Natural Language Processing

Class 9: Text Generation Tasks

Adam Faulkner

November 3, 2025

1. Machine Translation
2. Dialogue
3. Summarization
4. Paraphrase Generation

Era 1: Foundations and Initial Hype (1940s–1966)

- **1949: Warren Weaver's Memorandum** (Rockefeller Foundation). Suggested using computers, information theory, and cryptography for translation.
- **1954: Georgetown-IBM Experiment**. The first public demonstration of a system translating 49 Russian sentences into English.
- **Approach:** Initial Direct MT (word-for-word, dictionary lookup), shifting to more sophisticated Rule-Based MT (RBMT) focused on syntax.
- **Focus:** Cold War demand, primarily Russian-English for scientific/intelligence documents.
- **1966: ALPAC Report (Automatic Language Processing Advisory Committee)**. Concluded that MT was slower, less accurate, and more expensive than human translation.
    - **Impact:** Led to drastic funding cuts and triggered the **First AI Winter** in the field of MT research.

Era 2: Survival and Rule-Based Systems (1970s–1990s)

- **1970s:** The **RBMT** approach persists and finds commercial application in specific domains (e.g., restricted vocabulary).
- **1974-1980: First AI Winter** (broader impact, but MT already hit by ALPAC). Research continues at a lower, more realistic pace.
- **1980s:** Emergence of Example-Based MT (EBMT) (proposed by Makoto Nagao, 1984), using large corpora of past translations.
- **1987-Early 1990s: Second AI Winter** MT saw slow but steady commercial growth outside of pure academic AI.

Era 3: Statistical MT and The Internet (1990s–2015)

- **Early 1990s:** Emergence of **Statistical Machine Translation (SMT)**.
  - **Labs/Companies:** Pioneered by a team at IBM's Thomas J. Watson Research Center (e.g., the *Candide* system).
  - **Key Insight:** Translation is treated as a decoding problem, using statistical models to find the most probable target sentence given the source sentence ($\arg\max_T P(T|S)$).
- **1997: AltaVista Babel Fish** (using SYSTRAN) launches, bringing free, online translation to the masses.
- **2006: Google Translate** launches, initially using SMT and leveraging massive bilingual corpora from the web.
- **2007: MOSES** (open-source SMT engine) is released, democratizing research and development.
- SMT offered a major accuracy improvement over RBMT but still struggled with word reordering and grammatical fluency.

The Deep Learning Revolution (2016–Present)

- **2016: Neural Machine Translation (NMT)** takes over.
  - **Key Development: Google Brain Team** (and others) introduce an NMT system built on recurrent neural networks (RNNs), providing a massive jump in translation quality.
  - **Impact:** NMT systems translate entire sentences at once, capturing context and producing much more fluent, human-sounding output.
- **2017: The Transformer Architecture** (Google's *Attention Is All You Need* paper) is introduced.
  - **Impact:** The Transformer becomes the foundational model for modern NMT, **Large Language Models (LLMs)**, and Generative AI.
- **Present Day:**
  - Technology: MT is now a sub-field of Generative AI and LLMs (e.g., using models like GPT-4, Gemini) for more context-aware, multimodal, and adaptable translation.

Why MT is hard

- Words of the **target language** don't necessarily agree with the words of the **source language** in number or order

English:	*He wrote a letter to a friend*		
Japanese:	*tomodachi*	*ni tegami-o*	*kaita*
	friend	to letter	wrote

- Elements of the sentences are in very different places – in English, the verb is in the middle of the sentence, while in Japanese the verb *kaita* comes at the end.
- Japanese sentence doesn't require the pronoun *he*, while English does.

Why MT is hard

- Differences become more marked across language families.
- English: **Indo-European language family (West Germanic branch)**,
- Chinese: **Sino-Tibetan language family**
- In the following actual sentence from the United Nations, notice the many changes between the Chinese sentence and its English equivalent

大会/General Assembly 在/on 1982年/1982 12月/December 10日/10 通过了/adopted 第37号/37th 决议/resolution， 核准了/approved 第二次/second 探索/exploration 及/and 和平peaceful 利用/using 外层空间/outer space 会议/conference 的/of 各项/various 建议/suggestions 。

On 10 December 1982 , the General Assembly adopted resolution 37 in which it endorsed the recommendations of the Second United Nations Conference on the Exploration and Peaceful Uses of Outer Space .

Commonalities and differences across languages

- 7~000 languages
- **Universals:** Many universals arise from the functional role of language as a communicative system by humans.
  - Referring to people, eating, drinking
  - Registers: politeness/non-politeness
  - Nouns, verbs, commands, questions
- **Differences**: Word order often differs across languages
  - *Subject-Verb-Object:* German, French, English, Mandarin
  - *Subject-Object-Verb*: Hindi, Japanese
  - *Verb-Subject-Object:* Irish, Arabic

Commonalities and differences across languages

- **Morphological** (word structure) differences across languages
  - **Isolating** languages: English, Vietnamese, Cantonese. Little to no morphology
  - **Polysynthetic** languages: Siberian Yupik ("Eskimo"), in which a single word may have many morphemes, corresponding to a whole sentence in English.
    - **English**: *He said he would probably go.*
    - **Inuktitut**: *Ayagciqsugnarqnillruuq* ayag- (go) + -ciq- (future) + -sugnarqe- (probably) + -ni- (say) + -llru- (past) + -u- (indicative) + -q (3rd person singular)

  Most languages lay somewhere in the middle between these two extremes

Machine Translation: The Encoder-Decoder Model

- The standard architecture for MT is the **Encoder-Decoder Transformer** (or sequence-to-sequence model).
- This architecture takes a sequence of input tokens and generates a sequence of output tokens.

