Automatic Speech Recognition

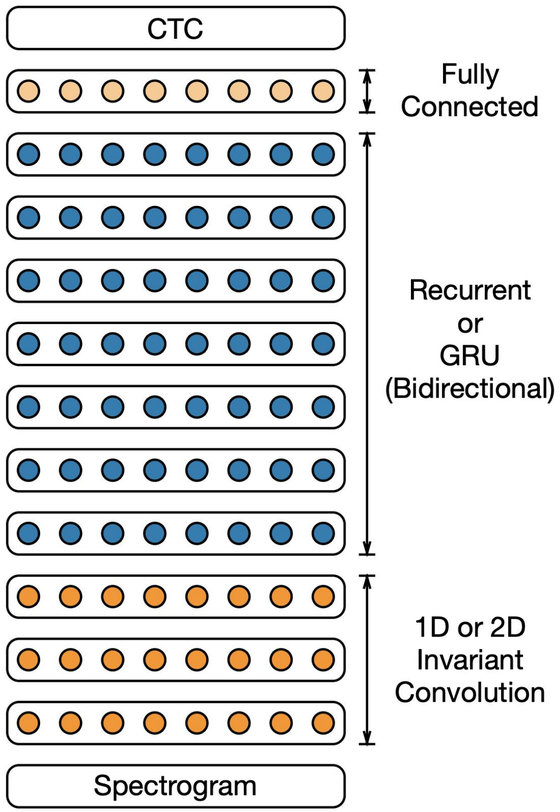
лекция 2

Апарин Георгий

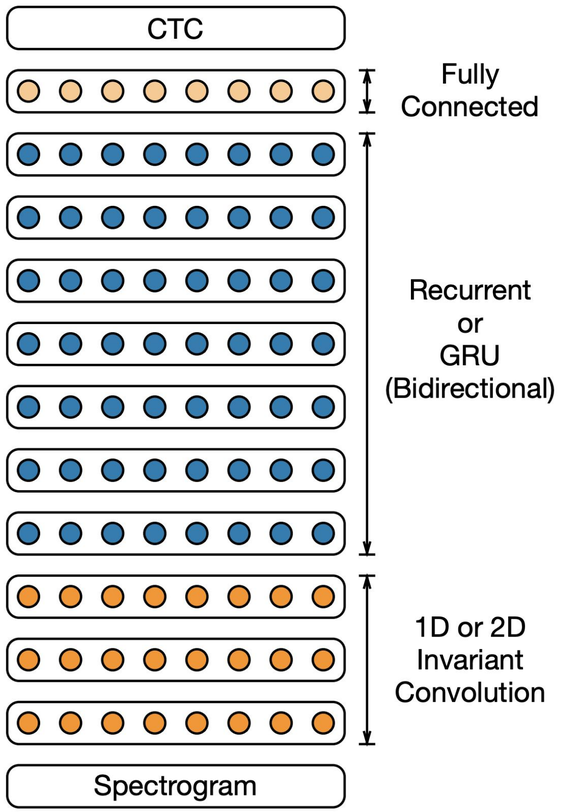
- Что такое доменные данные
- Какие бывают методы адаптации
- Shallow fusion
- Модели адаптации

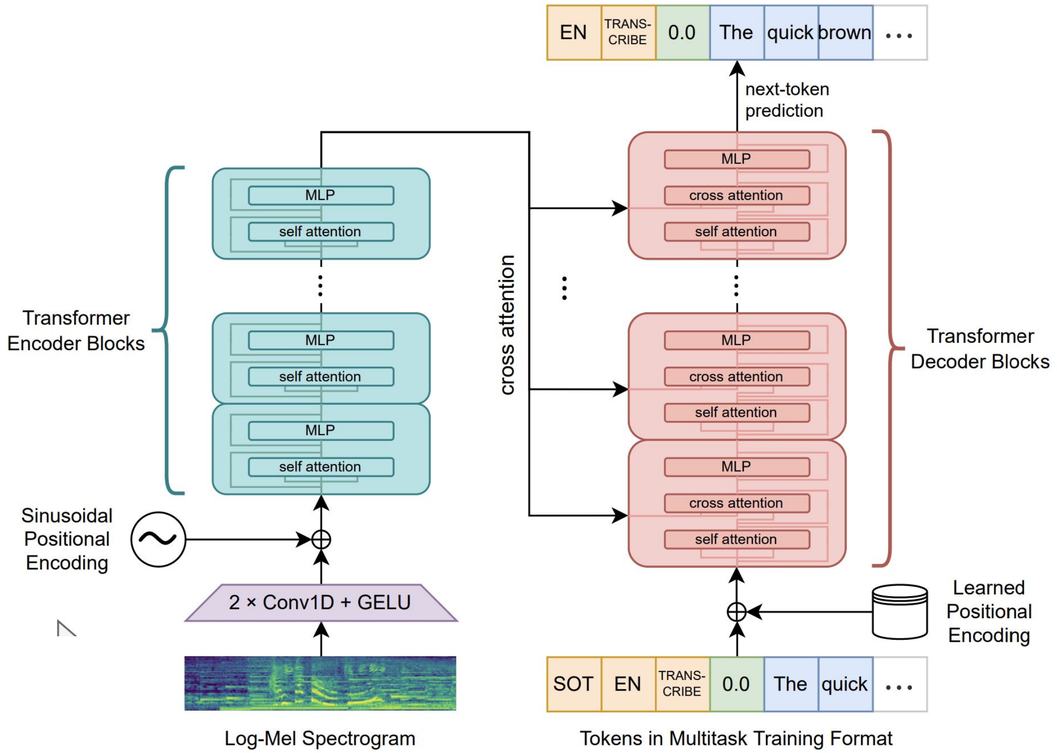
Доменные данные

Вспомним архитектуры

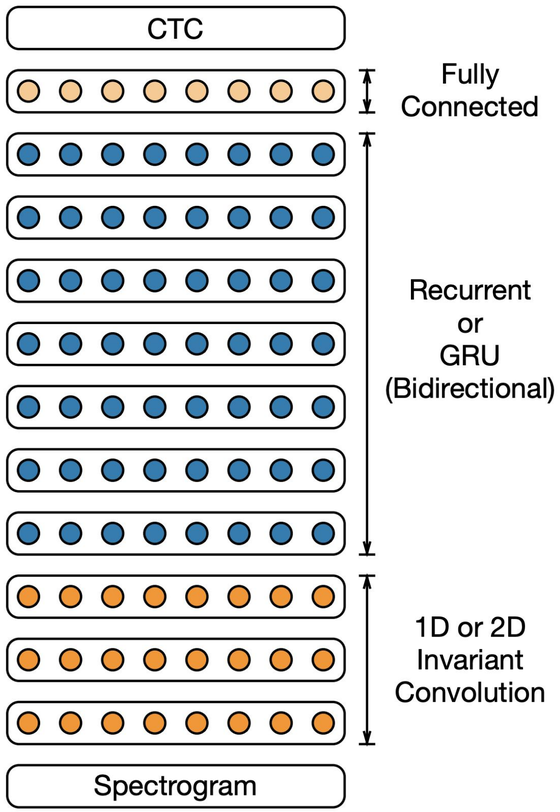


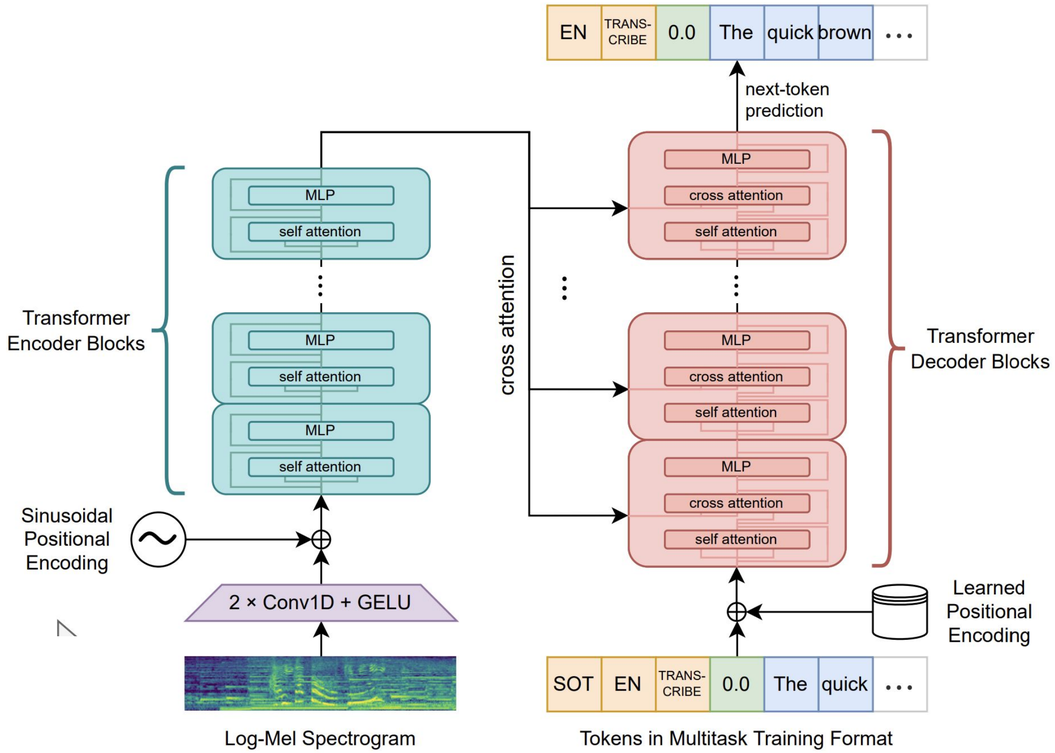
Вспомним архитектуры

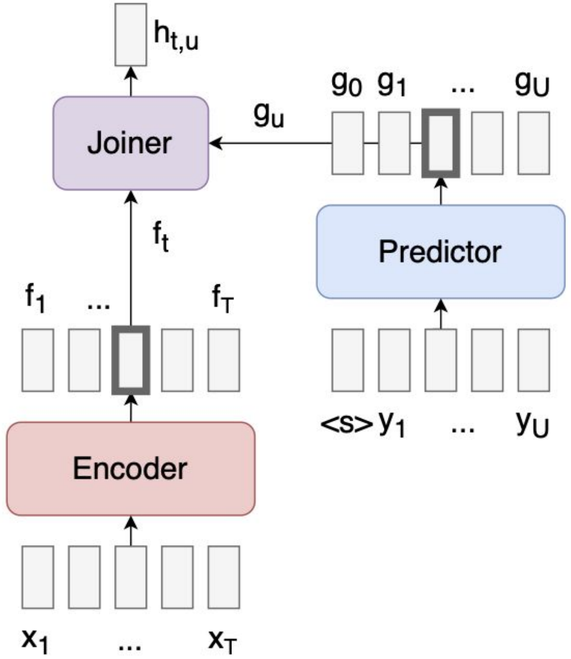




Вспомним архитектуры

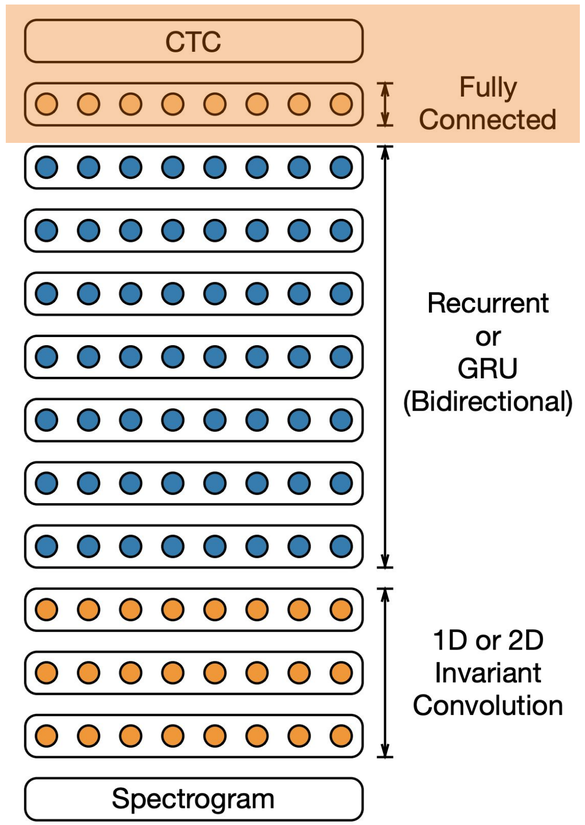


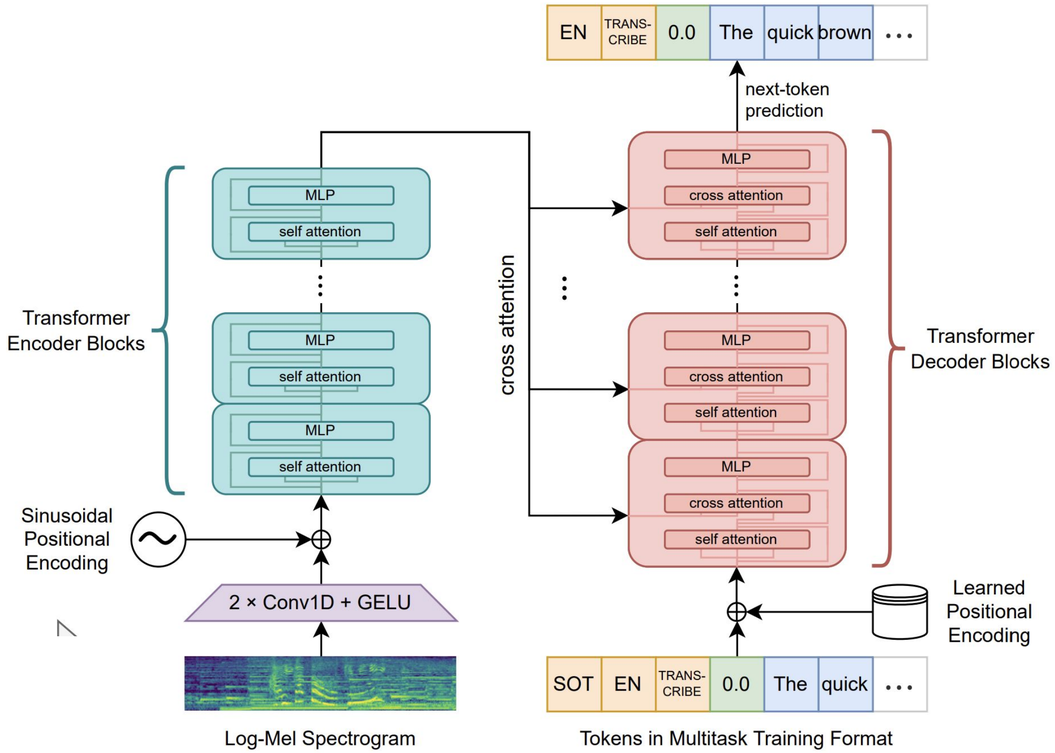


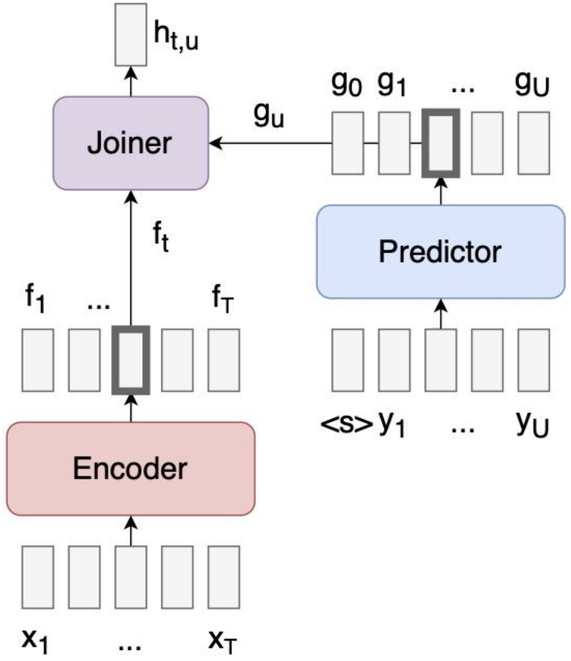


Вспомним архитектуры

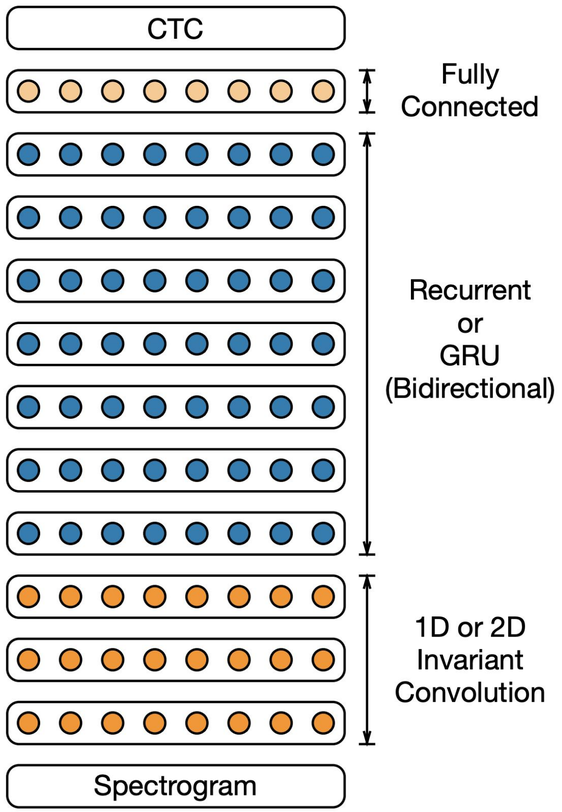
не авторегрессионное предсказание



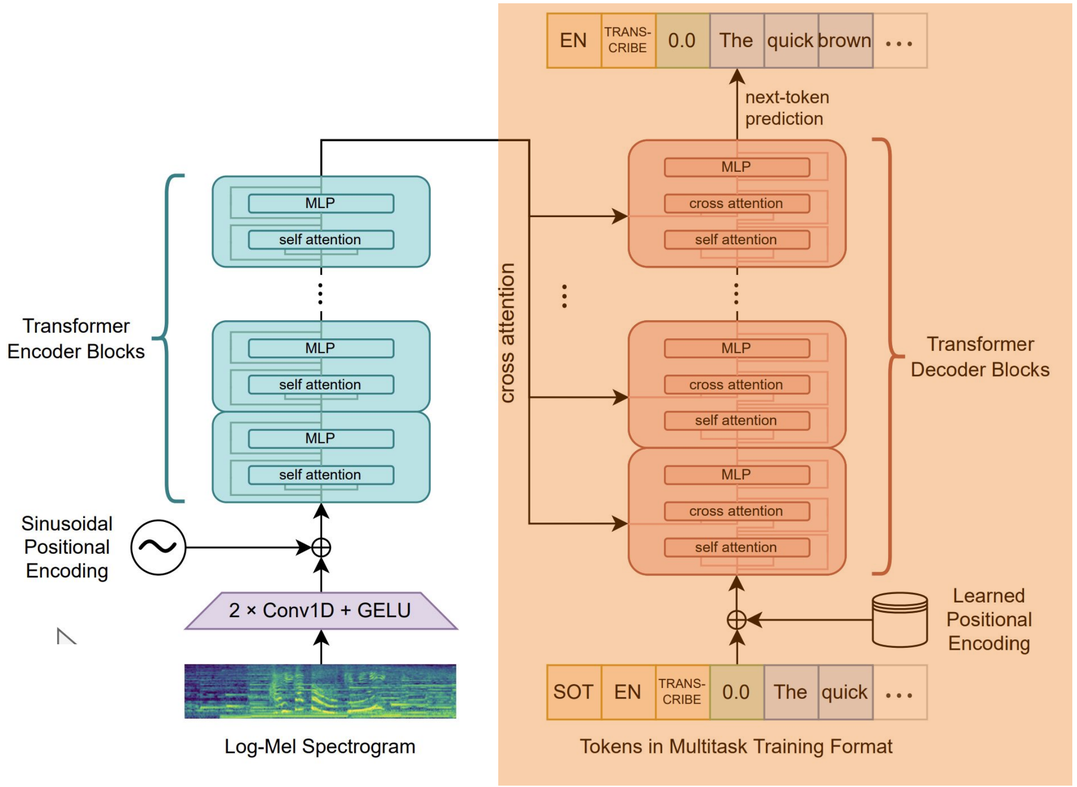


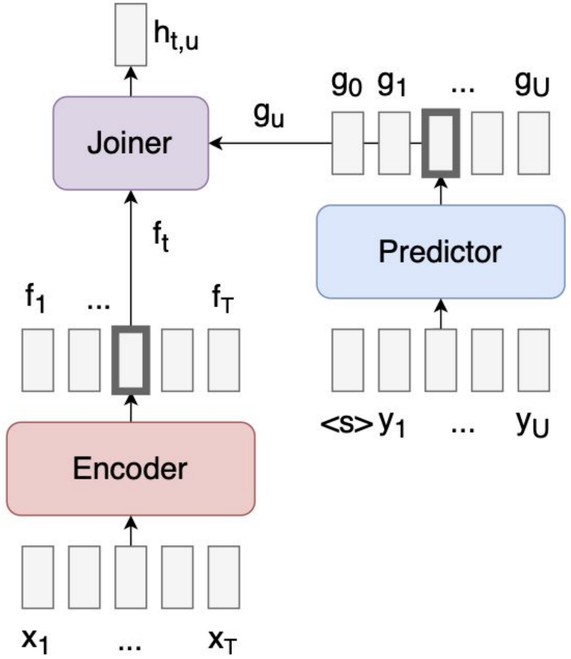


Вспомним архитектуры

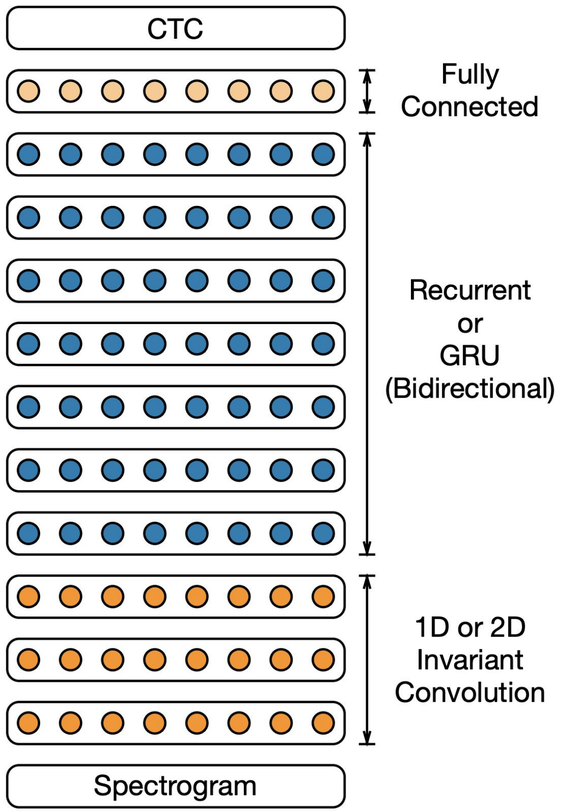


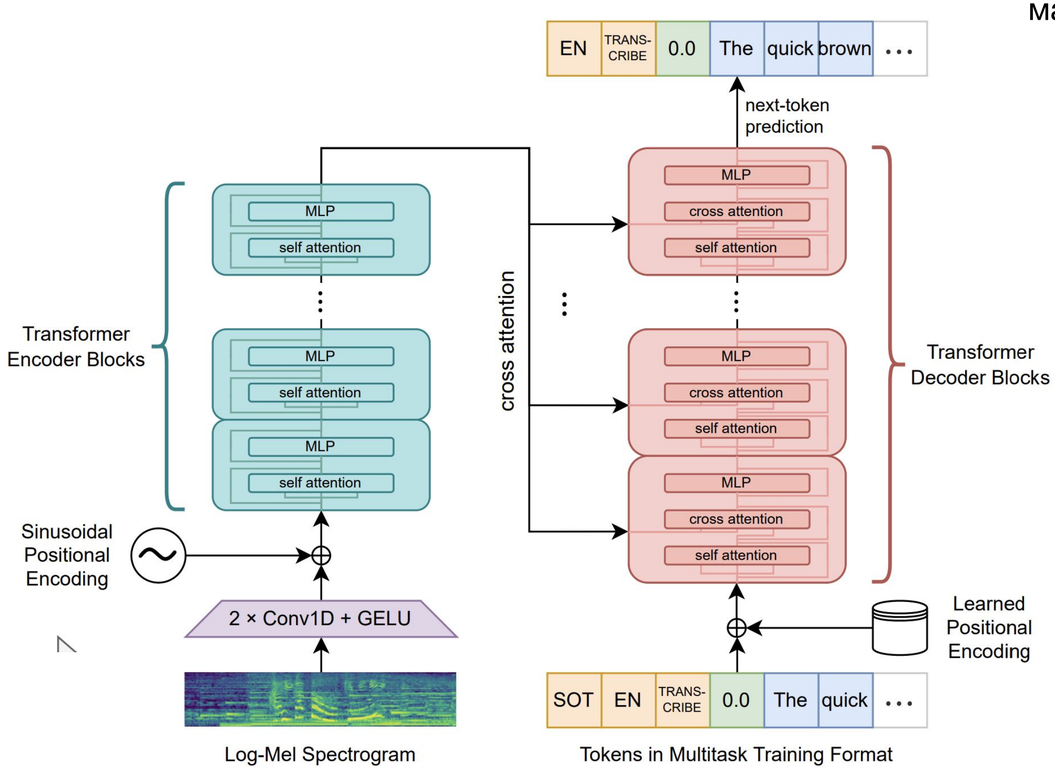
авторегрессионное предсказание
большой декодер (лингвистическая модель)



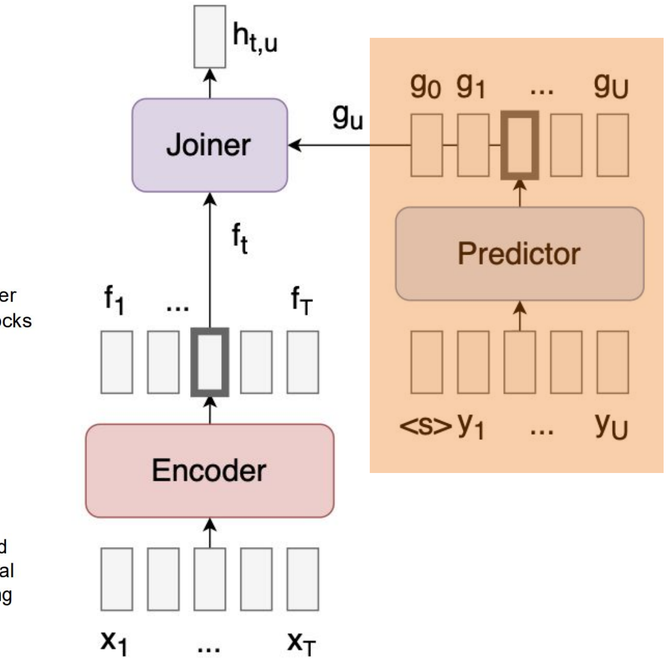


Вспомним архитектуры



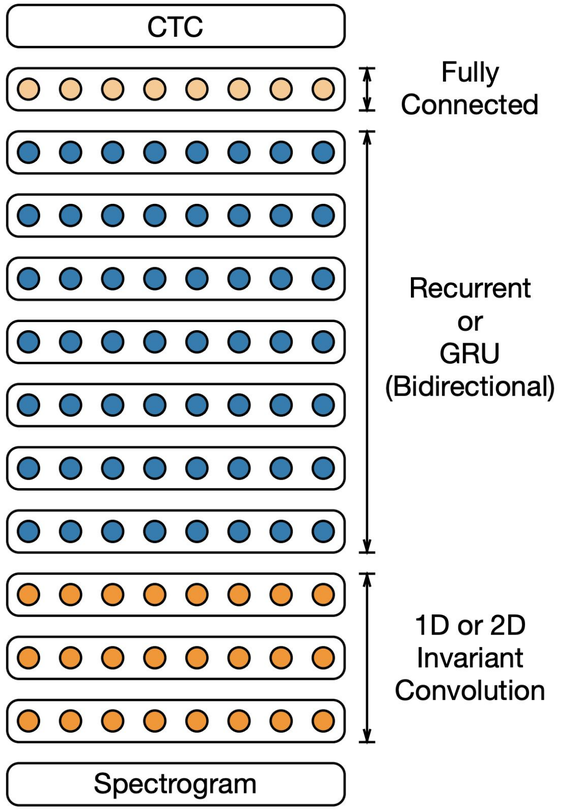


авторегрессионное предсказание
маленький декодер (лингвистическая модель)

