CSCI 1470/2470
Spring 2024

Ritambhara Singh

March 06, 2024
Wednesday

Seq2Seq modeling

Deep Learning

ChatGPT prompt “minimalist landscape painting of a deep underwater scene with a blue tang fish in the bottom right corner”

Review: Natural Language Prediction Tasks

"They went to the grocery store and **bought… bread?**

milk?

rock?

Generating artificial sentences: Here each word is a discrete unit; predicting the next part of the sequence means predicting words

Review: Natural Language Prediction Tasks

Input: X

Output: Y

"Good review?"

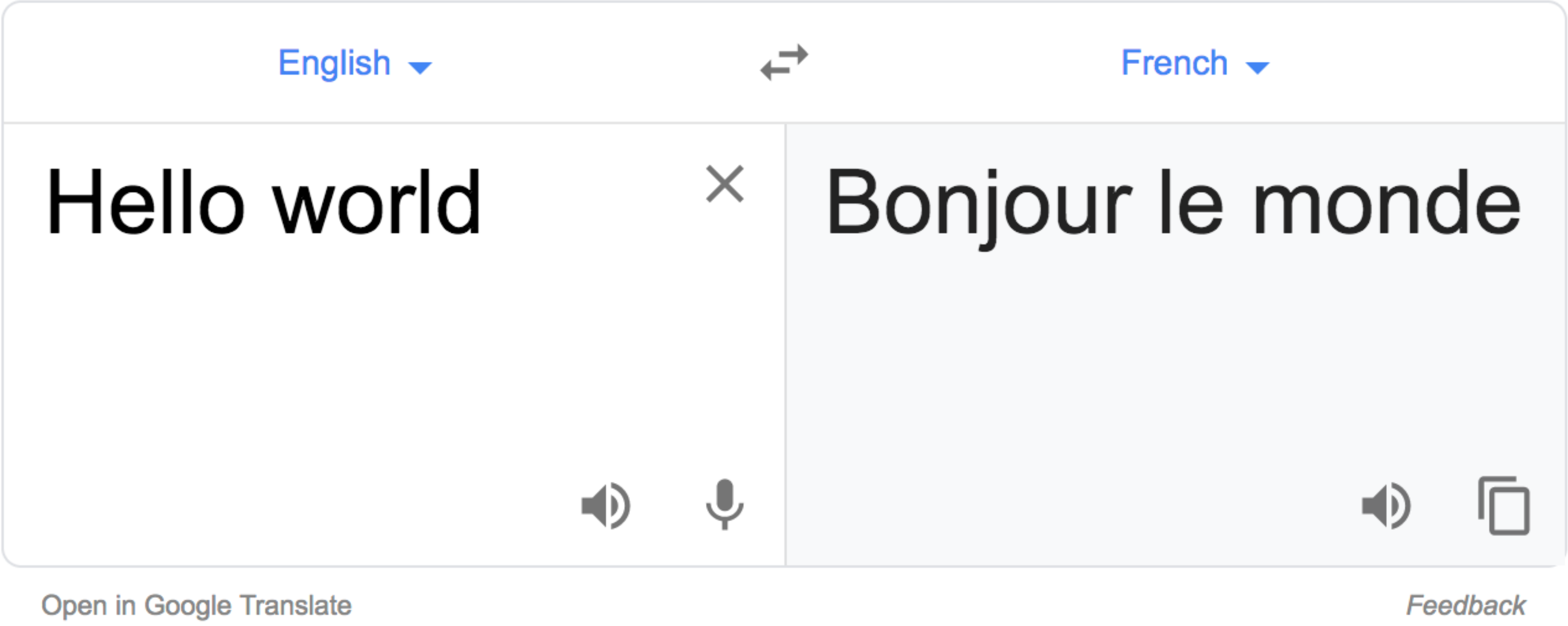
"The story telling was erratic and, at times, slow"

Function: f

"Loved the diverse cast of this movie"

Machine Translation (MT)

Software that transforms text in a source language into text in a target language



Why is this an interesting problem to solve?

- **Complex:** languages evolve rapidly and don’t have a clear and well-defined structure
  - Example of language change: “awful” originally meant “full of awe”, but is now strictly negative
- **Important**: billions per year spent on translation services
  - >CA$2.4 billion spent per year by Canadian government
  - >£100 million spent per year by UK government

- Original approach: create rule-based MT programs

- ***Why doesn't this work?***

Why rule-based MT doesn't work (1/3)

Basic rules are regularly broken

Example rule: in English, adjectives come before nouns

"black cat", "large building", etc.

Exception: "something"

"something black", "something large", etc.

Why rule-based MT doesn’t work (2/3)

Too many language pairs: Google Translate has 133 languages, or 8,778 pairs

Thus would require 8,778 sets of rules to cover all that Google Translate does

Detect language	Czech	Hebrew	Latin	Portuguese	Tajik
Afrikaans	Danish	Hindi	Latvian	Punjabi	Tamil
Albanian	Dutch	Hmong	Lithuanian	Romanian	Telugu
Amharic	English	Hungarian	Luxembourgish	Russian	Thai
Arabic	Esperanto	Icelandic	Macedonian	Samoan	Turkish
Armenian	Estonian	Igbo	Malagasy	Scots Gaelic	Ukrainian
Azerbaijani	Filipino	Indonesian	Malay	Serbian	Urdu
Basque	Finnish	Irish	Malayalam	Sesotho	Uzbek
Belarusian	French	Italian	Maltese	Shona	Vietnamese
Bengali	Frisian	Japanese	Maori	Sindhi	Welsh
Bosnian	Galician	Javanese	Marathi	Sinhala	Xhosa
Bulgarian	Georgian	Kannada	Mongolian	Slovak	Yiddish
Catalan	German	Kazakh	Myanmar (Burmese)	Slovenian	Yoruba
Cebuano	Greek	Khmer	Nepali	Somali	Zulu
Chichewa	Gujarati	Korean	Norwegian	Spanish	
Chinese	Haitian Creole	Kurdish (Kurmanji)	Pashto	Sundanese	
Corsican	Hausa	Kyrgyz	Persian	Swahili	
Croatian	Hawaiian	Lao	Polish	Swedish	

Why rule-based MT doesn't work (3/3)

Translations depend on context, and words shouldn't always be translated literally

	Spanish	**English**
Apertium (rule-based)	Me llamo John	I call me John

Why rule-based MT doesn't work (3/3)

Translations depend on context, and words shouldn't always be translated literally

	Spanish	**English**
Apertium (rule-based)	Me llamo John	I call me John
Google Translate	Me llamo John	My name is John

- Original approach: create rule-based MT programs (doesn’t work well!)

