LLM Glossary
(draft version)

February 28, 2024

Tessa Gengnagel, Fotis Jannidis, Rabea Kleymann, Julian Schröter, Heike Zinsmeister

Embedding is a representation of units such as words, sentences, or texts, which embed these units into a high-dimensional vector space. An embedding comprises a real-valued vector that can be interpreted as a point within this space. Embeddings are used to calculate distributional similarity and can be pre-trained on corpora using unsupervised large language models.

Distributional Hypothesis The Distributional Hypothesis states that words that occur in the same contexts tend to have similar meanings (Harris, 1954, p. 151-57). It has been operationalized in computational semantics as "words with similar distributional properties have similar meanings" (Sahlgren, 2008, p. 21). Distributional properties are understood as the sets of co-occurring words in a corpus (plus their frequencies), e.g. such that the words occur in a given window of n surrounding tokens of the target word, or occur in a syntactic dependency relation with the target word. In a strong interpretation of the Distributional Hypothesis, humans derive the meaning of words from distributional patterns (Lenci, 2008). The distributional hypothesis provides the theoretical basis for representing meaning geometrically in vector spaces.

Generative AI can be defined as a technology that leverages → neural networks, also known as deep learning models, to generate human-like content (e.g., images, texts) in response to complex and varied prompts (e.g., languages, instructions, questions). In recent years they are usually based on the → Transformer architecture.

Grounding refers to the process of providing LLMs use-case specific information alongside prompts in order to improve the accuracy and quality of response. This is necessitated by LLMs acting like engines rather than knowledge stores. The most common technique for grounding is Retrieval Augmented Generation (RAG) which provides additional context through information retrieval. This typically involves semantic search (which relies on an embedding model, vector index, and similarity search). More broadly speaking, grounding may also refer to the situatedness of communication in space, time, and environment (as well as embodiment, being grounded in bodily experiences) which informs the context and thereby meaning of communication. The lack of such a situatedness of cognition that characterizes LLMs is one of the root causes for the described mitigation strategies and still one of the major points of contention.

Hallucination describes a phenomenon, primarily in the context of chatbots, where fact-seeking prompts return replies containing non-sensical or fictitious statements masquerading as fact. The generated responses retain a simulation of plausibility without corresponding to actual information present in the source material they were trained on. This may be caused by a variety of reasons, including lack of grounding, overfitting and underfitting, source-reference divergence (both intentional and unintentional), and noisy training data, among others; most importantly, LLMs are language models rather than models of the world and have no awareness of factuality as such, generating words on the basis of probability rather than accuracy. The entirety and extent of the phenomenon are not yet fully understood. For users, it is not always possible to easily discern between fact and fiction, unless they possess the required domain knowledge themselves. Different solutions have been proposed to mitigate the issue, such as validating responses against information modelled in knowledge graphs.

Inference In the context of deep learning, the term 'inference' refers to the forward pass in a neural network and the resulting prediction, for example that a given email is spam. In cognitive science and philosophy inference refers to the process of reaching a conclusion based on reasoning on the basis of information. Very often three types of inference are distinguished: deduction, induction and abduction. Deduction refers to a reasoning which is always true: 1) all humans are mortal, 2) Sokrates is a human, 3) therefore Sokrates is mortal. In induction the reasoning starts with an observation 1) Socrates is human, 2) Socrates is mortal, 3) therefore all humans are mortal. In contrast to induction abduction aims the best explanation of some fact using a general rule: 1) Socrates is mortal, 2) all humans are mortal, 3) therefore Socrates is a human. In contrast

to deduction, which always provides true results, the results of induction and abduction are only probable (Douven, 2011).

Interpretation serves as an umbrella term for an attribution of meaning to entities (e.g., aesthetic works, actions, natural phenomena, etc.).

Knowledge Graph refers to a networked representation that delineates real-world objects and concepts into entities and models their relation to each other in a graph database. Structurally, this requires nodes connected by edges; semantically, it requires the labelling of components. The relationships in such a graph are commonly expressed in semantic triples, i.e. subject-predicate-object statements. A well-known example for a knowledge graph is Wikidata which "acts as central storage for the structured data" of its sister projects like Wikipedia. The notion of such webs of networked knowledge are closely related to principles and practices of Linked Open Data (LOD). Since knowledge graphs always implicitly endorse a certain model of the world inherent to their expressions (as conceptualized by humans), they are subject to biases, inaccuracies and lacunae.

