Machine Translation and Encoder-Decoder Models

May 25, 2020

ddebarr@uw.edu

http://cross-entropy.net/ML530/Sequence-to-Sequence_Models.pdf

Speech and Language Processing 3rd ed draft

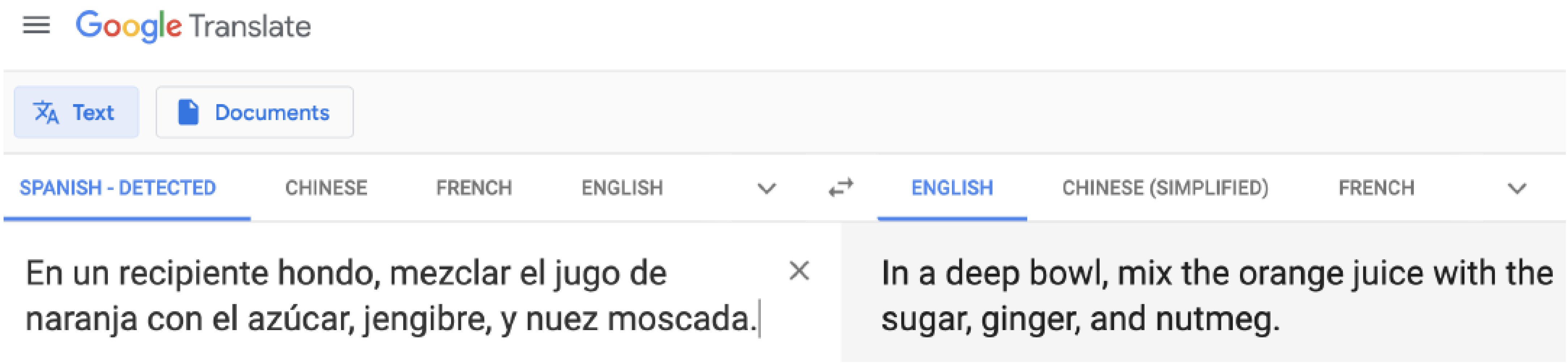
Chapter 11: Machine Translation and Encoder-Decoder Models

https://web.stanford.edu/~jurafsky/slp3/

1. Language Divergences and Typology
2. The Encoder-Decoder Model
3. Encoder-Decoder with RNNs
4. Attention
5. Beam Search
6. Some Practical Details on Building MT systems
7. MT Evaluation
8. Bias and Ethical Issues

Google Translate

"Google Translate alone translates hundreds of billions of words a day between over 100 languages."



Example Machine Translation (MT) Tasks

- Information Access
    - Recipes
    - News articles
    - Wikipedia pages
    - Government web site
- Computer-Assisted Translation (CAT); e.g. first pass at localization (adapting content to a new language)
- Optical Character Recognition (OCR) for menus and signs

Word Order Can Vary

- SVO vs SOV vs VSO languages

English: *He wrote a letter to a friend*	English is a Subject Verb Object (SVO) language
Japanese: *tomodachi ni tegami-o kaita* friend to letter wrote	Japanese is a Subject Object Verb (SOV) language
Arabic: *katabt risāla li ṡadq* wrote letter to friend	Arabic is a Verb Subject Object (VSO) language

- Chinese "exploration and peaceful using outer space conference" vs English "conference on the exploration and peaceful uses of outer space"

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 。

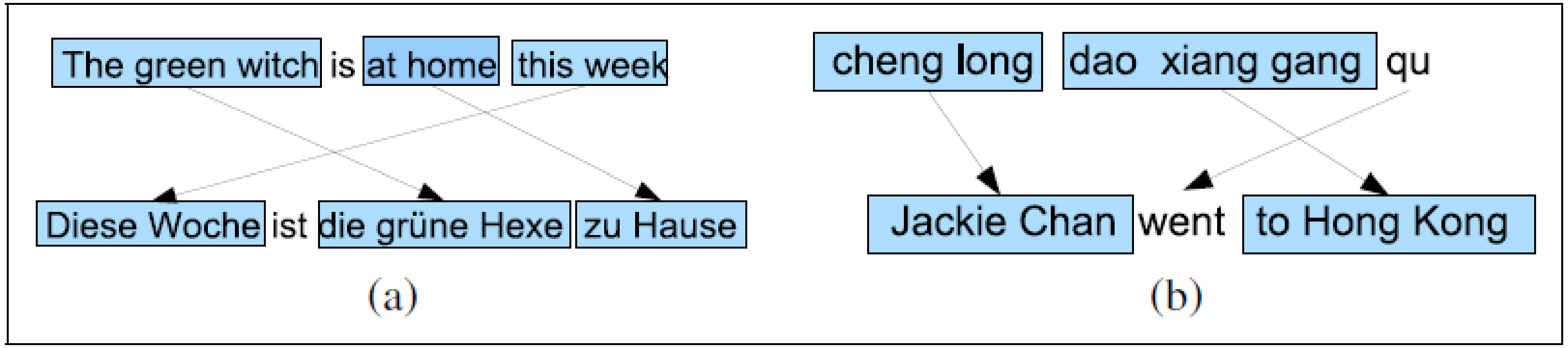
On 10 December 1982 , the General Assembly adopted resolution 37 in which it endorsed the recommendations of the Second United Nations Conference on the Exploration and Peaceful Uses of Outer Space .

Word Order Can Vary

- Adjective after vs before the noun

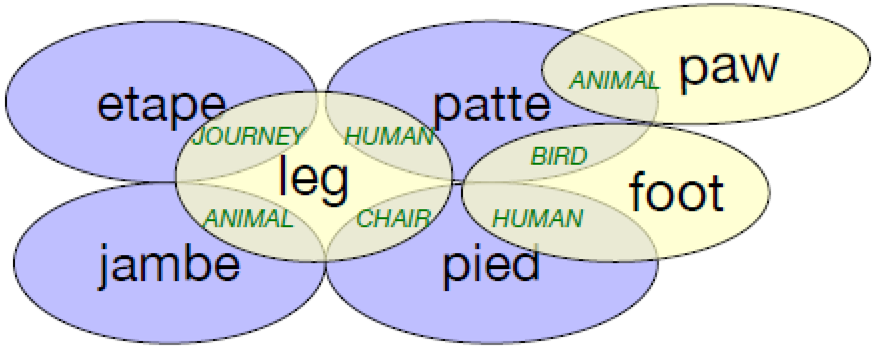
Spanish *bruja verde* **English** *green witch*

- Adverb moved to the initial position (English vs German); prepositional phrase before verb (Mandarin vs English)



Lexical Divergences

- Appropriate word can vary based on context
  - English “bass” in Spanish:
    - “lubina”, for the fish
    - “bajo”, for the musical instrument
  - English “leg”, “foot”, and “paw” translated to French



Lexical Divergences

- Lexical gap: target language may not contain the desired word
  - Mandarin “xiào” roughly translated to English as filial piety or loving child
- Direction of motion marked on the “verb” vs the “satellites” (particles, prepositional phrases, or adverbial phrases)
  - Direction marked on the particle “out” vs the verb

English:	*The bottle floated out.*			
Spanish:	La	botella	salió	flotando.
	The	bottle	exited	floating.