- **Deep learning can help!**
  - Instead of telling the computer rules, it could learn them for itself

What is the first thing we need?

Parallel Corpora

- We need pairs of equivalent sentences in two languages, called ***parallel corpora***

Canadian Hansards

- Hansards are transcripts of parliamentary debates
- Canada's official languages are English and French, so everything said in parliament is transcribed in both languages

Canadian Hansards: Examples

English	French
What a past to celebrate.	Nous avons un beau passé à célébrer.
We are about to embark on a new era in health research in this country.	Le Canada est sur le point d'entrer dans une nouvelle ère en matière de recherche sur la santé.

Canadian Hansards

- We can use this as a dataset for MT!
- Not perfect:
  - *Translations aren't literal*: in the example, "this country" is translated to "Le Canada"
  - *Biased in style*: not everyone speaks like politicians in parliamentary debate
  - *Biased in content*: some topics are never discussed in parliament

Other parallel corpora

- Europarl, a parallel corpus of 21 languages used in the European Parliament
- EUR-Lex, a parallel corpus of 24 languages used in EU law and public documents
- Japanese-English Bilingual Corpus of Wikipedia's Kyoto Articles

Any questions?

Problems with parallel corpora

- Expensive to produce
- Tend to be biased towards particular types of text – e.g. government documents containing formal language
- Translations aren’t necessarily literal - e.g. “this country” -> “Le Canada”
- Parallel corpora are necessary, **but never perfect**

Implementing learning-based MT

We have our dataset!

Example from Hansards

- For example, take the first entry in Hansard's:

edited hansard number 1

hansard révisé numéro 1

LM approach

- Language modelling works on a word-by-word basis, taking only previous words as input

$$P(w_{t,i}) = P(w_{t,i} \mid w_{s,i-1}, w_{s,i-2}, \dots, w_{s,0})$$

- Where $w_{t,i}$ is the i^{th} word in the target sentence, and $w_{s,\,i}$ is the i^{th} word in the source sentence

Will it work for MT task?

Why our LM approach doesn't work for MT

- Language modelling works on a word-by-word basis, taking only previous words as input

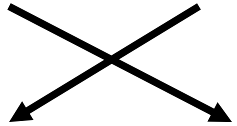
$$P(w_{t,i}) = P(w_{t,i} \mid w_{s,i-1}, w_{s,i-2}, \dots, w_{s,0})$$

- Where $w_{t,i}$ is the i^{th} word in the target sentence, and $w_{s,\,i}$ is the i^{th} word in the source sentence
- However, **it is not a given that the information we need comes in the preceding words**
- The order and length of the source and target sentences are not necessarily equal

Example from Hansards

- For example, take the first entry in Hansard's:

edited hansard number 1



hansard révisé numéro 1

What should we do?

Further examples

French: “Londres me manque”

Naive translation: “London I miss”

Correct translation: “I miss London”

French: “Je viens de partir”

Naive translation: “I come of to go”

Correct translation: “I just left”

Sequence to Sequence (seq2seq)

Thus, we cannot simply use the previous words – we need to ***summarize the source sentence first***

This is called ***sequence to sequence learning***, or ***seq2seq***

Sequence to Sequence (seq2seq)

Instead of:

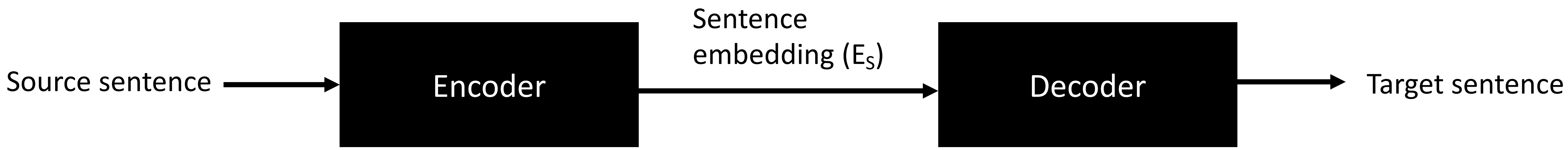
$$P\left(w_{i,t}\right) = P\left(w_{i,t} \,\middle|\, w_{i-1,s}, w_{i-2,s}, \dots, w_{0,s}\right)$$

Let's do:

$$P\left(w_{i,t}\right) = P\left(w_{i,t} \,\middle|\, E_S, w_{i-1,t}, w_{i-2,t}, \dots, w_{0,t}\right)$$

Where E_S is a summary, or ***embedding***, of the sentence taken from the source language, and w_i is the i^{th} word of the sentence in the target language

What will the neural net look like?

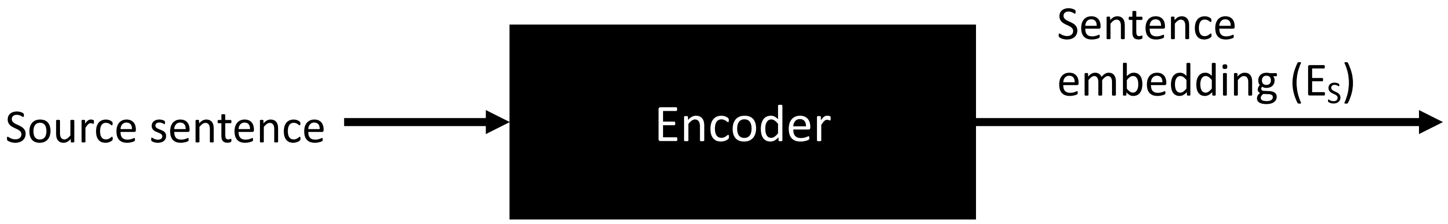


Origin of the encoder/decoder terminology: information theory

- The encoder “compresses” the source sentence into a compact “code”
- The decoder recovers the sentence (but in the target language) from this code

What will the neural net look like?

