Machine Translation

Many slides from Michael Collins, Chris Calison-Burch, Alan Ritter, Thang Luong

Overview

- Challenges in machine translation
- Classical machine translation
- A brief introduction to statistical MT

Challenges: Lexical Ambiguity

(Example from Dorr et. al, 1999)

book the flight ⇒ reservar
read the book ⇒ libro

Spanish

kill a man ⇒ matar
kill a process ⇒ acabar

Portuguese

Challenges: Differing Word Orders

- English word order is *subject – verb – object*
- Japanese word order is *subject – object – verb*

English: IBM bought Lotus
Japanese: *IBM Lotus bought*

English: Sources said that IBM bought Lotus yesterday
Japanese: *Sources yesterday IBM Lotus bought that said*

Syntactic Structure is not Preserved Across Translations (Example from Dorr et. al, 1999)

The bottle floated into the cave

La botella entro a la cuerva flotando
(the bottle entered the cave floating)

Syntactic Ambiguity Causes Problems

(Example from Dorr et. al, 1999)

John hit the dog with the stick

$\Downarrow$

John golpeo el perro con el palo/que tenia el palo

Pronoun Resolution (Example from Dorr et. al, 1999)

The computer outputs the data; it is fast.

$\Downarrow$

La computadora imprime los datos; es rapida

The computer outputs the data; it is stored in ascii.

$\Downarrow$

La computadora imprime los datos; estan almacendos en ascii

Overview

- Challenges in machine translation
- Classical machine translation
- A brief introduction to statistical MT

Direct Machine Translation

- Translation is word-by-word
- Very little analysis of the source text (e.g., no syntactic or semantic analysis)
- Relies on a large bilingual directionary. For each word in the source language, the dictionary specifies a set of rules for translating that word
- After the words are translated, simple reordering rules are applied (e.g., move adjectives after nouns when translating from English to French)

An Example of a set of Direct Translation Rules

(From Jurafsky and Martin, edition 2, chapter 25. Originally from a system from Panov 1960)

Rules for translating *much* or *many* into Russian:

if preceding word is *how* **return** *skol'ko*
else if preceding word is *as* **return** *stol'ko zhe*
else if word is *much*
    if preceding word is *very* **return** nil
    else if following word is a noun **return** *mnogo*
else (word is many)
    if preceding word is a preposition and following word is noun **return** *mnogii*
    else return *mnogo*

Some Problems with Direct Machine Translation

- Lack of any analysis of the source language causes several problems, for example:
  - Difficult or impossible to capture long-range reorderings

    English: Sources said that IBM bought Lotus yesterday
    Japanese: *Sources yesterday IBM Lotus bought that said*

  - Words are translated without disambiguation of their syntactic role
    e.g., *that* can be a complementizer or determiner, and will often be translated differently for these two cases

    They said *that* ...

    They like *that* ice-cream

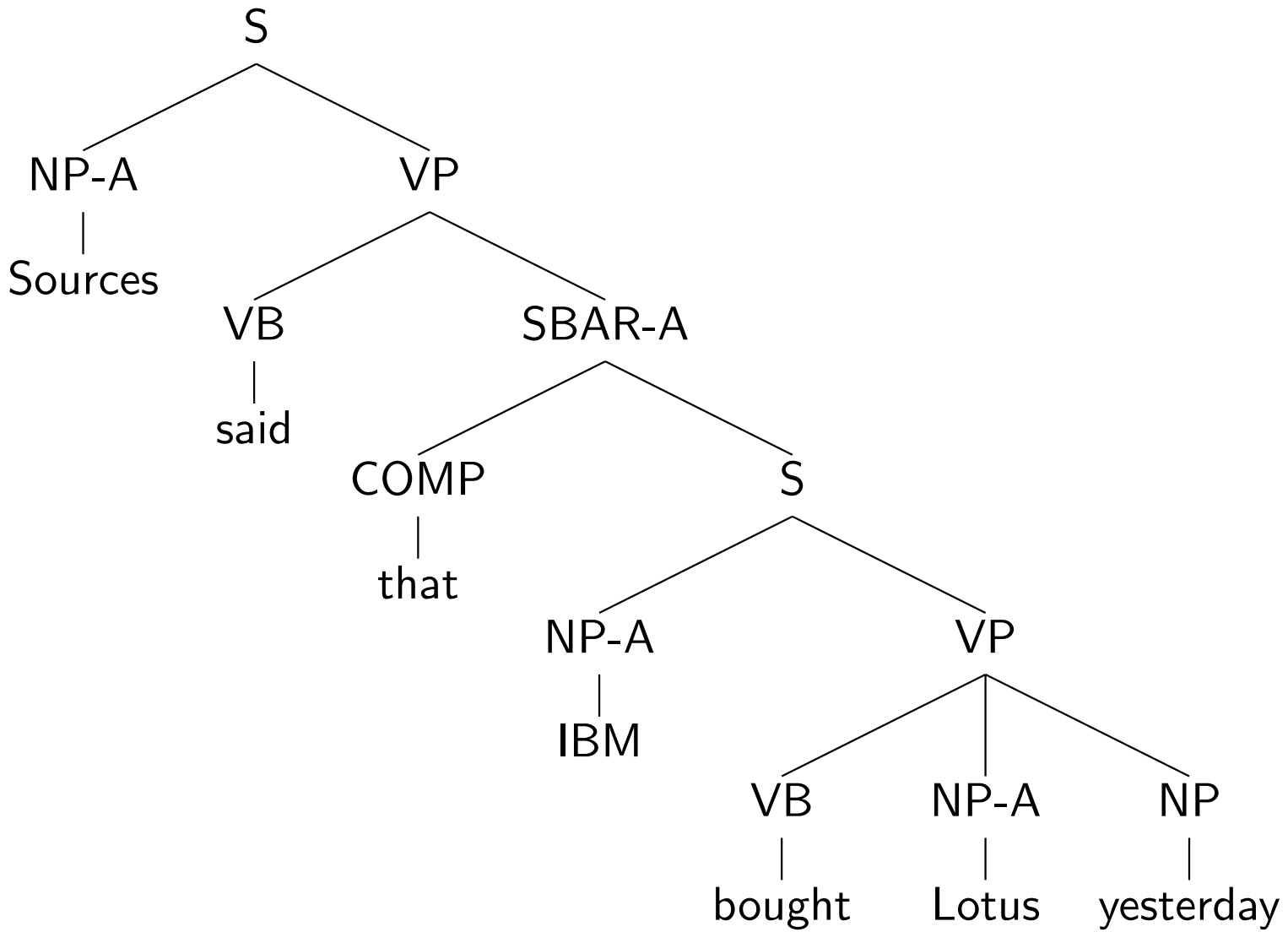
Transfer-Based Approaches

Three phases in translation:

