# 1. Neural Machine Translation with RNNs

**Solution (g)**

The masks are used to set the attention scores $\mathbf{e}_{t,i} = -\infty$ for all the positions $i$ that correspond to 'pad' tokens in the source sentence. This means that after we apply the softmax function, the attention distribution $\boldsymbol{\alpha}_{t,i} = 0$ for all positions $i$ that correspond to `pad' tokens. This means that the encoder hidden states $\mathbf{h}_i^{\mathbf{enc}}$ that correspond to `pad' tokens have no effect on the attention output $\mathbf{a}_t$.

It is necessary to apply the masks in this way because we do not want to put any attention on the encoder hidden states that correspond to `pad' tokens. These `padding' hidden states were computed because we have to pad the batch of source sentences up to maximum length $m$, but they are meaningless and we do not want them to affect our model.

**Solution (i)**

One advantage of dot product attention is that it is the most computationally efficient. It is only an O(h) operation, i.e. a dot product, and it does not require us to learn and store any weight matrix **W**. One con of dot product attention is that it requires both the encoder and decoder states to be of the same dimension. Additionally, it is less expressive, as there is no relative weighting of hidden units.

One advantage of multiplicative attention is that the hidden states of the encoder and decoder can be of different dimensions. This is why we used multiplicative attention, rather than dot product attention, in our NMT system. Another advantage over dot product attention is that it is more expressive, by learning the weights **W**. An advantage that multiplicative attention provides over additive attention is that it is more efficient, i.e. can be implemented solely through matrix multiplication and requires fewer weights.

One advantage of additive attention is that it is a fundamentally different operation than the dot product and matrix multiplications. Thus, it may be able to capture different relationships between words. One disadvantage of additive addition is that it is the most computationally complicated, relying on learning two weight matrices and a vector. The more parameters we add to a model, the more difficult it is to train the model.

# 2. Analyzing NMT Systems

**Solution (a)**

Cherokee uses a unique 85- character syllabary invented by Sequoyah in the early 1820s, which is highly different from English's alphabetic writing system. Cherokee is a polysynthetic language, meaning that words are composed of many morphemes that each have independent

meanings. A single Cherokee word can express the meaning of several English words. The semantics are often conveyed by the rich morphology, the word orders of Cherokee sentences are variable. There is no "basic word order" in Cherokee, and most word orders are possible (Montgomery- Anderson, 2008), while English generally follows the Subject-Verb-Object (SVO) word order. Plus, verbs comprise 75% of Cherokee, which is only 25% for English (Feeling, 1975, 1994).

**Solution (b)**
On the input side, they dramatically increase the vocabulary our models can handle and show resilience in the face of spelling mistakes and rare words. On the output side, character models are computationally cheaper due to the small size of their vocabulary. This attribute makes training techniques (such as cotraining a language model) feasible and fast even under a constrained budget.

**Solution (c)**
In the multilingual NMT scenario, the automatic translations used as the source part of the extended training data will likely contain a mixed-language that includes words from a vocabulary shared with other languages. The expectation is that, round after round, the model will generate better outputs by learning at the same time to translate and "correct" its own translations by removing spurious elements from other languages.

**Solution (d)**
**i.** Source Translation: *Yona utsesdo ustiyegv anitsilvsgi digvtanv uwoduisdei.*
Reference Translation: *Fern had a crown of daisies in her hair.*
NMT Translation: *Fern had her hair with her hair.*

**Reason for error**:The model might be producing **her hair** because it was attending to (or influenced by) the proper noun. Alternatively, if the training data contained more examples, the model might have learned a connection between daisies and hair, leading to incorrectly producing the word 'hair'. Another possible reason is that in the her hair is a more common phrase than a crown of daisies - NMT systems sometimes have a tendency default to general, unconditional target (English) Language Modeling.

**Possible Fix**: First inspect the attention distribution on the step when the decoder produced the phrase her hair. This might help us figure out which (if any) of the possible reasons above is responsible. If we believe the problem is the dataset, we might try to add more examples of to the dataset, or try any of the current research techniques to debias our word vectors and/or model (this is an open research area).

**ii.** Source Sentence: *Ulihelisdi nigalisda.*

Reference Translation: *She is very excited.*
NMT Translation: *It's joy.*

**Reason for error**: A deeper reason for this error is that NMT systems do not generally have logic, reasoning or numerical abilities - here the system considered only the semantic (not logic) similarity of 'excited & joy' and 'she & its'.