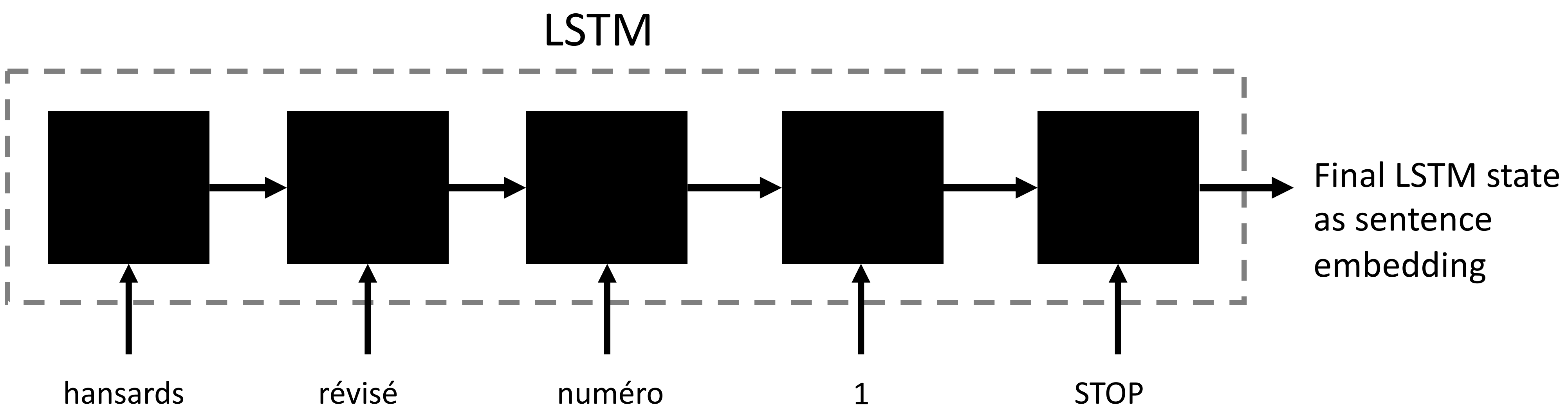
Any ideas?



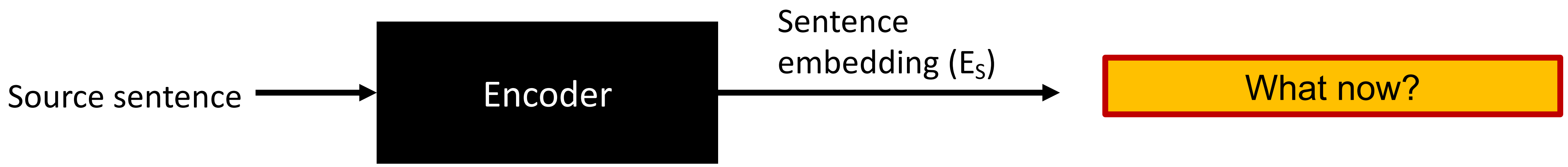
Encoder

- To generate the sentence embedding, we need an encoder
- Use an LSTM
- Feed in the source sentence
- Take the final LSTM state as the sentence embedding
- This will be a ***language-agnostic*** representation of the sentence
  - i.e. it will represent the *meaning* of the sentence without being tied to any particular language

Encoder architecture

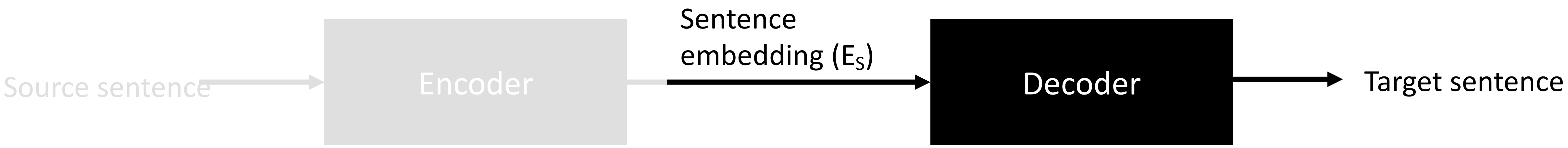


What will the neural net look like?



What will the neural net look like?

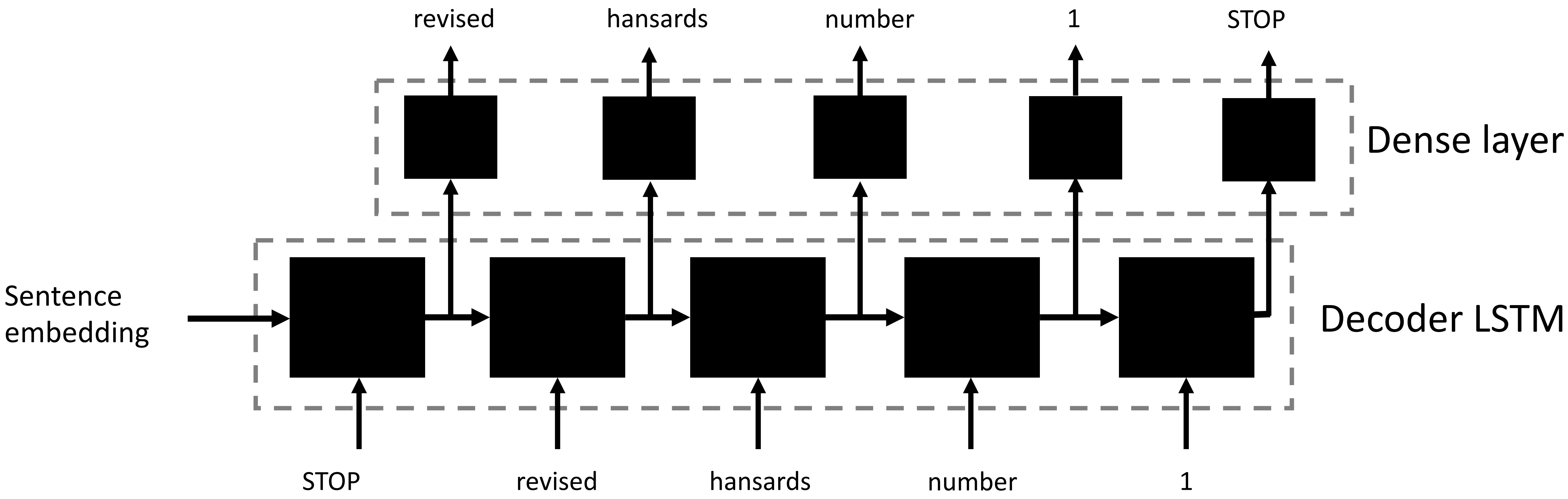
Any ideas?



