[Language Model] A statistical model of language can be represented by the conditional **probability** of the next word given all the previous one.

$$\hat{P}(w_1^T) = \prod^T \hat{P}(w_t|w_1^{t-1}),$$
$$w_i^j = (w_i, w_{i+1}, \cdots, w_{j-1}, w_j)$$

[Artificial Intelligence] → "simulating" human
1. ANI (Narrow): Machine Learning
2. AGI (General): Machine Intelligence
3. ASI (Super): Machine Consciousness

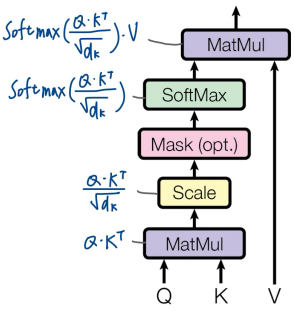
[Attention is all you need] Why attention? RNN: hard to parallel, sequential

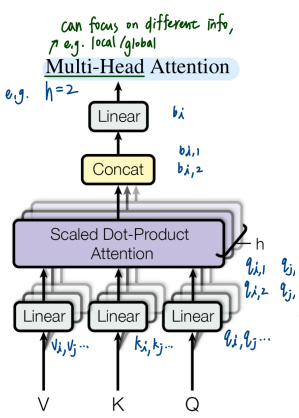
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

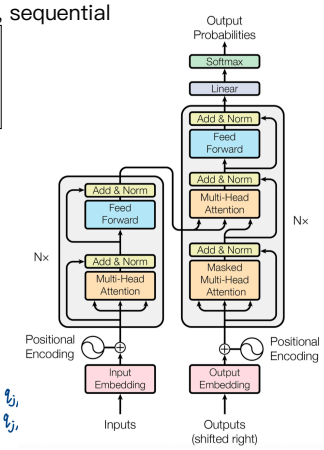
$$s\left(x_i\right) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

① Exponent of xi
② Total value of x exponent

Scaled Dot-Product Attention







[Moravec's Paradox]
Conscious/Easy/learnt recently
Unconscious/Hard/reinforced millions yrs

Decoder: difference between self/cross attention?

self-attention → all from decoder

$$\text{softmax}\left(\frac{Q_D K_D^T}{\sqrt{d_k}}\right) V_D$$

cross-attention → key and value from encoder

$$\text{softmax}\left(\frac{Q_D K_E^T}{\sqrt{d_k}}\right) V_E$$

[Perplexity] how well a model fits the test data and predictive power
word-error-rate (WER): simple model / perplexity: neural models
* compare 2 Language Model sharing common vocabulary
Compute perplexity across multiple sentences:
Token-level aggregation, not sentence-level averaging (overweight shorter sentences, exponential)
lower score is better, the model assigned a higher prob to the sentence

Corpus with M sentences, lengths N1, N2, …, NM
N = the number of tokens
H = sum of log-prob of tokens
PP = perplexity

$$N = \sum_{j=1}^{M} N_j. \qquad H = -\frac{1}{N} \sum_{j=1}^{M} \sum_{i=1}^{N_j} \log P(w_i^{(j)} \mid w_{<i}^{(j)}).$$
$$PP = \exp(H).$$

$$Q = [5]$$
$$K_1 = [10], K_2 = [5]$$
$$V_1 = [-1], V_2 = [2]$$

$QK_1 = 50$ $QK_2 = 25$ $QK^T = [50, 25]$ $q_k = 1$

$$\text{softmax}([50, 25])\, V = \left[\frac{e^{50}}{e^{50}+e^{25}}, \frac{e^{25}}{e^{50}+e^{25}}\right]\begin{bmatrix}-1\\2\end{bmatrix} = \frac{-e^{50}+2\cdot e^{25}}{e^{50}+e^{25}}$$

Imagine two sentences:
- Sentence 1: 4 tokens, log-probs = $-1.0, -0.5, -2.0, -0.2$.
- Sentence 2: 2 tokens, log-probs = $-0.1, -0.3$.

Token count: $N = 6$.
Total log-prob sum: $-1.0 - 0.5 - 2.0 - 0.2 - 0.1 - 0.3 = -4.1$.
Average: $-4.1/6 = -0.6833$.
Perplexity = $\exp(0.6833) \approx 1.98$.