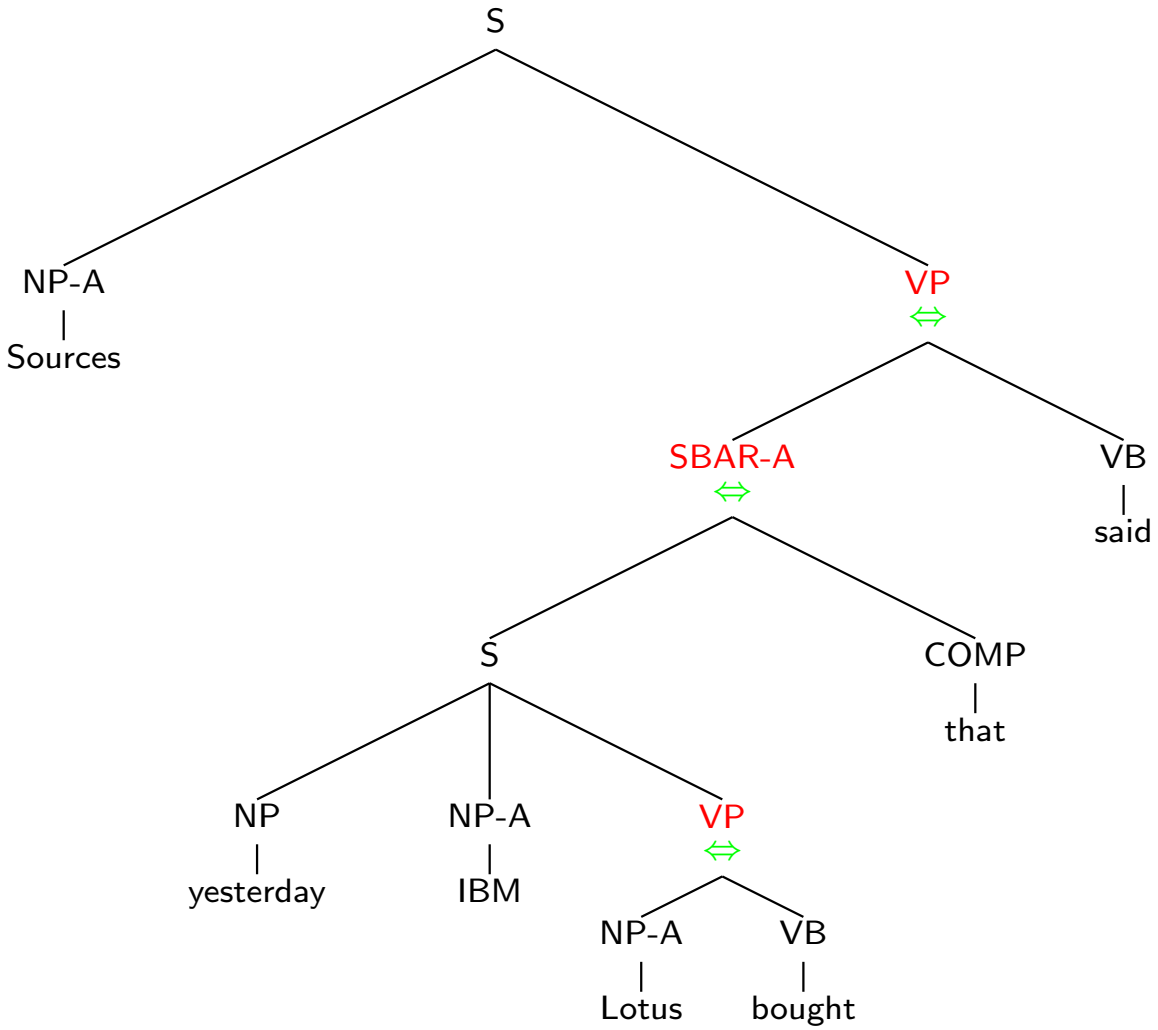
- Analysis: Analyze the source language sentence; for example, build a syntactic analysis of the source language sentence.
- Transfer: Convert the source-language parse tree to a target-language parse tree.
- Generation: Convert the target-language parse tree to an output sentence.

Transfer-Based Approaches

- The "parse trees" involved can vary from shallow analyses to much deeper analyses (even semantic representations).
- The transfer rules might look quite similar to the rules for direct translation systems. But they can now operate on syntactic structures.
- It's easier with these approaches to handle long-distance reorderings
- The *Systran* systems are a classic example of this approach



$\Rightarrow$ Japanese: *Sources yesterday IBM Lotus bought that said*


S
NP-A
Sources
VP
⇔
SBAR-A
⇔
VB
said
S
COMP
that
NP
yesterday
NP-A
IBM
VP
⇔
NP-A
Lotus
VB
bought

Interlingua-Based Translation

Two phases in translation:

- Analysis: Analyze the source language sentence into a (language-independent) representation of its meaning.
- Generation: Convert the meaning representation into an output sentence.

Interlingua-Based Translation

One Advantage: If we want to build a translation system that translates between n languages, we need to develop n analysis and generation systems. With a transfer based system, we'd need to develop $O(n^2)$ sets of translation rules.

Disadvantage: What would a language-independent representation look like?

Interlingua-Based Translation

- How to represent different concepts in an interlingua?
- Different languages break down concepts in quite different ways:

  German has two words for *wall*: one for an internal wall, one for a wall that is outside

  Japanese has two words for *brother*: one for an elder brother, one for a younger brother

  Spanish has two words for *leg*: *pierna* for a human's leg, *pata* for an animal's leg, or the leg of a table

- An interlingua might end up simple being an intersection of these different ways of breaking down concepts, but that doesn't seem very satisfactory...

Overview

- Challenges in machine translation
- Classical machine translation
- A brief introduction to statistical MT

A Brief Introduction to Statistical MT

- Parallel corpora are available in several language pairs
- Basic idea: use a parallel corpus as a training set of translation examples
- Classic example: IBM work on French-English translation, using the Canadian Hansards. (1.7 million sentences of 30 words or less in length).
- Idea goes back to Warren Weaver (1949): suggested applying statistical and cryptanalytic techniques to translation.

When I look at an article in Russian, I say: "This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode."