**Problem Fix:** The simple solution for this particular example would be adding subword abilities. A more complex solution would be to supply the NMT system with a knowledge base of English dictionary (word and their meanings, synonyms, antonyms, homonyms, etc.), and train the system to convert accordingly. A much more complex and general solution would be to work on imbuing NMT systems with reasoning abilities. This is an open research problem! The most dicult solution of all would be to convince the Oxford/Merriam Dictionaries to change the meaning or make it the same.

**iii.** Source Sentence: *Tsesdi hana yitsadawoesdi usdi atsadi!*
Reference Translation: *Don't swim there, Little fish!*
NMT Translation: *Don't know how a small fish!*

**Reason for error**: The model might be producing small because it was attending to (or influenced by) the pronouns earlier in the sentence. Alternatively, if the training data contained more examples of words with their uses, the model might have learned a connection between small and little, leading to incorrectly producing the word small. There are many possible reasons for this error. Perhaps our NMT system is not good at modeling long-term dependencies, so struggles to produce a long output sentence that makes sense. Perhaps the NMT system has not been trained on enough data to have learned how to rearrange word ordering appropriately. In particular, perhaps the decoder's language model is not strong enough to recognize that this output translation is unnatural English.

**Possible Fix:** To improve modeling of long term dependencies, we might make the model more powerful architecturally (e.g. increase hidden size, number of layers, add self-attention, or switch to Transformer). If we think the problem is insufficient data, we might train the NMT system on more data, or build our system on top of a pre trained system (e.g., ELMo or BERT). In particular if we think that the decoder's English Language Model is too weak, we might try initializing our decoder with a strong English LM trained on lots of data.

**Solution (f)**
**(i)** For $c_1$, $p_1 = 3/5$ and $p_2 = 2/4 = 1/2$ , len(c) = 5 and the closest reference length is len(r) = 4 (because 4 and 6 are equally close to 5, so we choose 4). Thus the brevity penalty BP = 1. Therefore BLEU = $exp(1/2\ log(3/5) + 1/2\ log(1/2)) = 0.548$.

For $c_2$, $p_1 = 4/5$ and $p_2 = 2/4 = 1/2$ . Again, len(c) = 5 and len(r) = 4 so BP = 1. Therefore BLEU = $exp(1/2 \log (4/5) + 1/2 \log (1/2)) = 0.632$.

According to these BLEU scores, NMT translation $\mathbf{c_2}$ is the better one. $\mathbf{c_2}$ is indeed the better translation - it has the correct meaning, whereas $\mathbf{c_1}$ translates the Spanish phrase too literally, leading to unnatural and nonsensical English.

**(ii)** For $c_1$, $p_1 = 3/5$ and $p_2 = 2/4 = 1/2$ . len(c) = 5 and len(r) = 6 so the brevity penalty BP = $exp(1 - (6/5))$. Therefore BLEU = $exp(1 - (6/5)) * exp(1/2 \log (3/5) + 1/2 \log (1/2)) = 0.448$.

For $c_2$, $p_1 = 2/5$ and $p_2 = 1/4$ . len(c) = 5 and len(r) = 6 so the brevity penalty BP = $exp(1-(6/5))$. Therefore
BLEU = $exp(1 -( 6/5 ))* exp(1/2 \log (2/5) + 1/2 \log (1/4)) = 0.259$.

According to these BLEU scores, NMT translation $\mathbf{c_1}$ is the better one. As noted before, $\mathbf{c_1}$ is not the better translation.

**(iii)** Often there are many valid ways to translate a source sentence. This is particularly true for idiomatic phrases such as the previous example. The BLEU metric is designed to accommodate this flexibility: an n-gram in c is rewarded if it appears in any one of the reference translations. If we have multiple reference translations, the BLEU metric will thus reward similarity to any of the several valid translations. But if we only have one reference translation, the BLEU metric only recognizes similarity to that particular translation - potentially penalizing other valid candidate translations.

(iv)

- Advantage: BLEU is automatic, so it is fast to compute (unlike human evaluation, which is slow).
- Advantage: BLEU is automatic, so it is free to compute (unlike human evaluation, which is expensive).
- Advantage: BLEU has a concrete definition (unlike human evaluation, which is hard to define and varies depending on the human judge). This means that researchers can reproduce each others' BLEU results and use BLEU to compare different systems.
- Disadvantage: BLEU requires a reference translation (whereas human evaluation doesn't, assuming the human judges are bilingual), and optimally requires multiple reference translations.
- Disadvantage: BLEU is based on absolute n-gram matching, so it doesn't reward synonyms, paraphrases, or different inflections of the same word (e.g. make and makes).
- There are lots more disadvantages of BLEU vs human evaluation (basically any way in

which BLEU does not fully capture the true notion of good translation) - e.g. not having world knowledge, not knowing idioms, not recognizing what `sounds good' and what doesn't, etc.