Knowledge Capsule II

Application of LLMs in Transportation:

From Transformers to human mobility prediction

Mini Lecture for Data, AI, and Multimodal Traffic Management Course

Weiming Mai | w.m.mai@tudelft.nl

Mahsa Movaghar | m.Movaghar@tudelft.nl

Friday, October 3rd, 2025

Image created using OpenAI's DALL·E (2024)

Session goals

By the end of this session, you should be able to learn and answer:

01

Introduction to LLM

- What are LLMs?
- What are the main components of an LLM model that distinguish it from other models?
- What are the challenges and concerns using LLMs?

02

Application of LLMs in Transportation

- How can you use LLMs in the Transportation domain?
- Why and how to use Transformers for predicting trajectories?
- From Transformers to Mobility prediction: Hands-on practice on using Transformers for trajectory prediction.

01

Introduction to LLM

- What are LLMs?
- What are the main components of an LLM model that distinguish it from other models?
- What are the challenges and concerns using LLMs?

What are LLMs?

Do not panic!!

- Nothing really new!

- Autocomplete or autocorrection feature on your mobile phone was a little cousins of GPT! So, it has been a while since we used them!

- They are called ***Large*** because they are trained on massive datasets, and they have billions of parameters!

A brief history of LLMs

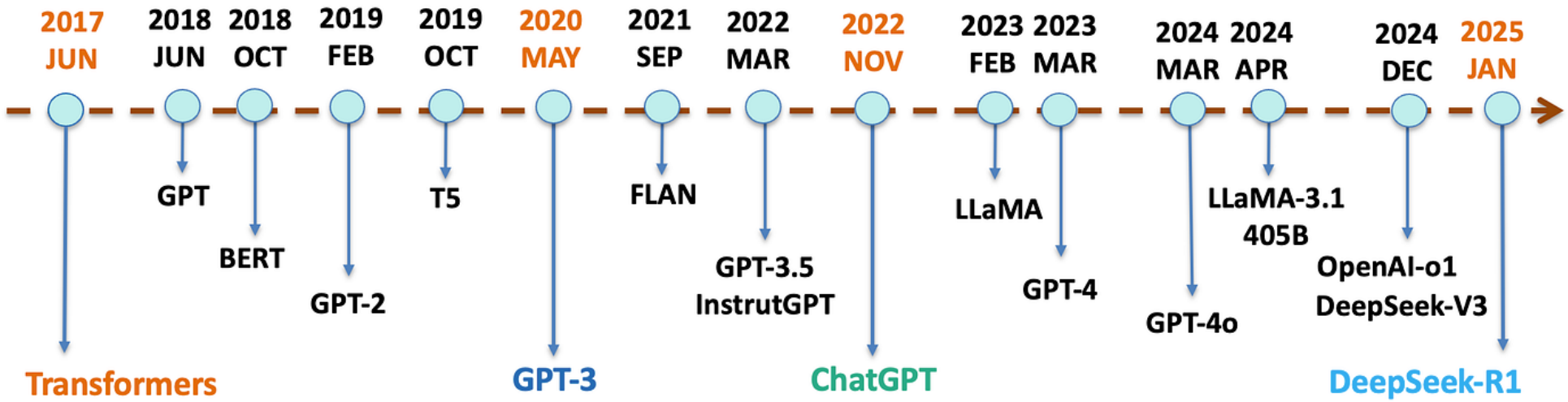


Image Source:
LLMs and Real-World Applications (Link)

LLMs VS. Deep Learning and Generative AI

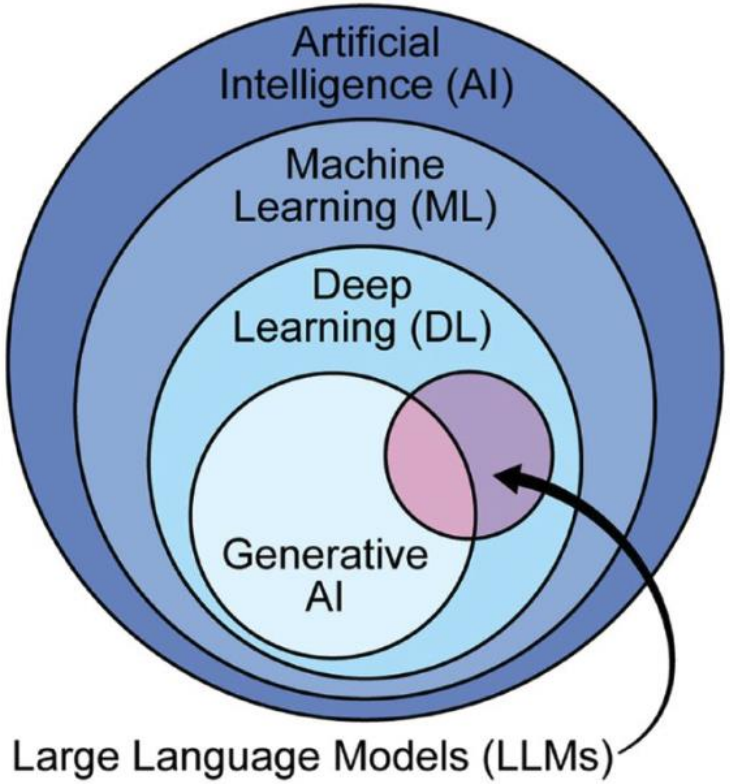
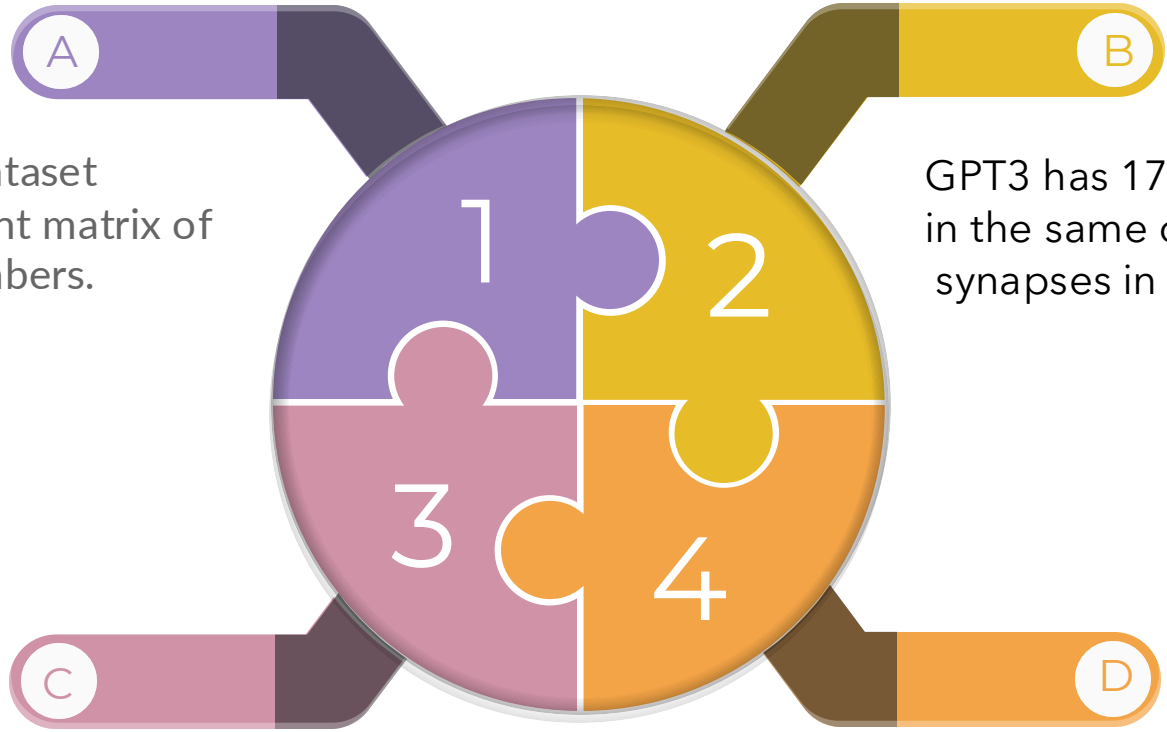


Image Sources:
Large language models: a primer and gastroenterology applications (DOI: 10.1177/17562848241227031)

What are LLMs?