Decoder

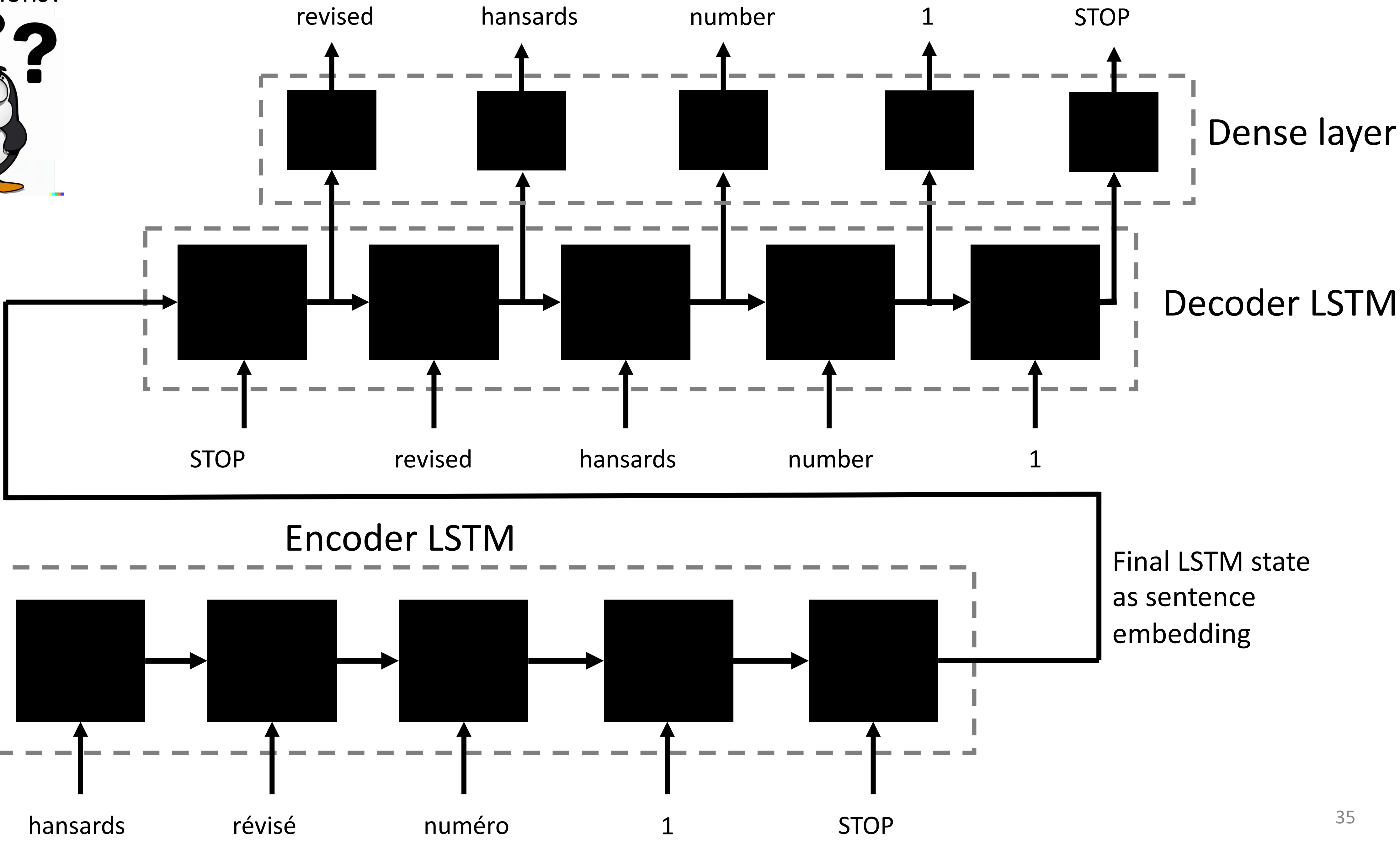
- We now have a sentence embedding representing the meaning of the source sentence
- Now, let's generate a sentence in the target language with the same meaning
- Use an LSTM again, **with the sentence embedding** as its initial hidden state
- The rest is just like language modeling:
  - Input to the LSTM is the previous word from the target sentence
  - Take each LSTM output and put it through a fully connected layer
  - Softmax to convert to probability distribution over next word in target language

Decoder architecture



Putting it all together…

Any questions?

