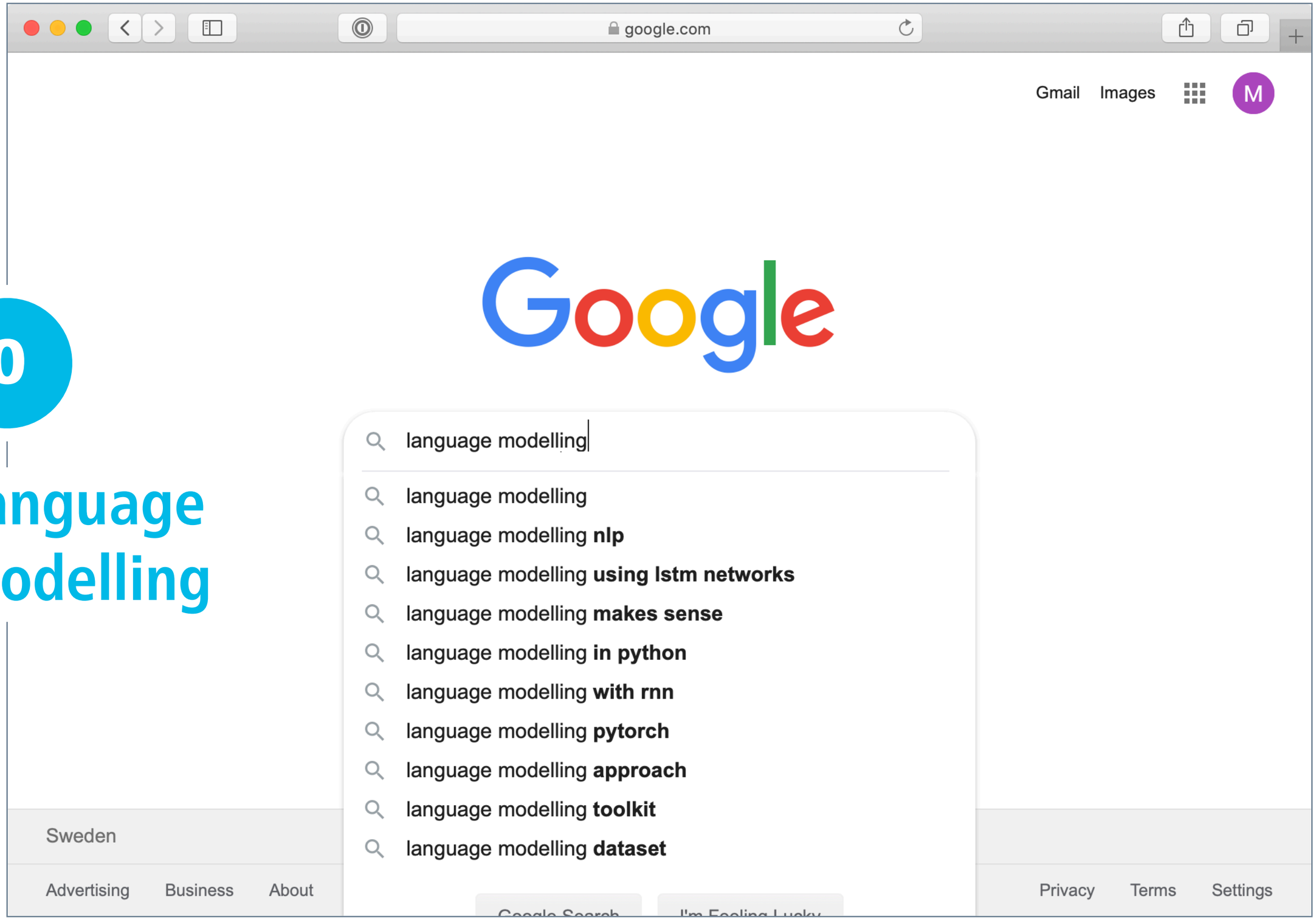
Natural Language Processing

Course overview

Marco Kuhlmann
Department of Computer and Information Science

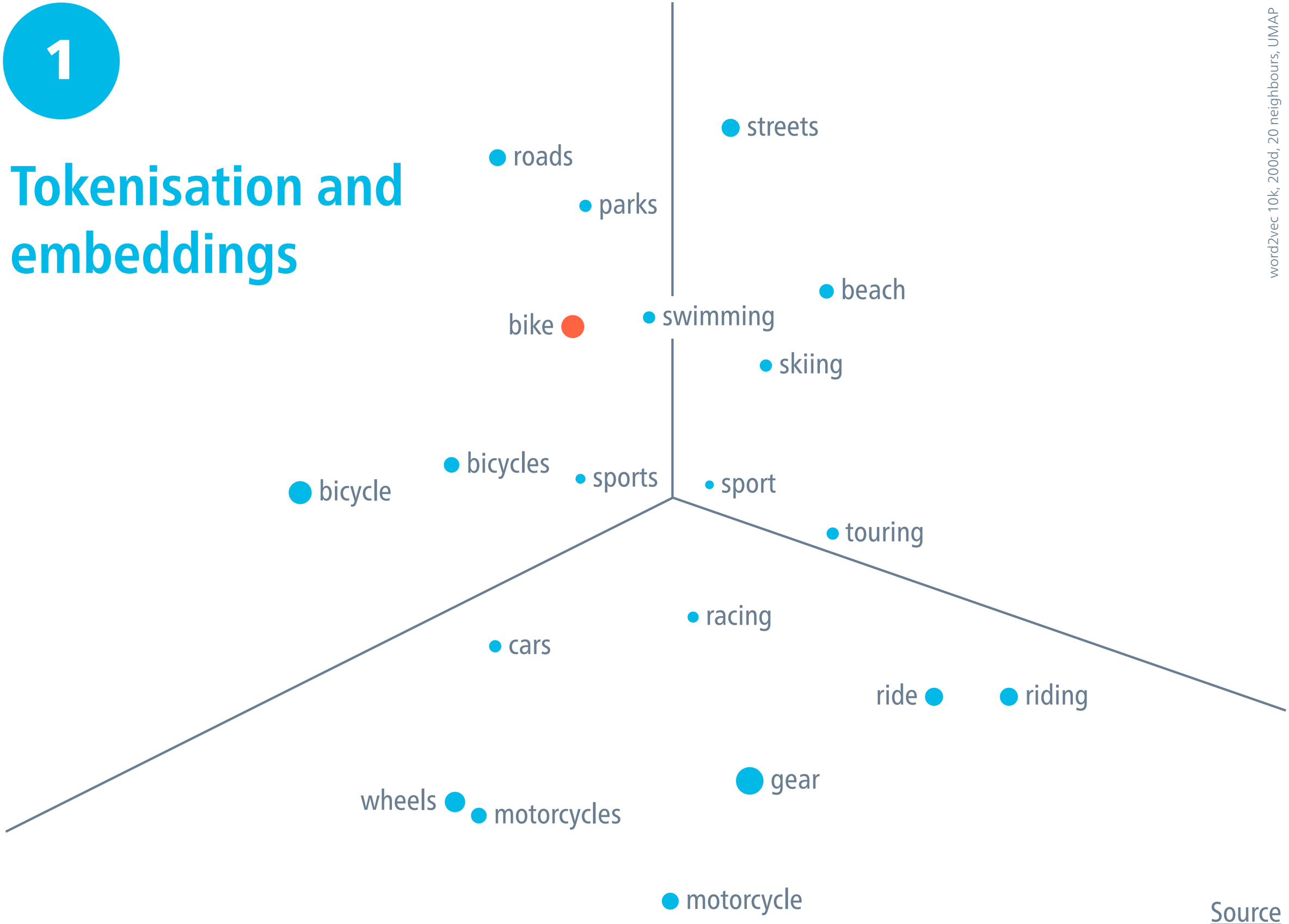
0

Language modelling



1

Tokenisation and embeddings

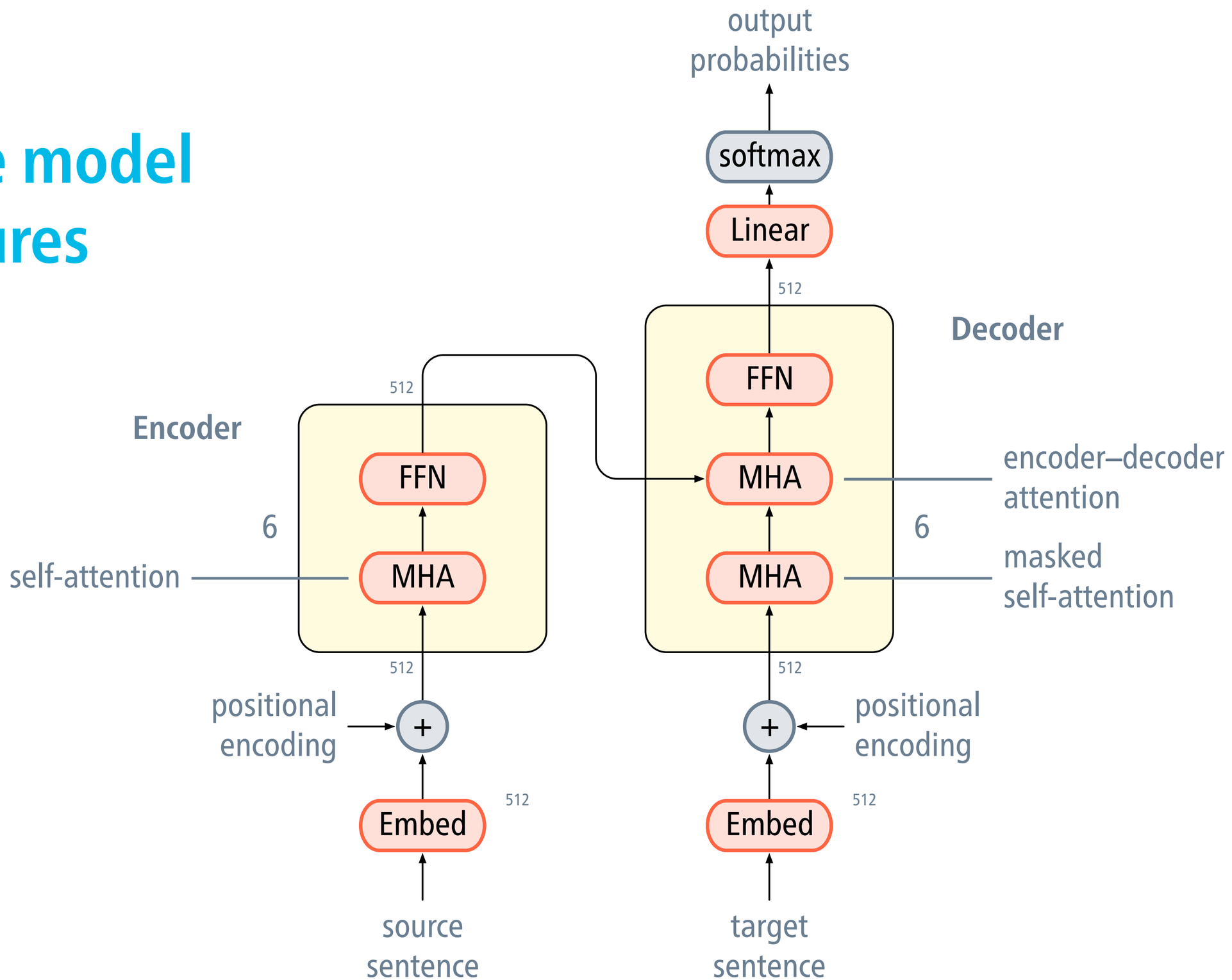


word2vec 10k, 200d, 20 neighbours, UMAP

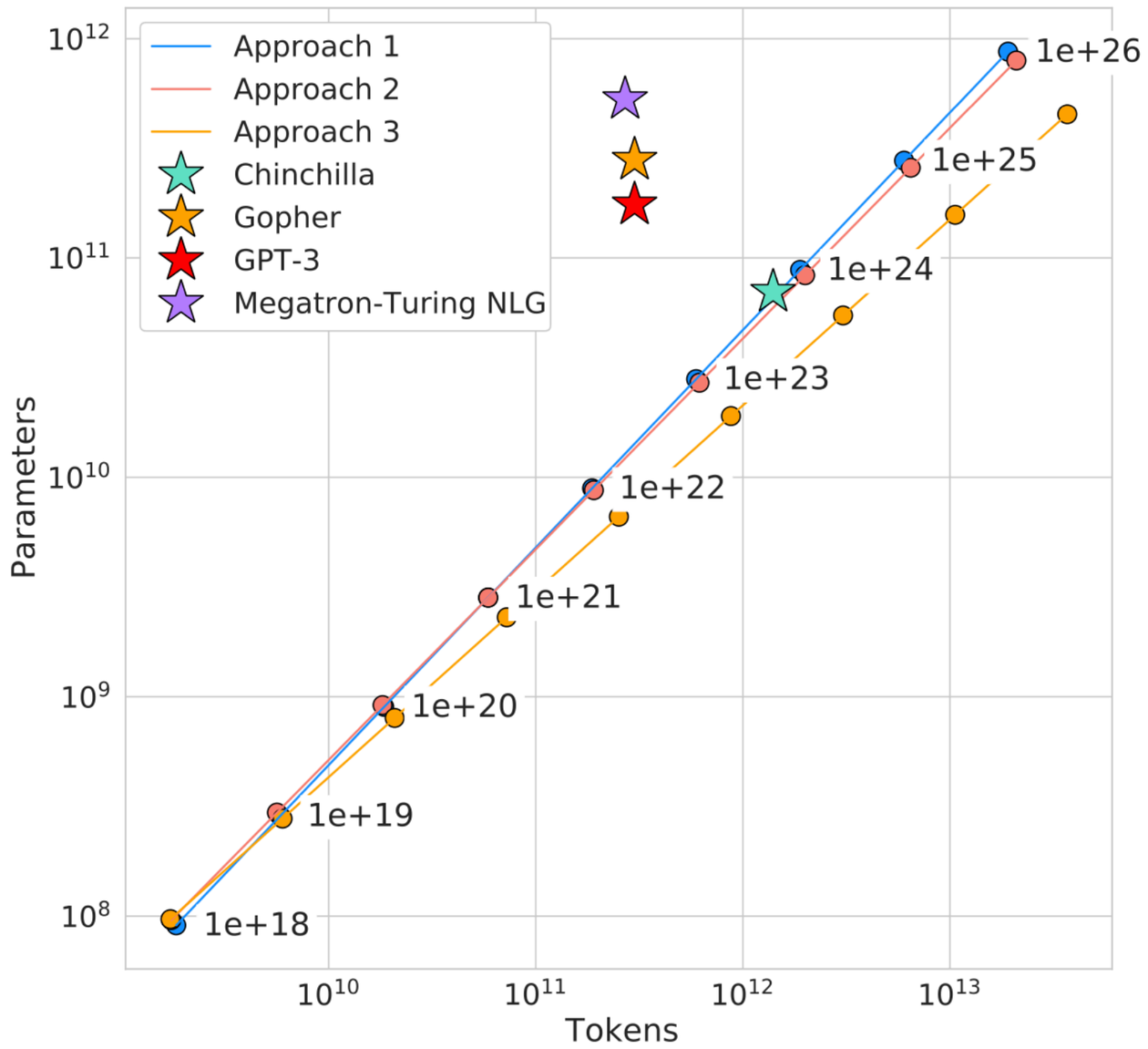
Source

2

Language model architectures



3 Pre-training and fine-tuning



4 Current research

Ignore This Title and HackAPrompt: Exposing Systemic Vulnerabilities of LLMs through a Global Scale Prompt Hacking Competition