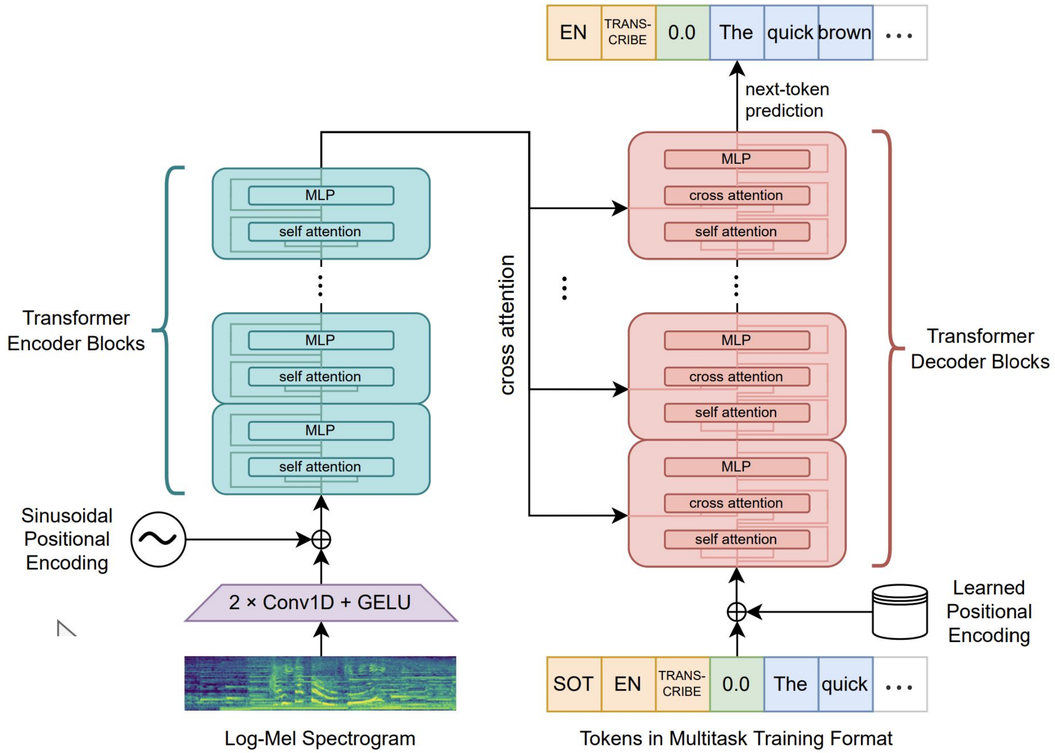


Вспомним архитектуры

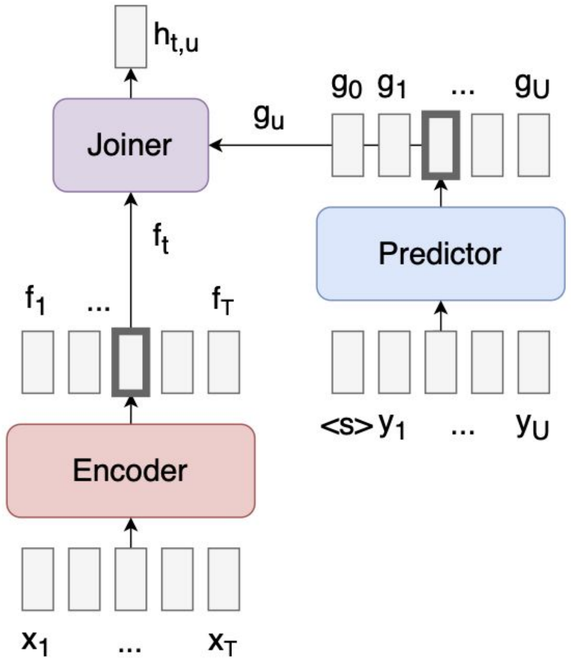
поиск лучшего выравнивания
между фреймами и токенами