Warren Weaver (1949)

The Noisy Channel Model

- Goal: translation system from French to English
- Have a model $p(e \mid f)$ which estimates conditional probability of any English sentence e given the French sentence f. Use the training corpus to set the parameters.
- A Noisy Channel Model has two components:

$$p(e) \quad \textbf{the language model}$$

$$p(f \mid e) \quad \textbf{the translation model}$$

- Giving:

$$p(e \mid f) = \frac{p(e, f)}{p(f)} = \frac{p(e)p(f \mid e)}{\sum_e p(e)p(f \mid e)}$$

and

$$\text{argmax}_e p(e \mid f) = \text{argmax}_e p(e)p(f \mid e)$$

More About the Noisy Channel Model

- The **language model** $p(e)$ could be a trigram model, estimated from any data (parallel corpus not needed to estimate the parameters)

- The **translation model** $p(f \mid e)$ is trained from a parallel corpus of French/English pairs.

- Note:
  - The translation model is backwards!
  - The language model can make up for deficiencies of the translation model.
  - Later we'll talk about how to build $p(f \mid e)$
  - Decoding, i.e., finding

$$\text{argmax}_e p(e)p(f \mid e)$$

    is also a challenging problem.

Example from Koehn and Knight tutorial

Translation from Spanish to English, candidate translations based on $p(Spanish \mid English)$ alone:

Que hambre tengo yo

$\rightarrow$

What hunger have	$p(s\|e) = 0.000014$
Hungry I am so	$p(s\|e) = 0.000001$
I am so hungry	$p(s\|e) = 0.0000015$
Have i that hunger	$p(s\|e) = 0.000020$

...

Example from Koehn and Knight tutorial (continued)

With $p(Spanish \mid English) \times p(English)$:

Que hambre tengo yo

$\rightarrow$

What hunger have	$p(s\|e)p(e) = 0.000014 \times 0.000001$
Hungry I am so	$p(s\|e)p(e) = 0.000001 \times 0.0000014$
I am so hungry	$p(s\|e)p(e) = 0.0000015 \times 0.0001$
Have i that hunger	$p(s\|e)p(e) = 0.000020 \times 0.00000098$

...

Recap: The Noisy Channel Model

- Goal: translation system from French to English
- Have a model $p(e \mid f)$ which estimates conditional probability of any English sentence e given the French sentence f. Use the training corpus to set the parameters.
- A Noisy Channel Model has two components:

$$p(e) \quad \textbf{the language model}$$

$$p(f \mid e) \quad \textbf{the translation model}$$

- Giving:

$$p(e \mid f) = \frac{p(e, f)}{p(f)} = \frac{p(e)p(f \mid e)}{\sum_e p(e)p(f \mid e)}$$

and

$$\text{argmax}_e p(e \mid f) = \text{argmax}_e p(e)p(f \mid e)$$

IBM Model 2: The Generative Process

To generate a French string f from an English string e:

- **Step 1:** Pick an alignment $a = \{a_1, a_2 \ldots a_m\}$ with probability

$$\prod_{j=1}^{m} \mathbf{q}(a_j \mid j, l, m)$$

- **Step 3:** Pick the French words with probability

$$p(f \mid a, e, m) = \prod_{j=1}^{m} \mathbf{t}(f_j \mid e_{a_j})$$

The final result:

$$p(f, a \mid e, m) = p(a \mid e, m)p(f \mid a, e, m) = \prod_{j=1}^{m} \mathbf{q}(a_j \mid j, l, m)\mathbf{t}(f_j \mid e_{a_j})$$

Recovering Alignments

- If we have parameters q and t, we can easily recover the most likely alignment for any sentence pair

- Given a sentence pair $e_1, e_2, \ldots, e_l$, $f_1, f_2, \ldots, f_m$, define

$$a_j = \arg \max_{a \in \{0 \ldots l\}} q(a|j, l, m) \times t(f_j|e_a)$$

for $j = 1 \ldots m$

e = And the program has been implemented

f = Le programme a ete mis en application

The Parameter Estimation Problem

- Input to the parameter estimation algorithm: $(e^{(k)}, f^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence

- Output: parameters $t(f|e)$ and $q(i|j, l, m)$

- A key challenge: **we do not have alignments on our training examples**, e.g.,

$$e^{(100)} = \text{And the program has been implemented}$$
$$f^{(100)} = \text{Le programme a ete mis en application}$$

Parameter Estimation if the Alignments are Observed

- First: case where alignments are observed in training data.
  E.g.,

$$e^{(100)} = \text{And the program has been implemented}$$

$$f^{(100)} = \text{Le programme a ete mis en application}$$
$$a^{(100)} = \langle 2, 3, 4, 5, 6, 6, 6 \rangle$$

- Training data is $(e^{(k)}, f^{(k)}, a^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence, each $a^{(k)}$ is an alignment
- Maximum-likelihood parameter estimates in this case are trivial:

$$t_{ML}(f|e) = \frac{\text{Count}(e,\ f)}{\text{Count}(e)} \qquad q_{ML}(j|i,l,m) = \frac{\text{Count}(j|i,l,m)}{\text{Count}(i,l,m)}$$

Input: A training corpus $(f^{(k)}, e^{(k)})$ for $k = 1 \ldots n$, where $f^{(k)} = f_1^{(k)} \ldots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \ldots e_{l_k}^{(k)}$.

Initialization: Initialize $t(f|e)$ and $q(j|i, l, m)$ parameters (e.g., to random values).

For $s = 1 \ldots S$

- Set all counts $c(\ldots) = 0$
- For $k = 1 \ldots n$
  - For $i = 1 \ldots m_k$, For $j = 0 \ldots l_k$

$$
\begin{aligned}
c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k, i, j) \\
c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k, i, j) \\
c(j|i, l, m) &\leftarrow c(j|i, l, m) + \delta(k, i, j) \\
c(i, l, m) &\leftarrow c(i, l, m) + \delta(k, i, j)
\end{aligned}
$$

where

$$
\delta(k, i, j) = \frac{q(j|i, l_k, m_k) t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i, l_k, m_k) t(f_i^{(k)}|e_j^{(k)})}
$$

- Recalculate the parameters:

$$
t(f|e) = \frac{c(e, f)}{c(e)} \qquad q(j|i, l, m) = \frac{c(j|i, l, m)}{c(i, l, m)}
$$

$$\delta(k,i,j) = \frac{q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}$$

$e^{(100)}$ = And the program has been implemented

$f^{(100)}$ = Le programme a ete mis en application

Justification for the Algorithm

- Training examples are $(e^{(k)}, f^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence

- The log-likelihood function:

$$L(t, q) = \sum_{k=1}^{n} \log p(f^{(k)}|e^{(k)}) = \sum_{k=1}^{n} \log \sum_{a} p(f^{(k)}, a|e^{(k)})$$

