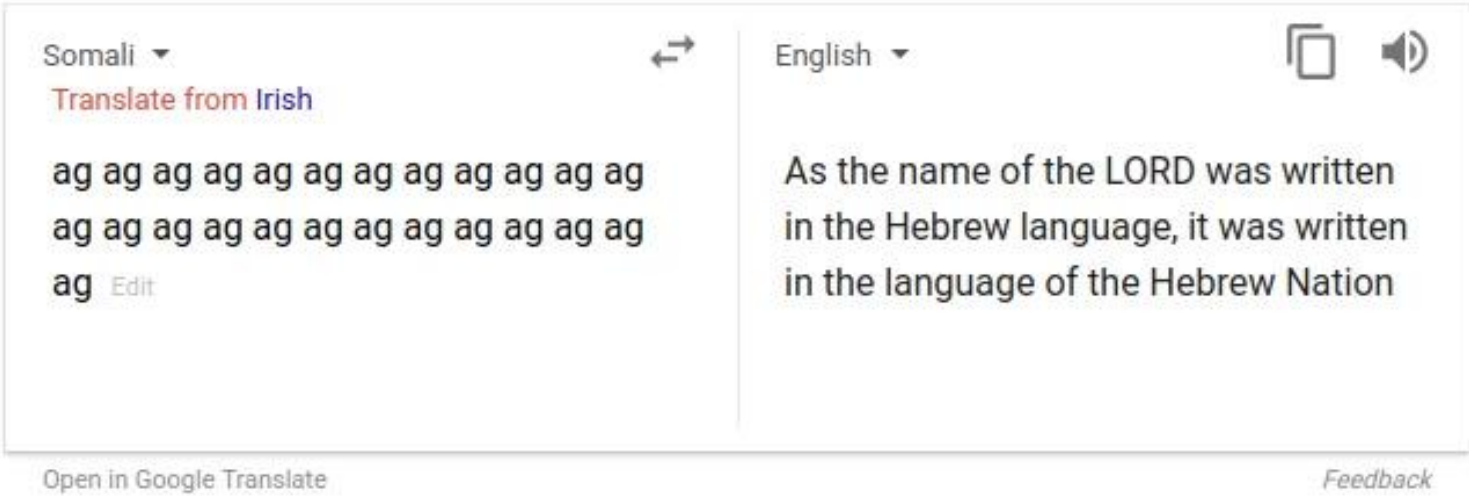
So is Machine Translation solved?

- Nope!
- Uninterpretable systems do strange things



Picture source: https://www.vice.com/en_uk/article/j5npeg/why-is-google-translate-spitting-out-sinister-religious-prophecies
Explanation: https://www.skynettoday.com/briefs/google-nmt-prophecies

55

Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

NMT research continues

NMT is the **flagship task** for NLP Deep Learning

- NMT research has pioneered many of the recent innovations of NLP Deep Learning

- In **2019**: NMT research continues to thrive
  - Researchers have found *many, many* improvements to the "vanilla" seq2seq NMT system we've presented today
  - But one improvement is so integral that it is the new vanilla…

