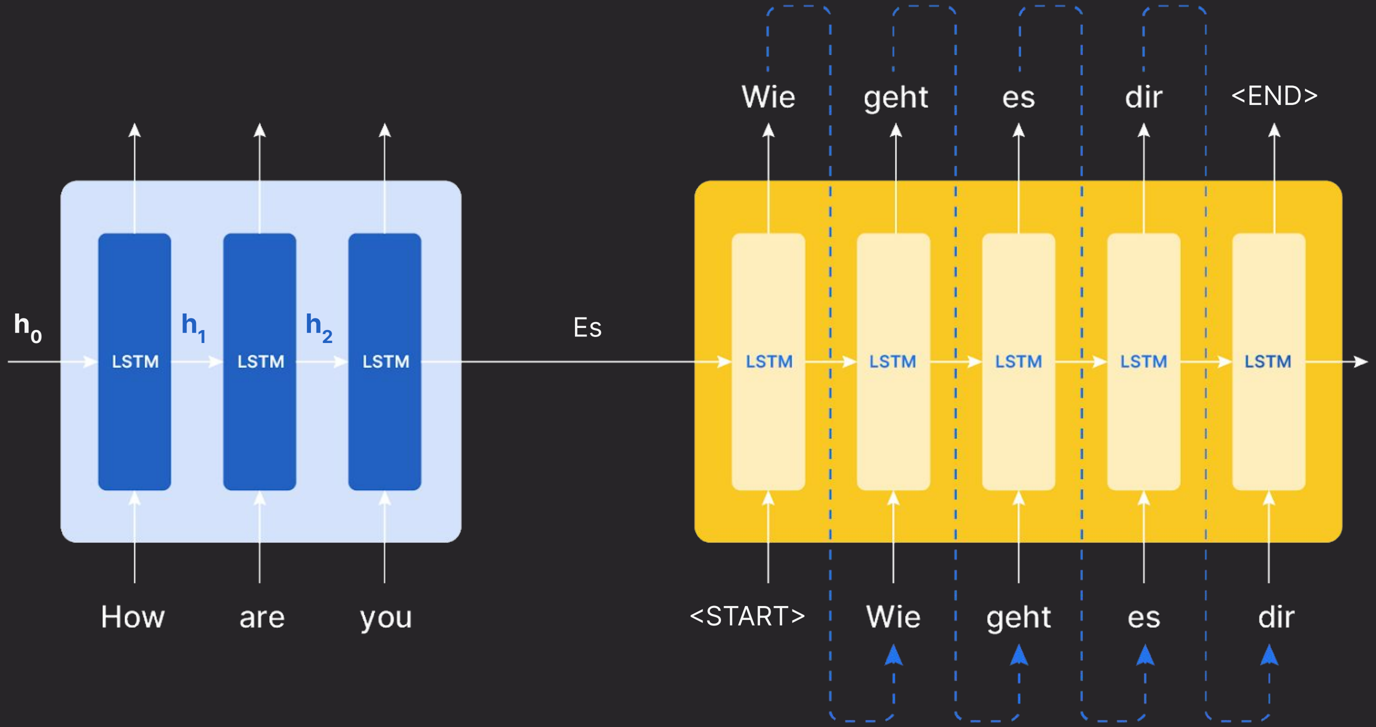
Attention Mechanism and Transformers

Video 5: Attention Mechanism

Analytics Vidhya

Recap

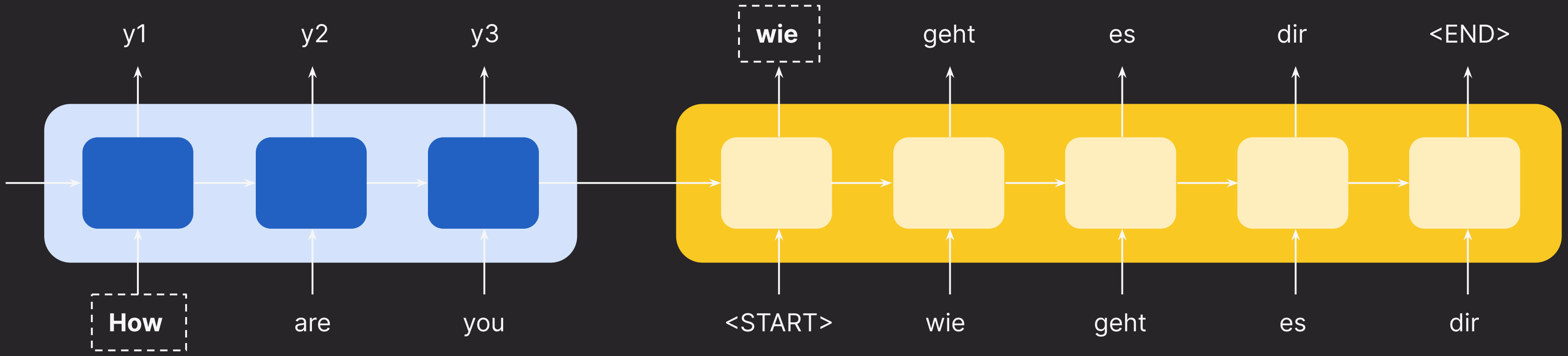


Wie
geht
es
dir
<END>
h_0
LSTM
h_1
LSTM
h_2
LSTM
Es
LSTM
LSTM
LSTM
LSTM
LSTM
How
are
you
<START>
Wie
geht
es
dir

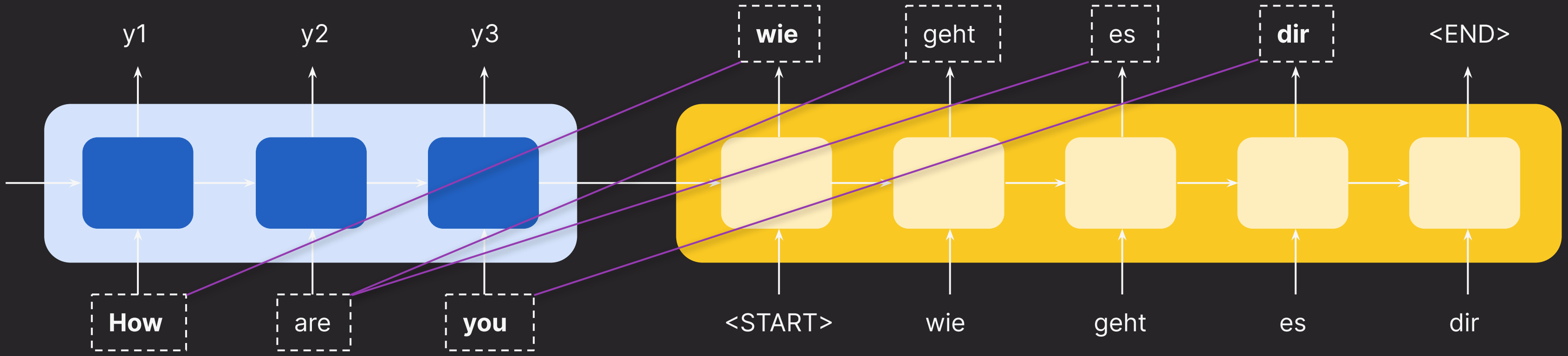
Limitations of Encoder-Decoder

- **Encoder state burden:** Carries all data from encoder to decoder where encoder errors may lead to translation failures.
- For long texts, **Encoder state may miss key info**
- The model **struggles to prioritize important details** at each time step

Limitations of Encoder-Decoder



Attention Mechanism



Attention

Analytics Vidhya

[Submitted on 1 Sep 2014 (v1), last revised 19 May 2016 (this version, v7)]

Neural Machine Translation by Jointly Learning to Align and Translate

Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio

Neural machine translation is a recently proposed approach to machine translation. Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. The models proposed recently for neural machine translation often belong to a family of encoder-decoders and consists of an encoder that encodes a source sentence into a fixed-length vector from which a decoder generates a translation. In this paper, we conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder-decoder architecture, and propose to extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly. With this new approach, we achieve a translation performance comparable to the existing state-of-the-art phrase-based system on the task of English-to-French translation. Furthermore, qualitative analysis reveals that the (soft-)alignments found by the model agree well with our intuition.

