

Large Language Model Applications

Spring 2026

Daphne Ippolito and Fernando Diaz

logistics

Logistics

Teaching staff

- Instructors
 - Daphne Ippolito, Office hours: Tuesdays 10-11 AM
 - Fernando Diaz, Office hours: Wednesdays 1:00-2:00
- Teaching assistants (office hours TBD)
 - Aditya Oke
 - Mingkai Deng
 - Barry Wang
 - Haoze He
 - Pranav Jaysankar
 - Xinran Zhao

Logistics

Course Website

<https://cmu-llms.org/>

Logistics

Grading

- **Homework:** Comprise a mixture of written and coding questions. 30% Homework
- **Project:** Students will complete a final project that focuses on exploring in depth a specific application of their choice. 40% Project
- **Scribe notes:** Each student will be assigned one lecture to act as scribe and prepare notes to share with the rest of the class one week after the lecture. 20% Scribe notes
- **In-class participation:** Poll Everywhere 10% In-class participation

Logistics

Deadlines

Homeworks

- Homework 1:
 - Due 2/05/2026
- Homework 2:
 - Due 2/26/2026
- Homework 3:
 - Due 3/26/2026

Project Milestones

- Initial Pitch:
 - Due 1/22/202
- Proposal:
 - Due 2/12/2026
- Midpoint check-in:
 - Due 3/12/2026
- Final Report and Presentation
 - Due 4/21/2026

Policy on Use of AI for Homeworks

All written words in your homework and project reports should be written by yourself, without the use of AI.

You may use AI to learn more about the subject material of the class and to help you write code. However, you must explain your AI usage in your homework submission.

See syllabus for more details.

Class Goals

What goes into building and deploying an LLM for a real-world application?

Students who successfully complete this course will be able to:

- Determine the best LLM tools for whatever applications are thrown at them
- Understand how these tools are implemented and feel confident in making modifications to these implementations
- Read and comprehend recent, academic papers on customizing LLMs for different applications
- Contextualize recent LLM advancements in the prior developments that got us here

survey

prehistory of language models

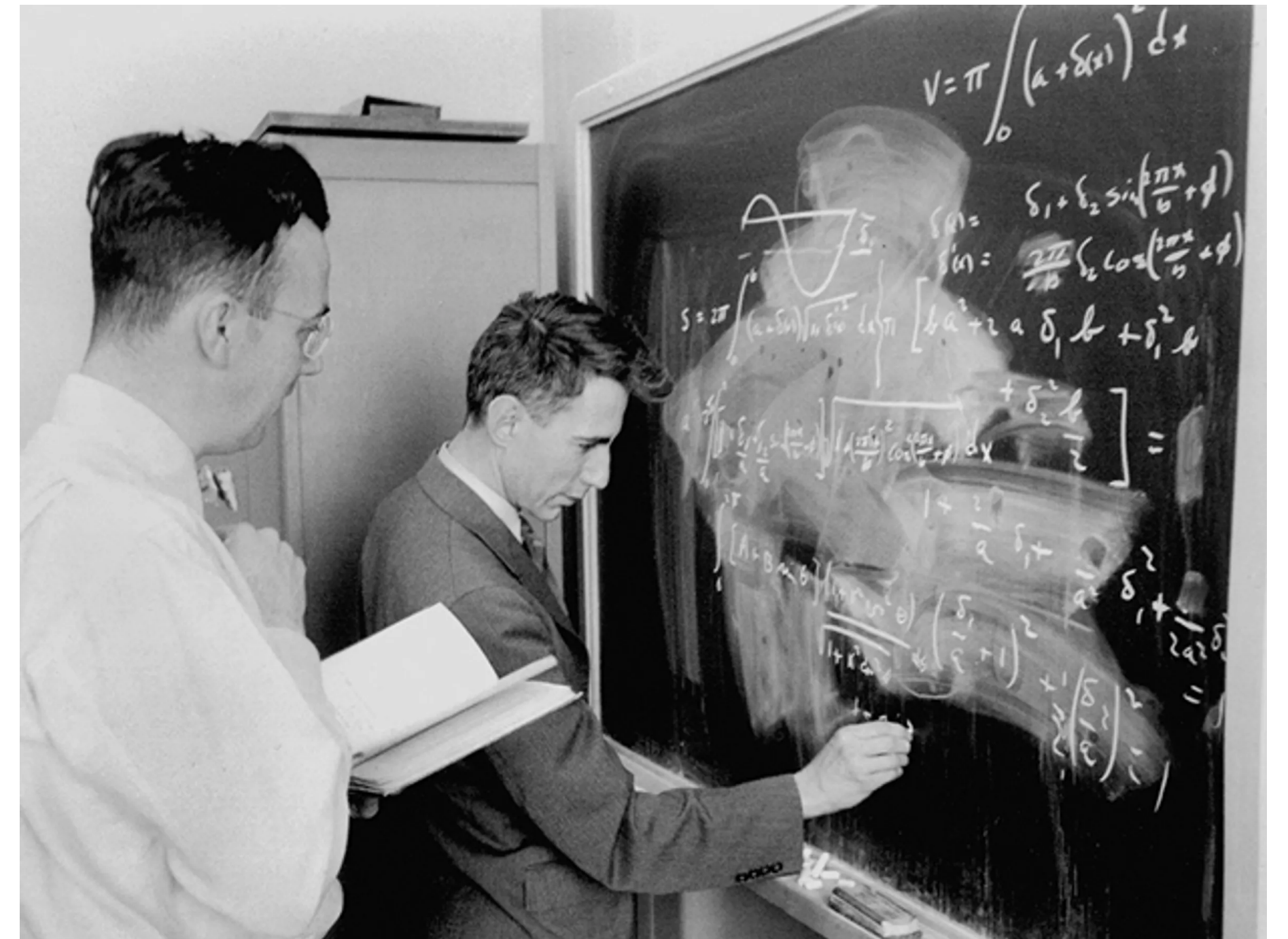
Introduction

- Modern large language models emerged from the convergence of multiple research traditions: information theory and statistical NLP, deep learning and representation learning, reinforcement learning, and NLP applications.
- Many of the key building blocks have existed for decades; what changed was their integration into a single scalable training-and-inference stack.
- Web-scale data, specialized hardware, and distributed systems turned language modeling from a niche component (e.g., MT/ASR/IR) into a general-purpose foundation for applications.
- The central technical through-line is simple: optimize next-token prediction at scale, then add retrieval and control layers to make the model useful, grounded, and steerable.

1. information theory

Information theory

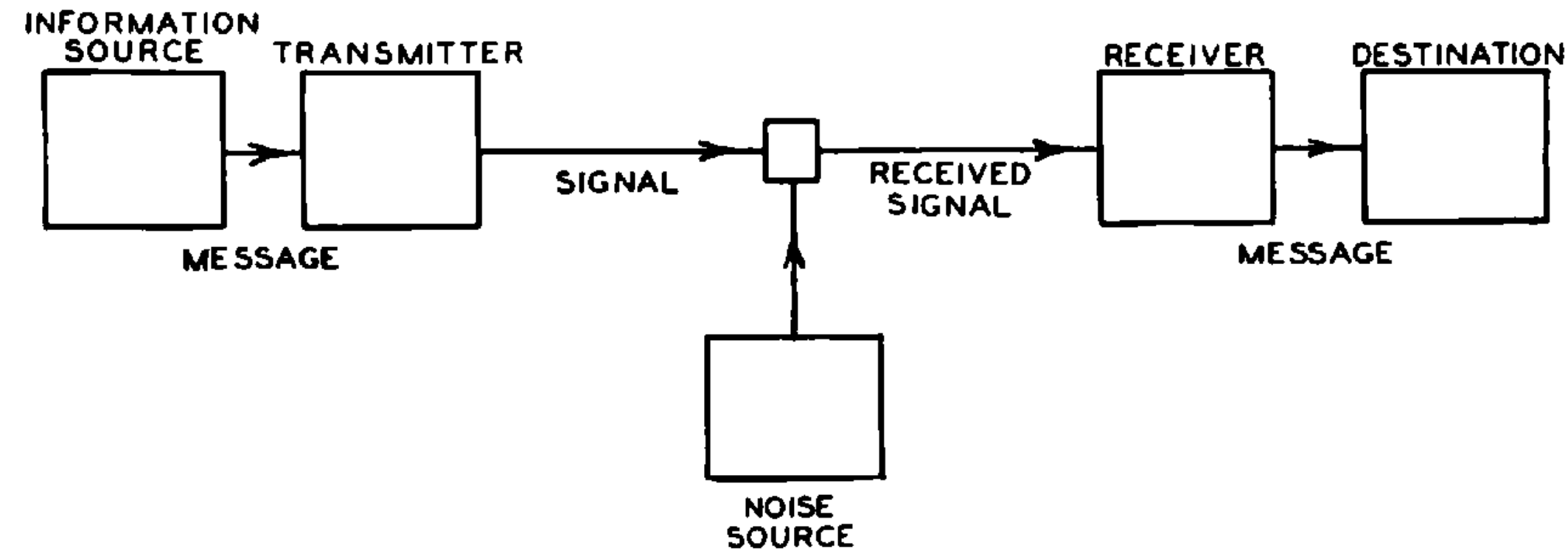
- **Bell Labs.** AT&T's premier research institution, driven to understand the fundamental limits of telephone transmission.
- **Claude Shannon.** Joined Bell Labs in 1941 to work on fire-control systems and cryptography during WWII. In 1948, published "A Mathematical Theory of Communication," introducing the bit, entropy, and channel capacity—the foundation of information theory.