Morphological Typology and Referential Density

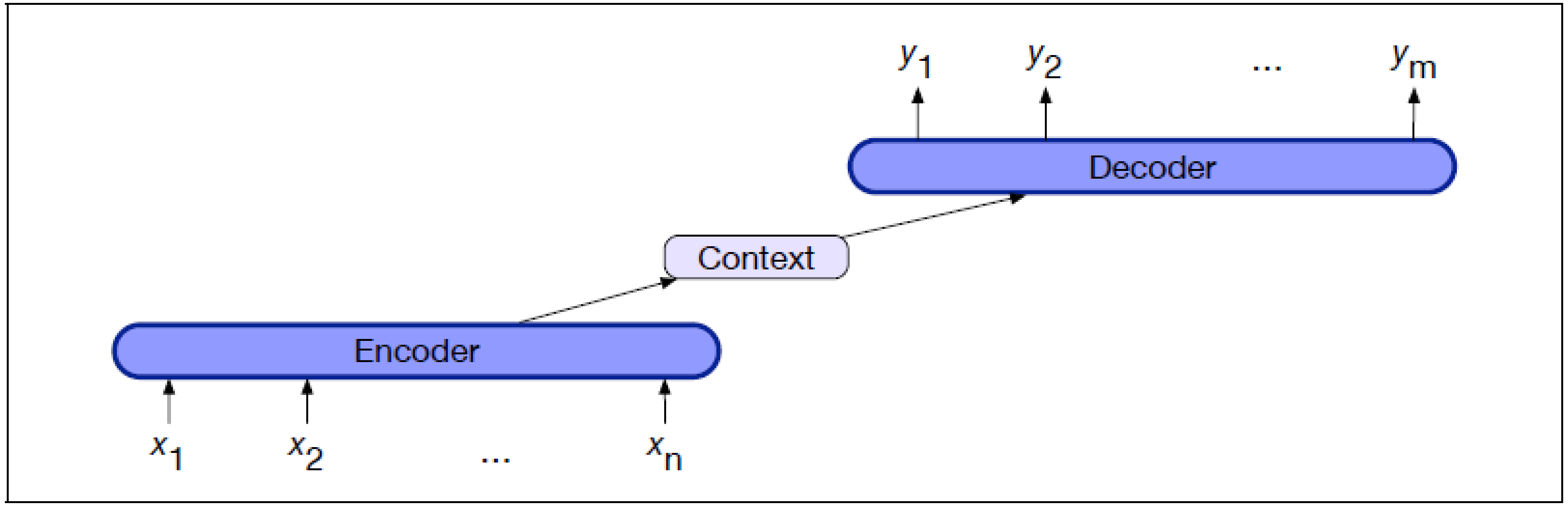
- Morphemes
  - Languages can vary by the number of morphemes per word; e.g. generally one for "isolating" languages such as Vietnamese or Cantonese to more than one for "polysynthetic" languages such as Siberian Yupik
  - Languages can vary based on the degree to which morphemes are segmentable (e.g. outgoing = out | go | ing)
    - Agglutinative languages such as Turkish have relatively clean boundaries for morphemes
    - Fusion languages such as Russian can conflate multiple morphemes into a single affix
- Referential Density
  - Languages that tend to use more pronouns are more "referentially dense"
  - Referentially sparse languages, like Chinese or Japanese, that require the hearer to do more inferential work to recover antecedents are called cold languages
  - Languages that are more explicit and make it easier for the hearer are called hot languages

Encoder-Decoder Architecture

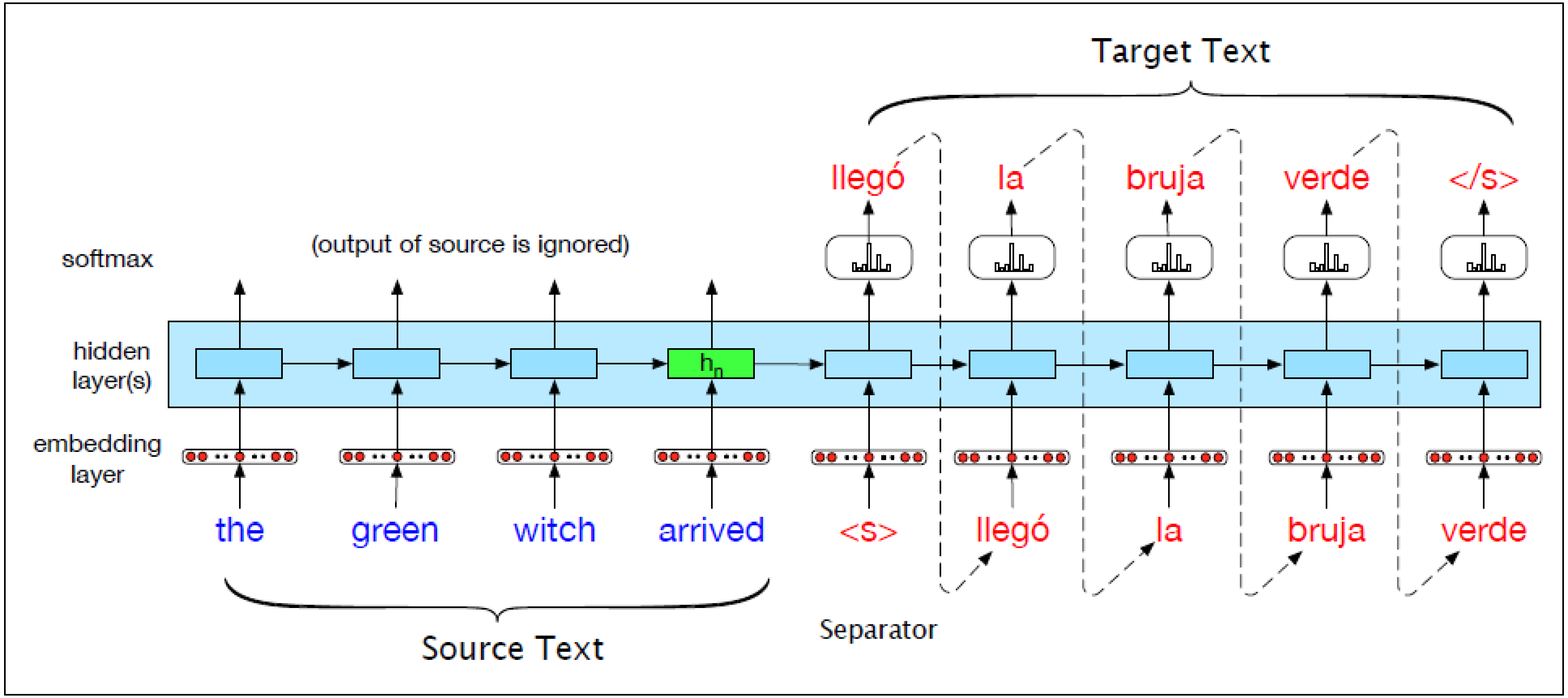
- Example Applications
  - Machine translation
  - Summarization (imagine applying this to text or a meeting)
  - Dialogue systems (imagine a chat bot that provides support)
  - Semantic parsing (imagine mapping words to a database query)
- Common encoder and decoder stacks
  - Recurrent Neural Networks
  - Convolutional Neural Networks
  - Transformers