Advanced AI systems built on **deep neural networks** that can process, analyze, and generate human-like text.



Tokenization and Embeddings

Massive textual dataset converted to a giant matrix of floating-point numbers.

Billions of parameters

GPT3 has 175 billion parameters in the same order as the number of synapses in a mouse brain.

Attention Mechanism

They have a mechanism to find and direct to the most important part of the input data.

Training and Resources

Training cost is mind-blowing 💰
Training GPT4 (around 13 trillion training tokens) is estimated at over 100 million $.

The training took approximately 100 days, consuming around 50GWH for a single training run.

Architecture of LLMs

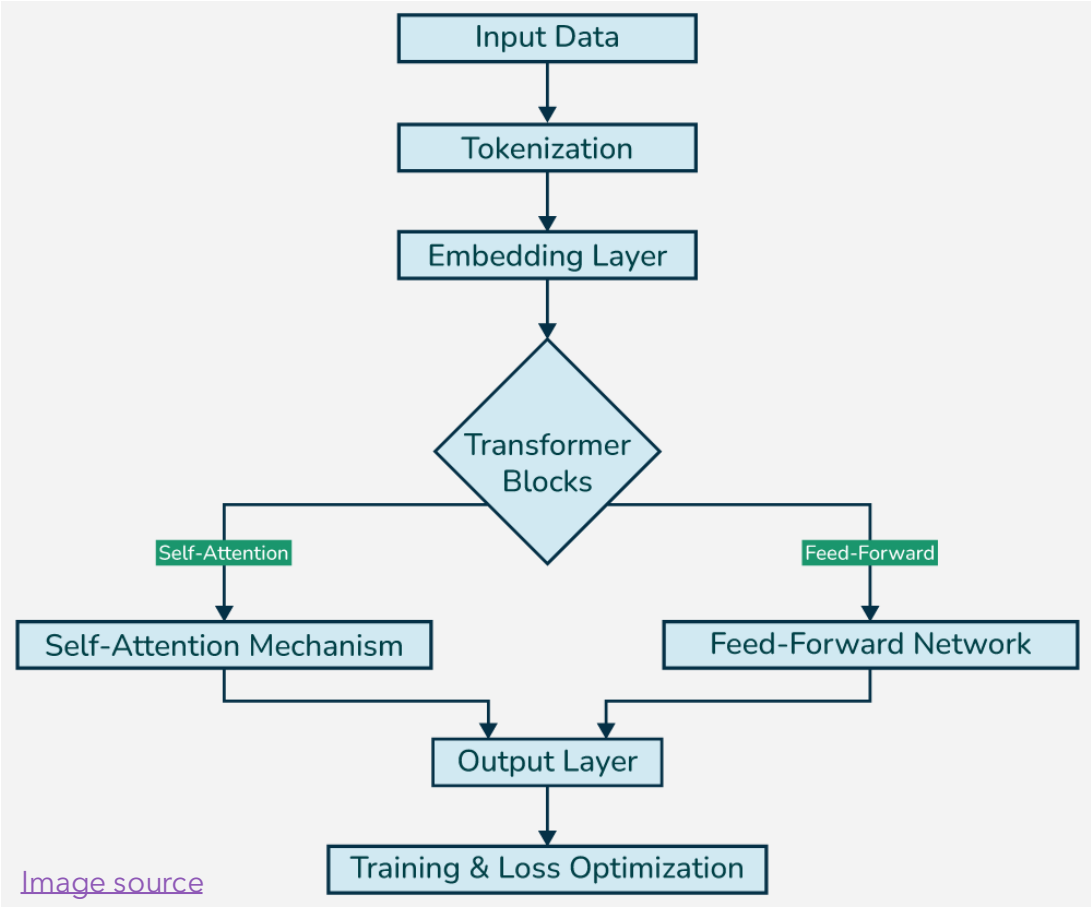


Image source

Architecture of LLMs: Tokenization

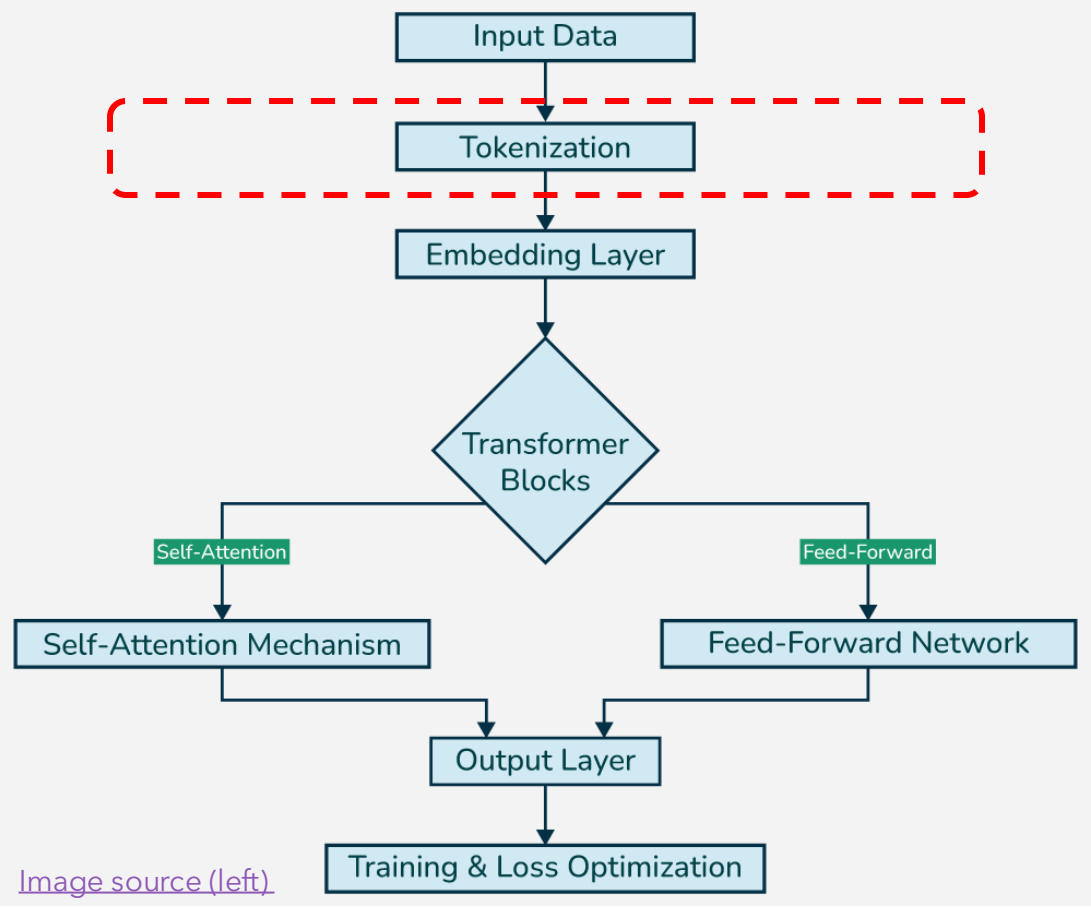


Image source (left)

This is a TRAIL course for AI in Traffic.