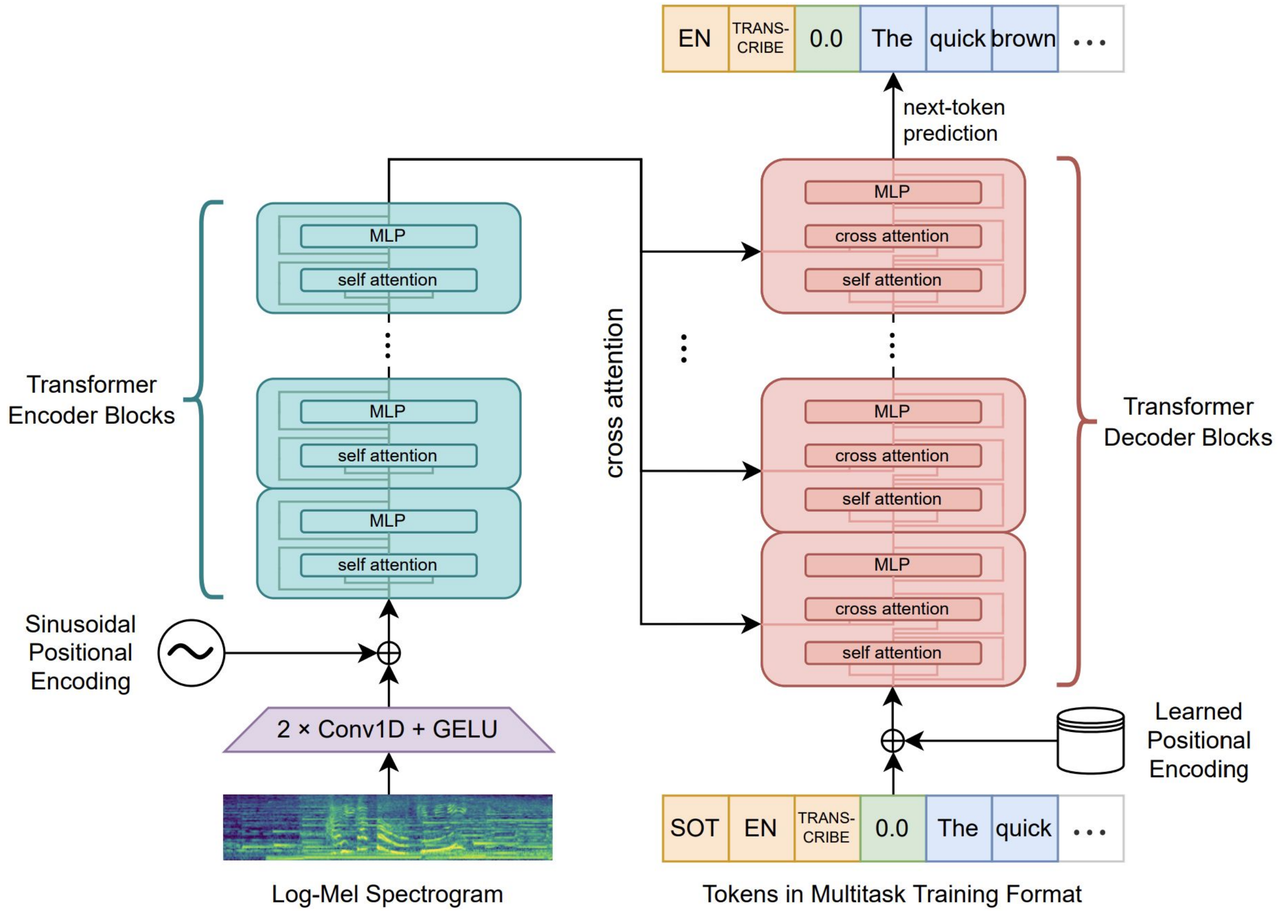
предсказание следующего токена



поиск лучшего выравнивания
между фреймами и токенами

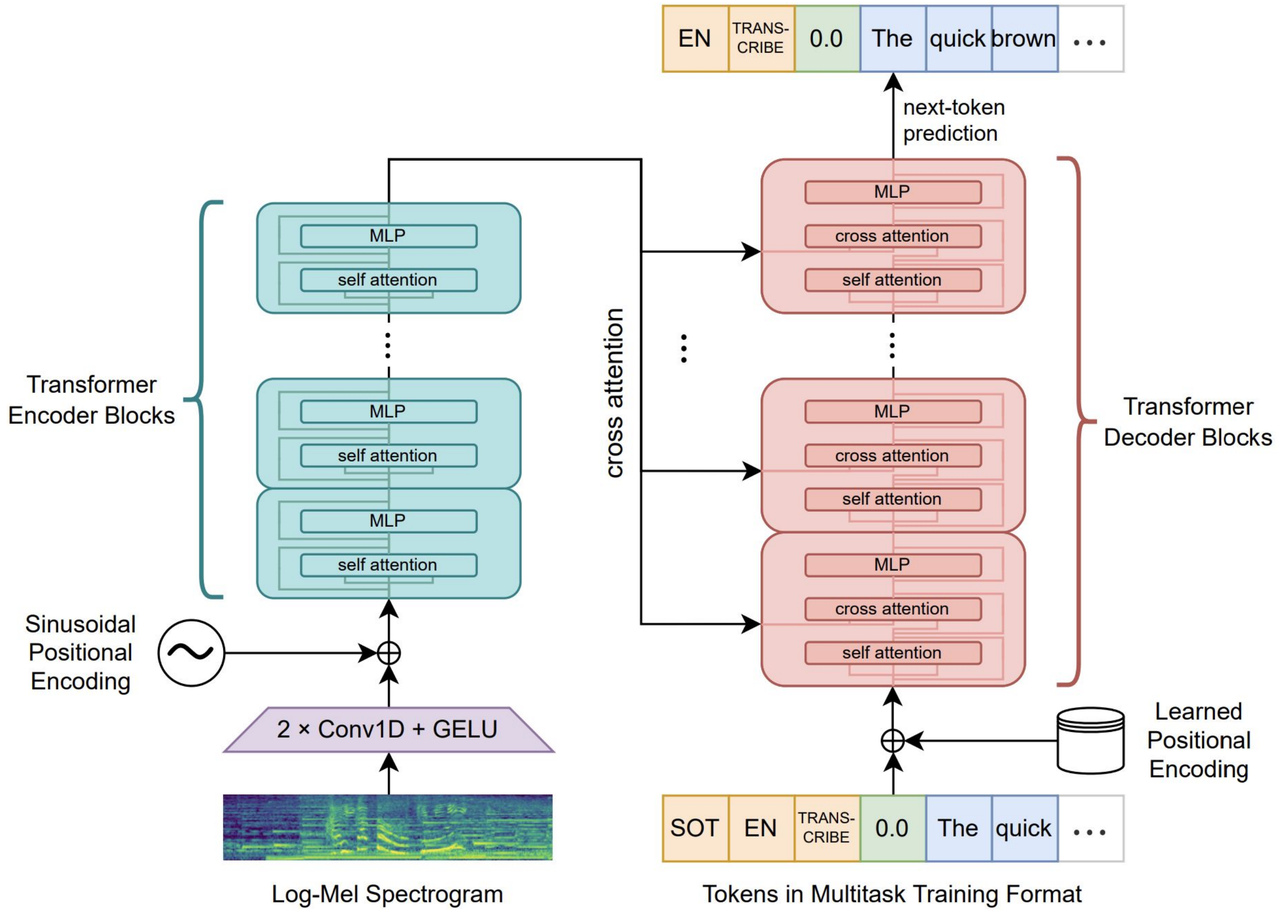


Типы вне-доменных данных



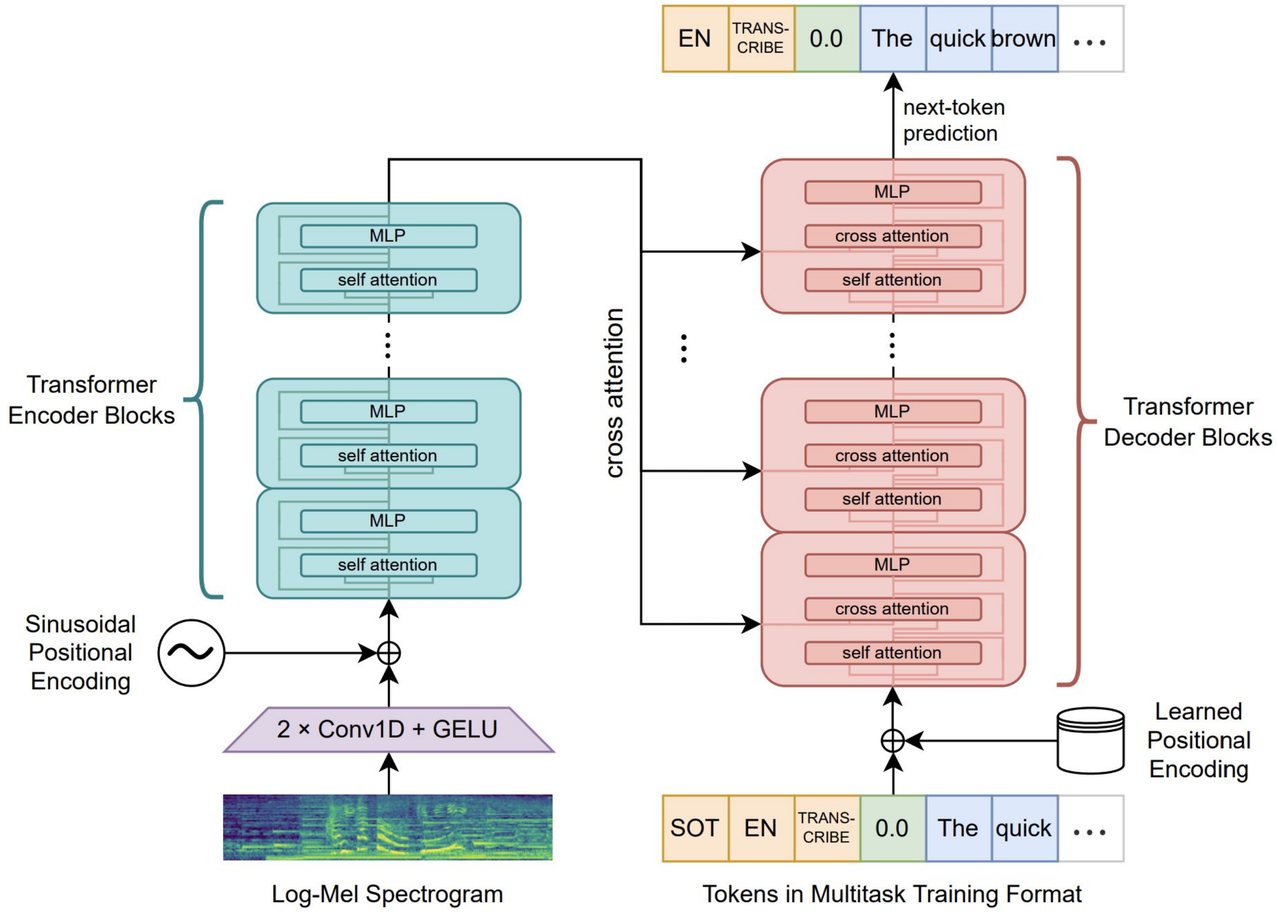
Типы вне-доменных данных

- Различные шумы



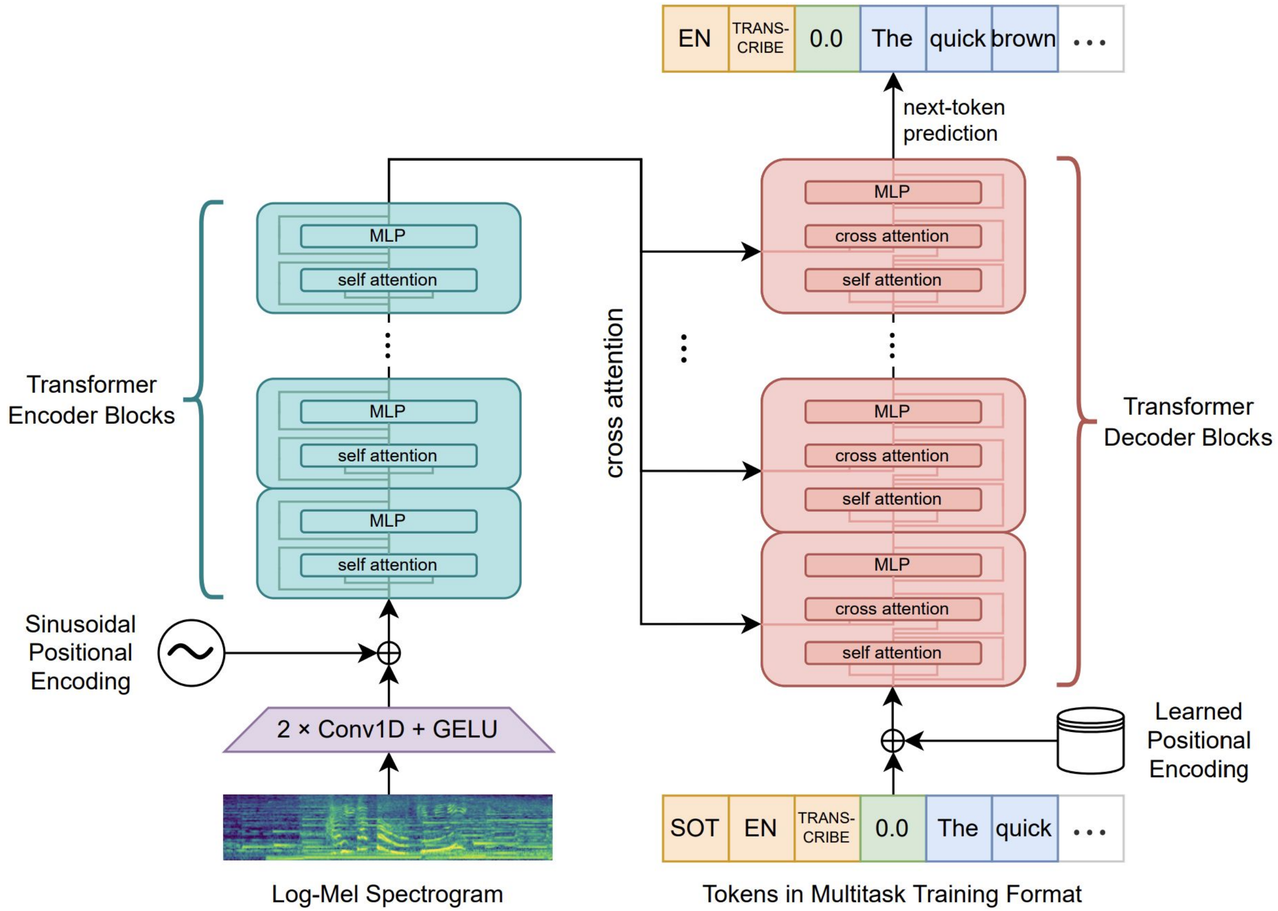
Типы вне-доменных данных

- Различные шумы
- Новые звуковые эвенты



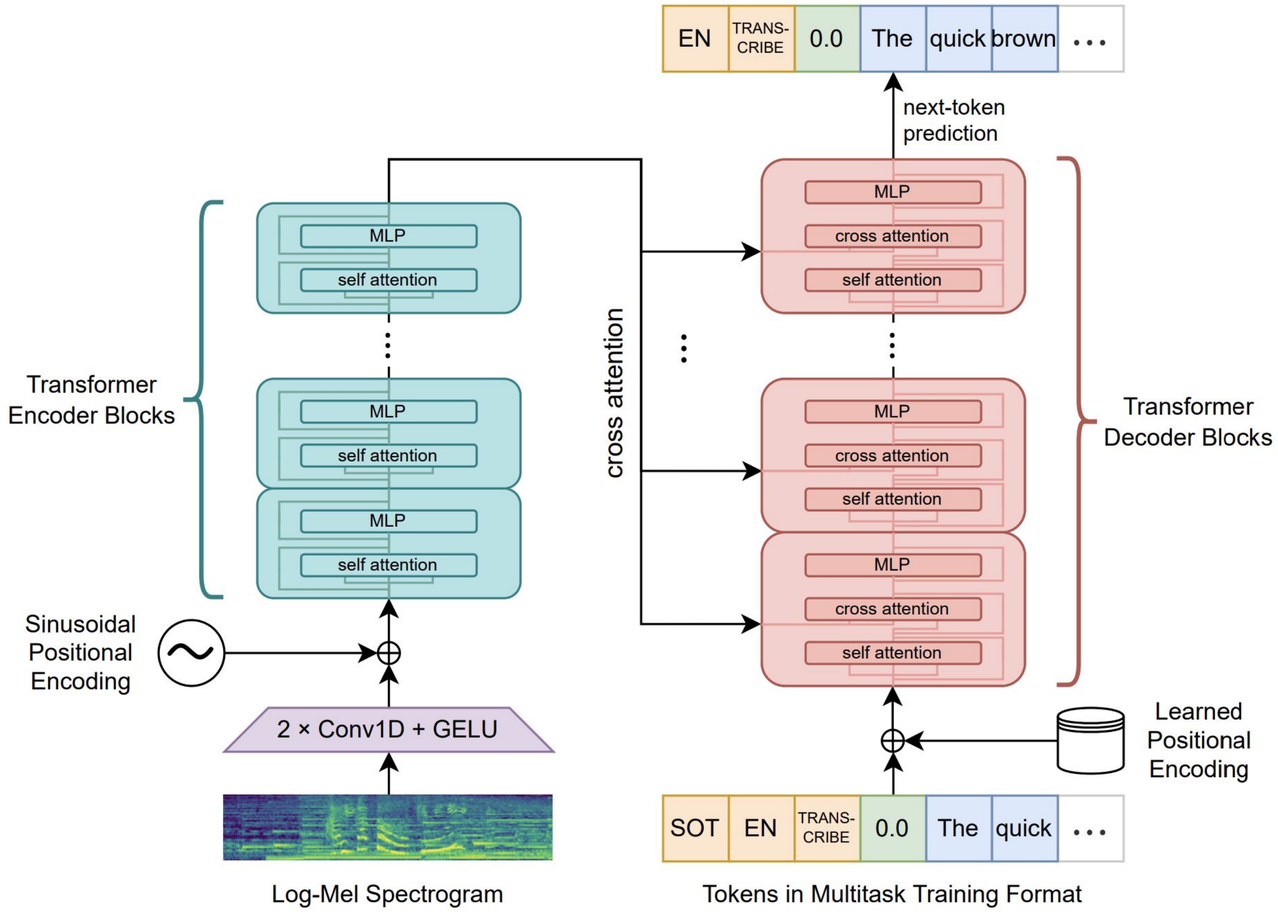
Типы вне-доменных данных

- Различные шумы
- Новые звуковые эвенты
- Эхо



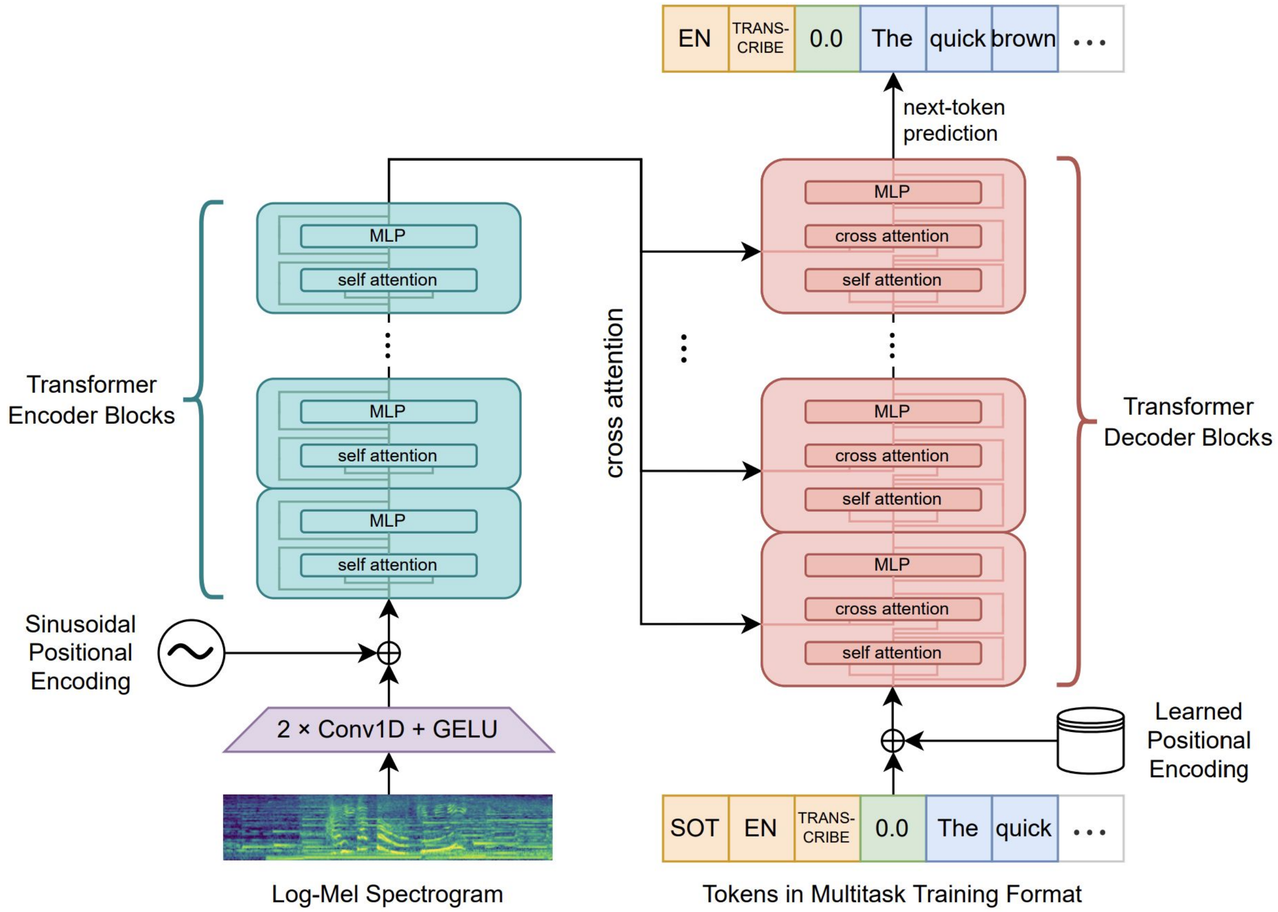
Типы вне-доменных данных

- Различные шумы
- Новые звуковые эвенты
- Эхо
- Акцент



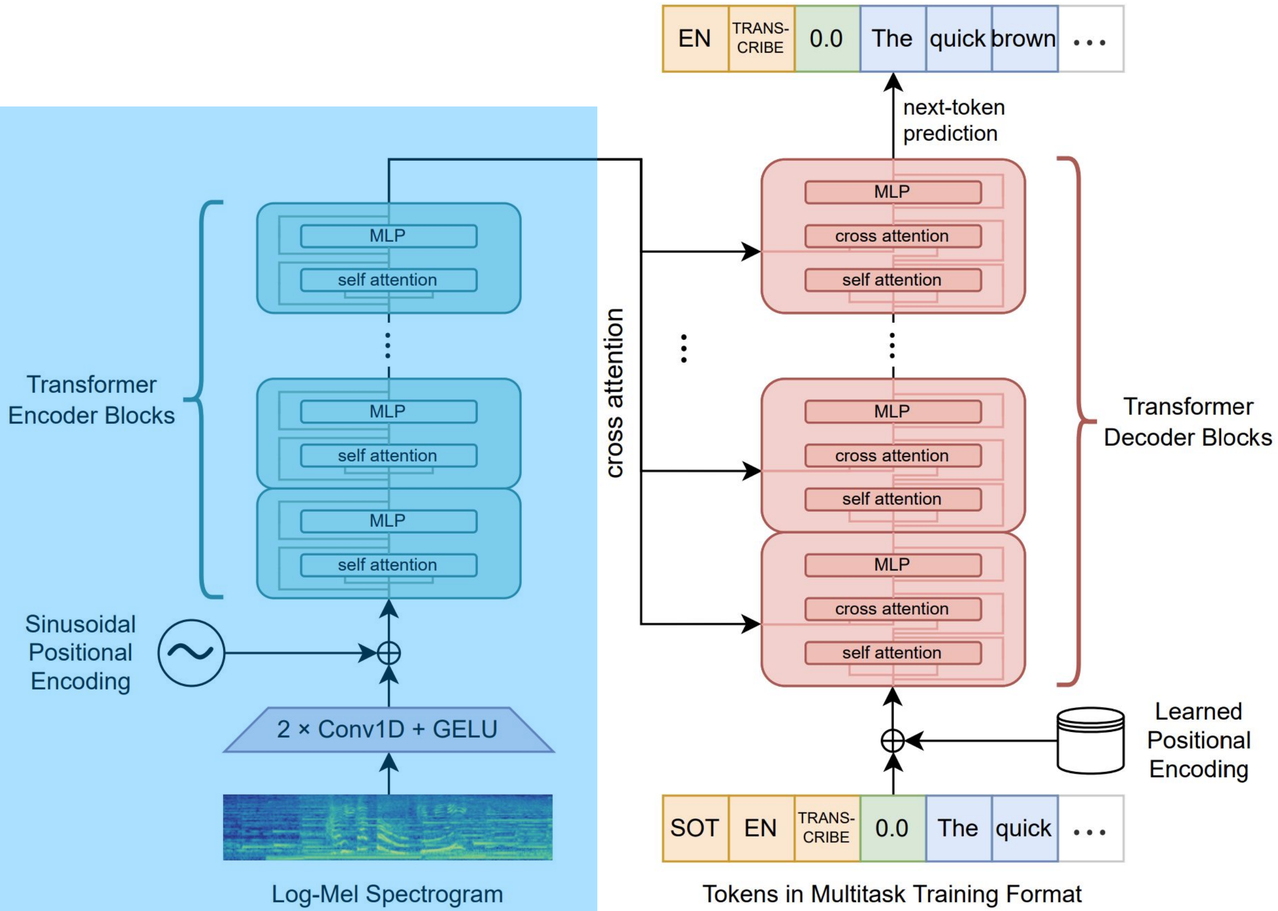
Типы вне-доменных данных

- Различные шумы
- Новые звуковые эвенты
- Эхо
- Акцент
- Громкость речи



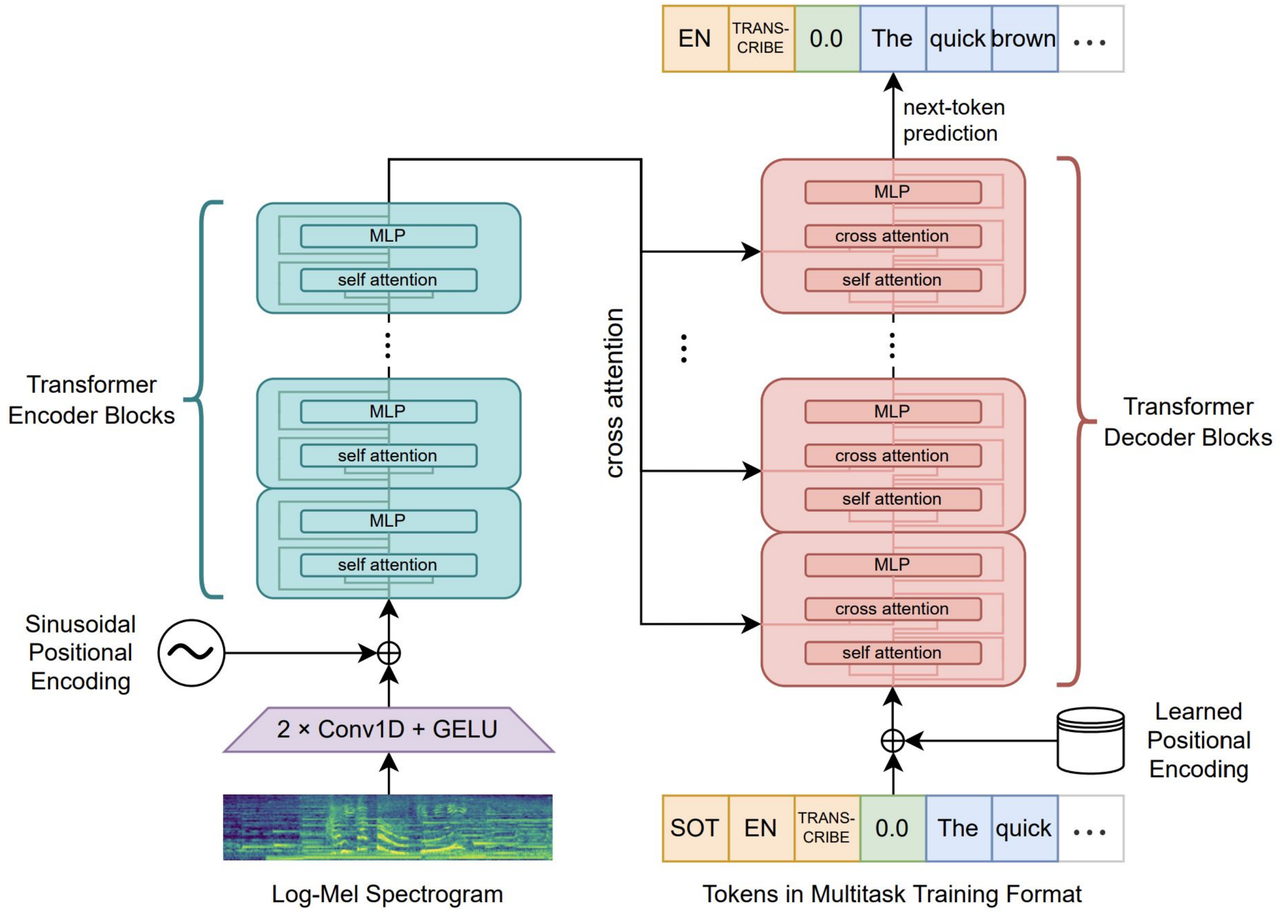
Типы вне-доменных данных

- Различные шумы
- Новые звуковые эвенты
- Эхо
- Акцент
- Громкость речи



Типы вне-доменных данных

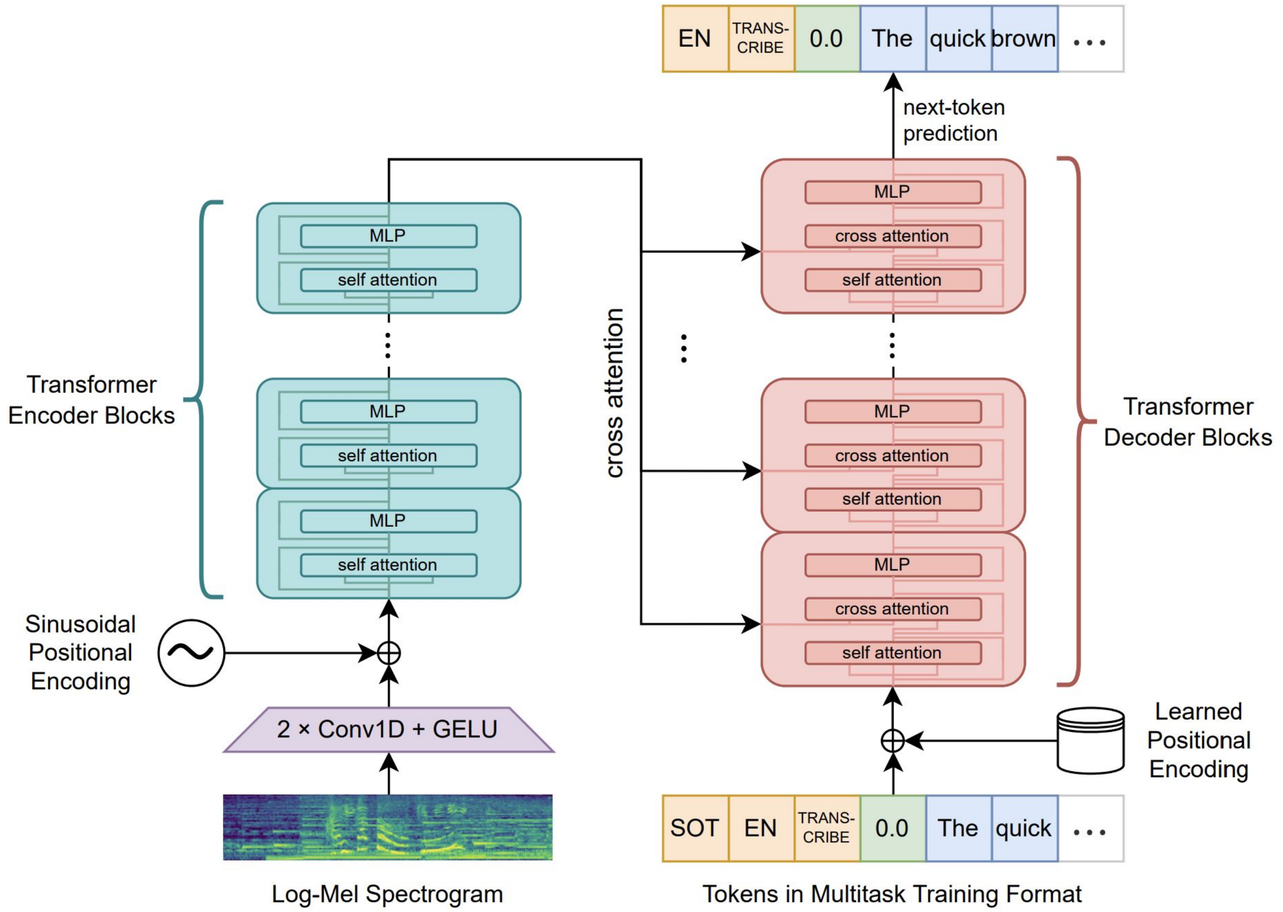
- Различные шумы
- Новые звуковые эвенты
- Эхо
- Акцент
- Громкость речи



- Новые слова

Типы вне-доменных данных

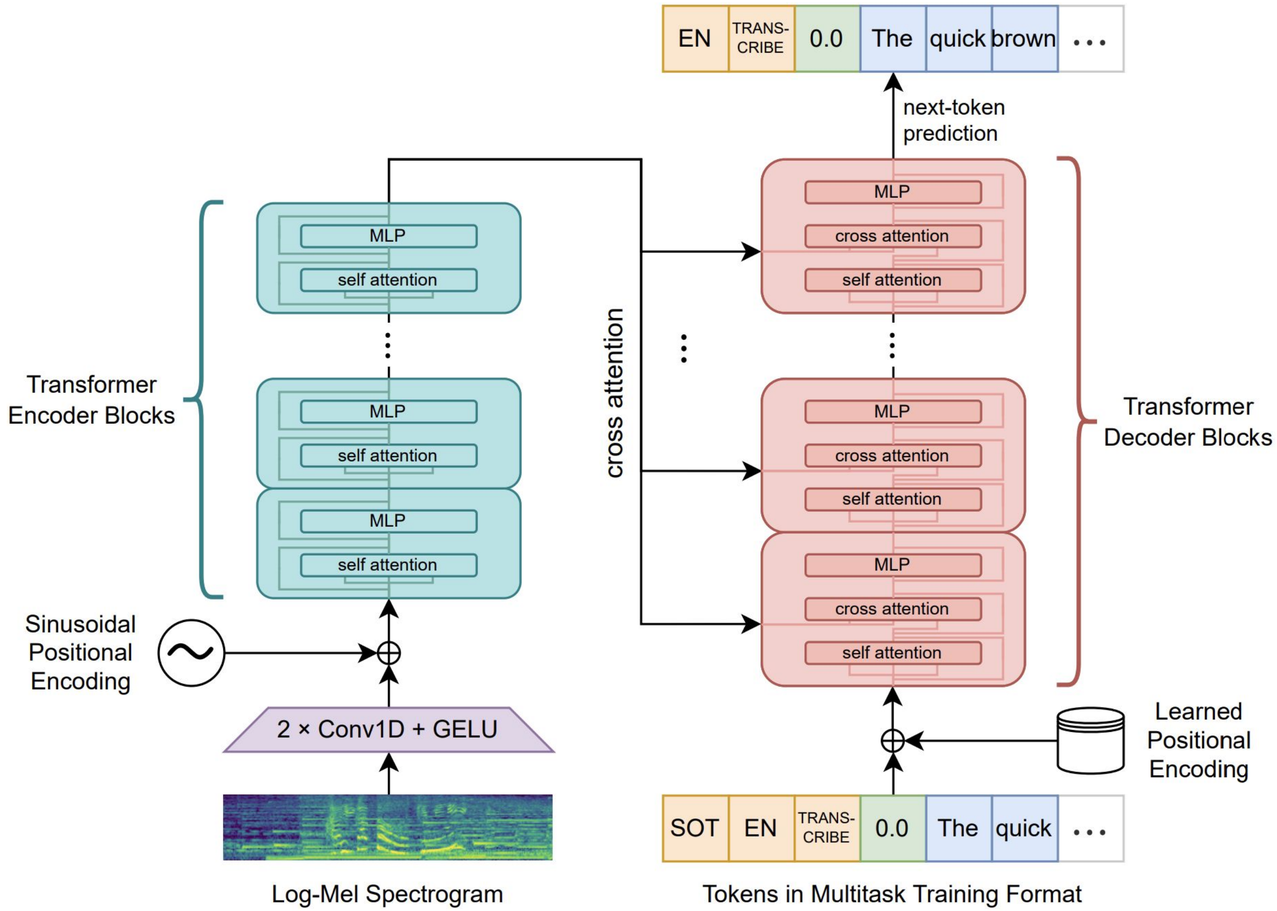
- Различные шумы
- Новые звуковые эвенты
- Эхо
- Акцент
- Громкость речи



- Новые слова
- Проф. термины

Типы вне-доменных данных

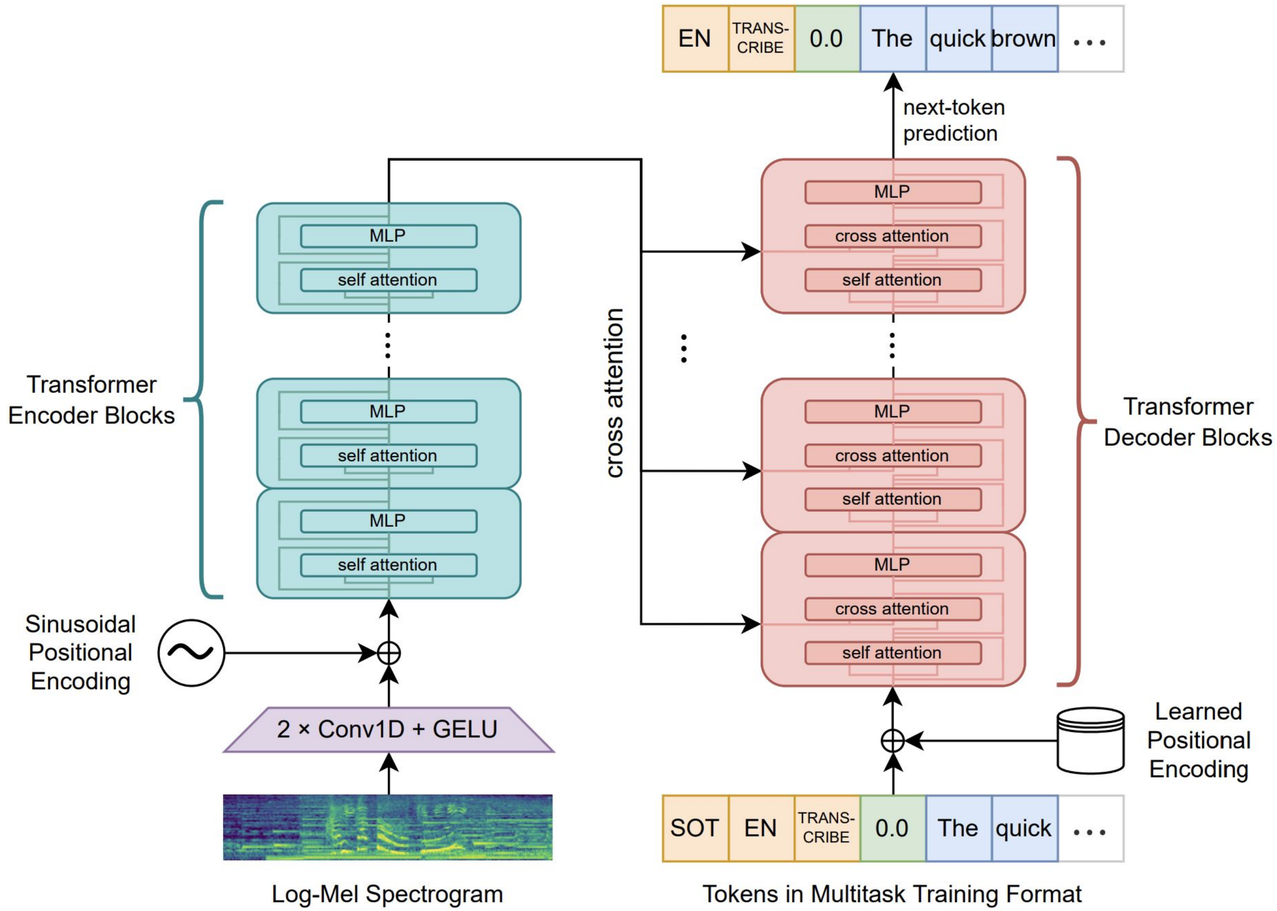
- Различные шумы
- Новые звуковые эвенты
- Эхо
- Акцент
- Громкость речи



- Новые слова
- Проф. термины
- Сленг

Типы вне-доменных данных

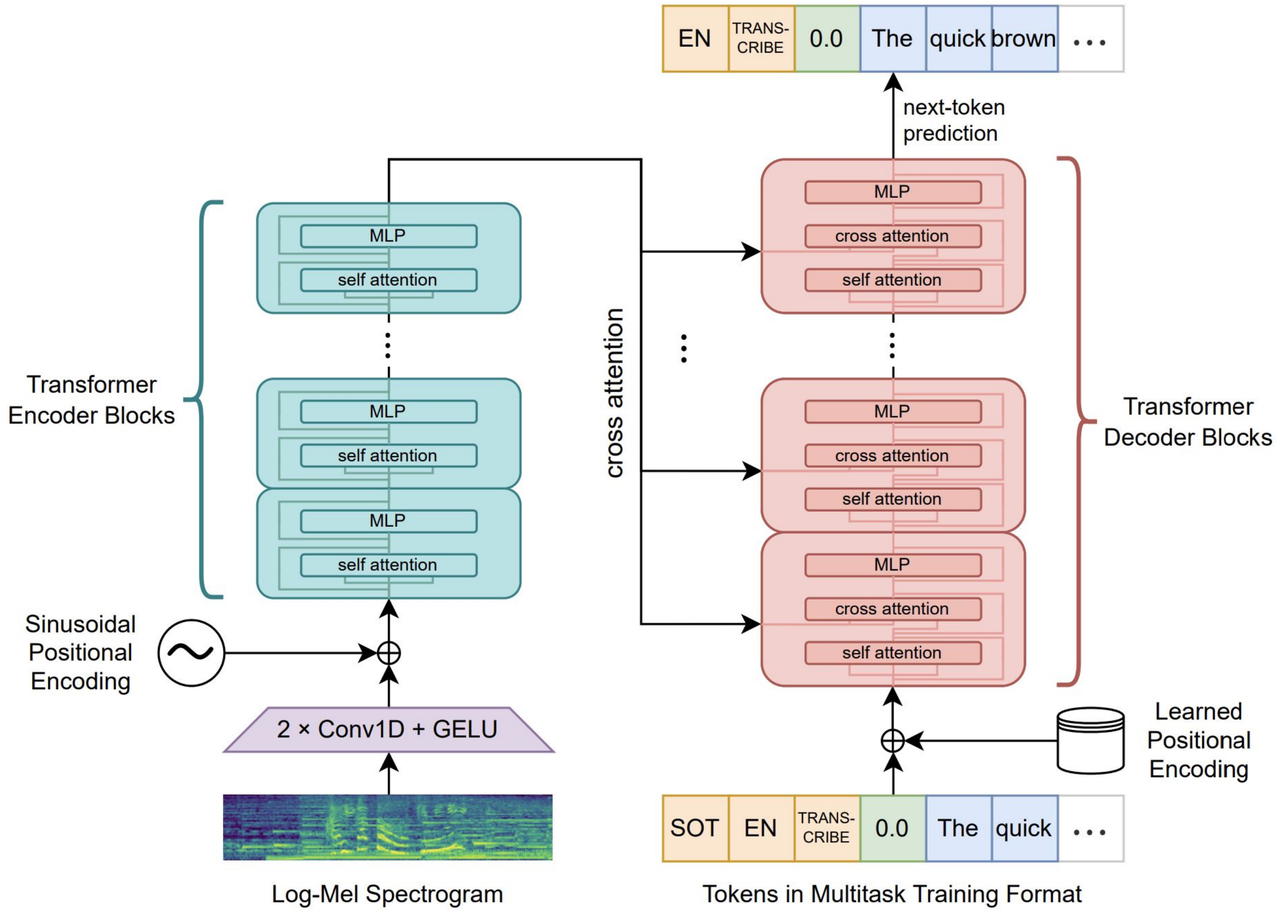
- Различные шумы
- Новые звуковые эвенты
- Эхо
- Акцент
- Громкость речи



- Новые слова
- Проф. термины
- Сленг
- Акцент (Диалект)

Типы вне-доменных данных

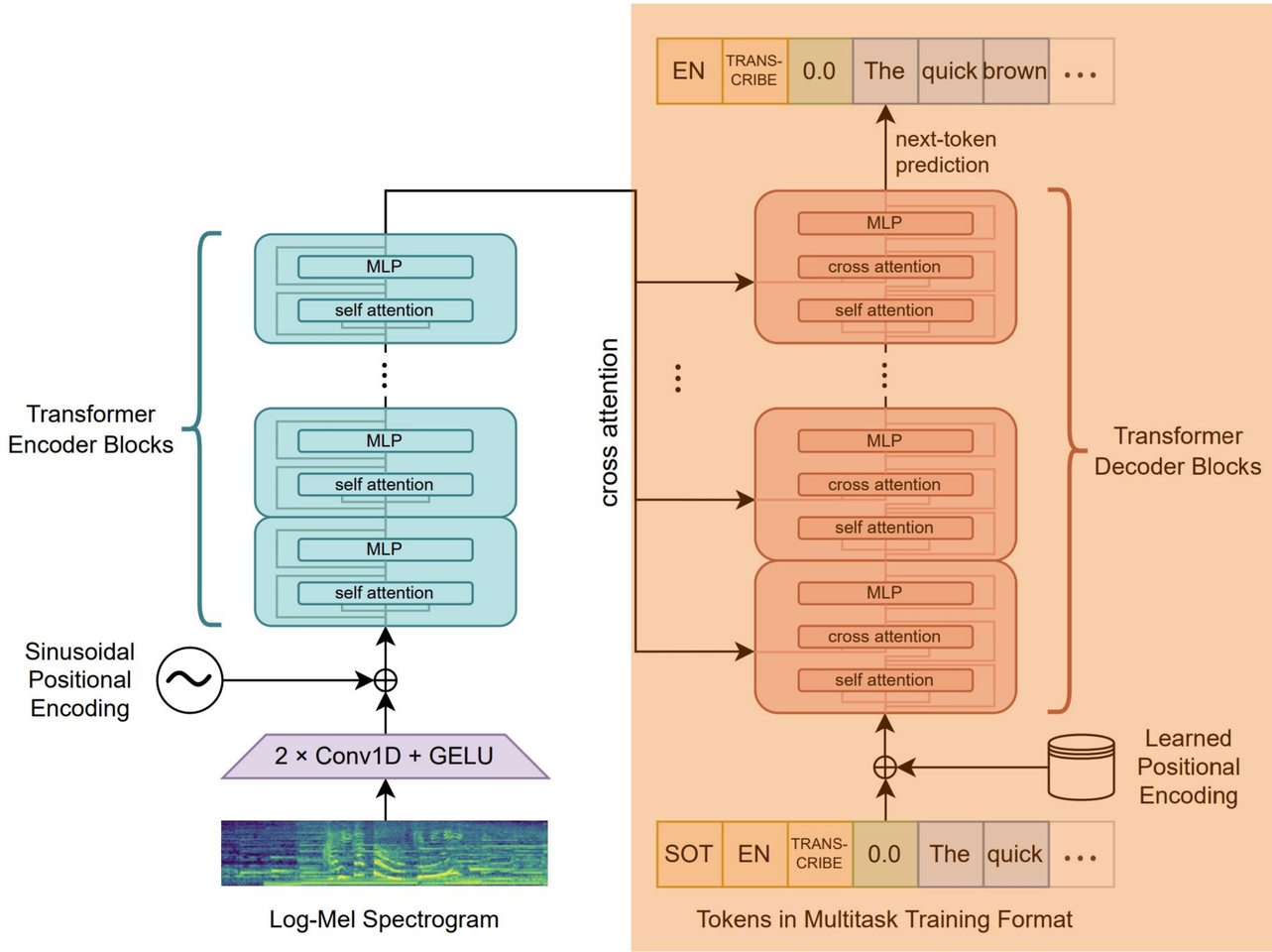
- Различные шумы
- Новые звуковые эвенты
- Эхо
- Акцент
- Громкость речи