Probability Formula:
$P(\text{'the bird flew over'}) = P(\text{'the'}) \times P(\text{'bird'}) \times P(\text{'flew'}) \times P(\text{'over'})$

Conditional Probabilities:
- $P(\text{'the'}) = \frac{\text{Count('the')}}{\text{Total Words}} = \frac{30}{90}$
- $P(\text{'bird'}) = \frac{\text{Count('bird')}}{\text{Total Words}} = \frac{4}{90}$
- $P(\text{'flew'}) = \frac{\text{Count('flew')}}{\text{Total Words}} = \frac{2}{90}$
- $P(\text{'over'}) = \frac{\text{Count('over')}}{\text{Total Words}} = \frac{2}{90}$

Total Sentence Probability:
$$P(W_{\text{uni}}) = \frac{30}{90} \times \frac{4}{90} \times \frac{2}{90} \times \frac{2}{90} = \frac{480}{65{,}610{,}000} \approx 7.31 \times 10^{-6}$$

Perplexity:
$$\text{Perplexity} = (P(\text{Sentence}))^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}}$$
$$\text{Perplexity}_{\text{uni}} = (P(W_{\text{uni}}))^{-1/4} = \left(\frac{480}{65{,}610{,}000}\right)^{-1/4} \approx \mathbf{19.23}$$

$Q = [-1, 1]$
$K_1 = [0, 1]$
$K_2 = [1, 0]$
$V_1 = [1, 2]$
$V_2 = [-1, -2]$

$$QK^T = [-1, 1]\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix} = [1, -1]$$
$$\frac{QK^T}{\sqrt{d_k}} = \frac{[1, -1]}{\sqrt{2}} = \left[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right]$$
$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot V = \left[\frac{e^{\frac{1}{\sqrt{2}}}}{e^{\frac{1}{\sqrt{2}}}+e^{-\frac{1}{\sqrt{2}}}}, \frac{e^{\frac{1}{\sqrt{2}}}}{e^{\frac{1}{\sqrt{2}}}+e^{-\frac{1}{\sqrt{2}}}}\right]\begin{bmatrix}1 & 2\\-1 & -2\end{bmatrix}$$

[N-Gram Model] est. prob of each word given N-1 words of prior context

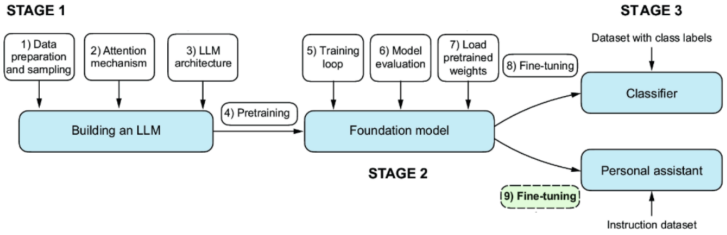
Bigram $P(w_1^n) = \prod_{k=1}^{n} P(w_k \mid w_{k-1})$ N-gram $P(w_1^n) = \prod_{k=1}^{n} P(w_k \mid w_{k-N+1}^{k-1})$

- Chain rule of probability

$$P(w_1^n) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1^2)\ldots P(w_n \mid w_1^{n-1}) = \prod_{k=1}^{n} P(w_k \mid w_1^{k-1})$$

[lower-order backoff method] 1st: unigram; 2nd: bigram, rest: trigram
P(the)P(cat|the) P(jumped|the cat)P(over|cat jumped)
[dummy start tokens]
P(the|<s><s>)P(cat|the,<s>)P(jumped|the cat)P(over|cat jumped)
P(<s> i want english food </s>)= P(i | <s>)*P(want | i)*P(English | want)*P(food | English)*P(</s> | food)

[Pre-training] initial training on large, generic data to learn fundamental features *self-supervised learning
[Fine-tuning (SFT)] improved performance on particular task with specific, label data



[Dialogue] both parties provide feedback, important application of LLMs (chatbots)
*Discourse: only speaker & passive listener
- Retrieval-based approaches: find most similar responses
- Neural approaches (seq2seq models): trained on large corpora of dialogues, reinforcement learning from human feedback (RLHF)
- Issues: hallucinations, bias…