The Task

Given a source sentence, generate a corresponding sentence in the target language.

- **Source (English):** *The green witch arrived*
- **Target (Spanish):** *Llegó la bruja verde*

Supervised Training of Encoder-Decoder-style MT

- **Method:** MT uses supervised machine learning.
- **Training Data:** A large set of **parallel sentences** is provided, matching source sentences with target sentences.
- **Input/Output Representation:** Sentences are split into sequences of **subword tokens** $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_m)$.

The Goal

The system is trained to maximize the probability of the target sequence given the source sequence:

$$P(y_1, \ldots, y_m | x_1, \ldots, x_n)$$

Encoder-Decoder Components

The architecture separates the input processing from the output generation.

1. The Encoder

- Takes the input tokens $x = [x_1, ..., x_n]$
- Produces an intermediate, dense representation called the **context** (h).

$$\mathbf{h} = \text{encoder}(\mathbf{x})$$

Encoder-Decoder Components

The architecture separates the input processing from the output generation.

1. The Encoder

- Takes the input tokens $x = [x_1, ..., x_n]$
- Produces an intermediate, dense representation called the **context** (h).

$$\mathbf{h} = \text{encoder}(\mathbf{x})$$

2. The Decoder

- Takes the context h.
- Generates the output **y** token by token, conditioning on previously generated tokens.

$$y_{t+1} = \text{decoder}(\mathbf{h}, y_1, ..., y_t) \quad \forall t \in [1, ..., m] \quad \text{(Eq. 12.9)}$$

Tokenization: Beyond Words

- MT systems require a vocabulary fixed in advance.
- Traditional space-separated words are insufficient due to:
  1. **Vocabulary Size:** Handling all possible word forms.
  2. **Language Differences:** Handling languages with (English) and without (Chinese, Thai) clear word separation.

Tokenization: Beyond Words

- MT systems require a vocabulary fixed in advance.
- Traditional space-separated words are insufficient due to:
  1. **Vocabulary Size:** Handling all possible word forms.
  2. **Language Differences:** Handling languages with (English) and without (Chinese, Thai) clear word separation.
- **Solution: Subword Tokenization**
- Algorithms like BPE (Byte-Pair Encoding) and its variants create tokens that can be words, subwords, or characters.
- A **shared vocabulary** for source and target languages simplifies the process and aids in copying entities (like names).

The Wordpiece Tokenization Algorithm

- Used in many modern systems (e.g., BERT, Google MT).
- **Difference from BPE:** Instead of merging the most frequent pairs, Wordpiece chooses merges based on which one **most increases the language model probability** of the tokenization.

The Wordpiece Tokenization Algorithm

- Used in many modern systems (e.g., BERT, Google MT).
- **Difference from BPE:** Instead of merging the most frequent pairs, Wordpiece chooses merges based on which one **most increases the language model probability** of the tokenization.

Example (Wu et al., 2016):

- **Words:** `Jet makers feud over seat width with big orders at stake`
- **Wordpieces:** `J et makers fe ud over seat width with big orders at stake`

The Wordpiece Tokenization Algorithm

- Used in many modern systems (e.g., BERT, Google MT).
- **Difference from BPE:** Instead of merging the most frequent pairs, Wordpiece chooses merges based on which one **most increases the language model probability** of the tokenization.

Example (Wu et al., 2016):

- **Words:** `Jet makers feud over seat width with big orders at stake`
- **Wordpieces:** `J et makers fe ud over seat width with big orders at stake`

Core Idea (Simplified Steps):

1. Initialize lexicon with individual characters.
2. Repeatedly train an n-gram language model on the training corpus.
3. Choose the concatenation of two existing wordpieces that offers the maximum increase in the language model probability of the corpus.
4. Repeat until the desired vocabulary size (V) is reached.

Acquiring Training Data: Parallel Corpora

- MT models are trained on a **parallel corpus**
- Parallel corpora contain two or more languages, typically sentence-aligned

Acquiring Training Data: Parallel Corpora

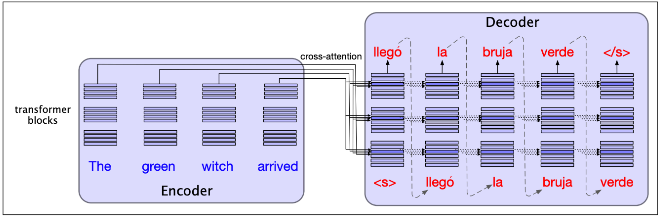
- MT models are trained on a **parallel corpus**
- Parallel corpora contain two or more languages, typically sentence-aligned
- **Sources of Large Corpora:**
  1. **Governmental / Institutional Proceedings:** Texts that must be legally translated into multiple official languages.
  2. **Public Domain Translations:** Classic literature, religious texts.

Acquiring Training Data: Parallel Corpora

- MT models are trained on a **parallel corpus**
- Parallel corpora contain two or more languages, typically sentence-aligned
- **Sources of Large Corpora:**
  1. **Governmental / Institutional Proceedings:** Texts that must be legally translated into multiple official languages.
  2. **Public Domain Translations:** Classic literature, religious texts.
- **Examples:**
  - **Europarl Corpus:** Proceedings of the European Parliament, containing 400k–2M sentences each from 21 European languages.
  - **United Nations Parallel Corpus:** Approximately 10 million sentences in the six official UN languages (Arabic, Chinese, English, French, Russian, Spanish).

