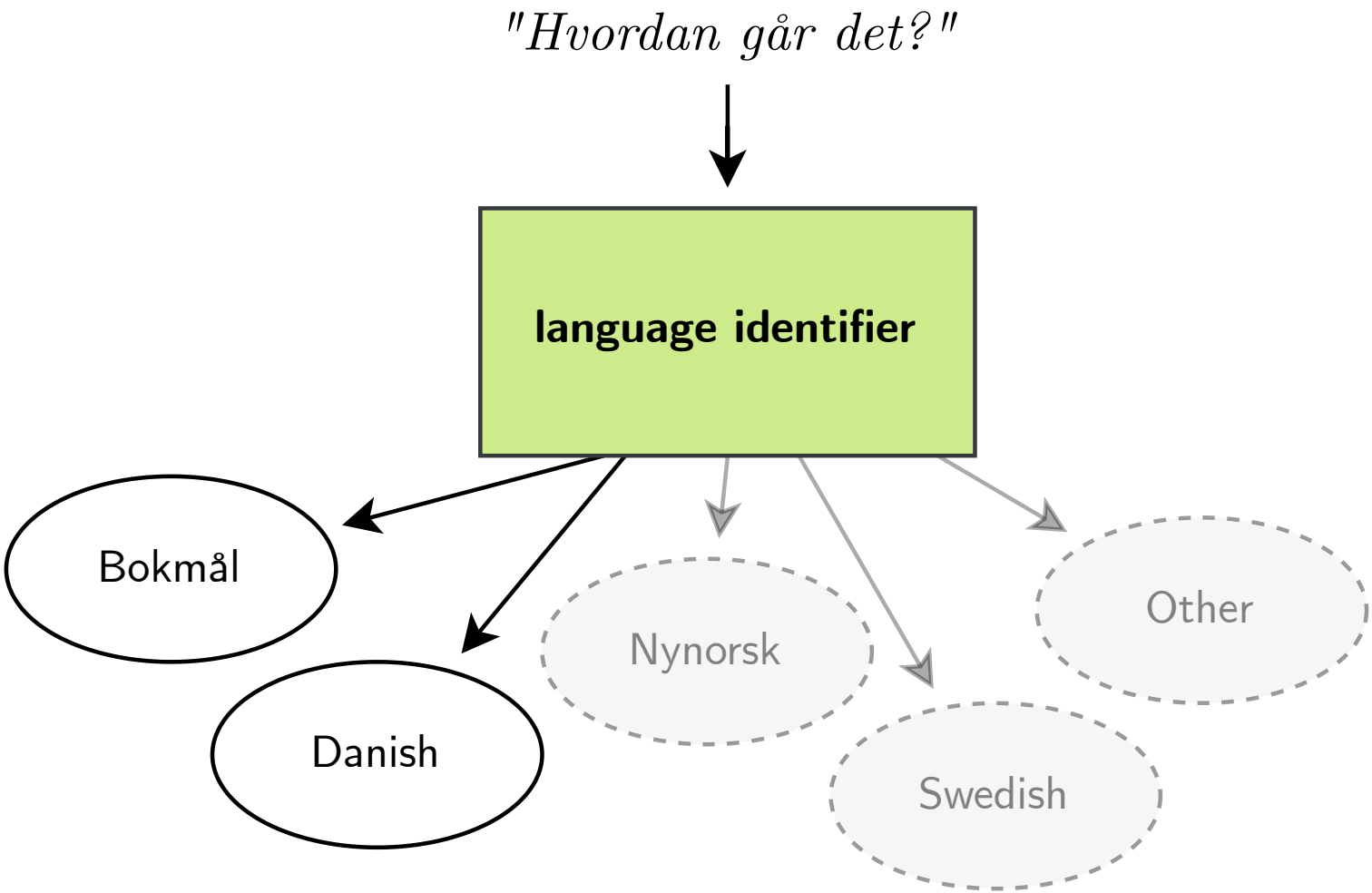
Task	Trained Decoder	Trained Encoder	Trained Encoder–Decoder
LAMBADA			
Accuracy	**7.53**	0.00	4.66
Perplexity	22.64	2349.43	32.26
BLiMP			
Linguistic Term			
Ellipsis	44.85	**71.60**	61.85
Binding	48.94	**75.69**	64.93
Quantifiers	53.10	47.73	**73.85**
Argument Structure	53.14	49.19	**60.60**
Subject Verb Agreement	50.40	51.57	**54.95**
Anaphor Agreement	**85.50**	36.75	53.25
Filler Gap Dependency	64.26	**71.43**	63.29
Determiner Noun Agreement	46.83	49.95	**70.35**
NPI Licensing	**59.24**	57.63	34.09
Island Effects	57.20	**72.04**	42.31
Control Raising	54.18	48.50	**58.72**
Irreglar Forms	52.05	42.85	**81.60**
Field			
Syntax	56.75	**63.93**	55.79
Syntax/Semantics	50.34	**66.40**	57.75
Semantics	**59.53**	48.56	52.62
Morphology	52.89	48.23	**64.57**
Average	54.75	**58.16**	58.13