Nokia Bell Labs

Information theory

Noisy Channel (1948)



Shannon's schematic diagram for a general communication system [Shannon 1948].

- Shannon wanted to answer a fundamental question: what is the maximum rate at which information can be reliably sent over a noisy channel?
- **Shannon's Communication Model.** Information source $\rightarrow$ transmitter $\rightarrow$ noisy channel $\rightarrow$ receiver $\rightarrow$ destination.
- Every channel has a maximum rate for reliable transmission. Below capacity, error-free communication is possible; above it, errors are unavoidable.
- Shannon defined information entropy. Language provided a convenient testbed for estimating it empirically.

Information theory

The Shannon Game (1951)

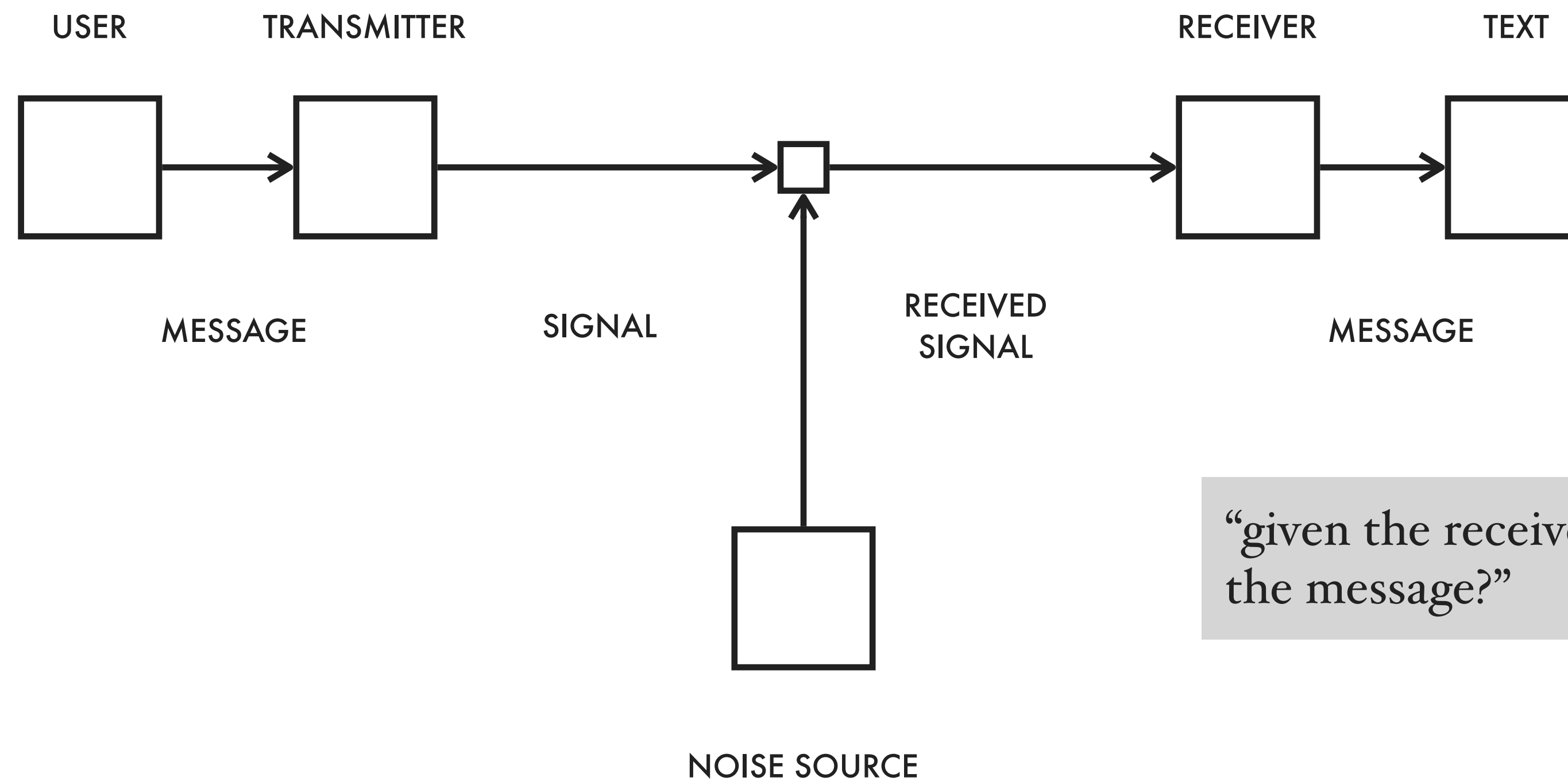
- Shannon defined entropy as the measure of information—the average uncertainty in a message. How do you estimate the entropy of something as complex as English?
- **Setup:** A subject guesses a text passage letter by letter (including spaces = 27 symbols). After each guess, they're told if they're correct; if wrong, they're given the correct letter. They see all revealed text so far.
- Humans exploit statistical redundancy in language. The "reduced text" (just the mistakes) contains the same information as the original: an identical predictor could reconstruct the full text from the errors alone.
- This is exactly next-token prediction. Shannon showed that modeling language = predicting language, and that prediction quality directly measures entropy.

```
(1) THE ROOM WAS NOT VERY LIGHT A SMALL OBLONG
(2) ----ROO-----NOT-V-----I-----SM----OBL----
(1) READING LAMP ON THE DESK SHED GLOW ON
(2) REA-----O-----D----SHED-GLO--O--
(1) POLISHED WOOD BUT LESS ON THE SHABBY RED CARPET
(2) P-L-S-----O---BU--L-S--O-----SH-----RE--C-----
```

A dash means guessed correctly; a letter means that letter was revealed after a wrong guess.