Encoder Context Used by Decoder

- Encoder runs once to produce context for the input sequence
- Decoder is fed a "start" symbol, then run once per output token [continues until it generates a "stop" symbol]



Context Passed from Encoder to Decoder

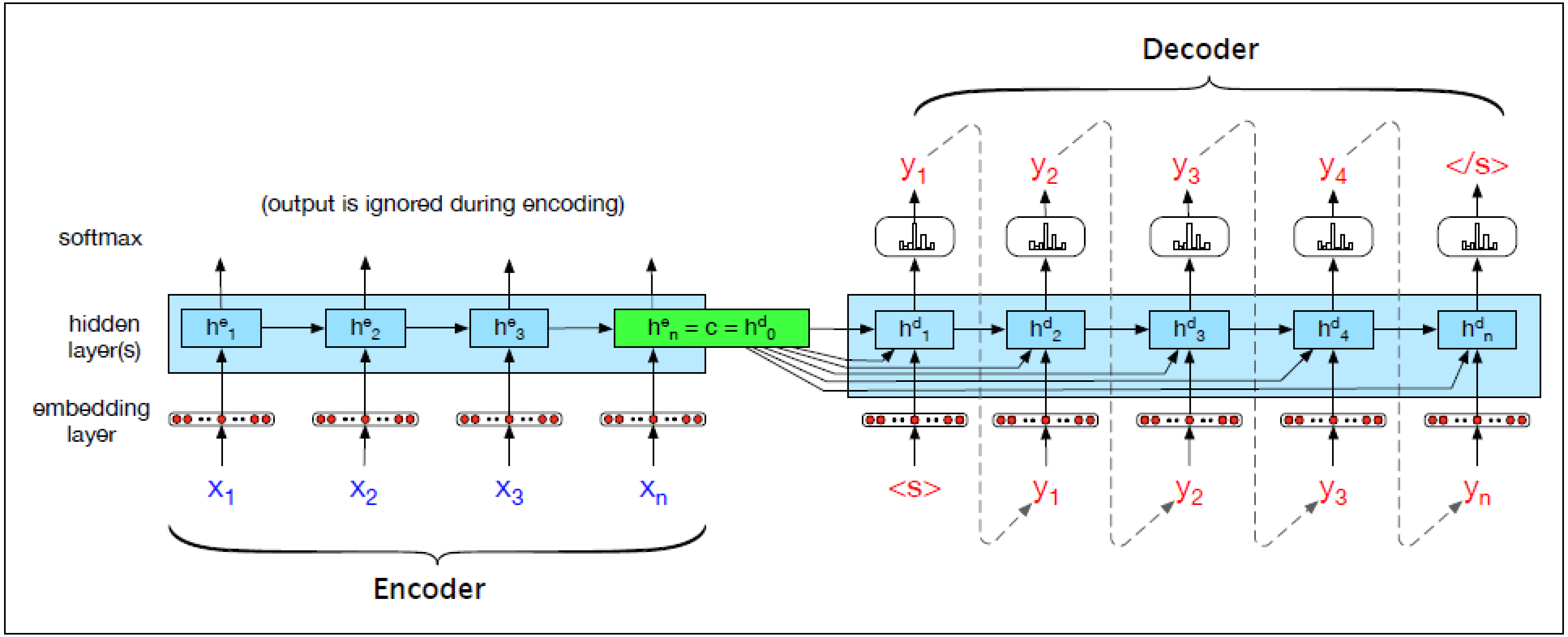


$$h_t = g(h_{t-1}, x_t)$$
$$y_t = f(h_t)$$

$$p(y|x) = p(y_1|x)p(y_2|y_1,x)p(y_3|y_1,y_2,x)...P(y_m|y_1,...,y_{m-1},x)$$

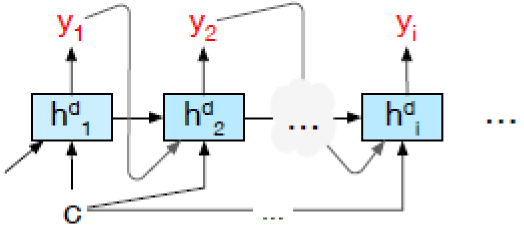
Context as Input for Each Decoder Step

Note that the context is now being fed as input to the decoder at each step



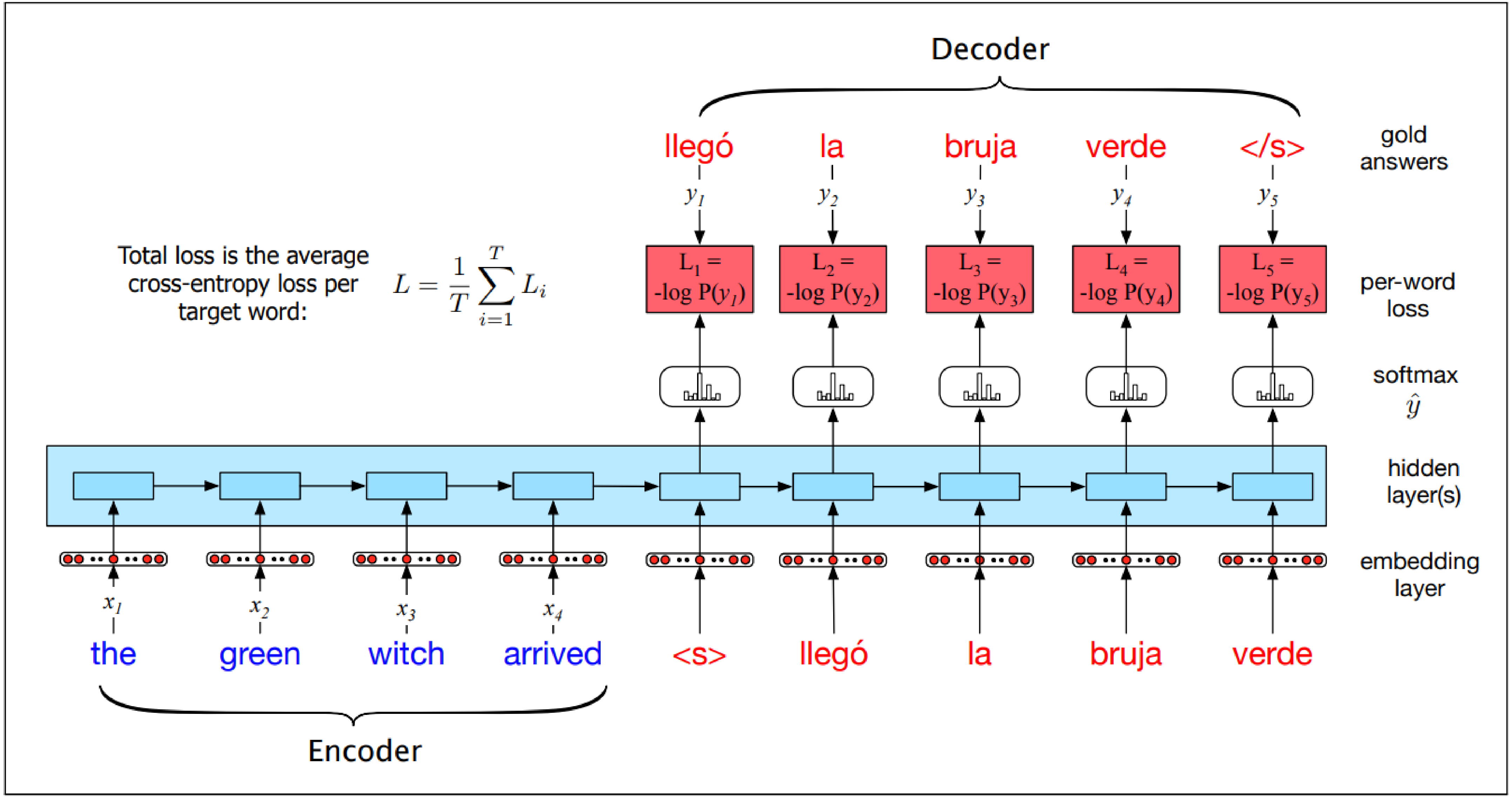