Table: Results on evaluation metrics of the baseline decoder, encoder, and encoder–decoder models.

Contents

1. Track 1: Pretraining tiny language models
2. Track 2: Scandinavian language identification
3. Track 3: Fact-checking with graph-evidence
4. Track 4: Artificial text detection
5. Track 5: Predicting a prompt from a generated text

Track 2: Scandinavian language identification

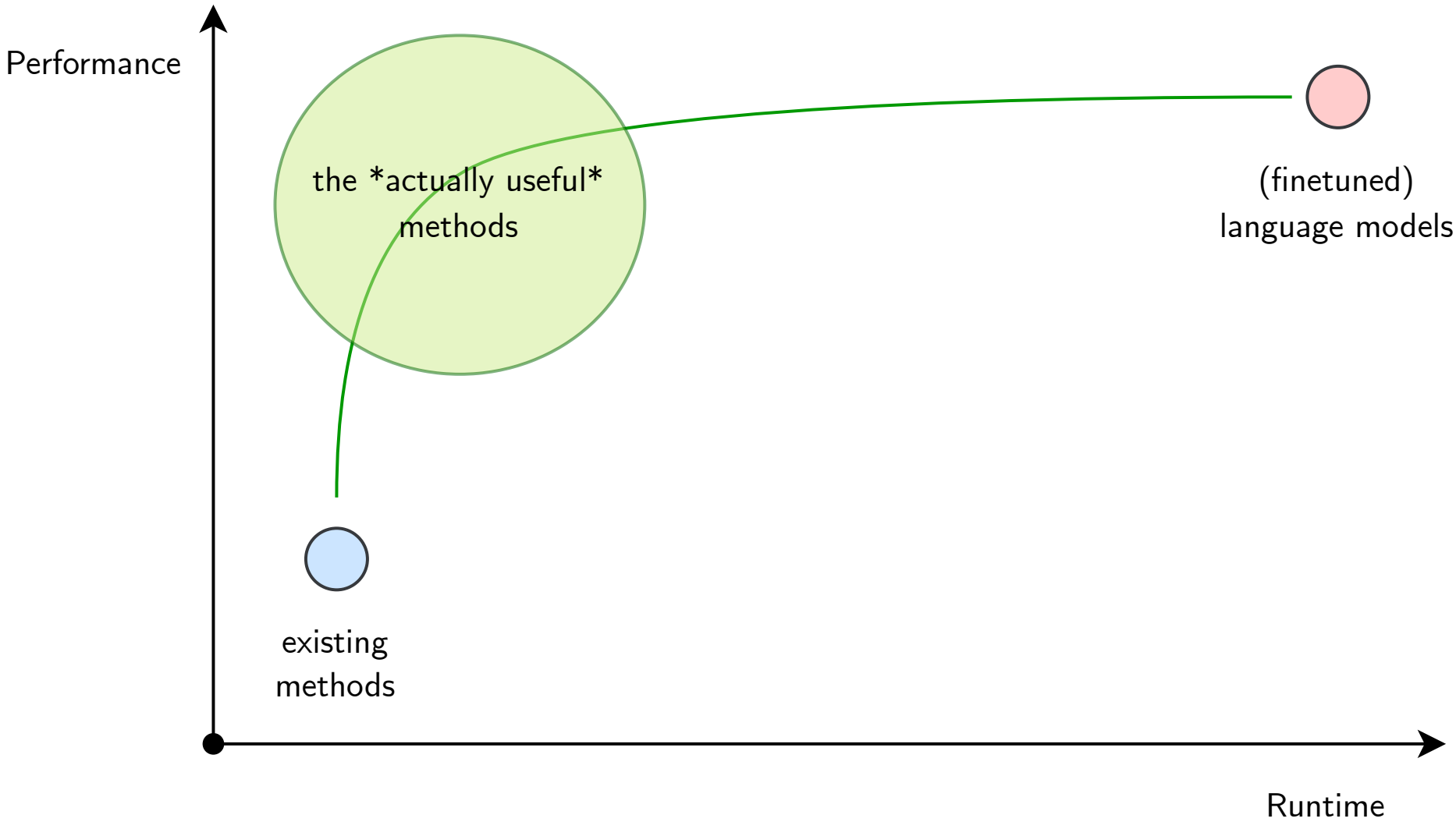


Track 2: Scandinavian language identification

- Might seem like a simple and boring task
- But in practice, it's very difficult and super important
  - It's the first step in many NLP pipelines – we can't a good MT model or LM without it
  - The existing identifiers are pretty terrible, which causes a lot of troubles

Track 2: Scandinavian language identification

An unscientific plot of language identifier runtime vs. performance



