Hands on Camembert

Building a dialog act classification model for French

Benjamin Muller Roman Castagne Nathan Godey INRIA Paris

Hands on Camembert

Session 1 [lecture]: Dialog Act Classification with Camembert

Session 2 [coding]: Camembert Language Modeling

Session 3 [coding]: Building a supervised model for dialog act prediction for French with Camembert

Hands on Camembert: Session 1

1. Framework: Focus on Dialog Act Prediction

2. Task-Specific Modeling
    a. Segmentation
    b. The Transformers Architecture
    c. Training

3. The Camembert model

Acknowledgement

We built Camembert in 2019 as part of a collaboration between **INRIA Paris (ALMANACH** team) and **Facebook AI**

Work done by **Louis Martin, Pedro Ortiz, Benjamin Muller** with the guidance of Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah and Benoît Sagot

Camembert derived from **BERT** (released by Google in 2018) and **ROBERTa** (released by Facebook in 2019).

Framework

Given a sequence of tokens $(w_1, .., w_T)$ goal is to find the best model $model_\theta$ to predict a label or sequence of label *Y*

$$model_\theta : \quad \mathcal{V}^T \quad \rightarrow \quad \Omega = \mathcal{L},\ \mathcal{V}'^{T'}$$

$$(w_1, .., w_T) \mapsto \hat{Y}$$

We may want to

- ➔ assign a category to each word (*sequence labeling*)
- ➔ assign a category to each sentence (*sequence classification*)
- ➔ Generate a sequence (*sequence generation*)

Framework

Given a sequence of tokens $(w_1, .., w_T)$ goal is to find the best model $model_\theta$ to predict a label or sequence of label *Y*

$$model_\theta : \quad \mathcal{V}^T \quad \rightarrow \quad \Omega = \mathcal{L}, \mathcal{V'}^{T'}$$