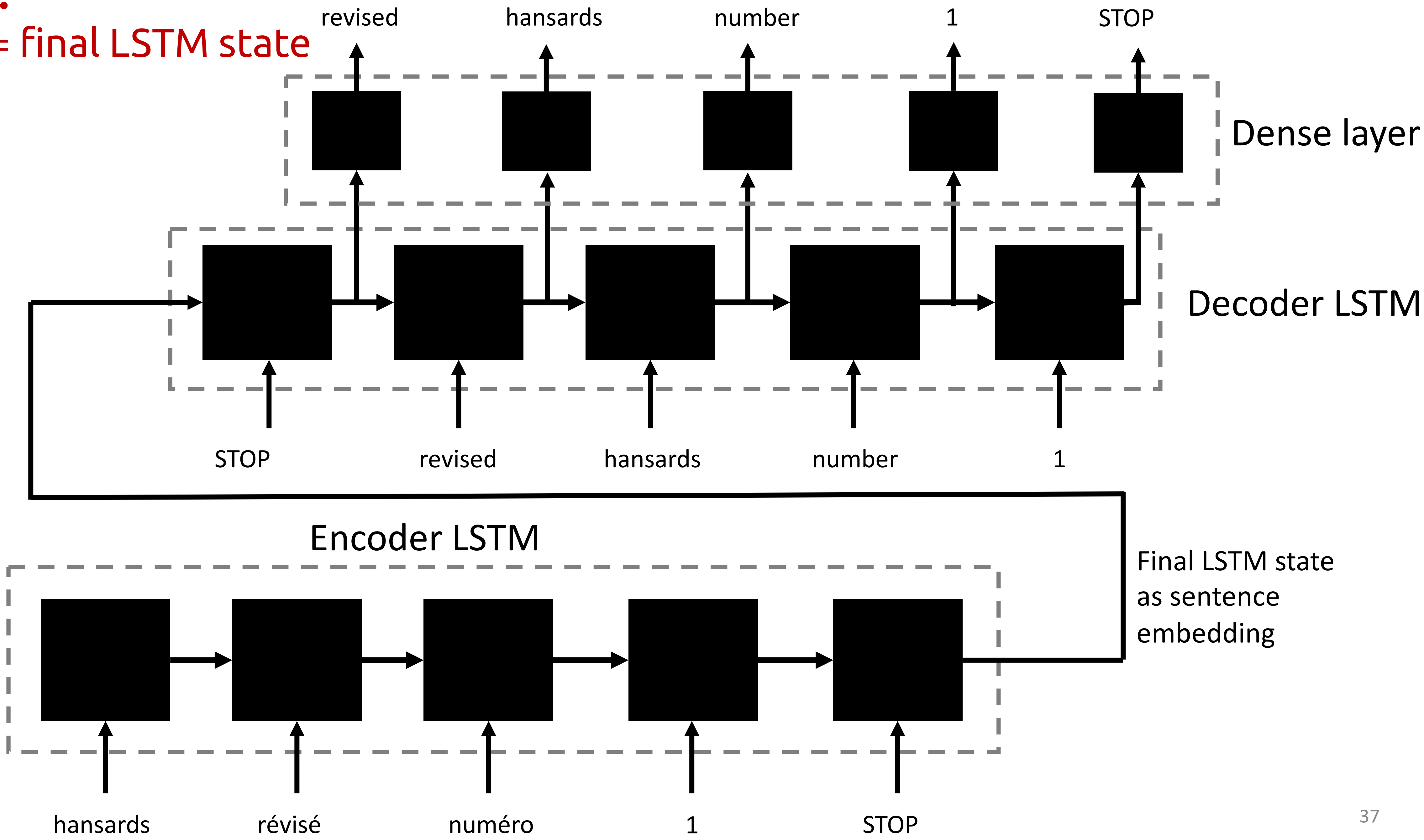


Architecture variations

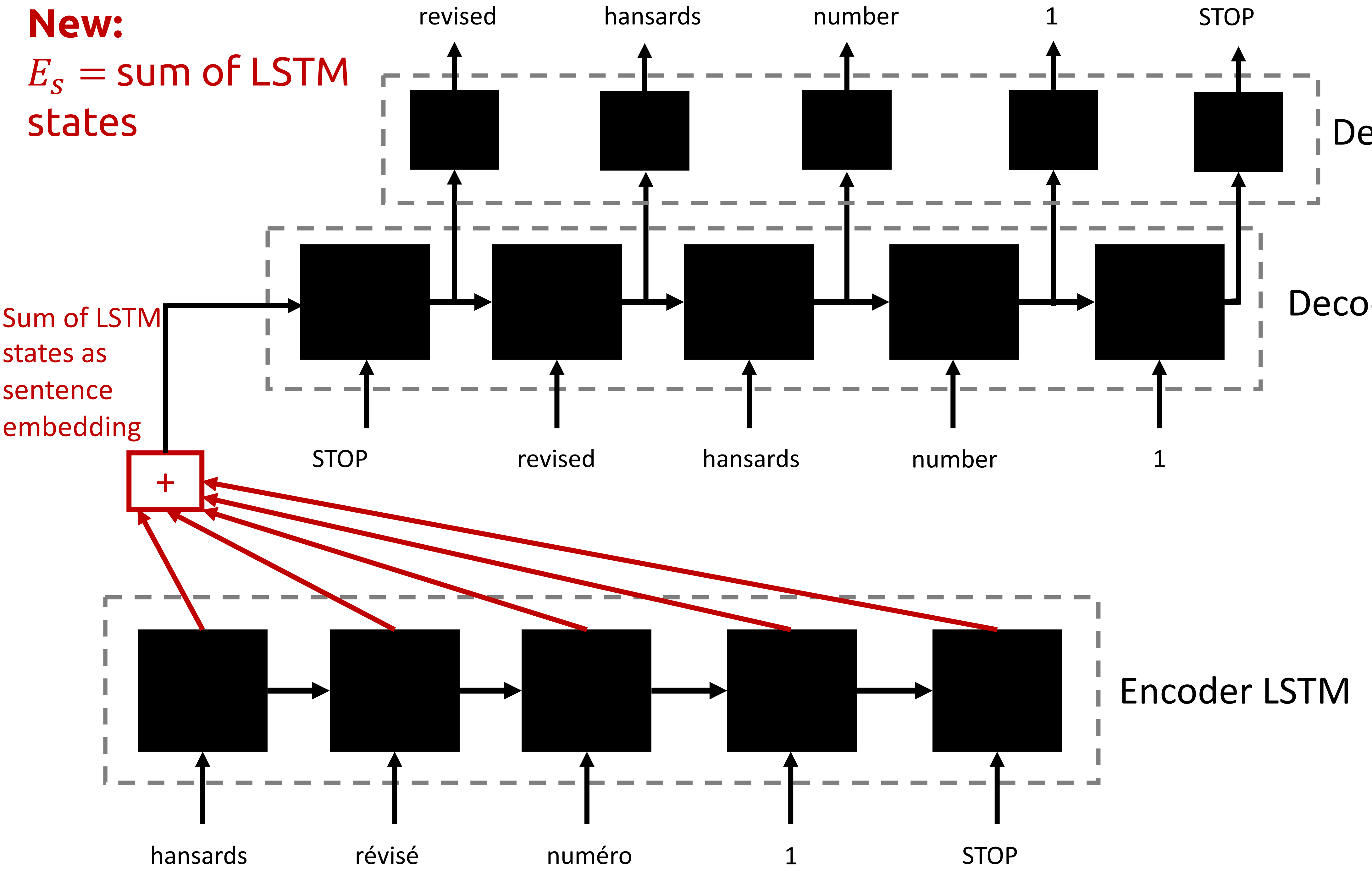
- No one correct answer on how to produce the sentence embedding
- One improvement: instead of taking the final state as the sentence embedding, sum the LSTM states
- Advantage: Less bias towards later words