Track 2: Scandinavian language identification

Language	Gold train	Gold valid	Silver train	Evaluation set
Bokmål	15 435	2 392	22 860	1 939
Danish	4 376	563	34 173	565
Nynorsk	14 029	1 859	24 382	1 511
Swedish	4 278	503	34 252	1 219
Other	8 805	1 165	—	1 742

Table: Number of sentences across the datasets and languages. Every sentence in the training and validation splits is annotated with a single label, while the sentence in the evaluation set can be multi-labeled.

Track 2: Scandinavian language identification

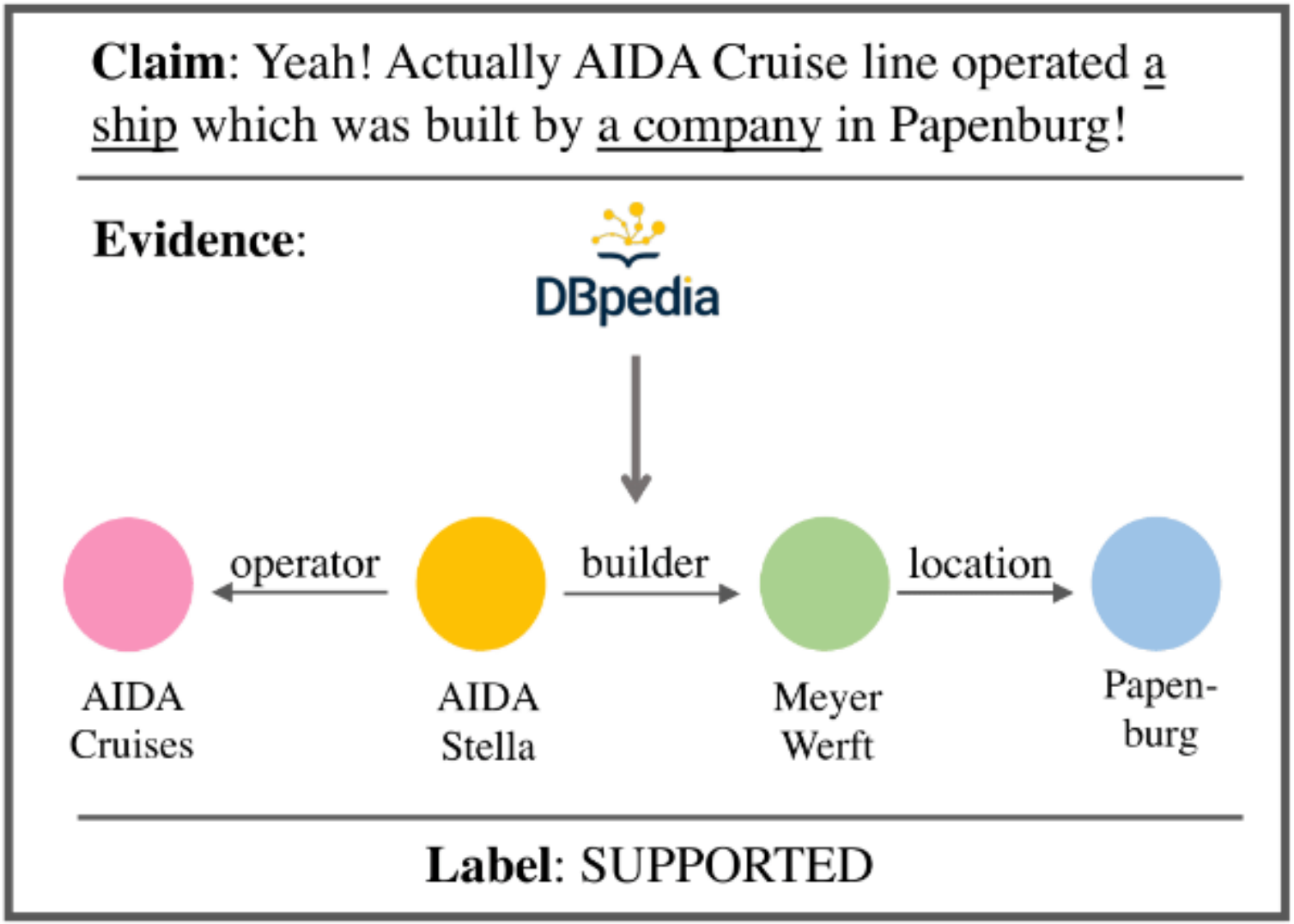
- Your mission: solve language identification and get to the *actually useful* area! :))
- There are lots of different directions to pursue, **See more in the detailed task description:** `https://github.uio.no/in5550/2024/tree/main/exam/scandinavian_language_identification`