Source and target sequences do *not* have to be the same length

Context Provided for Each Decoder Position



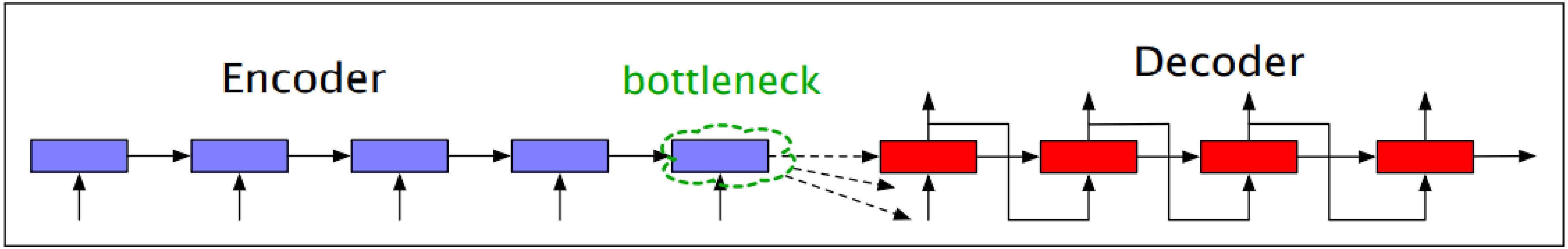
$$
\begin{aligned}
c &= h_n^e \\
h_0^d &= c \\
h_t^d &= g(\hat{y}_{t-1}, h_{t-1}^d, c) \\
z_t &= f(h_t^d) \\
y_t &= \mathrm{softmax}(z_t)
\end{aligned}
$$

Training the RNN Cells with Teacher Forcing



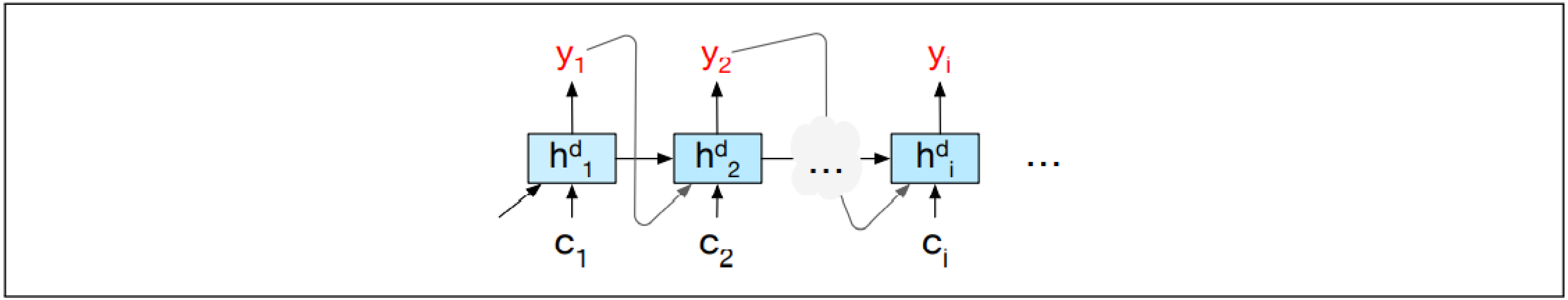
Motivation for Attention

Use of a single context vector to represent the entire input sequence can be viewed as a bottleneck (because we're using a small representation for a larger sequence)