Large Language Model (LLMs) are → neural networks trained on large textual corpora. Unlike traditional language models focused on specific tasks (e.g., text classification or summarization), LLMs are notable for their adaptability across a broader spectrum of linguistic applications. This adaptability is enabled by two core characteristics: → Transformer Architectures and self-supervised training on very large datasets. The training data often encompasses hundreds of billions of words sourced from diverse text genres. This enormous scale gives LLMs substantial statistical knowledge about language patterns and relationships. The combination of transformer architectures and very large training datasets leads to several key functional capabilities in LLMs: Text Generation and Natural Language Understanding. LLMs exhibit a remarkable ability to generate coherent and fluent text. This generative capability spans various forms, including the completion of unfinished text, the creation of summaries, and even the production of different creative text formats (e.g., poems, code, scripts). LLMs demonstrate also a degree of language comprehension. This understanding enables them to tackle tasks like question answering (factual and open-ended), text classification, and sentiment analysis.

LLMs come in 'pre-trained' form. While this pre-training provides a foundational linguistic understanding, they also possess two key methods for greater task-specificity: Fine-Tuning and In-Context Learning. Fine-Tuning involves further training the LLM on a targeted dataset aligned with a particular task or

domain. It helps the LLM adapt its statistical patterns to better match the desired application, improving performance. LLMs can be instructed via 'prompts' or 'demonstrations' - this is refered to as In-Context Learning. A well-designed prompt supplies the model with examples of the intended input and desired output format. LLMs can then 'learn' the pattern and generalize to similar unseen cases, often reducing or even eliminating the need for fine-tuning. LLMs are now widely used in the form of models which have been fine-tuned on the task to function as general chat programs. Usually this is done by reinforcement learning with human feedback ($\rightarrow$ Training).

LLMs possess some significant limitations. 1) Factual Inconsistencies or $\rightarrow$ hallucinations: LLMs, due to their statistical nature, are prone to generating factually incorrect or misleading statements. This tendency, coupled with fluent text generation, can make it difficult to distinguish LLM output from reliable information. 2) Biases: Training LLMs on real-world textual data means they can inherit and perpetuate biases existing within those corpora (Bender et al., 2021). Biased outputs raise serious concerns about fairness and ethical implications. 3) Stability: Small perturbations of the input, e.g. small changes to the prompt, may result in very different output. 4) Opacity: The sheer size and complexity of LLMs often result in them being 'black boxes'. Understanding exactly how these models reach decisions or produce outputs is a significant challenge for researchers, impacting explainability and efforts to mitigate the issues mentioned above. $\rightarrow$ Probing is used to understand the internal workings of LLMs to mitigate some of these problems.

Mathematical foundations Three notions have been essential not only to the design and to regular calculations in and with LLMs but to NLP in general. In language and text models, units are represented as vectors in an n-dimensional vector space model. In the classic bag-of-words model, each document is represented as one vector with n word frequencies as dimensions. Recent langauge models rely upon word vectors rather than text vectors, where the co-textual context forms the coordinates of each vector. A number of m units (words or documents) yields a m x n matrix for an n-dimensional vector space. Based on the bag-of-words-model, corpora are represented as document-term-matrices. In the model of word vectors, m words yield a m x n matrix of a sample of m words where each word is represented as an n dimensional vector. One mathematical strength of vector-space models and the representation of linguistic units in matrices is the computability. Comparability of linguistic units such as the similarity of word meaning or the stylistic similarity of whole documents can be calculated by classic geometric distance measures (such as Manhattan, Euclidean, and Co-

sine distance). One critical step forward in the development of language models was the analogy between semantic relation and vector calculus. It has been shown, that subtracting the vector of the word 'man' from the vector of 'king' and adding the vector of 'woman' results in a synthetic vector that has the lowest distance to the word vector of the word 'queen'. This result demonstrates that language models that are based on word vectors are capable of learning latent semantic relations without explicitly being trained on these semantic relations.

Meaning The concept of meaning is crucial to linguistics, literary studies and criticism as well as to philosophy. 'Meaning' is often disambiguated as 'word meaning', 'word sequence meaning', 'utterance meaning' and 'utterer's meaning' (Tolhurst, 1979). These distinctions refer to controversies regarding the status of intentionality, truth, the necessity of concrete reference, and of temporally and spacially specified situation (grounding) for the actualization of linguistic meaning. These distinctions and the underlying controversies are critical to current discussion on LLMs' capability of generating not only textual output but also meaningful content.

Neural Network Artificial neural networks are a family of algorithms used for machine learning. It is loosely based on a very simplified model of a biological brain. Its basic building blocks are nodes, also called neurons, and edges which control the flow of information (Russell, Norvig 2016: Chapter 22). Neural networks can be used in a supervised and a unsupervised framework. In a supervised setting neural networks are first trained to solve a task using data which contain the correct answer for the task. If, for example, the task is the classification of emails as 'spam' or 'not-spam', the training data consists of many emails, each with a label identifying the category it belongs to. Training is one of two states of a neural network; prediction is the other. The training creates a model and this model can then be used to predict the answer to the same task on new, yet unseen data.