Standard MT Architecture: Encoder-Decoder Transformer

- The standard architecture for Machine Translation (MT) is the **encoder-decoder transformer**.
- It is composed of two transformers:
  1. **Encoder**: Same as the basic transformer from Class 5 (Maps source tokens $X = x_1, ..., x_n$ to representation $H_{enc} = h_1, ..., h_n$).
  2. **Decoder**: An augmented transformer that is essentially a **conditional language model**.
- The decoder generates target words one by one, conditioning on the source sentence and previously generated target words.



Decoder Augmentation: The Cross-Attention Layer

- The main difference in the decoder transformer block is an extra layer: the **cross-attention layer**.
- The decoder block structure:
  1. Multi-Head **Causal Self-Attention** (masked, attends to prior decoder output)
  2. Layer Norm
  3. Multi-Head **Cross-Attention** (new layer)
  4. Layer Norm
  5. Feed Forward Layer
  6. Layer Norm
- **Cross-attention** (or encoder-decoder attention) allows the decoder to **attend to the encoder's output** (H_{enc}).

Mechanics of Cross-Attention

- Cross-attention uses:
  - **Query (Q)**: Comes from the output of the prior layer of the **decoder** ($H_{dec}^{[-1]}$).
  - **Key (K)** and **Value (V)**: Come from the final output of the **encoder** (H_{enc}).
- **Formulas**:

$$Q = H_{dec}^{[-1]} W_Q$$

$$K = H_{enc} W_K$$

$$V = H_{enc} W_V$$

- **Attention Mechanism**:

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

- This links the decoder's current state (Q) with the entire source representation (K and V).

Mechanics of Cross-Attention

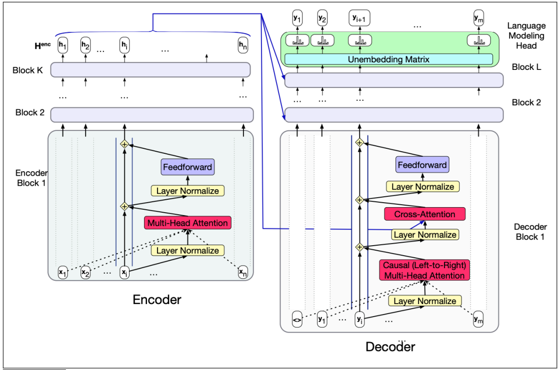


Figure 1: The transformer block for the encoder and the decoder, showing the residual stream view. The final output of the encoder $H^{enc} = h_1, ..., h_n$ is the context used in the decoder. The decoder is a standard transformer except with one extra layer, the **cross-attention layer**, which takes that encoder output H^{enc} and uses it to form its K and V inputs.

Training and Decoding the MT Model

- Uses the same **self-supervision model** as encoder-decoder RNNs.
- The network is trained **autoregressively** to predict the next token.
- **Loss Function**: **Cross-Entropy Loss** L_{CE} on the predicted next word probability:

$$L_{CE}(\hat{y}_t, y_t) = -log\hat{y}_t[w_{t+1}]$$

- **Teacher Forcing**: At each timestep, the **gold target token** from the training set is used as the next input, instead of the decoder's own (possibly erroneous) output.

Decoding in MT: Beam Search

- **Greedy Decoding** Problem: The locally best choice at time t might lead to a poor sequence overall, as it chooses:

$$\hat{w}_t = \text{argmax}_{w \in \mathcal{V}} P(w|w_{<t})$$

- A problem with greedy decoding is that what looks high probability at word t might turn out to have been the wrong choice once we get to word t +1. The beam search algorithm maintains multiple choices until later when we can see which one is best.

Beam Search

- **Beam Search Algorithm**:
  - The most common decoding algorithm for MT.
  - Models decoding as a search through a **search tree** (branches are token generation, nodes are prefixes).
  - It maintains **multiple choices** (a "beam") at each step, keeping the k most probable prefixes.
  - This allows the algorithm to explore paths that might have a slightly lower probability initially but lead to a much better overall sequence probability later on.

Beam Search

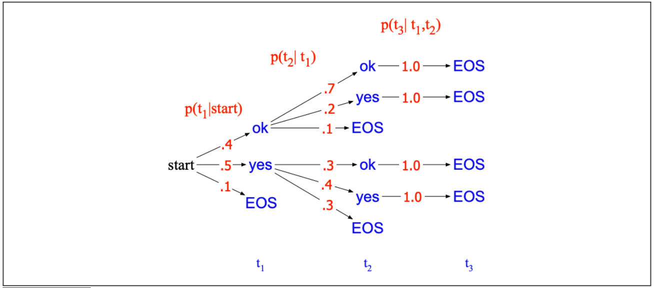


Figure 2: A search tree for generating the target string $T = t_1, t_2, \ldots$ from vocabulary $V = yes, ok, < s >$, showing the probability of generating each token from that state. Greedy search chooses *yes* followed by *yes*, instead of the globally most probable sequence *ok ok*.

Evaluation of MT systems: From BLEU to BERTScore

- The Bilingual Evaluation Understudy (BLEU) score is the most widely cited metric for evaluating Machine Translation (MT) quality.
- It is computed for a corpus of candidate translation sentences.
- The final score is a function of:
  1. **Modified *n*-gram Precision** (up to 4-grams).
  2. A **Brevity Penalty** (BP) to penalize short translations.
- The *n*-gram precision and brevity penalty are computed over the corpus as a whole.