- Новые слова
- Проф. термины
- Сленг
- Акцент (Диалект)
- Новый язык

Типы вне-доменных данных

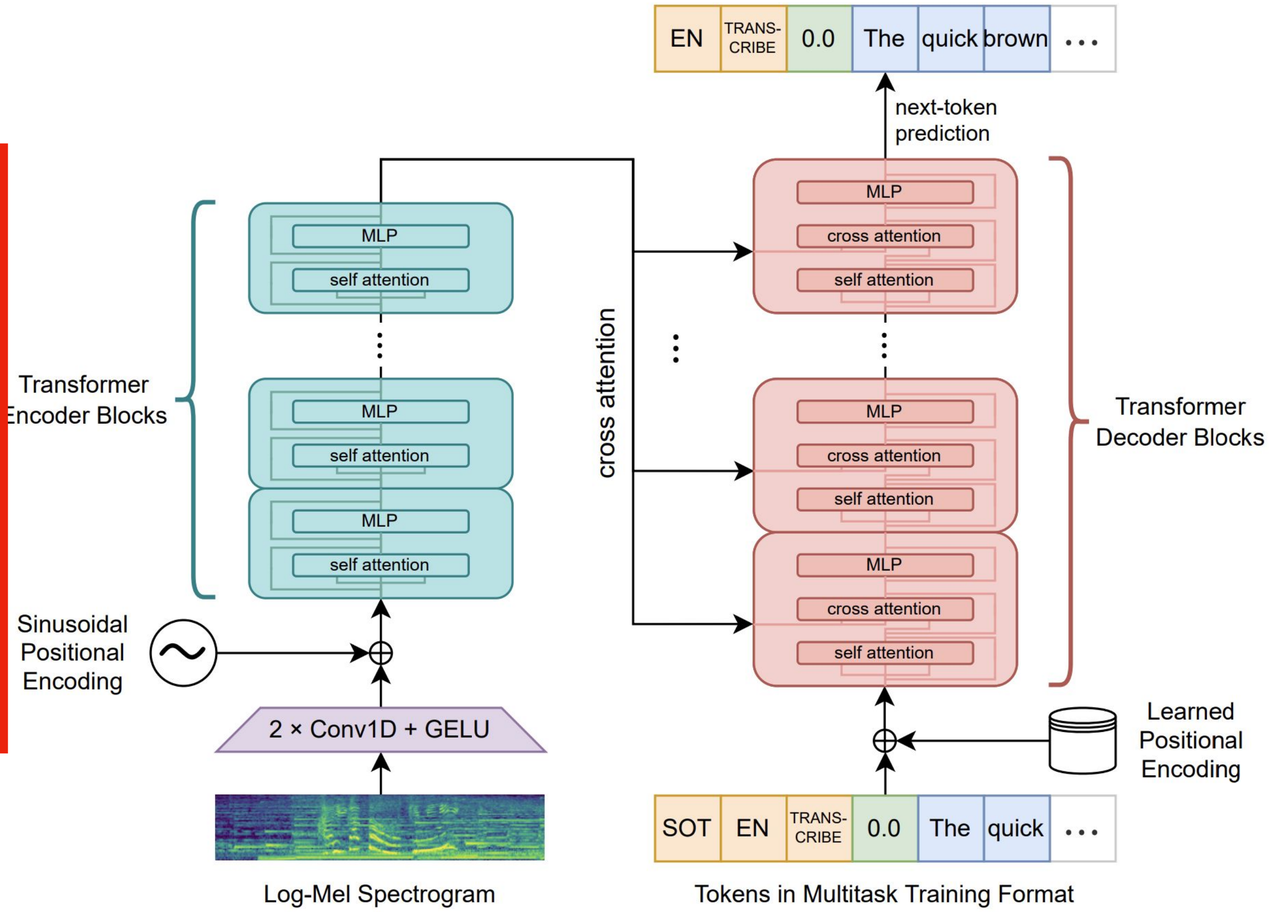
- Различные шумы
- Новые звуковые эвенты
- Эхо
- Акцент
- Громкость речи