Information theory

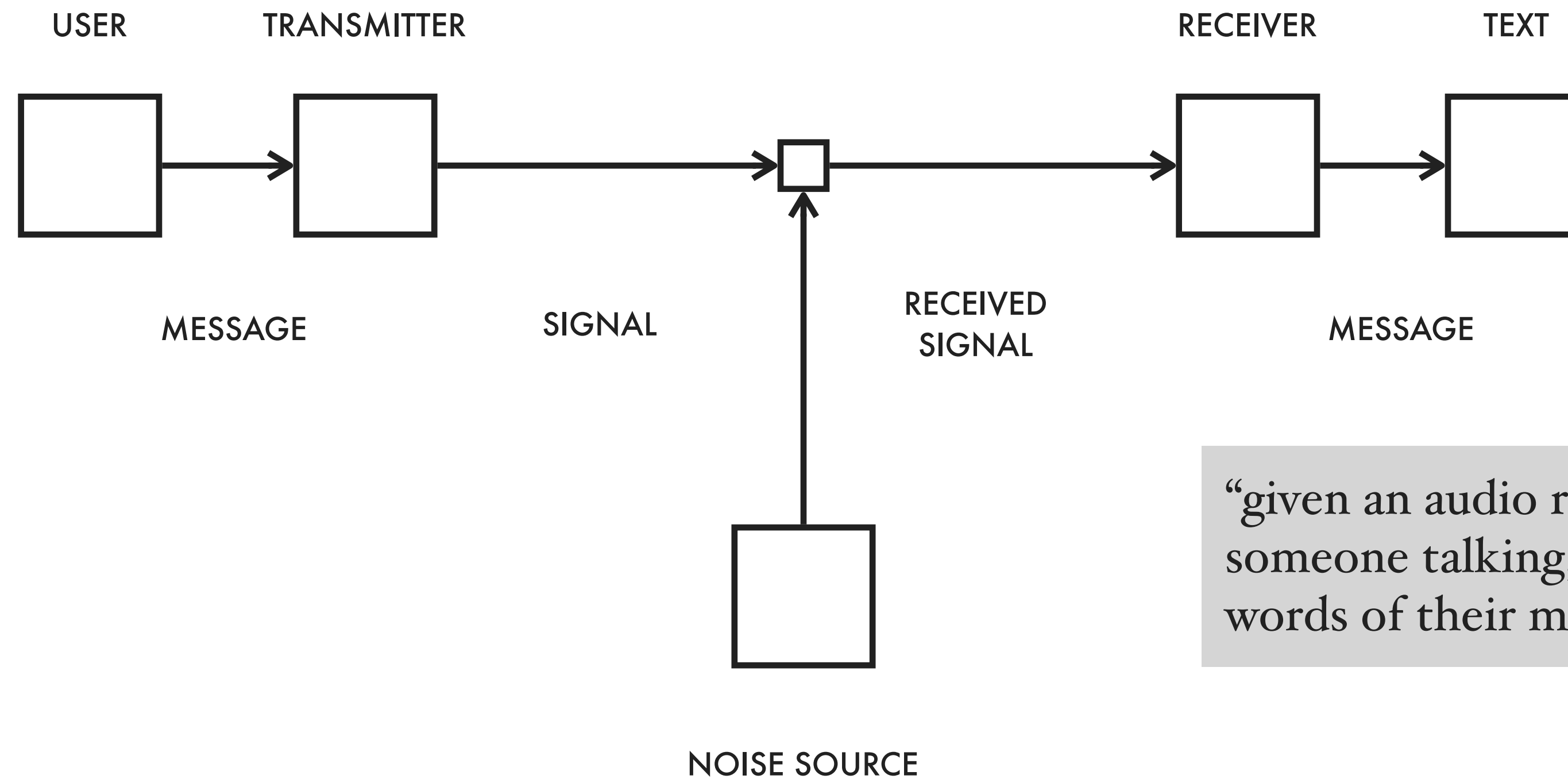
Noisy channel model



“given the received signal, what was the message?”

Information theory

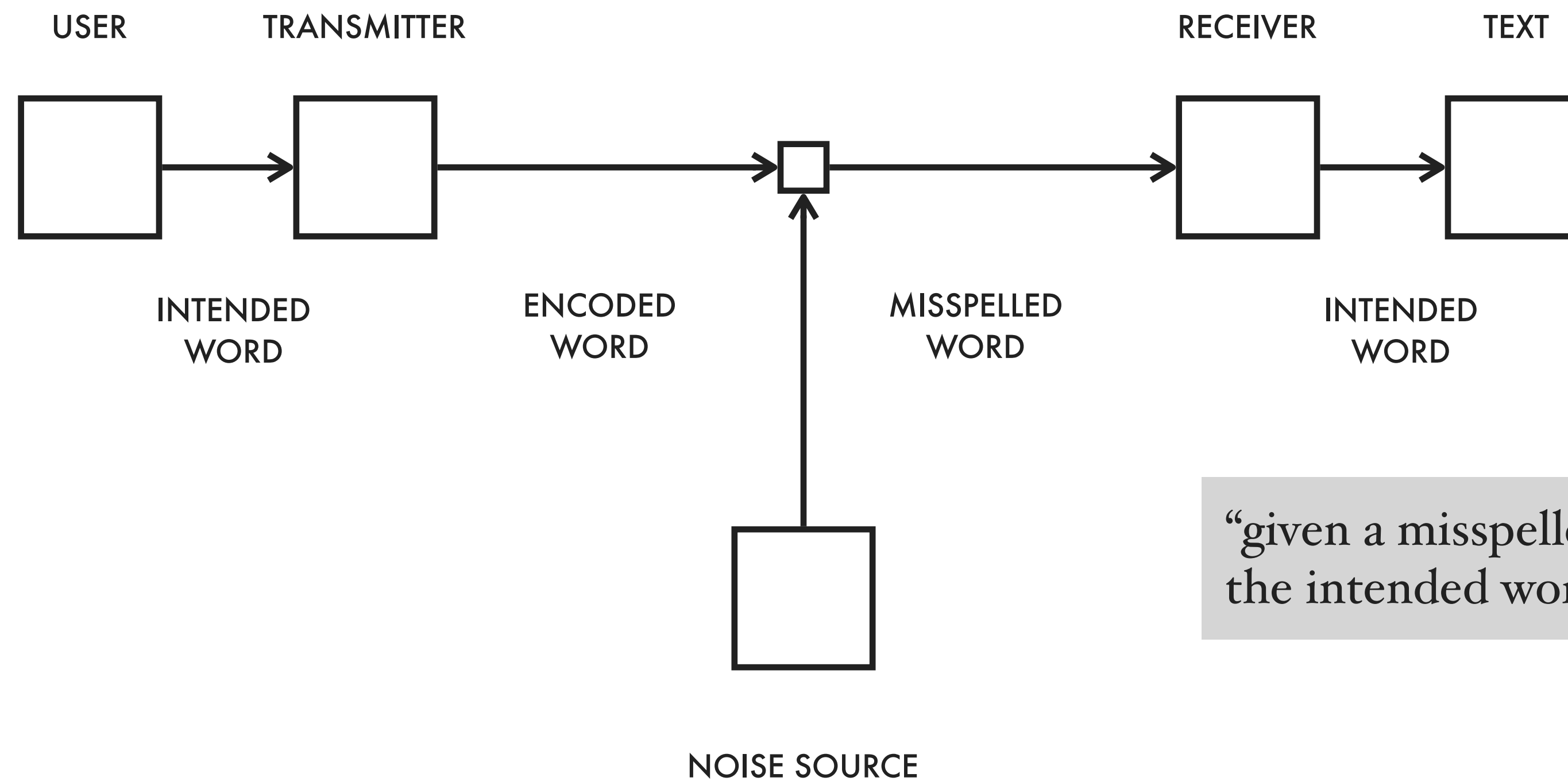
Noisy channel model for **speech-to-text**



“given an audio recording of someone talking, what were the words of their message?”