N-gram Precision

- **Definition**: The percentage of n-gram tokens in the candidate translation that also occur in the reference translation.
- **Corpus-Level Computation**:
  - **Numerator**: Sum over all sentences of the counts of all n-gram types that also occur in the reference translation (clipped).
  - **Denominator**: Total count of all n-grams in all candidate sentences.
- We compute this precision for unigrams ($n = 1$), bigrams ($n = 2$), trigrams ($n = 3$), and 4-grams ($n = 4$).
- The precisions are combined using the Geometric Mean:

$$\text{BLEU} \propto \exp\left(\sum_{n=1}^{4} w_n \log p_n\right)$$

(where p_n is the n-gram precision and w_n is the weight, typically $\frac{1}{4}$)

Limitations

- **Tokenization Sensitivity**: As a word-based metric, BLEU is highly sensitive to the word tokenization standard. Comparing systems with different tokenizers is impossible.
- **Morphological Complexity**: Does not work as well in languages with complex morphology (e.g., highly inflected languages).
- **Continued Use**: Despite limitations, BLEU is still sometimes used for evaluation, particularly for translation **into English**.

BERTScore: Measuring Similarity with Embeddings

- Core Idea: Measure the similarity between a reference sentence ($\mathbf{x}$) and a candidate translation ($\tilde{\mathbf{x}}$) by comparing their **token embeddings**.
- Process:
  1. Pass the reference $\mathbf{x}$ and candidate $\tilde{\mathbf{x}}$ through a pre-trained contextual embedding model, such as BERT.
  2. Compute a BERT embedding for each token (x_i and $\tilde{x}_j$).
- Token Scoring: The similarity between any pair of tokens $(x_i, \tilde{x}_j)$ is scored using cosine similarity:

$$\text{similarity}(x_i, \tilde{x}_j) = \frac{x_i \cdot \tilde{x}_j}{|x_i||\tilde{x}_j|}$$

BERTScore Metrics: Recall and Precision

- BERTScore provides a final score based on Precision, Recall, and the F_1 score.
- Matching Strategy: Tokens are greedily matched to the most similar token in the corresponding sentence.

1. Recall ($\mathbf{R_{BERT}}$): How well the reference tokens ($\mathbf{x}$) are covered by the candidate tokens ($\tilde{\mathbf{x}}$).

- Each token $x_i \in \mathbf{x}$ is matched to the most similar token in $\tilde{\mathbf{x}}$.

$$R_{\text{BERT}} = \frac{1}{|\mathbf{x}|} \sum_{x_i \in \mathbf{x}} \max_{\tilde{x}_j \in \tilde{\mathbf{x}}} \frac{x_i \cdot \tilde{x}_j}{|x_i||\tilde{x}_j|}$$

2. Precision ($\mathbf{P_{BERT}}$): How well the candidate tokens ($\tilde{\mathbf{x}}$) are supported by the reference tokens ($\mathbf{x}$).

- Each token $\tilde{x}_j \in \tilde{\mathbf{x}}$ is matched to the most similar token in $\mathbf{x}$.

$$P_{\text{BERT}} = \frac{1}{|\tilde{\mathbf{x}}|} \sum_{\tilde{x}_j \in \tilde{\mathbf{x}}} \max_{x_i \in \mathbf{x}} \frac{x_i \cdot \tilde{x}_j}{|x_i||\tilde{x}_j|}$$

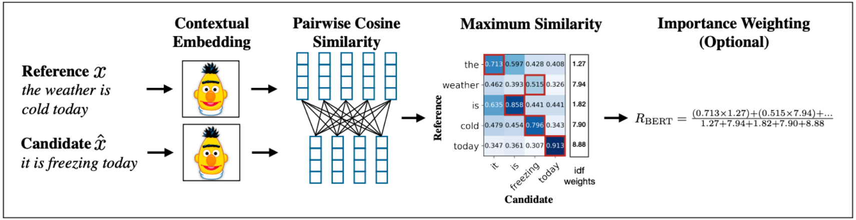
Final BERTScore

- The final BERTScore (typically F_1) is the harmonic mean of Precision and Recall:

$$F_1 = 2 \cdot \frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}$$

- Key Advantage: By using contextual embeddings (like BERT), the metric captures **semantic similarity** beyond exact word or n-gram matching (unlike BLEU).
- BERTScore provides a more nuanced measure of translation quality by utilizing deep contextual representations.

BertScore for MT evaluation



Early Rule-Based Systems (1960s-1970s)

- Focus: Pattern matching and predefined rules.
- ELIZA (1966):
  - Developed by Joseph Weizenbaum at MIT.
  - Simulated a Rogerian psychotherapist.
  - Relied on keyword matching and simple rephrasing rules.
  - Limitation: No real understanding; simply reflected input.
  - Example: "I am sad." → "Why are you sad?"
- SHRDLU (1970s):
  - Developed by Terry Winograd.
  - Operated in a "blocks world" micro-domain.
  - Could understand and respond to natural language commands and questions within its limited world.
  - Advantage: More sophisticated understanding within its domain.
  - Limitation: Brittle outside its highly constrained environment.

Knowledge-Based Systems (1970s-1990s)

- Focus: Structured knowledge representation (ontologies, databases, expert systems).
- Systems like MYCIN:
  - Expert system for diagnosing infectious diseases.
  - Used if-then rules and a vast knowledge base.
  - Could explain its reasoning.
- TREC QA Track (1999 onwards):
  - The Text REtrieval Conference (TREC) introduced a QA track.
  - Prompted research into retrieving answers from large text collections.
  - Shifted focus from pre-defined knowledge to information retrieval and extraction.
- Limitations:
  - Manual knowledge engineering was expensive and time-consuming.
  - Difficulty scaling to open domains.
  - Still relied heavily on symbolic reasoning, less on statistical patterns.