- Новые слова
- Проф. термины
- Сленг
- Акцент (Диалект)
- Новый язык

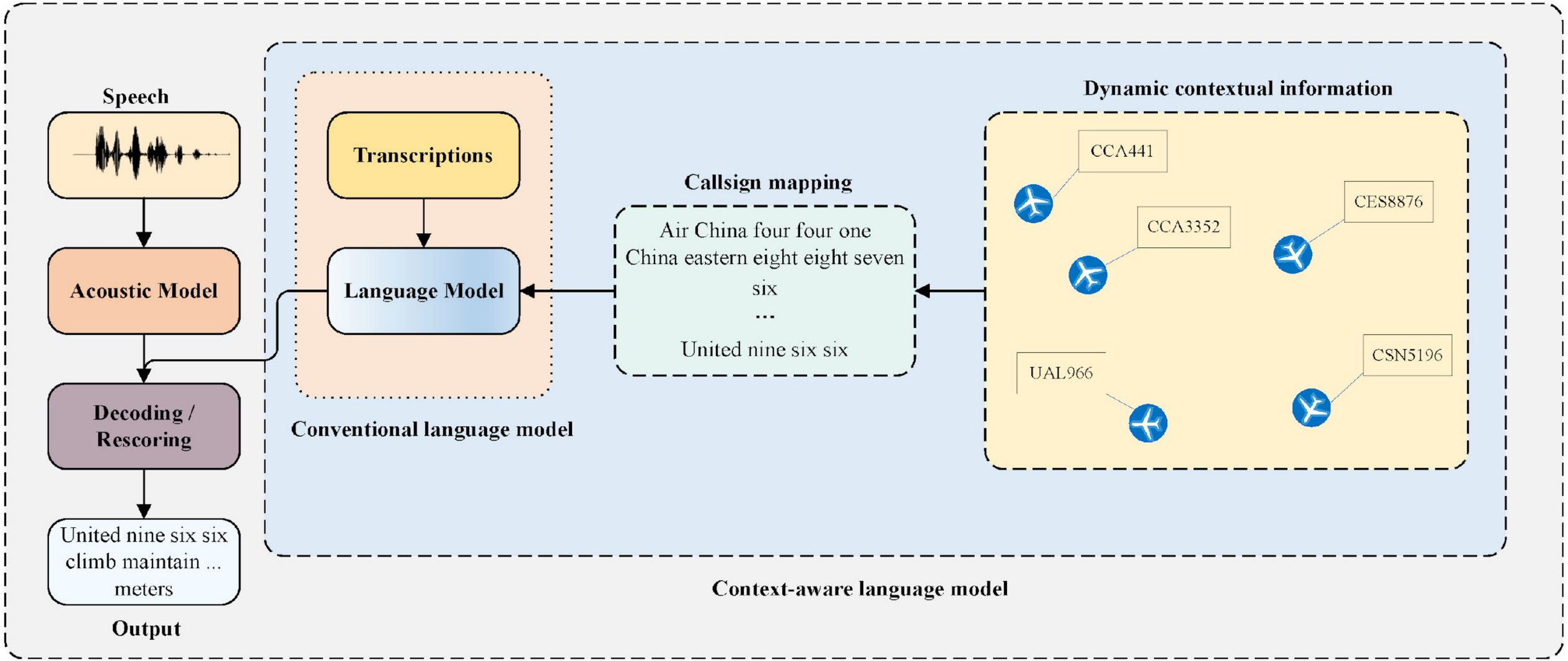
Типы вне-доменных данных

- Различные шумы
- Новые звуковые эвенты
- Эхо
- Акцент
- Громкость речи

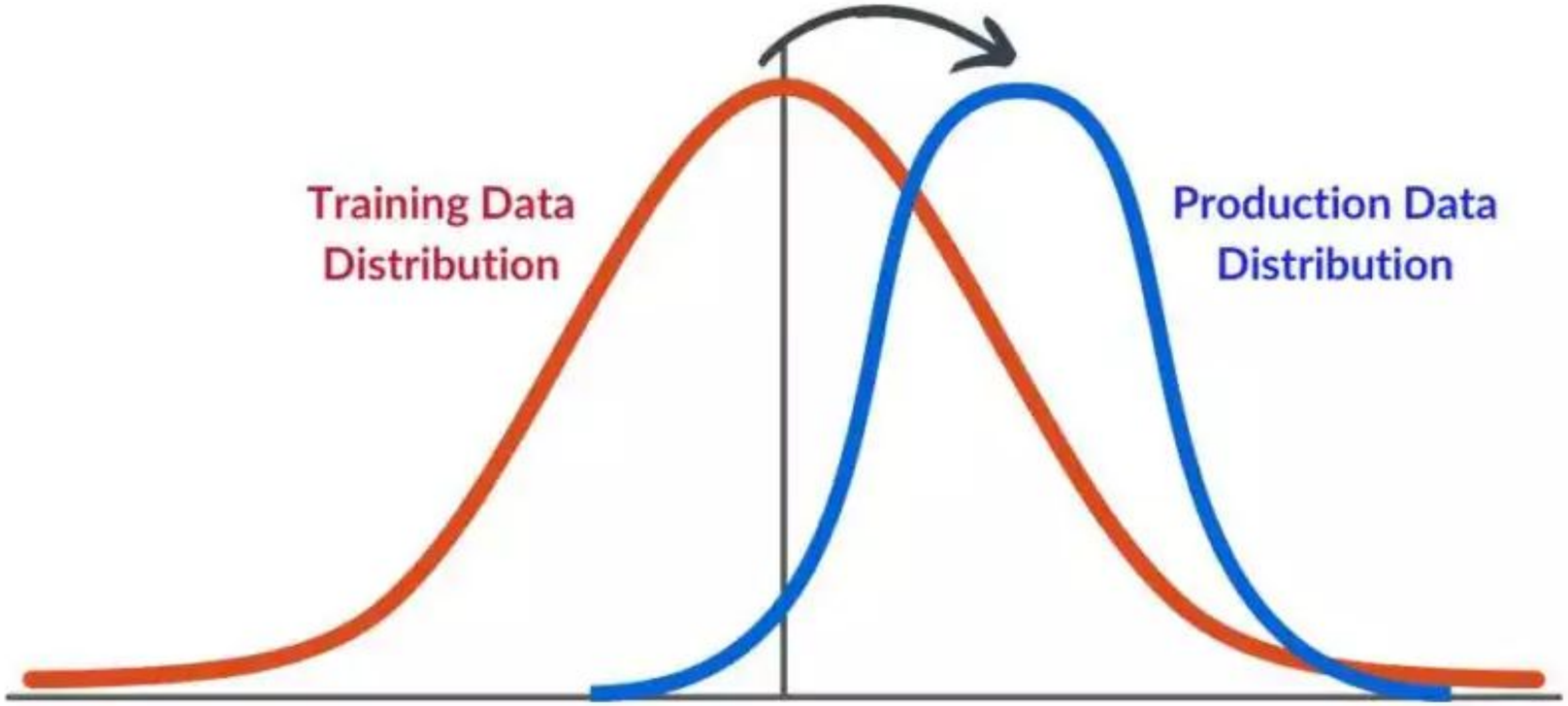


- Новые слова
- Проф. термины
- Сленг
- Акцент (Диалект)
- Новый язык

Типы вне-доменных данных

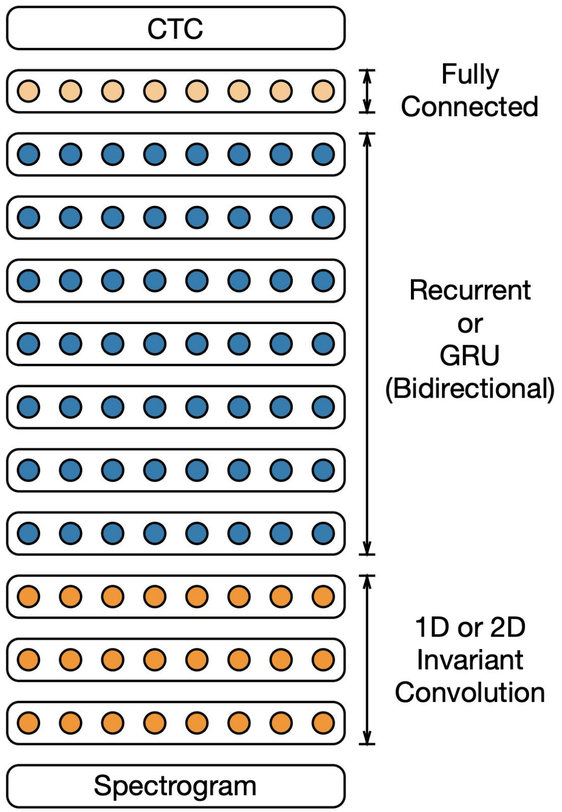


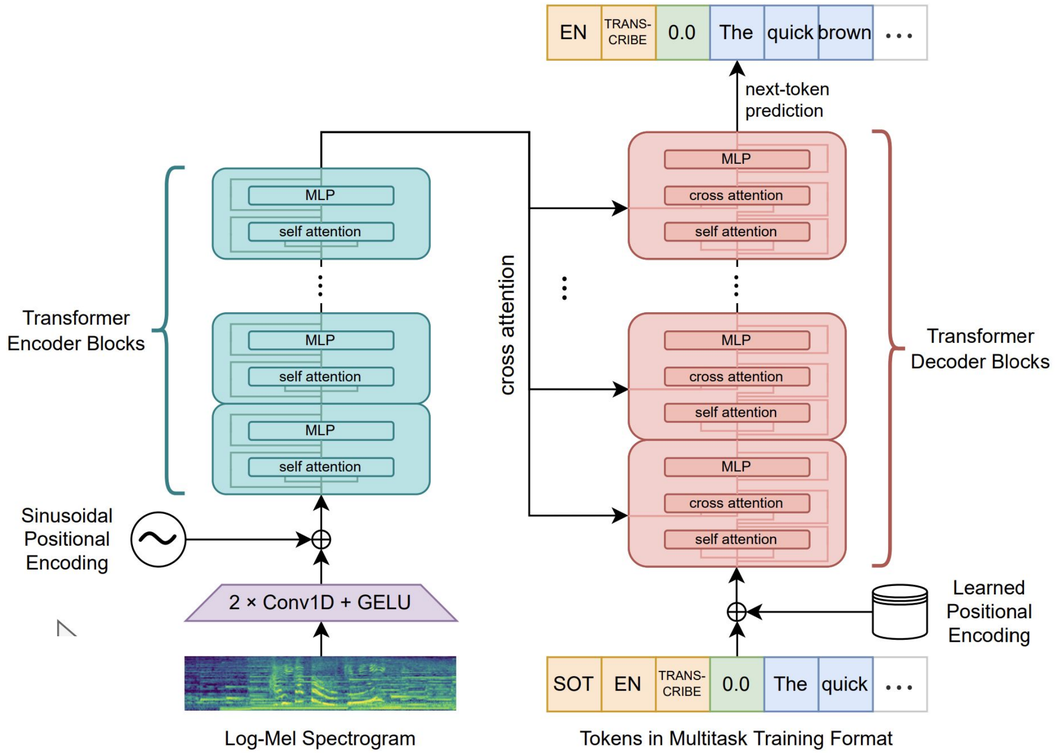
Типы вне-доменных данных

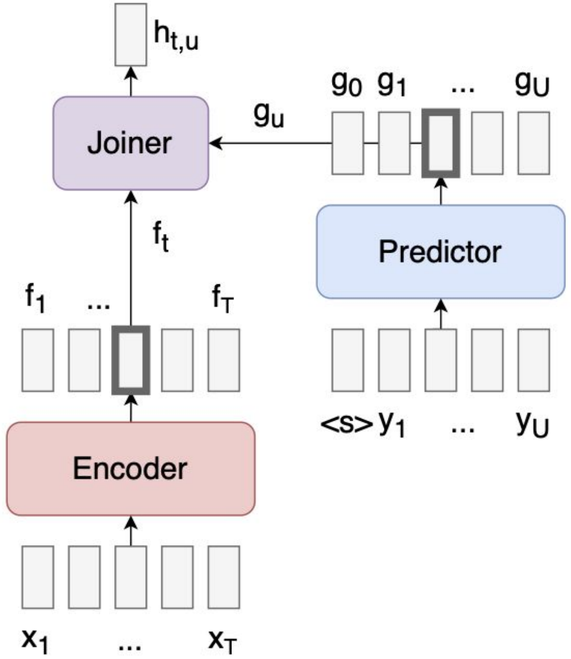


Методы адаптации

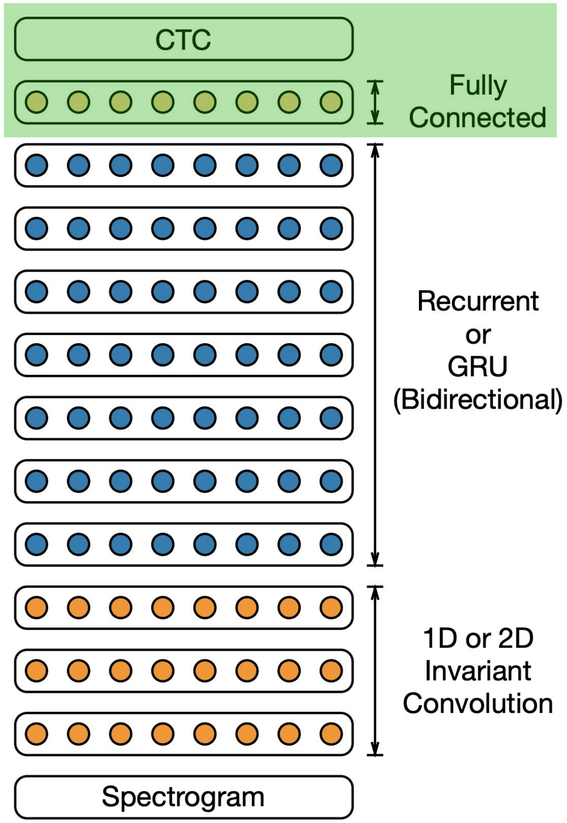
Методы адаптации

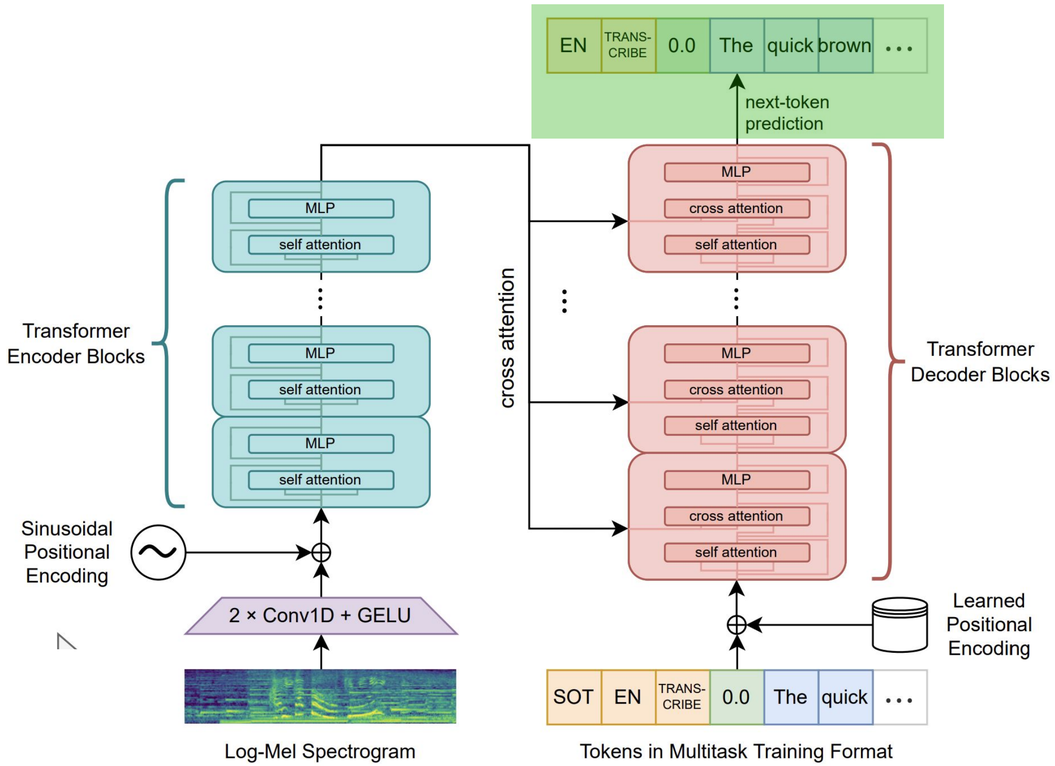


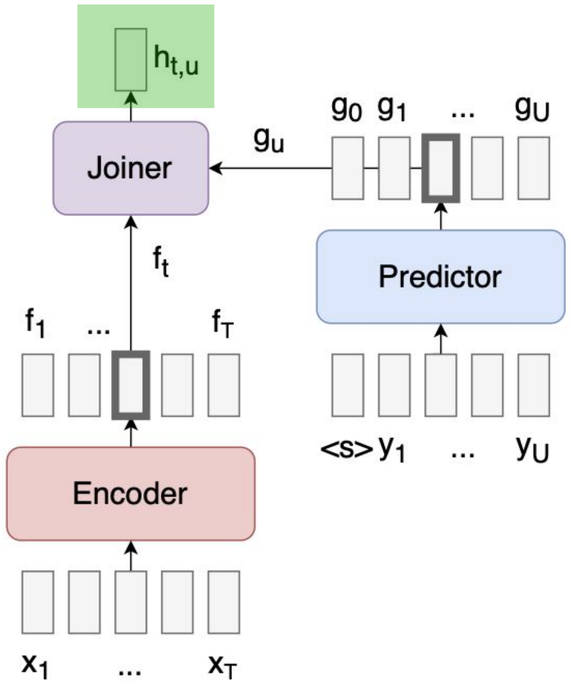




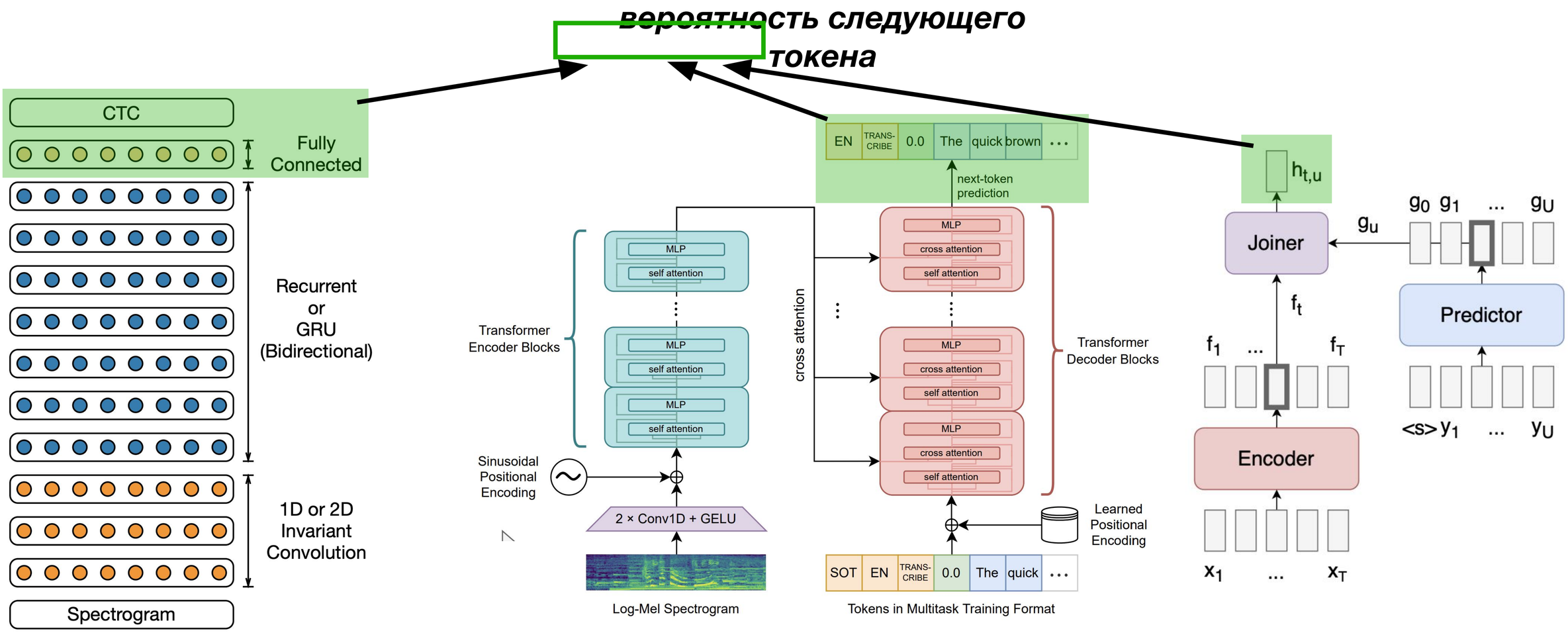
Методы адаптации



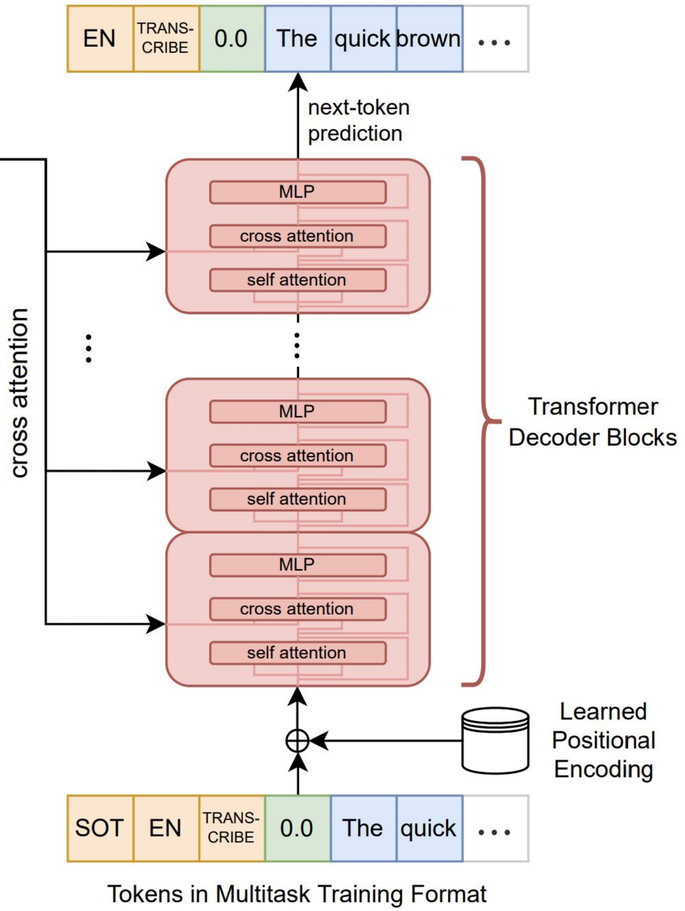




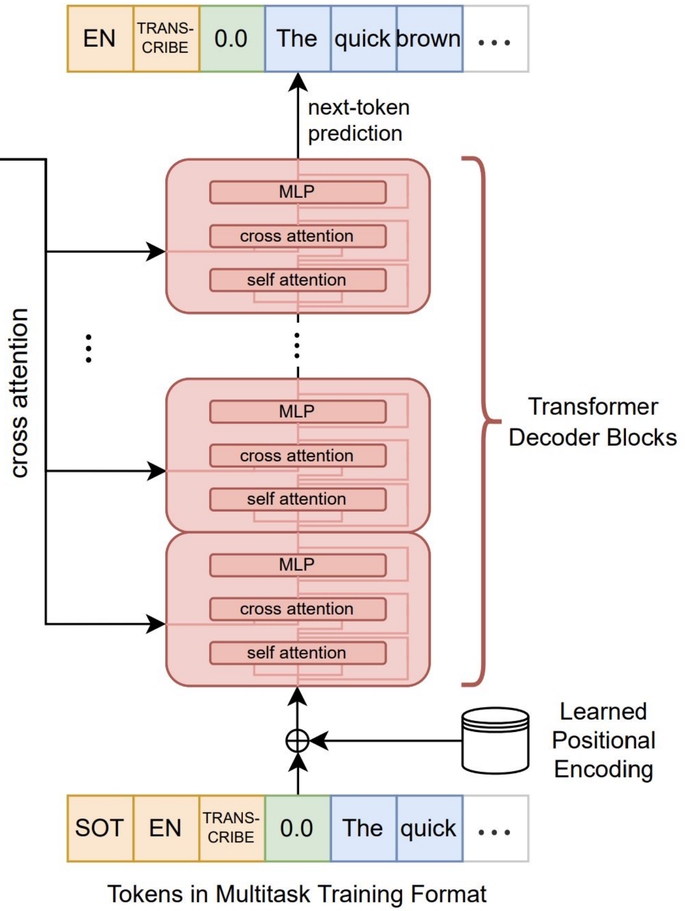
Методы адаптации



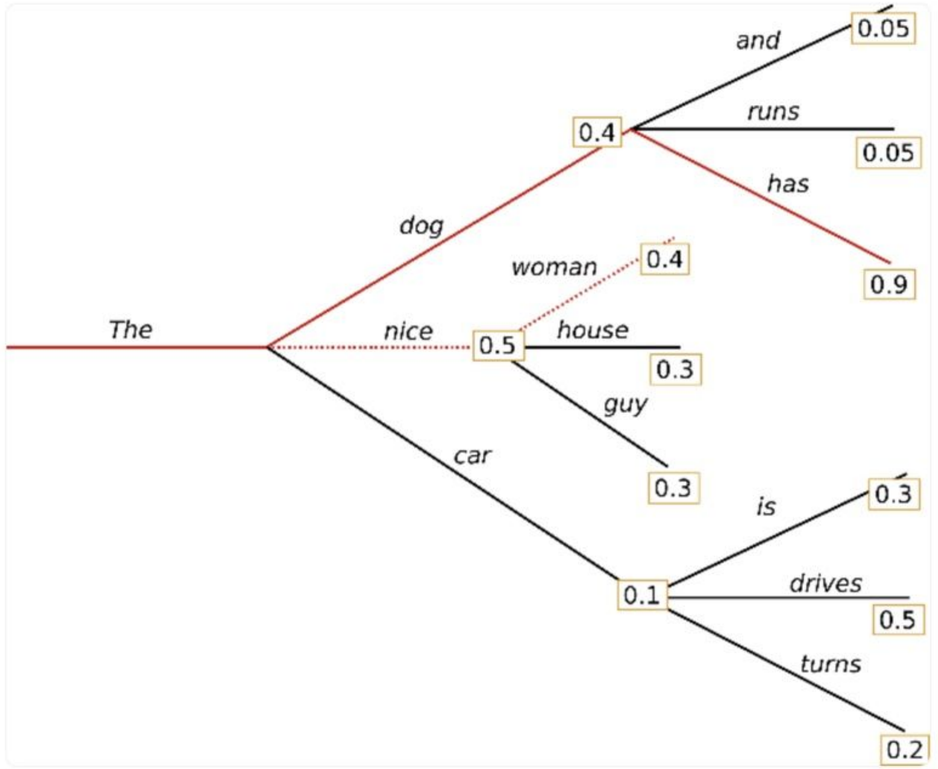
Методы адаптации



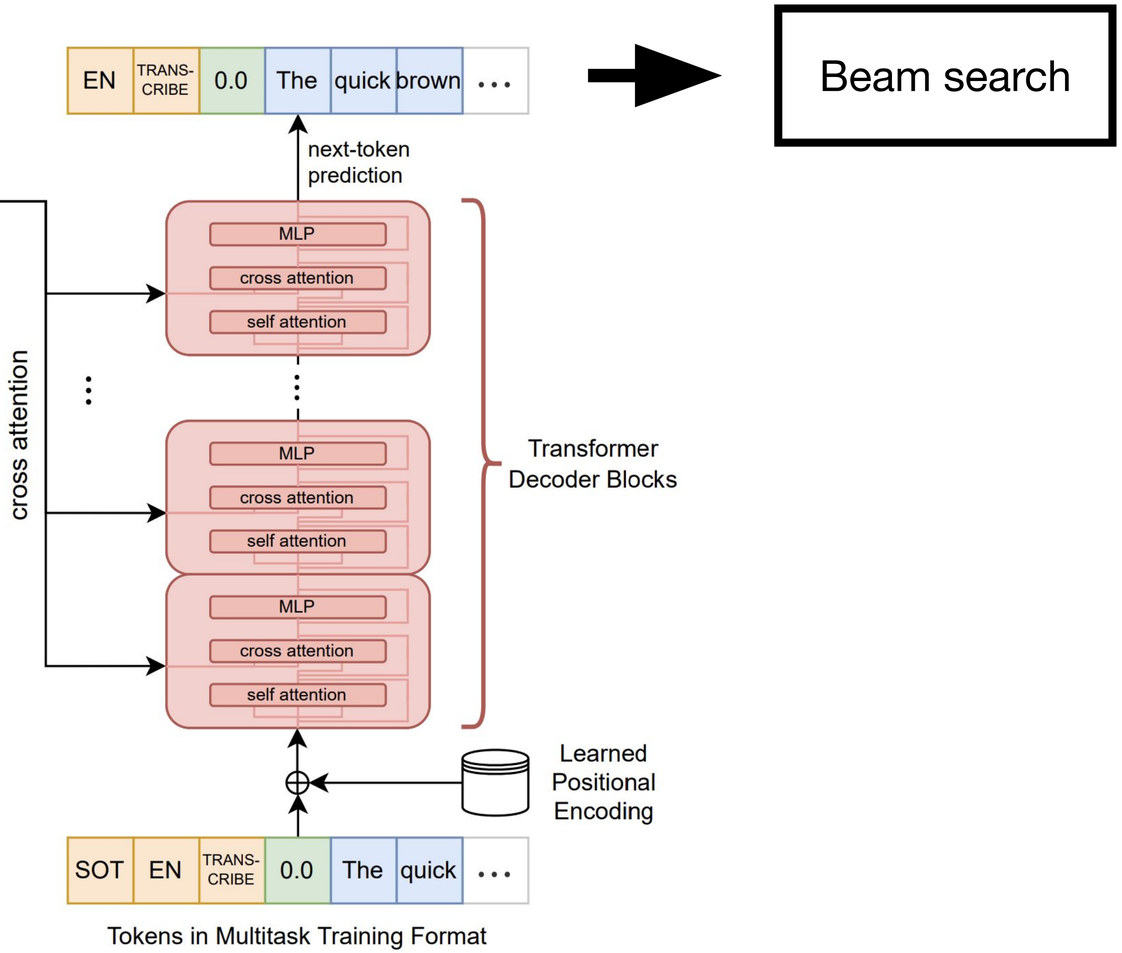
Методы адаптации



Beam

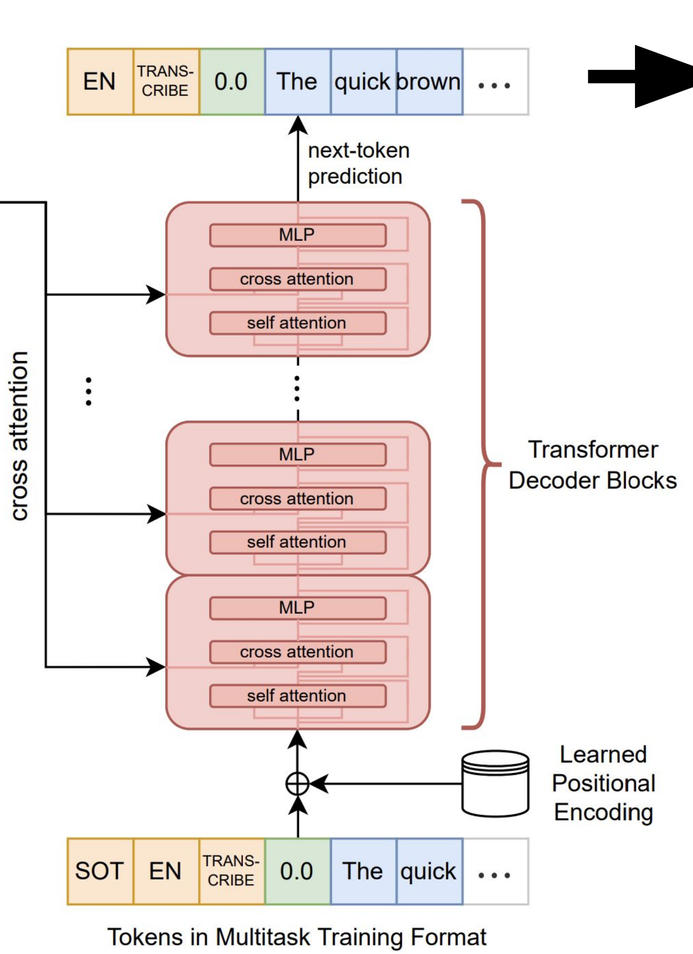


Методы адаптации



Beam search

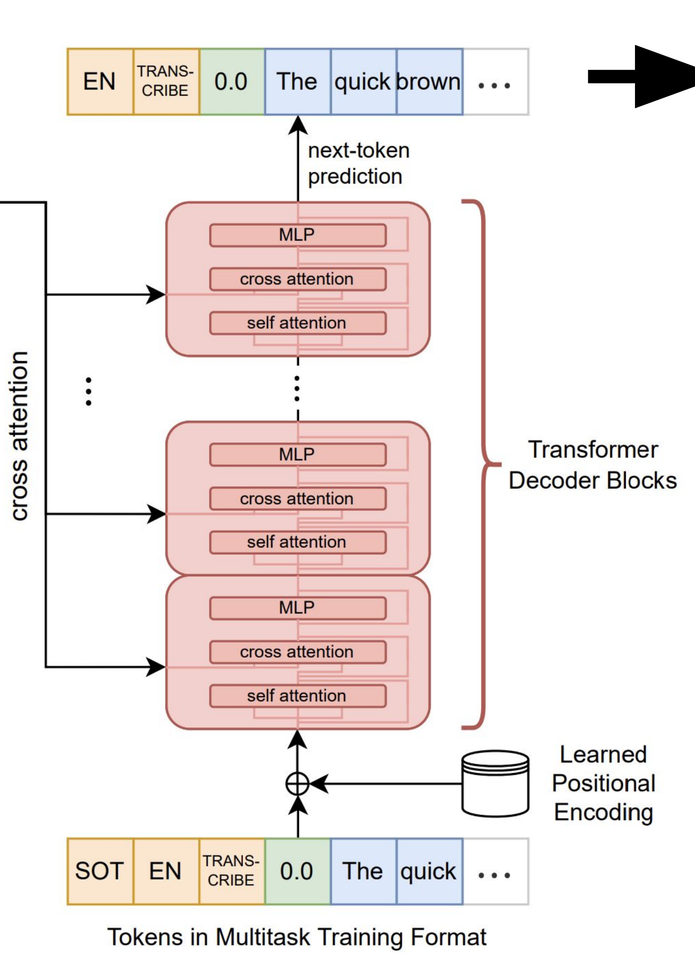
Методы адаптации



Beam search

Адаптация может происходить на уровне
пересчёта скоров в лучах
и на уровне реранкинга итоговых последовательной

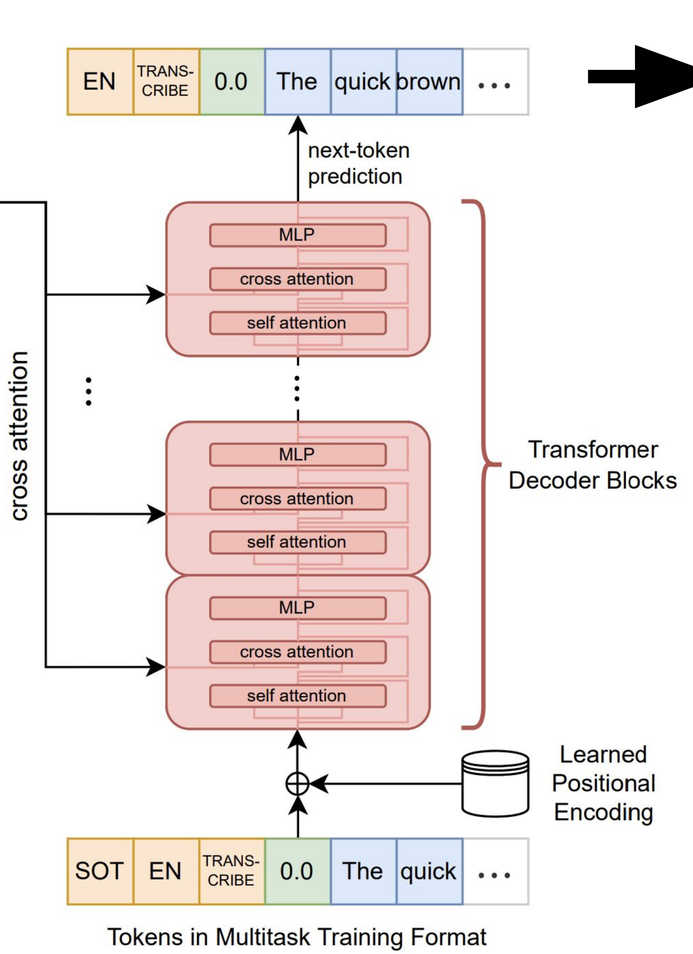
Методы адаптации

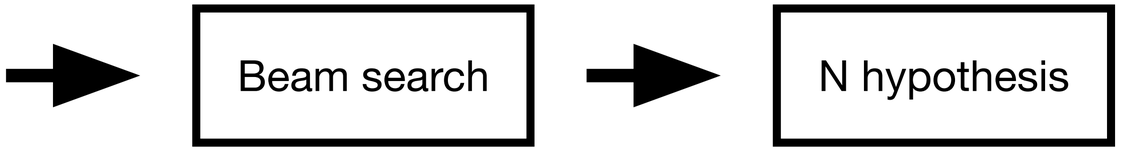


Beam search

Адаптация может происходить на уровне
пересчёта скоров в лучах
и на уровне **реранкинга итоговых последовательной**

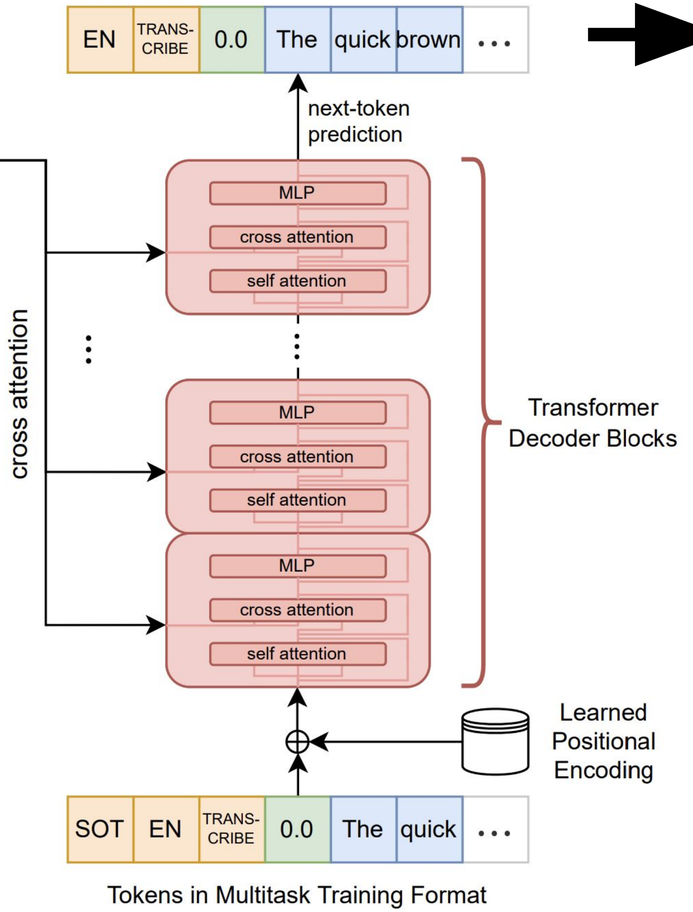
Методы адаптации

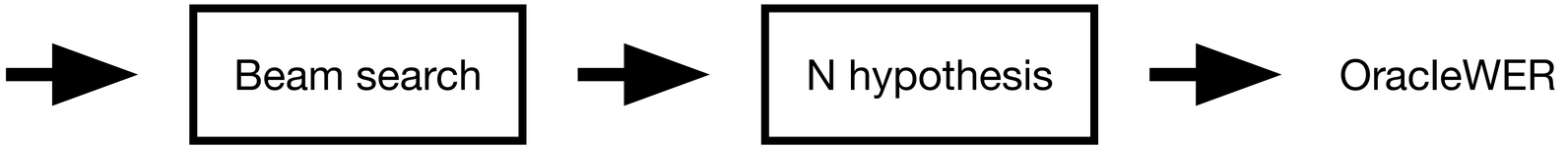




Адаптация может происходить на уровне
пересчёта скоров в лучах
и на уровне **реранкинга итоговых последовательной**

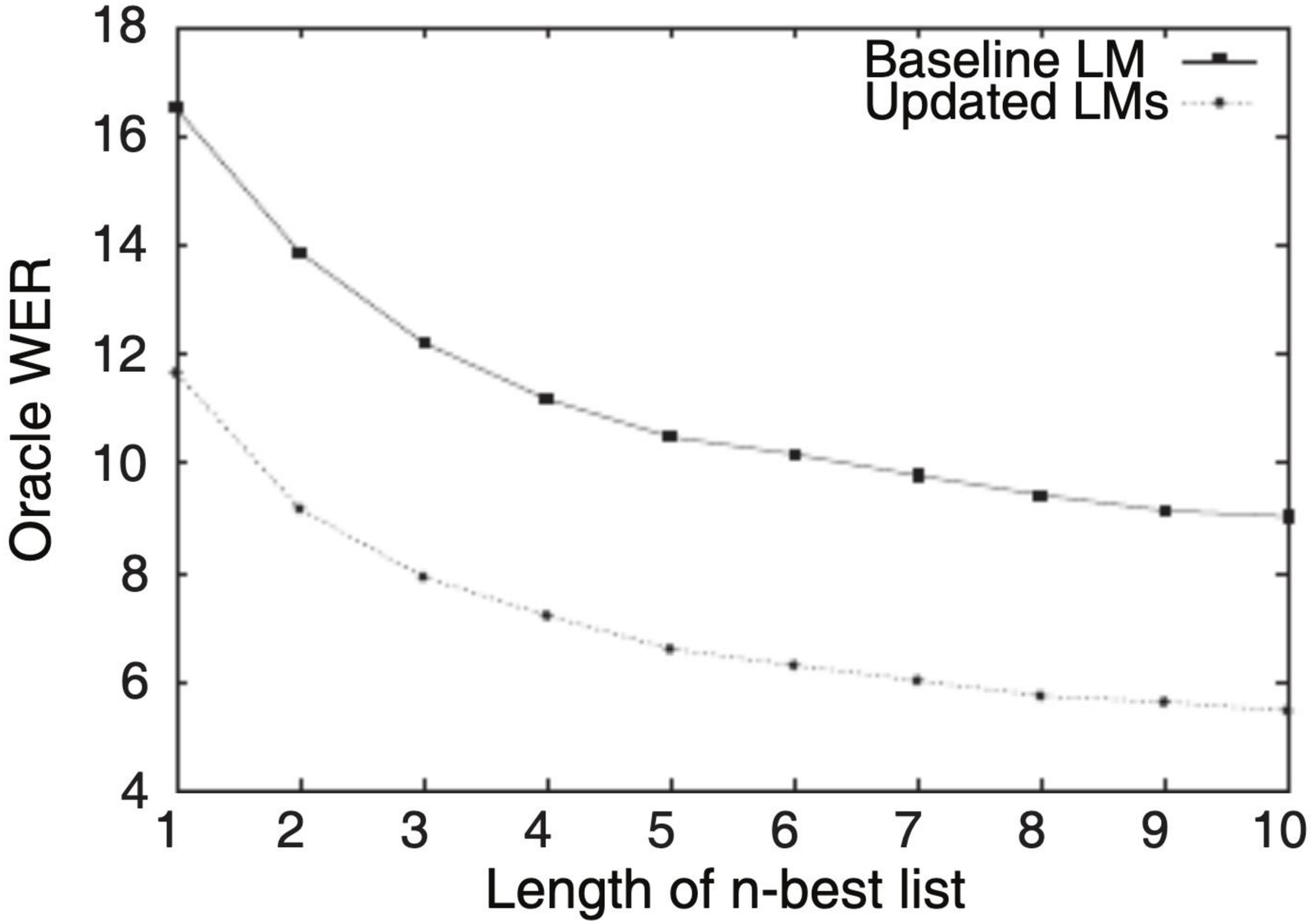
Методы адаптации



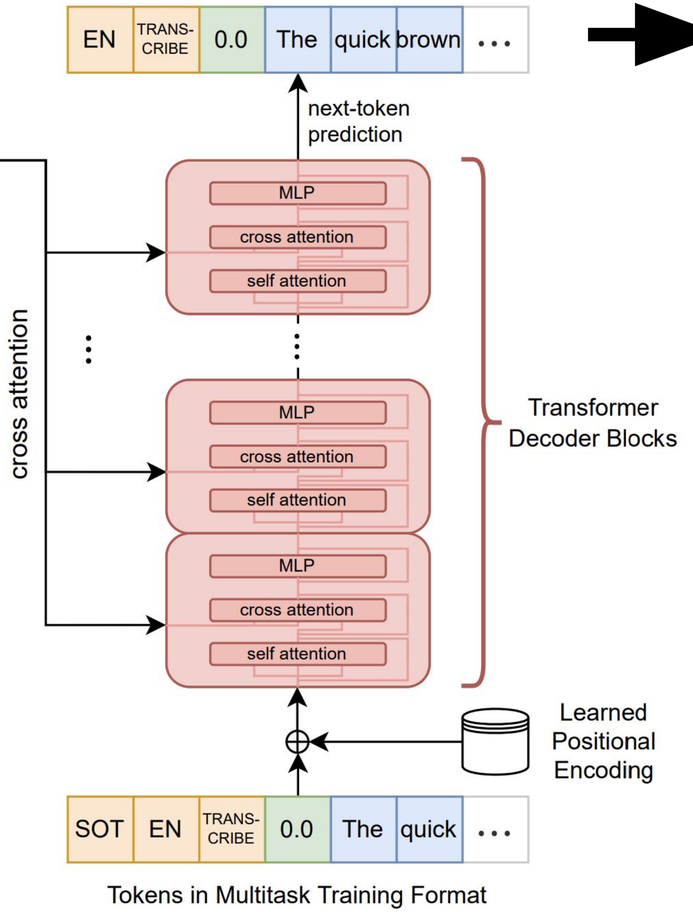


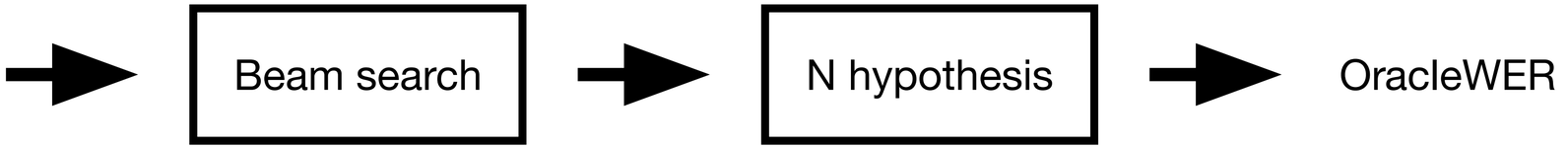
Адаптация может происходить на уровне
пересчёта скоров в лучах
и на уровне **реранкинга итоговых последовательной**

Методы адаптации



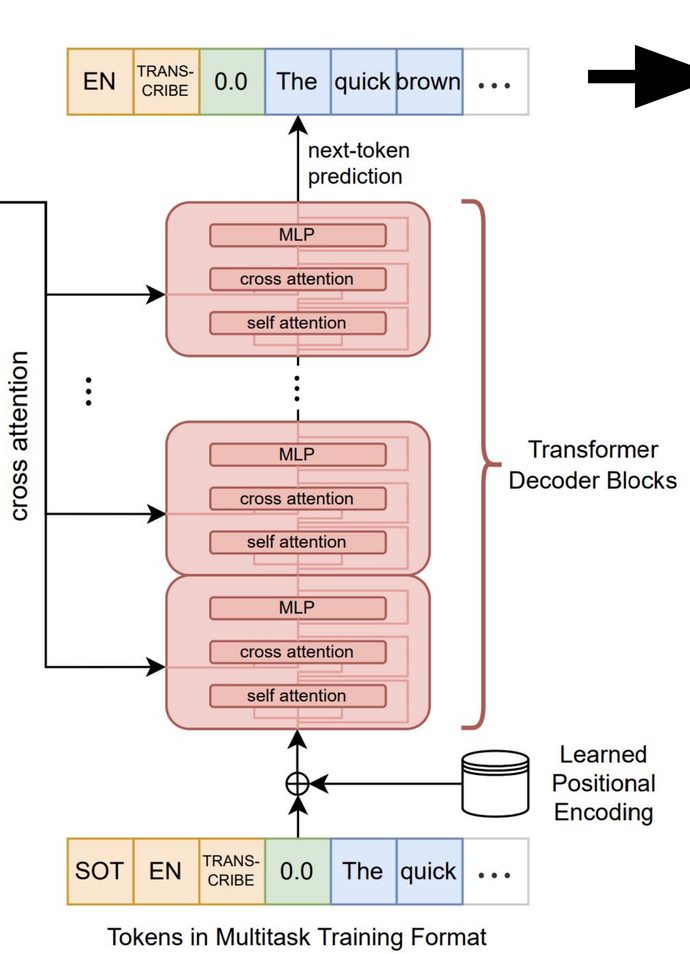
Методы адаптации

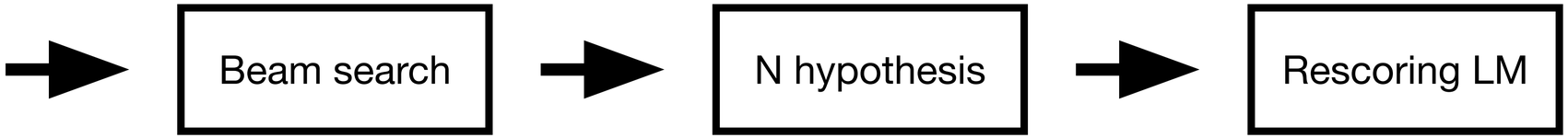




Адаптация может происходить на уровне
пересчёта скоров в лучах
и на уровне **реранкинга итоговых последовательной**

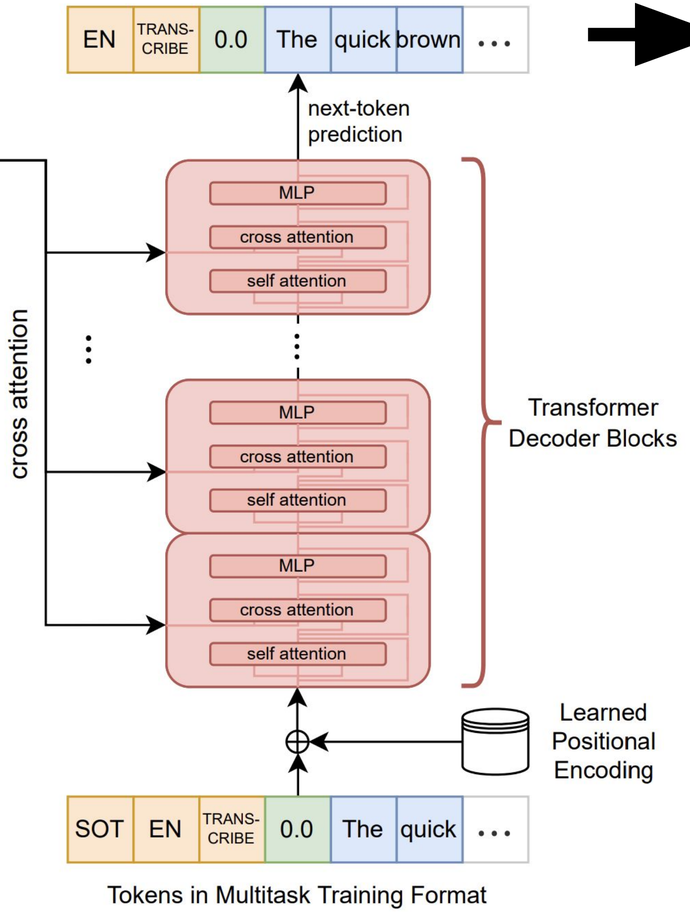
Методы адаптации





Адаптация может происходить на уровне
пересчёта скоров в лучах
и на уровне **реранкинга итоговых последовательной**