[Bias]
Sources:
- Training datasets: data predominantly represents certain demographics or perspectives
- Algorithms: amplify bias by autonomous learning behaviors of complex models
- Human subjectivity: data preparation processes like labeling and annotation

Debiasing:
- Preprocessing: Counterfactual Data Augmentation (CDA), rebalanced by alter specific association
- Training: involve regularization terms & various loss functions, contrastive loss
- Post-processing: most important, prompt engineering

[Hallucinations] generated content from nowhere (knowledge boundaries)
- Intrinsic: directly conflict with provided source, e.g. prompt LLM with paper with 3 authors, but LLM says only 2 authors.
- Extrinsic: output cannot be verified, e.g. asked LLM to give a survey, but return incorrect material
- Knowledge Boundaries: 1. Inability of LLMs to memorize, 2. Intrinsic boundary, does not include rapidly evolving world knowledge

[Embedding]

1. Why do we often use cosine similarity instead of Euclidean distance when comparing embeddings?

• Cosine compares direction and scale–invariant. If two vectors point in same direction but different lengths, cosine still reports high similarity.

• Euclidean distance is sensitive to magnitude.

2. If we rotate all word embeddings by the same angle in space, do relative similarities between words change? Why or why not?

No, since rotation preserves the angles and distances between all vectors in the space, both cosine similarity (which depends on the angle) and Euclidean distance (which depends on distance) remain constant for any pair of words.

3. Let u=(2,1), v=(1.3)

(a) dot product $u \cdot v = 2 \cdot 1 + 1 \cdot 3 = 5$

(b) norms $|u| = \sqrt{2^2+1^2} = \sqrt{5}$, $|v| = \sqrt{1^2+3^2} = \sqrt{10}$

(c) cosine similarity $\frac{u \cdot v}{|u||v|} = \frac{5}{\sqrt{5}\sqrt{10}} = \frac{1}{\sqrt{2}} \approx 0.707$

$$\cos\theta = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\|\vec{b}\|} \qquad \|\vec{a}\| = \sqrt{a_1^2 + a_2^2 + a_3^2 + \cdots + a_n^2} \qquad \|\vec{b}\| = \sqrt{b_1^2 + b_2^2 + b_3^2 + \cdots + b_n^2}$$

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

4. If embeddings from clusters for "sports news" and "finance news", what does this mean?

the model encodes topical information, sentences about similar subjects are placed close together, so embeddings separate content by topic.

Q1 You are given 3D embeddings: orange = (0,1,2) grape = (1,1,1) truck= (4,4,0)

1. Using cosine similarity, which is closer to orange: grape or truck? Show dot products and norms.
2. Using Euclidean distance, which is closer to orange: grape or truck? Show distances.
3. Briefly explain (1–2 sentences) why the two metrics can disagree.

$orange \cdot grape = 0 \cdot 1 + 1 \cdot 1 + 2 \cdot 1 = 3 \rightarrow \cos(o,g) = \frac{3}{\sqrt{5}\cdot\sqrt{3}} = \frac{3}{\sqrt{15}}$

$orange \cdot truck = 0 \cdot 4 + 1 \cdot 4 + 2 \cdot 0 = 4 \rightarrow \cos(o,t) = \frac{4}{\sqrt{5}\cdot\sqrt{32}} = \frac{1}{\sqrt{10}}$

$E(o,g) = (-1,0,1) = \sqrt{2}$

$E(o,t) = (-4,-3,2) = \sqrt{29}$

Q2 2D embeddings (rounded; small noise present):

violin = (2.0, 5.0) musician = (1.0, 2.0)

painter = (2.1, 1.9) painting = (3.0, 5.1)

cello = (2.0, 4.2)

Define t = violin – musician + painter.

Compute t. $t = (3.1, 4.9)$ $|t| = \sqrt{34}$

Compute cos(t, painting) and cos(t, cello)

$\cos(t, painting) = \frac{3 \cdot 3.1 + 5.1 \cdot 4.9}{\sqrt{34}\cdot\sqrt{34}} \approx 1$

$\cos(t, cello) = \frac{2 \cdot 3.1 + 4.2 \cdot 4.9}{\sqrt{34}\ \sqrt{21.04}}$

If the cosines are very close, propose one tie–break rule that does not require retraining.