Statistical Methods and Deep Learning (2000s-2010s)

- Statistical QA:
  - Emergence of techniques like Hidden Markov Models (HMMs) and Support Vector Machines (SVMs) for tasks like Named Entity Recognition and Relation Extraction.
  - Focused on identifying answer spans within documents based on statistical features.
- Deep Learning Revolution (mid-2010s onwards):
  - Word Embeddings (Word2Vec, GloVe): Enabled capture of semantic relationships between words.
  - Recurrent Neural Networks (RNNs) LSTMs: Improved modeling of sequential data.
  - Attention Mechanism: Allowed models to focus on relevant parts of the input.
  - Datasets: Creation of large-scale QA datasets like SQuAD (Stanford Question Answering Dataset) accelerated progress. Models learned to extract answers from given passages.
- IBM Watson (2011):
  - Won Jeopardy!, demonstrating advanced NLP capabilities including QA.
  - Combined deep analytics with sophisticated information retrieval.

Transformers and Large Language Models (Late 2010s-Present)

- Pre-trained Language Models (PLMs):
  - BERT: Fine-tuned for QA, achieved state-of-the-art results on SQuAD.
  - GPT-series (Generative Pre-trained Transformer): Demonstrated strong generative capabilities.
- Large Language Models (LLMs):
  - Zero-shot and Few-shot QA: Can answer questions without specific fine-tuning or with minimal examples.
  - Conversational QA: Support multi-turn dialogues, maintaining context.
  - Reasoning: Show surprising capabilities in complex reasoning, summarization, and creative text generation for answers.

What is a Conversational Agent?

Definition

A **Dialogue System** (or **Conversational Agent**) is a program that communicates with users in natural language (text, speech, or both).

- These systems facilitate human-computer interaction by understanding and generating human language.
- Conversational agents fall into two primary classes, based on their goal:

What is a Conversational Agent?

Definition

A **Dialogue System** (or **Conversational Agent**) is a program that communicates with users in natural language (text, speech, or both).

- These systems facilitate human-computer interaction by understanding and generating human language.
- Conversational agents fall into two primary classes, based on their goal:
  1. **Task-Oriented Dialogue Agents**
  2. **Chatbots**

Task-Oriented Dialogue Agents

Core Function

- Use conversation to help the user **complete specific, defined tasks**.

Examples in Digital Assistants

- Siri, Alexa, Google Now/Home, Cortana.
- Functions include:
  - Giving directions.
  - Controlling smart appliances.
  - Finding restaurants or making calls.

Wider Applications

- Answering questions on corporate websites.
- Interfacing with robots.
- Social Good Initiatives:
  - Examples like **DoNotPay** (a "robot lawyer").
  - Tasks: Challenging incorrect parking fines, applying for emergency housing, or claiming asylum.

Class 2: Chatbots

Definition

Chatbots are systems designed for **extended, unstructured conversations** ('chats').

- **Primary Goal:** Mimic the free-flowing, unstructured characteristic of human-human interaction.
- **Purpose:** Mainly for entertainment or social engagement.

Class 2: Chatbots

Definition

Chatbots are systems designed for **extended, unstructured conversations** ('chats').

- **Primary Goal:** Mimic the free-flowing, unstructured characteristic of human-human interaction.
- **Purpose:** Mainly for entertainment or social engagement.
- **Secondary Goal:** They can also be used to make task-oriented agents more natural and conversational, enhancing the user experience beyond simple utility.

Introduction to Corpus-based Chatbots

- Chatbots that mine conversations of human-human interactions, rather than using hand-built rules.
- Data-Intensive: Require hundreds of millions or even billions of words for training
- Systems typically generate a single response turn appropriate for the entire conversation so far

Training Datasets

- **Natural Spoken Conversational Corpora:**
  - Switchboard corpus (American English telephone conversations).
  - CALLHOME and CALLFRIEND (various languages).
- **Movie Dialogue:**
  - Resembles natural conversation in many ways (Forchini, 2013).
  - (Danescu-Niculescu-Mizil and Lee 2011, Lison and Tiedemann 2016).
- **Crowdsourced Datasets:**
  - Created specifically for dialogue systems (e.g., workers take on personas).
  - **Topical-Chat:** 11K crowdsourced conversations spanning 8 broad topics.

Response Generation Methods

Most corpus-based chatbots produce responses using one of two primary methods:

- **Retrieval Methods:** Use Information Retrieval (IR) to grab an appropriate response from a corpus.
- **Generation Methods:** Use a language model or encoder-decoder to generate a novel response.

These algorithms draw on techniques from Question Answering Systems, which also focus on single, context-appropriate responses.

Response by Retrieval (IR-based)

- The user's turn **q** is treated as a query.
- The system retrieves and repeats an appropriate turn **r** from a conversation corpus **C**.
- **Scoring Metric (Similarity):** Choose the **r** ∈ **C** most similar to **q**.
- **Classic IR (e.g., TF-IDF):**

$$\text{response}(q, C) = \text{argmax}_{r \in C} \frac{q \cdot r}{|q||r|}$$

- **Alternative:** Find the most similar turn **t** to **q**, and return the turn immediately following **t** as the response.

Response by Retrieval (Neural IR)

- Uses neural techniques, such as a bi-encoder model.
- Two separate encoders are trained: BERT_Q for the query and BERT_R for the candidate response.
- **Encodings (e.g., using [CLS] token):**

$$h_q = \text{BERT}_Q(q)\,[\text{CLS}]$$

$$h_r = \text{BERT}_R(r)\,[\text{CLS}]$$

- **Scoring:** The dot product $\mathbf{h_q} \cdot \mathbf{h_r}$ serves as the similarity score.
- **Neural Response Selection:**

$$\text{response}(q, C) = \text{argmax}_{r \in C}\, h_q \cdot h_r$$

- **Extensions:** Can use longer conversational context, more sophisticated architectures, or incorporate user/sentiment information.