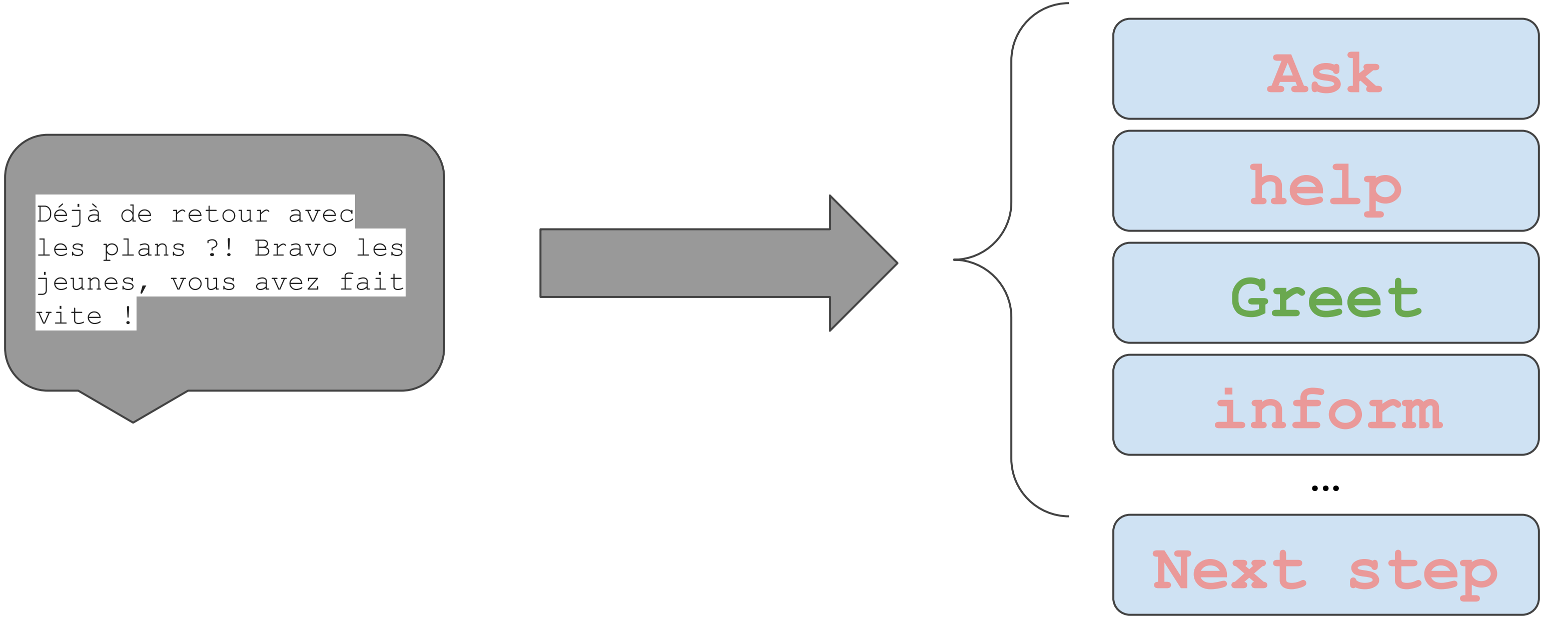
$$(w_1, .., w_T) \mapsto \hat{Y}$$

In this tutorial

We will build a **Dialog Act Classification model for French**

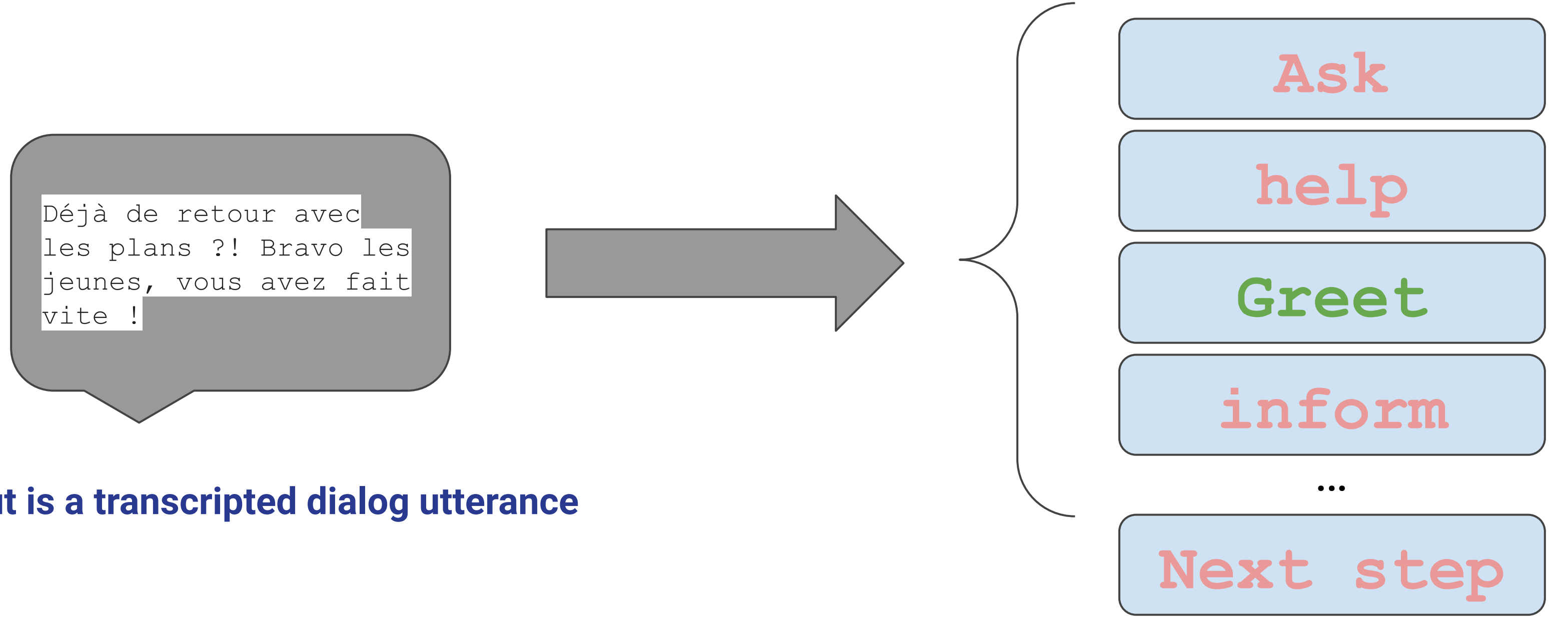
Dialog Act Classification Model

We build a **Dialog Act Classification model for French**



Dialog Act Classification Model

We build a **Dialog Act Classification model for French**

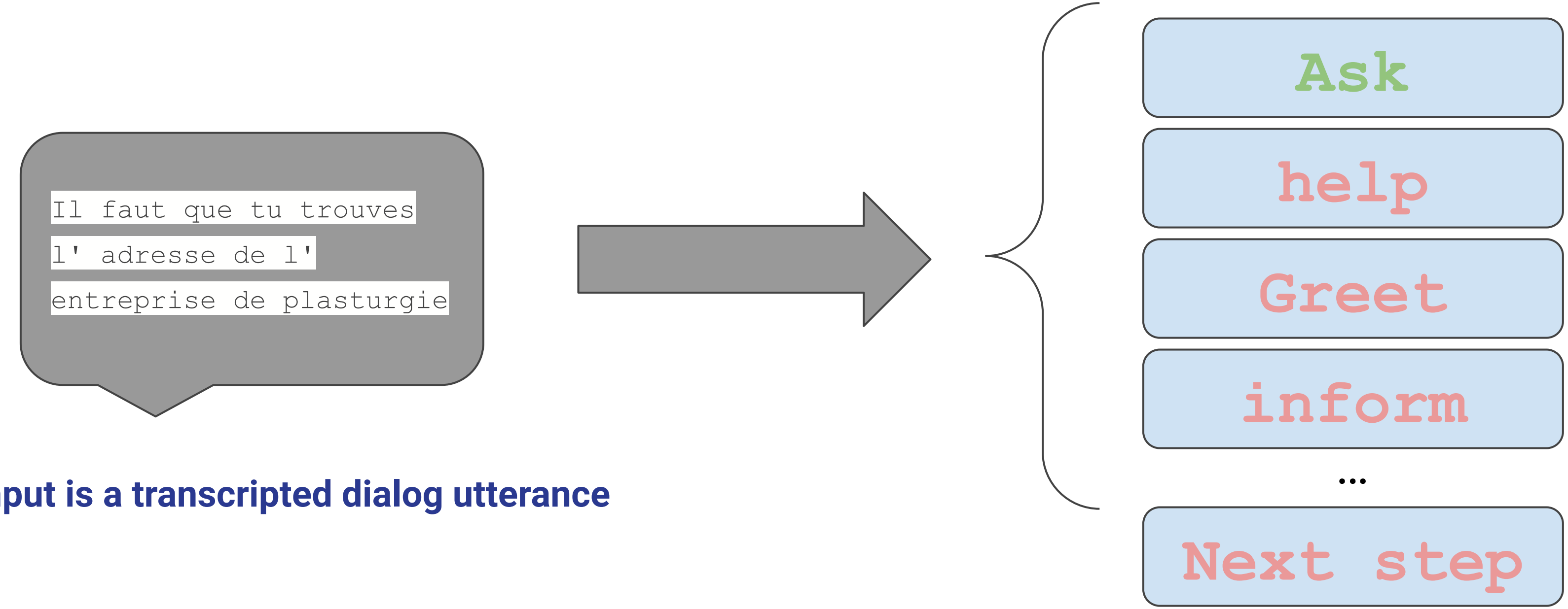


Input is a transcripted dialog utterance

Label is a dialog act [29 possible labels]

Dialog Act Classification Model

We build a **Dialog Act Classification model for French**



Input is a transcripted dialog utterance

Label is a dialog act [29 possible labels]

Building a NLP Model

1. How do we segment the input text ?

2. How do we parametrize the *model*?

3. How do we train the model?

Building a NLP Model: In this Tutorial

1. **How do we segment the input text ?**

➔ We use **sentencepiece** tokenization which is a data-driven sub-word segmentation algorithm

2. How do we parametrize the *model*?

3. How do we train the model?

Tokenization

Tokenization is the first step of everything we do in NLP

It consists in segmenting raw text to define **our modeling units (tokens)**

E.g.: `Il faut que tu trouves l' adresse de l' entreprise de plasturgie`

Tokenization

Tokenization is the first step of everything we do in NLP
It consists in segmenting raw text to define **our modeling units (tokens)**

E.g.: *`Il faut que tu trouves l' adresse de l' entreprise de plasturgie`*

★ Word level segmentation

➔ `['Il', 'faut', 'que', 'tu', 'trouves', "l'", 'adresse', 'de', "l'", 'entreprise', 'de', 'plasturgie']`

Tokenization

Tokenization is the first step of everything we do in NLP
It consists in segmenting raw text to define **our modeling units (tokens)**

E.g.: `Il faut que tu trouves l' adresse de l' entreprise de plasturgie`

★ Word level segmentation

➔ `['Il', 'faut', 'que', 'tu', 'trouves', "l'", 'adresse', 'de', "l'", 'entreprise', 'de', 'plasturgie']`

What happens for an unknown word at test time?

Tokenization

Tokenization is the first step of everything we do in NLP

It consists in segmenting raw text to define **our modeling units (tokens)**

E.g.: `Il faut que tu trouves l' adresse de l' entreprise de plasturgie`

★ Word level segmentation → **Limit: Out-of-Vocabulary Problem**

➔ `['Il', 'faut', 'que', 'tu', 'trouves', "l'", 'adresse', 'de', "l'", 'entreprise', 'de', UNK]`

Tokenization

Tokenization is the first step of everything we do in NLP

It consists in segmenting raw text to define **our modeling units (tokens)**

E.g.: `Il faut que tu trouves l' adresse de l' entreprise de plasturgie`

★ Word level segmentation → **Limit: Out-of-Vocabulary Problem**

★ Character-Level Segmentation

`['I', 'l', ' ', 'f', 'a', 'u', 't', ' ', 'q', 'u', 'e', ' ', 't', 'u', ' ', 't',....`

Tokenization

Tokenization is the first step of everything we do in NLP

It consists in segmenting raw text to define **our modeling units (tokens)**

E.g.: *Il faut que tu trouves l' adresse de l' entreprise de plasturgie*

- ★ Word level segmentation → **Limit: Out-of-Vocabulary Problem**
- ★ Character-Level Segmentation → **Limit: Too Long Sequences**

`['I', 'l', ' ', 'f', 'a', 'u', 't', ' ', 'q', 'u', 'e', ' ', 't', 'u', ' ', 't',....`

Tokenization

Tokenization is the first step of everything we do in NLP
It consists in segmenting raw text to define **our modeling units (tokens)**

E.g.: `Il faut que tu trouves l' adresse de l' entreprise de plasturgie`

★ Word level segmentation → **Limit: Out-of-Vocabulary Problem**

★ Character-Level Segmentation → **Limit: Too Long Sequences**

★ **Sentencepiece**
Segment at the word-level except for infrequent words that are segmented at the subword level

Tokenization

Tokenization is the first step of everything we do in NLP
It consists in segmenting raw text to define **our modeling units (tokens)**

E.g.: `Il faut que tu trouves l' adresse de l' entreprise de plasturgie`

★ Word level segmentation → **Limit: Out-of-Vocabulary Problem**

★ Character-Level Segmentation → **Limit: Too Long Sequences**

★ **Sentencepiece** → **Tradeoff between both approaches**
Segment at the word-level except for infrequent words that are segmented at the subword level
`['_Il','_faut','_que','_tu','_trouve','s','_l',"'",'_adresse','_de','_l',"'",'_entreprise','_de','_','plast','ur','gie']`

Building a NLP Model: In this Tutorial

1. How do we segment the input text ?

➔ We use sentencepiece tokenization which is a data-driven sub-word segmentation algorithm

2. **How do we parametrize the *model*?**

➔ **We use the transformer architecture**

3. How do we train the model?

Recall

Given a sequence of tokens $(w_1, .., w_T)$ goal is to find the best model $model_\theta$ to predict a label or sequence of label *Y*

$$model_\theta : \qquad \mathcal{V}^T \qquad \rightarrow \quad \Omega = \mathcal{L}, \ \mathcal{V}'^{T'}$$

$$(w_1, .., w_T) \mapsto \hat{Y}$$

We want to build a **Dialog Act Classification model for French**

How to parametrize the Model?

We parametrize the model with a **Transformer** architecture

- The Transformer is a deep-learning architecture
- It **can be used to parametrize any** sequence labelling, classification and generation **task**
- It is **the most popular and accurate architecture for most NLP tasks**

How to parametrize the Model?

We parametrize the model with a **Transformer** architecture

- The Transformer is a deep-learning architecture
- It **can be used to parametrize any** sequence labelling, classification and generation **task**
- It is **the most popular and accurate architecture for most NLP tasks**

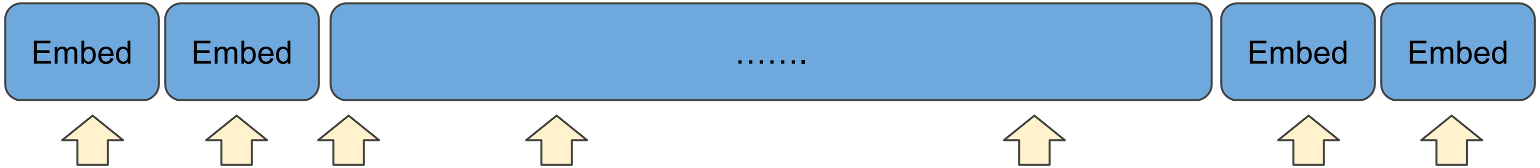
In a nutshell

- Transformers are *trainable functions* made of **the stack of multiple linear and non-linear functions**
- All the parameters are trained **to minimize a loss function for a given task**
- In this tutorial, we use **a Transformer (Camembert) to perform dialog act prediction**

Sequence Classification with a Transformer

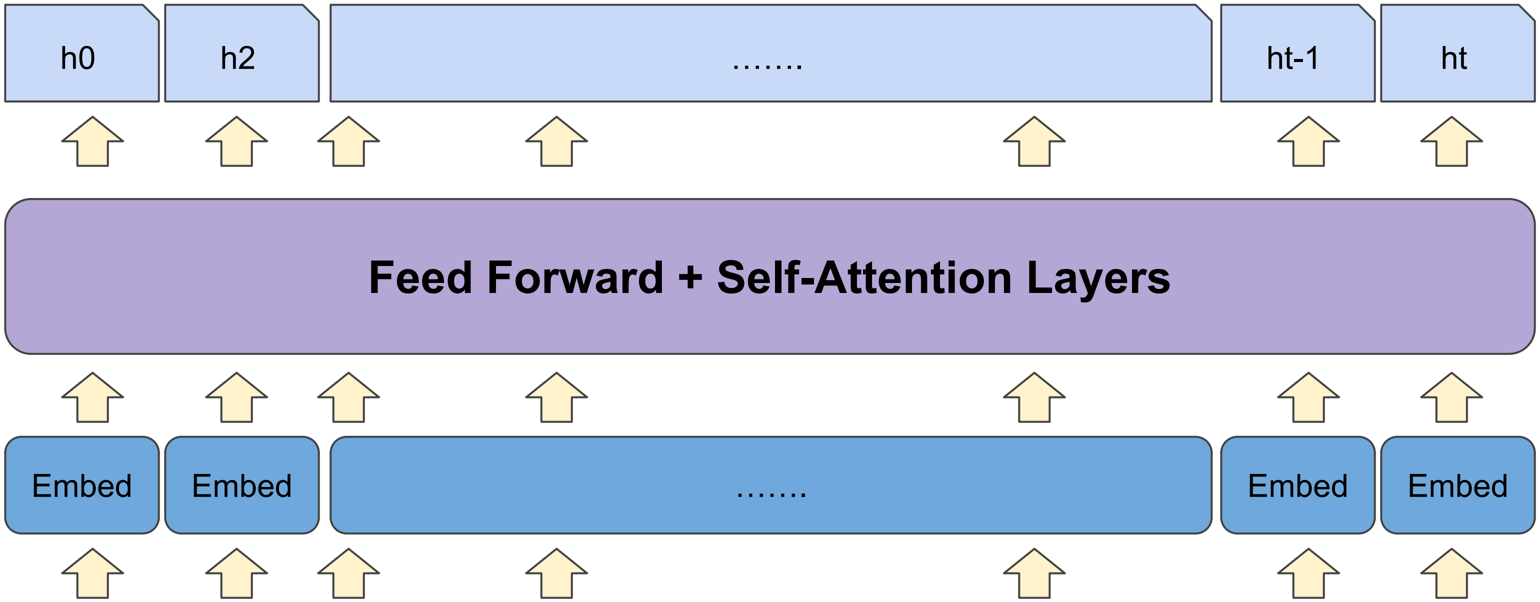
```
[<s> '_Il',,"'",'_entreprise','_de','_','plast','ur','gie', </s>]
```