Attention

Each decoder step gets its own weighted average of the encoder states



$$h_i^d = g(\hat{y}_{i-1}, h_{i-1}^d, c_i)$$

Dot Product Attention

We can say that the query (decoder) embeddings are being matched against the key (encoder) embeddings to produce weights to compute weighted averages of the value (encoder) embeddings

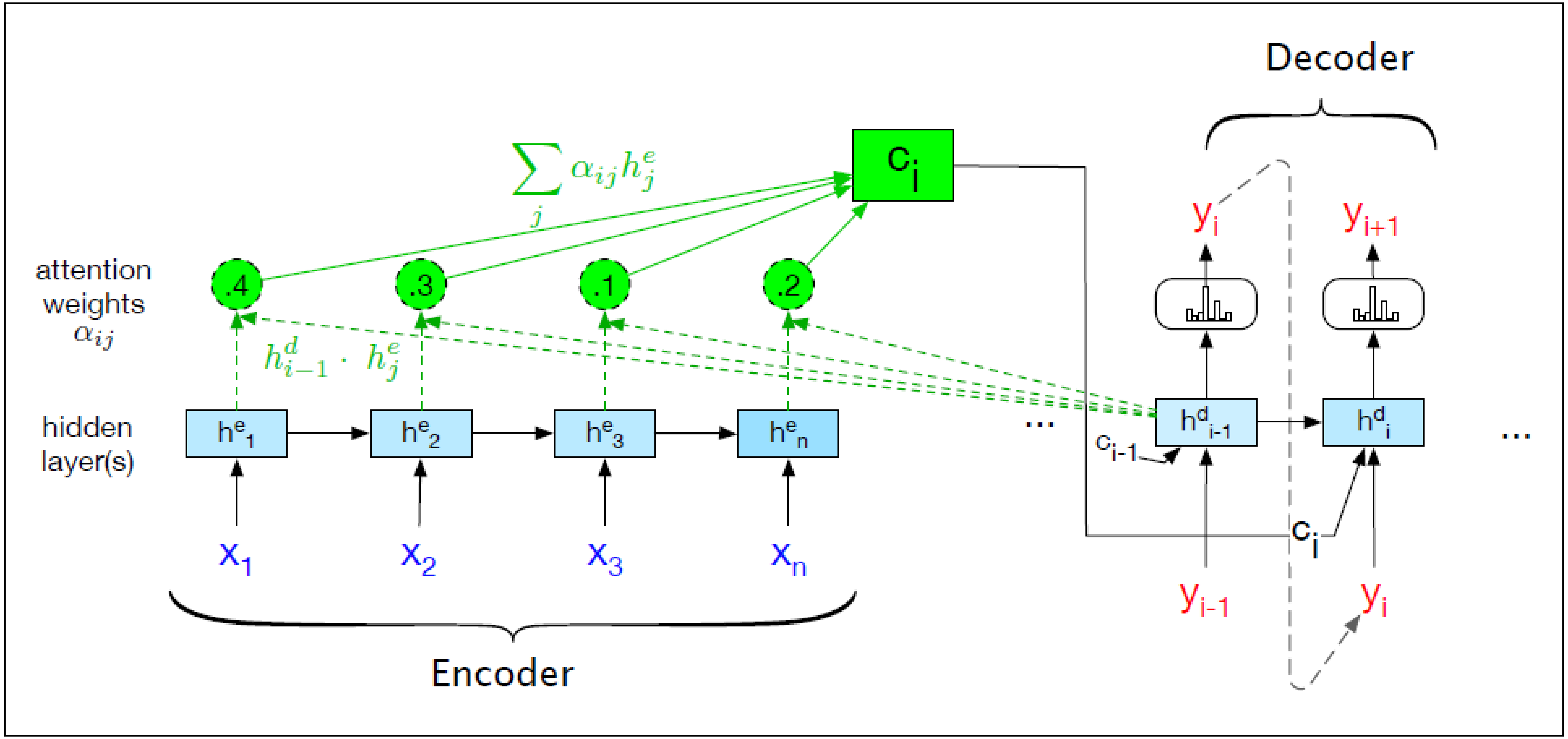
$$score(h_{i-1}^d, h_j^e) = h_{i-1}^d \cdot h_j^e$$