Response by Generation

- Response production is modeled as an encoder-decoder task (transduction).
- The system learns to "translate" the user's prior turn into the system's turn (A machine learning version of ELIZA).
- Pioneered as a kind of translation task and generalized to encoder-decoder models
- The model generates each token $\mathbf{r_t}$ of the response by conditioning:

$$\hat{r}_t = \text{argmax}_{w \in V} P(w|q, r_1 \dots r_{t-1})$$

- The response is conditioned on the encoding of the entire query $\mathbf{q}$ and the response generated so far ($\mathbf{r_1 \dots r_{t-1}}$).

Response by Generation

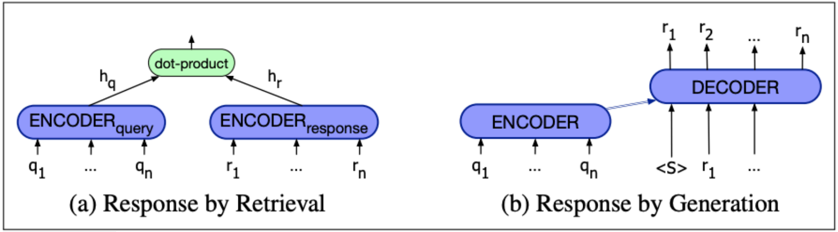


Figure 3: Two architectures for generating responses for a neural chatbot. In response by retrieval (a) we choose a response by finding the turn in the corpus whose encoding has the highest dot-product with the users turn. In response by generation (b) we use an encoder-decoder to generate the response.

Overview of Chatbot Evaluation

Chatbots are primarily evaluated by humans, who assign a score based on interaction or transcription.

1. **Participant Evaluation**
   - The human who *talked* directly to the chatbot assigns the score.
   - Focuses on immediate conversational experience.
2. **Observer Evaluation**
   - A *third party* reads a transcript of the human/chatbot conversation.
   - Allows for objective comparison across systems.

Participant Evaluation

In this common paradigm, the human evaluator interacts with the model:

- **Interaction Length:** Chat for a fixed number of turns (e.g., six turns).
- **Rating Scope:** Rates the chatbot on **8 dimensions** capturing conversational quality:
  - Making Sense, Fluency, Humanness
  - Avoiding Repetition, Listening, Inquisitiveness
  - Engagingness, Interestingness

Participant Evaluation: Dimension Examples

Engagingness

- *Question:* How much did you enjoy talking to this user?
- *Scale:* Not at all → A little → Somewhat → A lot

Avoiding Repetition

- *Question:* How repetitive was this user?
- *Scale:* Repeated themselves over and over → Sometimes said the same thing twice → Always said something new

Making Sense

- *Question:* How often did this user say something which did **NOT** make sense?
- *Scale:* Never made any sense → Most responses didn't make sense → Some responses didn't make sense → Everything made perfect sense

Observer Evaluation

Third-party annotators review complete conversation transcripts.

- **Turn-Level Scoring:**
  - Raters assign a score to *each system turn*.
  - Example: Marking the **coherence** of each turn
- **High-Level Scoring:**
  - Assigning a single, overall score to compare two systems (A vs. B).
  - The **acute-eval** metric is commonly used for this purpose.

The Acute-Eval Metric (Li et al., 2019)

Acute-eval is a high-agreement observer evaluation technique for comparative ranking.

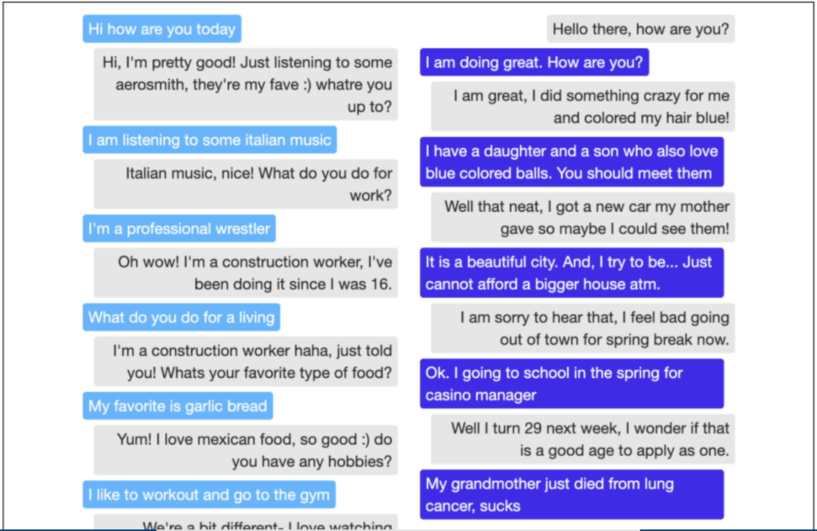
- **Process:** Annotators look at two separate human-computer conversations (Conversation A and Conversation B).
- **Goal:** Choose the dialogue system participant (A or B) that performed better overall.
- **Wording:** The questions are specifically worded to lead to high agreement among annotators.
- **4 Key Comparative Questions** are asked:
  1. Engagingness
  2. Interestingness
  3. Humanness
  4. Knowledgeable

Acute-Eval: The Comparative Questions

Annotators choose between Speaker A and Speaker B based on:

- **Engagingness:**
  - Who would you prefer to talk to for a long conversation?
- **Interestingness:**
  - If you had to say one of these speakers is interesting and one is boring, who would you say is more interesting?
- **Humanness:**
  - Which speaker sounds more human?
- **Knowledgeable:**
  - If you had to say that one speaker is more knowledgeable, who would you choose?