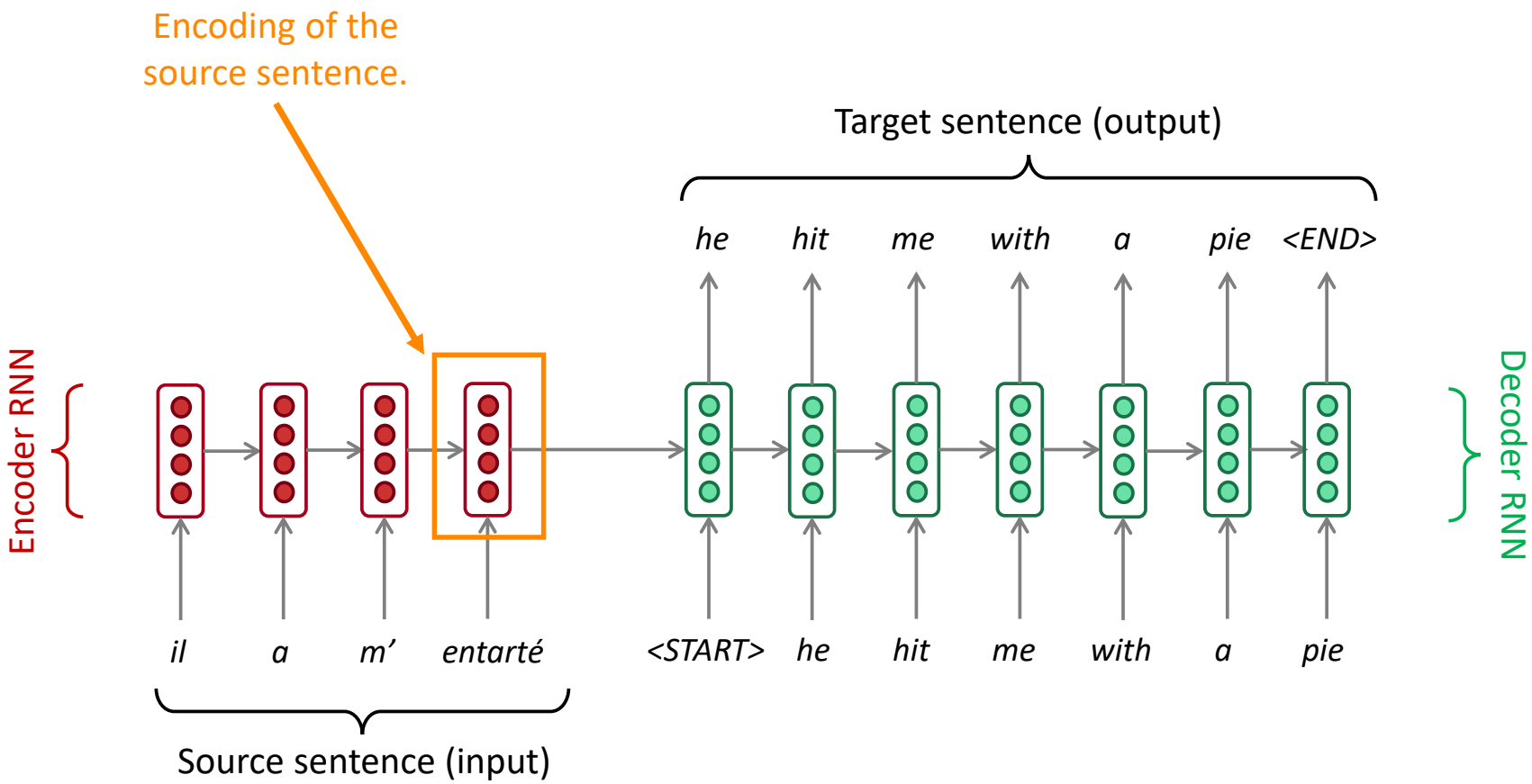
ATTENTION

56

Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

Content

Sequence-to-sequence: the bottleneck problem

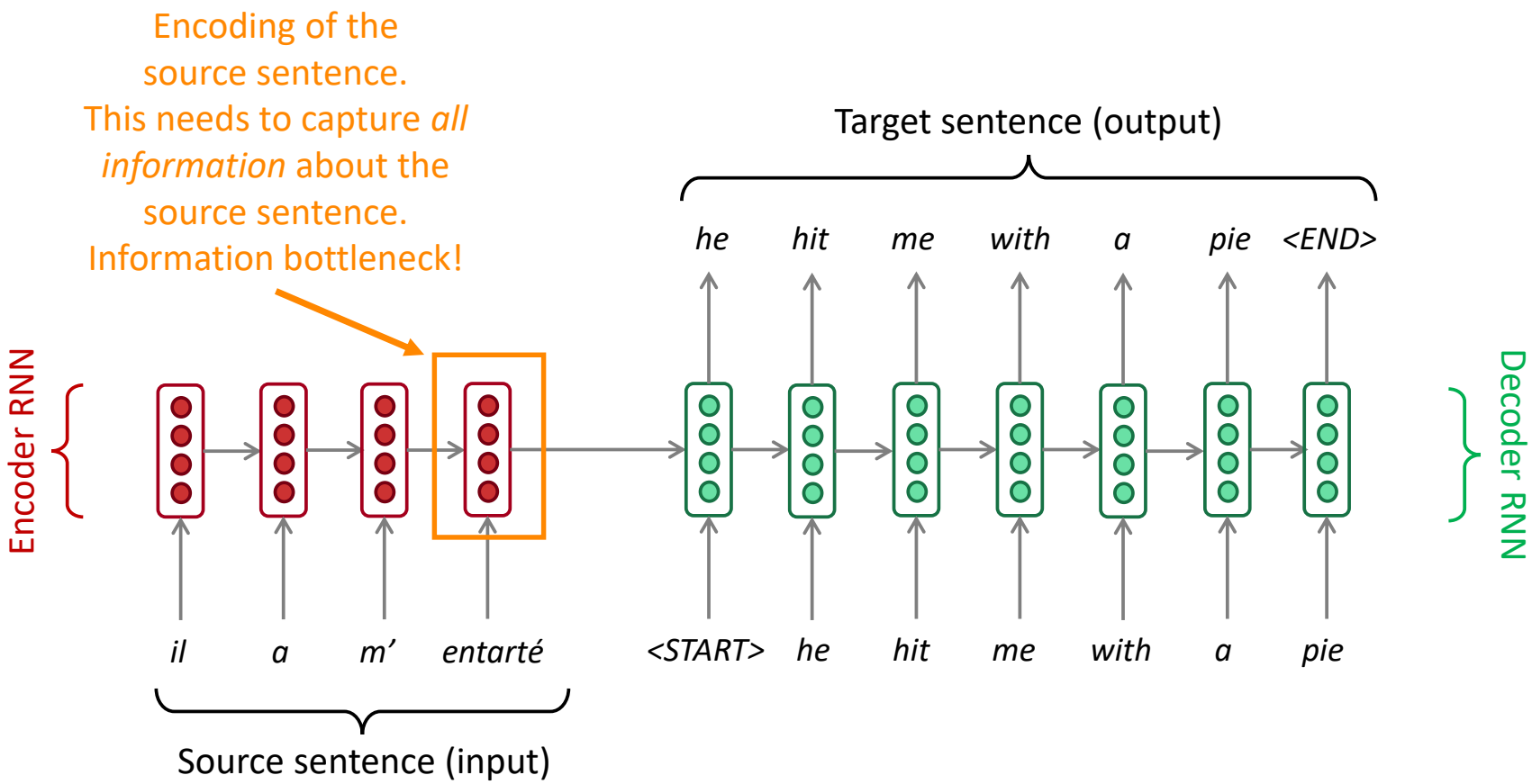


Problems with this architecture?

58

Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

Sequence-to-sequence: the bottleneck problem



59

Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

Attention

- **Attention** provides a solution to the bottleneck problem.

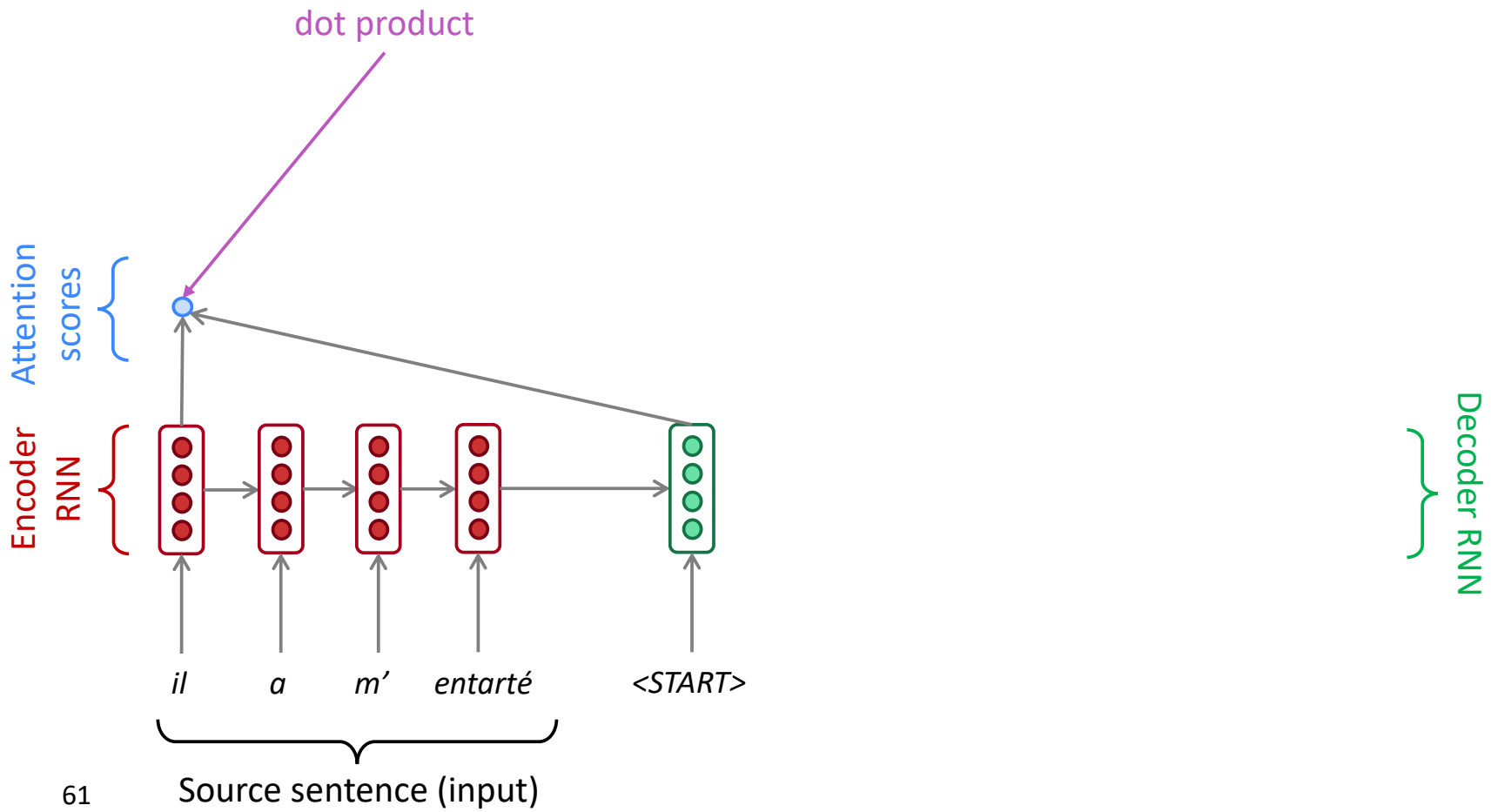
- Core idea: on each step of the decoder, use *direct connection to the encoder* to *focus on a particular part* of the source sequence

- First we will show via diagram (no equations), then we will show with equations