$$\alpha_{ij} = \text{softmax}(score(h_{i-1}^d, h_j^e)\ \forall j \in e)$$

$$= \frac{\exp(score(h_{i-1}^d, h_j^e))}{\sum_k \exp(score(h_{i-1}^d, h_k^e))}$$

$$c_i = \sum_j \alpha_{ij}\, h_j^e$$

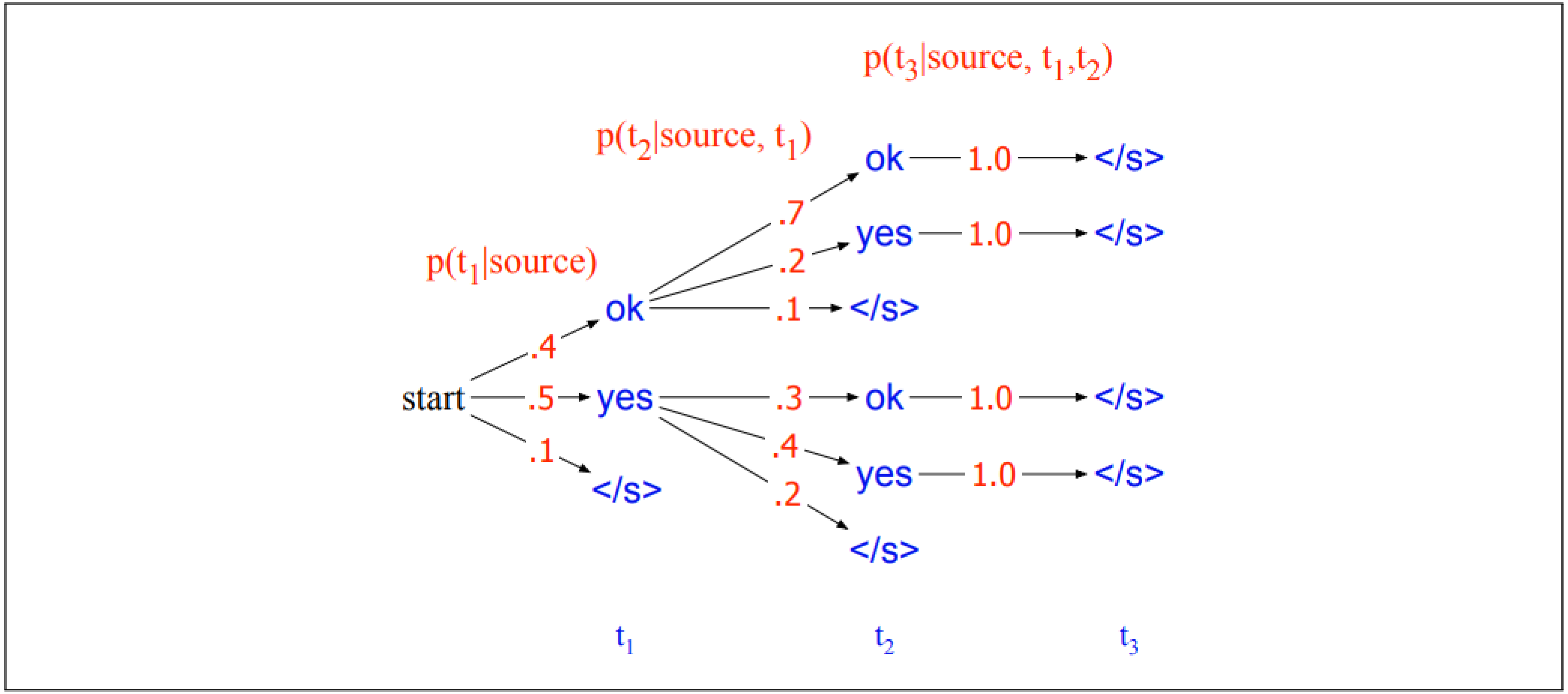
Computing Context c_i



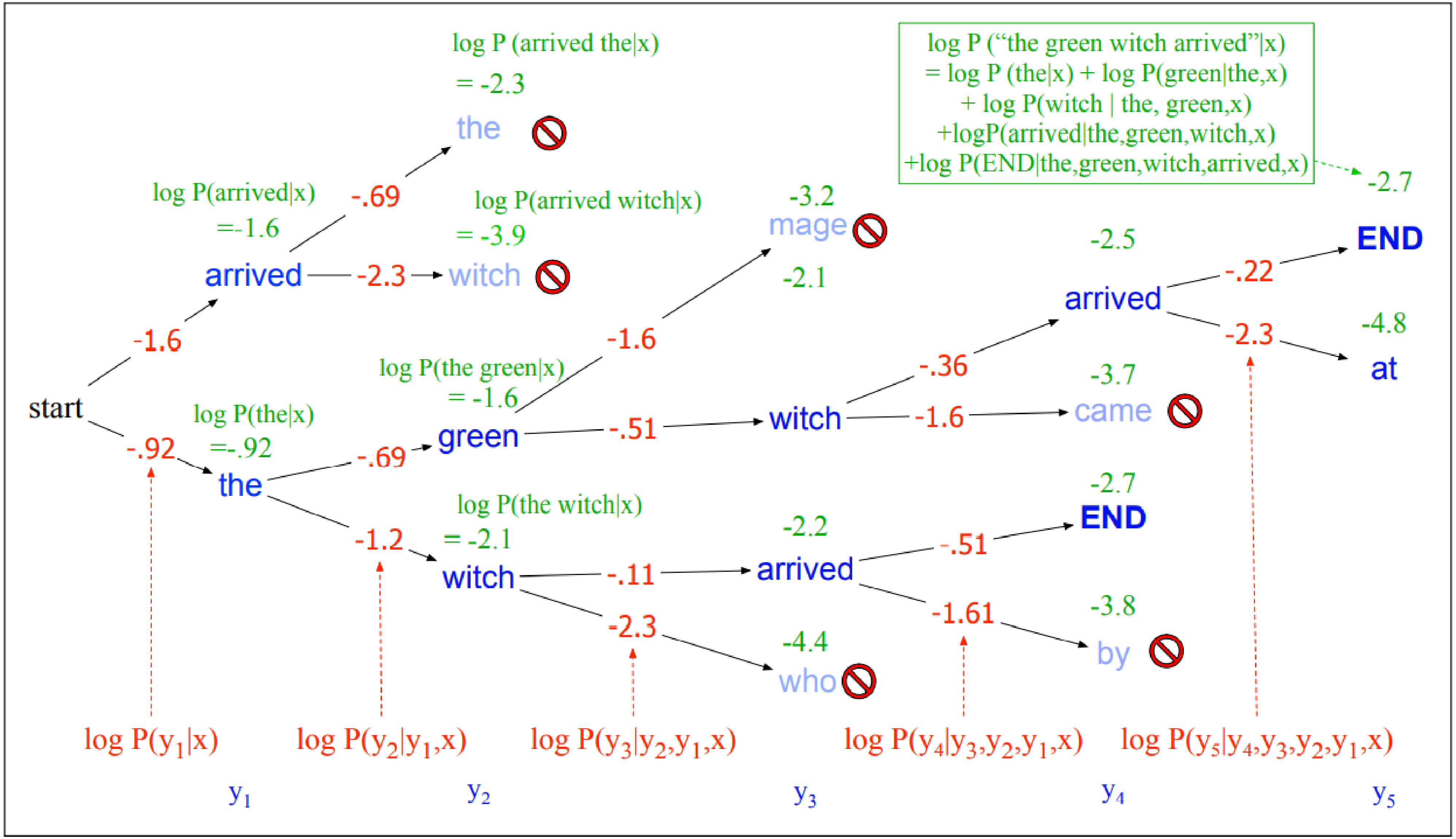
$$score(h^d_{i-1}, h^e_j) = h^d_{t-1} W_s h^e_j$$

