Machine Translation

Machine Translation: Language Divergences and Typology, Machine Translation using Encoder-Decoder, Details of the Encoder-Decoder Model, Translating in Low-Resource Situations, MT Evaluation, Bias and Ethical Issues.

Textbook 2: Ch. 13. (Exclude 13.4)

Overview:

- Machine Translation: The use of computers to translate from one language to another.
- MT for information access is probably one of the most common uses of NLP ○ We might want to translate some instructions on the web, perhaps the recipe for a favorite dish, or the steps for putting together some furniture.
    - We might want to read an article in a newspaper, or get information from an online resource like Wikipedia or a government webpage in some other language. ○ Google Translate alone translates hundreds of billions of words a day between over 100 languages.

- Another common use of machine translation is to aid human translators.
    - This task is often called computer-aided translation or CAT.
    - CAT is commonly used as part of localization: the task of adapting content or a product to a particular language community.
- Finally, a more recent application of MT is to in-the-moment human communication needs. This includes incremental translation, translating speech on-the-fly before the entire sentence is complete, as is commonly used in simultaneous interpretation.
- Image-centric translation can be used for example to use OCR of the text on a phone camera image as input to an MT system to translate menus or street signs.

5.1 Language Divergences and Typology

There are about 7,000 languages in the world, some aspects of human language seem to be universal. For example, language seems to have words for referring to people, for talking about eating.

Structural linguistic universals; for example, every language seems to have nouns and verbs.

Languages are different in many ways. People have noticed this since ancient times (see Fig.

5.1).

Fig 5.1: The Tower of Babel, Pieter Bruegel 1563. Wikimedia Commons, from the Kunsthistorisches Museum, Vienna.

***Story of The Tower of Babel* (Bruegel, 1563):**

- Bruegel's painting depicts the biblical story from **Genesis 11**, where humanity, *speaking a single language, tries to build a tower to reach the heavens*.
- As a divine response, *God confuses their language*, causing miscommunication and halting the project. It's a cautionary tale about human ambition and the limits of communication.

To build better machine translation (MT) systems, we need to understand why translations can be different (Dorr, 1994).

- ***Differences about words themselves***. For example, each language has a different word for "dog." These are called **idiosyncratic or lexical differences**, and we handle them one by one.
- ***Differences about patterns***. For example, some languages put the verb before the object. Others put the verb after the object. These are **systematic differences** that we can model more generally. The study of these patterns across languages is called **linguistic typology**.

Different types of Language Divergences and Typology:

- Word Order Typology
- Lexical Divergences
- Morphological Typology
- Referential density

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

5.1.1 Word Order Typology

- English and Japanese, languages differ in the basic word order of verbs, subjects, and objects in simple declarative clauses.

- German, French, English, and Mandarin, are all SVO (Subject-Verb-Object) languages.
- Hindi and Japanese, by contrast, are SOV languages, the verb tends to come at the end of basic clauses.
- Irish and Arabic are VSO languages.

Two languages that share their basic word order type often have other similarities. For example, *VO languages generally have prepositions*, whereas *OV languages generally have postpositions*.

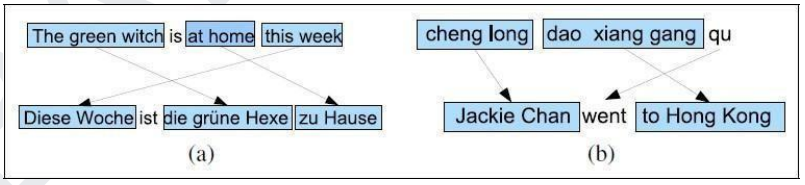
English: *He wrote a letter to a friend*
Japanese: *tomodachi ni tegami-o kaita*
friend to letter wrote
Arabic: *katabt risāla li ṣadq*
wrote letter to friend

VO ❼ verb ***wrote*** is followed by its object ***a letter*** and the prepositional phrase ***to a friend***, in which the preposition ***to*** is followed by its argument ***a friend***.