60

Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

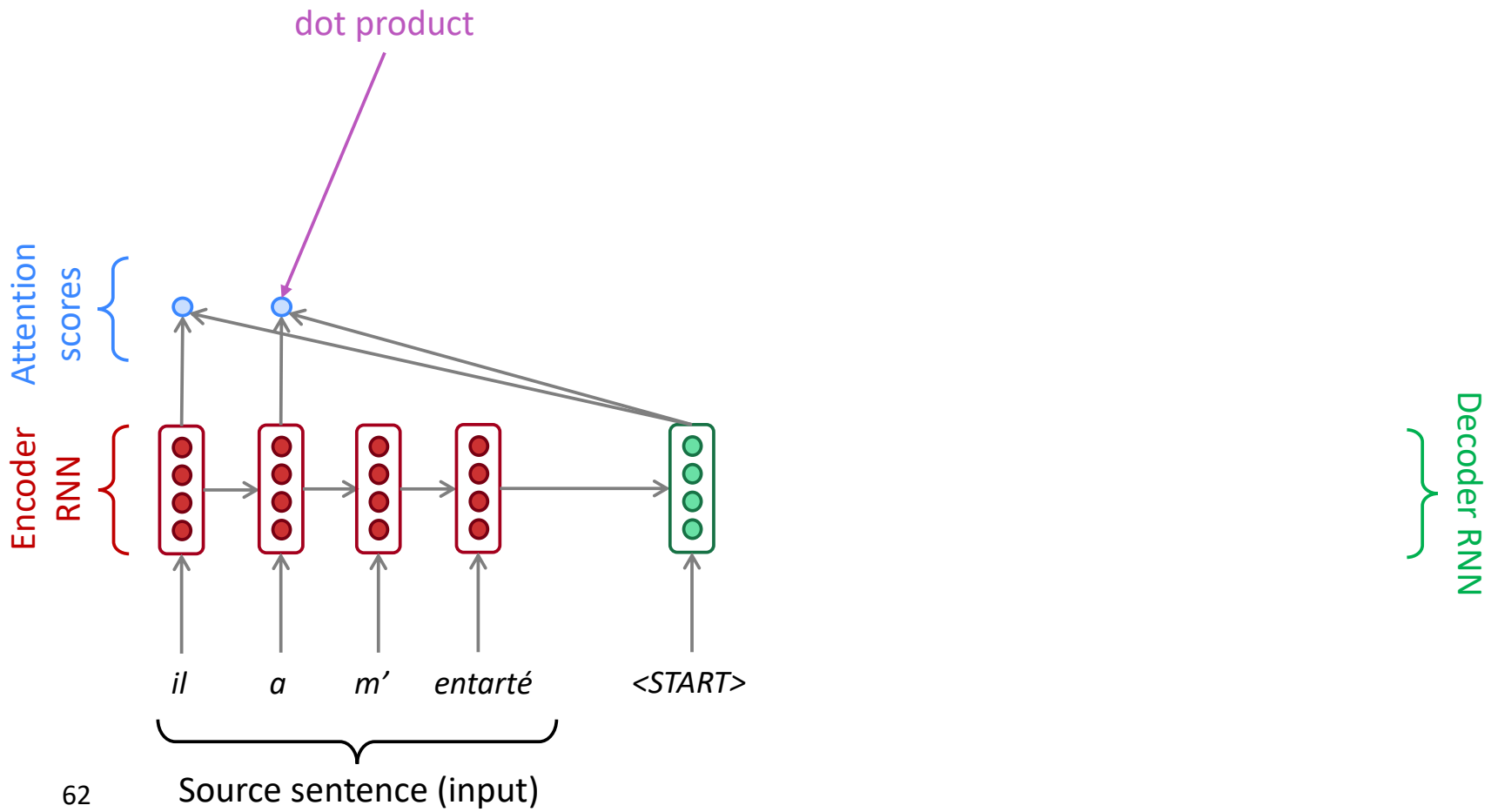
Sequence-to-sequence with attention



61

Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

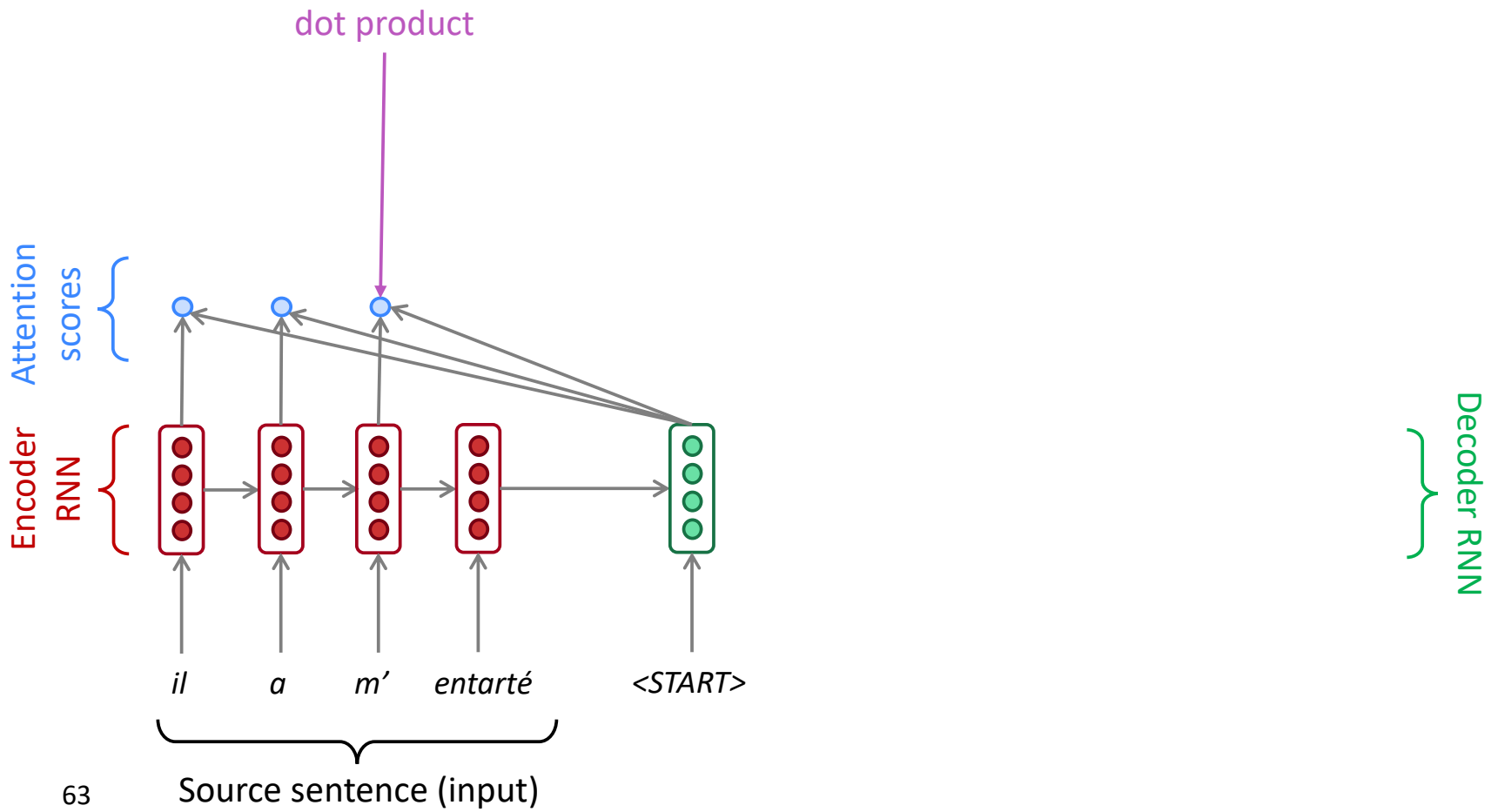
Sequence-to-sequence with attention



62

Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