Comments: Accepted at ICLR 2015 as oral presentation

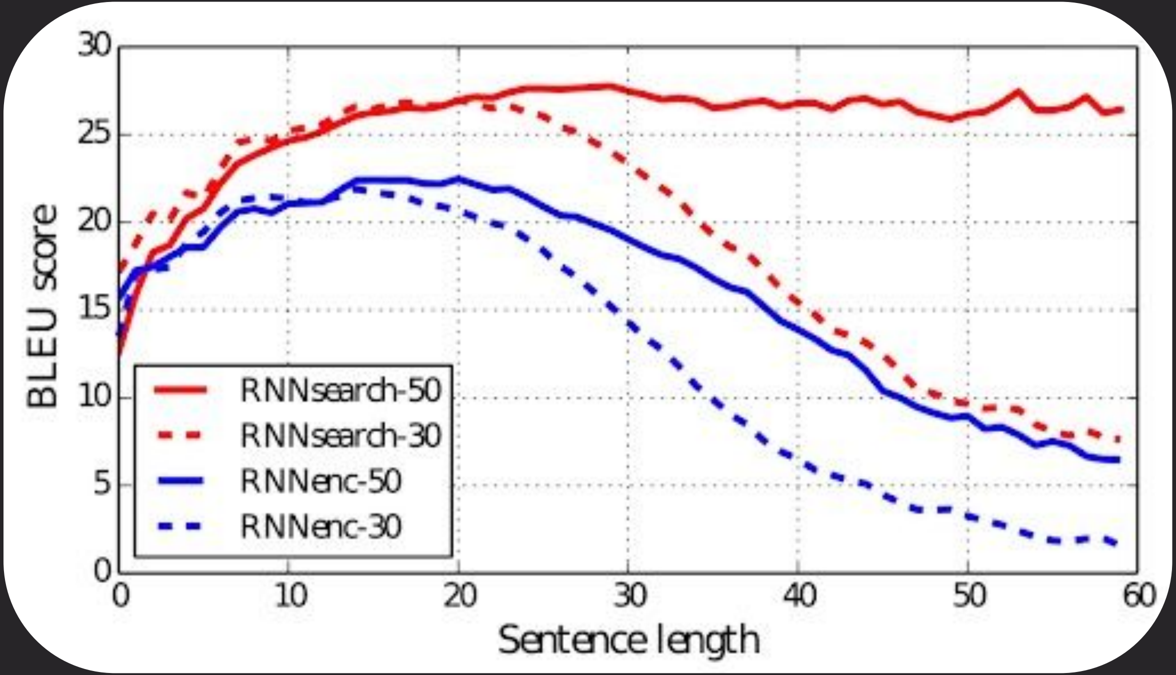
Subjects: **Computation and Language (cs.CL)**; Machine Learning (cs.LG); Neural and Evolutionary Computing (cs.NE); Machine Learning (stat.ML)

Cite as: arXiv:1409.0473 **[cs.CL]**

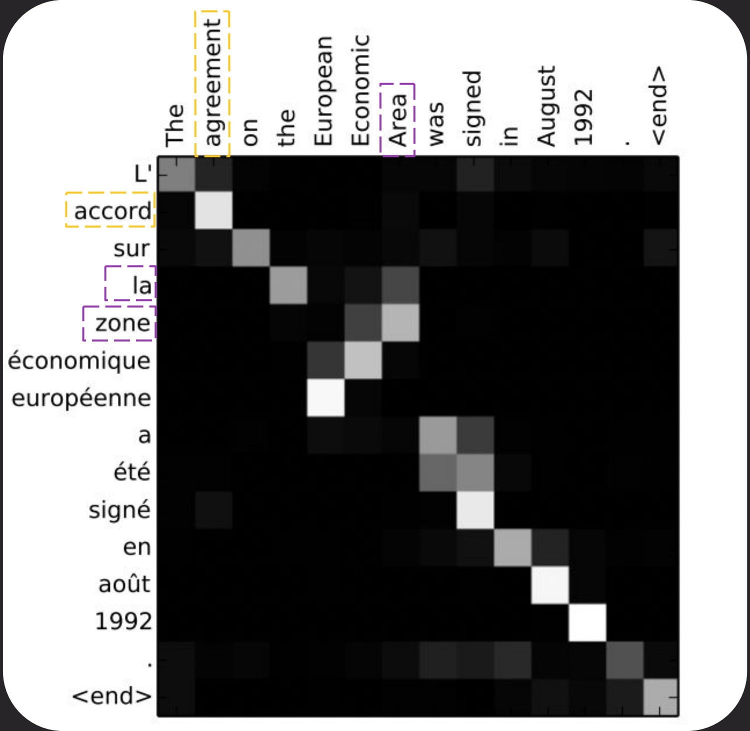
(or arXiv:1409.0473v7 **[cs.CL]** for this version)

https://doi.org/10.48550/arXiv.1409.0473

Impact of Attention Mechanism



Impact of Attention Mechanism

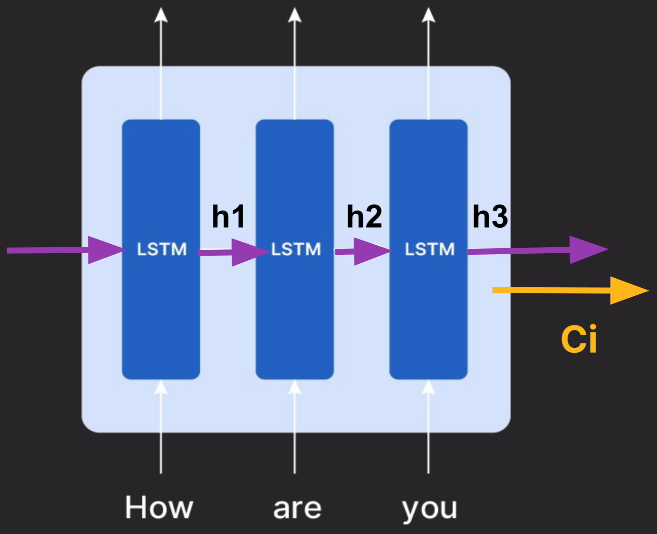


- Europèenne → European
- Agreement → Accord
- Area → Zone

Translate the following to German:

English - "how are you" → **German**- "wei geht es dir"

Attention Context Vector



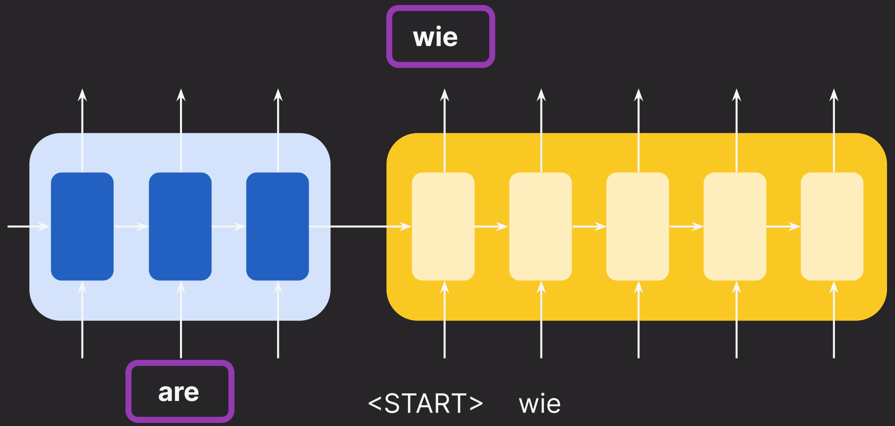
- Attention context vector is the weighted sum of the hidden states of the encoder.

Attention context vector (Ci)