Sequence Classification with a Transformer



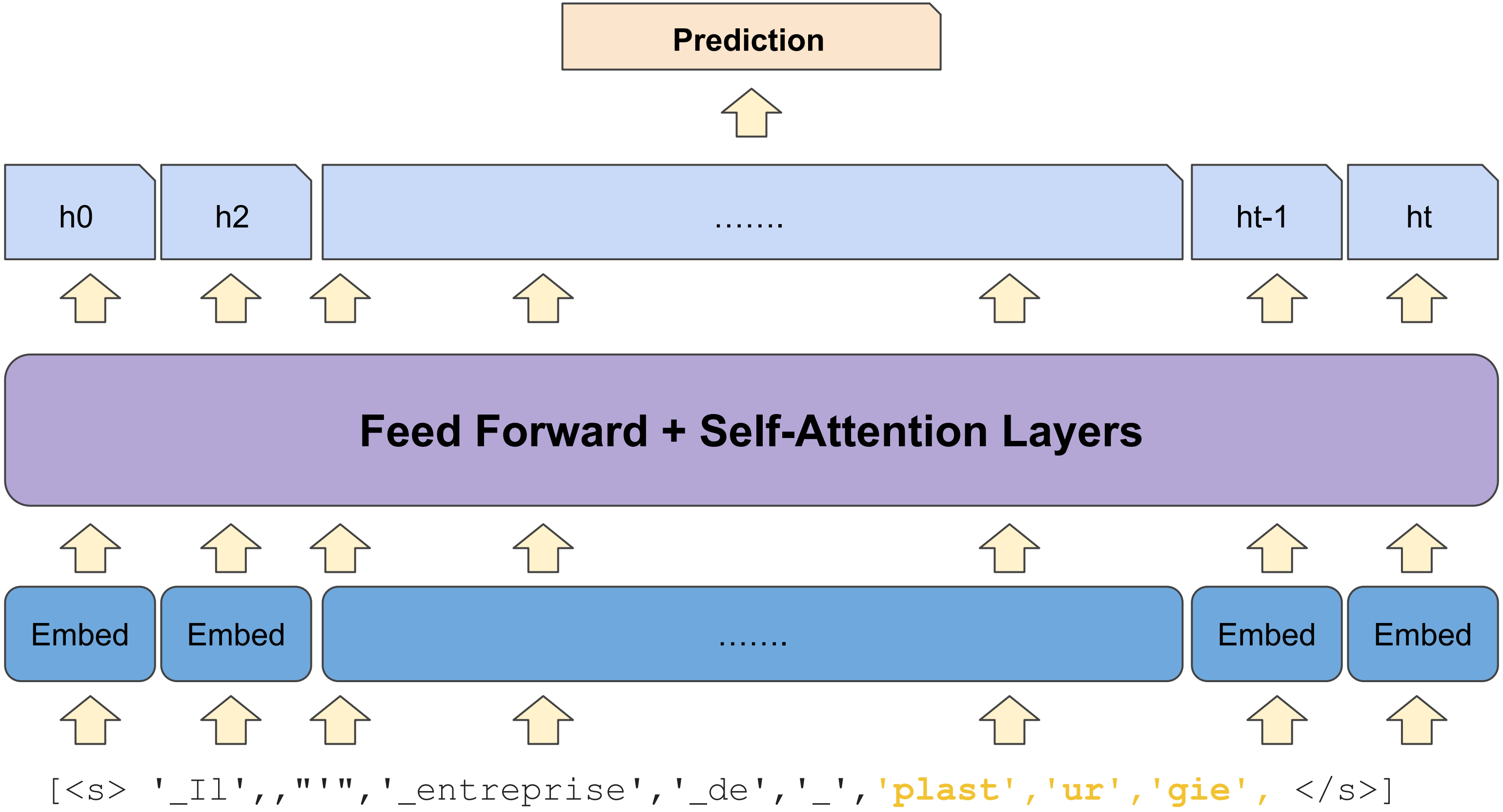
`[<s> '_Il',,"'",'_entreprise','_de','_','plast','ur','gie', </s>]`

Sequence Classification with a Transformer

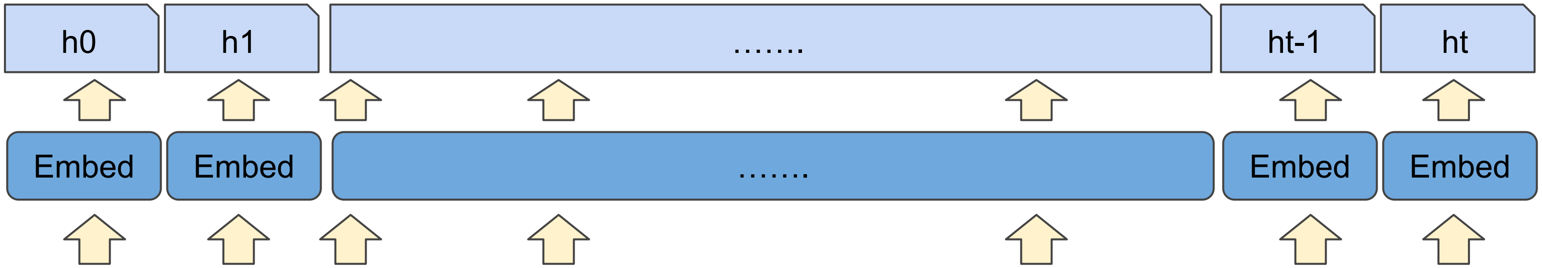


[<s> '_Il',,"'",'_entreprise','_de','_','plast','ur','gie', </s>]