"This"
"is"
"a"
"TRAIL"
"course"
"for"
"AI"
"in"
"Traffic"

Architecture of LLMs: Embedding

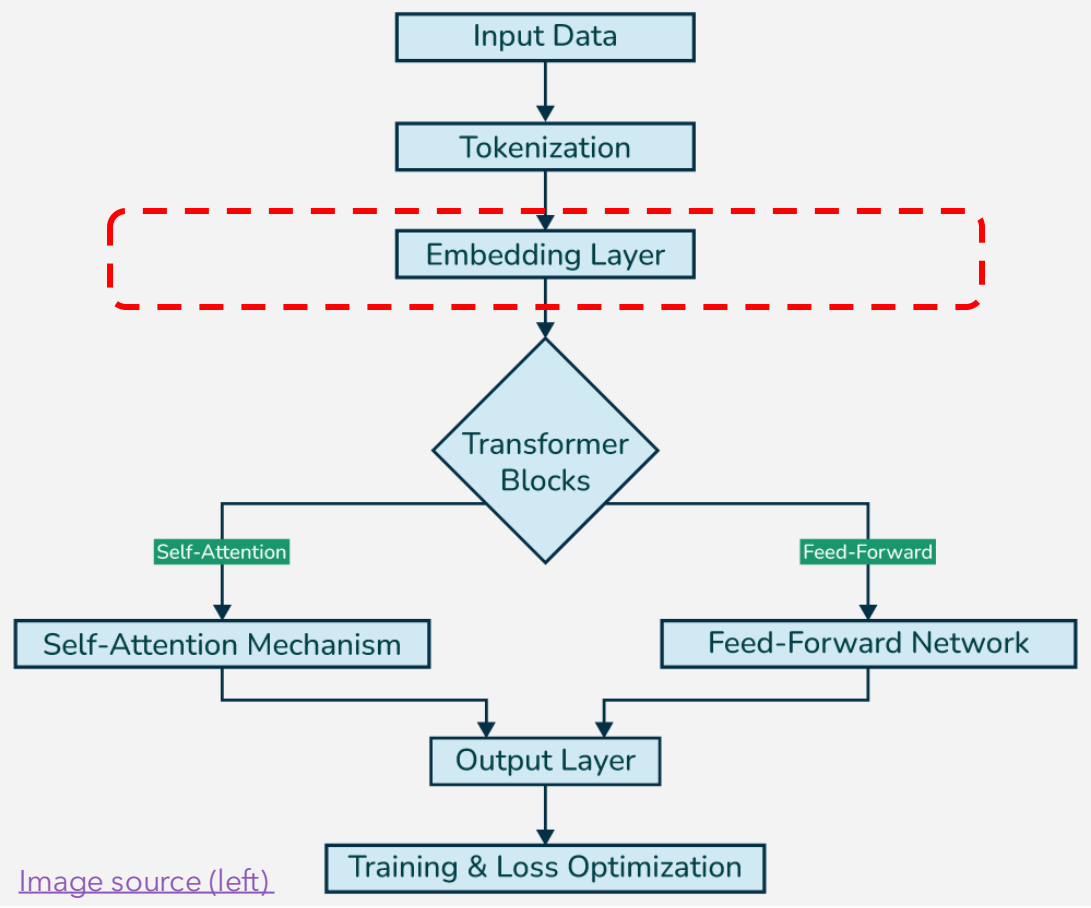


Image source (left)

This is a TRAIL course for AI in Traffic.

"This"	0.86	0.75	…	0.45
"is"	0.35	0.56	…	0.22
"a"	0.26	0.61	…	0.91
"TRAIL"				
"course"				
"for"				
"AI"				
"in"	0.86	0.75	…	0.45
"Traffic"	0.43	0.65	…	0.82

Architecture of LLMs: Transformer

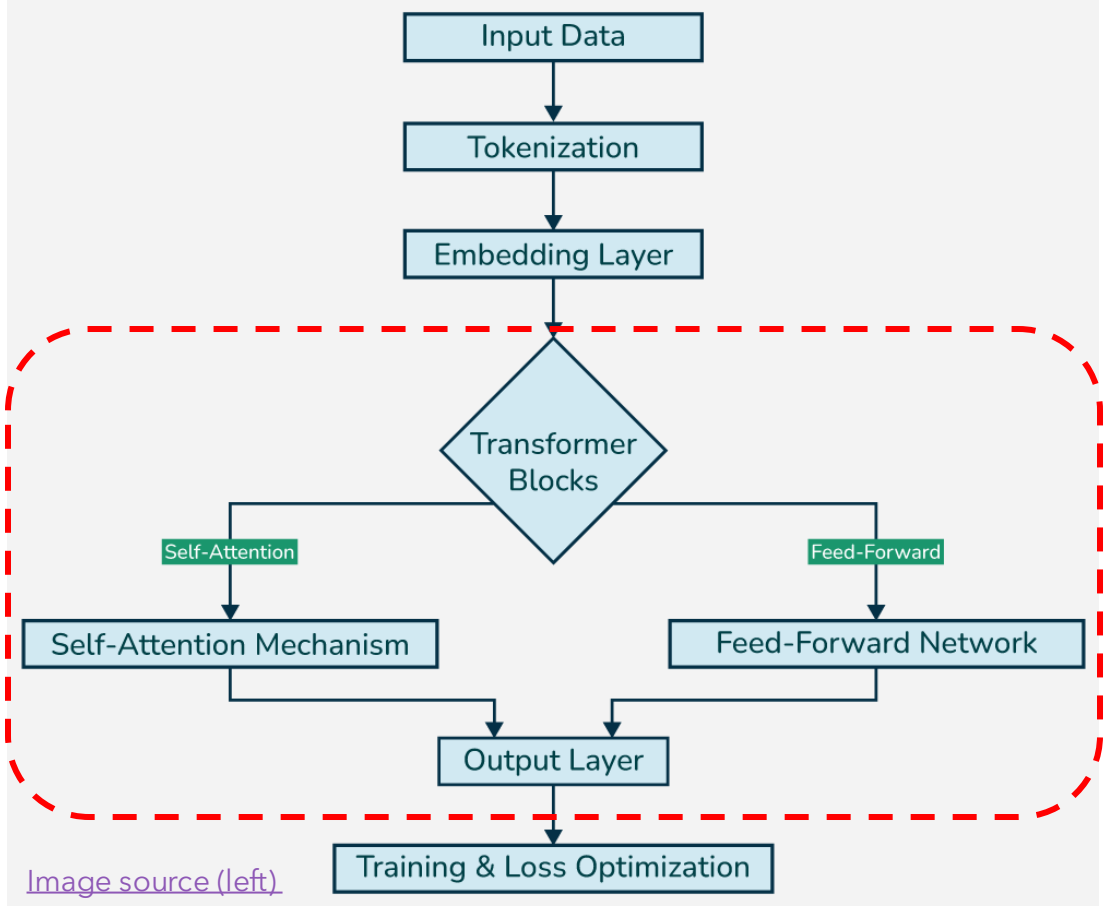


Image source (left)