$$C_i = \sum \alpha_{ij} h_i$$

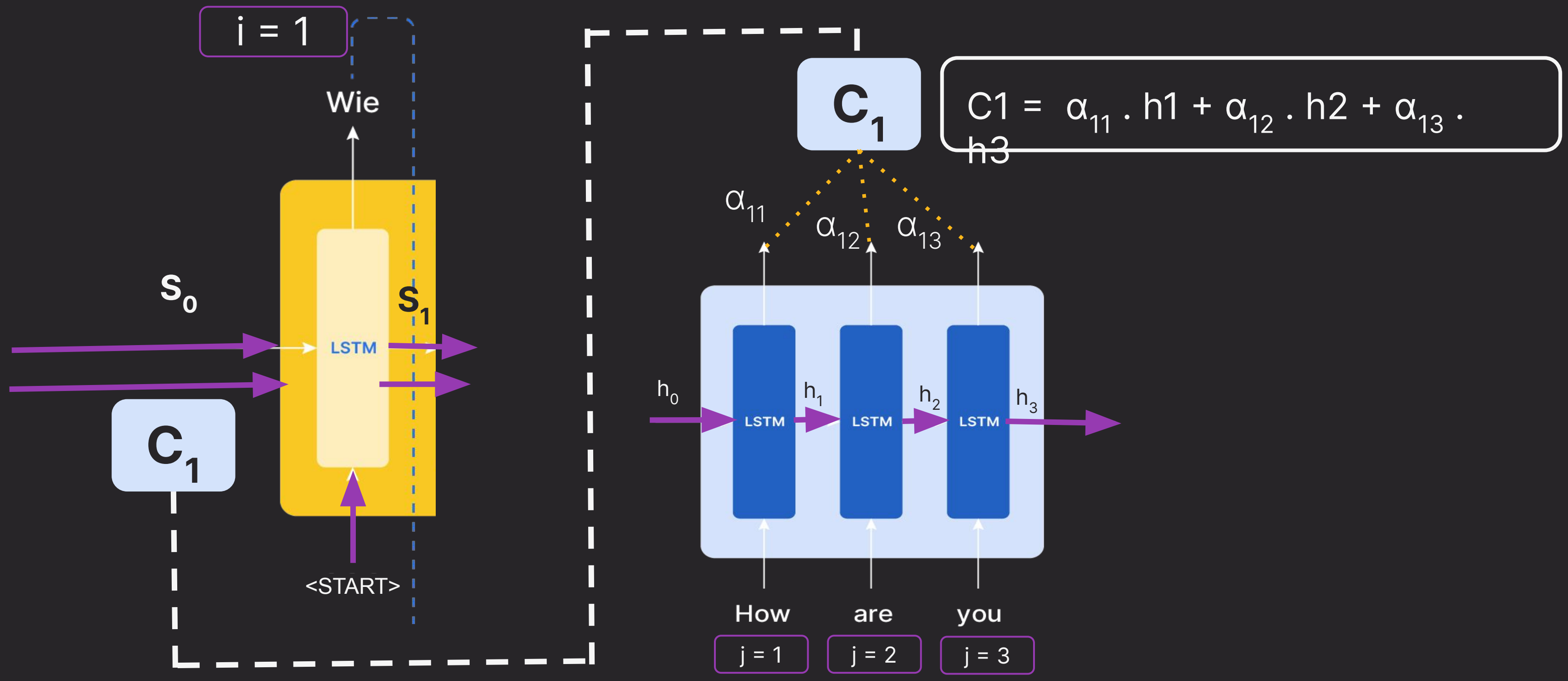
Attention weights

α_{ij} weights for ith time step in decoder and jth in the encoder

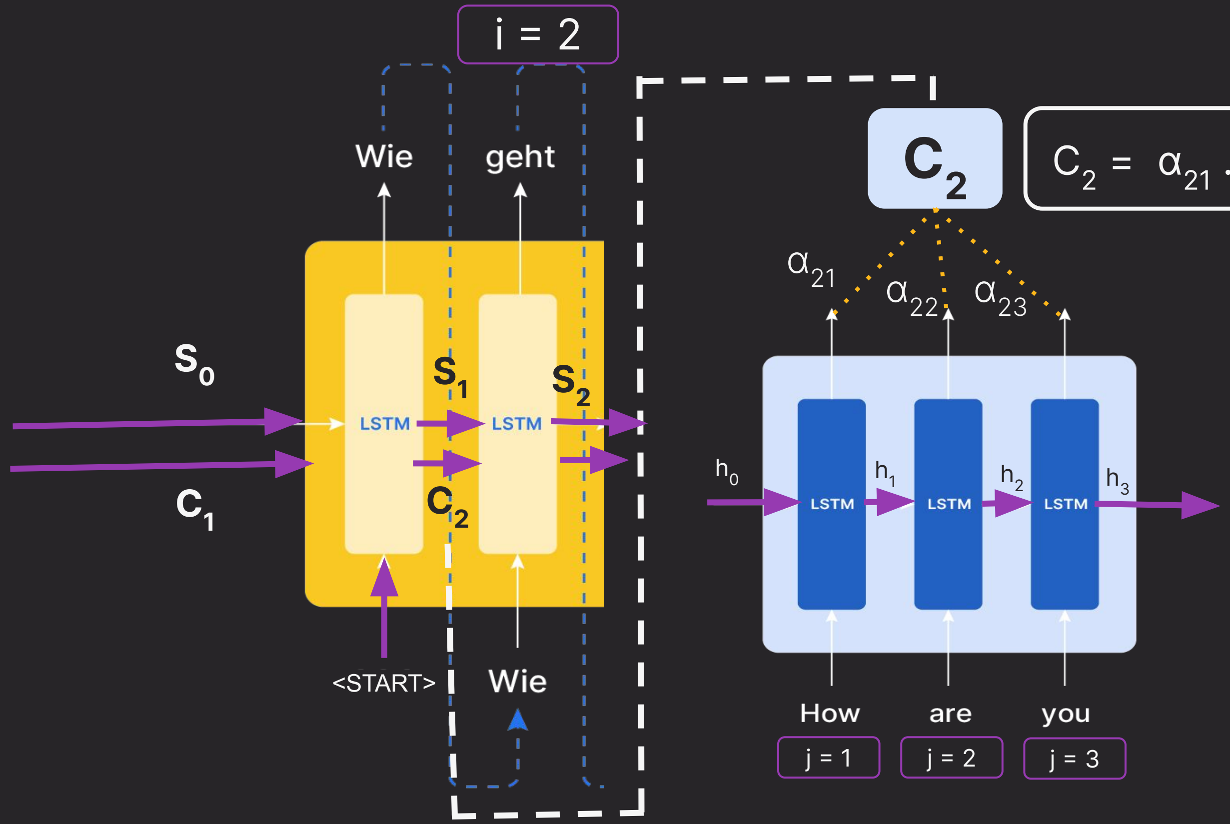


	How	Are	you
Wie	α_{11}	α_{12}	α_{13}
geht	α_{21}	α_{22}	α_{23}
er	α_{31}	α_{32}	α_{33}
dis	α_{41}	α_{42}	α_{43}