- The maximum-likelihood estimates are

$$\arg\max_{t,q} L(t, q)$$

- The EM algorithm will converge to a *local maximum* of the log-likelihood function

Summary

- Key ideas in the IBM translation models:
  - Alignment variables
  - Translation parameters, e.g., $t(\text{chien}|\text{dog})$
  - Distortion parameters, e.g., $q(2|1,6,7)$
- The EM algorithm: an iterative algorithm for training the q and t parameters
- Once the parameters are trained, we can recover the most likely alignments on our training examples

$$e = \text{And the program has been implemented}$$

$$f = \text{Le programme a ete mis en application}$$

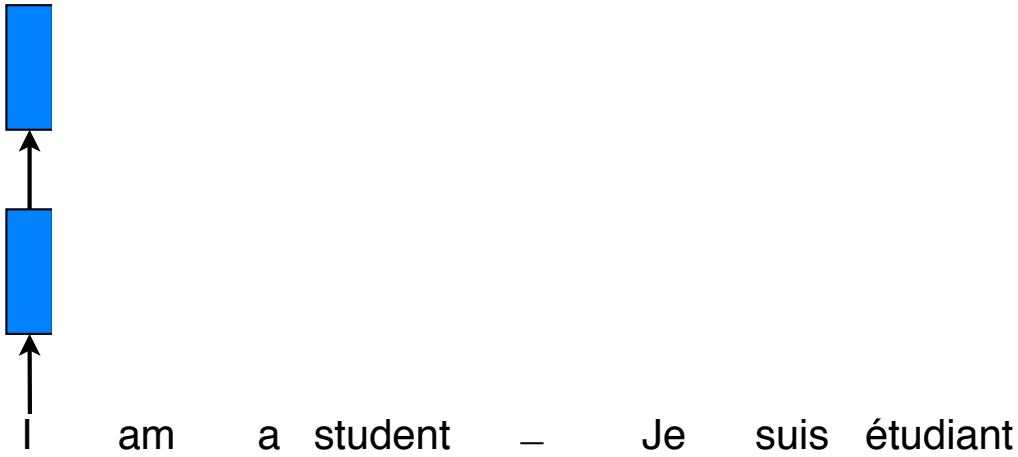
Sixi roasted husband

Meat Muscle Stupid Bean Sprouts

Sixi roasted husband

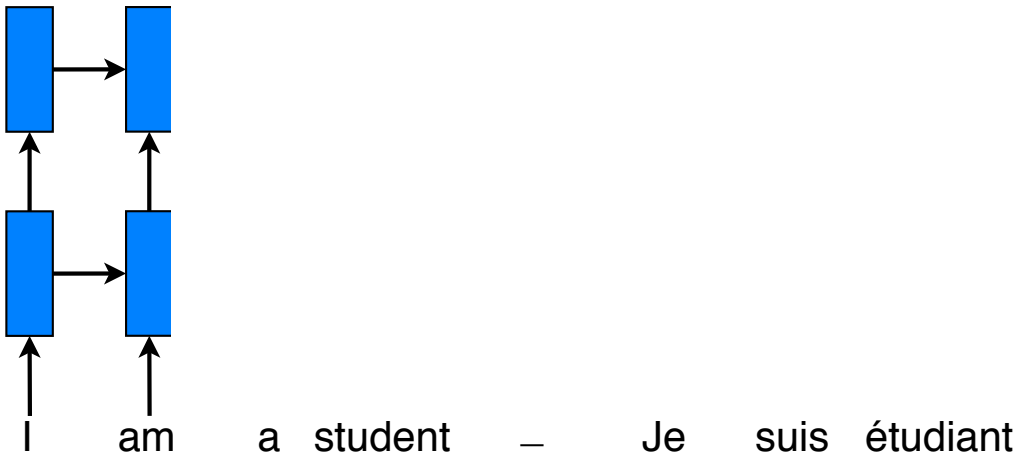
Meat Muscle Stupid Bean Sprouts

Neural Machine Translation (NMT)



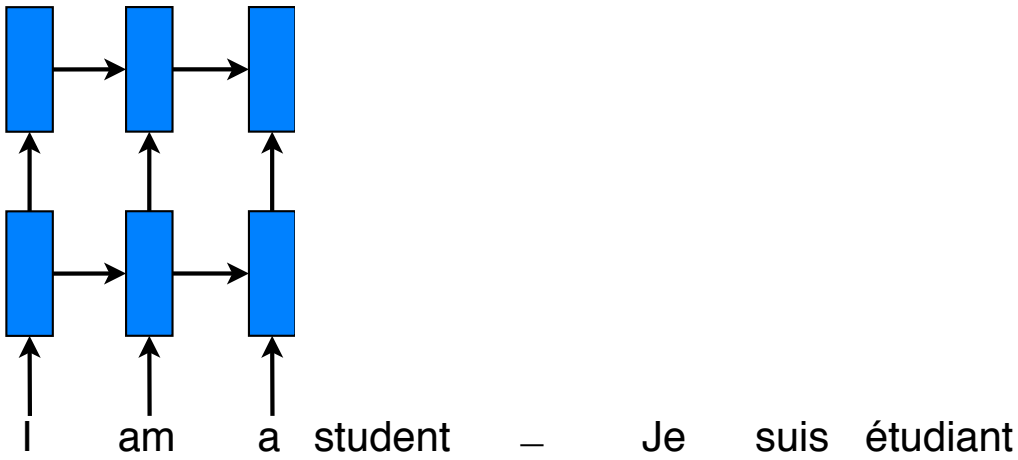
- RNNs trained end-to-end (Sutskever et al., 2014).

Neural Machine Translation (NMT)



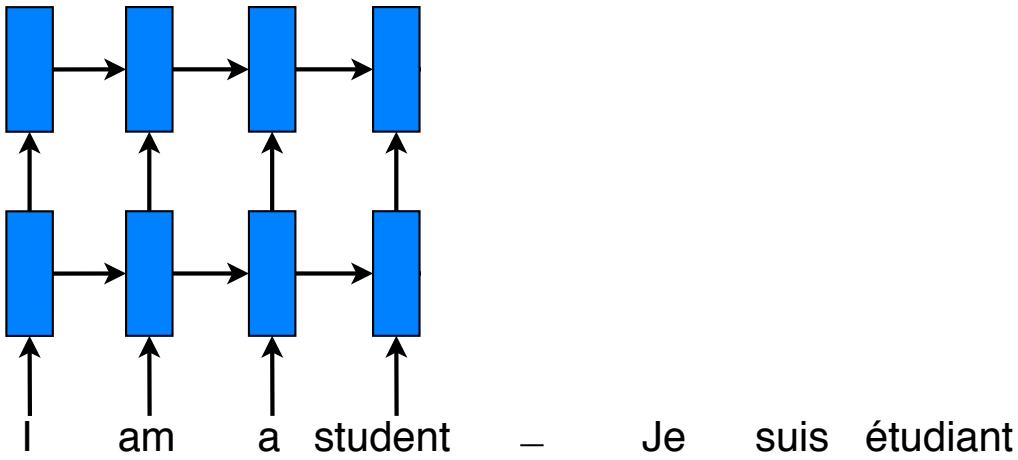
- RNNs trained end-to-end (Sutskever et al., 2014).