Information theory

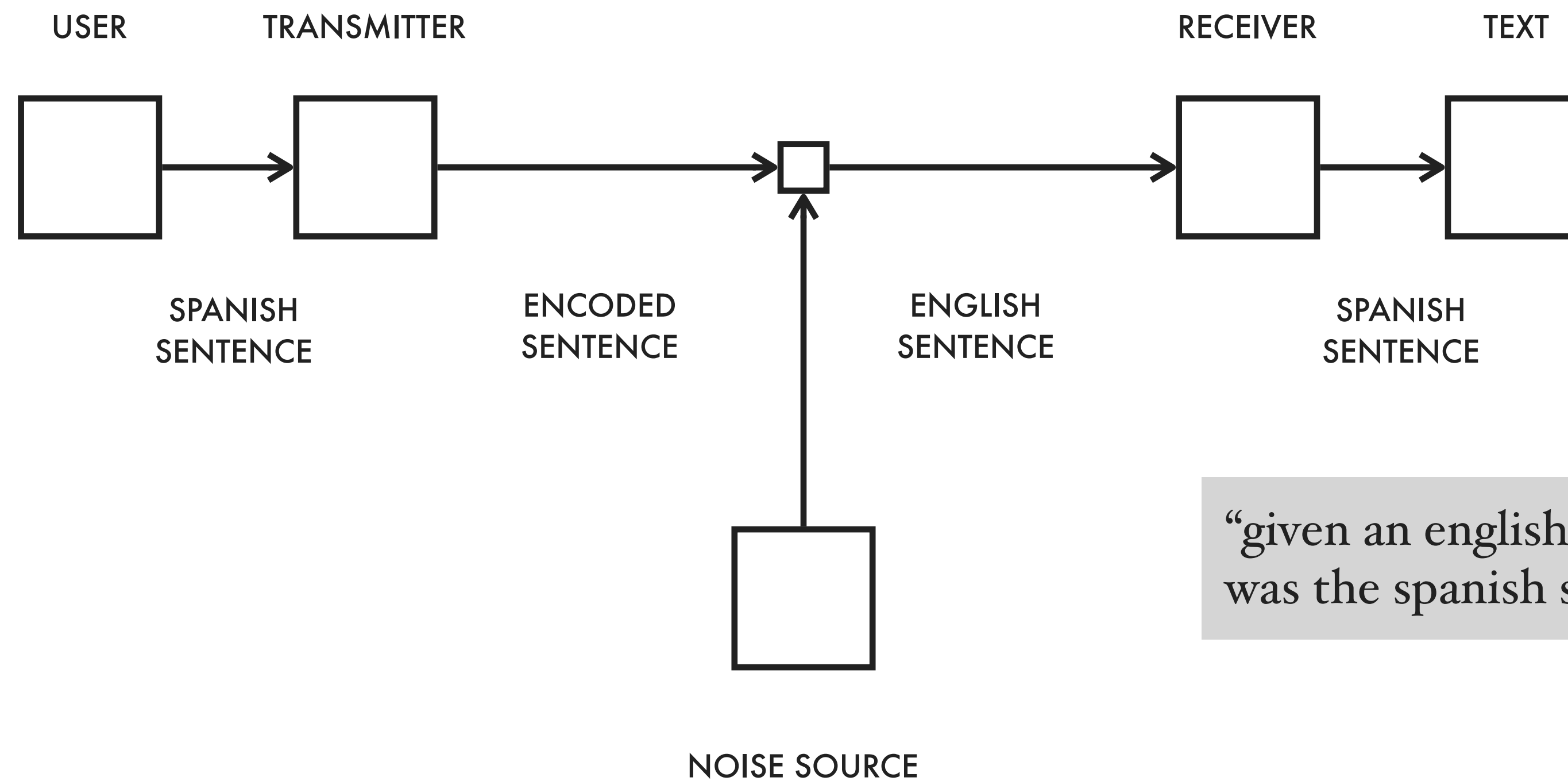
Noisy channel model for spelling correction



“given a misspelled word, what was the intended word?”

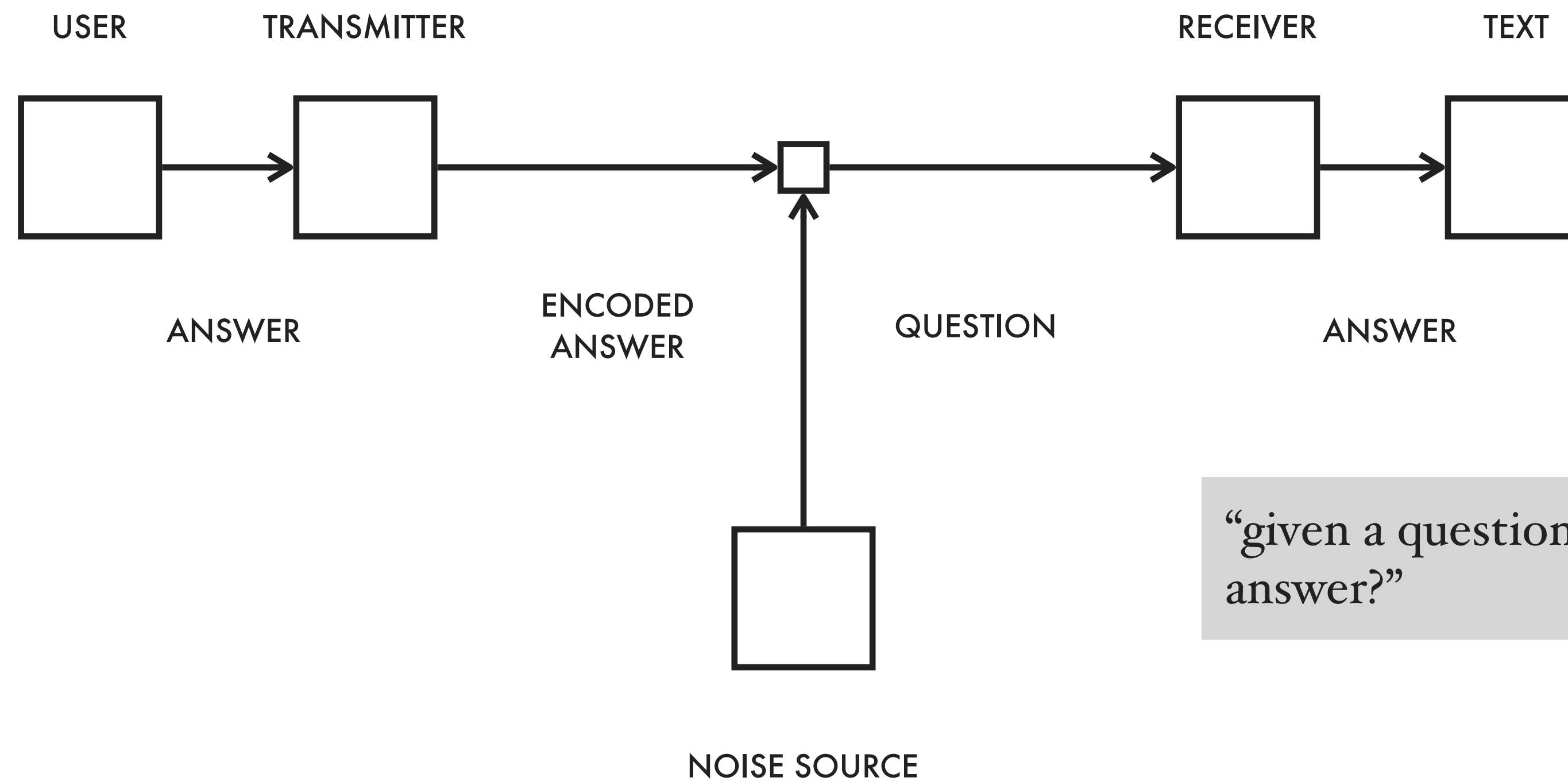
Information theory

Noisy channel model for translation



Information theory

Noisy channel model for **question answering**



Information theory

Noisy channel model

- The noisy channel:
 - We make some observation x (e.g., recorded audio signal; string of characters user typed in)
 - x is a noisy version of y^* , the actual word that was intended
- Modeling the noisy channel:
 - Model the likelihood of the noisy channel producing any particular x as $\Pr(x | y)$
 - We can then compute the most probable word $\tilde{y}$ as:

$$\tilde{y} = \operatorname{argmax}_{y \in \mathcal{Y}} \underbrace{\Pr(x | y)}_{\text{channel model}} \underbrace{\Pr(x)}_{\text{prior}}$$

Information theory

Noisy channel model

- The noisy channel:
 - We make some observation x (e.g., recorded audio signal; string of characters user typed in)
 - x is a noisy version of y^* , the actual word that was intended
- Modeling the noisy channel:
 - Model the likelihood of the noisy channel producing any particular x as $\Pr(x | y)$
 - We can then compute the most probable word $\tilde{y}$ as:

$$\tilde{y} = \operatorname{argmax}_{y \in \mathcal{Y}} \underbrace{\Pr(x | y)}_{\text{channel model}} \underbrace{\Pr(x)}_{\text{prior}}$$



language model!