The nodes of a neural network are usually organized in layers, which receive their input from the previous layer and send their results to the next layer. Each node receives information from its input edges, processes the information and sends an output to its output edges connected to nodes from the next layer. How the nodes of a layer are connected to the nodes of the same layer, to the previous layer and the next layer is determined by the architecture of the network. Because there are usually many layers in networks after 2005, they are also called deep neural networks.

In a single node the processing of the information usually consists of a multi-

plication of the input with a list of numbers specific to the node called 'weights' (and the addition of a 'bias' term) (Goodfellow et al., 2016). The result of this multiplication is sent through an activation function, which has often a very simple form, for example setting all negative numbers to zero. It has been shown that with this step even simple feedforward layers can, if they are large enough, approximate any possible function (Hornik et al., 1989). While the architecture of a neural network is stable, the weights are changed during the training. In the beginning of the training the weights are set to random numbers. Then training data is sent through the network and the predicted output is compared to the label of the data. Based on this comparison in a process called 'backpropagation' all the weights of the network are slightly adjusted to provide a better result for the next forward pass. At the end of the training process the weights have values which allow the network as a whole to achieve its task in an optimal way. After the training the weights are not changed. The weights, the task-adapted numbers in each node, not only help to solve the tasks; they also represent those aspects of the information which are needed to solve the task. If, for example, the task is an image classification, then the weights will represent aspects of the image. The weights of layers closer to the input represent more basic information like lines, while weights of layers closer to the output represent more abstract information like an 'eye'. Very often the weights of networks trained on a very generic task, for example masked word prediction, are used as semantic representations; in this context the weights are called → embeddings.

The architectures which have been used in the first two decades of the 21st Century, e.g. Convolutional Network, Gated Recurrent Unit, Long Short-Term Memory, have been superseded by the → Transformer architecture (Vaswani et al., 2017), which has been very successful across many different media and tasks. While it has been common to train networks from one end (the input data) to the other end (the answer to the task), in recent years a two-step approach became established: first, neural networks are trained on a generic task and a huge amount of data, which results in semantically rich representations, secondly the model is 'fine-tuned' to a specific task. Because the first step is so time-consuming and expensive, new workflows were developed where a model was pre-trained on a huge amount of data using a generic task like next word prediction. Then this foundational model (Bommasani et al., 2021) is fine-tuned for many different tasks. Since 2023 very large models like ChatGPT 4, Llama 2 (Touvron et al., 2023) or Gemini (Team, 2023) have been published. After the first phase, during which the models were trained on next word prediction, in a second phase reinforcement learning with human feedback was used to "further align model behavior with human preferences and instruction following." (Touvron et al., 2023). GPT 4 and Gemini are also multi-modal model, capable of processing

text, audio, images and videos. Very recently published or announced models like Mixtral, Gemini 1.5 and (probably) GPT 4 use eight or more different models in a mixture of experts setup, where each part of the input is sent to only one of the models (Shazeer et al. 2017). All these newer models provide a chat interface which allows it users to phrase the task using natural language. Their very rich semantic representation very often enables zero-shot learning making a fine-tuning to the specific task a user wants to solve not necessary any longer. In this context the phrasing of the task, the → prompt engineering, becomes more and more important to achieve reliable results (Wei et al., 2023).

Probing is a technique that indirectly analyzes the opaque internal representations of a trained large language model through targeted tasks. The goal is to understand, for example, the model's linguistic or geographic knowledge or gender biases. The generated output is used as a proxy for the original model.

Prompt engineering Since the establishment of instruct-layer based language models that operate as chat bots (most popularly ChatGPT by OpenAI), the design and construction of prompts has become more and more a proper field of applied computer science, of job profiles and thus of professional engineering. The growing importance of prompt engineering also raises question regarding the transformation of methods and analytical techniques in digital humanities research. For example, a significant branch of computational literary studies (CLS) is occupied with operationalizing the detection of complex textual features such as different types of speech and thought representation. In such fields, rule-based programming that detects specific linguistic features has largely been replaced by algorithmic models based on training data during the last two decades. While rule and model-based analysis requires programming skills, prompt engineering is largely a matter of interacting with a language model on the level of natural language use. It is an open question to what extent rule- and model-based text analysis will be replaced by prompt engineering also in digital humanities methodology. Different techniques that are oriented towards natural conversation have emerged. Among these, tree-of-thought (ToT) and chain-of-thought (CoP) prompting have gained the highest interest in the Humanities. CoT is used to solve a problem as a series of intermediate steps prior to giving a final answer. It gains its strength from forcing the model to base its consecutive answers on its prior output and thus to refine its answers during the chain of thoughts. Ted Underwood could show the increase of performance based on CoT in a large scale macro analysis of the passage of time in narrative fiction. ToT generalizes CoT by asking the model to provide different solutions and to

run the model on each solution respectively. It is likely that model based and prompt based text analysis are going to be in competition for the next years.