Attention Context Vector

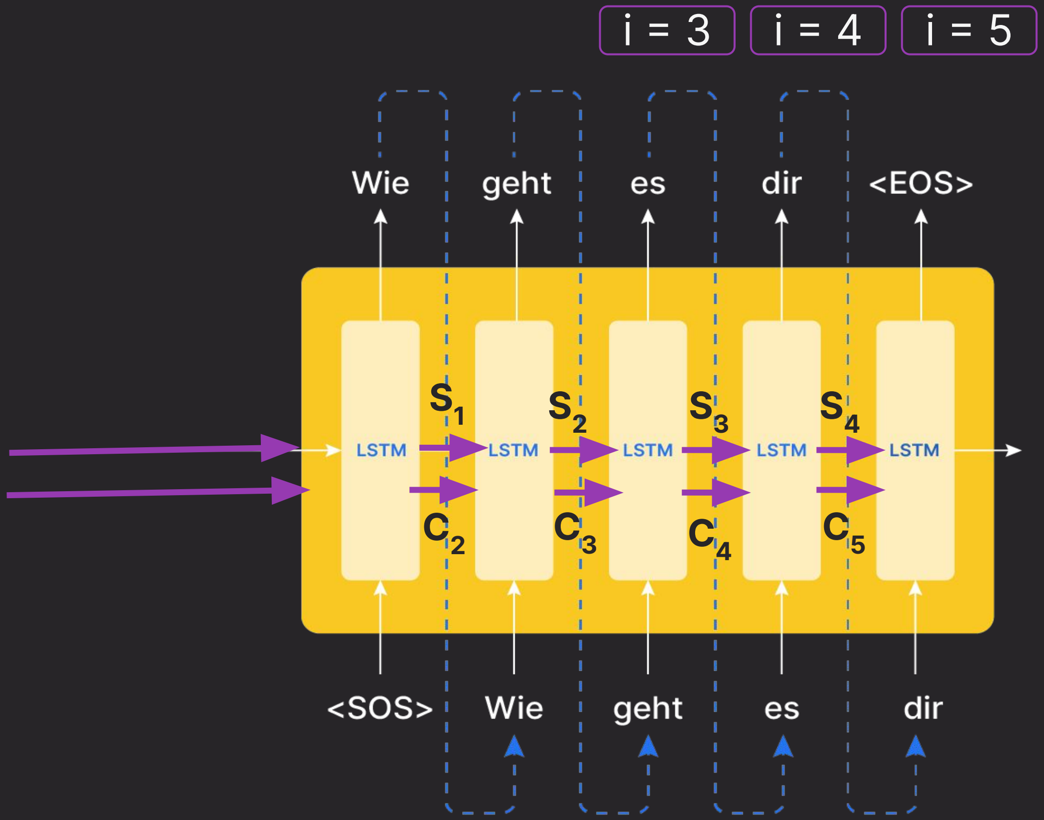


Attention Context Vector



$$C_2 = \alpha_{21} . h_1 + \alpha_{22} . h_2 + \alpha_{23} . h_3$$

Attention Context Vector



$$C_3 = \alpha_{31} . h_1 + \alpha_{32} . h_2 + \alpha_{33} . h_3$$

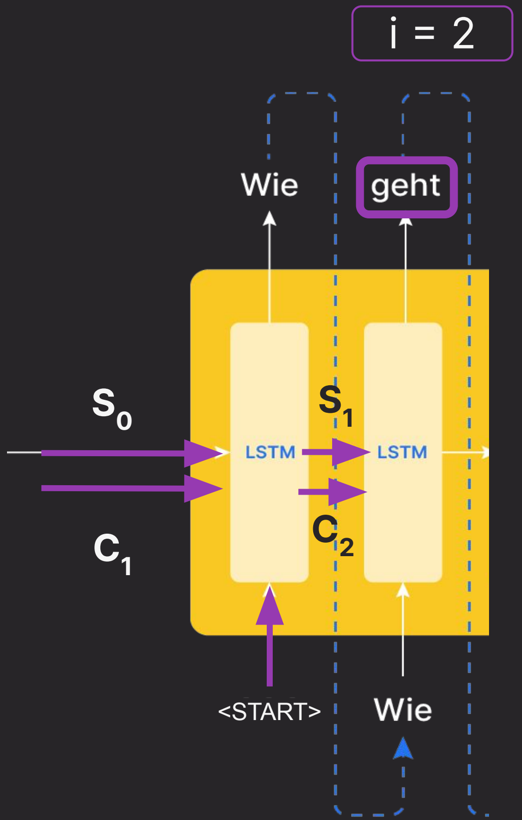
$$C_4 = \alpha_{41} . h_1 + \alpha_{42} . h_2 + \alpha_{43} . h_3$$