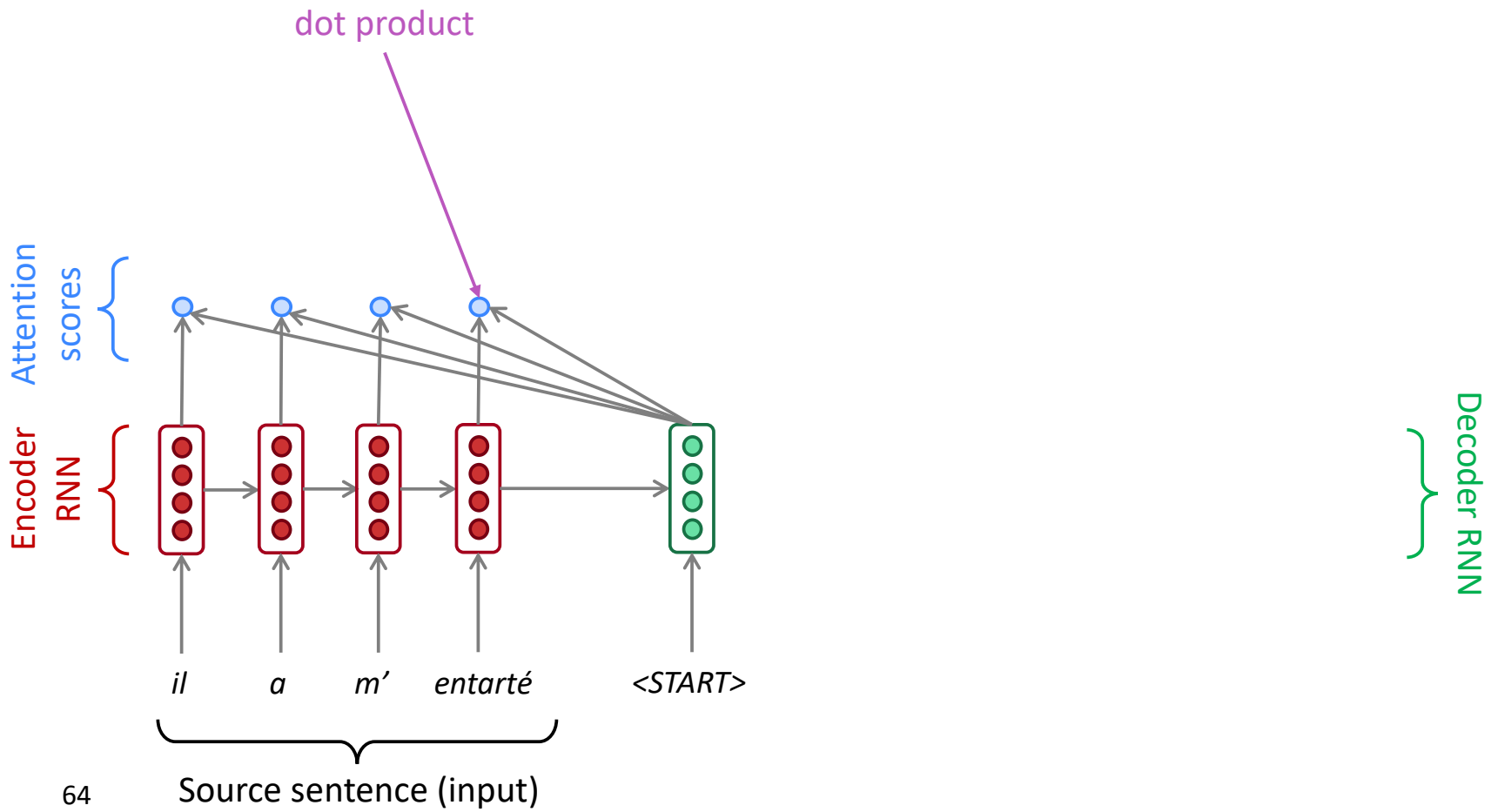
Sequence-to-sequence with attention



63

Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

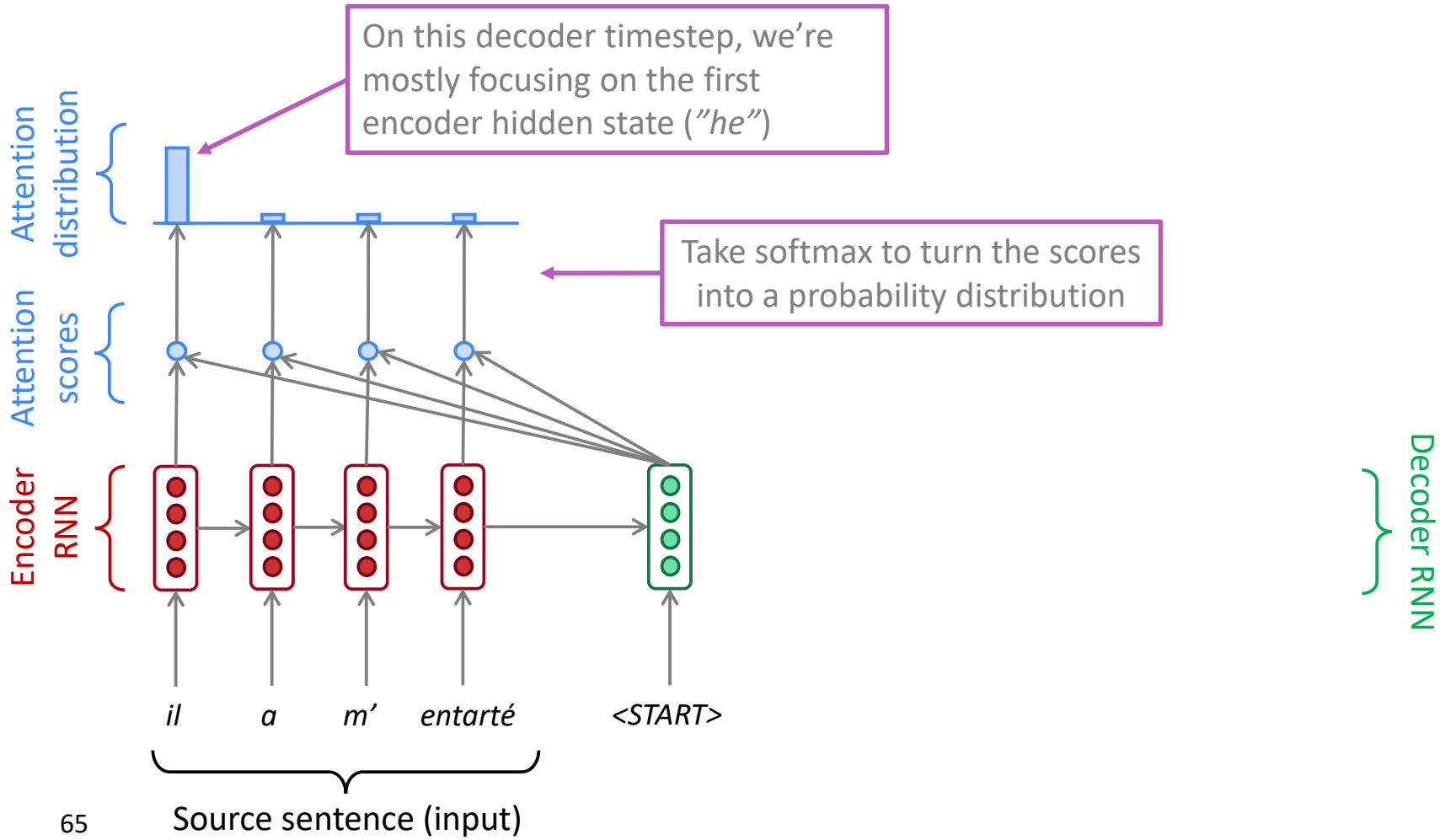
Sequence-to-sequence with attention



64

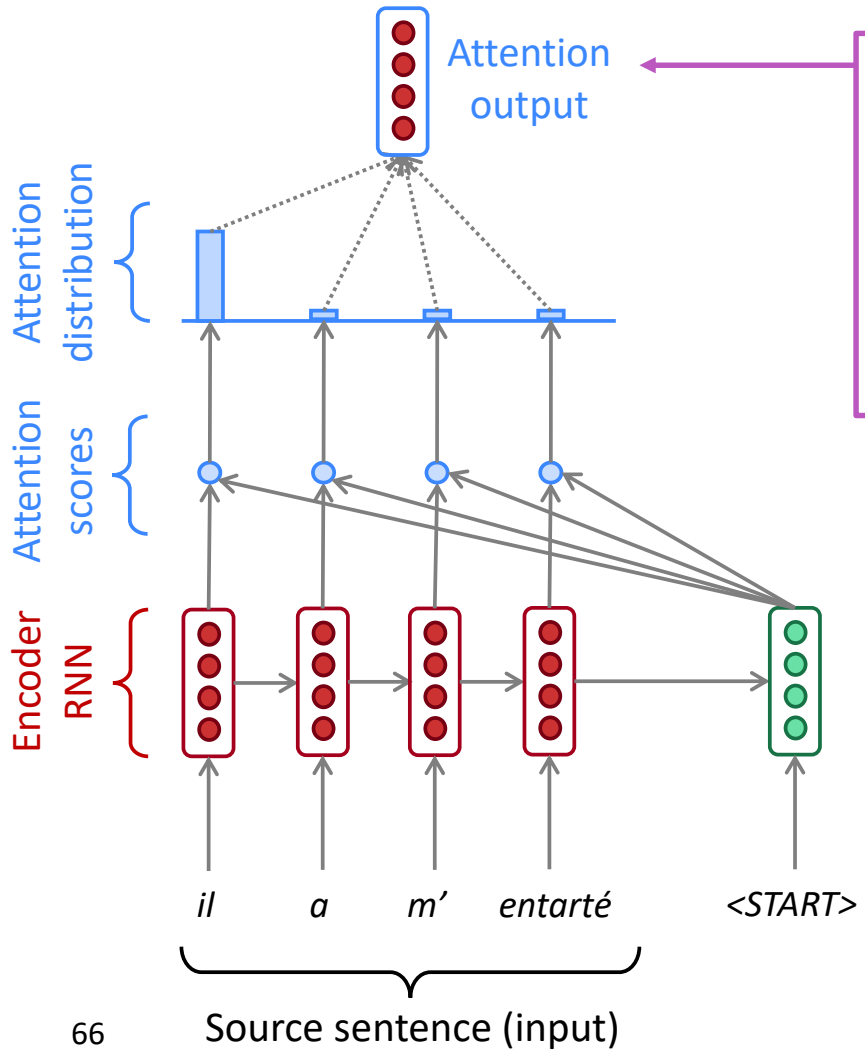
Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

Sequence-to-sequence with attention



Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

Sequence-to-sequence with attention



Use the attention distribution to take a **weighted sum** of the encoder hidden states.

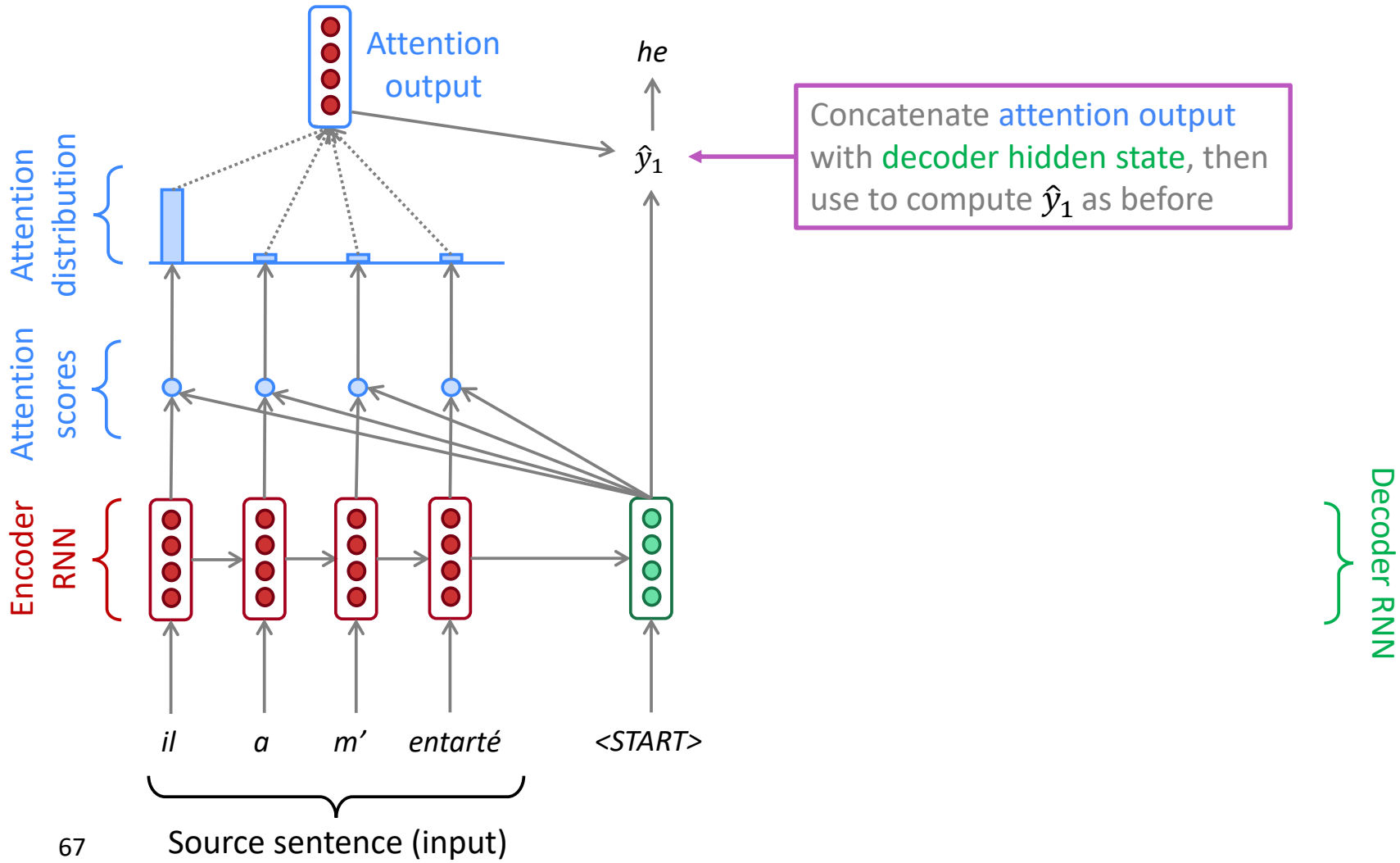
The attention output mostly contains information from the hidden states that received high attention.