- Arabic, with a ***VSO*** order, also has the ***verb before the object and prepositions***.
- Other kinds of ordering preferences vary idiosyncratically - In some SVO languages (like English and Mandarin) adjectives tend to appear before nouns, while in others languages like Spanish and Modern Hebrew, adjectives appear after the noun.

Spanish *bruja verde* **English** *green witch*

Fig. shows examples of other word order differences. All of these word order differences between languages can cause problems for translation, requiring the system to do huge structural reorderings as it generates the output.



5.1.2 Lexical Divergences

Translate the individual words from one language to another.

The appropriate word can vary depending on the context.

- Bass - English source-language; fish lubina or the musical instrument bajo - in Spanish.
- A wall – English; Wand (walls inside a building), and Mauer (walls outside a building) – in German.
- Brother - English uses the word brother for any male sibling, Chinese and many other languages have distinct words for older brother and younger brother (Mandarin gege and didi, respectively)
- One language places more grammatical constraints on word choice than another.

- English marks nouns for whether they are singular or plural. Mandarin doesn't. Or French and Spanish.

- Lexically dividing up conceptual space, leading to many-to-many mappings.
  - English leg, foot, and paw ❼ French.
  - For example, when leg is used about an animal ❼patte; leg of a journey ❼ etape; leg is of a chair ❼ pied.

- Lexical gap: Mandarin *xiào* or Japanese *oyakōkō* , English does not have a word that corresponds neatly for phrases like: filial piety or loving child, or good son/daughter.

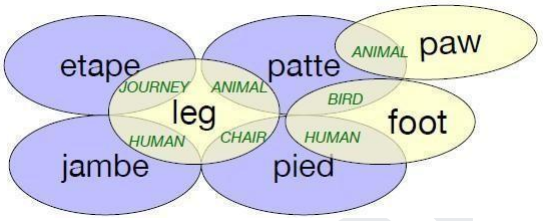


Fig.: The complex overlap between English leg, foot, etc.,

- **Verb-framed** languages (Spanish, French, Japanese) ○ **Verb** shows the **direction or path** of movement.
  - The **manner** (how something moves) is optional or added separately.

  Examples:
  - *Entrar* = to enter ○ *Salir* = to go out ○ *Subir* = to go up
- **Satellite-framed** languages (English, German, Russian) ○ The **verb** shows the **manner** of movement.
  - The **direction or path** is shown in a **satellite** (like a preposition or particle).

  Examples:
  - *Run out* = run (manner) + out (path) ○ *Jump over* = jump (manner) + over (path) ○ *Slide down* = slide (manner) + down (path)

5.1.3 Morphological Typology

- Languages differ morphologically along two main dimensions:

1. Number of Morphemes per Word:

- Isolating languages (e.g., Vietnamese, Cantonese): ○ One morpheme per word.
  - Words are simple and not combined.
- Polysynthetic languages (e.g., Siberian Yupik):
  - Many morphemes in one word.

- One word can express a full sentence.

2. Segmentability of Morphemes:

- Agglutinative languages (e.g., Turkish):
  - Morphemes are clearly separable.
  - Each morpheme represents one meaning or function.
- Fusional languages (e.g., Russian):
  - Morphemes blend together.
  - One morpheme may carry multiple meanings (e.g., case, number, gender). In Machine Translation:

- Translating morphologically rich languages requires understanding parts of words.
- Modern systems often use subword models (like BPE or WordPiece) to handle this.

5.1.4 Referential density

- Measures how often a language uses explicit pronouns.
- **High referential density (hot languages):**
  - Use pronouns more frequently.
  - Easier for the listener (e.g., English).
- **Low referential density (cold languages):**
  - Omit pronouns often.
  - Listener must infer more (e.g., Chinese, Japanese).

Hot vs. Cold Languages:

- **Hot languages** = more explicit, easier for the listener.
- **Cold languages** = less explicit, require more inference.

Translation Challenge:

- **Translating from pro-drop (cold) to non-pro-drop (hot) languages** is tricky.
- The system must **infer missing pronouns** and insert the correct ones.