Sentence Embedding is a numeric representation of a sentence, presented as a vector of real numbers. Contextualized embeddings play a role in this process, combining static embeddings to convey the context-dependent meaning of each token. BERT (Devlin et al., 2019) is a model that adopts this approach, built upon a Transformer architecture.

Theory of Mind is a concept developed in cognitive science that describes the ability of humans to incorporate the intentions and beliefs of others, including their respective levels of knowledge about the situation, into their mental representation of the situation.

Transformer is an architecture for → neural networks. Since its first description (Vaswani et al., 2017) where it was applied to machine translation, it has become the standard architecture not only for natural language processing but also image processing and multi-modal models. Transformers are especially good in handling sequential data. The paper describes an architecture where the input is passed through an encoder and then through a decoder. Each of them consists of a specific building block which is repeated n times (the size of n is part of the specific implementation). The main component in these building blocks is multi-head self-attention. Self attention checks the connection between every element of the input to each of its other elements giving those elements a higher weight which are important for the solution of the task. Multi-head means that the same process is happening in parallel on the same input whereby different task-relevant aspects of the input are highlighted in each head, for example semantic and syntax. In contrast to earlier architectures for sequential data Transformers don't process them sequentially step by step, for example word by word, but in parallel. This is made possible by positional encodings of the input. One of the main limitations of Transformers was the limited size of the input it can handle, because during self attention each input element is connected to each other element, which means that if you double the size of input elements there are now 4 times as many connections. Recent models use different strategies to handle this limitation and are able to handle up to 1 mio tokens context.

Training is the process of optimizing the parameters of a statistical or neural model using training data. In simplified terms, the training of a statistical model adjusts its rule probabilities according to the distribution of the patterns that

are found in the training data such as how often a particular syntactic structure occurs. In the case of → large language models, it is not probabilities that are optimized but the 'loss function' that measures the discrepancy between predictions of the model and what is found in the training data. Here, the training objective is to minimize the loss function. In general, the training data can be unstructured text or annotated instances, such as word tokens annotated with part-of-speech labels. For example, in the case of pre-training LLMs, the training data are large amounts of raw text; in the case of fine-tuning LLMs, the training data often include annotations of the target output. A side effect of training a model is that the model learns whatever biases are in the training data. Given that the world is an unjust place, models trained on texts produced in this world appropriate the prejudices and stereotypes inherent in the texts—and may even reinforce them. The debiasing of LLMs is an important but unresolved task. A special case of training is Reinforcement Learning from Human Feedback (RLHF), which was extensively used to improve ChatGPT. It describes a feedback loop in which the model's output is evaluated by a human. Compared to training on data, RLHF is like teaching a child something through guidance and feedback instead of letting them learn through trial and error.

Word Embedding establish a link between operations in the vector space and the semantic/syntactic properties of words, by mapping words from the vocabulary to points in an n-dimensional vector space. This approach, originally outlined by Rumelhart et al. (1988), became popular with the introduction of word2vec by Mikolov et al. (2013).

References

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. https://doi.org/10.1145/3442188.3445922 On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623, Virtual Event Canada. ACM.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn,

Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. https://doi.org/10.48550/ARXIV.2108.07258 On the Opportunities and Risks of Foundation Models. Publisher: arXiv Version Number: 3.

Igor Douven. 2011. `https://plato.stanford.edu/ENTRIES/abduction/` Abduction. Last Modified: 2021-05-18.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Zellig S. Harris. 1954. https://doi.org/10.1080/00437956.1954.11659520 Distributional Structure. *WORD*, 10(2-3):146–162.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. https://doi.org/10.1016/0893-6080(89)90020-8 Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

Alessandro Lenci. 2008. Distributional semantics in linguistic and cognitive research. *Italian journal of linguistics*, 20(1):1–31.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. https://arxiv.org/abs/1301.3781v3 Efficient Estimation of Word Representations in Vector Space. *arXiv*.

Magnus Sahlgren. 2008. The distributional hypothesis. *Italian Journal of Disability Studies*, 20:33–53.

Gemini Team. 2023. https://doi.org/10.48550/arXiv.2312.11805 Gemini: A Family of Highly Capable Multimodal Models. ArXiv:2312.11805.

William E. Tolhurst. 1979. On What a Text is and How it Means. *The British Journal of Aesthetics*, 19(1):3–14.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. https://doi.org/10.48550/arXiv.2307.09288 Llama 2: Open Foundation and Fine-Tuned Chat Models. ArXiv:2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. `https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf` Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. https://doi.org/10.48550/arXiv.2201.11903 Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. ArXiv:2201.11903.