Old:

E_s = final LSTM state



New:
E_s = sum of LSTM states

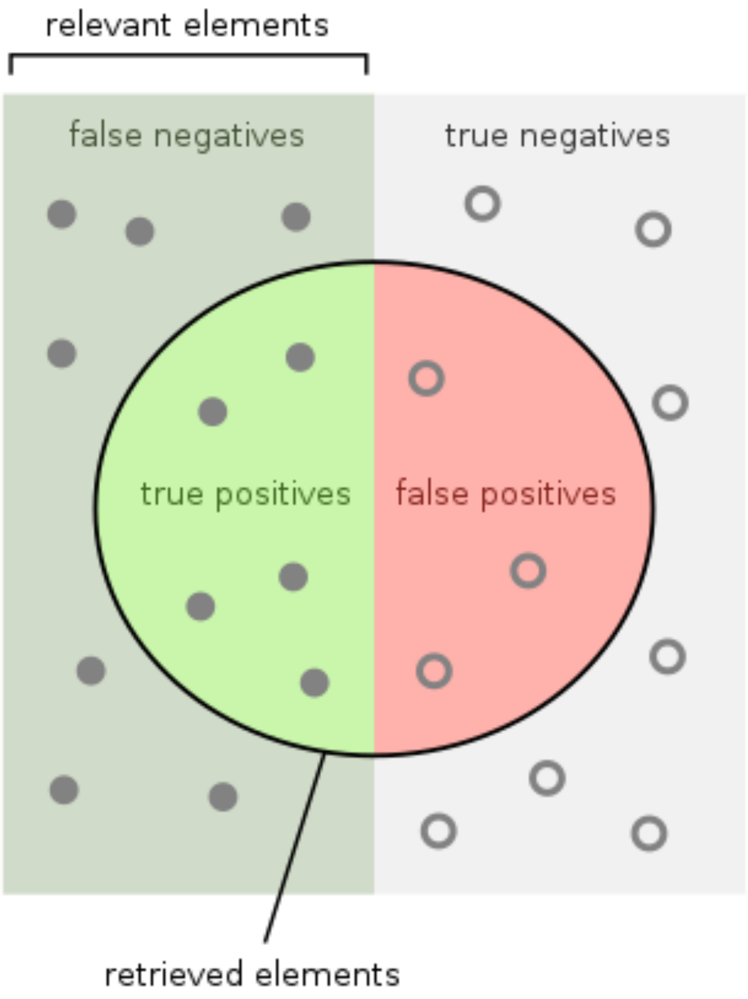


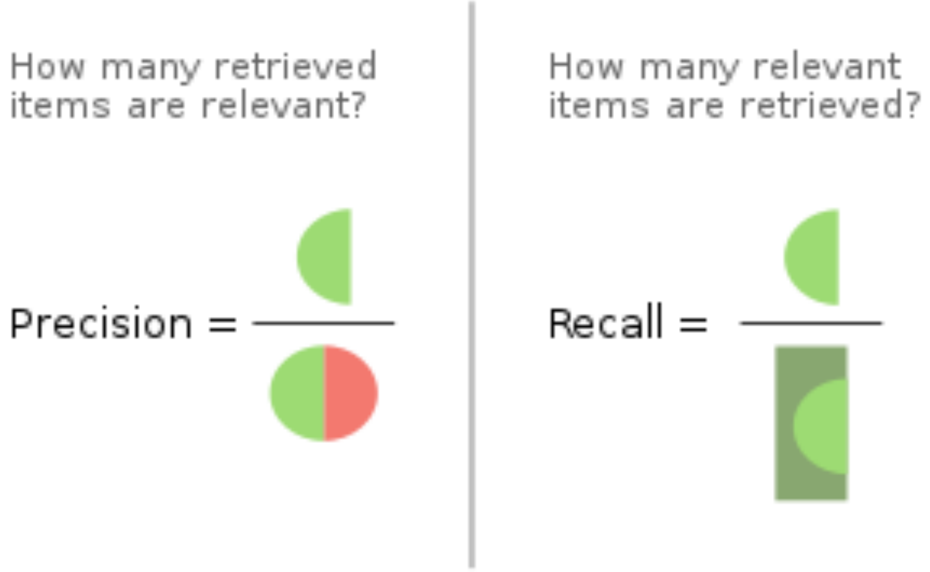
Any questions?

Evaluating MT models

i.e. How do we know if a translation is good?

Precision and Recall





https://en.wikipedia.org/wiki/Precision_and_recall

BLEU

- **B**i-**L**ingual **E**valuation **U**nderstudy

- Based on ***precision***:
  fraction of words generated that are in a given ground-truth ("correct" translated sentence)
  - Or, more commonly, that are in one of several given correct translations

- Instead of naïve precision (per word), use n-grams of each sentence
  - For example, in "Sam saw the black cat", check for "Sam saw the", "saw the black", etc. instead of "Sam", "saw", etc.

ROUGE

- **R**ecall-**O**riented **U**nderstudy for **G**isting **E**valuation
- Based on ***recall***:
  fraction of words in the correct translated sentence that are generated
  - Or, more commonly, that are in one of several given correct translations
- Like BLEU, also looks for n-grams instead of individual words

Calculate BLEU and ROGUE scores (naively!)

Generated: “BLEU prefers shorter sentences”

Ground-Truth: “BLEU prefers shorter sentences more than ROUGE”

Generated: “BLEU prefers shorter sentences more than ROUGE”

Ground-Truth: “BLEU prefers shorter sentences”

Do we prefer BLEU or ROUGE?