Sequence Classification with a Transformer

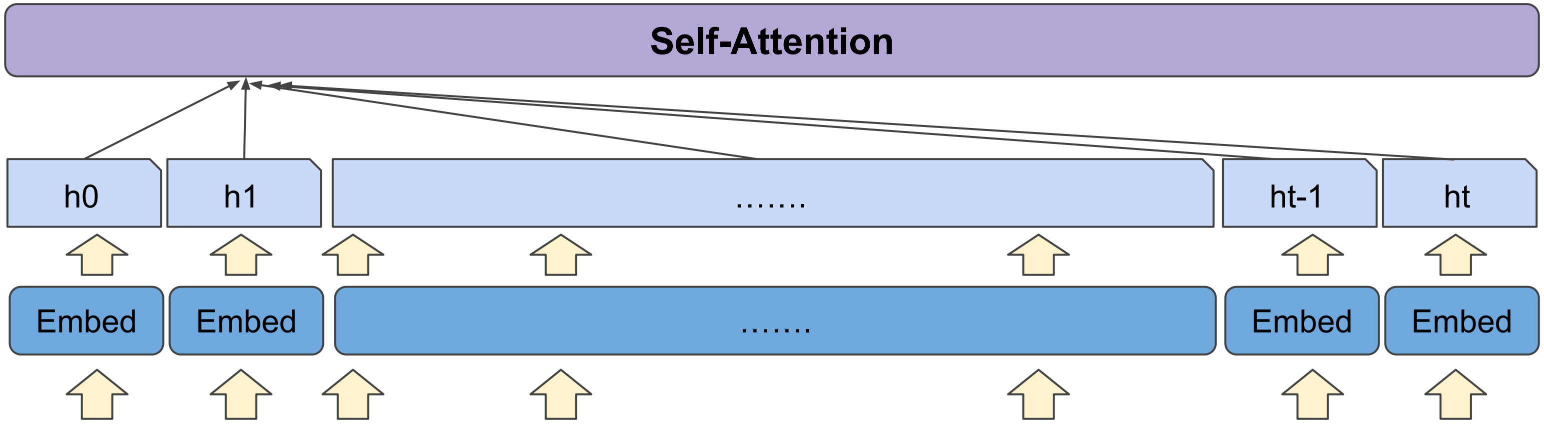


Self-Attention Layer



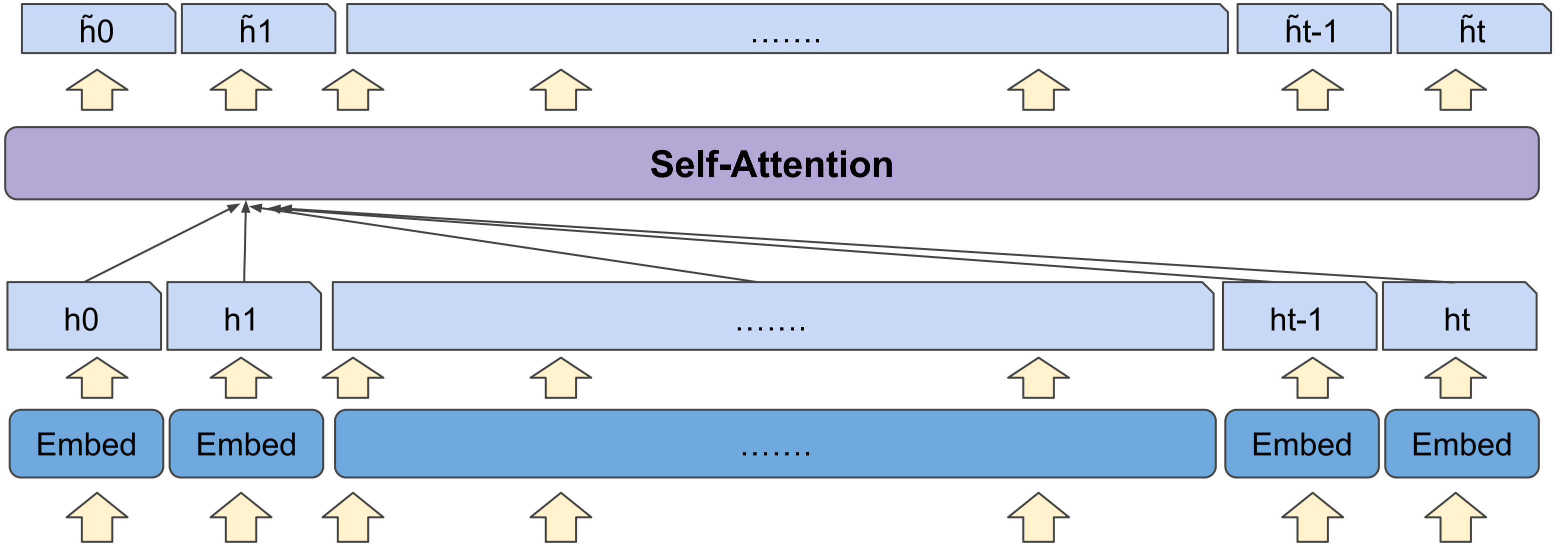
[<s> '_Il',,"'",'_entreprise','_de','_','plast','ur','gie', </s>]

Self-Attention Layer



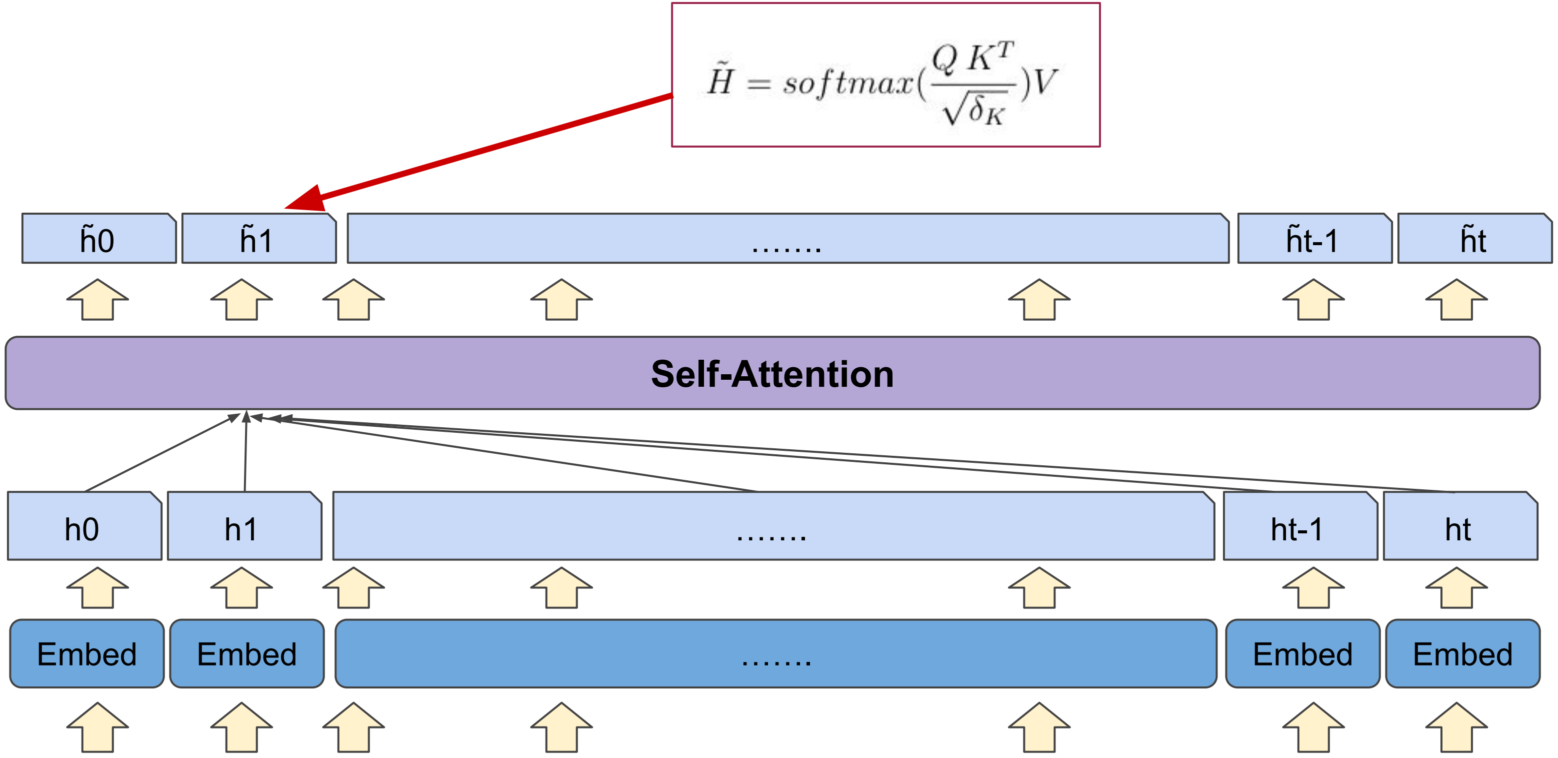
[<s> '_Il',,"'",'_entreprise','_de','_','plast','ur','gie', </s>]

Self-Attention Layer



[<s> '_Il',,"'",'_entreprise','_de','_','plast','ur','gie', </s>]

Self-Attention Layer



[<s> '_Il',,"'",'_entreprise','_de','_','plast','ur','gie', </s>]

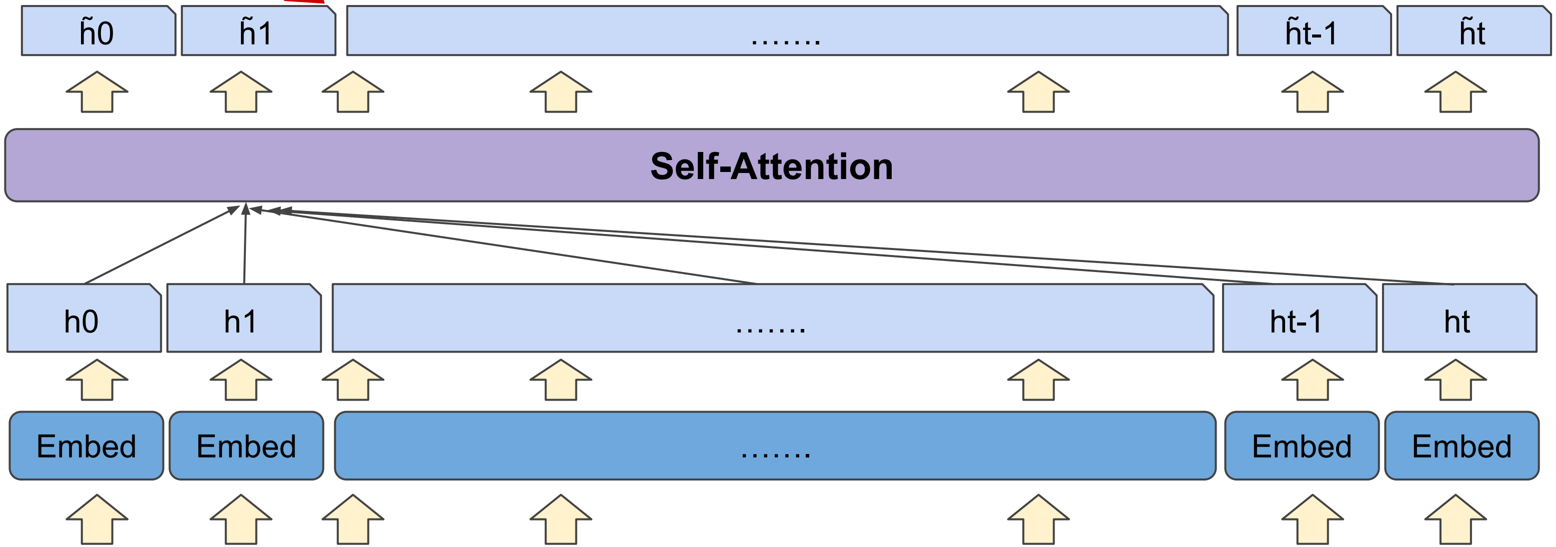
Self-Attention Layer

$$q_t = W_Q h_t \ , \forall\, t \in [|1, T|] \text{ with } W_Q \in \mathrm{R}^{\delta_q \times \delta}$$

$$k_t = W_K h_t \ , \forall\, t \in [|1, T|] \text{ with } W_K \in \mathrm{R}^{\delta_k \times \delta}$$