Euclidean distance

Q3 A low–resource corpus includes many rare words like "microinjury" and "revascularization".

1. In 2–3 sentences, explain how Word2Vec, GloVe, and FastText would differ in handling such words at test time.
2. Give one concrete example where subword n–grams provide a useful vector for a word not seen during training.

Feature	Word2Vec	GloVe (Global Vectors)	FastText
Model	predictive model (local) 1. CBOW: context → target 2. Skip–gram: target → context	count–based model (global): pre–computed word2word co–occurence matrix	subword model: extension of skip–gram, word vector = sum of n–gram sub vectors
Learning	sliding window of context	large matrix of how often word appears	character n–grams to form final word embedding
Unknown	Bad, OOV words	Bad, OOV words	Good, character n–grams
Efficiency	Fast	Memory intensive for large matrix	Slowest, vast num of n–grams

Continuous Bag Of Words (CBOW)

1. One–hot input vectors
2. Lookup embeddings from weight matrix W
3. Average the context embeddings
4. Compute output scores using output matrix W'
5. Apply softmax

Word2Vec/GloVe Approach

• "playing" → single atomic vector lookup

• If "playing" not in vocabulary → <UNK>

FastText's Innovation

"playing" → break into character n–grams: <pl, pla, play, playi, playin, playing><la, lay, layi, layin, laying><ay, ayi, ayin, aying><yi, yin, ying><in, ing><ng>

• Final embedding = sum of all n–gram embeddings

Aspect	Word2Vec	GloVe
Core Approach	Local context window prediction	Global matrix factorization
Training Data	Individual context windows	Pre-computed co-occurrence matrix
Learning Method	Implicit relationship learning	Explicit relationship optimization
Mathematical Goal	Maximize P(context|word)	Minimize $\|\|w_i^T \tilde{w}_j + b_i + \tilde{b}_j \approx \log X_{ij}\|\|^2$

Category	Count-Based Methods (Global Matrix Factorization)	Prediction-Based Methods (Local Context Windows)
Examples	- Latent Semantic Analysis (1990) - Hyperspace Analogue to Language (1996) - Positive Pointwise Mutual Information (2010)	- Word2Vec (2013) - Neural Language Models
Strengths	- Use global corpus statistics - Mathematically clear	- Capture complex patterns - Great analogy performance

[Core Concepts & The 3 Levels of Analysis]

What is NLP? The branch of AI focused on enabling computers to communicate using human language.

The #1 Challenge: Ambiguity. Natural language is full of ambiguity that must be resolved; computer languages are designed to be unambiguous.

Example: "I saw the man on the hill with a telescope." (Who has the telescope?)

1. **Syntax (Structure):** Grammatical ordering of words.
   - The dog bit the boy. vs. The boy bit the dog.
2. **Semantics (Literal Meaning):** The meaning of words and sentences.
   - plant (a flower) vs. plant (a factory)
3. **Pragmatics (Contextual Meaning):** How social context shapes interpretation.
   - A waiter says: "The ham sandwich wants another beer."

Category	Task	Description
Syntactic (Structure)	Part–of–Speech (POS) Tagging	Annotating each word with its grammatical type (noun, verb, adjective, etc.).
	Parsing	Generating the grammatical tree structure of a sentence.
Semantic (Meaning)	Word Sense Disambiguation (WSD)	Determining the correct meaning of a word in context (e.g., a river
	Semantic Role Labeling (SRL)	Identifying
	Textual Entailment	Deciding if one sentence logically implies another.
Pragmatic (Context)	Anaphora Resolution	Figuring out what a pronoun or noun phrase refers to (e.g., "John put the carrot on the plate and ate
Applications	Information Extraction (IE)	Identifying and extracting specific entities (people, places) and their relationships from text.
	Machine Translation (MT)	Translating text from one language to another.
	Summarization & QA	Creating short summaries of long documents and answering questions based on a body of text.