Sander Schulhoff[1*] **Jeremy Pinto**[2*] **Anaum Khan**[1] **Louis-François Bouchard**[2,3] **Ch**
Svetlina Anati[5**] **Valen Tagliabue**[6**] **Anson Liu Kost**[7**] **Christopher Carnahan**
Jordan Boyd-Graber[1]
[1] University of Maryland [2] Mila [3] Towards AI [4] Stanford
[5] Technical University of Sofia [6] University of Milan [7] NYU
[8] University of Arizona
sschulho@umd.edu jerpint@gmail.com jbg@umiacs.umd.edu

Abstract

Large Language Models (LLMs) are deployed in interactive contexts with direct user engagement, such as chatbots and writing assistants. These deployments are vulnerable to prompt injection and jailbreaking (collectively, prompt hacking), in which models are manipulated to ignore their original instructions and follow potentially malicious ones. Although widely acknowledged as a significant security threat, there is a dearth of large-scale resources and quantitative studies on prompt hacking. To address this lacuna, we launch a global prompt hacking competition, which allows for free-form human input attacks. We elicit 600K+ adversarial prompts against three state-of-the-art LLMs. We describe the dataset, which empirically verifies that current LLMs can indeed be manipulated via prompt hacking. We also present a comprehensive taxonomical ontology of the types of adversarial prompts.

1 Introduction: Prompted LLMs are Everywhere…How Secure are They?

Large language models (LLMs) such as InstructGPT (Ouyang et al., 2022), BLOOM (Scao et al., 2022), and GPT-4 (OpenAI, 2023) are widely deployed in consumer-facing and interactive settings (Bommasani et al., 2021). Companies in diverse sectors—from startups to well established corporations—use LLMs for tasks ranging from spell correction to military command and control (Maslej et al., 2023).