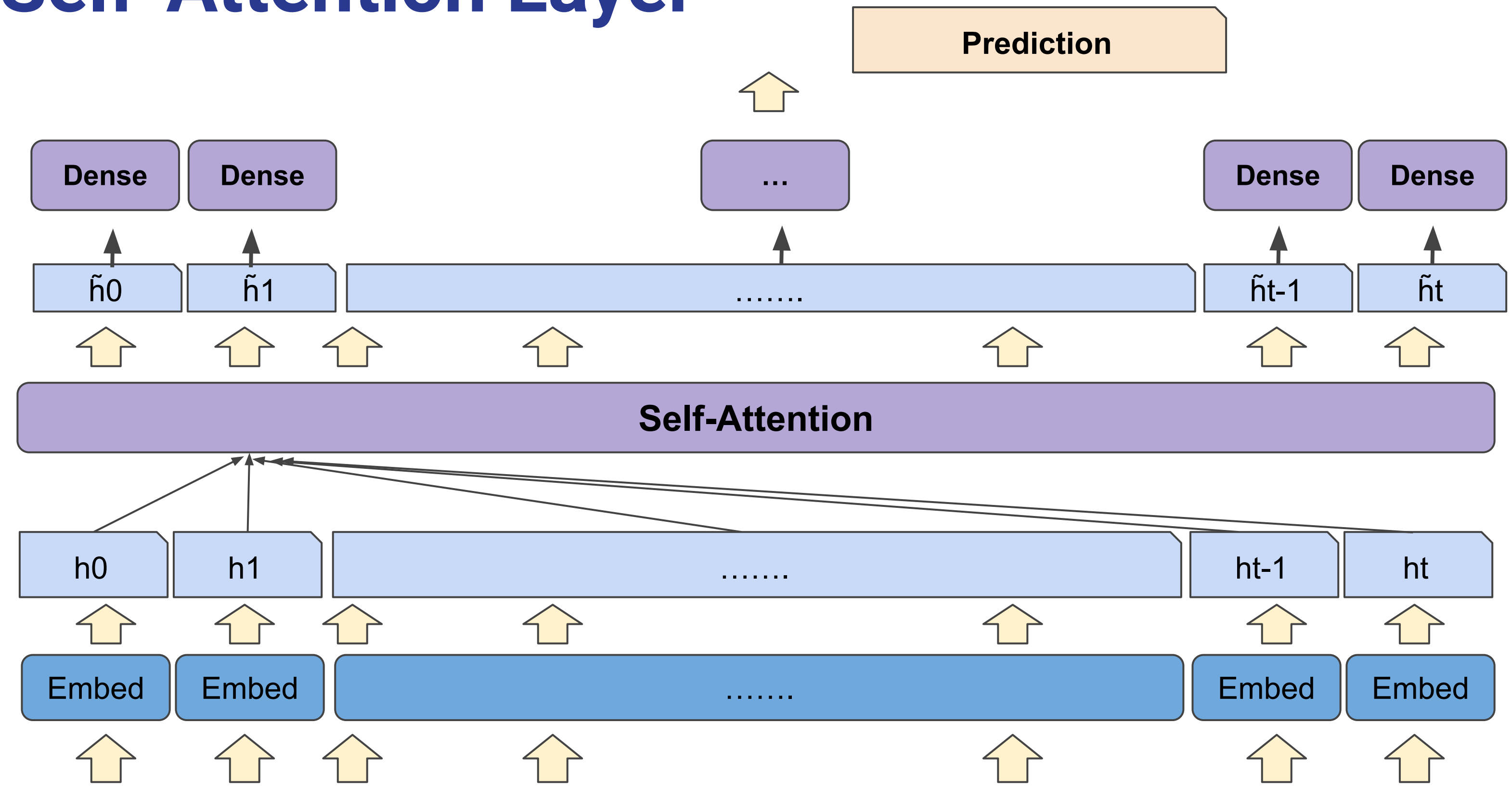
$$v_t = W_V h_t \ , \forall\, t \in [|1, T|] \text{ with } W_V \in \mathrm{R}^{\delta_v \times \delta}$$

$$\tilde{H} = softmax\left(\frac{Q\,K^T}{\sqrt{\delta_K}}\right)V$$



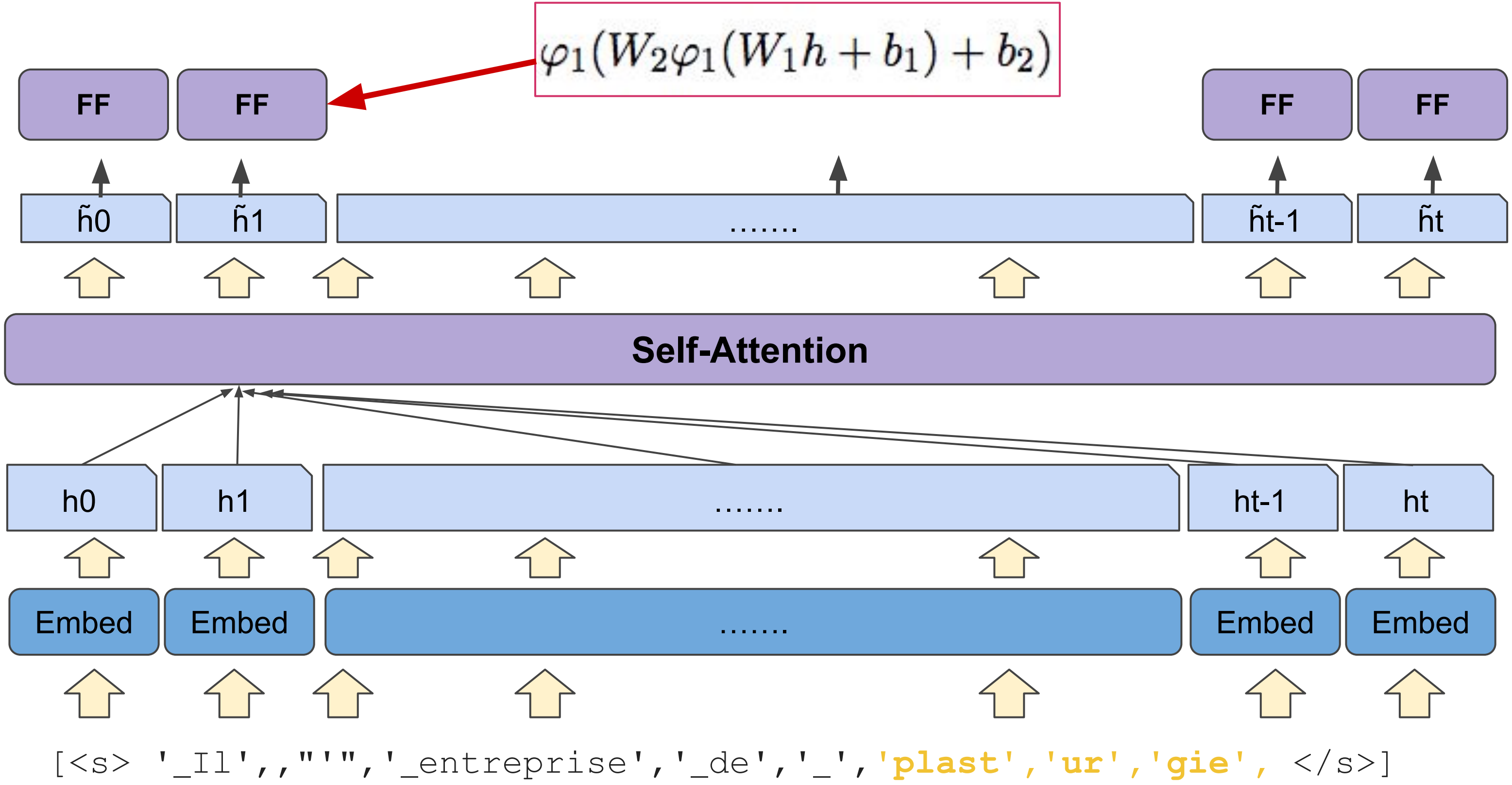

```
[<s> '_Il',,"'",'_entreprise','_de','_','plast','ur','gie', </s>]
```

Self-Attention Layer

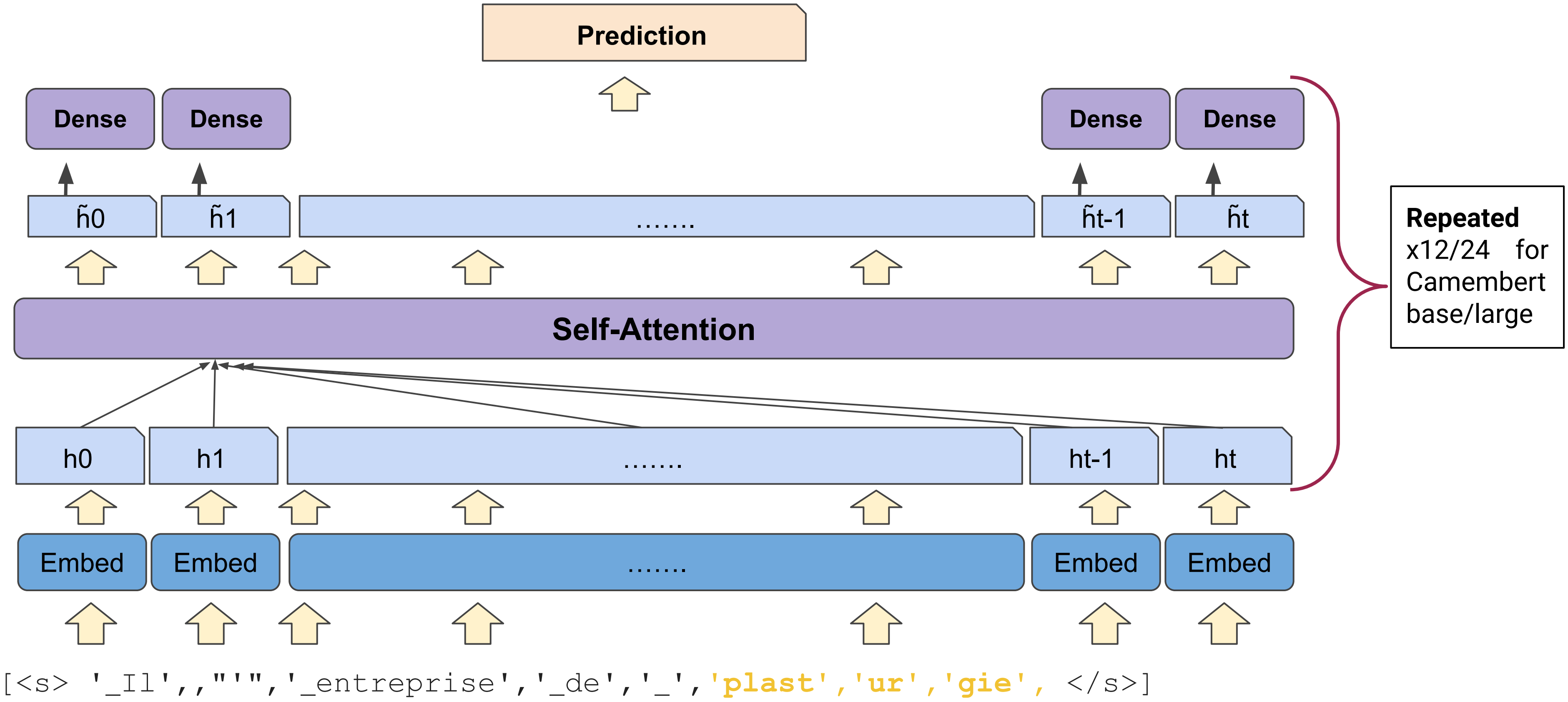


[<s> '_Il',,"'",'_entreprise','_de','_','plast','ur','gie', </s>]