Contents

1. Track 1: Pretraining tiny language models
2. Track 2: Scandinavian language identification
3. Track 3: Fact-checking with graph-evidence
4. Track 4: Artificial text detection
5. Track 5: Predicting a prompt from a generated text

Track 3: Fact-checking with graph-evidence

Will be using FactKG[Kim et al., 2023]



Track 3: Fact-checking with graph-evidence

Two settings:

1. Simplified setting
2. Full setting

Track 3: Fact-checking with graph-evidence

Two settings:

1. Simplified setting
   - input: Claim, Graphical evidence
   - output: label
2. Full setting
   - input: Claim, Entities
     - Must retrieve the subgraphs
     - Graphical evidence is given for training data
   - output: label

Track 3: Fact-checking with graph-evidence

Simplified setting data

Claim	**Subgraph evidence**	**Label**
Yeah! Actually AIDA Cruise line operated a ship which was built by a company in Papenburg!	$AIDA_Cruises$: $\sim$ $operator$, $builder$, $location$; $Papenburg$: $\sim$ $location$, $\sim$ $builder$, $operator$	Supported

Track 3: Fact-checking with graph-evidence

Baselines:

1. Simplified setting
   - claim-only
   - evidence-based
     - Uses paths constructed from the given relations and DBpedia
2. Full setting
   - claim-only

Track 3: Fact-checking with graph-evidence

Baselines:

1. Simplified setting
   - claim-only
   - evidence-based
     - Uses paths constructed from the given relations and DBpedia
2. Full setting
   - claim-only

All baselines employ a fine-tuned BERT model.

Track 3: Fact-checking with graph-evidence

Your improvements:

1. Simplified setting
   - Different ways of using the graphical evidence: encoding paths, subgraphs (GNNs), combination of the two
2. Full setting
   - Different ways of extracting relevant knowledge

Track 3: Fact-checking with graph-evidence

Your improvements:

1. Simplified setting
   - Different ways of using the graphical evidence: encoding paths, subgraphs (GNNs), combination of the two
2. Full setting
   - Different ways of extracting relevant knowledge

   For both these settings you can of course experiment with different language models, for both the claim and textual representations of paths!

Contents

1. Track 1: Pretraining tiny language models
2. Track 2: Scandinavian language identification
3. Track 3: Fact-checking with graph-evidence
4. Track 4: Artificial text detection
5. Track 5: Predicting a prompt from a generated text