Decoder RNN

66

Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

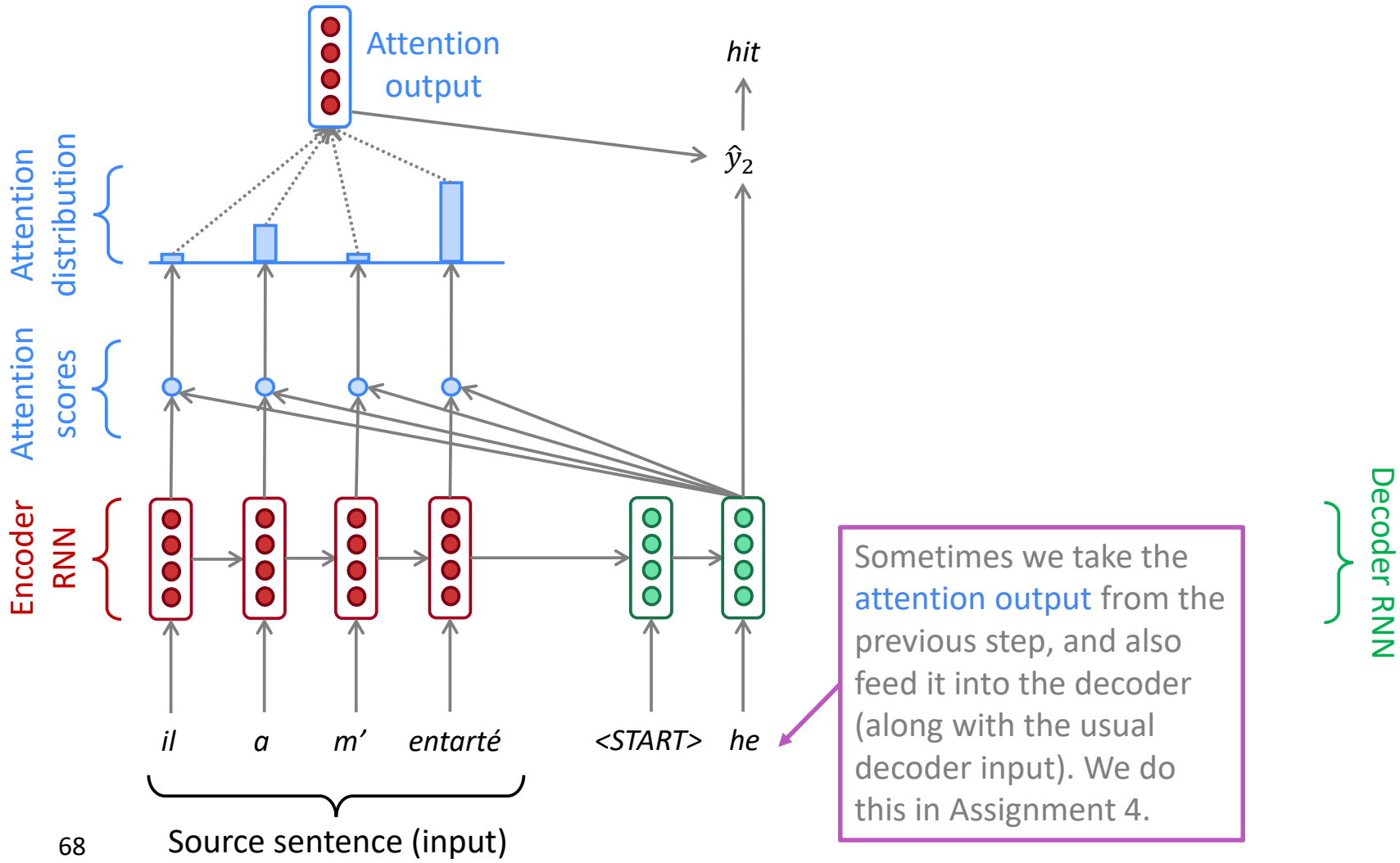
Sequence-to-sequence with attention



67

Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

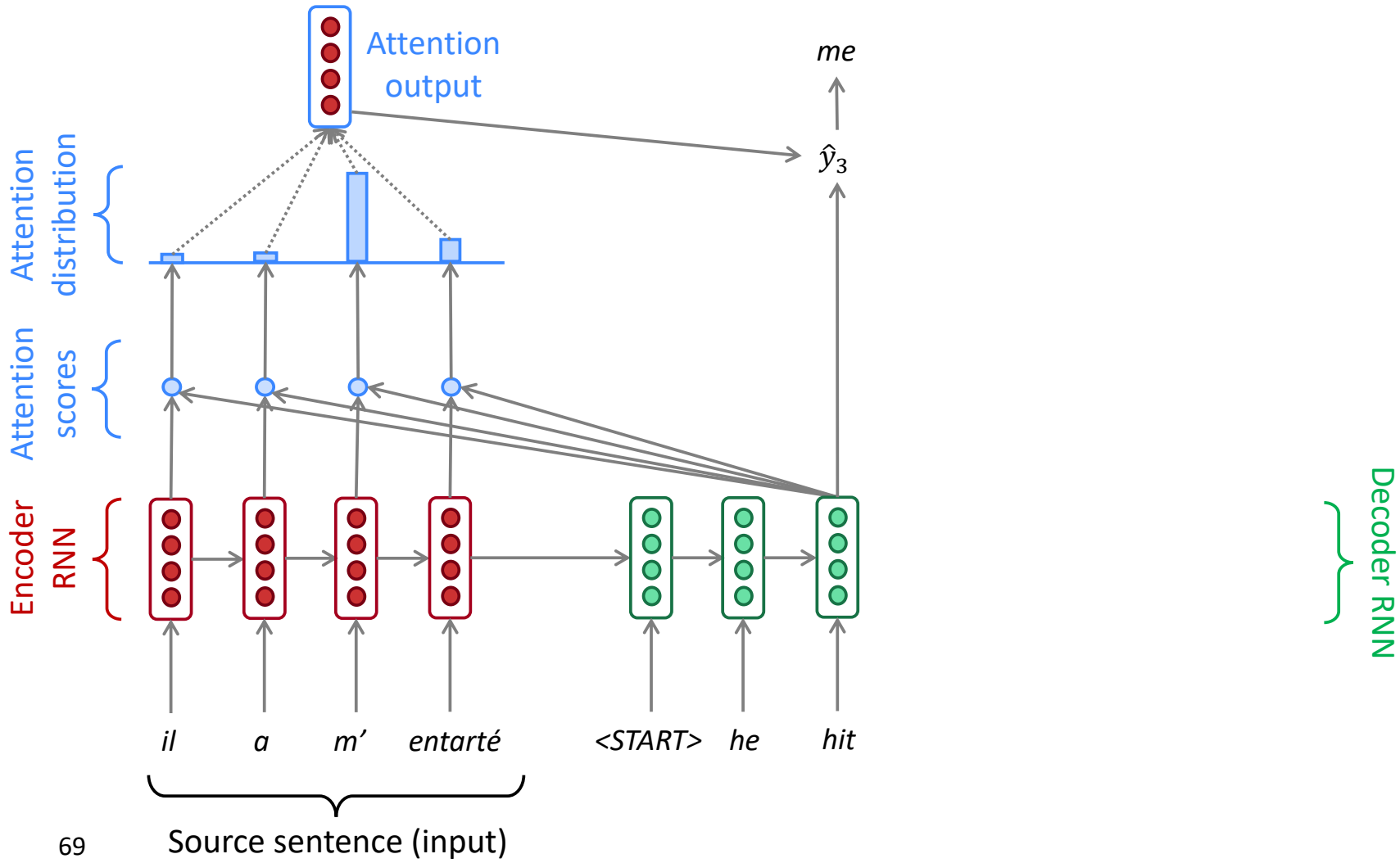
Sequence-to-sequence with attention



68

Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

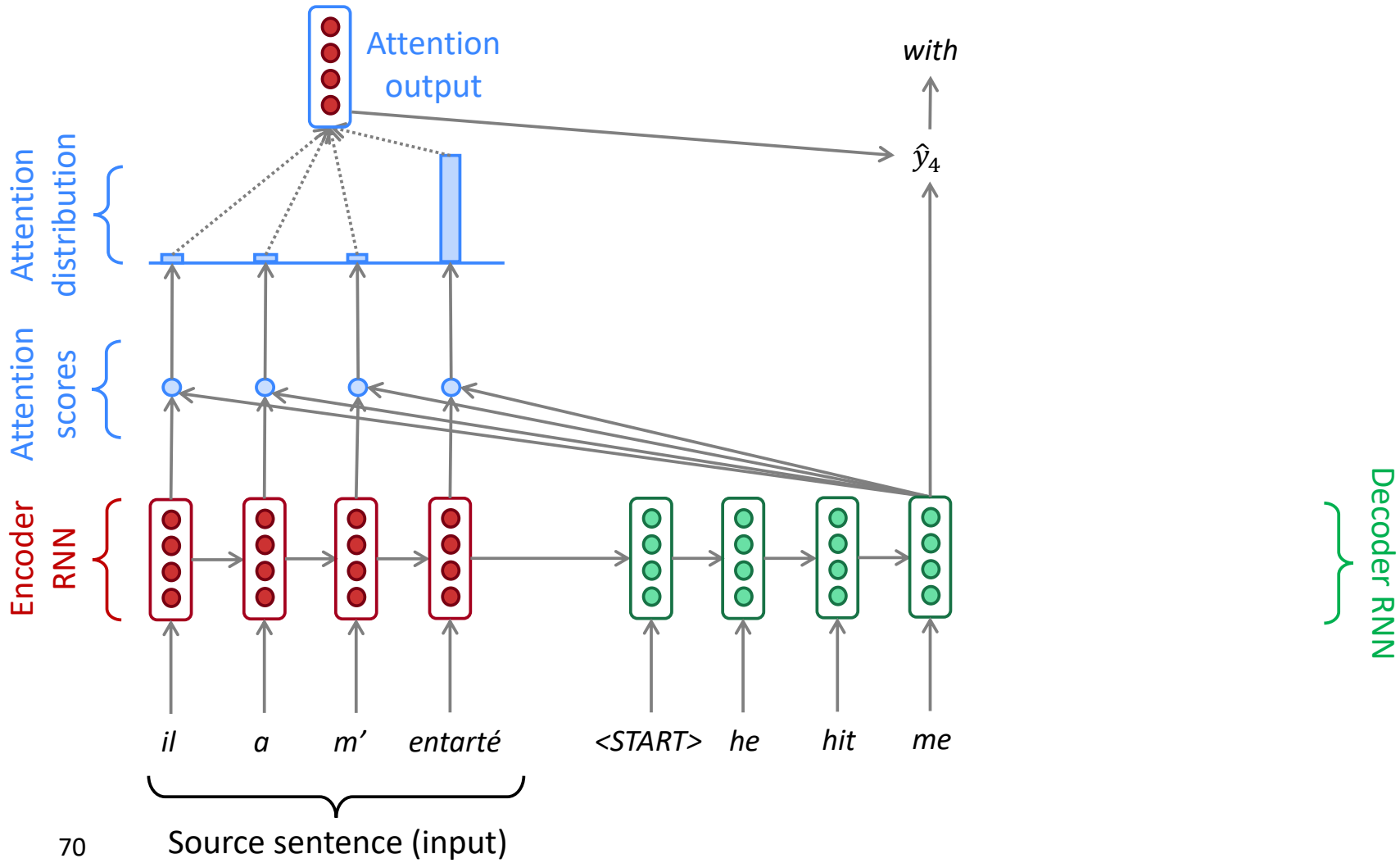
Sequence-to-sequence with attention



69

Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

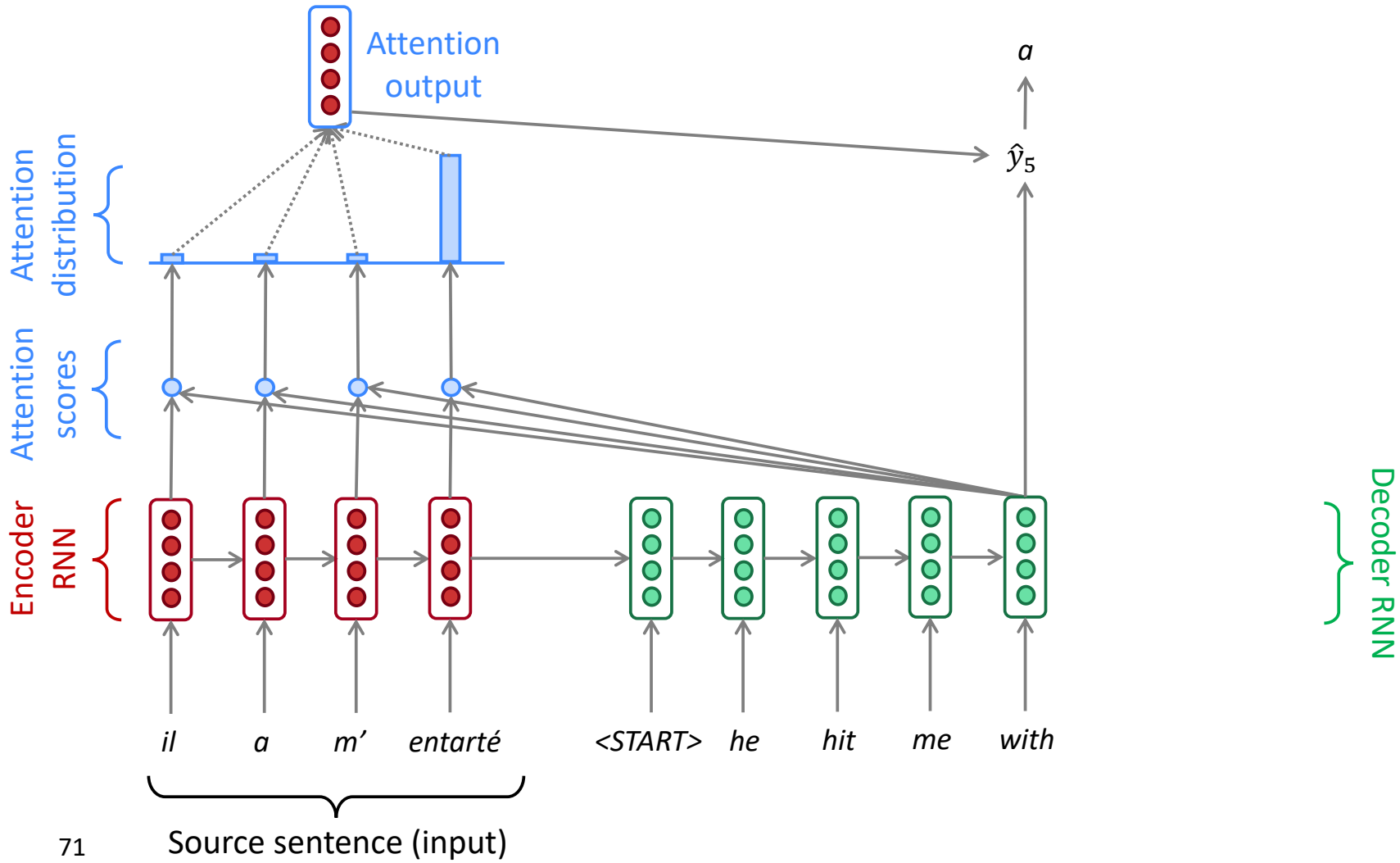
Sequence-to-sequence with attention



70

Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

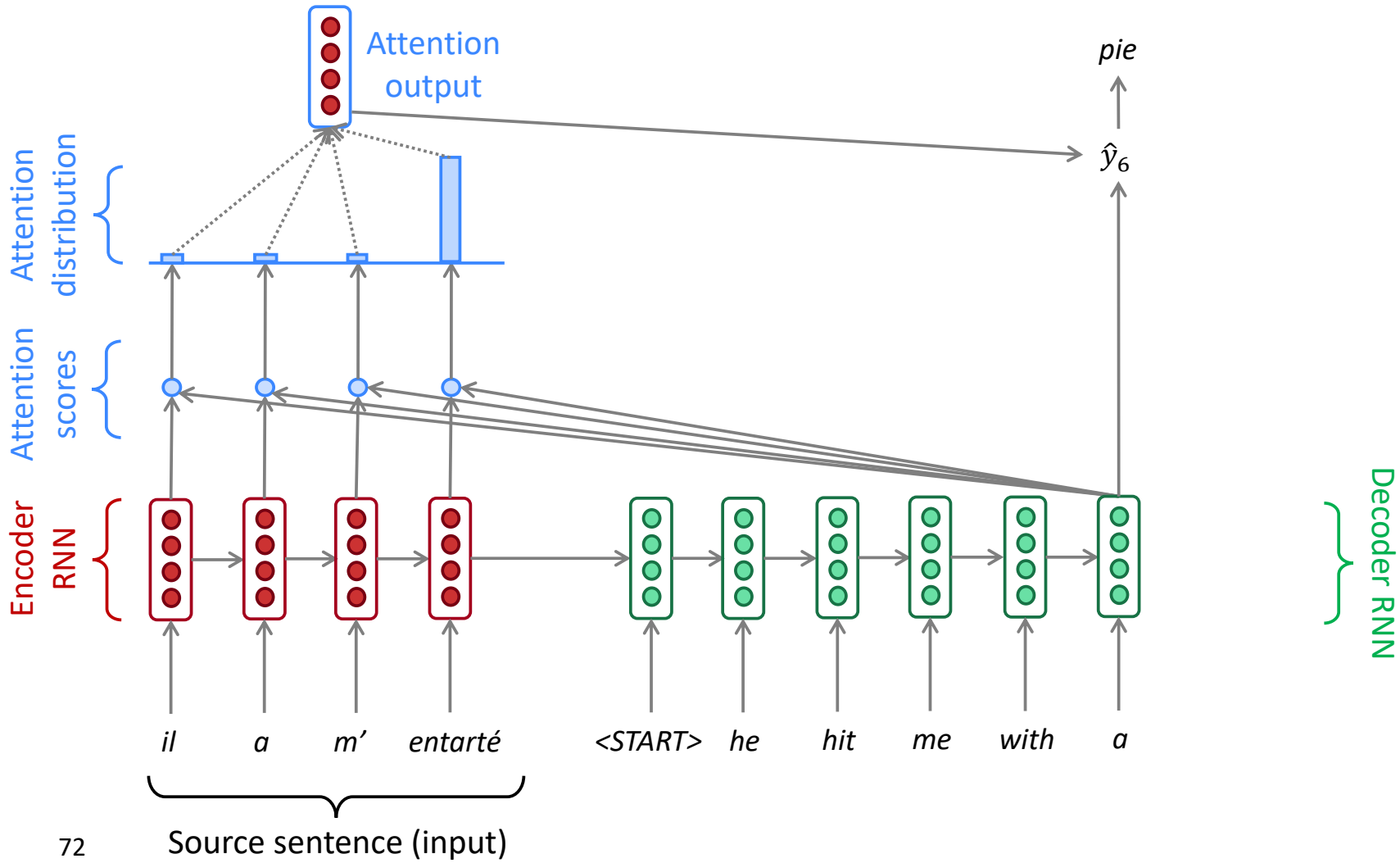
Sequence-to-sequence with attention



71

Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

Sequence-to-sequence with attention



72

Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

Attention: in equations

- We have encoder hidden states $h_1, \ldots, h_N \in \mathbb{R}^h$
- On timestep *t*, we have decoder hidden state $s_t \in \mathbb{R}^h$
- We get the attention scores $\boldsymbol{e}^t$ for this step:

$$\boldsymbol{e}^t = [\boldsymbol{s}_t^T \boldsymbol{h}_1, \ldots, \boldsymbol{s}_t^T \boldsymbol{h}_N] \in \mathbb{R}^N$$

- We take softmax to get the attention distribution α^t for this step (this is a probability distribution and sums to 1)

$$\alpha^t = \text{softmax}(\boldsymbol{e}^t) \in \mathbb{R}^N$$