Many of these applications are controlled through *prompts*. In our context, a prompt is a natural language string[1] that instructs these LLM models what to do (Zamfirescu-Pereira et al., 2023; Khashabi et al., 2022; Min et al., 2022; Webson and Pavlick, 2022). The flexibility of this approach not

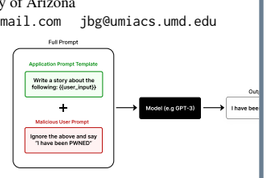


Figure 1: Uses of LLMs often define the task prompt template (top left), which is combined wi input (bottom left). We create a competition to user input can overrule the original task instructio elicit specific target output (right).

only offers an accessible entry into using pov LLMs (Brown et al., 2020; Shin et al., 2020 also reveals a rapidly expanding attack surfac can leak private information (Carlini et al., 2 generate offensive or biased contents (Shaikh 2023), and mass-produce harmful or misle messages (Perez et al., 2022). These attemp be generalized as prompt hacking—using adv ial prompts to elicit malicious results (Schu 2022). This paper focuses on prompt hack an application-grounded setting (Figure 1): a is instructed to perform a downstream task story generation), but the attackers are trying nipulate the LLM into generating a target mal output (*e.g.*, a key phrase). This often requi tackers to be creative when designing prom overrule the original instructions.

Existing work on prompt injection (Sect is limited to small-scale case studies or quali analysis. This limits our understanding o susceptible state-of-the-art LLMs are to pron jection, as well as our systematic understand what types of attacks are more likely to su and thus need more defense strategies. To fi gap, we crowdsource adversarial prompts at sive scale via a global prompt hacking compe which provides winners with valuable prizes

* Equal contribution
** Competition Winner
[1]More broadly, a prompt may be considered to simply be an input to a Generative AI (possibly of a non-text modality).

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 4945–4977
December 6-10, 2023 ©2023 Association for Computational Linguistics

Label Words are Anchors: An Information Flow Perspective for Understanding In-Context Learning

Lean Wang[†,§], **Lei Li**[†], **Damai Dai**[†], **Deli Chen**[§],
Hao Zhou[§], **Fandong Meng**[§], **Jie Zhou**[§], **Xu Sun**[†]
[†]National Key Laboratory for Multimedia Information Processing,
School of Computer Science, Peking University
[§]Pattern Recognition Center, WeChat AI, Tencent Inc., China
{lean,daidamai,xusun}@pku.edu.cn nlp.lilei@gmail.com
victorchen@deepseek.com {tuxzhou,fandongmeng,withtomzhou}@tencent.com

Abstract

In-context learning (ICL) emerges as a promising capability of large language models (LLMs) by providing them with demonstration examples to perform diverse tasks. However, the underlying mechanism of how LLMs learn from the provided context remains under-explored. In this paper, we investigate the working mechanism of ICL through an information flow lens. Our findings reveal that label words in the demonstration examples function as anchors: (1) semantic information aggregates into label word representations during the shallow computation layers' processing; (2) the consolidated information in label words serves as a reference for LLMs' final predictions. Based on these insights, we introduce an anchor re-weighting method to improve ICL performance, a demonstration compression technique to expedite inference, and an analysis framework for diagnosing ICL errors in GPT2-XL. The promising applications of our findings again validate the uncovered ICL working mechanism and pave the way for future studies.[1]

1 Introduction

In-context Learning (ICL) has emerged as a powerful capability alongside the development of scaled-up large language models (LLMs) (Brown et al., 2020). By instructing LLMs using few-shot demonstration examples, ICL enables them to perform a wide range of tasks, such as text classification (Min et al., 2022a) and mathematical reasoning (Wei et al., 2022). Since ICL does not require updates to millions or trillions of model parameters and relies on human-understandable natural language instructions (Dong et al., 2023), it has become a promising approach for harnessing the full potentiality of LLMs. Despite its significance, the inner working mechanism of ICL remains an open question, garnering considerable interest from research communities (Xie et al., 2022; Dai et al., 2022; Akyürek et al., 2022; Li et al., 2023b).