Architecture of LLMs: Transformer

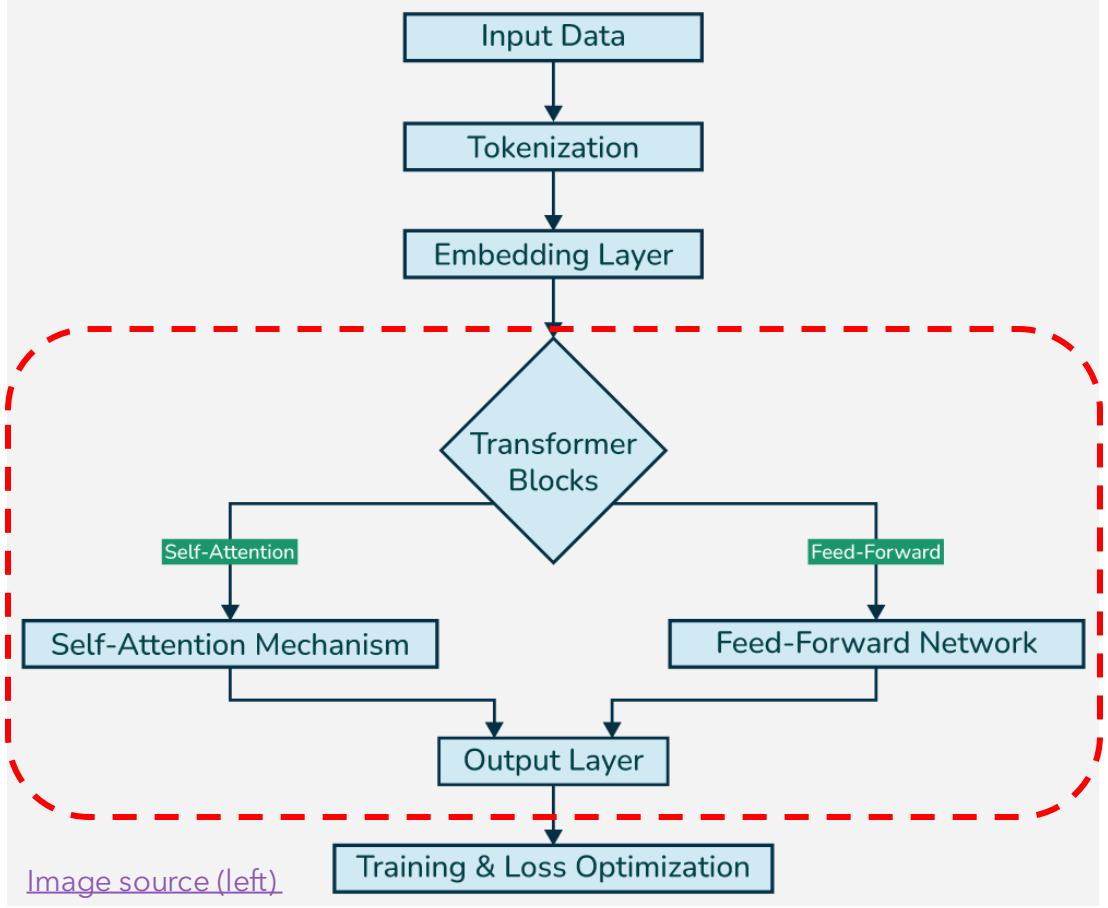
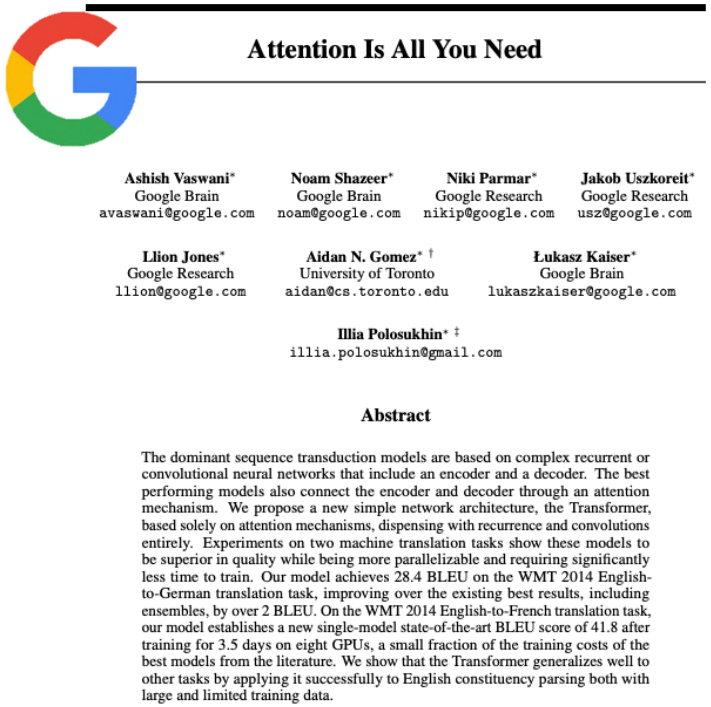


Image source (left)



Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Source: 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA. (link)

- A transformer is a Layer for implementing an attention mechanism, which can "**transform"** input sequences into meaningful outputs by "**focusing attention"** on the right word.
- Attention mechanism (Transformers) for LLMs is like Waze for words → it constantly re-routes focus to the most relevant words in a sentence, just like Waze that redirects drivers around traffic jams.

Architecture of LLMs: Transformer

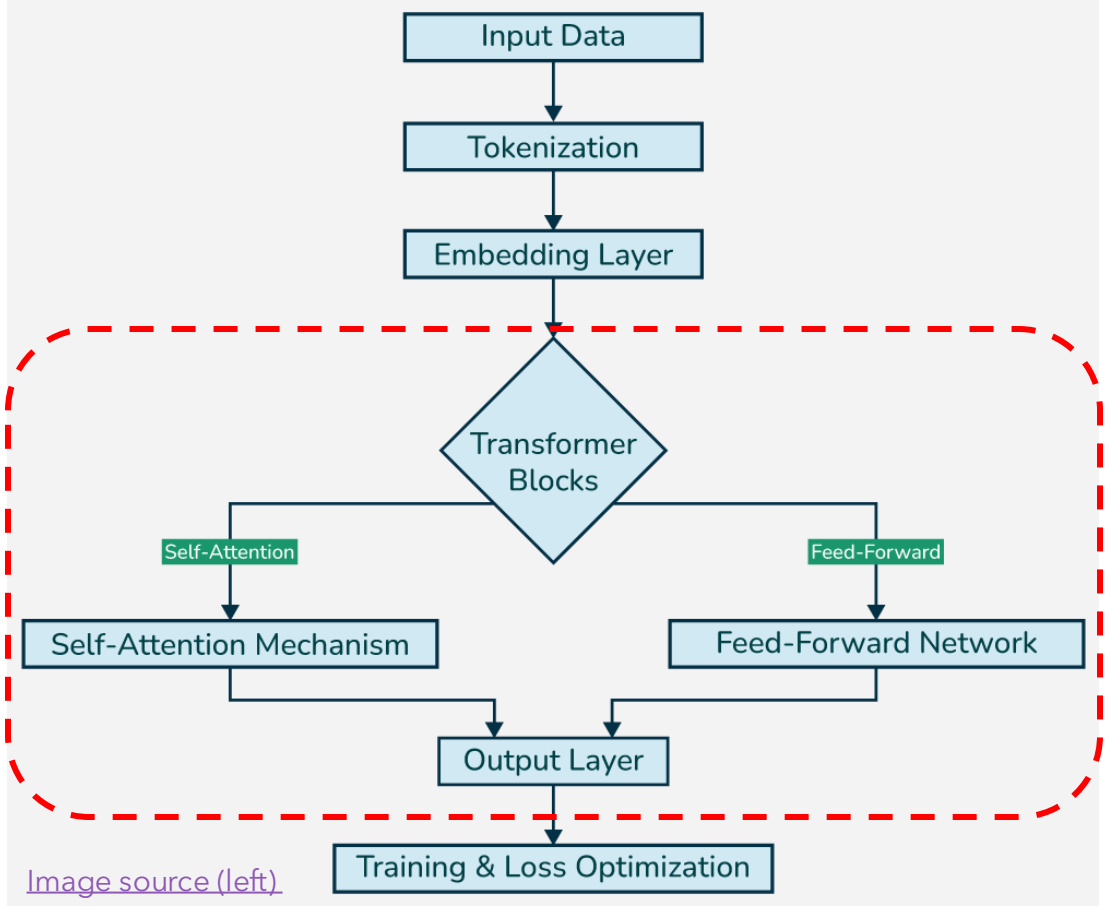
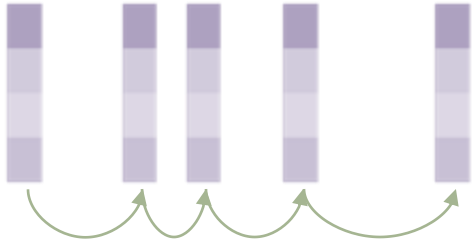


Image source (left)

Previous models

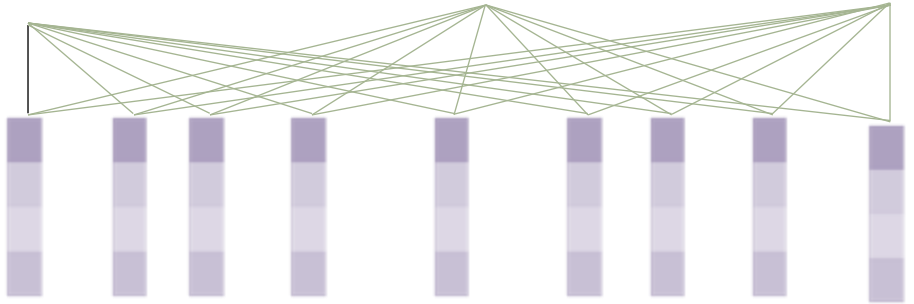
This is a TRAIL course for AI in Traffic.