Acute-Eval


Hi how are you today
Hi, I'm pretty good! Just listening to some aerosmith, they're my fave :) whatre you up to?
I am listening to some italian music
Italian music, nice! What do you do for work?
I'm a professional wrestler
Oh wow! I'm a construction worker, I've been doing it since I was 16.
What do you do for a living
I'm a construction worker haha, just told you! Whats your favorite type of food?
My favorite is garlic bread
Yum! I love mexican food, so good :) do you have any hobbies?
I like to workout and go to the gym
We're a bit different- I love watching
Hello there, how are you?
I am doing great. How are you?
I am great, I did something crazy for me and colored my hair blue!
I have a daughter and a son who also love blue colored balls. You should meet them
Well that neat, I got a new car my mother gave so maybe I could see them!
It is a beautiful city. And, I try to be... Just cannot afford a bigger house atm.
I am sorry to hear that, I feel bad going out of town for spring break now.
Ok. I going to school in the spring for casino manager
Well I turn 29 next week, I wonder if that is a good age to apply as one.
My grandmother just died from lung cancer, sucks

Why Not Use Automatic Evaluation for Chatbots?

- Automatic evaluations (e.g., BLEU, ROUGE, Embedding Dot Products) are generally not used for chatbots.
- Reason: These computational measures correlate very poorly with human judgments of performance
- Why the poor performance?
  - The space of possible, valid responses in a dialogue turn is very large.
  - Simple word-overlap or semantic similarity metrics work best when the response space is small and lexically overlapping.
  - This is true for generation tasks like machine translation or summarization, but definitely not true for dialogue.

Absolute Task Success

- When the task is **unambiguous**, evaluation can be simple.
- Measure **absolute task success**:
    - Did the system book the right plane flight?
    - Did it put the correct event on the calendar?

User Satisfaction and Task Error Rate

User Satisfaction Rating

- Provides a **fine-grained** idea of user happiness.
- Users interact with the system, perform a task, and then complete a **questionnaire**.

Task Error Rate (Extrinsic Metric)

- A **less fine-grained**, but important, measure of success.
- Quantifies **how often** the correct task (e.g., meeting) was **not** completed correctly at the end of the interaction.

Efficiency Cost: Measuring System Efficiency

- Measures the system's **efficiency** at helping users.
- **Key Metrics:**
  - **Total Elapsed Time** for the dialogue (in seconds).
  - **Number of Turns** (total or system turns).
  - **Total Number of Queries** (Polifroni et al., 1992).
- **Error-Related Metrics:**
  - Number of system **non-responses**.
  - **Turn Correction Ratio**: $\frac{\text{System/User turns used solely to correct errors}}{\text{Total number of turns}}$

What is a Wizard-of-Oz System?

- A crucial tool for building and testing dialogue systems.
- Users interact with what they think is a **software agent**.
- In reality, it is a **human "wizard"** disguised by a software interface.

Origin of the Name

- Comes from the children's book/movie *The Wizard of Oz*
- In the story, the "wizard" was a **simulation** controlled by a man behind a curtain or screen.

System Architecture and Function

- A Wizard-of-Oz system allows testing an architecture **before full implementation**.
- Only the **interface software** and **databases** need to be in place.

The Wizard's Role

- Gets **input** from the user.
- Uses a graphical interface to run **sample queries** against a database.
- **Outputs sentences** to the user, either by typing them or by selecting/combining from a menu.

Wizard of Oz setup

1. Machine Translation
2. Dialogue
3. Summarization
4. Paraphrase Generation

Statistical and Feature-Based Extraction (Early 2000s)

- **Dominant Approach: Extractive Summarization**
  - Summary is created by selecting and concatenating key sentences from the source document (verbatim).
- **Techniques Employed:**
  - **Statistical Methods:** Sentence scoring based on features like **term frequency-inverse document frequency (TF-IDF)**, sentence location, and key phrase matching.
  - **Graph-Based Ranking:** Algorithms like **TextRank** and **LexRank** treat sentences as nodes in a graph and score them based on connectivity (similarity).
  - **Supervised Machine Learning:** Training classifiers (e.g., SVM, Naive Bayes) to label sentences as 'summary-worthy' or 'not summary-worthy' using hand-crafted features.
- **Limitation:** Produced summaries often **lacked fluency** and coherence due to disjointed sentences.

Extractive versus abstractive summarization

(a) Extractive Summarization

Source Text: Peter and Elizabeth took a taxi to attend the night party in the city.

While in the party, Elizabeth collapsed and was rushed to the hospital.

Summary: Peter and Elizabeth attend party city. Elizabeth rushed hospital.

(b) Abstractive Summarization

Source Text: Peter and Elizabeth took a taxi to attend the night party in the city.

While in the party, Elizabeth collapsed and was rushed to the hospital.

Summary: Elizabeth was hospitalized after attending a party with Peter.

4

Deep Learning and the Rise of Abstractive Methods (Mid-2010s)

- **Shift in Focus:** Renewed push toward **Abstractive Summarization**.
    - Goal: Generate **novel sentences** that paraphrase and synthesize the original content, mimicking human summarizers.
- **Key Architecture: Sequence-to-Sequence (Seq2Seq) Models**
    - Employed **Recurrent Neural Networks (RNNs)**, like LSTMs and GRUs, in an **Encoder-Decoder** framework.
    - **Attention Mechanism** became crucial for the decoder to focus on relevant parts of the input text during summary generation.
- **Challenge:** Abstractive models were difficult to train, often suffered from **repetition** (a recurring word problem), and faced the **Out-of-Vocabulary (OOV)** problem.