5.2 Machine Translation using Encoder-Decoder (sequence to- sequence model)

- Translate each sentence independently – ○ ***source language ❼ target language***.
  - The green witch arrived (English) ❼ Lleg´o la bruja verde (Spanish).
- MT uses supervised machine learning:
  - System is given a large set of parallel sentences. ○ Learns to map source sentences into target sentences. ○ Split the sentences into a sequence of subword tokens.

- The systems are then trained to ***maximize the probability*** of the sequence of tokens in the ***target language*** $y_1,\ldots,y_m$ given the sequence ***of tokens in the source language*** $x_1,\ldots,x_n$: $P(y_1,\ldots,y_m|x_1,\ldots,x_n)$

- Rather than use the input tokens directly, the encoder-decoder architecture is used.
  - ***Encoder*** takes the input words $x = [x_1,\ldots,x_n]$ and produces an intermediate context ***h.***
  - ***Decoder***, the system takes ***h*** and, word by word, generates the output ***y***:

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

5.2.1 Tokenization

- Machine translation systems use a **fixed vocabulary** decided in advance.
- The vocabulary is built by running a tokenization algorithm on **both source and target language texts** together.
- This vocabulary is made using **subword tokenization**, not by splitting at spaces.
- An example of a subword tokenization method is **BPE (Byte Pair Encoding)**.
- **One shared vocabulary** is used for both source and target languages.
- This sharing makes it **easy to copy names and special words** from one language to another.
- Subword tokenization works well for languages with **spaces** (like English, Hindi) and **no spaces** (like Chinese, Thai).
- Modern systems use **better algorithms** than simple BPE.
  - For example, **BERT** uses **WordPiece**, a smarter version of BPE.
- WordPiece chooses – **merges that improve the language model probability**, not just based on frequency.
- Wordpieces use a special symbol at the beginning of each token;

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 algorithm*** is *given a training corpus and a desired vocabulary size V*, and proceeds as follows:

1. ***Initialize the wordpiece lexicon with characters*** (for example a subset of Unicode characters, collapsing all the remaining characters to a special unknown character token).
2. Repeat until there are V wordpieces:

   (a) ***Train an n-gram*** language model on the training corpus, using the current set of wordpieces.

(b) ***Consider the set of possible new wordpieces made by concatenating two wordpieces from the current lexicon***. Choose the one new wordpiece that most increases the language model probability of the training corpus.

Unigram Model:

- Unlike BPE, which requires specifying the number of merges, ***WordPiece*** and the ***unigram algorithm*** let users define a target vocabulary size, typically between 8K–32K tokens.
- The unigram algorithm, often referred to as SentencePiece (its implementation library), starts with a large initial vocabulary of characters and frequent character sequences.
- Unigram ***iteratively removes low-probability tokens using statistical modeling*** (like the EM algorithm) until reaching the desired size.
- It generally outperforms BPE by avoiding overly fragmented or non-meaningful tokens and better handling common subword patterns.

```
Original:  corrupted          Original:  Completely preposterous suggestions
BPE:       cor rupted         BPE:       Comple t ely prep ost erous suggest ions
Unigram:   corrupt ed         Unigram:   Complete ly pre post er ous suggestion s
```

5.2.2 Creating the Training data

Machine translation models are trained on a *parallel corpus*, sometimes called a bitext, a text that appears *in two (or more) languages*.

- **Europarl corpus:** Proceedings of the European Parliament, contains between 400,000 and 2 million sentences each from 21 European languages.
- **The United Nations Parallel Corpus:** contains on the order of 10 million sentences in the six official languages of the United Nations (Arabic, Chinese, English, French, Russian, Spanish).
- **OpenSubtitles corpus & ParaCrawl corpus:** Movie and TV subtitles & general web text, 223 million sentence pairs between 23 EU languages and English extracted from the CommonCrawl.

Sentence alignment

Standard training corpora for MT come as aligned pairs of sentences.

When creating new corpora, for example for underresourced languages or new domains, these sentence alignments must be created.