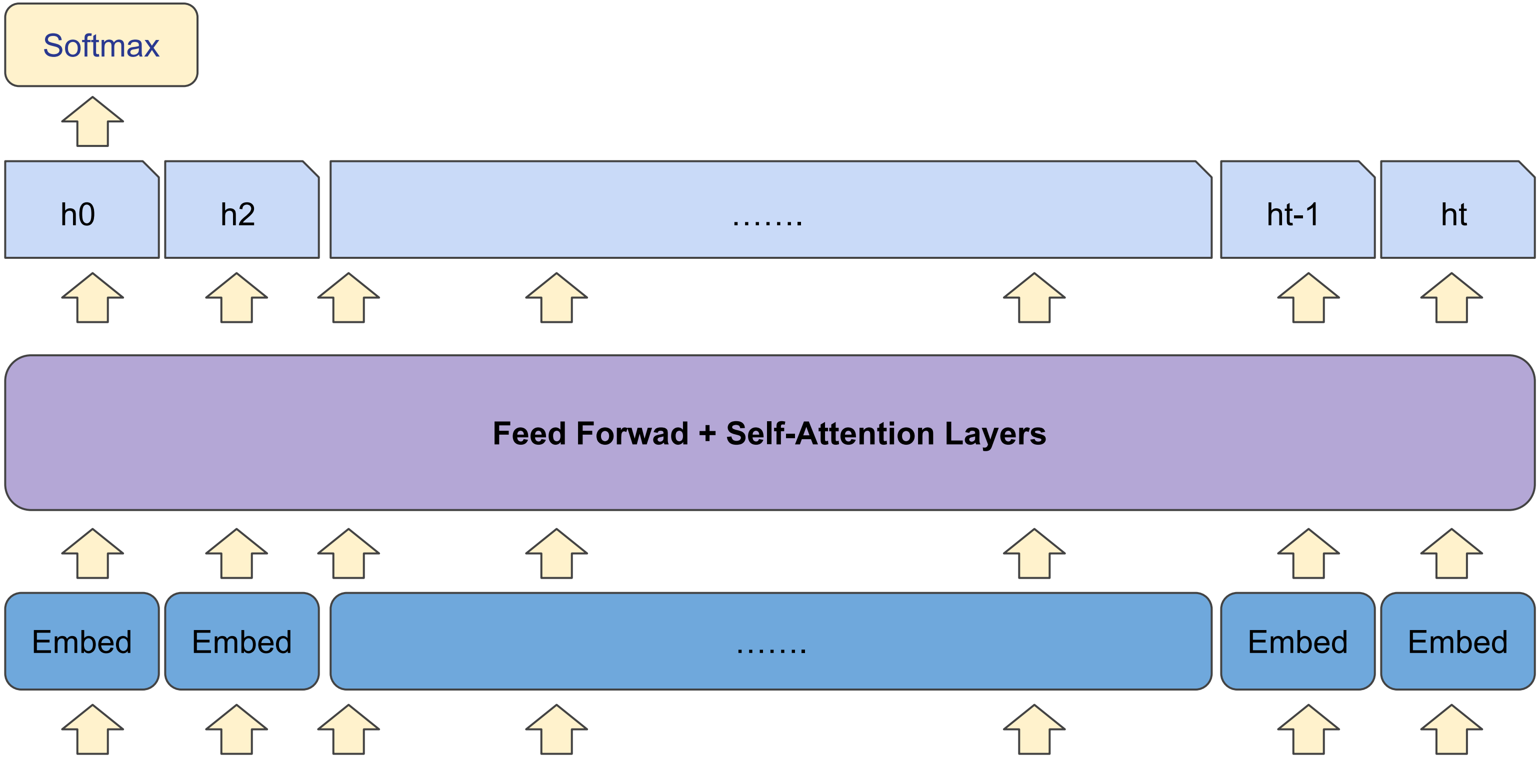
Dense Layer


$\varphi_1(W_2\varphi_1(W_1 h + b_1) + b_2)$
FF
FF
FF
FF
h̃0
h̃1
…….
h̃t-1
h̃t
Self-Attention
h0
h1
…….
ht-1
ht
Embed
Embed
…….
Embed
Embed
[<s> '_Il',,"'",'_entreprise','_de','_','plast','ur','gie', </s>]

Transformer



Sequence Classification



[<s> '_l',,"'",'_entreprise','_de','_','plast','ur','gie', </s>]

Sequence Classification: Output Function

We compute a distribution over the labels using the softmax function.
We have K=31 **speech acts labels**

$$softmax(s) = \left(\frac{e^{s_i}}{\sum_k e^{s_k}}\right)_{i \in [|1,K|]}, \text{ for } s \in \mathbb{R}^K$$

*s is also referred to as the **logit** vector*

Sequence Classification: Output Function

We compute a distribution over the labels using the softmax function.
We have *K*=31 **speech acts labels**

$$softmax(s) = \left(\frac{e^{s_i}}{\sum_k e^{s_k}}\right)_{i \in [|1,K|]}, \text{ for } s \in \mathbb{R}^K$$

s is also referred to as the ***logit*** *vector*

➔ At test time, we make predictions by picking the label that has the **maximum output likelihood** (argmax)

Building a NLP Model

1. How do we segment the input text ?

➔ We use sentencepiece tokenization which is a data-driven sub-word segmentation algorithm

2. How do we parametrize the *model*?

➔ We use the transformer architecture

3. **How do we train the model?**

➔ **Training the transformers on sequence classification**

Training for Sequence Classification

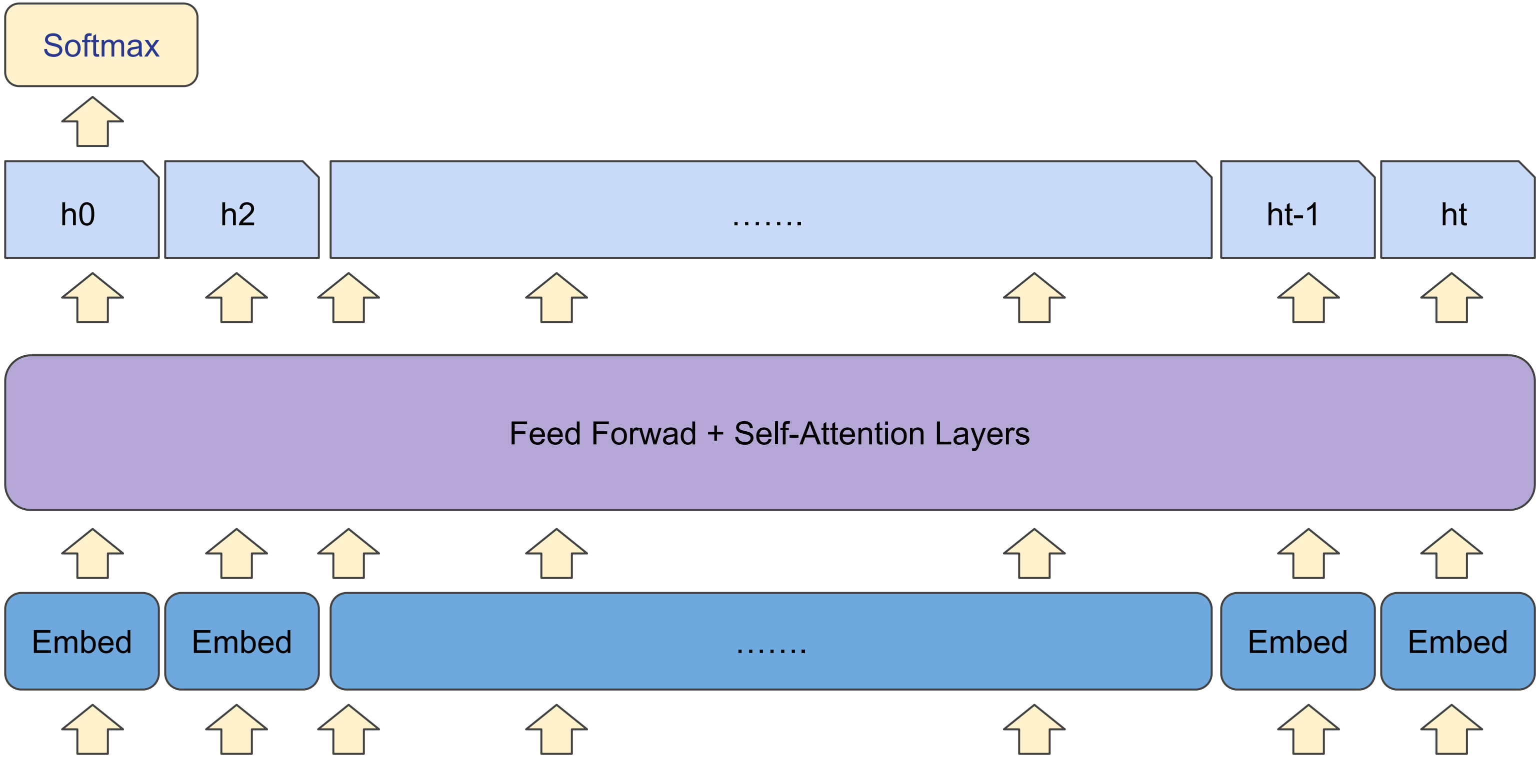
As most deep learning models, we train our model by:

1. **Initializing** all the parameters of the model

2. Defining **a loss function** **to compare the prediction of the model with observed labels**

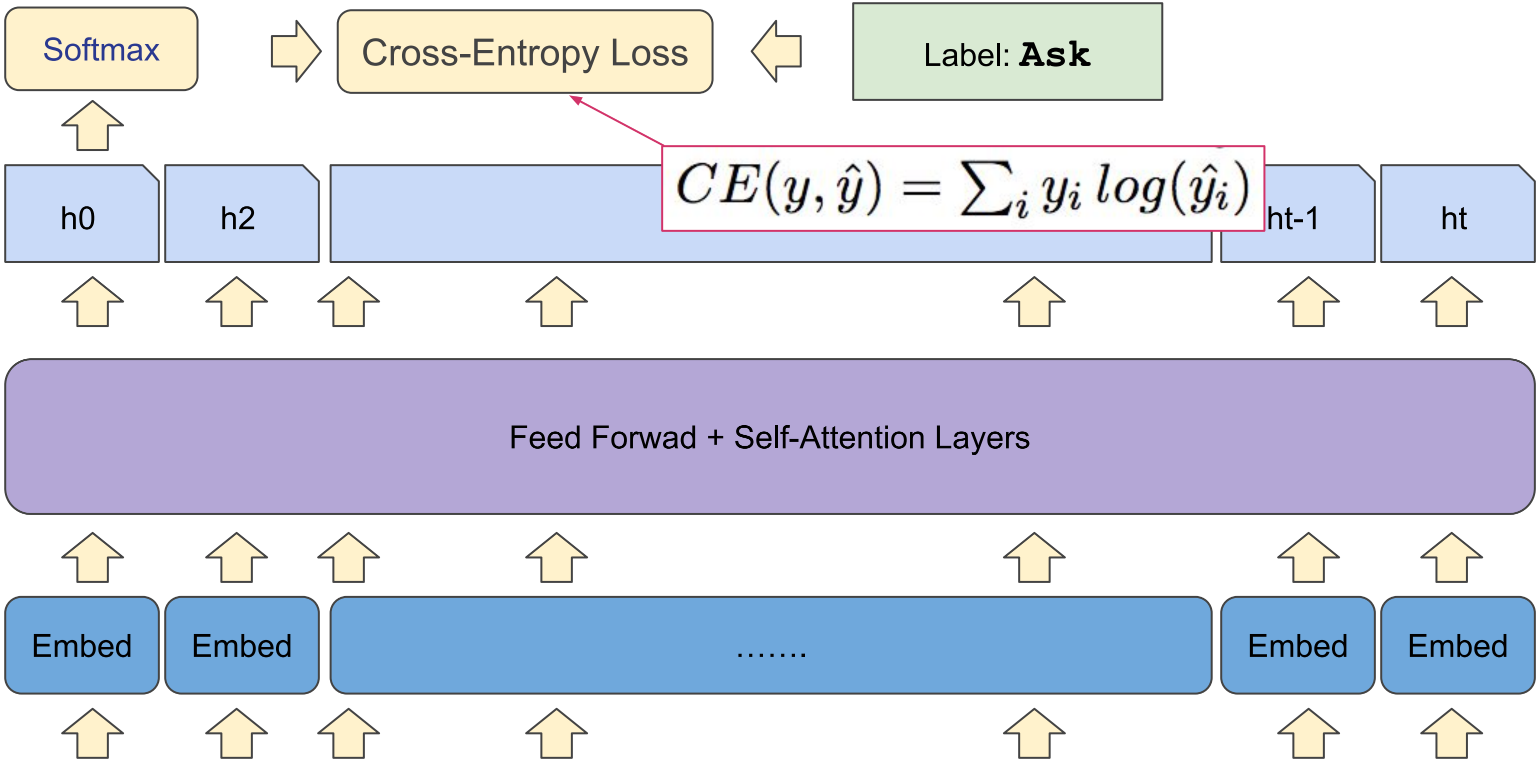
3. Find the parameters of the model that minimize **the loss** **with Gradient Descent**
   NB: all the parameters are trained end-to-end

Sequence Classification



[<s> '_l',,"'",'_entreprise','_de','_','plast','ur','gie', </s>]

Sequence Classification



`[<s> '_l',,"'",'_entreprise','_de','_','plast','ur','gie', </s>]`

Training with Stochastic Gradient Descent

Algorithm 2 Stochastic Gradient Descend

Given observations $((x_i), (y_i))$ of two variables (X, Y)
Given a loss function l. An architecture dnn_θ
The goal is to find the best θ **s.t.** $E(l(Y, dnn_\theta(X)))$ **is small.** Given a learning rate α
for *step* $<$ max **do**
  Sample (x, y)
  # Forward pass:
  $\hat{y} = dnn_\theta(x)$ and $l(y, \hat{y})$
  # Backward pass:
  $\nabla_\theta\, l(y, \hat{y})$ # compute gradients
  $\theta := \theta - \alpha \nabla_\theta\, l(y, \hat{y})$ # parameter update
end

Training with Stochastic Gradient Descent

Algorithm 2 Stochastic Gradient Descend

Given observations $((x_i), (y_i))$ of two variables (X, Y)
Given a loss function l. An architecture dnn_θ
The goal is to find the best θ **s.t.** $E(l(Y, dnn_\theta(X))$ **is small**. Given a learning rate α
for $step < max$ **do**
  Sample (x, y)
  # Forward pass:
  $\hat{y} = dnn_\theta(x)$ and $l(y, \hat{y})$
  # Backward pass:
  $\nabla_\theta\, l(y, \hat{y})$ # compute gradients
  $\theta := \theta - \alpha \nabla_\theta\, l(y, \hat{y})$ # parameter update
end

Stochastic Gradient Descent

Algorithm 2 Stochastic Gradient Descend

Given observations $((x_i), (y_i))$ of two variables (X, Y)
Given a loss function l. An architecture dnn_θ
The goal is to find the best θ **s.t.** $E(l(Y, dnn_\theta(X)))$ **is small.** Given a learning rate α
for *step* $<$ max **do**
  Sample (x, y)
  # Forward pass:
  $\hat{y} = dnn_\theta(x)$ and $l(y, \hat{y})$
  # Backward pass:
  $\nabla_\theta\, l(y, \hat{y})$ # compute gradients
  $\theta := \theta - \alpha \nabla_\theta\, l(y, \hat{y})$ # parameter update
end

Training with Stochastic Gradient Descent

Algorithm 2 Stochastic Gradient Descend

Given observations $((x_i), (y_i))$ of two variables (X, Y)
Given a loss function l. An architecture dnn_θ
The goal is to find the best θ **s.t.** $E(l(Y, dnn_\theta(X)))$ **is small**. Given a learning rate α
for *step* $< \ max$ **do**
  Sample (x, y)
  # Forward pass:
  $\hat{y} = dnn_\theta(x)$ and $l(y, \hat{y})$
  # Backward pass:
  $\nabla_\theta \, l(y, \hat{y})$ # compute loss
  $\theta := \theta - \alpha \nabla_\theta \, l(y, \hat{y})$ # parameter update
end

Training with Stochastic Gradient Descent

Algorithm 2 Stochastic Gradient Descend

Given observations $((x_i), (y_i))$ of two variables (X, Y)
Given a loss function l. An architecture dnn_θ
The goal is to find the best θ **s.t.** $E(l(Y, dnn_\theta(X)))$ **is small**. Given a learning rate α
for *step* $<$ max **do**
  Sample (x, y)
  # Forward pass:
  $\hat{y} = dnn_\theta(x)$ and $l(y, \hat{y})$
  # Backward pass:
  $\nabla_\theta\, l(y, \hat{y})$ # compute loss
  $\theta := \theta - \alpha \nabla_\theta\, l(y, \hat{y})$ # parameter update
end

Training step

Learning rate

In this tutorial, you will use **Adam a variant of SGD** to train our model.

How to initialize our Transformer?

Randomly initialize all the parameters and train all the parameters from scratch for Speech Act Classification

Limits: Training Transformers this way is suboptimal

How to initialize our Transformer?

Randomly initialize all the parameters and train all the parameters from scratch for Speech Act Classification

Limits: Training Transformers this way is suboptimal

➔ **Likely to overfit**

➔ **Poor Generalization**

How to initialize our Transformer?