...

Transformers

This is a TRAIL course for AI in Traffic.



Architecture of LLMs: Transformer

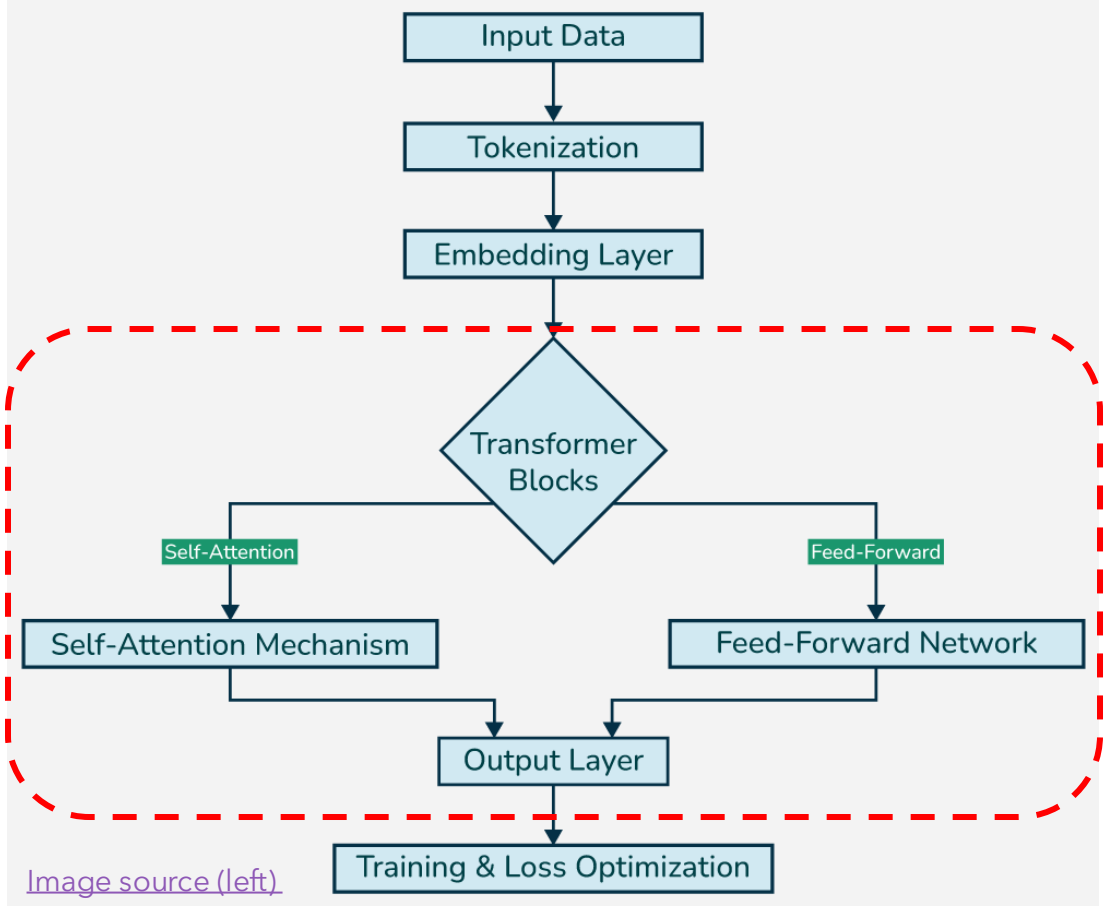
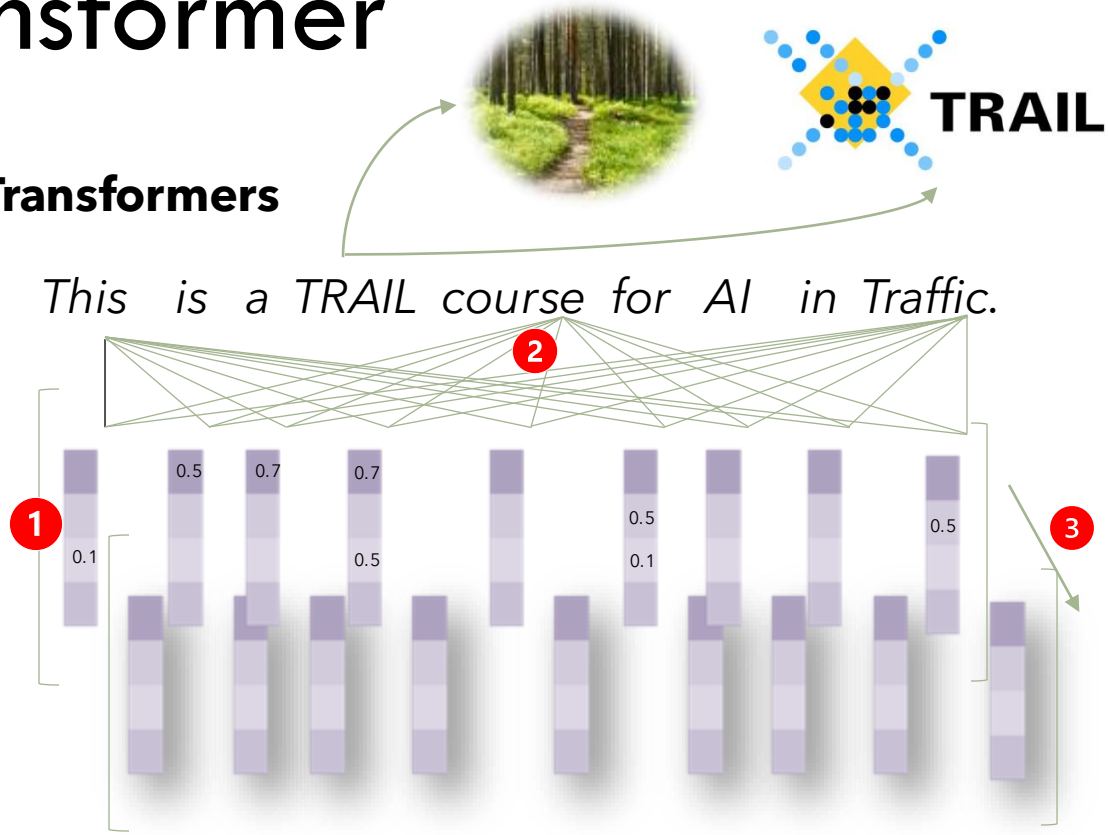


Image source (left)

Transformers



1. Transformers do not read text from the start to the finish, they soak it all in at once, in parallel.
2. Attention operation: Gives a chance that all numbers listed talk to one another. → Refining the meaning they encode based on the context around. → The combination of numbers will change to encode the more specific notion of the rest of the sentence.
3. Feed-forward capacity.