Neural Machine Translation (NMT)



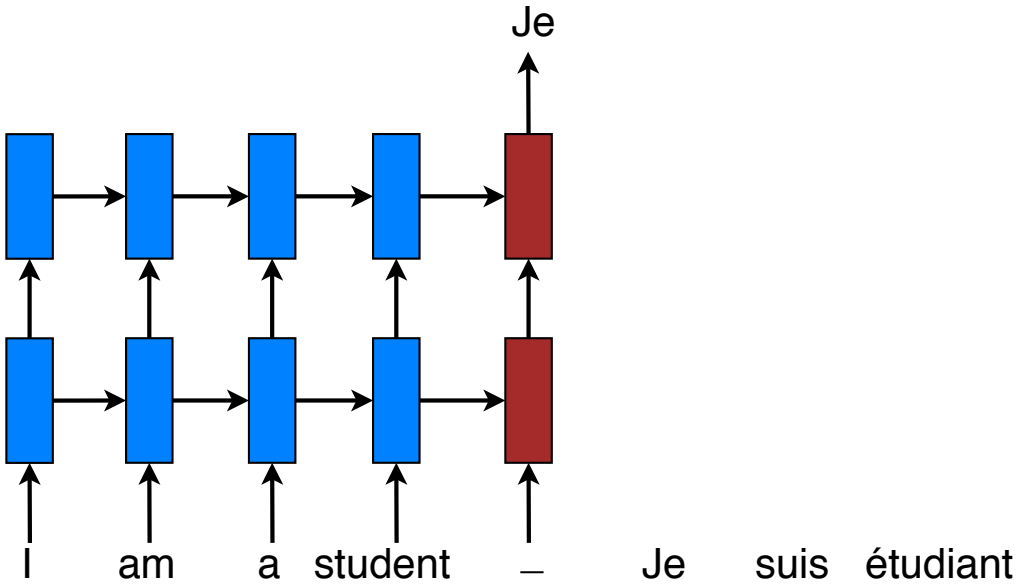
- RNNs trained end-to-end (Sutskever et al., 2014).

Neural Machine Translation (NMT)



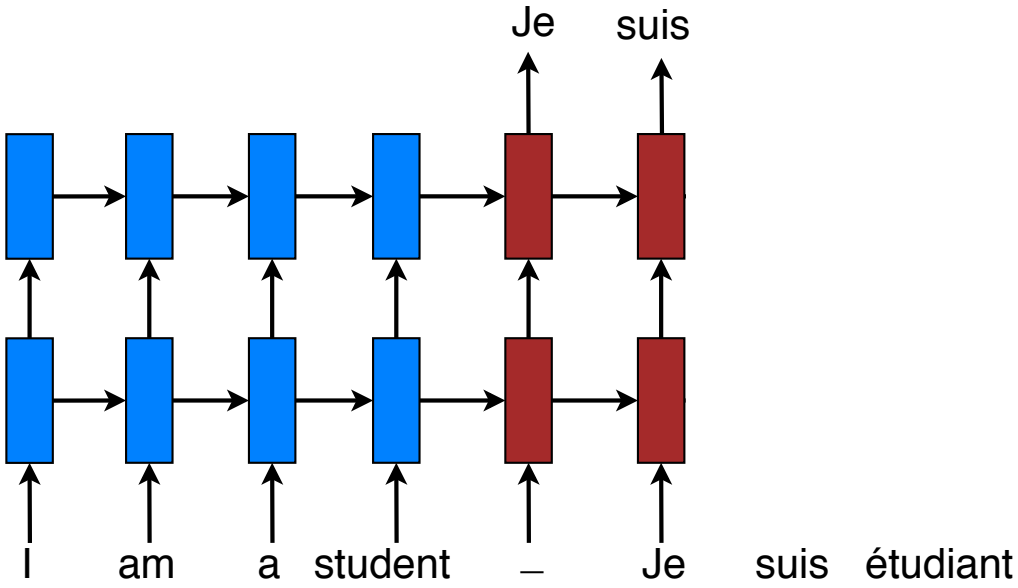
- RNNs trained end-to-end (Sutskever et al., 2014).

Neural Machine Translation (NMT)



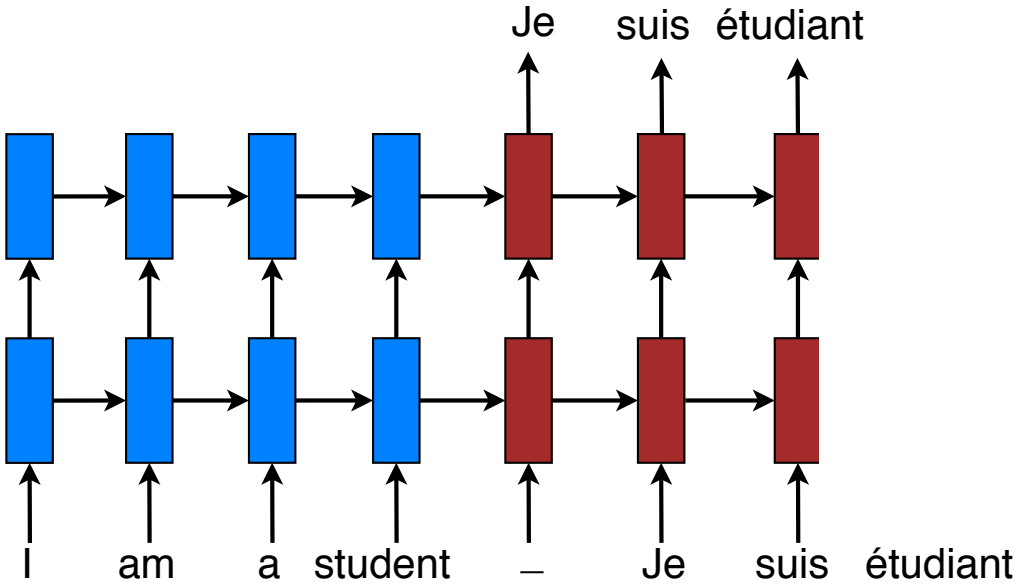
- RNNs trained end-to-end (Sutskever et al., 2014).