Statistical Language Models

n -gram models of language

- we can estimate word likelihoods using frequencies in a corpus
- given a sequence $w_1, \dots, w_n$, we factorize the joint probability as,

$$\begin{aligned}\Pr(w_1, \dots, w_n) &= \prod_{i=1}^n \Pr(w_i | w_s, w_1, \dots, w_{i-1}) \\ &= \Pr(w_n | w_s, w_1, \dots, w_{n-1}) \times \Pr(w_{n-1} | w_s, w_1, \dots, w_{n-2}) \times \dots \times \Pr(w_2 | w_s w_1) \times \Pr(w_1 | w_s)\end{aligned}$$

where w_s is a special token indicating the start of the sequence

- we can approximate this with n -grams. for $n = 3$,

$$\begin{aligned}\Pr(w_1, \dots, w_n) &= \prod_{i=1}^n \Pr(w_i | w_{i-2}, w_{i-1}) \\ &= \Pr(w_n | w_{n-2}, w_{n-1}) \times \Pr(w_{n-1} | w_{n-3}, w_{n-2}) \times \dots \times \Pr(w_2 | w_s w_1) \times \Pr(w_1 | w_s)\end{aligned}$$

- the frequencies of all possible n -grams can be pre-computed and stored in a lookup table



Name a problem with statistical
language models.

Curse of Dimensionality

Word sequences are extremely high dimensional.

A word sequence on which the model will be tested is likely to be different from all the word sequences seen during training

2. neural networks

Neural networks

- McCulloch and Pitts laid the theoretical groundwork in 1943 by showing how neurons could be modeled as simple logical units, establishing that networks of such units could, in principle, compute anything
- Frank Rosenblatt's Perceptron generated enormous excitement in the late 1950s as a learning machine that could be trained to recognize patterns, prompting bold predictions about thinking machines
- Arthur Samuel's checkers program demonstrated that machines could improve through experience, playing thousands of games against itself and eventually beating skilled human players
- The field revived dramatically when Rumelhart, Hinton, and Williams showed in 1986 that backpropagation could train multi-layer networks

Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4):115--133, 1943.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain.. Psychol Rev, 65(6):386--408, Nov 1958.

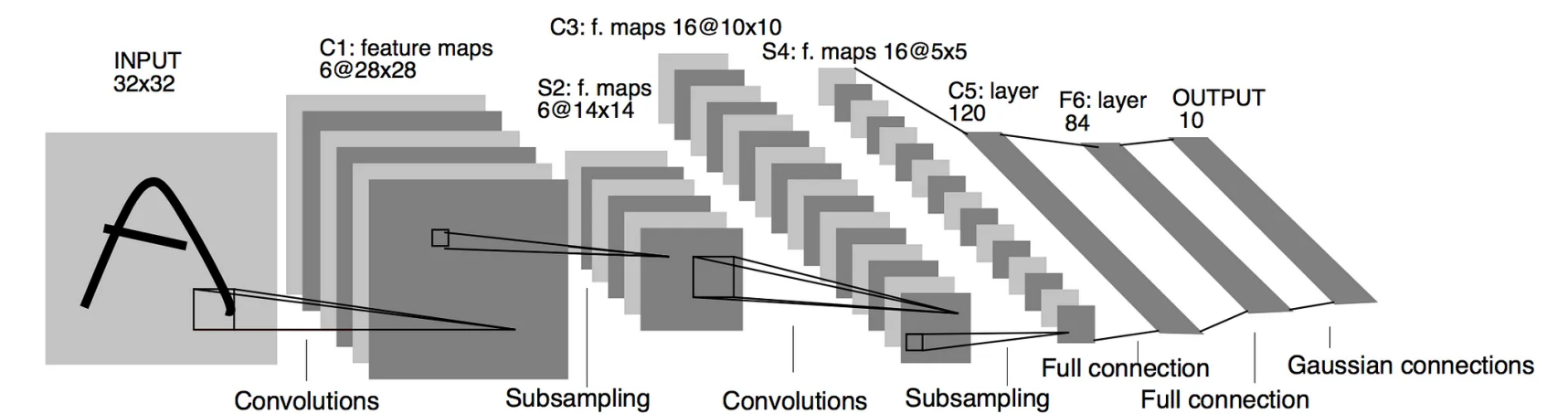
A. L. Samuel. Some studies in machine learning using the game of checkers. IBM Journal of Research and Development, 3(3):210-229, 1959.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. Nature, 323(6088):533--536, 1986.

Neural networks

1990s

- Elman developed recurrent neural networks that could process sequences, introducing the idea of hidden states that carry information forward through time
- Hochreiter and Schmidhuber invented Long Short-Term Memory networks to solve the vanishing gradient problem, enabling networks to learn dependencies across long sequences of text
- LeCun *et al.* refined backpropagation and convolutional architectures through the 1990s, establishing training techniques and architectural patterns that would scale to much larger models



LeNet architecture [Lecun et al. 1998]

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179-211, 1990.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735-1780, 1997.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278-2324, 1998.

Neural Language Models

“A Neural Probabilistic Language Model” (2003)

- Key goal:
 - Suppose “The cat ate tuna for dinner.” is high likelihood according to the training data.
 - “The kitty consumed fish for supper” should also be high likelihood even if it *never* occurs in the training data.

Neural Language Models

“A Neural Probabilistic Language Model” (2003)

In a nutshell, the idea of the proposed approach can be summarized as follows:

1. associate with each word in the vocabulary a distributed *word feature vector* (a real-valued vector in $\mathbb{R}^m$),
2. express the joint *probability function* of word sequences in terms of the feature vectors of these words in the sequence, and
3. learn simultaneously the *word feature vectors* and the parameters of that *probability function*.

Neural Language Models

“A Neural Probabilistic Language Model” (2003)