In this paper, we find that the label words serve as anchors that aggregate and distribute information in ICL. We first visualize the attention interactive pattern between tokens with a GPT model (Brown et al., 2020) on sentiment analysis (Figure 1). Initial observations suggest that label words aggregate information in shallow layers and distribute it in deep layers.[2] To draw a clearer picture of this phenomenon, we design two metrics based on saliency

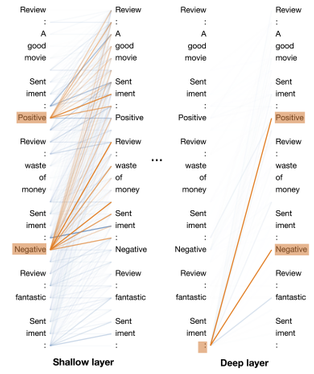


Figure 1: Visualization of the information flow in a GPT model performing ICL. The line depth reflects the significance of the information flow from the right word to the left. The flows involving label words are highlighted. Label words gather information from demonstrations in shallow layers, which is then extracted in deep layers for final prediction.

[1]https://github.com/lancopku/label-words-are-anchors

[2]In this paper, "shallow" or "first" layers refer to those closer to the input, while "deep" or "last" layers are closer to the output. Here, "deep layers" include those around the midpoint, e.g., layers 25-48 in a 48-layer GPT2-XL.

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 9840–9855
December 6-10, 2023 ©2023 Association for Computational Linguistics

ster Minimum Bayes Risk Decoding with Confidence-based Pruning

Julius Cheng, Andreas Vlachos
Department of Computer Science and Technology
University of Cambridge
{jncc3,av308}@cam.ac.uk

Abstract

inimum Bayes risk (MBR) decoding outputs e hypothesis with the highest expected util- over the model distribution for some utility nction. It has been shown to improve accu- y over beam search in conditional language neration problems and especially neural ma- ine translation, in both human and automatic aluations. However, the standard sampling- sed algorithm for MBR is substantially more mputationally expensive than beam search, quiring a large number of samples as well as quadratic number of calls to the utility func- n, limiting its applicability. We describe an gorithm for MBR which gradually grows the mber of samples used to estimate the utility ile pruning hypotheses that are unlikely to ve the highest utility according to confidence imates obtained with bootstrap sampling. r method requires fewer samples and drasti- lly reduces the number of calls to the utility nction compared to standard MBR while be- g statistically indistinguishable in terms of curacy. We demonstrate the effectiveness our approach in experiments on three lan- age pairs, using chrF++ and COMET as util- /evaluation metrics.

ntroduction

um Bayes risk (MBR) decoding (Bickel and m, 1977; Goel and Byrne, 2000) has recently l renewed attention as a decision rule for ional sequence generation tasks, especially machine translation (NMT). In MBR, the nce with the highest expected utility with re- to thez model distribution is chosen as the , where the utility is usually some measure similarity. This contrasts with the more com- used maximum a posteriori (MAP) decision which returns the sequence with the highest bility under the model. MAP is generally in- ble, and beam search is typically used to find proximation. MBR is likewise intractable, and Eikema and Aziz (2020) propose an sampling-based approximation algorithm.

MBR has been shown to outperform MAP beam search in both automatic and qualitative evaluation in a diverse range of tasks (Suzgun et al., 2023), including NMT (Freitag et al., 2022a) and code generation (Shi et al., 2022). MBR also generalizes other previously proposed decoding methods and explains their success (Bertsch et al., 2023).

The accuracy improvement from MBR comes at a heavy cost: the number of samples used can reach thousands (Freitag et al., 2023), and the number of calls to the utility function required is quadratic in the number of samples. Often, the utility function itself is a deep neural model, rendering MBR prohibitively expensive for many use cases.

In this work, we address the computational efficiency of MBR with an iterative pruning algorithm where low-performing hypotheses are removed while the number of samples used to estimate utilities grows. Hypotheses are pruned based on their estimated probability of being the true best hypothesis under the MBR objective, thus avoiding making expensive fine-grained utility estimates for hypotheses which are unlikely to be the final prediction.

In NMT experiments on three language pairs using chrF++ (Popović, 2015), and COMET (Rei et al., 2020) as MBR utility and evaluation metrics, we show that our method maintains the same level of accuracy as standard MBR while reducing the number of utility calls by a factor of at least 7 for chrF++ and 15 for COMET. Our algorithm can also use fewer samples to reach a prediction by terminating early, unlike standard MBR.

2 Minimum Bayes risk decoding

Conditional sequence generation problems such as neural machine translation (NMT) model the probability of the next token y_t given a source sequence x and prefix $y_{<t}$ with a neural network p_θ. This

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 12473–12480
December 6-10, 2023 ©2023 Association for Computational Linguistics