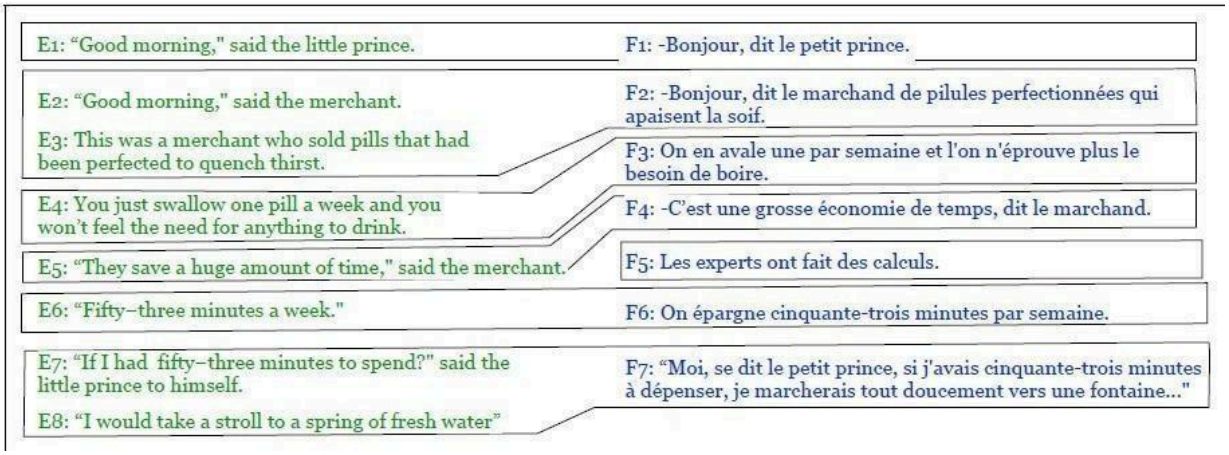


Fig: A sample alignment between sentences in English and French, with sentences extracted from Antoine de Saint-Exupery's Le Petit Prince and a hypothetical translation. Sentence alignment takes sentences $e_1,...,e_n$, and $f_1,\ldots\ldots,f_m$ and finds minimal sets of sentences that are translations of each other, including single sentence mappings like (e1, f1), (e4, f3), (e5, f4), (e6, f6) as well as 2-1 alignments (e2/e3,f2), (e7/e8,f7), and null alignments (f5).

we generally need two steps to produce sentence alignments:

1. A cost function that takes a span of source sentences and a span of target sentences and returns a score measuring how likely these spans are to be translations.
2. An alignment algorithm that takes these scores to find a good alignment between the documents.

Cost function between two sentences or spans ***x, y*** from the source and target documents respectively:

$$c(x,y) = \frac{(1-\cos(x,y))\text{nSents}(x)\ \text{nSents}(y)}{\sum_{s=1}^{S} 1-\cos(x,y_s) + \sum_{s=1}^{S} 1-\cos(x_s,y)}$$

where ***nSents()*** gives the number of sentences (this biases the metric toward many alignments of single sentences instead of aligning very large spans). The denominator helps to normalize the similarities, and so $x_1,...,x_S, y_1,...,y_S$ are randomly selected sentences sampled from the respective documents.