Методы адаптации

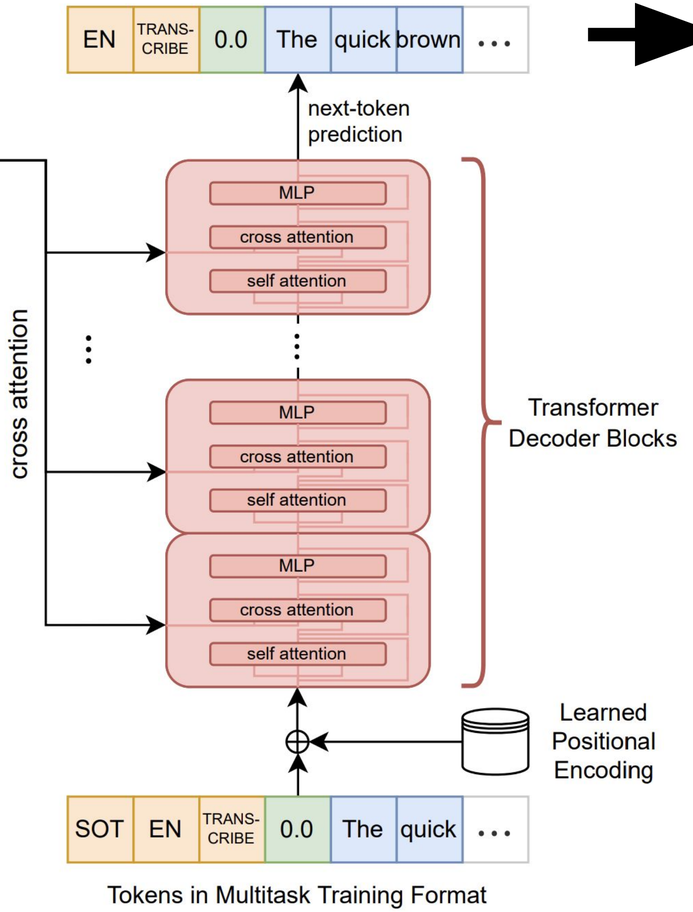


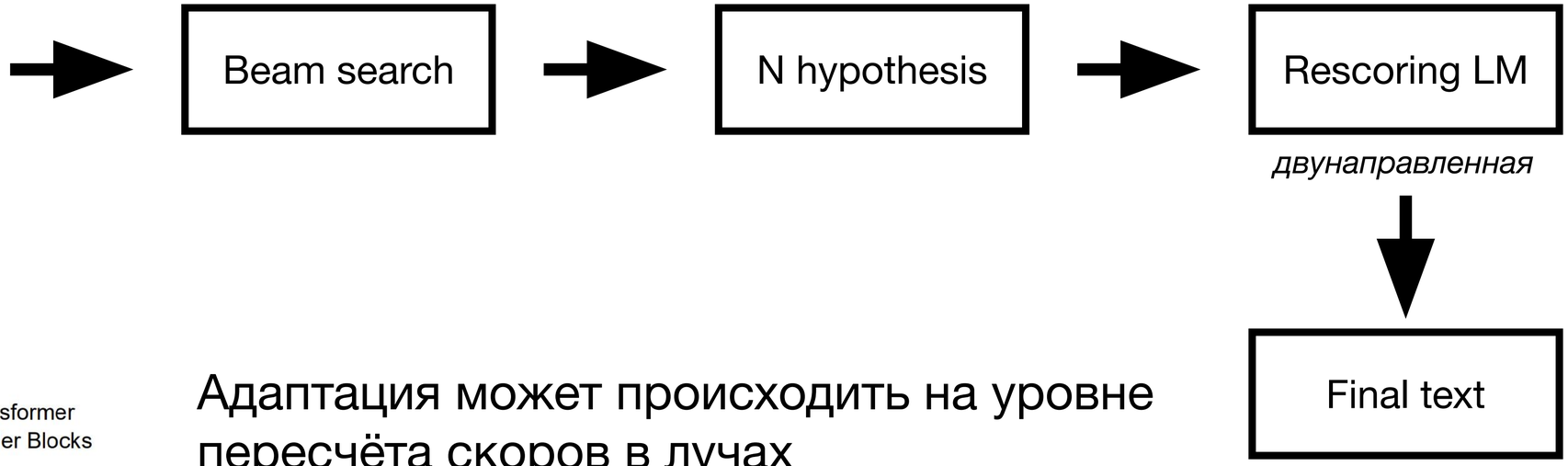
Beam search → N hypothesis → Rescoring LM

двунаправленная

Адаптация может происходить на уровне пересчёта скоров в лучах
и на уровне **реранкинга итоговых последовательной**

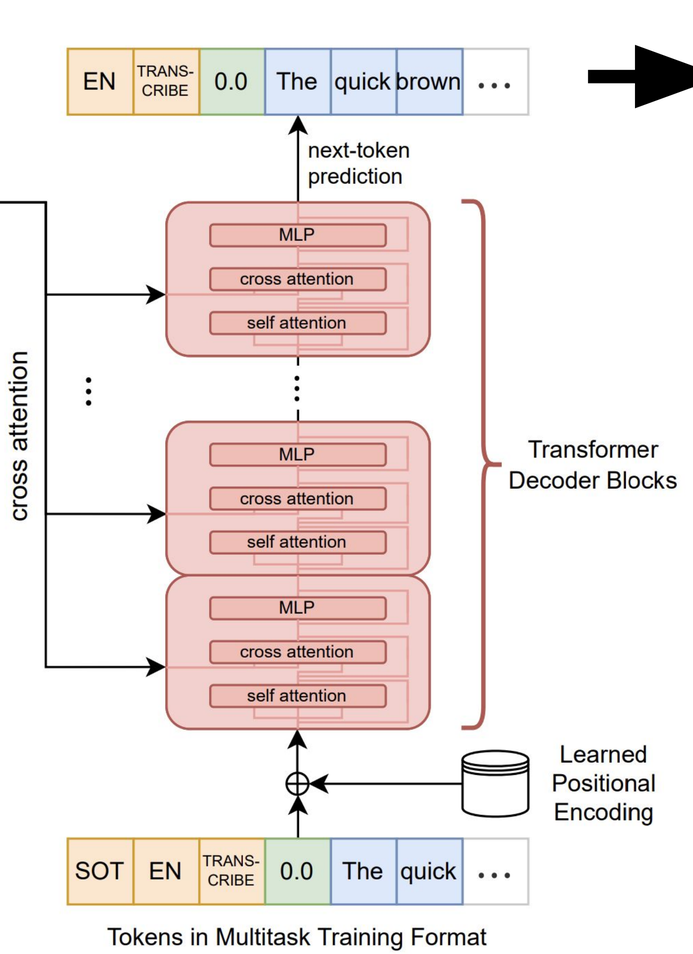
Методы адаптации





Адаптация может происходить на уровне пересчёта скоров в лучах
и на уровне **реранкинга итоговых последовательной**

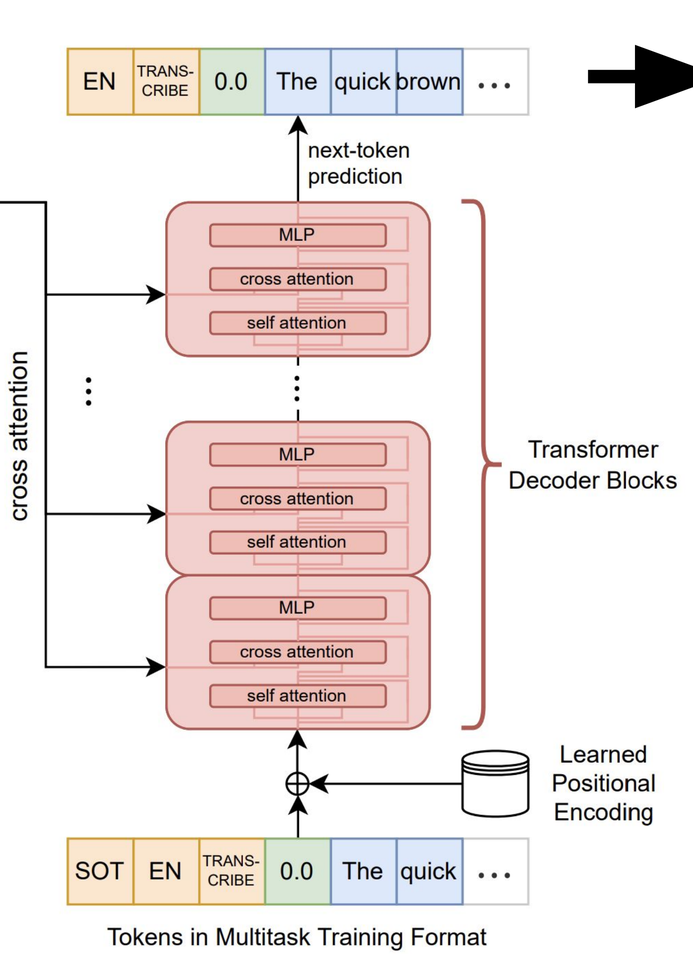
Методы адаптации



Beam search

Адаптация может происходить на уровне
пересчёта скоров в лучах
и на уровне реранкинга итоговых последовательной

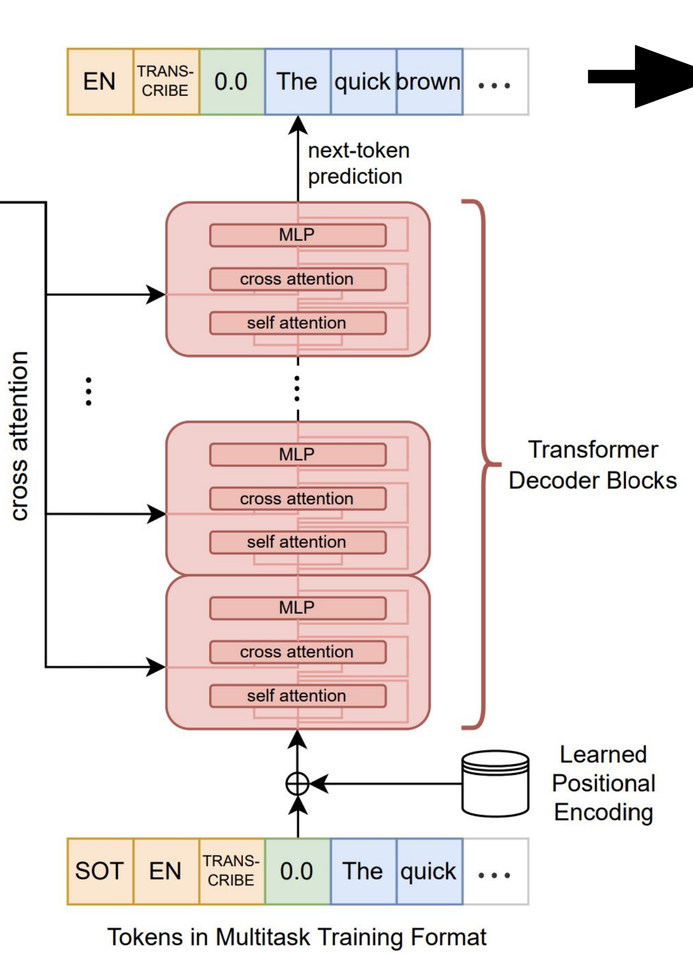
Методы адаптации

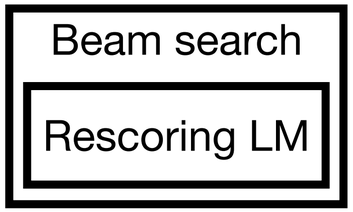


Beam search

Адаптация может происходить на уровне
пересчёта скоров в лучах
и на уровне реранкинга итоговых последовательной

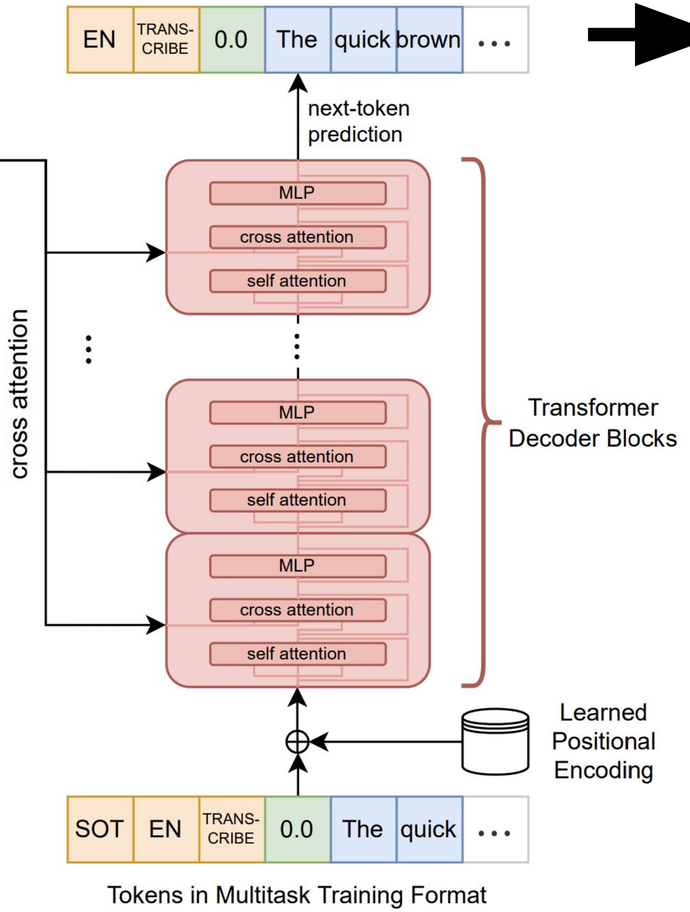
Методы адаптации

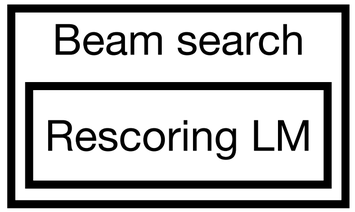




Адаптация может происходить на уровне
пересчёта скоров в лучах
и на уровне реранкинга итоговых последовательной

Методы адаптации

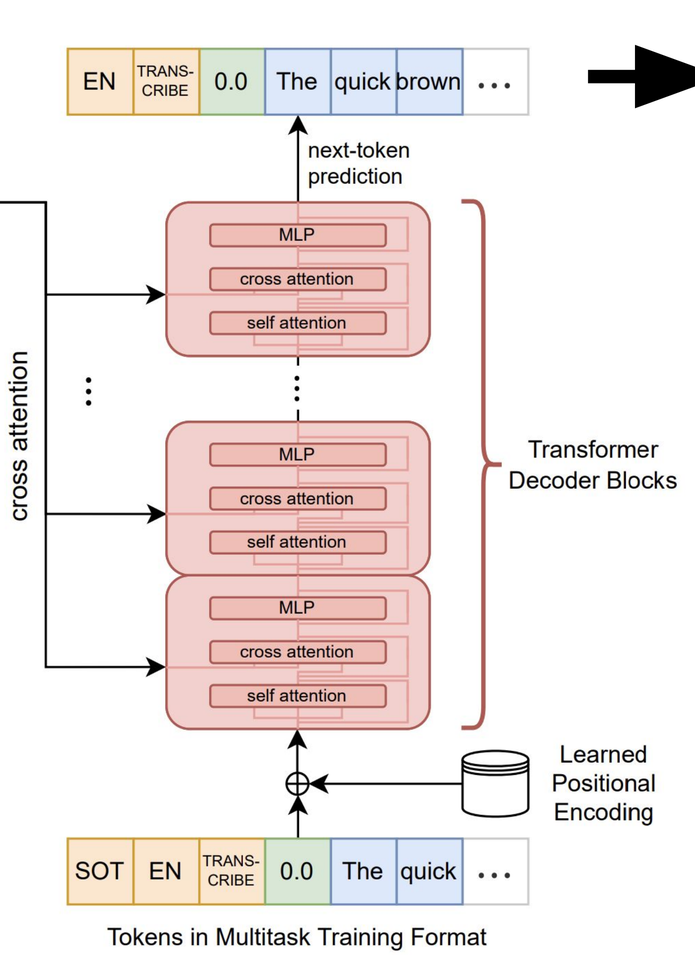


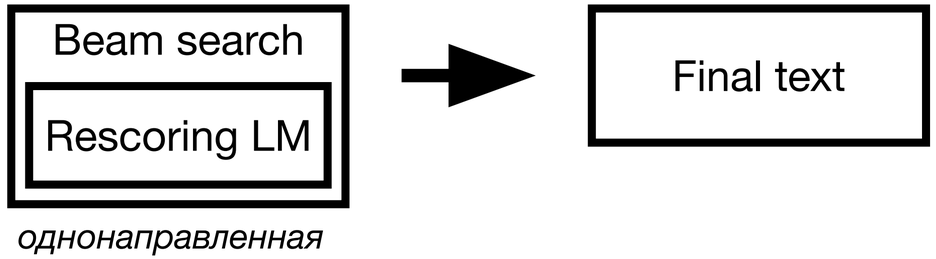


однонаправленная

Адаптация может происходить на уровне
пересчёта скоров в лучах
и на уровне реранкинга итоговых последовательной

Методы адаптации

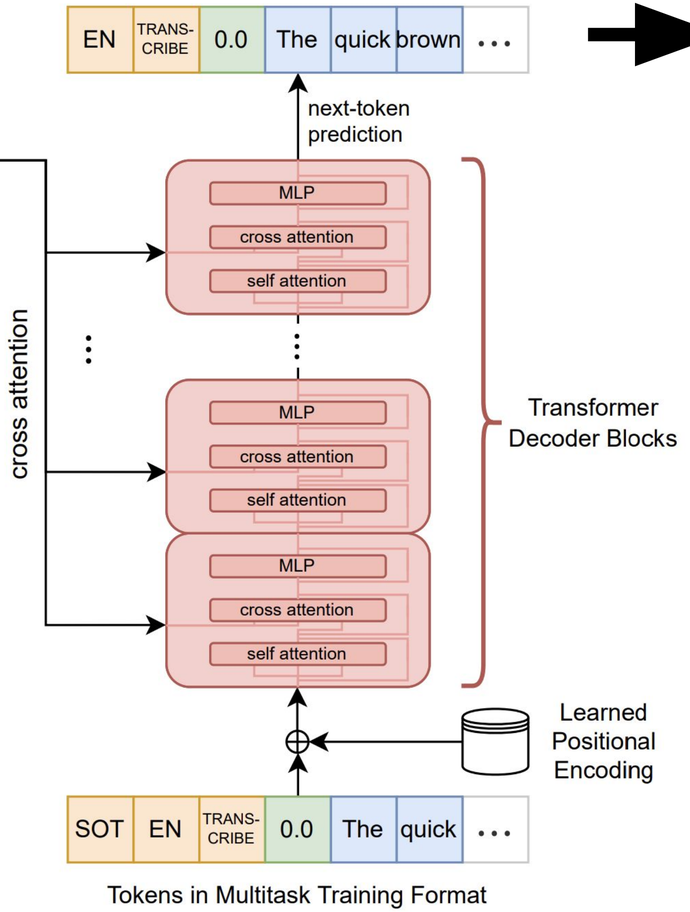


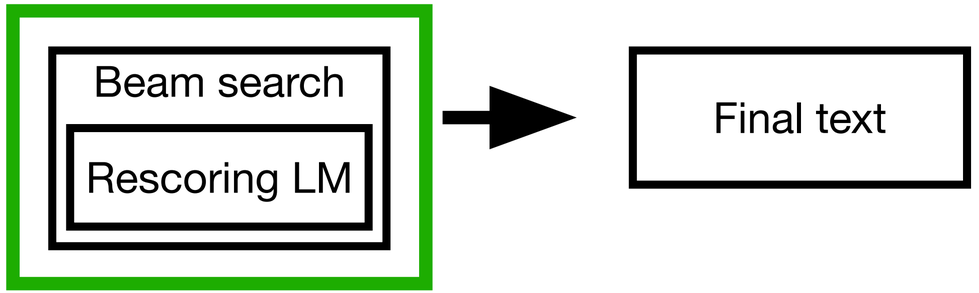


однонаправленная

Адаптация может происходить на уровне
пересчёта скоров в лучах
и на уровне реранкинга итоговых последовательной

Методы адаптации





shallow fusion

Адаптация может происходить на уровне
пересчёта скоров в лучах
и на уровне реранкинга итоговых последовательной

Shallow fusion

Функция смешения

Listen, Attend, Spell: 2015

$$s(\mathbf{y}|\mathbf{x}) = \frac{\log P(\mathbf{y}|\mathbf{x})}{|\mathbf{y}|_c} + \lambda \log P_{\mathrm{LM}}(\mathbf{y})$$

Функция смешения

Listen, Attend, Spell: 2015

$$s(\mathbf{y}|\mathbf{x}) = \frac{\log P(\mathbf{y}|\mathbf{x})}{|\mathbf{y}|_c} + \lambda \log P_{\mathrm{LM}}(\mathbf{y})$$

Функция смешения

Listen, Attend, Spell: 2015

$$s(\mathbf{y}|\mathbf{x}) = \frac{\log P(\mathbf{y}|\mathbf{x})}{|\mathbf{y}|_c} + \lambda \log P_{\mathrm{LM}}(\mathbf{y})$$

Функция смешения

Listen, Attend, Spell: 2015

$$s(\mathbf{y}|\mathbf{x}) = \frac{\log P(\mathbf{y}|\mathbf{x})}{|\mathbf{y}|_c} + \lambda \log P_{\mathrm{LM}}(\mathbf{y})$$

Функция смешения

Listen, Attend, Spell: 2015

$$s(\mathbf{y}|\mathbf{x}) = \frac{\log P(\mathbf{y}|\mathbf{x})}{|\mathbf{y}|_c} + \lambda \log P_{\mathrm{LM}}(\mathbf{y})$$

Функция смешения

Listen, Attend, Spell: 2015

$$s(\mathbf{y}|\mathbf{x}) = \frac{\log P(\mathbf{y}|\mathbf{x})}{|\mathbf{y}|_c} + \lambda \log P_{\mathrm{LM}}(\mathbf{y})$$

Функция смешения

Listen, Attend, Spell: 2015

$$s(\mathbf{y}|\mathbf{x}) = \frac{\log P(\mathbf{y}|\mathbf{x})}{|\mathbf{y}|_c} + \lambda \log P_{\mathrm{LM}}(\mathbf{y})$$

Функция смешения

Listen, Attend, Spell: 2015

$$s(\mathbf{y}|\mathbf{x}) = \frac{\log P(\mathbf{y}|\mathbf{x})}{|\mathbf{y}|_c} + \lambda \log P_{\mathrm{LM}}(\mathbf{y})$$

Функция смешения

Google: 2017

$$\mathbf{y}^* = \arg\max_y \log p(y|x) + \lambda \log p_{LM}(y) + \gamma c(x, y)$$

Функция смешения

Google: 2017

$$\mathbf{y}^* = \boxed{\arg\max_y \log p(y|x)} + \lambda \log p_{LM}(y) + \gamma c(x, y)$$

Функция смешения

Google: 2017

$$\mathbf{y}^* = \arg\max_y \log p(y|x) + \lambda \log p_{LM}(y) + \gamma c(x, y)$$

Функция смешения

Google: 2017

$$\mathbf{y}^* = \arg\max_y \log p(y|x) + \lambda \log p_{LM}(y) + \gamma c(x, y)$$