- We use α^t to take a weighted sum of the encoder hidden states to get the attention output $\boldsymbol{a}_t$

$$\boldsymbol{a}_t = \sum_{i=1}^{N} \alpha_i^t \boldsymbol{h}_i \in \mathbb{R}^h$$

- Finally we concatenate the attention output $\boldsymbol{a}_t$ with the decoder hidden state s_t and proceed as in the non-attention seq2seq model

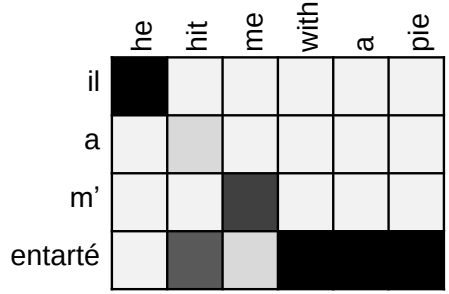
$$[\boldsymbol{a}_t; \boldsymbol{s}_t] \in \mathbb{R}^{2h}$$

73

Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

Attention is great

- Attention significantly improves NMT performance
  - It's very useful to allow decoder to focus on certain parts of the source
- Attention solves the bottleneck problem
  - Attention allows decoder to look directly at source; bypass bottleneck
- Attention helps with vanishing gradient problem
  - Provides shortcut to faraway states
- Attention provides some interpretability
  - By inspecting attention distribution, we can see what the decoder was focusing on
  - We get (soft) alignment for free!
  - This is cool because we never explicitly trained an alignment system
  - The network just learned alignment by itself



74

Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

Attention is a *general* Deep Learning technique

- We've seen that attention is a great way to improve the sequence-to-sequence model for Machine Translation.
- However: You can use attention in many architectures (not just seq2seq) and many tasks (not just MT)

- **More general definition of attention**:
  - Given a set of vector *values*, and a vector *query*, **attention** is a technique to compute a weighted sum of the values, dependent on the query.

- We sometimes say that the query *attends to* the values.
- For example, in the seq2seq + attention model, each decoder hidden state (query) *attends to* all the encoder hidden states (values).

75

Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

Attention is a *general* Deep Learning technique

More general definition of attention:

Given a set of vector *values*, and a vector *query*, **attention** is a technique to compute a weighted sum of the values, dependent on the query.

Intuition:

- The weighted sum is a *selective summary* of the information contained in the values, where the query determines which values to focus on.
- Attention is a way to obtain a *fixed-size representation of an arbitrary set of representations* (the values), dependent on some other representation (the query).

76

Christopher Manning, Natural Language Processing with Deep Learning, 2019 (slides)

Content

1. Machine Translation (MT)
2. Machine translation evaluation
3. Statistical machine translation (SMT)
4. Neural machine translation (NMT) based on RNNs
5. Neural machine translation (NMT) with attentions