5.3 Details of the Encoder-Decoder Model

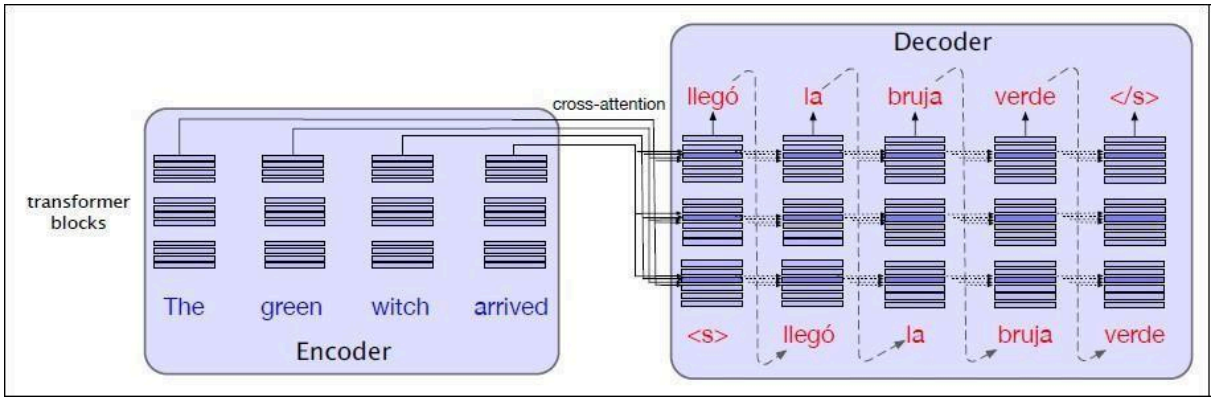
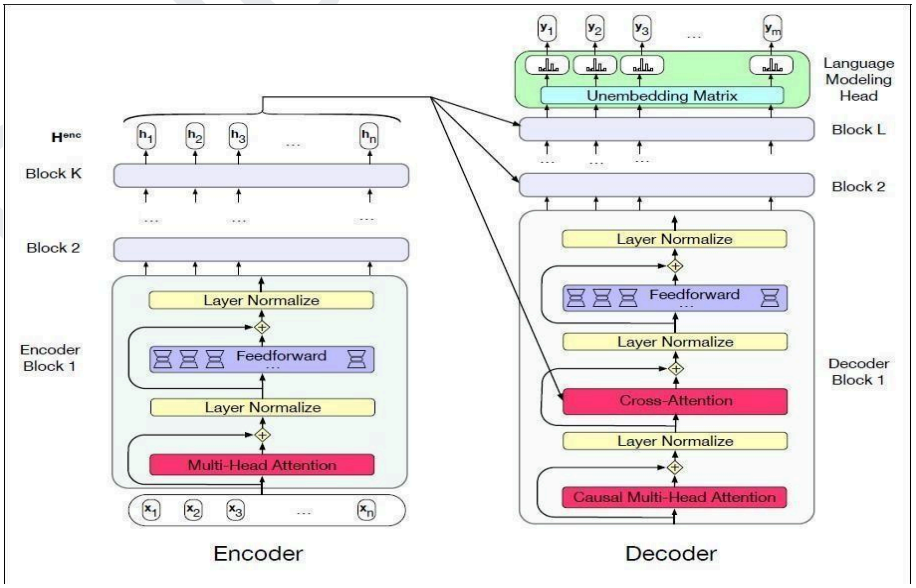


Fig. The encoder-decoder transformer architecture for machine translation.

- Fig. shows the intuition of the architecture at a high level.
- The encoder-decoder architecture is made up of two transformers: an encoder, which is the same as the basic transformers, and a decoder, which is augmented with a special new layer called the ***cross-attention layer***.
- The encoder takes the source language input word tokens X = $x_1, \ldots\ldots, x_n$ and ***maps them to*** an output representation $H_{enc} = h_1, \ldots, h_n$; via a stack of encoder blocks.
- The decoder attends to the encoder representation and generates the target words one by one.
    - At each timestep conditioning on the source sentence and the previously generated target language words to generate a token.
- In order to attend to the source language, the transformer blocks in the decoder have an extra cross-attention layer.
- A ***self-attention*** layer that attends to the input from the previous layer, followed by layer norm, a feed forward layer, and another layer norm.



Each **encoder block** consists of:

1. **Multi-Head Self-Attention** ○ Allows the model to focus on different positions in the input sequence simultaneously.

- Uses scaled dot-product attention in multiple "heads."

2. **Add & Layer Normalization** ○ Adds the input and the output of the attention layer (residual connection).
   - Applies layer normalization.
3. **Feed-Forward Neural Network (FFN)** ○ Two linear layers with a ReLU (or GELU) in between.
   - Applies to each position independently.
4. **Add & Layer Normalization** (again) ○ Adds the input and output of the FFN and normalizes.