Architecture of LLMs: Transformer

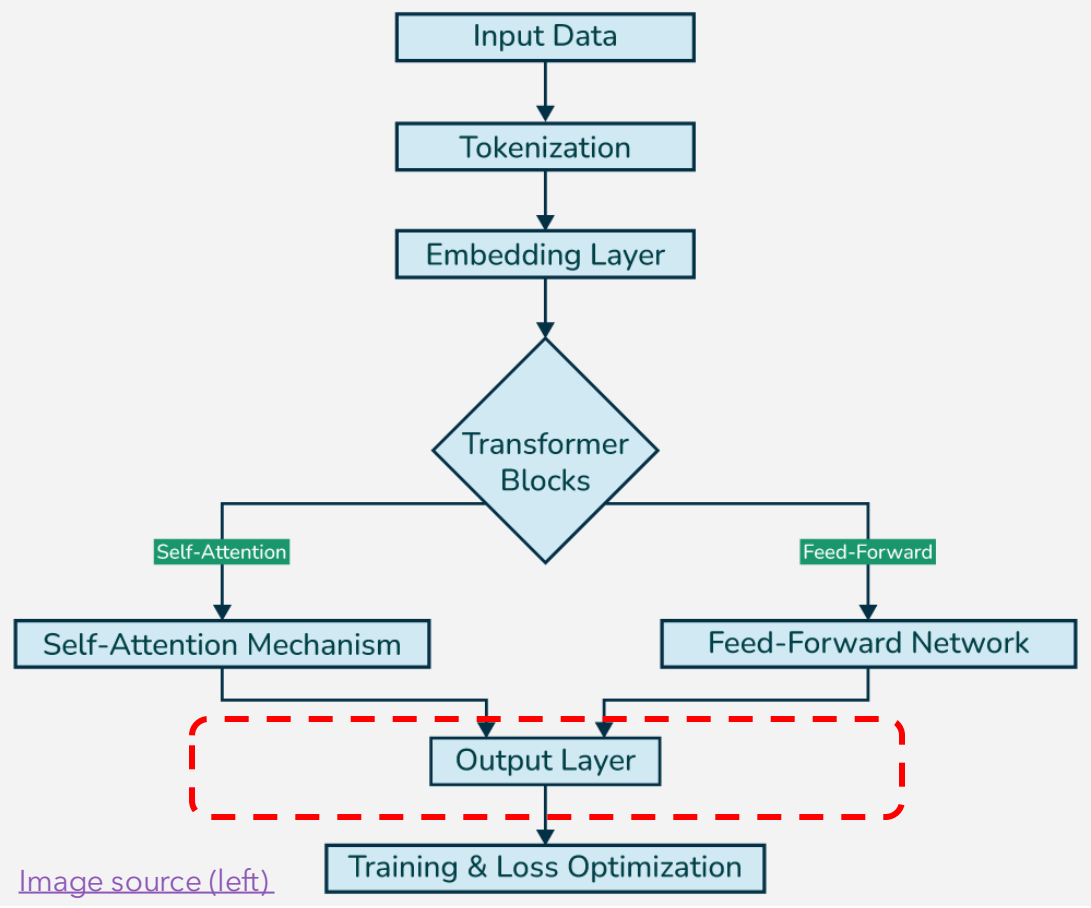
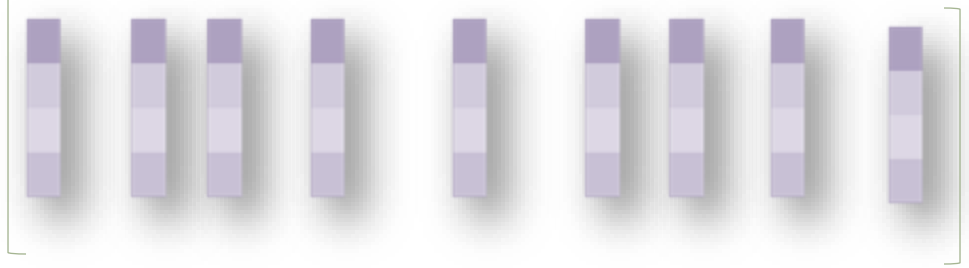


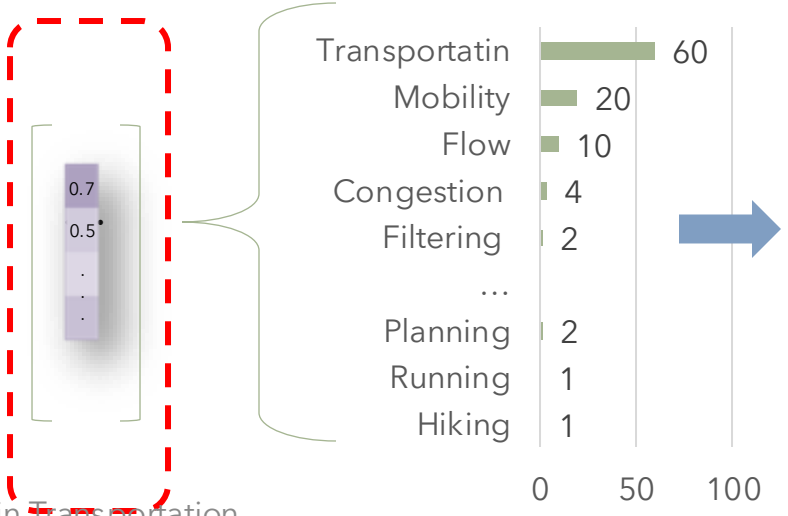
Image source (left)

Prediction of the next word, which is like a probability for every possible next word.

This is a TRAIL course for AI in Traffic.



*In this course we learn to use AI in **[???]***

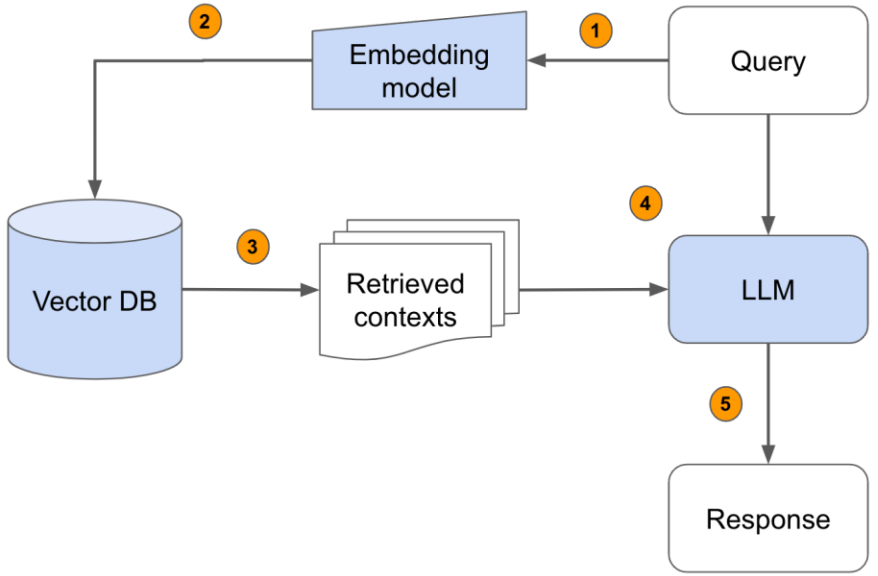


- **Sampling methods:**
- Deterministic:
  - Greedy
  - Beam Search
- Probabilistic
  - Top-k Sampling

New add-on for LLMs: RAG structure

Challenge with the previous models:

- **Without RAG:** LLM only uses what it memorized during training. → Like a driver who is driving relying only on his memory! 🚗
- **With RAG:** LLM can also use fresh and specific external data.→ Like a driver who is driving using GPS & live traffic updates!



RAG structure in LLMs (image source)

- **Retrieval:** When a user asks a question RAG system searches a separate content store to query some information related to the question.
- **Augmented:** That retrieved information will then be added to the initial query of the user and make the original prompt augmented.
- **Generation:** LLM then processes the augmented prompt using its training data and the newly provided context.