Neural Machine Translation (NMT)



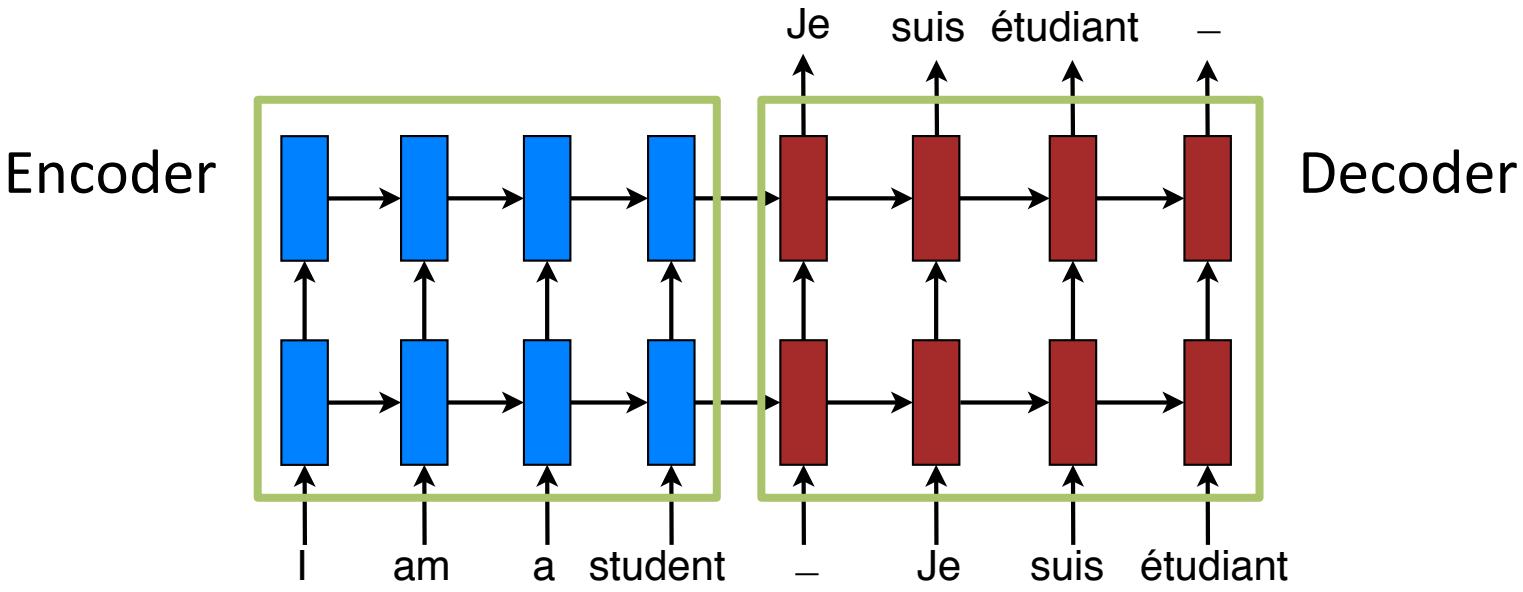
- RNNs trained end-to-end (Sutskever et al., 2014).

Neural Machine Translation (NMT)



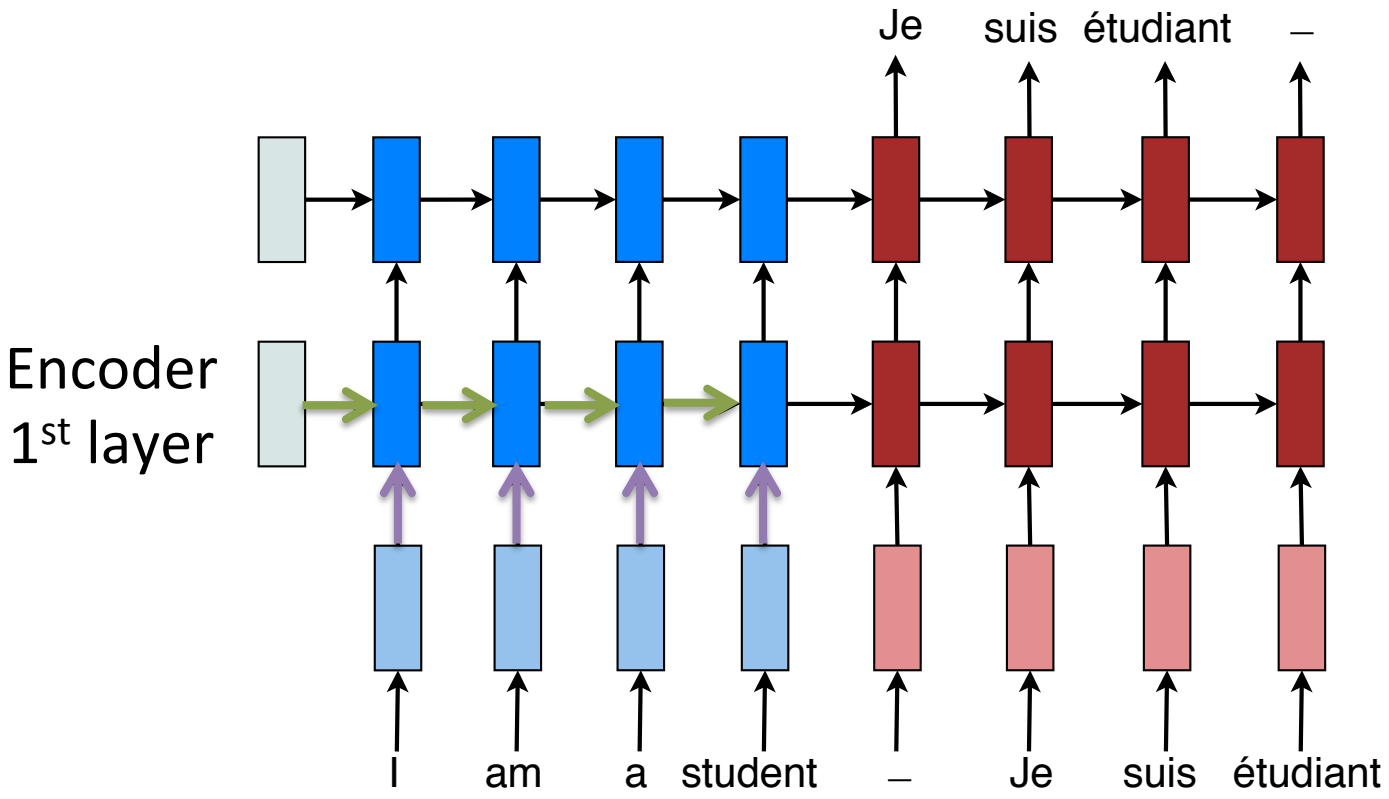
- RNNs trained end-to-end (Sutskever et al., 2014).