LLMs (Late 2010s – Present)

- **Extractive** Models: **BERT**-based models for sentence classification.
- **Abstractive** Models: **BART, T5, Pegasus** (Encoder-Decoder architectures optimized for sequence generation).

Current State: Large Language Models (LLMs)

- Models like **GPT-3/4** and advanced open-source models (e.g., Llama) show superior **fluency, coherence,** and an ability to perform **zero-shot/few-shot** summarization.
- Driven by **Prompt Engineering** and fine-tuning on diverse summarization tasks.

1. Machine Translation
2. Dialogue
3. Summarization
4. **Paraphrase Generation**

Paraphrasing the Gettysburg Address

Original Sentence (Gettysburg Address)

"Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal."

Potential Paraphrases

- **Lexical/Simple Change:** "Eighty-seven years ago, our forefathers established a new country here, founded on freedom, asserting that everyone is born equal."
- **Syntactic/Structural Change:** "On this land, eighty-seven years prior, our ancestors founded a new nation. It was conceived in the spirit of liberty and dedicated to the principle of universal equality."
- **More Abstract/Generative:** "Eight decades and seven years ago, the founders of our nation laid the groundwork for a new country, born from the ideals of freedom and the belief that all individuals are created with equal rights."

Lexical and Rule-Based Methods (Early 2000s)

- **Focus:** Primarily on **lexical (word) and simple syntactic variations** to generate paraphrases.
- **Key Techniques:**
  - **Thesaurus Substitution:** Replacing single words with synonyms from lexical resources like WordNet.
  - **Hand-Crafted Rules:** Using manual or automatically extracted rules for specific phrase-level or syntactic transformation (e.g., Active → Passive voice).
  - **Pattern Extraction:** Using techniques like **Multi-Sequence Alignment (MSA)** on comparable corpora (e.g., news articles about the same event) to find recurring sentence patterns/templates
- **Limitation:** These methods lacked fluency and could only generate a **limited diversity** of paraphrases, often requiring significant manual effort.

MT-style approaches (Mid-2000s)

- **Core Idea:** Paraphrase generation is treated as **Monolingual Machine Translation** (translating English to English).
- **Methodology:**
  - **Data:** Trained on large volumes of **monolingual parallel data** (sentence pairs with similar meaning) typically extracted from clustered news articles.
  - **Model:** Used **Phrase-Based SMT** (PB-SMT) tools. The noisy channel model finds the optimal paraphrase T^* of a sentence S by maximizing $P(S|T)P(T)$.
- **Benefit:** Greatly improved **coverage** and **scalability** compared to earlier rule-based systems.
- **Limitation:** Still focused primarily on phrase-level changes and inherited issues of SMT models.

Paraphrase generation as a monolingual MT task

Feature	Standard Machine Translation (MT)	Paraphrase Generation (PG)
Source Language (L_s)	L_A (e.g., French)	L_A (e.g., English)
Target Language (L_t)	L_B (e.g., English)	L_A (e.g., English)
Goal	Preserve **meaning** across **languages**	Preserve **meaning** across **forms**
Input/Output	"Je suis fatigué" $\rightarrow$ "I am tired"	"The car is red" $\rightarrow$ "The automobile is crimson"

Neural Networks and Seq2Seq (Early to Mid-2010s)

- **Paradigm Shift:** Transition from statistical models to **Neural Networks** for sequence generation.
- **Key Architecture: Sequence-to-Sequence (Seq2Seq)** models.
  - Used **Recurrent Neural Networks (RNNs)**, particularly **LSTMs**, in an Encoder-Decoder structure.
  - The model is trained to map an input sentence to its corresponding paraphrase.
  - Models often incorporated **Attention Mechanisms** and specialized layers (like **Stacked Residual LSTMs**) for better training and context modeling.
- **Advanced Techniques:**
  - Incorporating **Pointer Networks** or **Copy Mechanisms** to effectively copy common words/phrases from the input, which is essential for controlled paraphrasing.
  - Using **Reinforcement Learning** (RL) with an external evaluator/discriminator to further fine-tune the generator for better quality and diversity (Li et al., 2018).

Transformers and Large Language Models (LLMs) (Late 2010s – Present)

- **Current State:** Dominated by the **Transformer Architecture**.
- **Core Models:**
  - **T5 (Text-to-Text Transfer Transformer):** A unified framework that excels at paraphrase generation when fine-tuned on the task.
  - **BART/Pegasus:** Encoder-Decoder models that generate highly **fluent and coherent** paraphrases by leveraging massive pre-training.
  - **LLMs (e.g., GPT-4, Llama):** Provide state-of-the-art performance, generating diverse and high-quality paraphrases with high **semantic fidelity** through prompting and zero/few-shot learning.
- **Ongoing Challenges:**
  - Balancing **Semantic Fidelity** (preserving meaning) vs. **Diversity** (changing structure/lexicon).
  - Preventing **Hallucination** (generating factually incorrect but fluent text).
  - Developing metrics beyond simple n-gram overlap (like **BLEU** and **ROUGE**) that truly capture semantic equivalence.

Next class: November 11

Assignment 3: Due November 11

Large Language Models: Data, modeling, and tokenization

- Jurafsky & Martin Chapter 7: Large Language Models
- The Pile: An 800GB Dataset of Diverse Text for Language Modeling
- Masked language modeling (HF tutorial)
- Causal language modeling (HF tutorial)
- Fast WordPiece Tokenization
- Neural machine translation of rare words with subword units.