New add-on for LLMs: RAG structure

- Does this solve the bias and hallucination challenge with LLMs?
  - ✅ It will be reduced,
  - ✅ Outputs will be factual,

  But
  - ❌ Hallucination still happens → if retrieval pulls in irrelevant or low-quality documents.
  - ❌ Bias in generation still remains → the LLM still has its own **training biases** (language stereotypes, skewed distributions, …).
  - ❌ Bias in retrieval remains → If the external knowledge source is biased, RAG *inherits* that bias.

- In summary, RAGs are behaving more like a patch than an actual cure, so you still need to:
  - 📈 Be careful about the ***quality*** of retrieved data.
  - ⚖️ Consider ***bias detection and mitigation*** strategies.
  - 🙋 Have a ***human judgement and supervision*** for critical domains such as safety.

  - And,

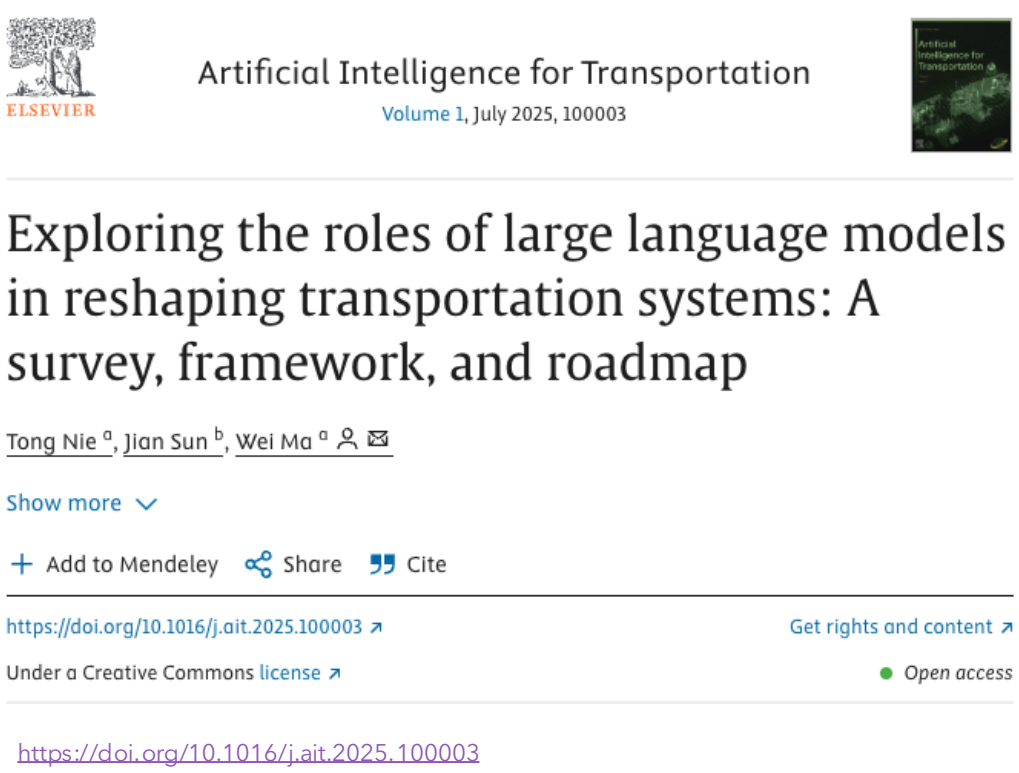
  - 💡 You can also think of creating a ***knowledge graph*** for your domain!

02

Application of LLMs in Transportation

- How can you use LLMs in the Transportation domain?
- Why and how to use Transformers for predicting trajectories?
- From Transformers to Mobility prediction: Hands-on practice on using Transformers for trajectory prediction.

Applications of LLMs in Transportation



ELSEVIER

Artificial Intelligence for Transportation

Volume 1, July 2025, 100003

Exploring the roles of large language models in reshaping transportation systems: A survey, framework, and roadmap

Tong Nie [a], Jian Sun [b], Wei Ma [a]

Show more

+ Add to Mendeley Share Cite

https://doi.org/10.1016/j.ait.2025.100003 ↗ Get rights and content ↗

Under a Creative Commons license ↗ Open access

https://doi.org/10.1016/j.ait.2025.100003

nm magazine

Hét vakblad voor netwerkmanagement in verkeer en vervoer.

HOME | ARTIKELEN | NIEUWS | AGENDA | DOWNLOAD | PARTNERS | OVER NM MAGAZINE | CONTACT

ZOEKEN ...

HOME › 2025 #2 › Large Language Models – de toekomst van mobiliteit?

INFORMATIE

Large Language Models – de toekomst van mobiliteit?

Large Language Models **zijn AI-systemen die menselijke taal begrijpen en zich er ook in kunnen uiten. Ze zijn de basis onder populaire applicaties als ChatGPT, Gemini en Copilot. Maar inmiddels is de technologie zó breed inzetbaar dat ze ook doordringt in de mobiliteitssector. Hoe werken de** *Large Language Models*? **Hoe kunnen ze van nut zijn in ons vakgebied? En wat zijn de mitsen en maren?**

De auteurs

Ting Gao, Mahsa Movaghar, Theivaprakasham Hari en Alex Roocroft zijn onderzoekers van het DAIMOND Lab van TU Delft. Zij worden begeleid door dr. ir. Marco Rinaldi en dr. Yanan Xin, co-directeurs van het Lab, en prof. dr. ir. Serge Hoogendoorn, adviseur van het Lab.

Read more online

Applications of LLMs in Transportation

1. **Traffic Signal Optimization (TSC):** Frameworks like **LLMLight** employ GPT-family models as intuitive decision makers for optimizing traffic lights, mimicking the contextual reasoning of human operators.

2. **Scenario Synthesis: LLMScenario** uses LLMs to generate parameters for safety-critical traffic scenarios, especially rare corner cases, by translating textual descriptions into actionable simulation parameters.

3. **Geospatial Mobility Modeling: MobilityGPT** is a geospatially-aware generative model that reformulates human mobility modeling as an autoregressive generation task using the GPT architecture, capturing complex dependencies in movement.

4. **Multimodal Demand Prediction:** LLMs are integrated to fuse heterogeneous data (e.g., GPS trajectories, social media, event data) for applications like electric vehicle charging demand prediction (**ChatEV**) or taxi usage forecasts.

5. **Crash Safety Analysis: CrashLLM** fine-tunes LLMs to predict fine-grained accident outcomes (e.g., severity, injury numbers) by framing crash event feature learning as a text reasoning problem, and LLMs are used to identify underreported alcohol involvement in crash narratives.

6. And much more there and will be …

From Transformers to Mobility Prediction

Why predict the trajectory of pedestrians?

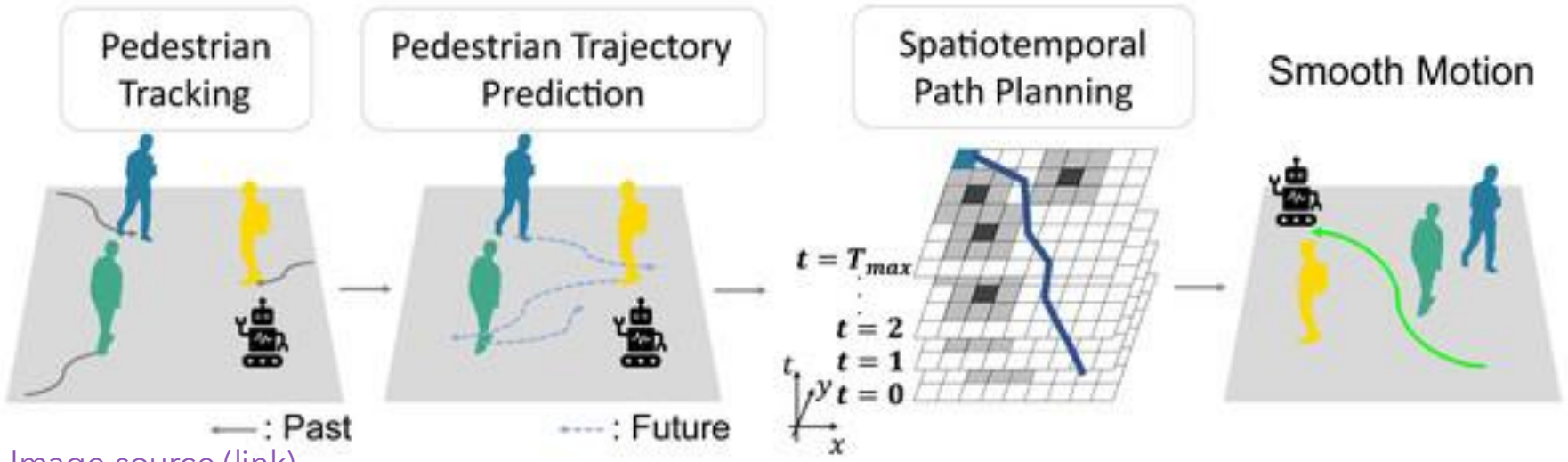


Image source (link)

From Transformers to Mobility Prediction

Why transformers for prediction?