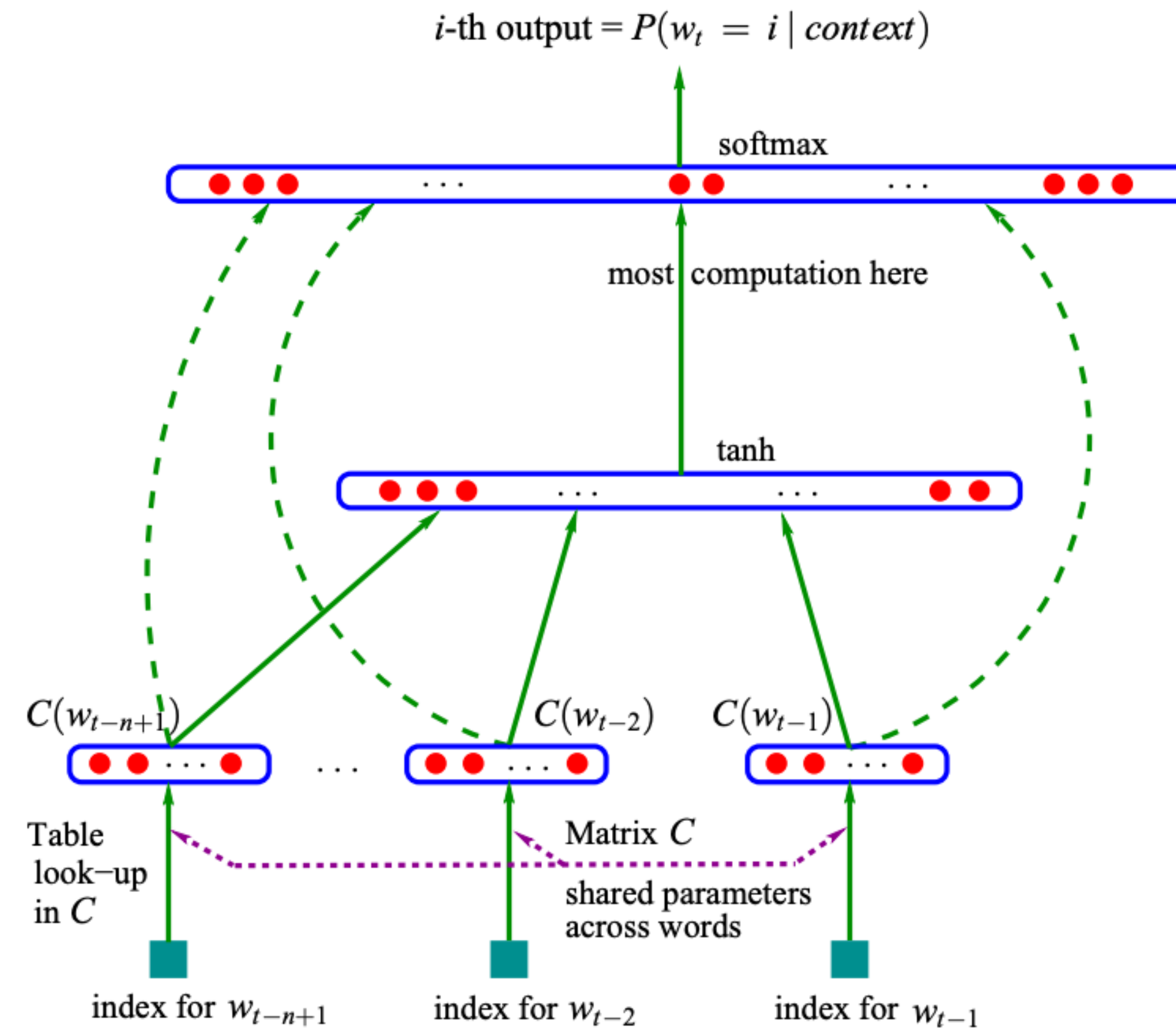


Figure 1: Neural architecture: $f(i, w_{t-1}, \dots, w_{t-n+1}) = g(i, C(w_{t-1}), \dots, C(w_{t-n+1}))$ where g is the neural network and $C(i)$ is the i -th word feature vector.

Highlights of Neural Networks for Language

2003-2017

- Word embeddings
 - Word2vec, GLoVe
- Sentence embeddings
 - SkipThought vectors, Elmo
- Advancements in recurrent neural networks
 - RNNs -> LSTMs -> GRUs
- Neural language models deployed at scale
 - Google Neural Machine Translation System
 - Facebook DeepText



Name a problem with word embeddings (e.g. word2vec).



Name a problem with recurrent
neural networks (e.g. LSTMs).

3. reinforcement learning

Reinforcement learning

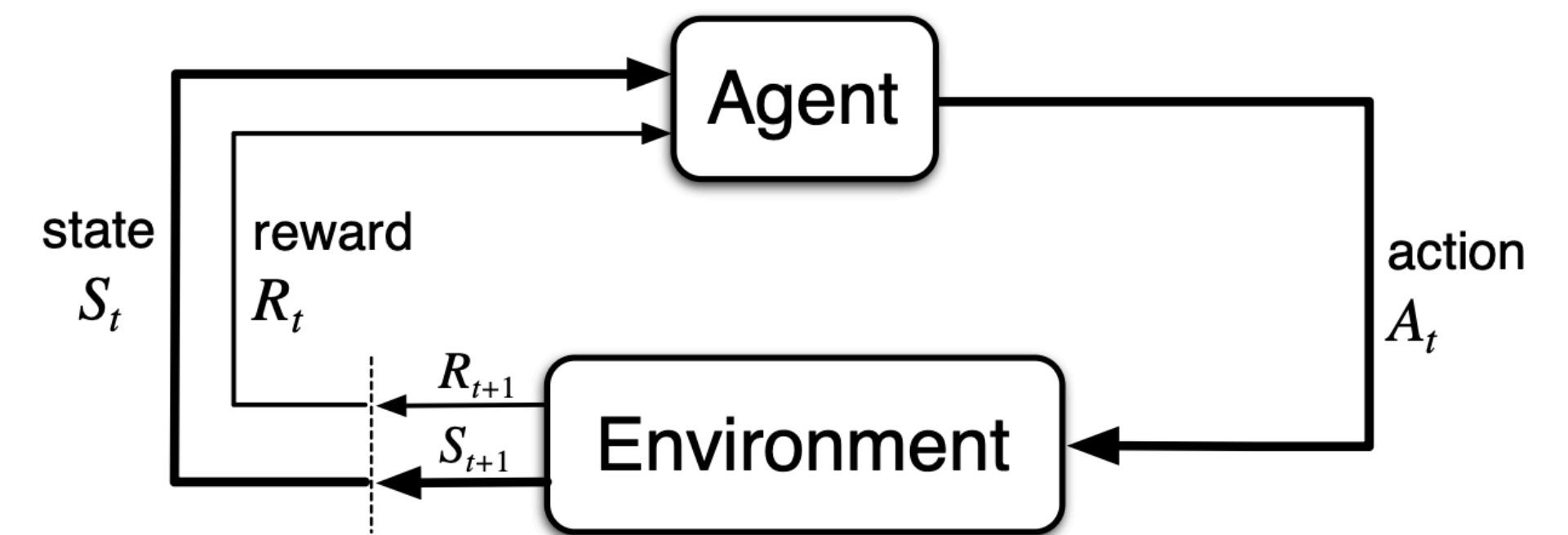
- Klopf's critique: By late 1970s, ML had narrowed to supervised learning (pattern matching with labels). Missing: the hedonic drive to achieve goals, not just match targets.
- The hedonistic neuron (1982): Neurons aren't homeostats (maintaining equilibrium)—they're heterostats (maximizing reward). Intelligence is about optimizing, not stabilizing.
- Sutton & Barto's formalization: Evaluative feedback ("that was good/bad") vs. instructive feedback ("you should have done X"). "The world doesn't tell you what to do; you learn from experience."
- Pole-balancing demo (1983): Actor-critic architecture—ASE chooses actions, ACE learns to predict future reward, converting sparse failure signals into dense internal signal.

A. Harry Klopf. The hedonistic neuron: a theory of memory, learning, and intelligence. Hemisphere Publishing Corporation, 1982.

Andrew G. Barto, Richard S. Sutton, and Charles W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. IEEE Transactions on Systems, Man, and Cybernetics, SMC-13(5):834-846, 1983.

Reinforcement learning

- Instead of adopting a (static) supervised learning paradigm, reinforcement learning considers an agent taking actions in a dynamic environment and observing rewards.
- Agent interacts with the environment to (a) explore, gathering information about how to act, and (b) exploit, using information to make rewarding decisions.
- How an agent uses information can be implemented using machine learning and optimization techniques.



The agent-environment interaction in a Markov decision process. [Sutton and Barto 2020]