Each **decoder block** adds an additional attention layer:

1. **Masked Multi-Head Self-Attention** ○ Prevents attending to future tokens (important during training).
2. **Add & Norm**
3. **Encoder-Decoder Attention (*Cross attention*)** ○ The decoder attends to encoder outputs. ○ Helps generate context-aware outputs.
4. **Add & Norm**
5. **Feed-Forward Neural Network**
6. **Add & Norm**

Cross attention is given by:

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

Where,

$$\mathbf{Q} = \mathbf{H}^{dec[\ell-1]}\mathbf{W}^{\mathbf{Q}}; \quad \mathbf{K} = \mathbf{H}^{enc}\mathbf{W}^{\mathbf{K}}; \quad \mathbf{V} = \mathbf{H}^{enc}\mathbf{W}^{\mathbf{V}}$$

- Multi-head attention the input to each attention layer is $\boldsymbol{X}$,
- The final output of the encoder $\boldsymbol{H^{enc} = h, \ldots, h_n}$. $\boldsymbol{H^{enc}}$ is of shape ***[n, d]***
- The query ($\boldsymbol{Q}$) comes from the output from the prior decoder $\mathbf{H}^{dec[\ell-1]}$, which is multiplied by the cross-attention layer's query weights $\boldsymbol{W^Q}$
- $\boldsymbol{K}$ = Multiply the encoder output $\boldsymbol{H^{enc}}$ by the cross-attention layer's key weights $\boldsymbol{W^K}$
- $\boldsymbol{V}$ = Multiply the encoder output $\boldsymbol{H^{enc}}$ by the cross-attention layer's key weights $\boldsymbol{W^v}$

- To train an encoder-decoder model, we use the same self-supervision model we used for training encoder-decoders RNNs.

- The network is given the source text and then starting with the separator token is trained autoregressively to predict the next token using cross-entropy loss.
- Cross-entropy is determined by the probability the model assigns to the correct next word.
- We use ***teacher forcing*** in the decoder, at each time step in decoding we force the system to use the gold target token from training as the next input.

5.4 Translating in low-resource situations

- For some languages, and especially for English, online resources are widely available.
- There are many large parallel corpora that contain translations between English and many languages.
- But, the vast majority of the world's languages *do not have large parallel training texts* available.

How to get good translation with lesser resourced languages? – Data Sparsity

Two commonly used approaches: Data augmentation [Backtranslation], and Multilingual models.

5.4.1 Data Augmentation

- Statistical technique for dealing with insufficient training data.
- Adding new synthetic data that is generated from the current natural data.
- **Backtranslation** is a common data augmentation technique in machine translation (MT) using monolingual target-language data (text written **only in the target language)** to create synthetic parallel data.
- It addresses the scarcity of parallel corpora by generating synthetic bitexts from abundant monolingual data.
- The process involves training a **target-to-source MT model** using available parallel data, then using it to translate monolingual target-language text into the source language.
- The resulting **synthetic source-target pairs** are added to the training data to improve the original source-to-target MT model.
- Backtranslation includes configurable parameters like **decoding methods** (greedy, beam search, sampling) and **data ratio settings** (e.g., upsampling real bitext).
- It is highly effective—studies suggest it provides about **two-thirds the benefit** of training on real bitext.

Example:

🎯 **Goal: Build a machine translation system from Navajo → English**

📦 **What we have:**

- A **small Navajo-English parallel corpus** (real translations).
- A **large monolingual English corpus** (just English sentences, no Navajo translations).

🔁 **Backtranslation Process:**

1. **Train a reverse model**:
   Use the small parallel corpus to train an **English → Navajo** MT system.
2. **Translate monolingual English**:
   Use this reverse model to translate English sentences into Navajo.
   Example:
   - English (original): "*The sun is shining.*"
   - Translated Navajo (synthetic): "*Dóó chizh yá'át'éeh.*" (machine-generated)
3. **Create synthetic parallel data**:
   Pair the original English sentence with its machine-translated Navajo version.
   → Synthetic bitext: *(Dóó chizh yá'át'éeh, The sun is shining)*
4. **Retrain the main model**:
   Combine this synthetic bitext with the original bitext and retrain the **Navajo → English** model.