Can also train a weight matrix; allowing embeddings of different sizes

Search: Greedy vs Beam vs Monte Carlo



Beam Decoding with Beam Width = 2



Length Normalization

Without normalization

$$
\begin{aligned}
score(y) &= \log P(y|x) \\
&= \log\left(P(y_1|x)P(y_2|y_1,x)P(y_3|y_1,y_2,x)\ldots P(y_t|y_1,\ldots,y_{t-1},x)\right) \\
&= \sum_{i=1}^{t} \log P(y_i|y_1,\ldots,y_{i-1},x)
\end{aligned}
$$

With normalization, where ‘T’ = target sequence length

$$
score(y) = -\log P(y|x) \;=\; \frac{1}{T}\sum_{i=1}^{t} -\log P(y_i|y_1,\ldots,y_{i-1},x)
$$

Beam Search Pseudocode

```
function BEAMDECODE(c, beam_width) returns best paths

  y0, h0 ← 0
  path ← ()
  complete_paths ← ()
  state ← (c, y0, h0, path)          ;initial state
  frontier ← ⟨state⟩        ;initial frontier

  while frontier contains incomplete paths and beamwidth > 0
     extended_frontier ← ⟨⟩
     for each state ∈ frontier do
        y ← DECODE(state)
        for each word i ∈ Vocabulary do
           successor ← NEWSTATE(state, i, y_i)
           new_agenda ← ADDTOBEAM(successor, extended_frontier, beam_width)

     for each state in extended_frontier do
        if state is complete do
           complete_paths ← APPEND(complete_paths, state)
           extended_frontier ← REMOVE(extended_frontier, state)
           beam_width ← beam_width - 1
     frontier ← extended_frontier

  return completed_paths
```

```
function NEWSTATE(state, word, word_prob) returns new state

function ADDTOBEAM(state, frontier, width) returns updated frontier

  if LENGTH(frontier) < width then
     frontier ← INSERT(state, frontier)
  else if SCORE(state) > SCORE(WORSTOF(frontier))
     frontier ← REMOVE(WORSTOF(frontier))
     frontier ← INSERT(state, frontier)
  return frontier
```

Tokenization: Byte Pair Encoding (BPE)

function BYTE-PAIR ENCODING(strings C, number of merges k) **returns** vocab V

$V \leftarrow$ all unique characters in C # initial set of tokens is characters
for $i = 1$ **to** k **do** # merge tokens til k times
  $t_L, t_R \leftarrow$ Most frequent pair of adjacent tokens in C
  $t_{NEW} \leftarrow t_L + t_R$ # make new token by concatenating
  $V \leftarrow V + t_{NEW}$ # update the vocabulary
  Replace each occurrence of t_L, t_R in C with t_{NEW} # and update the corpus
return V

Tokenization: WordPiece

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

We gave the BPE algorithm in detail in Chapter 2; here's more details on the wordpiece algorithm, which 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.

A vocabulary of 8K to 32K word pieces is commonly used.

'_' marks the beginning of a word; Jet and feud are split into subtokens;
See https://huggingface.co/transformers/tokenizer_summary.html

Example Machine Translation Corpora

- Parallel corpus (sometimes called bitext): text appears in two (or more) languages
- Europarl: proceedings of the European parliament
- OpenSubtitles: subtitles from movies and television
- ParaCrawl: extracted from the Common Crawl dataset

Sentence Alignment

Given two documents that are translations of each other, we generally need two steps to produce sentence alignments:

- 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.
- an alignment algorithm that takes these scores to find a good alignment between the documents.

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

where x, y denote one or more sequential sentences from the source/target document; $\cos(x,y)$ is the cosine similarity between embeddings[2] of x, y; $\mathrm{nSents}(x)$, $\mathrm{nSents}(y)$ denote the number of sentences in x, y; and $x_1,\ldots,x_S$, $y_1,\ldots,y_S$ are sampled uniformly from the given document.

Dynamic programming is usually used for the alignment algorithm

Backtranslation

- Parallel corpora may be limited for particular languages or domains, but we can often find a large monolingual corpus in the target language
  - Train a target-to-source translation model using your current bitext corpus
  - Predict translations for the monolingual corpus
  - Add these pairs to your training corpus for constructing the source-to-target translation model
- "one estimate suggests that a system trained on backtranslated text gets about 2/3 of the gain as would training on the same amount of natural bitext" (Edunov et al, 2018)

Machine Translation Evaluation

- Translations can be evaluated along two dimensions:
  - adequacy: how well the translation captures the exact meaning of the source sentence; sometimes called faithfulness or fidelity
  - fluency: how fluent the translation is in the target language (is it grammatical, clear, readable, natural)
- Humans can be asked to either …
  - Rate the proposed translation on a multi-point scale; e.g. 5-point or 100-point scales
  - Rate which translation of a pair of translations they prefer
- For monolingual raters, we can ask them to compare the proposed translation to a reference translation

BiLingual Evaluation Understudy (BLEU) Score

Source

la verdad, cuya madre es la historia, émula del tiempo, depósito de las acciones, testigo de lo pasado, ejemplo y aviso de lo presente, advertencia de lo por venir.

Reference

truth, whose mother is history, rival of time, storehouse of deeds, witness for the past, example and counsel for the present, and warning for the future.