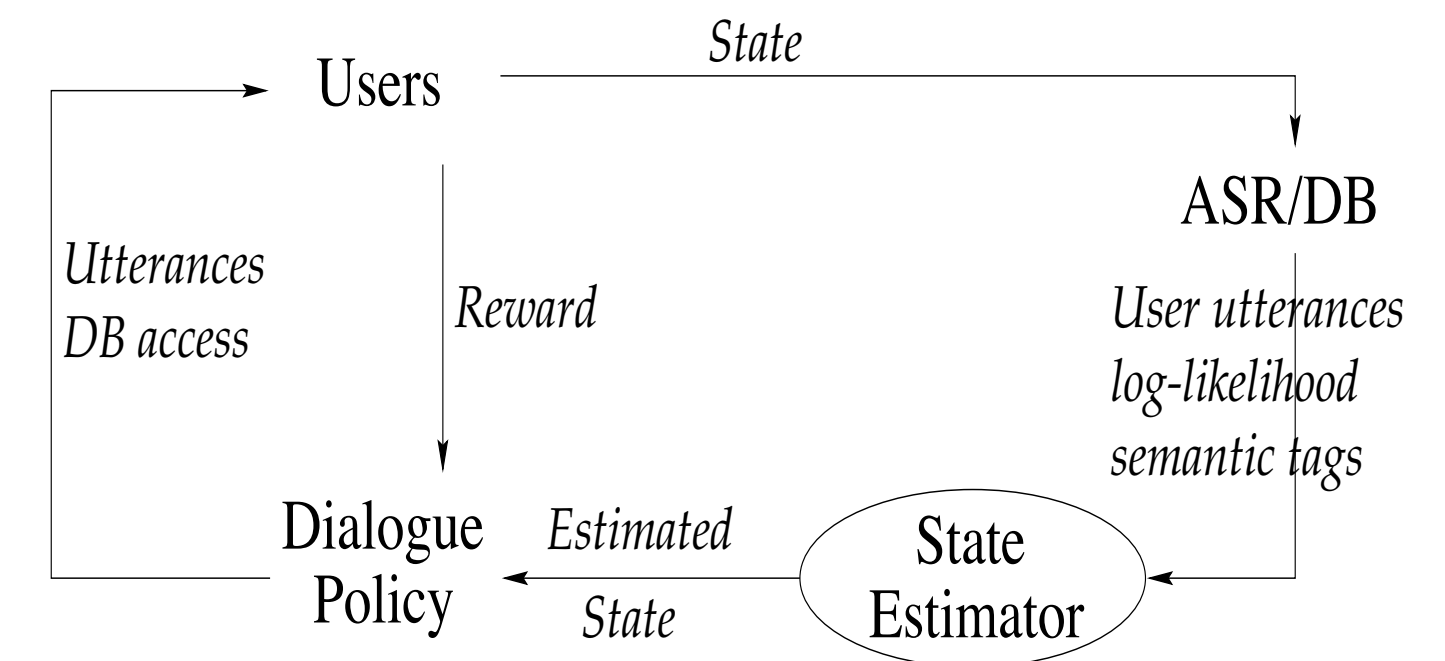
Reinforcement learning

Dialogue management

- Hand-designed dialogue policies are costly to iterate and explore limited options. RL enables automatic optimization over 2^{42} policies from user interaction data.
- MDP Formulation:
 - **States (62)**: Dialogue context—current attribute, ASR confidence, value status, retries, grammar, history
 - **Actions (42 choice-states)**: Initiative type (system/user/mixed) + confirmation strategy (explicit/none)
 - **Rewards**: Binary task completion (+1 exact match, -1 otherwise)
- Task completion of 52-64%. Outperformed hand-designed baselines. Discovered non-obvious strategy: start user initiative, back off to mixed/system when reasking, with attribute-specific confirmation thresholds.



Abstracted dialogue system. [Singh *et al.* 2002]



A dialogue system viewed as an MDP. [Singh *et al.* 2002]

Reinforcement learning

Recommendation and scaling RL

- Yahoo! news recommendation (2007-2012): Personalized article selection as a contextual bandit problem, single step RL.
 - User features as context, articles as arms, clicks as rewards. Deployed at web scale with LinUCB algorithm.
 - Extended exploration-exploitation tradeoffs to high-dimensional feature spaces. Enabled real-time learning from millions of user interactions per day.
- Simplified RL (no long-term planning, immediate rewards) made deployment practical. Bandits demonstrated RL could work in production systems with massive state spaces and real-world noise.
- Shifted recommendation systems from offline supervised learning to online learning with exploration. A/B testing became adaptive experimentation. Established RL as viable for consumer-facing products.



snapshot of the “Featured” tab in the Today Module on Yahoo! Front Page. [Li *et al.* 2010]

Sandeep Pandey, Deepak Agarwal, Deepayan Chakrabarti, and Vanja Josifovski. Bandits for taxonomies: a model-based approach. In SDM, 2007.

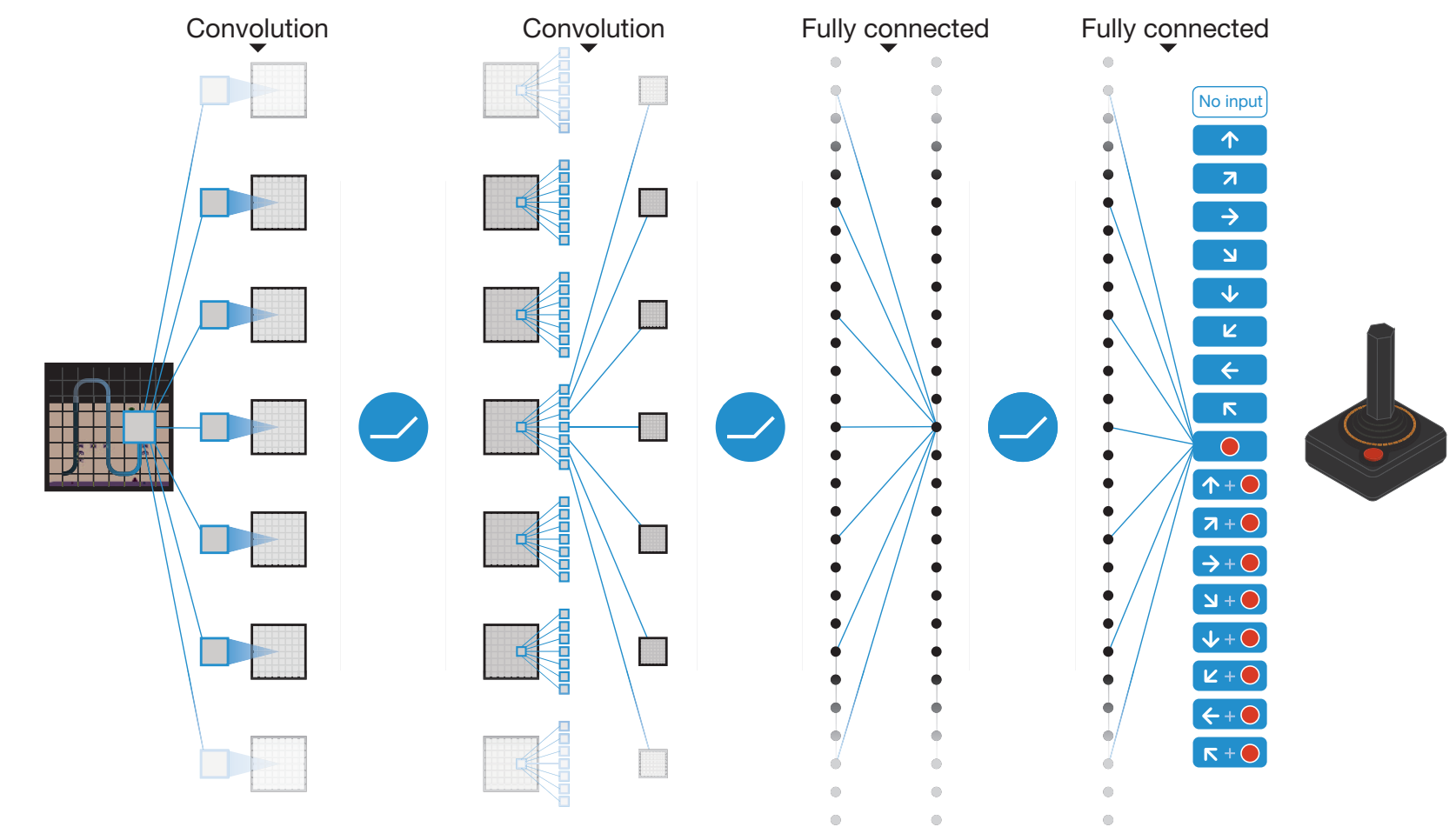
Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In Proceedings of the world wide web conference, 2010.

Miroslav Dudik, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In Proceedings of the 28th international conference on machine learning, icml 2011, 1097-1104, 2011.

Reinforcement learning

Neural network as a function approximation

- Convolutional neural networks as Q-function approximators, learning directly from raw pixels. Single architecture mastered 49 different games from the same algorithm.
 - Combined Q-learning (Watkins 1989) with deep CNNs, experience replay (Lin 1992), and target networks. Solved the deadly triad—function approximation, bootstrapping, and off-policy learning could now coexist.
- Hedonistic neuron's reward maximization, evaluative feedback from experience, and actor-critic separation (here: ϵ -greedy policy + learned Q-function). Scaled what Sutton & Barto formalized.
- Proved end-to-end learning was possible—no hand-crafted features, no domain-specific engineering. RL + deep learning became inseparable. Launched modern deep RL era.