- Trajectory prediction is a **sequence-to-sequence problem** → Based on previous observations, we want to predict next step.
- Trajectories have **short-term and long-term patterns** → Self-attention is what we want :)! → So, we can look back at any point in history not just the most recent ones.
- Trajectories have **spatiotemporal features.** → Transformers are capable of handling multi-modal tokens.
- Trajectories are **huge** datasets. → Transformers process all tokens in parallel. → Faster than a sequential RNN.
- Trajectories are **stochastic phenomena** rather than deterministic.

- Do you see any advantage of using Transformers instead of traditional models like LSTM, RNN,..?

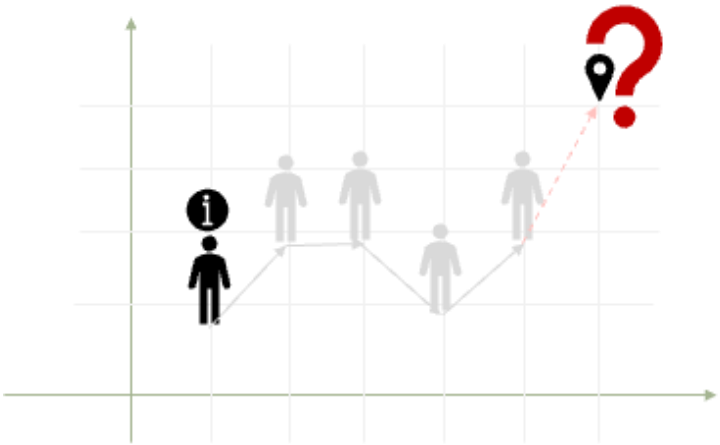
From Transformers to Mobility Prediction

How to use transformers for trajectory prediction?

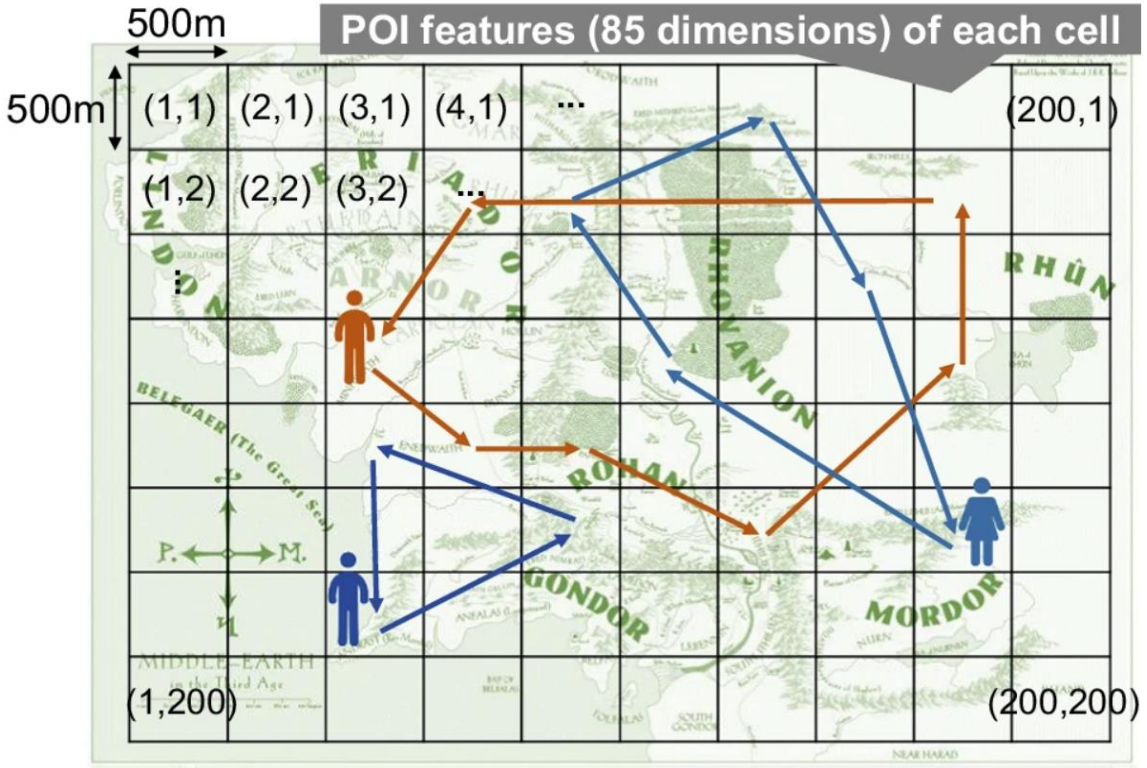
HINT

- Locations → words
- Trajectories → sentences
- Predicting next move → predicting next words.

1. Clean and split the dataset into past sequence (input) and future sequences (target).
2. Tokenization → turn trajectories into language.
   - Convert coordinates into discrete tokens: (45,120) → (x_45,y_120)
   - Add temporal features: time slot of 10 → t_10, Monday → dow_1
   - Add special tokens: [START],[END],[PAD]
   - Now each trajectory looks like a sentence.
3. Encoder input → past trajectory
4. Decoder input → time/day tokens for the steps we want to predict.
5. Decode output → the sequence of future coordinates.



Human Mobility Prediction



ACM SIGSPATIAL GIS Cup 2025

- Each area is divided into 500 meters x 500 meters cells, which span a 200 x 200 grid.
- The human mobility datasets contain the movement of individuals across a 75-day period, discretized into 30-minute intervals.

Hands-on practice

Takeaways

1. LLMs refer to large, general-purpose language models that can be **pre-trained** and then **fine-tuned** for specific purposes.

2. Through **embedding/tokenization** process, LLMs compress all the input data into a giant matrix of floating-point numbers.

3. LLMs, thanks to the **transformers**, process all words at once, while humans read sequentially, and shift attention to more important tokens.

4. They do not memorize texts, otherwise they learn **probabilistic** patterns of languages.

5. Transformers are a good approach to predict trajectories because traditional models like RNN/LSTM can not …. **handle longer sequential data/ capturing long-term patterns.**

Nudges

1. 😱 They sometimes hallucinate words, citations, and so on. They ***generate words*** that do not really exist! → Like when Google Maps directs you to a fietspad that does not exist! *You should not use them to drive blindly!*

2. 💡 They do code, but they do not understand the ***reasons and needs*** behind the software!

3. 🦜 Like parrots, they ***mimic patterns***, but they don't understand.

4. 🚧 ***Training*** part of LLM models is the most cost-driven part. Once trained, it's easy to run, like driving on a built road.

5. 🎛️ Don't forget! Although they are doing great, they are still doing text prediction; they ***do not have internal logic***!

6. 📈 They ***can reflect bias***! If the training data has a stereotype, the model will learn them! Exactly like you have a traffic model that overestimates cars and underestimates cyclists!