Generated: “BLEU prefers shorter sentences”

Ground-Truth: “BLEU prefers shorter sentences more than ROUGE”

BLEU score:

ROUGE score:

Do we prefer BLEU or ROUGE?

Generated: “BLEU prefers shorter sentences more than ROUGE”

Ground-Truth: “BLEU prefers shorter sentences”

BLEU score:

ROUGE score:

Both are biased

- BLEU favors shorter sentences
- ROUGE favors longer sentences

What should we do?

Both are biased

- BLEU favors shorter sentences
- ROUGE favors longer sentences

- So, let's use a metric that combines both BLEU and ROUGE
  - i.e. a single metric that tries to assess ***both*** precision and recall (a common thing to do in information retrieval)

How?

F_1 score

- BLEU favors shorter sentences
- ROUGE favors longer sentences
- F_1 score is the ***harmonic mean*** of BLEU and ROUGE
- $0 \leqq F_1 \leqq 1$
- **The higher the F_1 score, the better the translation**

$$F_1 = \frac{2}{\frac{1}{BLEU} + \frac{1}{ROUGE}} = \frac{2(BLEU \cdot ROUGE)}{BLEU + ROUGE}$$

Why combine using the harmonic mean?

- More appropriate than arithmetic mean for ***rate*** quantities
  - Precision and recall are both rates (i.e. percentage of matching words)

- More info on why: On Average, You’re Using the Wrong Average

- Added benefit: punishes extreme values --- a BLEU score of 0 and a ROUGE score of 1 would result in an F_1 score of 0, not 0.5
  - Note that it’s not actually possible for one sentence to have both a BLEU of 0 and a ROUGE of 1, but you get the idea...

Problems with F_1

- Does the "correct" translation even exist?
  - Sam saw a cat which was black
  - Sam saw a black thing which was a cat
  - A black cat was seen by Sam
  - Sam saw a black cat
- All above sentences are valid – but some are more or less "natural"
- F_1 cannot know this
  - And it may give high scores to unnatural translations if they have high word overlap with known good translations!

Problems with F_1 Morphologically rich languages

- Here are two translations of “Her village is large” into Shipibo, which is spoken in Peru:

  Jawen jemara ani iki

  Jawen jemaronki ani iki

- Sentence 1: The speaker is claiming the village is large because they have seen it with their own eyes
- Sentence 2: The speaker is claiming the village is large because they were told so by someone else

Problems with F_1
No understanding of meaning

Target: "F1 score is a flawed metric for evaluating machine translation systems"

Generated 1: "F1 score is an imperfect metric for evaluating machine translation systems"

F_1 score: 0.599

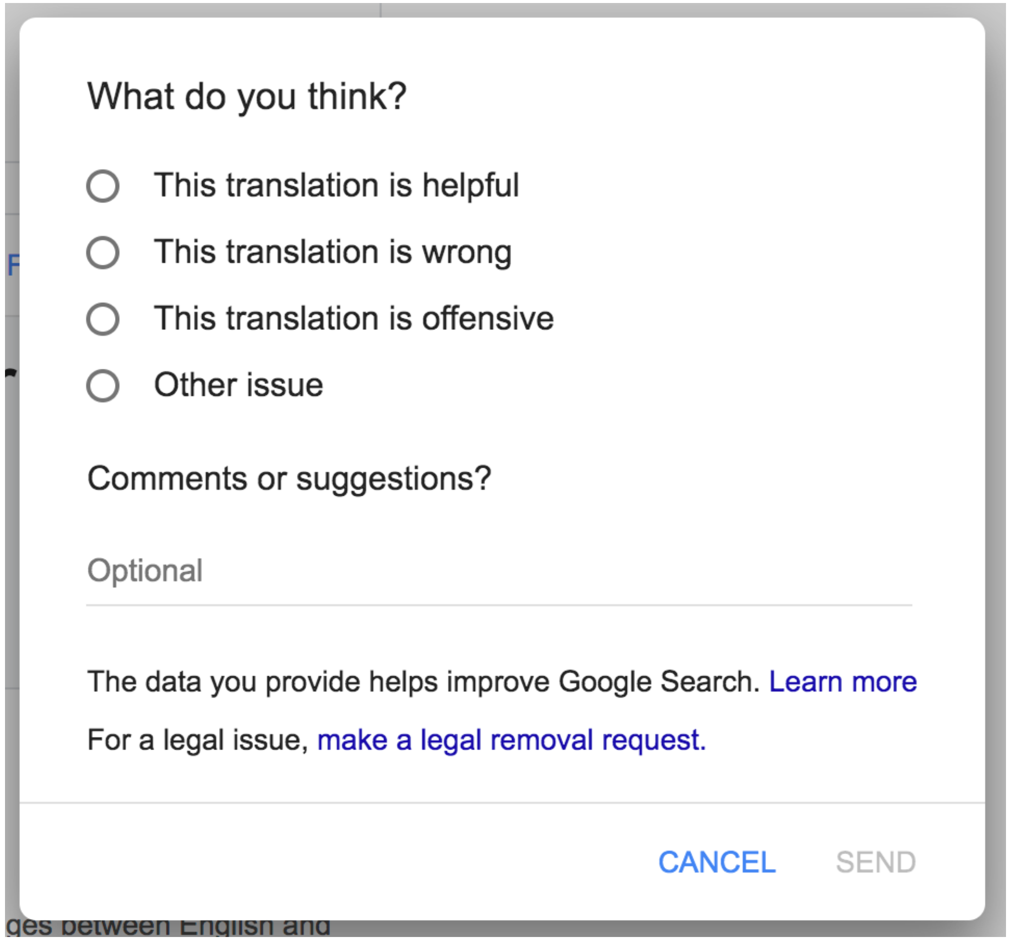
Generated 2: "F1 score is a ***great*** metric for evaluating machine translation systems"

F_1 score: 0.710

Is this accurate?