Deep RL architecture [Mnih *et al.* 2015]

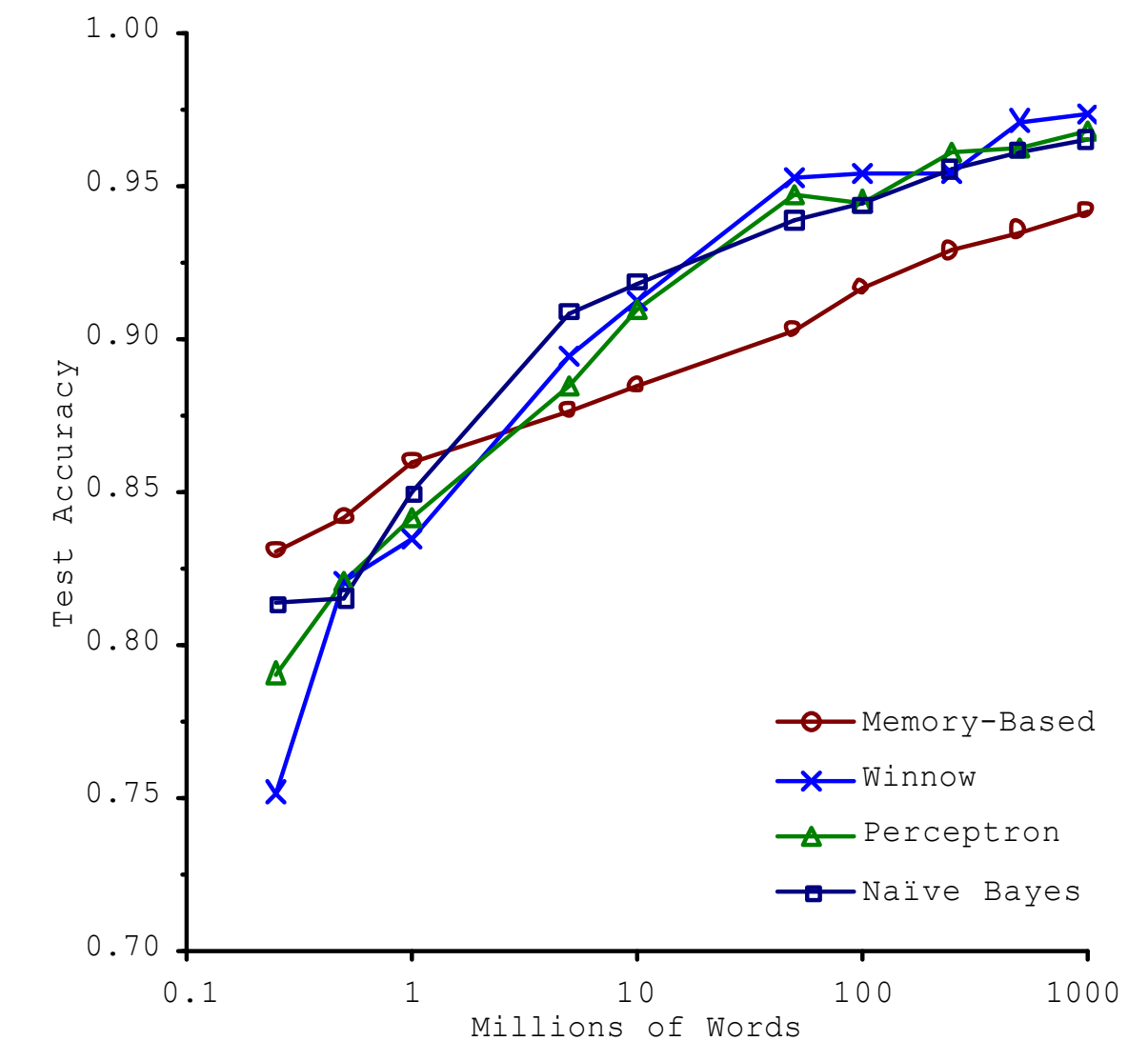
4. data

Data

- Statistical and machine learning approaches thrive on data. A parallel history of accessible datasets also set the stage for LLMs.
- **Early Days (1960s-1990s).** Hundreds to thousands of sentences. Brown Corpus (1961): ~1M words, a landmark that remained standard for decades. Heavy feature engineering necessary due to limited data.
- **The LDC Era (1990s-2000s).** Penn Treebank, TIMIT, parallel corpora. Scale: millions of words with quality annotation. Predominantly English with limited expansion to major languages.
- **News and Gigaword (2000s).** Reuters, Gigaword: billions of words of newswire text. Enabled larger-scale language modeling but limited diversity, formal, journalistic English dominated.

Data

- **The Web Era (2000s-2010s).** Common Crawl, Wikipedia: trillions of words across diverse genres and topics. Hundreds of languages included, though English-dominated. Quality highly variable = new challenges with noise and bias.
- **Modern Era (2020s).** Hundreds of billions to trillions of tokens. Multimodal (text, images, code) and multi-domain. Focus shifted from quantity to curation, filtering, and addressing representation gaps for low-resource languages.



disambiguation performance versus data. [Banko and Brill 2002]

5. funding

Role of the government funding

- **World War II.** Military need for code-breaking and machine translation spurred early computer science and AI research.
- **Scholarship.** Science and research leveraged for Cold War initiatives. Machine translation to make non-English research publications assessable; information retrieval to support large research corpora.
- **Intelligence.** Intelligence Advanced Research Projects Activity (IARPA) funds NLP for intelligence applications; NIST standardized evaluations (Message Understanding Conferences, ACE); TREC benchmarks drove information retrieval and question-answering research.

Role of the government funding

“At first thought it may seem curious that it was a Bell Telephone Laboratories group which came forward with new ideas and techniques to apply to the [anti-aircraft] problems. But for two reasons this was natural. First, this group not only had long and highly expert experience with a wide variety of electrical techniques. Second, there are surprisingly close and valid analogies between the fire control prediction problem and certain basic problems in communications engineering.”

—Warren Weaver, former head of research on control systems at the National Defense Research Committee, 1945

Role of the government funding

“Charles Wayne, starting a new speech-recognition program at DARPA in 1986, adopted the idea of quantitative comparison of alternative algorithms on a fixed task. In the context of this program, the formal quantitative competition was not only among algorithms but among research groups, with all the sites involved in the project sharing a predefined automatic evaluation metric and a body of material for training and testing...As a result of this demonstrated success, DARPA’s investment in Human Language Technology continued, and grew to include text retrieval, information extraction, topic detection and tracking, summarization, machine translation, and many other problems.”

Role of the government funding

“IR tools are ubiquitous today, but in the early 1990s IR as a field was relatively immature with limited ongoing research. Around that time, Donna Harman, a NIST IR researcher, was assisting the Defense Advanced Research Projects Agency (DARPA) with its TIPSTER IR research program; Harman saw an opportunity to radically improve IR research by developing a NIST program that would leverage TIPSTER’s accomplishments to provide new data and standard evaluation methodologies to IR researchers and create an objective competition among IR system creators...

[F]or every \$1 that NIST and its partners invested in TREC, at least \$3.35 to \$5.07 in benefits accrued to IR researchers. The internal rate of return (IRR) was estimated to be over 250% for extrapolated benefits and over 130% for unextrapolated benefits.”

Role of the government funding

“The Image Group conducted research for nearly a decade in areas of Optical Character Recognition (OCR) over the period of 1989 through 1998. Much of the focus at that time was on Hand Print Recognition from forms, which also included elements of Document Image Understanding. The Image Group's work in this area was in partnership with the IRS and the Bureau of Census, having direct impact and support to tax modernization efforts and the 1990 and 2000 decennial censuses. Advances in OCR were spurred on over this period as a result of increasing computation power of computers and from break-throughs in new generations of neural network and machine learning algorithms.”

convergence

Convergence

- Although each thread of this history can be seen independently, they also interacted
 - data + deep neural networks + information theory $\rightarrow$ semantic representation learning (neural probabilistic language model)
 - deep neural networks + RL $\rightarrow$ deep RL (Atari, AlphaGo, etc.)
 - data + (simple) RL $\rightarrow$ contextual bandits
- Large language models are arose from the convergence of all of these trends.

This course

- The first part of this course will cover contemporary methods that touch on these foundations
 - important to understand how these models are **designed**
- The second part will cover how modern LLMs are integrated into applications.
 - important to understand how these models are **applied**