Neural Machine Translation (NMT)



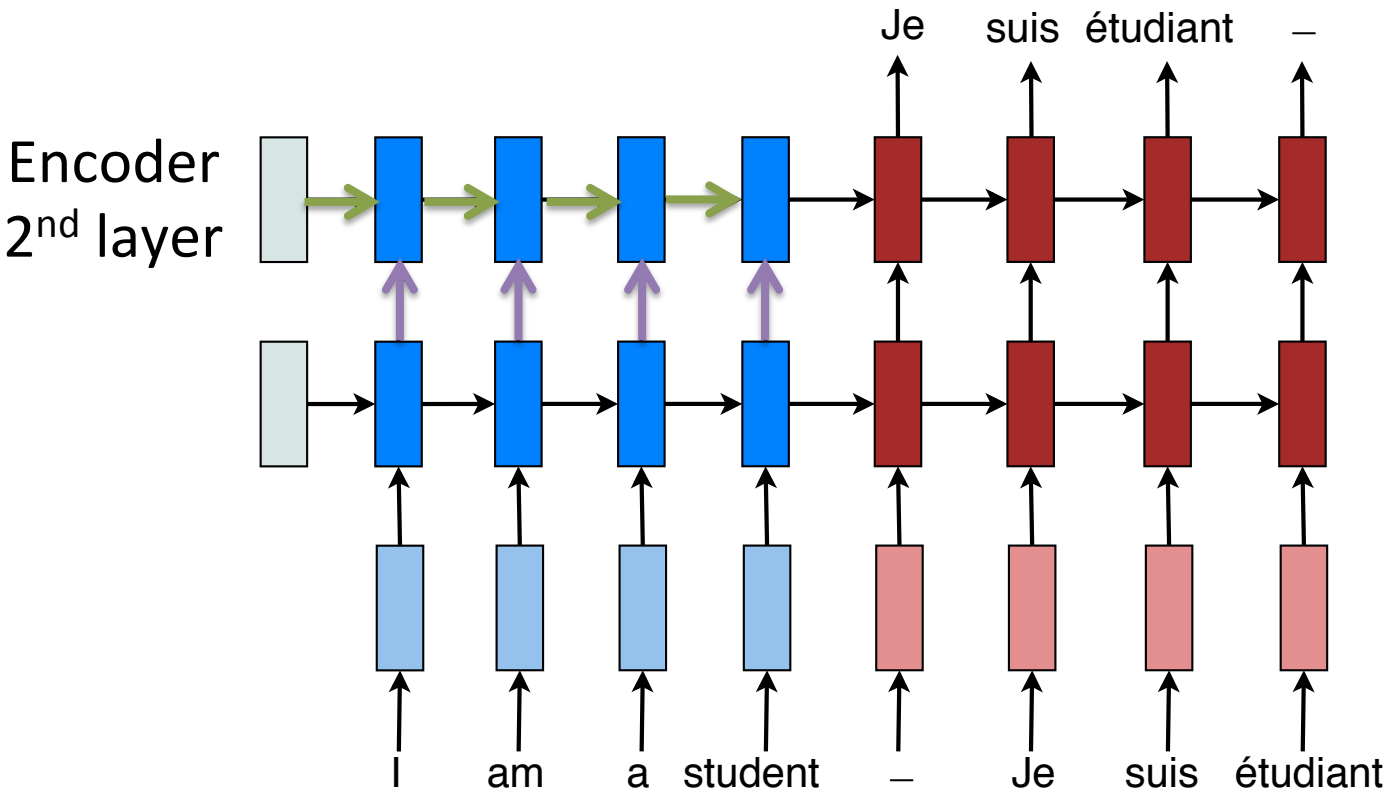
- RNNs trained end-to-end (Sutskever et al., 2014).
- Encoder-decoder approach.

Recurrent Connections



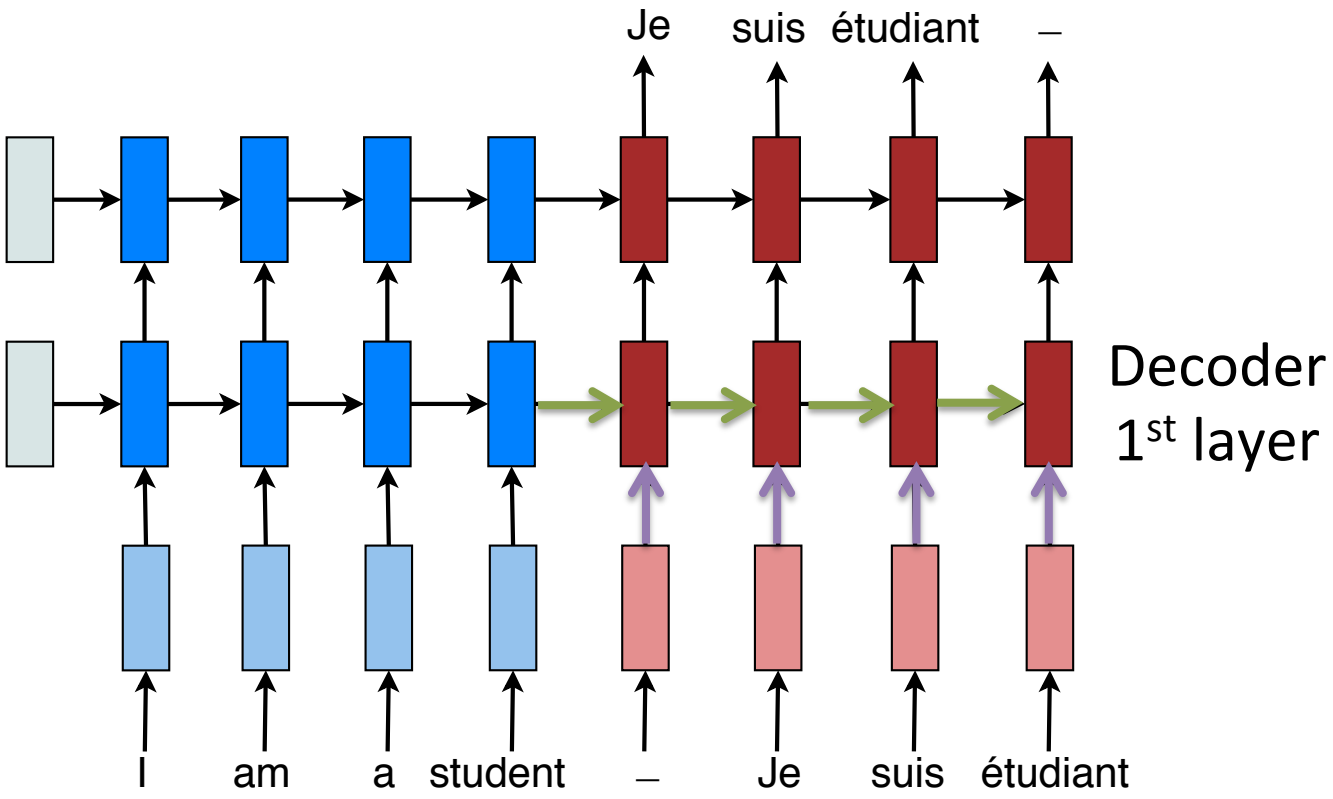
- Different across layers and encoder / decoder.