5.4.2 Multilingual models

- The models we've described so far are for bilingual translation: one source language, one target language.
- It's also possible to build a multilingual translator. In a multilingual translator, we train the system by giving it parallel sentences in many *different pairs of languages*.
- We tell the system which language is which by adding a *special token* l_s to the encoder specifying the source language we're translating from, and a *special token* l_t to the decoder telling it the target language we'd like to translate into.

$$\begin{aligned} \mathbf{h} &= \text{encoder}(x, l_s) \\ y_{i+1} &= \text{decoder}(\mathbf{h}, l_t, y_1, \ldots, y_i) \quad \forall i \in [1, \ldots, m] \end{aligned}$$

One advantage of a multilingual model is that they can improve the translation of lower-resourced languages by drawing on information from a similar language in the training data that happens to have more resources.

5.4.3 Sociotechnical issues

- **Limited native speaker involvement**: Many low-resource language projects lack participation from native speakers in content curation, technology development, and evaluation.

- **Low data quality**: some multilingual datasets were of acceptable quality, often containing errors, repeated content, or boilerplate text—likely due to insufficient native speaker oversight.
- **English-centric bias**: Many MT systems prioritize language pairs involving English, limiting the diversity of language coverage.
- **Efforts to broaden coverage**: Recent large-scale projects aim to support MT across hundreds of languages, expanding beyond English-centric training.
- **Participatory design**: Researchers advocate for involving native speakers in all stages of MT development, including through online communities, mentoring, and collaborative infrastructure.
- **Improved evaluation method**: Instead of direct evaluation, post-editing MT outputs (then measuring the difference) helps reduce bias from linguistic variation and simplifies training evaluators.

5.5 MT Evalaution

Translations are evaluated along two dimensions:

1. **Adequacy:** how well the translation captures the exact meaning of the source sentence. Sometimes called faithfulness or fidelity.
2. **Fluency:** how fluent the translation is in the target language (is it grammatical, clear, readable, natural).

5.5.1 Using Human Raters to Evaluate MT

- Human evaluation is the most accurate method for assessing machine translation (MT) quality, focusing on two main dimensions: **fluency** (how natural and readable the translation is) and **adequacy** (how much meaning from the source is preserved).
- Raters, often crowdworkers, assign scores on a scale (e.g., 1–5 or 1–100) for each.
- Bilingual raters compare source and translation directly for adequacy, while monolingual raters compare MT output with a human reference. Alternatively, raters may choose the better of two translations.
- Proper training is crucial, as raters often struggle to distinguish fluency from adequacy.
- To ensure consistency, outliers are removed and ratings are normalized.
- Another evaluation method involves post-editing: raters minimally correct MT output, and the extent of editing reflects translation quality.

5.5.2 Automatic Evaluation

Human evaluation can be time consuming and expensive. In this regard, automatic metrics are often used as temporary proxies.

Automatic metrics are less accurate than human evaluation, but can help test potential system improvements, and even be used as an automatic loss function for training.

Automatic Evaluation by Character Overlap: chrF

- The simplest and most robust metric for MT evaluation is called ***chrF***, which stands for character F-score.
- A good machine translation will tend to *contain characters and words that occur in a human translation of the same sentence*.
- Consider a test set from a parallel corpus, in which each source sentence has both a gold human target translation and a candidate MT translation we'd like to evaluate.
- ***chrP percentage of character*** 1-grams, 2-grams, ..., k-grams in the hypothesis that occur in the reference, averaged.
- ***chrR percentage of character*** 1-grams, 2-grams,..., k-grams in the reference that occur in the hypothesis, averaged.
- The metric then computes an F-score by combining ***chrP and chrR*** using a weighting parameter $\boldsymbol{\beta}$. It is common to set $\boldsymbol{\beta} = 2$, thus weighing recall twice as much as precision:

$$\text{chrF}\beta = (1+\beta^2)\frac{\text{chrP}\cdot\text{chrR}}{\beta^2\cdot\text{chrP}+\text{chrR}}$$