Функция смешения

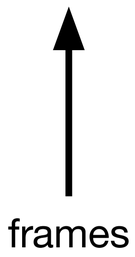
Google: 2017

$$c(x, y) = \sum_j \log(\min(\sum_i a_{i,j}, 0.5))$$

Функция смешения

Google: 2017

$$c(x,y) = \sum_{j} \log(\min(\sum_{i} a_{i,j}, 0.5))$$



frames

Функция смешения

Google: 2017

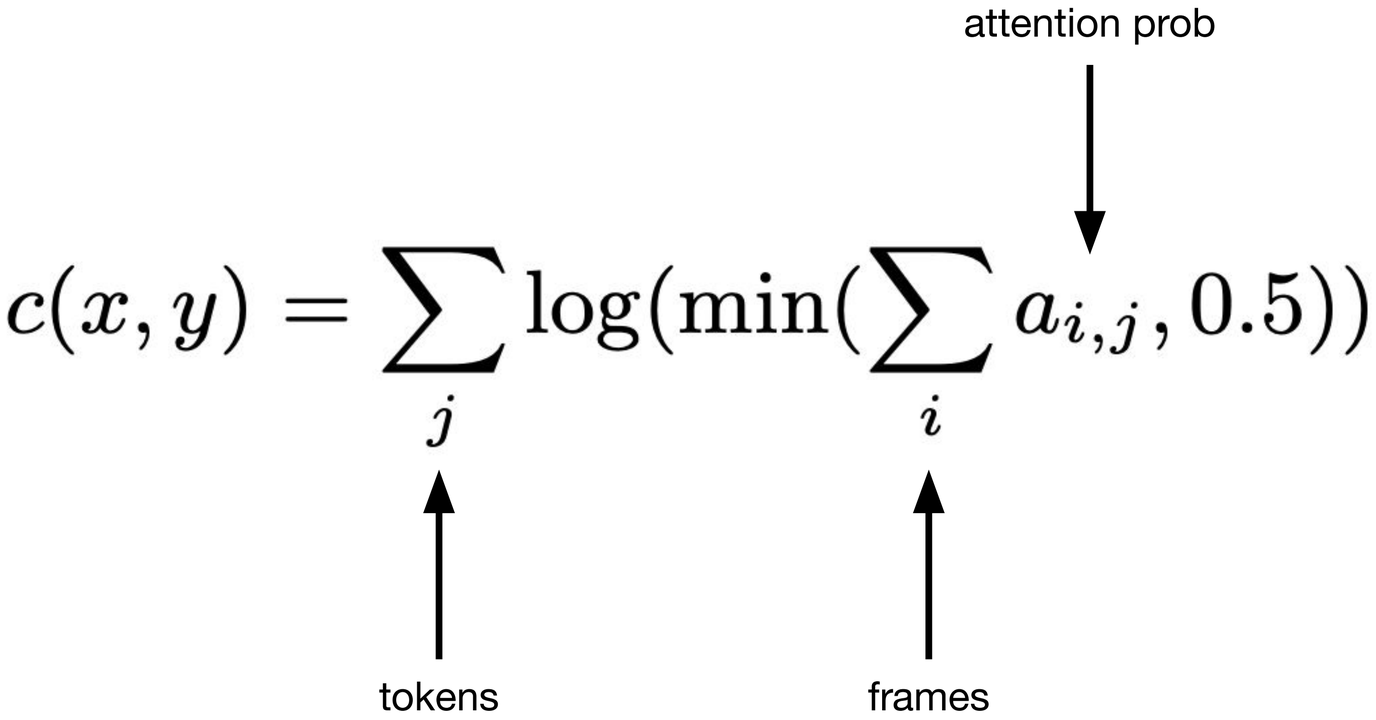
$$c(x,y) = \sum_{j} \log(\min(\sum_{i} a_{i,j}, 0.5))$$