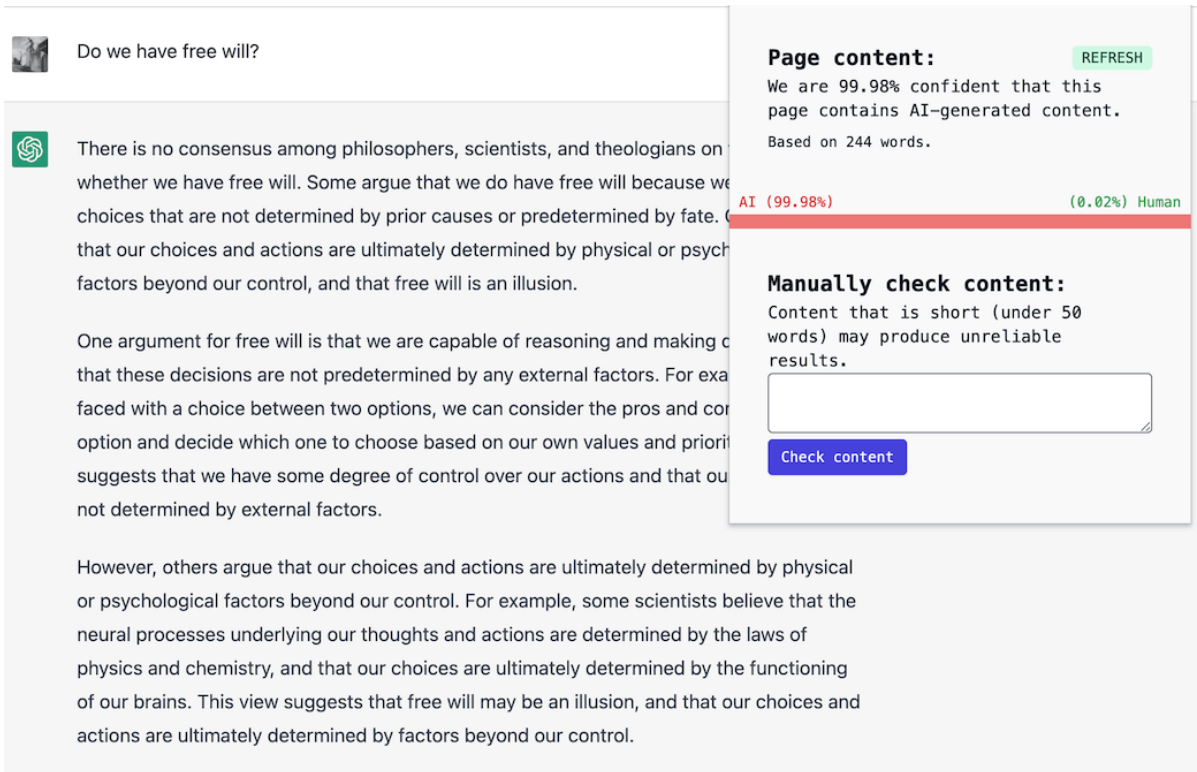
Track 4: Artificial text detection

Main info

Track chairs: Vladislav Mikhailov

See more in the detailed task description:

`https://github.uio.no/in5550/2024/tree/main/exam/artificial_text_detection`



Track 4: Artificial text detection

Background

- Modern generative LMs are widely used on a daily basis: creative writing, code-writing assistants, etc.
- Machine-generated data can be well-represented in pretraining and instruction-tuning corpora [Wang et al., 2023]
- However, such LMs are also widely misused for malicious purposes, e.g., generating fake news, product reviews, and abusive content.

Track 4: Artificial text detection

Background

- Modern generative LMs are widely used on a daily basis: creative writing, code-writing assistants, etc.
- Machine-generated data can be well-represented in pretraining and instruction-tuning corpora [Wang et al., 2023]
- However, such LMs are also widely misused for malicious purposes, e.g., generating fake news, product reviews, and abusive content.

Artificial Text Detection

- A niche field aimed to mitigate the risks of misusing LMs.
- Creating resources and artificial text detectors to assist humans in identifying generated content.

... the recent research is addressing more complex setups, including non-English languages and out-of-domain data sources and text generators.

Track 4: Artificial text detection

Goal

- Develop your own system for a task formulation of your choice.

Track 4: Artificial text detection

Goal

- Develop your own system for a task formulation of your choice.

Data

- SemEval-2024 Task 8: Multidomain, Multimodel and Multilingual Machine-Generated Text Detection [Wang et al., 2024]
- Robustness towards language, domain, and text generator.

Track 4: Artificial text detection

Goal

- Develop your own system for a task formulation of your choice.

Data

- SemEval-2024 Task 8: Multidomain, Multimodel and Multilingual Machine-Generated Text Detection [Wang et al., 2024]
- Robustness towards language, domain, and text generator.