For $\beta = 2$, that would be:

$$\text{chrF2} = \frac{5\cdot\text{chrP}\cdot\text{chrR}}{4\cdot\text{chrP}+\text{chrR}}$$

Example:

REF: witnessforthepast, (18 unigrams, 17 bigrams)
HYP1: witnessofthepast, (17 unigrams, 16 bigrams)

Next let's see how many unigrams and bigrams match between the reference and hypothesis:

unigrams that match: w i t n e s s f o t h e p a s t , (17 unigrams)
bigrams that match: wi it tn ne es ss th he ep pa as st t, (13 bigrams)

We use that to compute the unigram and bigram precisions and recalls:

unigram P: 17/17 = 1 unigram R: 17/18 = .944
bigram P: 13/16 = .813 bigram R: 13/17 = .765

Finally we average to get chrP and chrR, and compute the F-score:

$$\begin{aligned}\text{chrP} &= (17/17+13/16)/2 = .906\\ \text{chrR} &= (17/18+13/17)/2 = .855\\ \text{chrF2,2} &= 5\frac{\text{chrP}*\text{chrR}}{4\text{chrP}+\text{chrR}} = .86\end{aligned}$$

Alternative overlap metric: BLEU

- **BLEU** is a traditional metric used to evaluate machine translation quality.
- It is **precision-based**, not based on recall.
- It uses **n-gram precision**, meaning it checks how many n-grams (1 to 4 words in a row) in the translation match the reference.
- It calculates a **geometric mean** of unigram to 4-gram precision scores.

- BLEU includes a **brevity penalty** to avoid favoring translations that are too short.
- It uses **clipped counts**, which means it limits how often a match is counted to avoid overestimating quality.
- BLEU is **word-based** and **sensitive to tokenization**, so consistent tokenization is important.
- BLEU doesn't perform well with **languages that have complex word forms**.
- Tools like **SACREBLEU** are used to ensure consistent evaluation across different systems.

Statistical Significance Testing for MT evals

- **chrF** and **BLEU** are overlap-based metrics used to compare machine translation (MT) systems.
- They help answer: **Did the new translation system improve over the old one?**
- To check if a difference in scores is **statistically significant**, we use tests like the **paired bootstrap test** or **randomization test**.
- For a **confidence interval** on one system's chrF score:
  - Create many **pseudo-testsets** by sampling the original test set with replacement.
  - Calculate the chrF score for each one.
  - Drop the top and bottom 2.5% of scores → the rest give a **95% confidence interval**.
- To **compare two systems (A and B)**:
  - Use the same pseudo-testsets. o Compare chrF scores on each.
  - Count the percentage of sets where **A scores higher than B**.

chrF: Limitations

- Metrics like ***chrF*** and ***BLEU*** are helpful but have important limitations.
- ***chrF*** is local — it doesn't handle large phrase movements or reordering well.
- ***chrF*** can't assess document-level features like discourse coherence.
- These metrics are not good for comparing very different systems (e.g., human-aided vs. machine translation).
- ***chrF*** works best for comparing small changes within the same system.

5.6 Bias and Ethical Issues

- MT systems can show **gender bias**, especially when translating from **gender-neutral languages** (like Hungarian or Spanish) to **gendered ones** (like English).
- They may **default to male pronouns** due to lack of gender info or cultural stereotypes.

- Example: Hungarian gender-neutral pronoun "ő" becomes:
  - "she" if the job is *nurse* ○
  - "he" if the job is *CEO*
- These biases reflect and **reinforce gender stereotypes**, which is a serious ethical concern.
- The **WinoMT dataset** tests MT systems on sentences involving **non-stereotypical gender roles**.
- MT systems often perform **worse** on such sentences.
- Example: A system may mistranslate **"The doctor asked the nurse to help her"** if it expects the doctor to be male.