Recurrent Connections



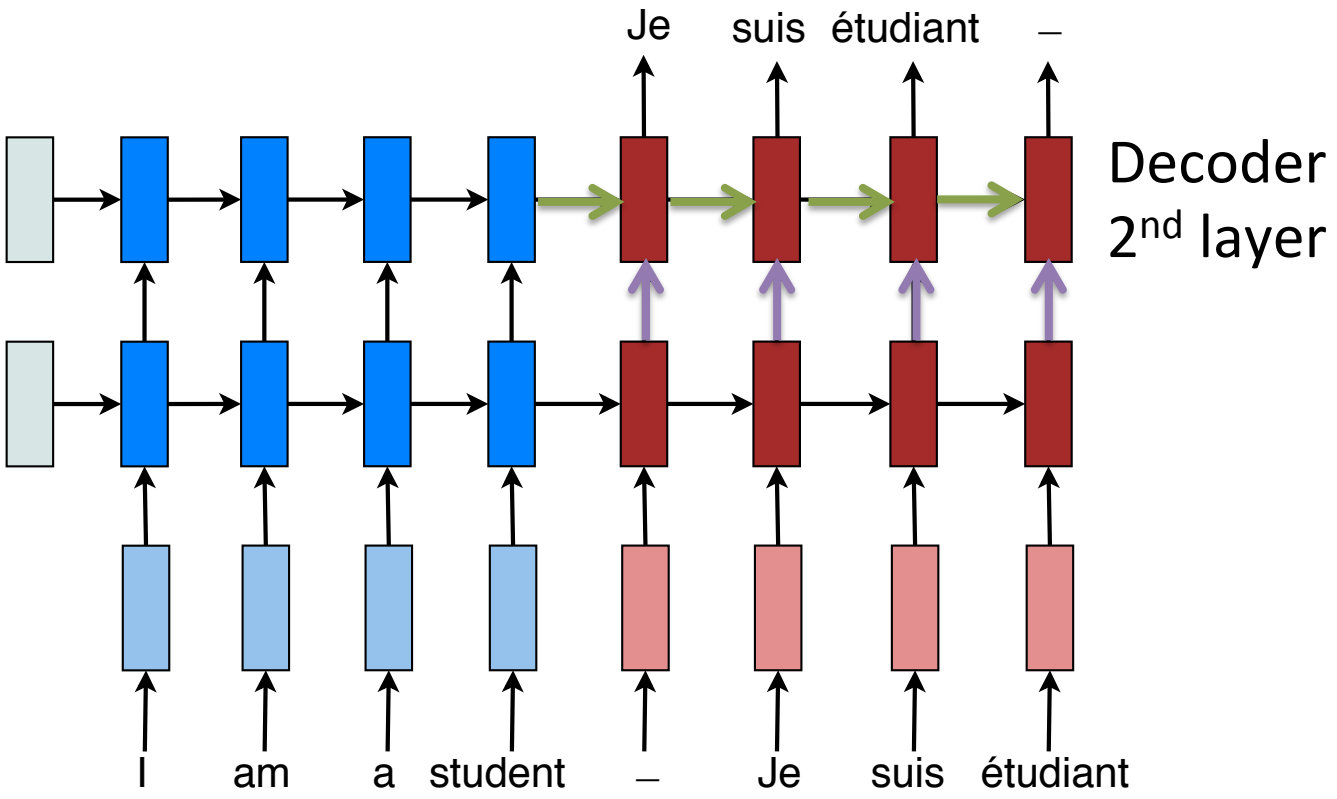
- Different across layers and encoder / decoder.

Recurrent Connections



- Different across layers and encoder / decoder.

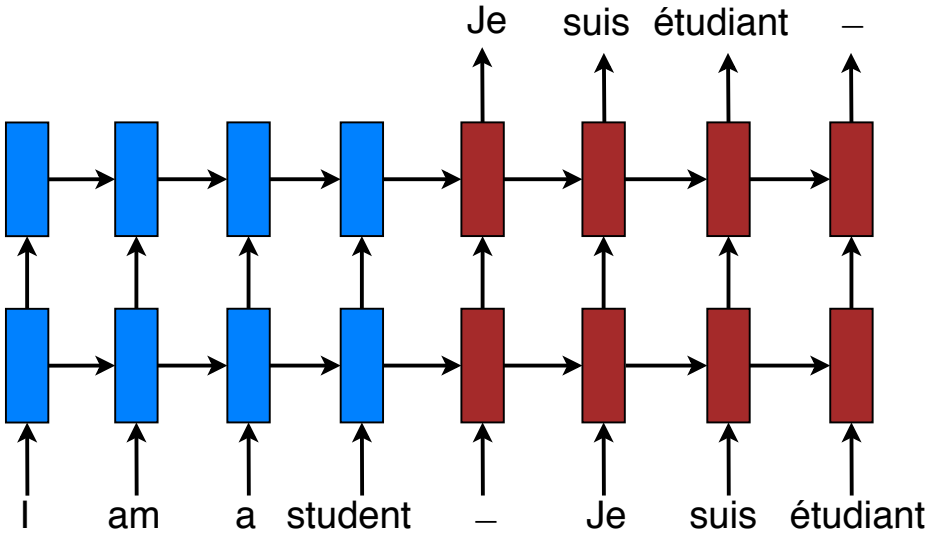
Recurrent Connections



- Different across layers and encoder / decoder.

Training vs. Testing

- *Training*
  - Correct translations are available.



- *Testing*
  - Only source sentences are given.