Task Formulations

- Subtask A: is the text created by a human or a model?
  - Monolingual setup (English).
  - Multilingual setup (7 diverse languages).
- Subtask B: which model generated the text, or was it human who created it?
- Subtask C: when is the text continued by a model?

Track 4: Artificial text detection

General workflow

- Select one or more task formulations.
- Analyze the participants' submissions, think about how they can be improved or come up with your own solutions.
- Try various types of detectors (zero-shot, feature-based, and Transformer-based detectors) and their ensembles.
- Conduct ablation studies to analyze the performance w.r.t. robustness, training data mixtures, dataset criteria.
- Compare the performance of your resulting system against the corresponding public leaderboards.

... or compete with one another!

Contents

1. Track 1: Pretraining tiny language models
2. Track 2: Scandinavian language identification
3. Track 3: Fact-checking with graph-evidence
4. Track 4: Artificial text detection
5. Track 5: Predicting a prompt from a generated text

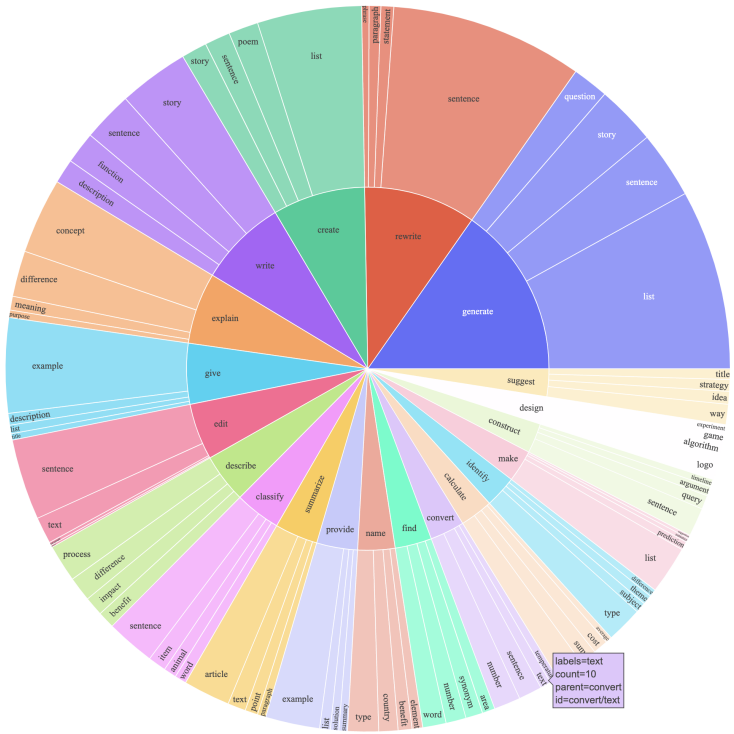
Track 5: Predicting a prompt from a generated text

Main info

Track chairs: Andrey Kutuzov and Louis Give

See more in the detailed task description:

`https://github.uio.no/in5550/2024/tree/main/exam/prompt_prediction`



Track 5: Predicting a prompt from a generated text

The main goal

To predict the prompt used to generate a given text with a given LLM

Input (LLM generated text)	Output (Prompt)
Dear [Hiring Manager's Name], I am writing to express my keen interest in the entry-level Financial Analyst position at [Company Name], as advertised. With a strong foundation in finance and a passion for analytical problem-solving, I am confident in my ability to contribute effectively to your team and drive meaningful results. [...]	Write a cutting edge cover letter for an entry-level role as a financial analyst. Ensure it makes my application stand out.

Track 5: Predicting a prompt from a generated text



Why?

- to better understand automatically generated texts
- to reveal patterns in the generation of harmful and fake content
- related to artificial text detection

Track 5: Predicting a prompt from a generated text

Why?

- to better understand automatically generated texts
- to reveal patterns in the generation of harmful and fake content
- related to artificial text detection

Background

- Prompt tuning [Lester et al., 2021, Ha et al., 2023, Madaan et al., 2023]
- Prefix tuning [Li and Liang, 2021]

.... but not many systems directly aimed at recovering the prompt from generated text.

Track 5: Predicting a prompt from a generated text

Possible data sources

- human-generated prompt/instruction and response datasets
  - e.g., `databricks-dolly-15k` [Conover et al., 2023]
- synthetic datasets
  - `Stanford Alpaca`, etc
- New prompts can be generated by paraphrasing the old ones.
- or from scratch by an LLM of your choice.