$$C_5 = \alpha_{51} . h_1 + \alpha_{52} . h_2 + \alpha_{53} . h_3$$

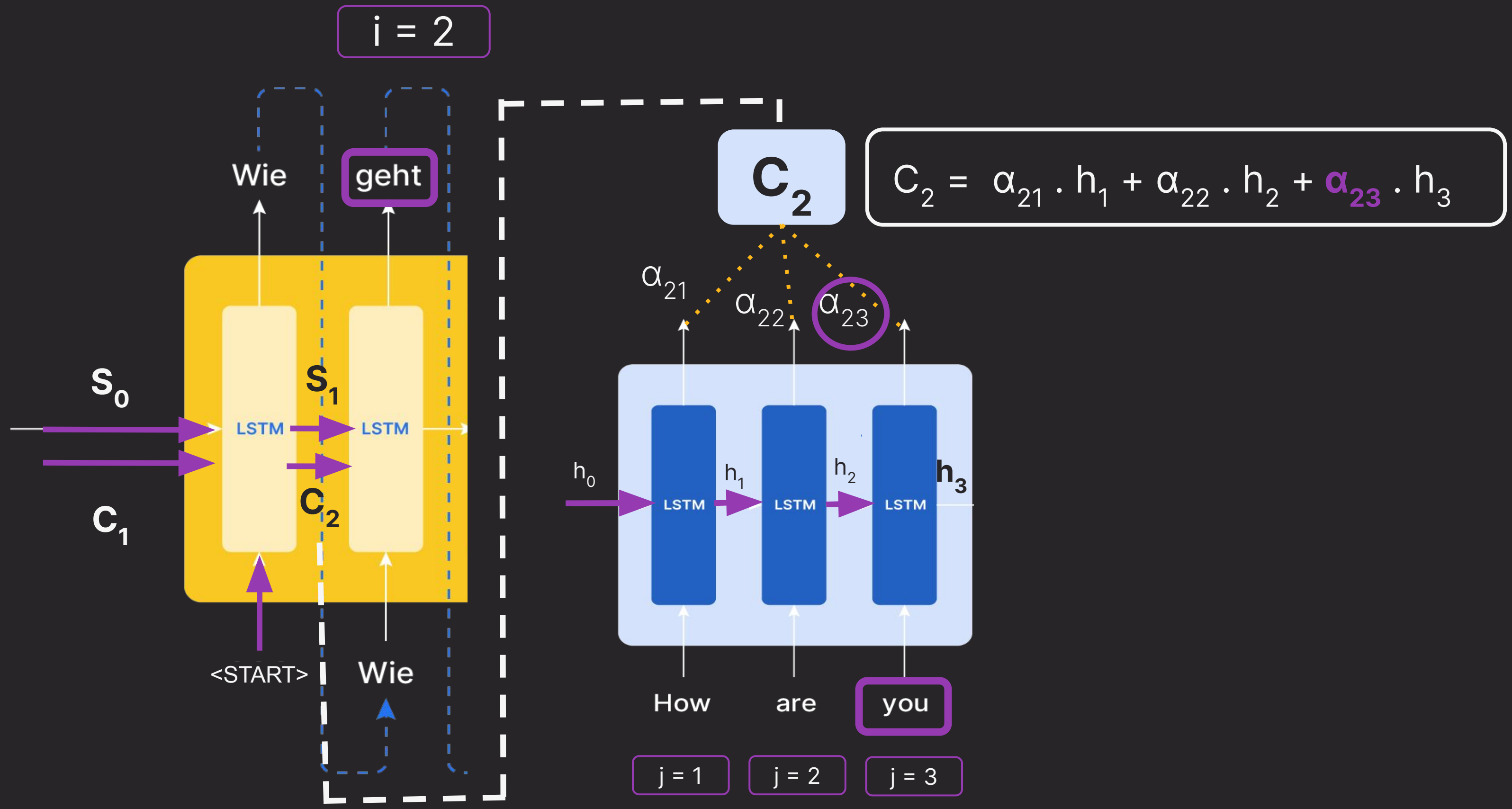
$$C_i = \sum \alpha_{ij} h_i$$

Attention Weights

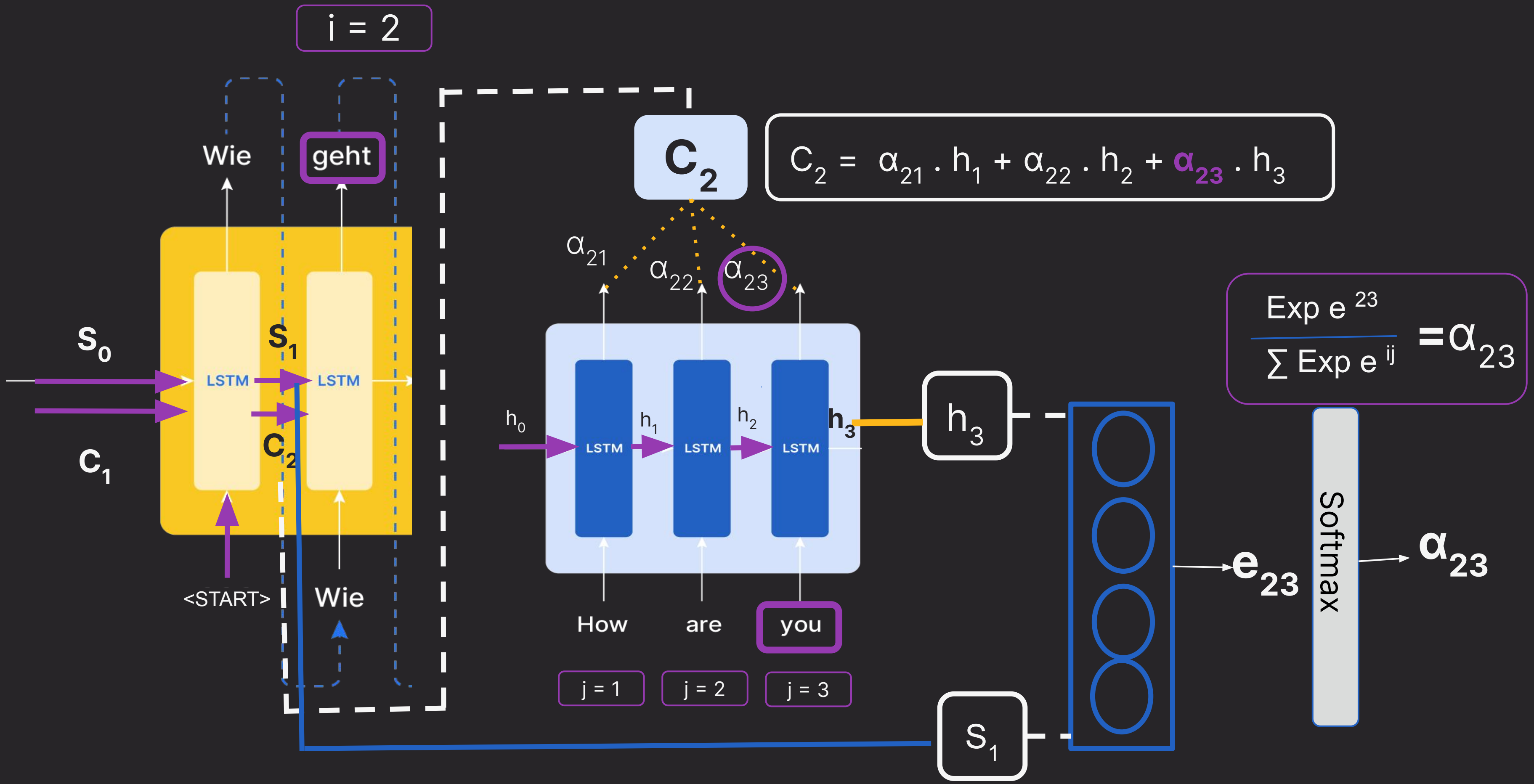
	How	Are	you
Wie	α_{11}	$\boldsymbol{\alpha_{12}}$	α_{13}
geht	α_{21}	$\boldsymbol{\alpha_{22}}$	$\boldsymbol{\alpha_{23}}$
er	α_{31}	α_{32}	α_{33}
dis	α_{41}	α_{42}	α_{43}



Analytics Vidhya



i = 2
Wie
geht
S_0
C_1
S_1
C_2
LSTM
LSTM
<START>
Wie
C_2
$C_2 = \alpha_{21} . h_1 + \alpha_{22} . h_2 + \alpha_{23} . h_3$
α_{21}
α_{22}
α_{23}
h_0
h_1
h_2
h_3
LSTM
LSTM
LSTM
How
are
you
j = 1
j = 2
j = 3



i = 2
Wie
geht
S_0
C_1
S_1
C_2
LSTM
LSTM
<START>
Wie
C_2
$C_2 = \alpha_{21} . h_1 + \alpha_{22} . h_2 + \alpha_{23} . h_3$
α_{21}
α_{22}
α_{23}
h_0
h_1
h_2
h_3
LSTM
LSTM
LSTM
How
are
you
j = 1
j = 2
j = 3
h_3
S_1
$\frac{\text{Exp e}^{23}}{\sum \text{Exp e}^{ij}} = \alpha_{23}$
e_{23}
Softmax
α_{23}