Human evaluation

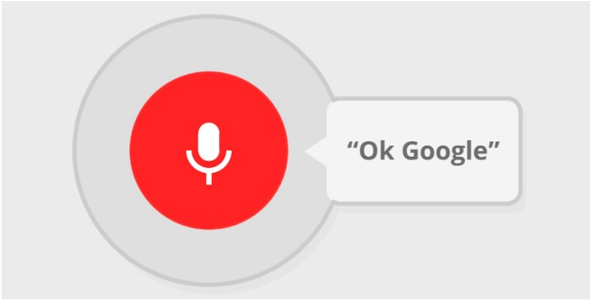
- The alternative is to have humans evaluate each translation
- However, this is very time consuming
- Google Translate attempts this with its “Translate Community” – volunteers who rate translations and suggest improvements

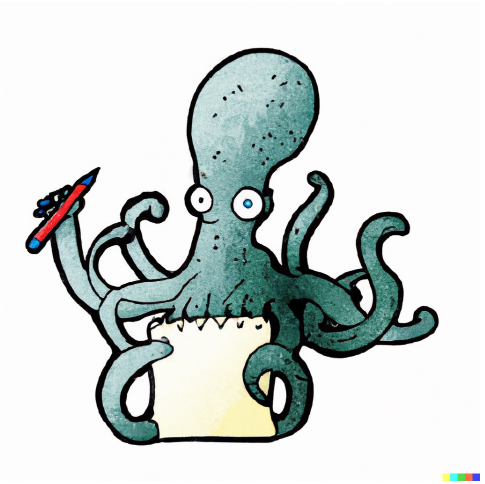


Recap

Machine translation

- Can train DL models using Parallel Corpora
- Seq-2-seq prediction (encoder-decoder)
- Various applications





Evaluation

- BLEU
- ROGUE
- F1 SCORE

$$F_1 = \frac{2}{\frac{1}{BLEU} + \frac{1}{ROUGE}} = \frac{2(BLEU \cdot ROUGE)}{BLEU + ROUGE}$$

Extra: Other Applications of seq2seq

- **Text summarization**

- Source: Long text pessage
- Target: Shortened version of input text passage

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.

https://blog.floydhub.com/gentle-introduction-to-text-summarization-in-machine-learning/

Extra: Other Applications of seq2seq

- Text summarization
- **Chatbots**

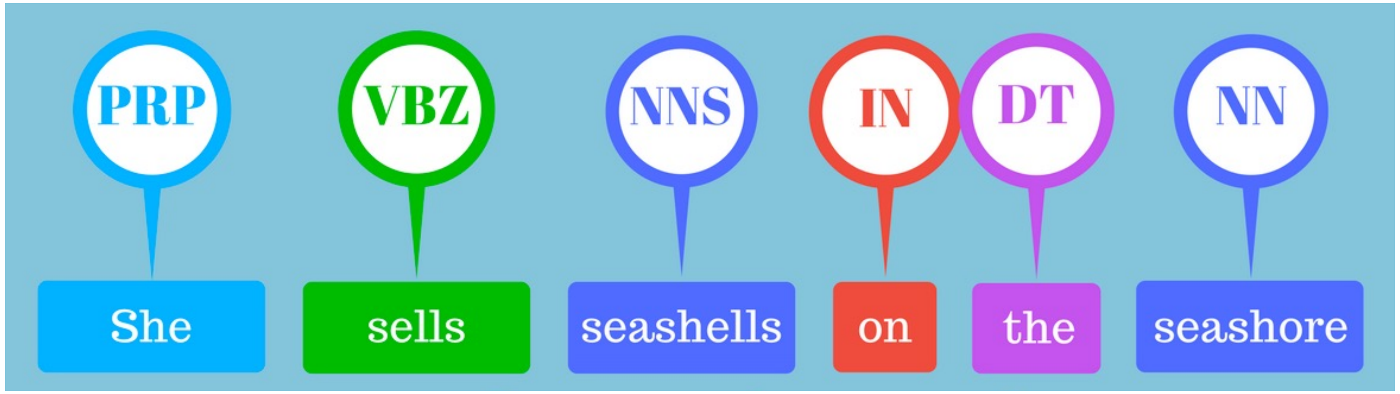
- Source: user question
- Target: chatbot response

https://www.yodlee.com/blog/chatbots-in-banking/

Extra: Other Applications of seq2seq

- Text summarization
- Chatbots
- **Part of speech tagging**

- Source: natural language sentence
- Target: part-of-speech labels for each word in the input sentence

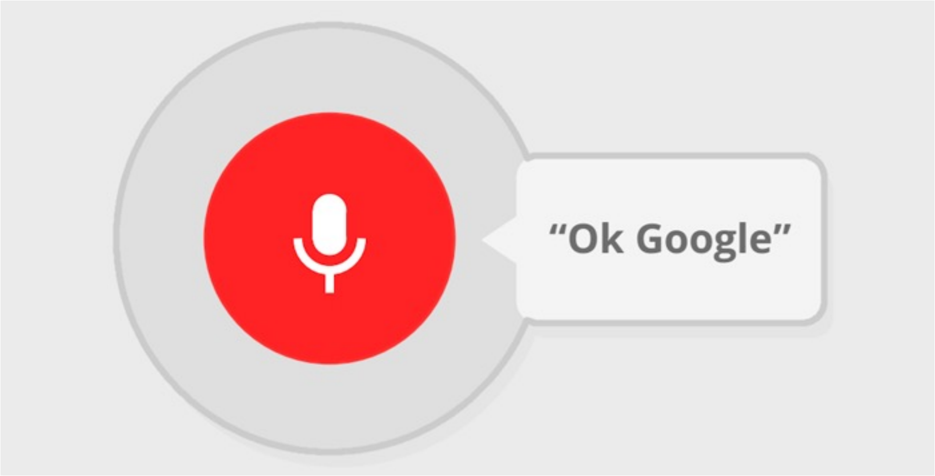


https://medium.com/analytics-vidhya/pos-tagging-using-conditional-random-fields-92077e5eaa31

Extra: Other Applications of seq2seq

- Text summarization
- Chatbots
- Part of speech tagging
- **Speech recognition**

- Input: sequence of audio samples
- Output: sequence of text words



Extra: Other Applications of seq2seq

- Text summarization
- Chatbots
- Part of speech tagging
- Speech recognition
- **Speech generation**

- Input: sequence of text words
- Output: sequence of audio samples

- Google Cloud Text to Speech