Would be interesting to compare simple prompts and creative prompts (*'Write me a poem'*)

Track 5: Predicting a prompt from a generated text

General workflow

- Try LLMs in a zero-shot or few-shot setups
- Try to fine-tune LLMs on the prompt prediction task
- Fine-tuning large models with LoRA and other efficient fine-tuning techniques (to fit on Fox)
- Standard NLG evaluation metrics.
- Research questions:
  - Is it really feasible to recover original prompts from generated text?
  - Does it make more sense to cast this problem as a classification task or as a sequence to sequence task (generation)?

WNNLP 2024: What Makes a Good Scientific Paper?



Empirical (Experimental)

- Motivate architecture choice(s) and hyper-parameters;
- systematic exploration of relevant parameter space;
- comparison to reasonable baseline or previous work.

Replicable (Reproducible)

- Everything relevant to run and reproduce in GitHub.

Analytical (Reflective)

- Identify and relate to previous work;
- explain choice of baseline or points of comparison;
- meaningful, precise discussion of results;
- 'negative' results can be interesting too;
- look at the data: discuss some examples:
- error analysis: identify remaining challenges.

References I


Conover, M., Hayes, M., Mathur, A., Xie, J., Wan, J., Shah, S., Ghodsi, A., Wendell, P., Zaharia, M., and Xin, R. (2023).
Free dolly: Introducing the world's first truly open instruction-tuned llm.


Gerlach, M. and Font-Clos, F. (2020).
A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics.
Entropy, 22(1):126.


Godfrey, J. J., Holliman, E. C., and McDaniel, J. (1992).
Switchboard: Telephone speech corpus for research and development.
In Acoustics, speech, and signal processing, ieee international conference on, volume 1, pages 517–520. IEEE Computer Society.

References II



Ha, H., Lee, J., Han, W., and Chun, B.-G. (2023).
Meta-learning of prompt generation for lightweight prompt engineering on language-model-as-a-service.
In Bouamor, H., Pino, J., and Bali, K., editors, Findings of the Association for Computational Linguistics: EMNLP 2023, pages 2433–2445, Singapore. Association for Computational Linguistics.



Kim, J., Park, S., Kwon, Y., Jo, Y., Thorne, J., and Choi, E. (2023).
FactKG: Fact verification via reasoning on knowledge graphs.
In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 16190–16206, Toronto, Canada. Association for Computational Linguistics.

References III



Lester, B., Al-Rfou, R., and Constant, N. (2021).
The power of scale for parameter-efficient prompt tuning.
In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.



Li, X. L. and Liang, P. (2021).
Prefix-tuning: Optimizing continuous prompts for generation.
In Zong, C., Xia, F., Li, W., and Navigli, R., editors, 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.

References IV



Lison, P. and Tiedemann, J. (2016).
OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles.
In Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).



MacWhinney, B. (2000).
The childes project.
Computational Linguistics, 26(4):657–657.

References V



Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B. P., Hermann, K., Welleck, S., Yazdanbakhsh, A., and Clark, P. (2023).
Self-refine: Iterative refinement with self-feedback.
In Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, Advances in Neural Information Processing Systems, volume 36, pages 46534–46594. Curran Associates, Inc.



Paperno, D., Kruszewski, G., Lazaridou, A., Pham, N. Q., Bernardi, R., Pezzelle, S., Baroni, M., Boleda, G., and Fernández, R. (2016).
The LAMBADA dataset: Word prediction requiring a broad discourse context.
In Erk, K. and Smith, N. A., editors, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1:

References VI

Long Papers), pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.


Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Ess-Dykema, C. V., and Meteer, M. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech.
Computational linguistics, 26(3):339–373.


Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. (2023).
Self-instruct: Aligning language models with self-generated instructions.
In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

References VII


Wang, Y., Mansurov, J., Ivanov, P., Su, J., Shelmanov, A., Tsvigun, A., Afzal, O. M., Mahmoud, T., Puccetti, G., Arnold, T., Whitehouse, C., Aji, A. F., Habash, N., Gurevych, I., and Nakov, P. (2024). Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection.


Warstadt, A., Parrish, A., Liu, H., Mohananey, A., Peng, W., Wang, S.-F., and Bowman, S. R. (2020).
BLiMP: The benchmark of linguistic minimal pairs for English.
Transactions of the Association for Computational Linguistics, 8:377–392.