Randomly initialize all the parameters and train all the parameters from scratch for Speech Act Classification

Limits: Training Transformers this way is suboptimal

Solution: **Pretrain** the transformer on **Masked-Language Modeling**

Finally: Camembert

Camembert is a large transformer model **(300M+ parameters)**

Pretrained with

- **a Masked-Language Modeling Objective**
- On **138 GB of Web-Crawled Data (32B tokens)** in French (OSCAR)

Finally: Camembert

Camembert is a large transformer model **(300M+ parameters)**

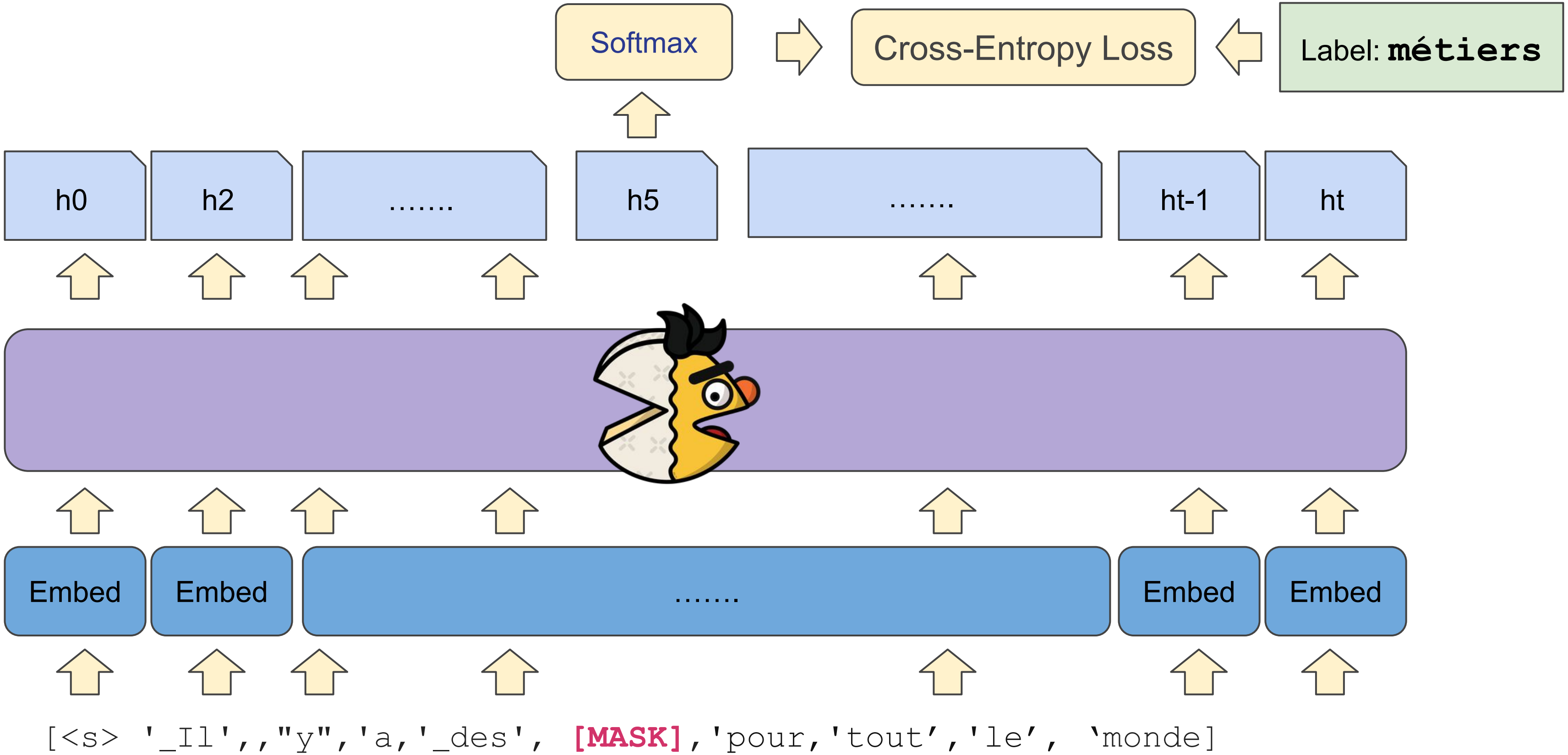
Pretrained with

- **a Masked-Language Modeling Objective**
- On **138 GB of Web-Crawled Data (32B tokens)** in French (OSCAR)

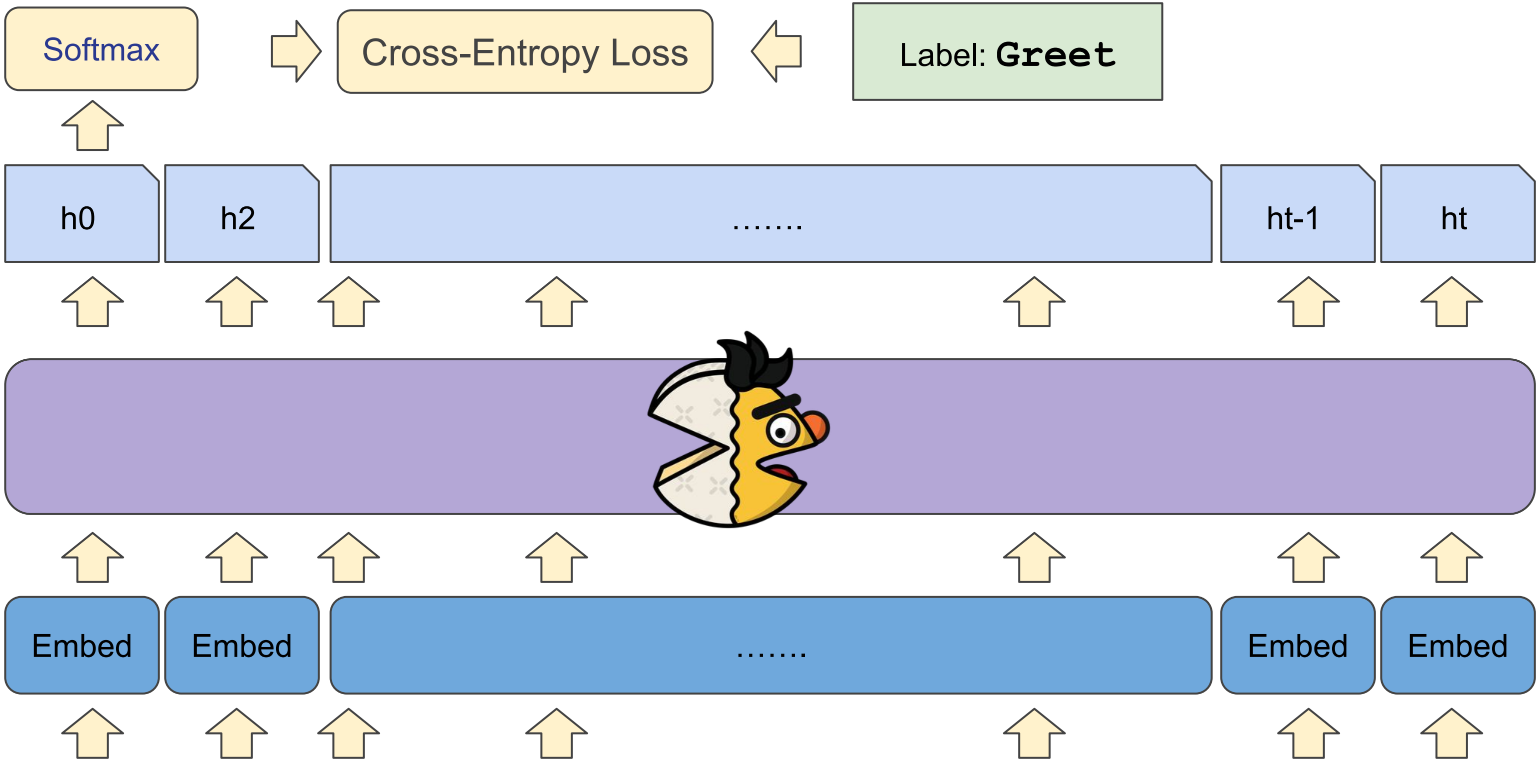
Camembert can be used as the initialization of any transformer architecture to build a model for French

In other words, **Camembert can be fine-tuned** (i.e. trained further) on any sequence labeling and sequence classification task

Pretraining of Camembert with MLM



Fine-tuning of Camembert



[<s> '_Il',,"'",'_entreprise','_de','_','plast','ur','gie', </s>]

Camembert

Camembert is a large transformer model **(300M+ parameters)**

Pretrained with

- **a Masked-Language Modeling Objective**
- On **138 GB of Web-Crawled Data (32B tokens)** in French (OSCAR)

Camembert can be fine-tuned (i.e. trained further) on any sequence labeling and sequence classification task

Camembert **delivers state-of-the-art performance** on multiple NLP benchmark for French

Camembert Performance on standard tasks

After pretraining, we can reuse the entire camembert model and **fine-tune** it on our task

Model	F1
SEM (CRF) (Dupont, 2017)	85.02
LSTM-CRF (Dupont, 2017)	85.57
mBERT (fine-tuned)	87.35
CamemBERT (fine-tuned)	89.08
LSTM+CRF+CamemBERT (embeddings)	**89.55**

Named-Entity Recognition

	FQuAD1.1-test		FQuAD1.1-dev	
Model	F1	EM	F1	EM
Human Perf.	91.2	75.9	92.1	78.3
CamemBERT$_{\text{BASE}}$	88.4	78.4	88.1	78.1
CamemBERT$_{\text{LARGE}}$	**92.2**	**82.1**	**91.8**	**82.4**
FlauBERT$_{\text{BASE}}$	77.6	66.5	76.3	65.5
FlauBERT$_{\text{LARGE}}$	80.5	69.0	79.7	69.3
mBERT	86.0	75.4	86.2	75.5
XLM-R$_{\text{BASE}}$	85.9	75.3	85.5	74.9
XLM-R$_{\text{LARGE}}$	89.5	79.0	89.1	78.9

Question-Answering

In Summary

- **Camembert is a large transformers model**
- Pretrained with the Masked-Language Objective
- **That can be re-used for any sequence labelling or sequence prediction tasks**
- **And likely to deliver good downstream performance**

In Summary

- **Camembert is a large transformers model**
- Pretrained with the Masked-Language Objective
- **That can be re-used for any sequence labelling or sequence prediction tasks**
- **And likely to deliver good downstream performance**

In the next sessions of the tutorial

➔ you will use the **transformers library** 🤗 **to download** **and play with it!**
➔ We showcase camembert for dialog act classification but the same approach can be used for any sequence classification or labeling task