Candidate 1

truth, whose mother is history, voice of time, deposit of actions, witness for the past, example and warning for the present, and warning for the future

Candidate 2

the truth, which mother is the history, émula of the time, deposition of the shares, witness of the past, example and notice of the present, warning of it for coming

Unigram Precision Example

- Without “clipping”: 23 / 26 = 88.5%
- With “clipping”: 22 / 26 = 84.6% [modified n-gram precision]

	Token Counts		
	Prediction	Reference	Matches
actions	1	0	
and	2	2	2
deposit	1	0	
example	1	1	1
for	3	3	3
future	1	1	1
history	1	1	1
is	1	1	1
mother	1	1	1
of	2	2	2
past	1	1	1
present	1	1	1
the	3	3	3
time	1	1	1
truth	1	1	1
voice	1	0	
warning	2	1	2
whose	1	1	1
witness	1	1	1

Reference:
truth, whose mother is history, rival of time, storehouse of deeds, witness for the past, example and counsel for the present, and warning for the future.

vs

Prediction:
truth, whose mother is history, voice of time, deposit of actions, witness for the past, example and warning for the present, and warning for the future

BLEU Expression

$$\text{prec}_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} \text{Count}_{\text{match}}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} \text{Count}(n\text{-}gram')}$$

$$\text{BP} = \min\left(1, \exp\left(1 - \frac{\text{ref\_len}}{\text{sys\_len}}\right)\right)$$

$$\text{BLEU} = \text{BP} \times \left(\prod_{n=1}^{4} \text{prec}_n\right)^{\frac{1}{4}}$$

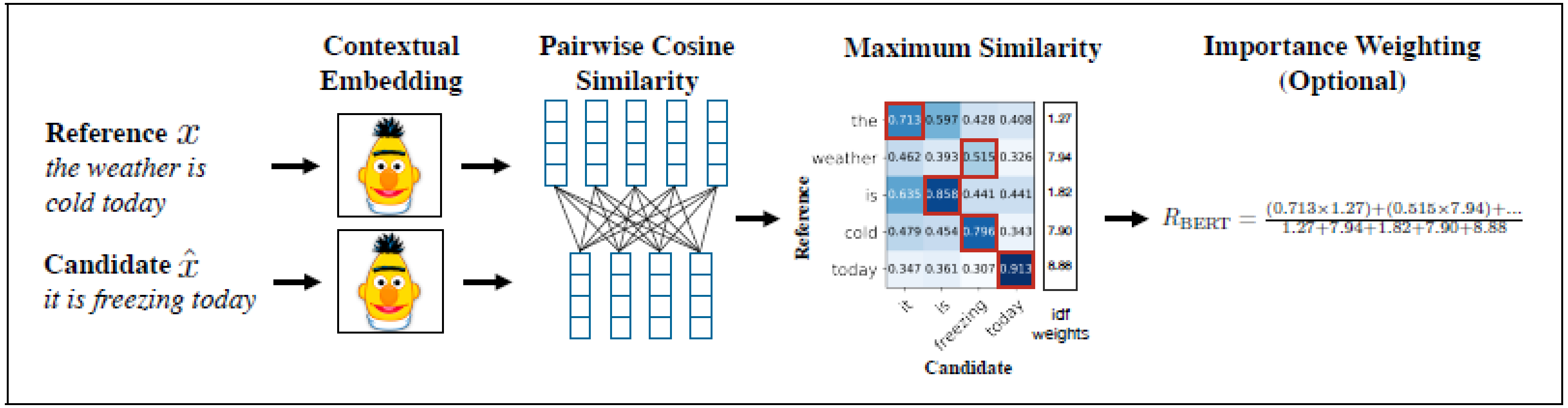
BP: Brevity Penalty

Geometric mean of modified n-gram precisions

Embedding-Based Evaluation

BERT Score:

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



Bias

When translating from a gender neutral language like Hungarian into English, MT systems assign gender based on whether an occupation is male-dominated or female-dominated

Hungarian (gender neutral) source	English MT output
ő egy ápoló	she is a nurse
ő egy tudós	he is a scientist
ő egy mérnök	he is an engineer
ő egy pék	he is a baker
ő egy tanár	she is a teacher
ő egy esküvőszervező	she is a wedding organizer
ő egy vezérigazgató	he is a CEO

WinoMT challenge: MT systems do worse when they are asked to translate sentences that describe people with non-stereotypical gender roles, like "The doctor asked the nurse to help her in the operation."

Ethical Issues

- MT systems need ways to assign confidence values to candidate translations, so they can avoid giving an incorrect translation that may cause harm (e.g. in urgent medical and legal situations)
- MT systems need low-resource algorithms that can help improve translation for languages that do not have large parallel corpora available for training

Speech and Language Processing 3rd ed draft

Chapter 11: Machine Translation and Encoder-Decoder Models

https://web.stanford.edu/~jurafsky/slp3/

1. Language Divergences and Typology
2. The Encoder-Decoder Model
3. Encoder-Decoder with RNNs
4. Attention
5. Beam Search
6. Some Practical Details on Building MT systems
7. MT Evaluation
8. Bias and Ethical Issues

English-to-French Translation

- Translating an English sentence to a French sentence, using characters as tokens
  https://keras.io/examples/nlp/lstm_seq2seq/
  - There are two LSTM cells: one for the encoder and one for the decoder
  - Greedy search used for decoding
- Another example, based on a keras example that seems to have disappeared from the Internet:
  https://github.com/serge-sotnyk/seq2seq-compress/blob/master/cnn_seq2seq.ipynb
  Note how attention is used in this example …
    - We take the dot product of each decoder output vector (one per output sequence time step; queries) and each encoder output vector (one per input sequence time step; keys)
    - The softmax activation changes the rows to weights that sum to one; i.e. attention weights
    - We now use the attention weights to compute a separate weighted average of encoder outputs (values) for each decoder time step, then concatenate the decoder output at each time step with the weighted average of encoder outputs for that time step

```
attention = Dot(axes=[2, 2])([x_decoder, x_encoder])
attention = Activation('softmax')(attention)
context = Dot(axes=[2, 1])([attention, x_encoder])
decoder_combined_context = Concatenate(axis=-1)([context, x_decoder])
```

BlenderBot: Chatbot Example

- Blog post
  https://ai.facebook.com/blog/state-of-the-art-open-source-chatbot/
- Install and run
  conda install pytorch torchvision cudatoolkit=10.1 -c pytorch
  git clone https://github.com/facebookresearch/ParlAI.git
  cd ParlAI
  python setup.py develop --user
  pip install 'git+https://github.com/rsennrich/subword-nmt.git#egg=subword-nmt'
  python parlai/scripts/safe_interactive.py -t blended_skill_talk -mf zoo:blender/blender_90M/model

[context]: your persona: i love to dance.

your persona: i'm a teacher.

Enter Your Message: Have you ever tried an interpretive dance class?

[TransformerGenerator]: no i haven ' t , but i ' d love to try it . what do you do for fun ?

Enter Your Message: I love to do the twist. How about you?

[TransformerGenerator]: i like to watch hockey . it ' s my favorite . what ' s your favorite sport to watch ?