tokens

frames

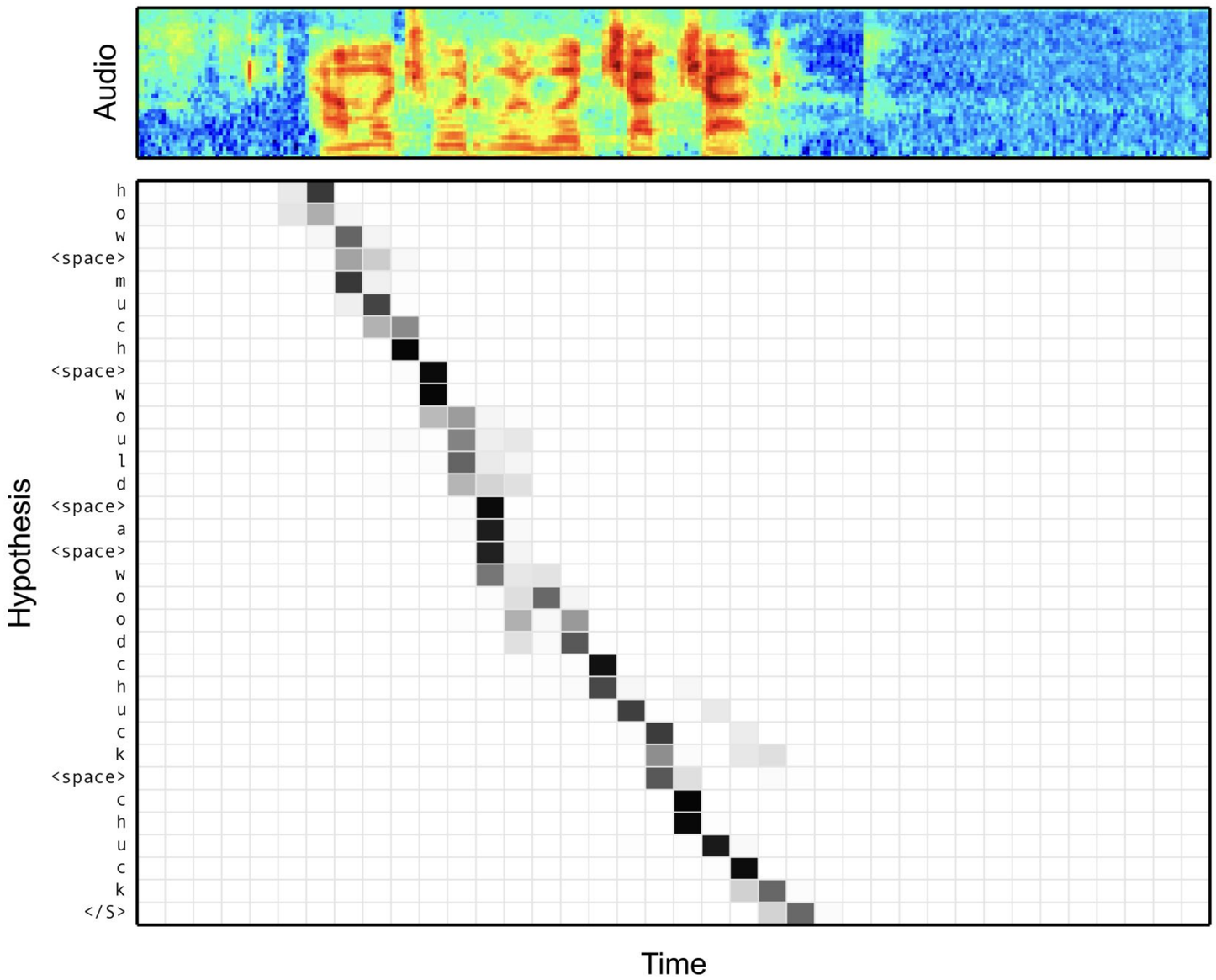
Функция смешения

Google: 2017



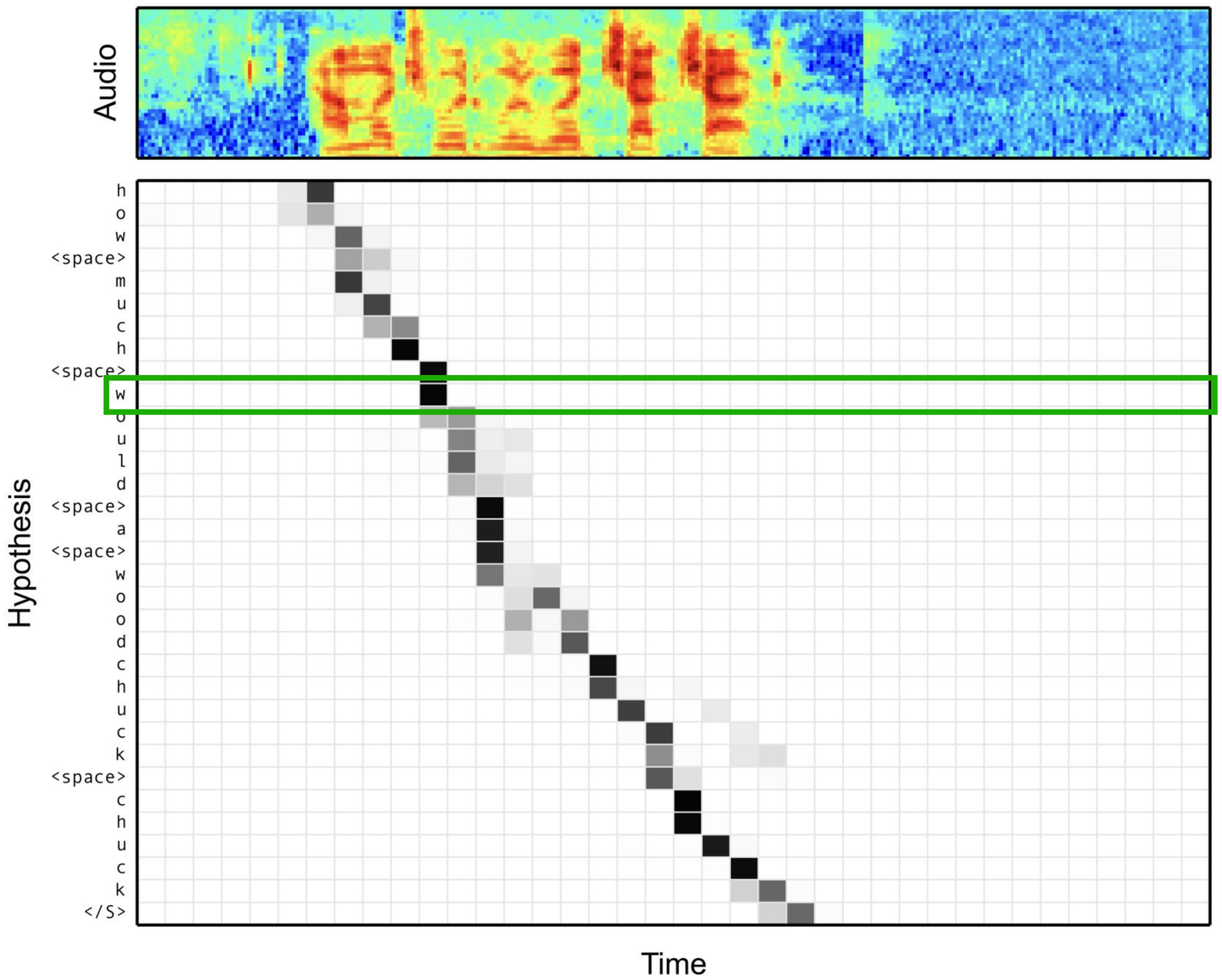
Функция смешения

Google: 2017



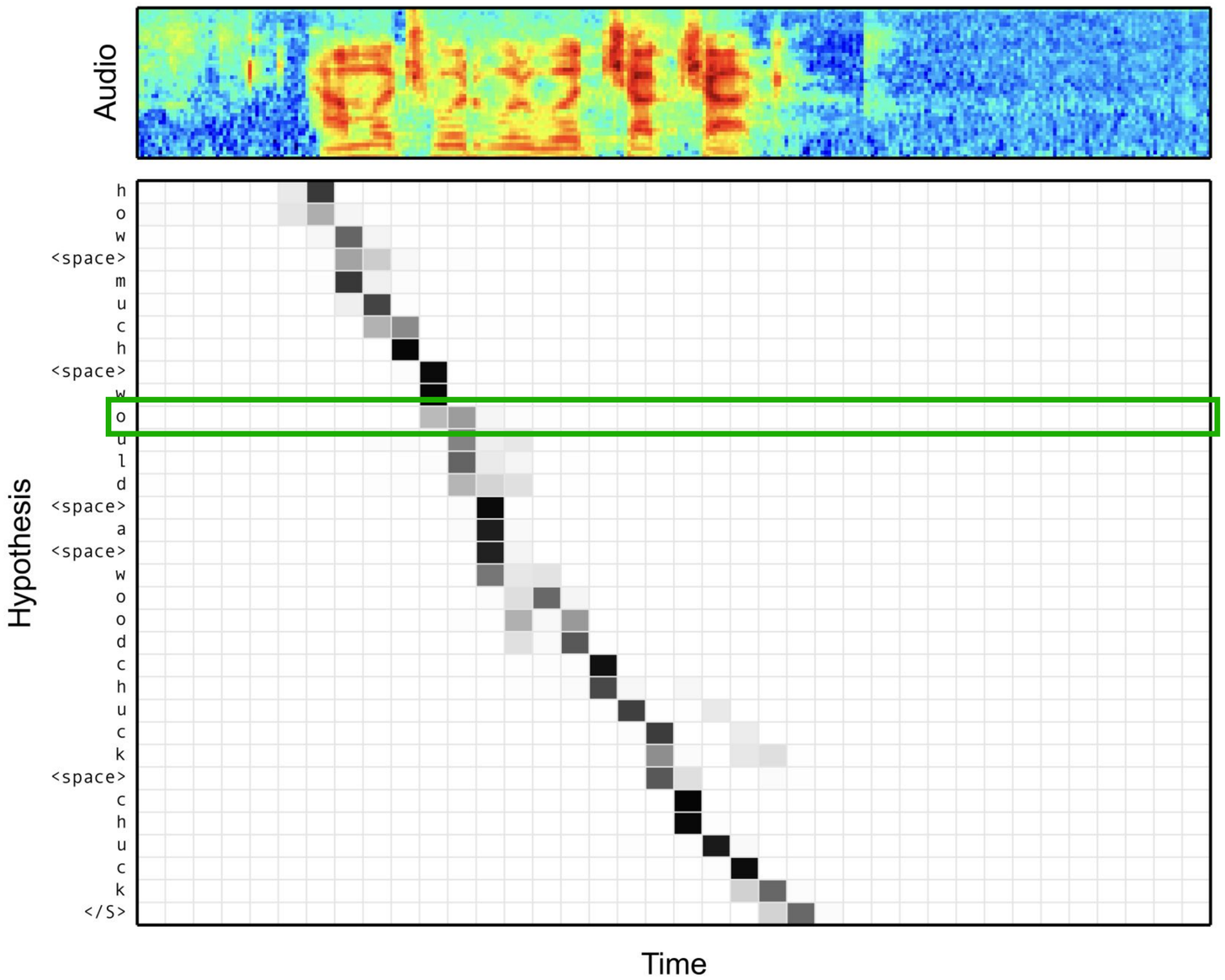
Функция смешения

Google: 2017



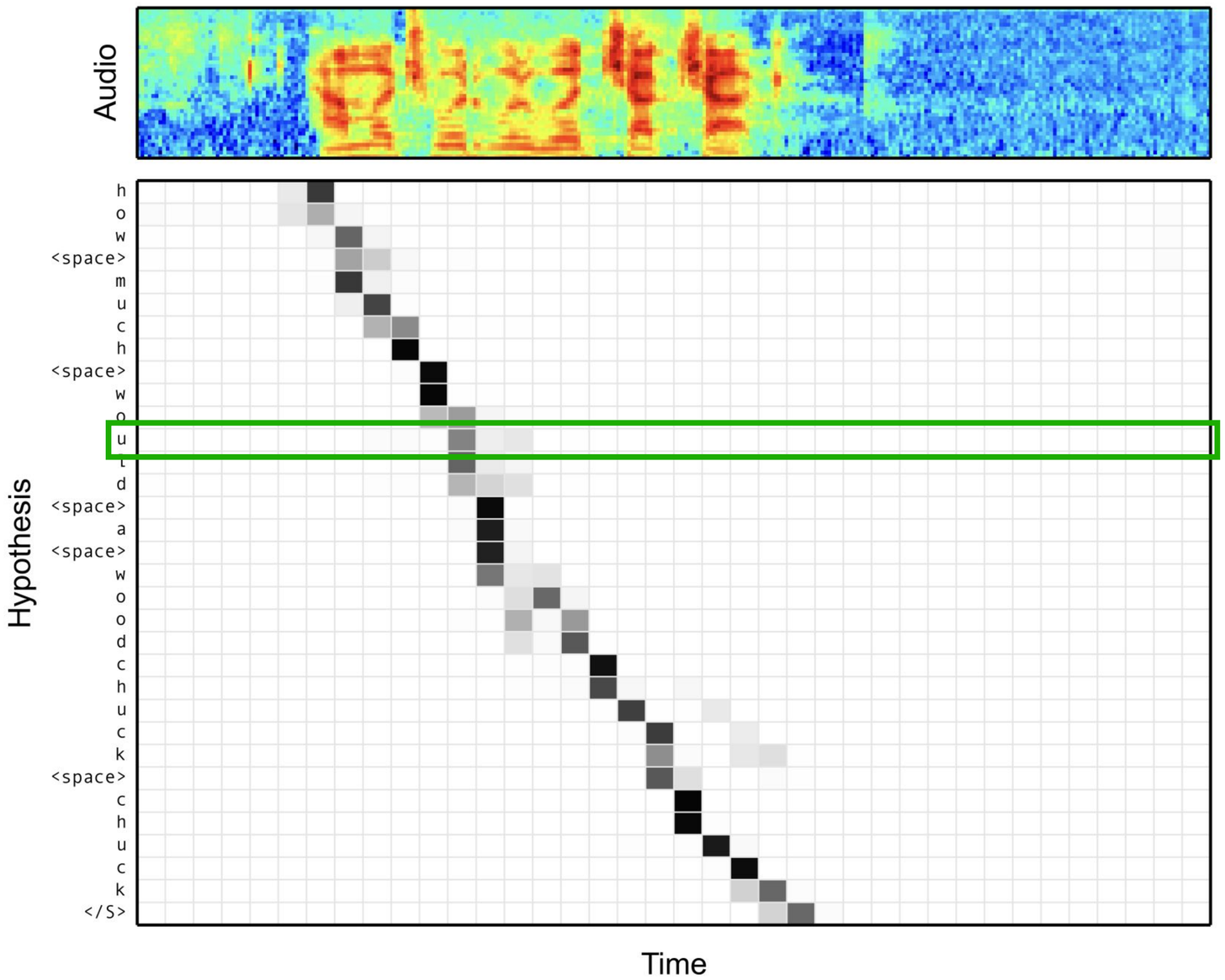
Функция смешения

Google: 2017



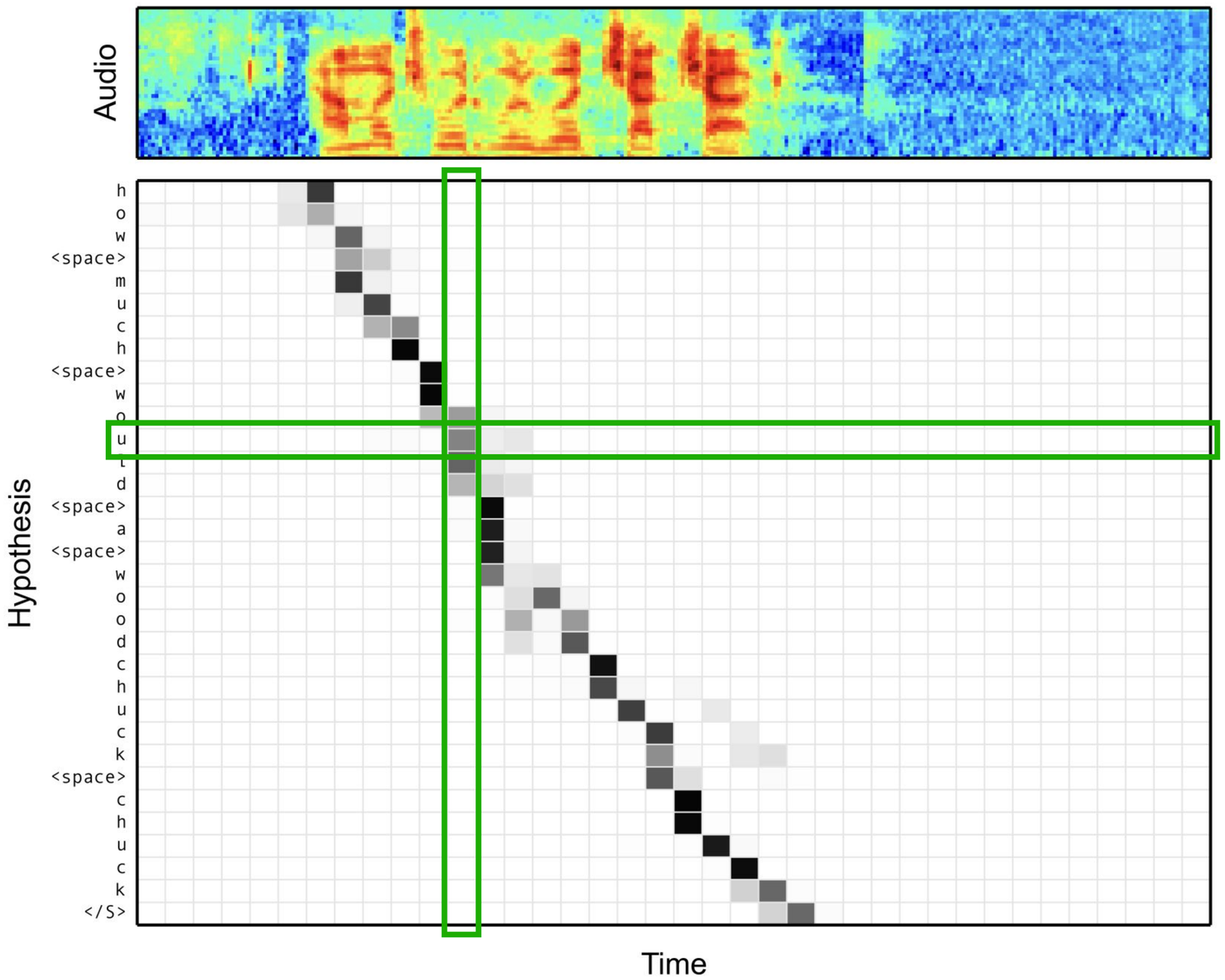
Функция смешения

Google: 2017



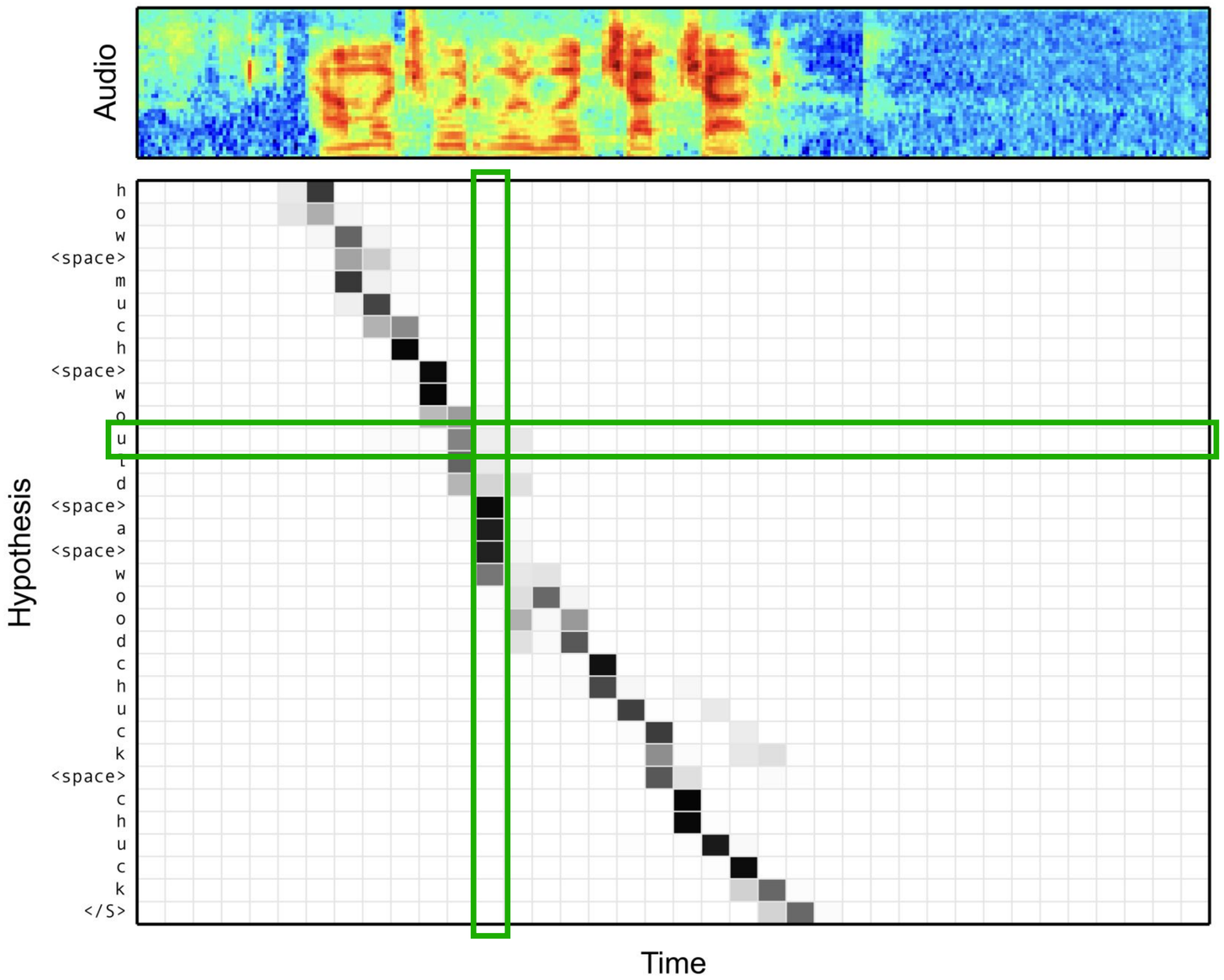
Функция смешения

Google: 2017



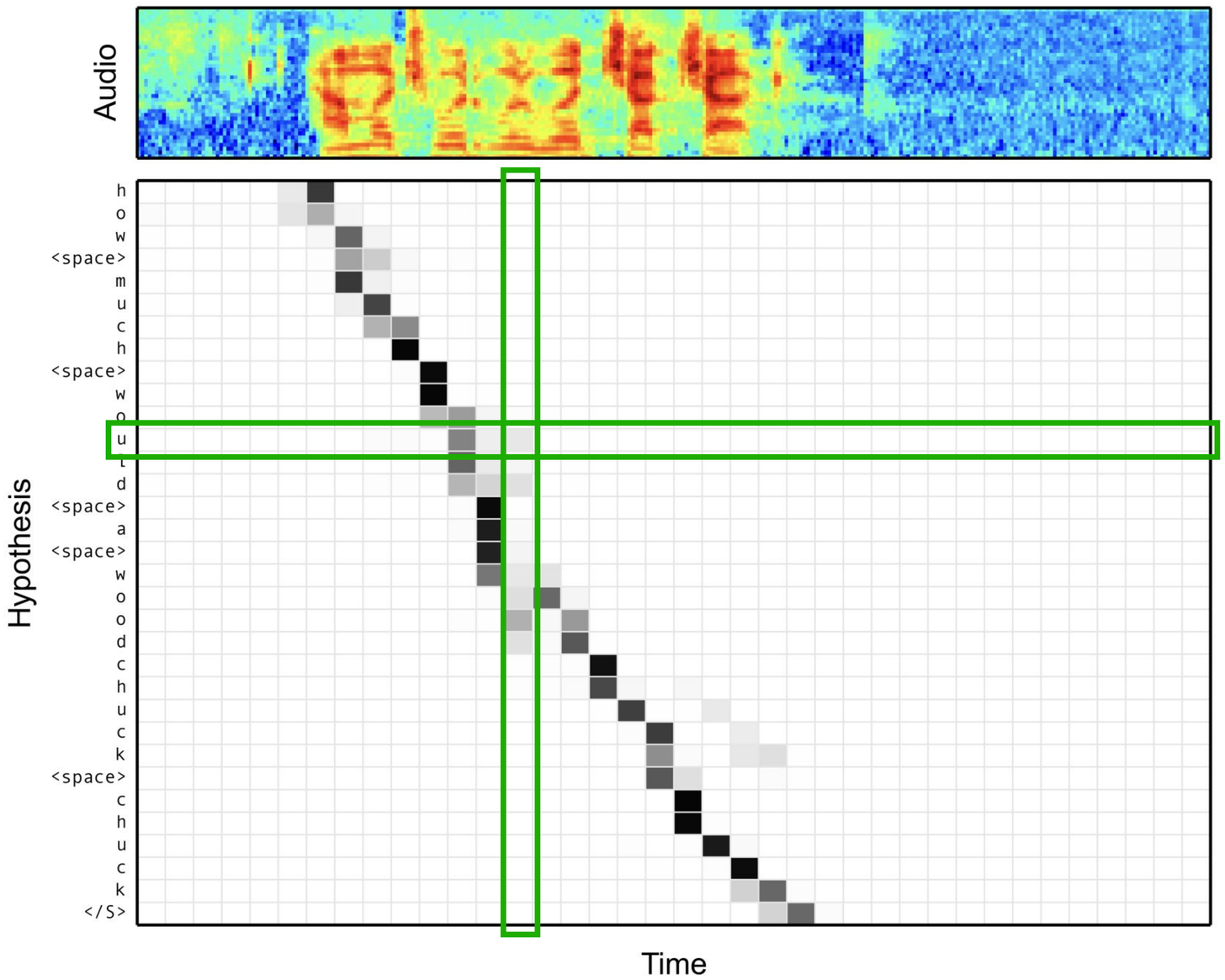
Функция смешения

Google: 2017



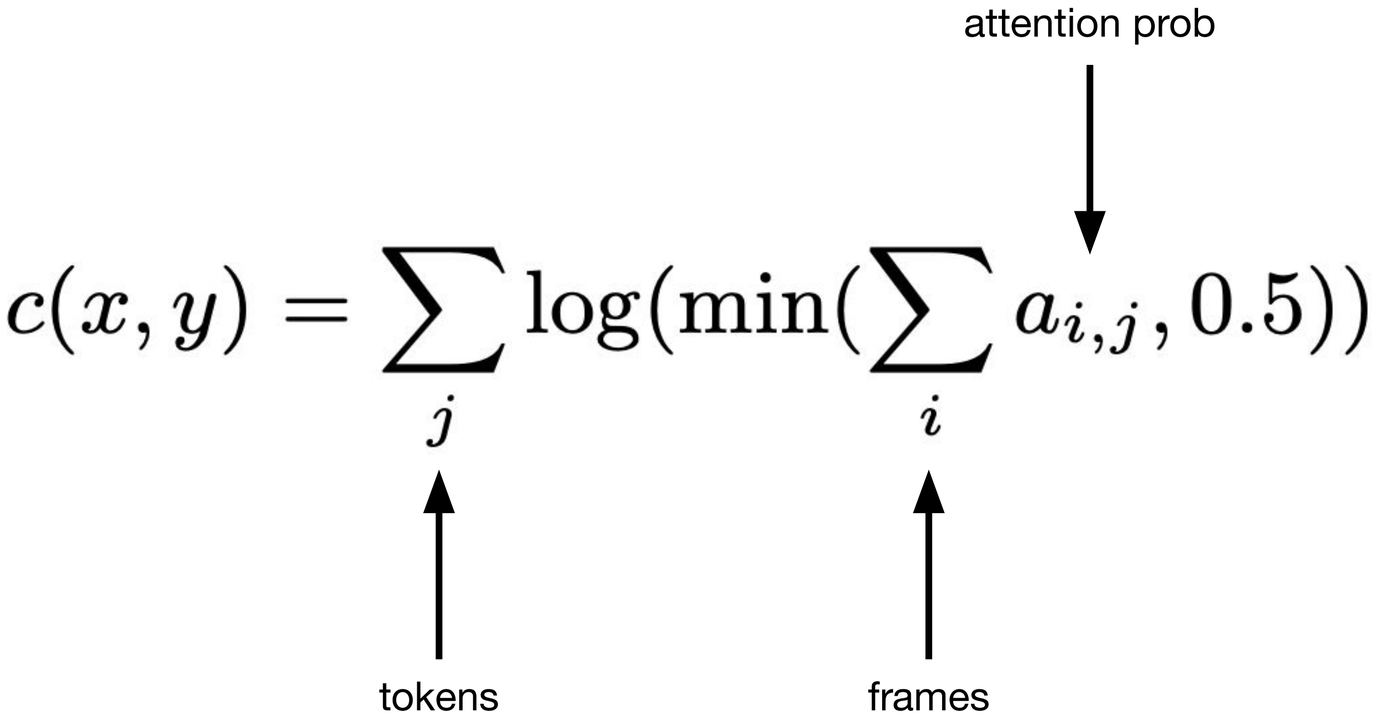
Функция смешения

Google: 2017



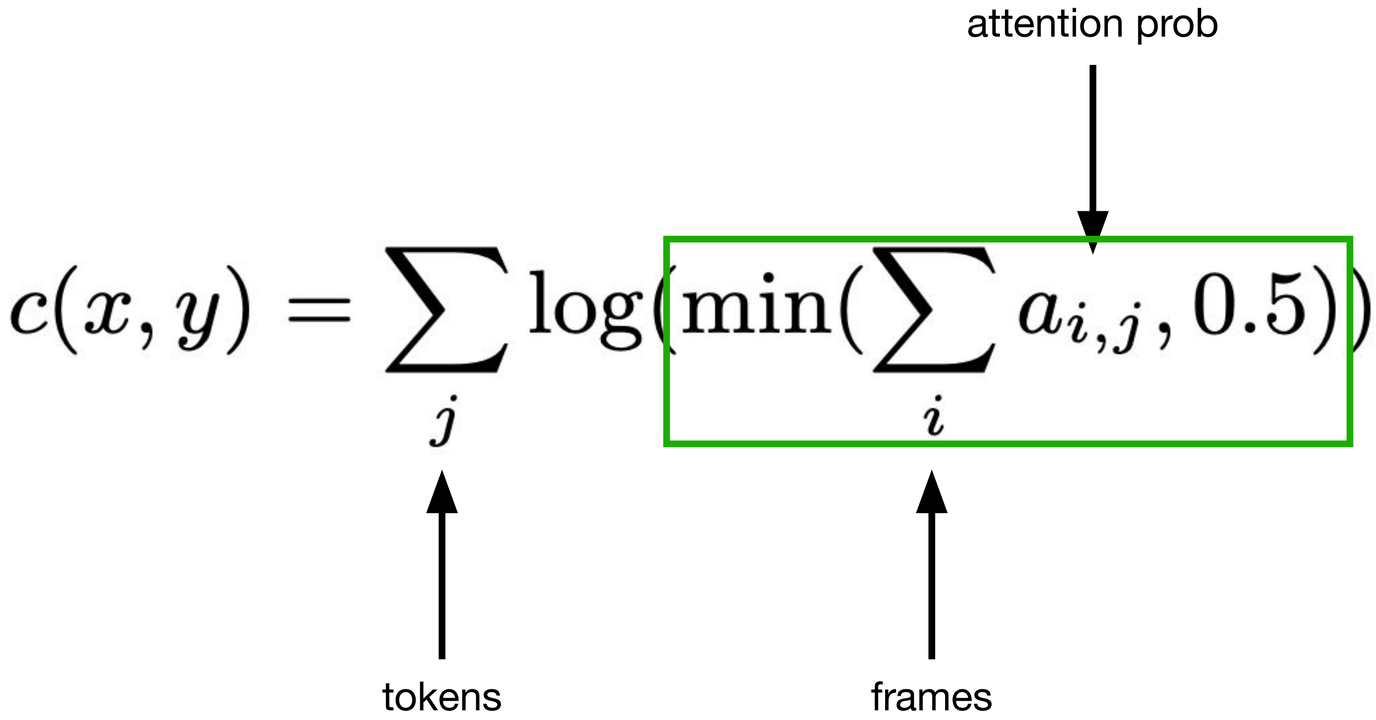
Функция смешения

Google: 2017



Функция смешения

Google: 2017



Функция смешения

Google: 2017

$$\mathbf{y}^* = \arg\max_y \log p(y|x) + \lambda \log p_{LM}(y) + \gamma c(x, y)$$

Функция смешения

Google: 2017

$$\mathbf{y}^* = \arg\max_y \log p(y|x) + \lambda \log p_{LM}(y) + \gamma c(x, y)$$

Функция смешения

Google: 2018

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} \log P(\mathbf{y}|\mathbf{x}) + \lambda \log P_{LM}(\mathbf{y}) + \gamma \mathrm{len}(\mathbf{y})$$

Функция смешения

Google: 2018

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} \log P(\mathbf{y}|\mathbf{x}) + \lambda \log P_{LM}(\mathbf{y}) + \gamma \mathrm{len}(\mathbf{y})$$

Функция смешения

SPMI: 2022

$$\hat{Y} = \arg\max_{Y} [\log P_{\text{RNN-T}}(Y|X) + \lambda_0 \log P_{\text{ILM}}(Y) + \lambda_1 \log P_{\text{ELM}}(Y) + \beta|Y|]$$

Функция смешения

SPMI: 2022

Internal LM

$$\hat{Y} = \arg\max_{Y} [\log P_{\text{RNN-T}}(Y|X) + \lambda_0 \log P_{\text{ILM}}(Y) + \lambda_1 \log P_{\text{ELM}}(Y) + \beta|Y|]$$

Функция смешения

SPMI: 2022

Internal LM

$$\hat{Y} = \arg\max_{Y} [\log P_{\text{RNN-T}}(Y|X) + \lambda_0 \log P_{\text{ILM}}(Y)$$

$$+\lambda_1 \log P_{\text{ELM}}(Y) + \beta|Y|]$$

external LM

Функция смешения

SPMI: 2022

Internal LM

$$\hat{Y} = \arg\max_{Y} \left[\log P_{\text{RNN-T}}(Y|X) + \lambda_0 \log P_{\text{ILM}}(Y) \right.$$

$$\left. + \lambda_1 \log P_{\text{ELM}}(Y) + \beta |Y| \right]$$

external LM

регуляризация на длину

Модели адаптации

Лингвистическая модель (LM)

Лингвистическая модель (LM)

- Оценивает лог-вероятность луча на каждом шаге

Лингвистическая модель (LM)

- Оценивает лог-вероятность луча на каждом шаге
- Токенизация не обязательно должна быть как в основной модели

Лингвистическая модель (LM)

- Оценивает лог-вероятность луча на каждом шаге
- Токенизация не обязательно должна быть как в основной модели
- Модель однонаправленная, учитывается только предыдущий контекст

Лингвистическая модель (LM)

- Оценивает лог-вероятность луча на каждом шаге
- Токенизация не обязательно должна быть как в основной модели
- Модель однонаправленная, учитывается только предыдущий контекст
- Для быстрого инференса обучают преимущественно маленькие модели

Типы лингвистических моделей

Аспект	N-Gram	RNN	Transformer

Типы лингвистических моделей

Аспект	**N-Gram**	**RNN**	**Transformer**
Обучение	Быстро	Медленно	Медленно

Типы лингвистических моделей

Аспект	**N-Gram**	**RNN**	**Transformer**
Обучение	Быстро	Медленно	Медленно
Инференс	Мгновенно	Медленно	Медленно

Типы лингвистических моделей

Аспект	**N-Gram**	**RNN**	**Transformer**
Обучение	Быстро	Медленно	Медленно
Инференс	Мгновенно	Медленно	Медленно
Видеопамять	Не требуется	Фиксирована (веса + hidden)	Растёт от длины контекста

Типы лингвистических моделей

Аспект	**N-Gram**	**RNN**	**Transformer**
Обучение	Быстро	Медленно	Медленно
Инференс	Мгновенно	Медленно	Медленно
Видеопамять	Не требуется	Фиксирована (веса + hidden)	Растёт от длины контекста
Качество	Среднее	Высокое	Высокое

Типы лингвистических моделей

Аспект	**N-Gram**	**RNN**	**Transformer**
Обучение	Быстро	Медленно	Медленно
Инференс	Мгновенно	Медленно	Медленно
Видеопамять	Не требуется	Фиксирована (веса + hidden)	Растёт от длины контекста
Качество	Среднее	Высокое	Высокое

Типы лингвистических моделей

Аспект	**N-Gram**	**RNN**	**Transformer**
Обучение	Быстро	Медленно	Медленно
Инференс	Мгновенно	Медленно	Медленно
Видеопамять	Не требуется	Фиксирована (веса + hidden)	Растёт от длины контекста
Качество	Среднее	Высокое	Высокое

N-Gram LM

N-Gram LM

$$P(w_i \mid w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})}$$

N-Gram LM

<s> Yes, I am watching the TV right now </s>
<s> The laptop I gave you is very good </s>
<s> I am very happy to be here </s>

$$P(w_i \mid w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})}$$

N-Gram LM

<s> Yes, I am watching the TV right now </s>
<s> The laptop I gave you is very good </s>
<s> I am very happy to be here </s>

$$P(w_i \mid w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})}$$

P(am | I) = count("**I am**") / count("**I**") = 2 / 3 = 0.667

N-Gram LM

<s> Yes, I am watching the TV right now </s>
<s> The laptop I gave you is very good </s>
<s> I am very happy to be here </s>

$$P(w_i \mid w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})}$$

P(am | I) = count("**I am**") / count("**I**") = 2 / 3 = 0.667

P(I | <s>) = count("**<s> I**") / count("**<s>**") = 1 / 3 = 0.333

N-Gram LM

<s> Yes, I am watching the TV right now </s>
<s> The laptop I gave you is very good </s>
<s> I am very happy to be here </s>

$$P(w_i \mid w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})}$$

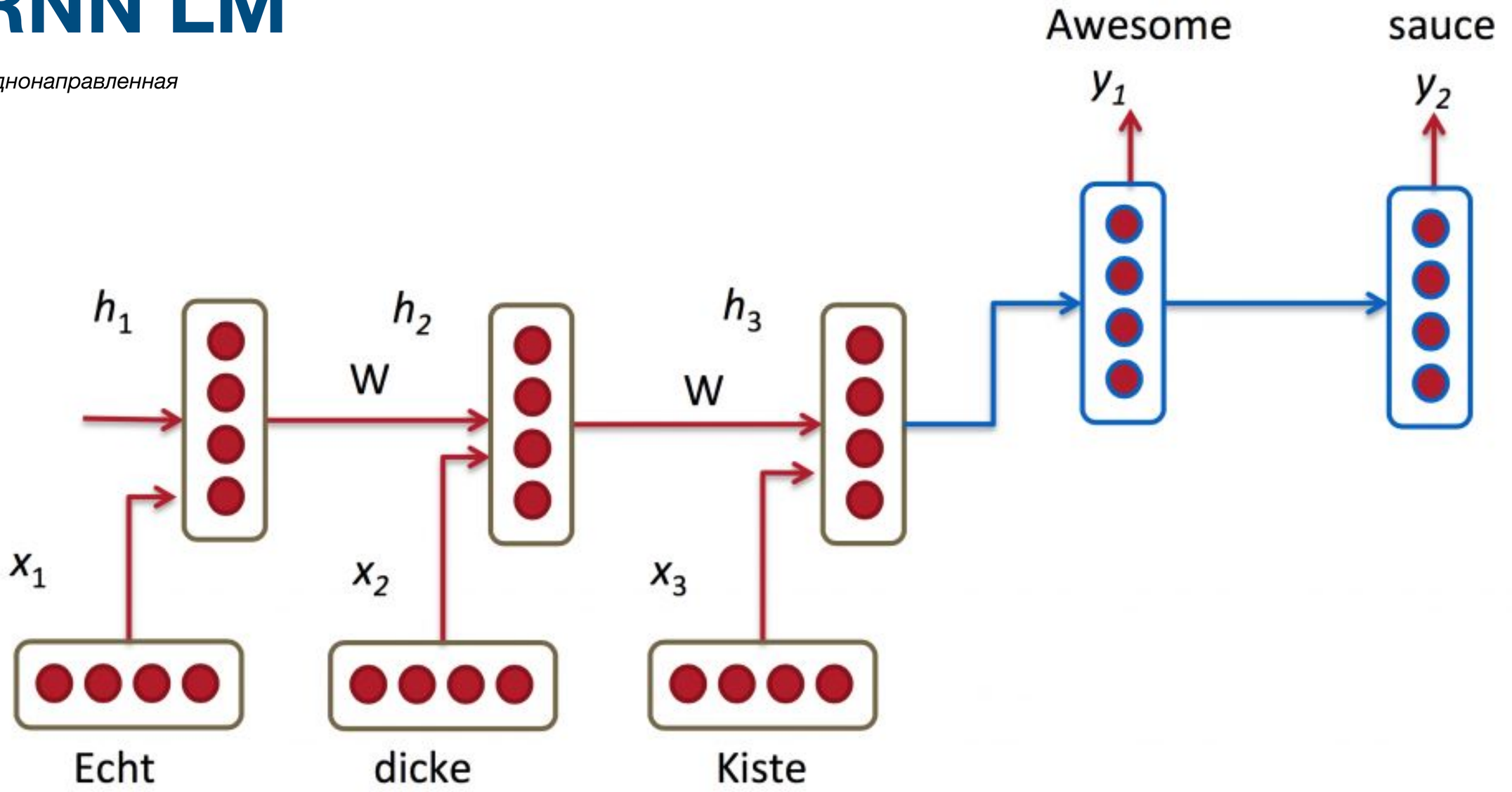
P(am | I) = count("**I am**") / count("**I**") = 2 / 3 = 0.667

P(I | <s>) = count("**<s> I**") / count("**<s>**") = 1 / 3 = 0.333

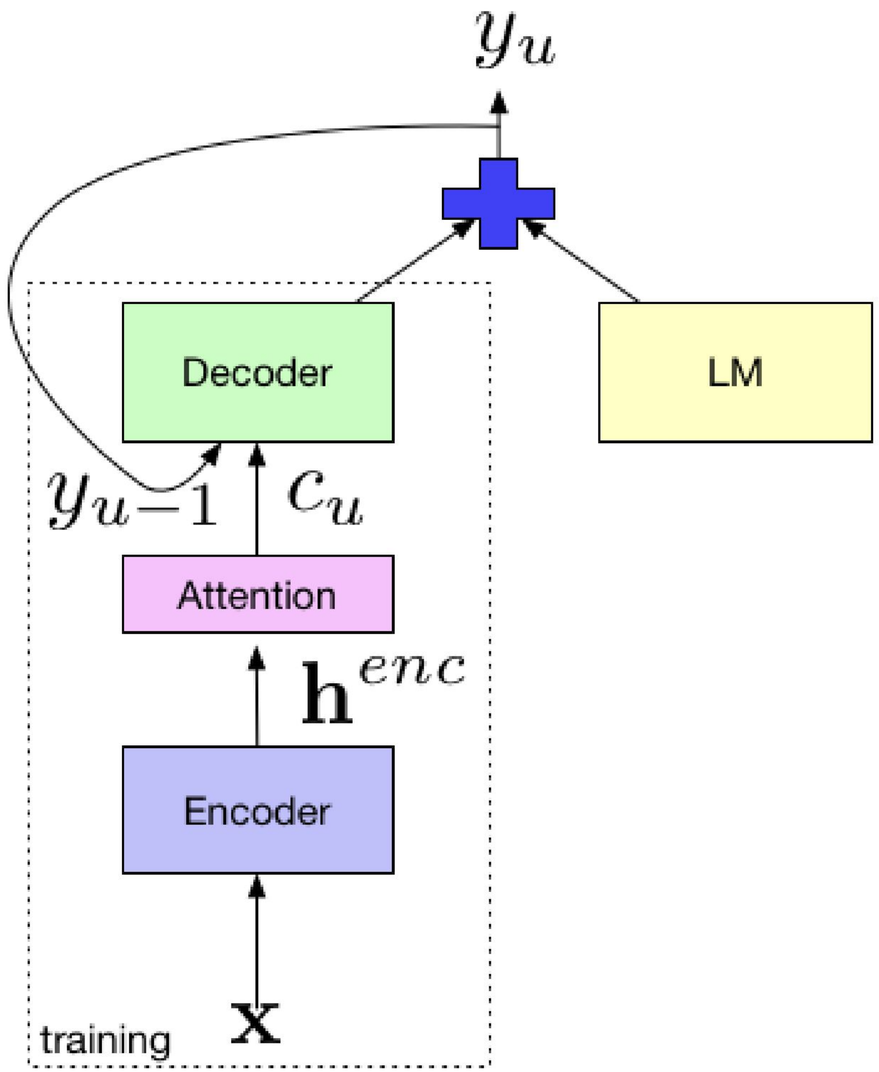
P(now | right) = count("**right now**") / count("**right**") = 1 / 1 = 1

RNN LM

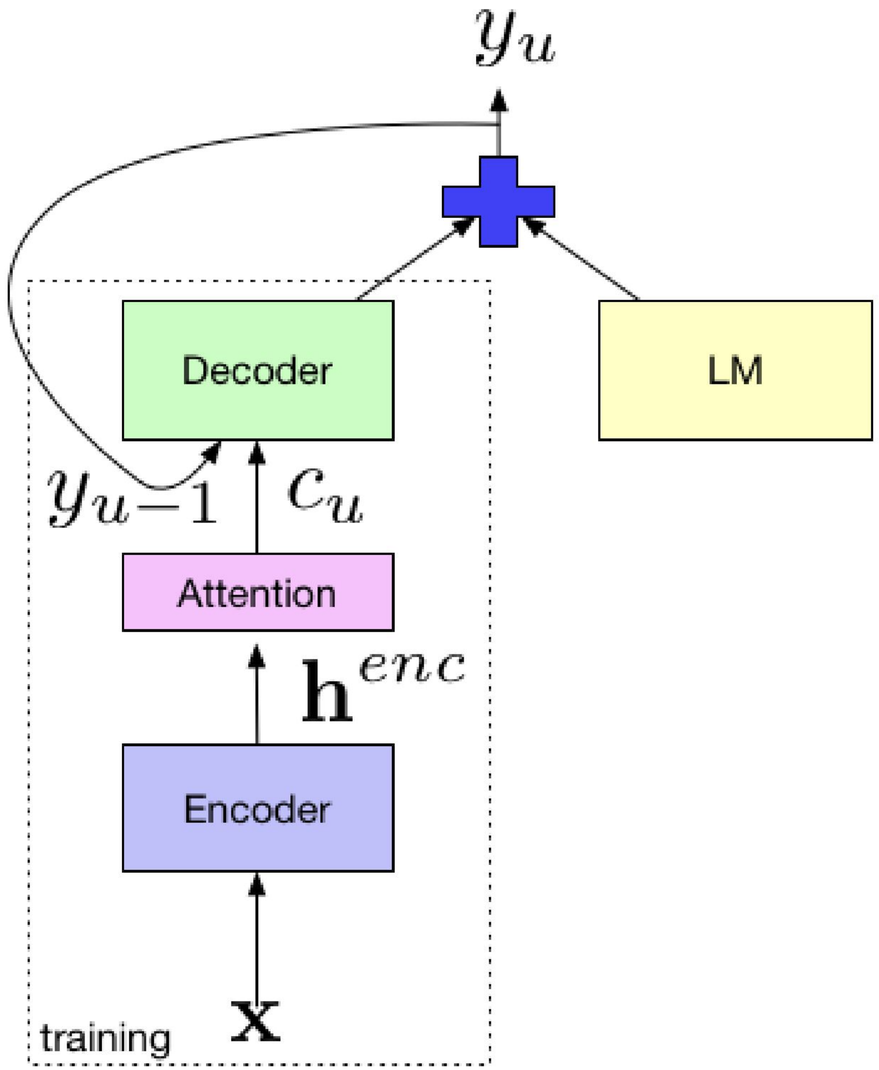
Однонаправленная

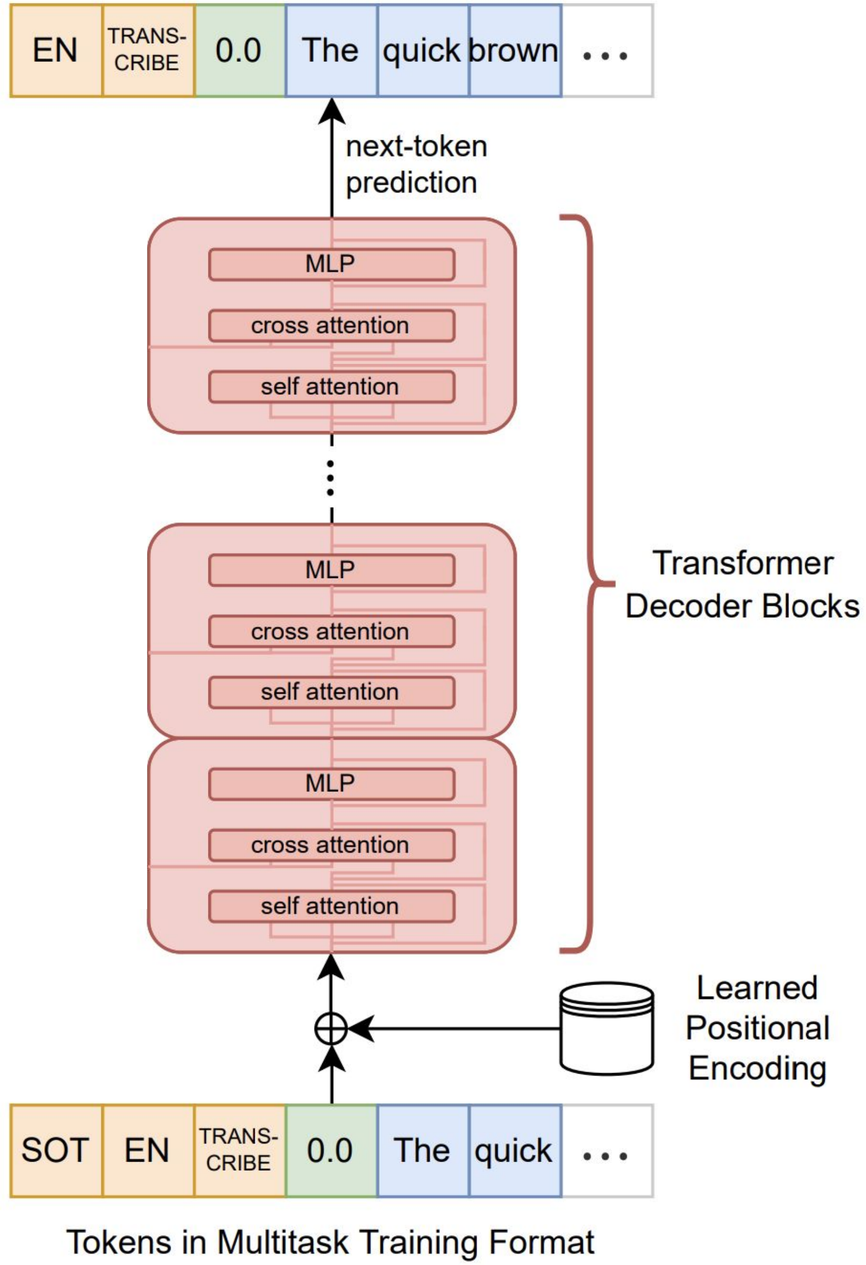


Инференс LM


y_u
Decoder
LM
y_{u-1}
c_u
Attention
$\mathbf{h}^{enc}$
Encoder
$\mathbf{x}$
training

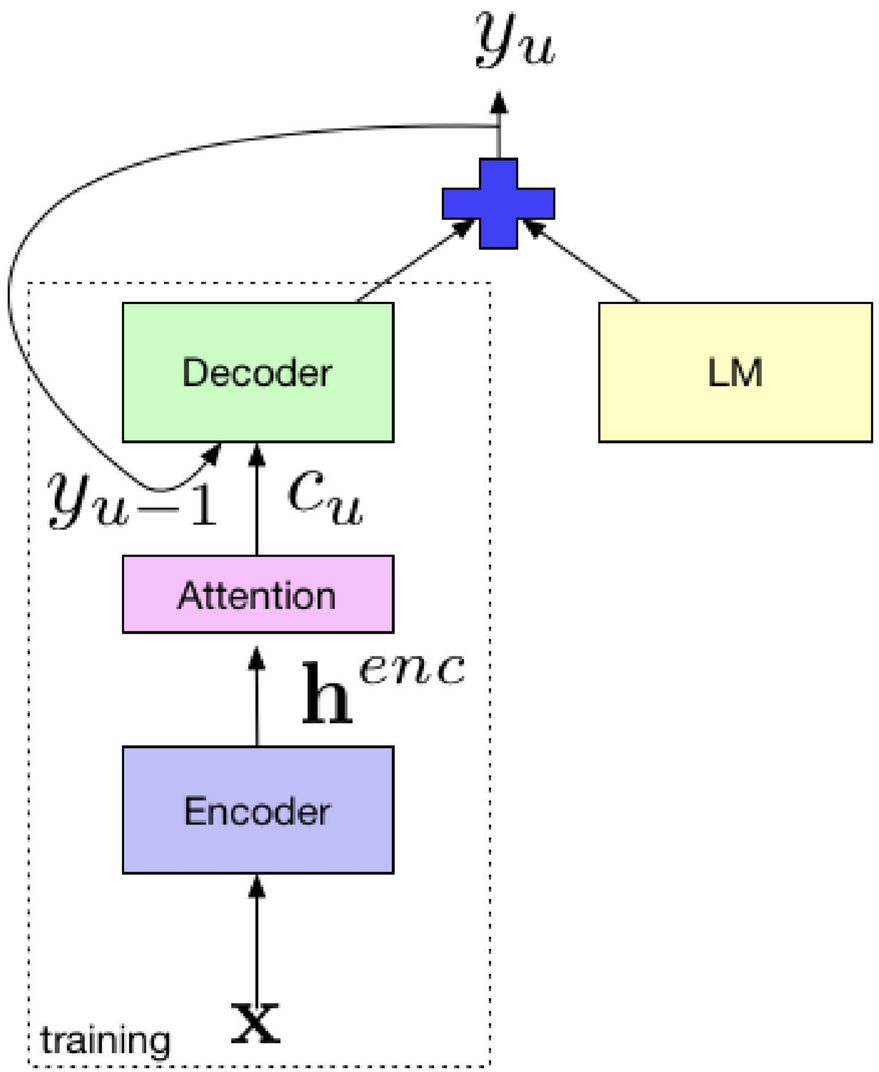
Инференс LM

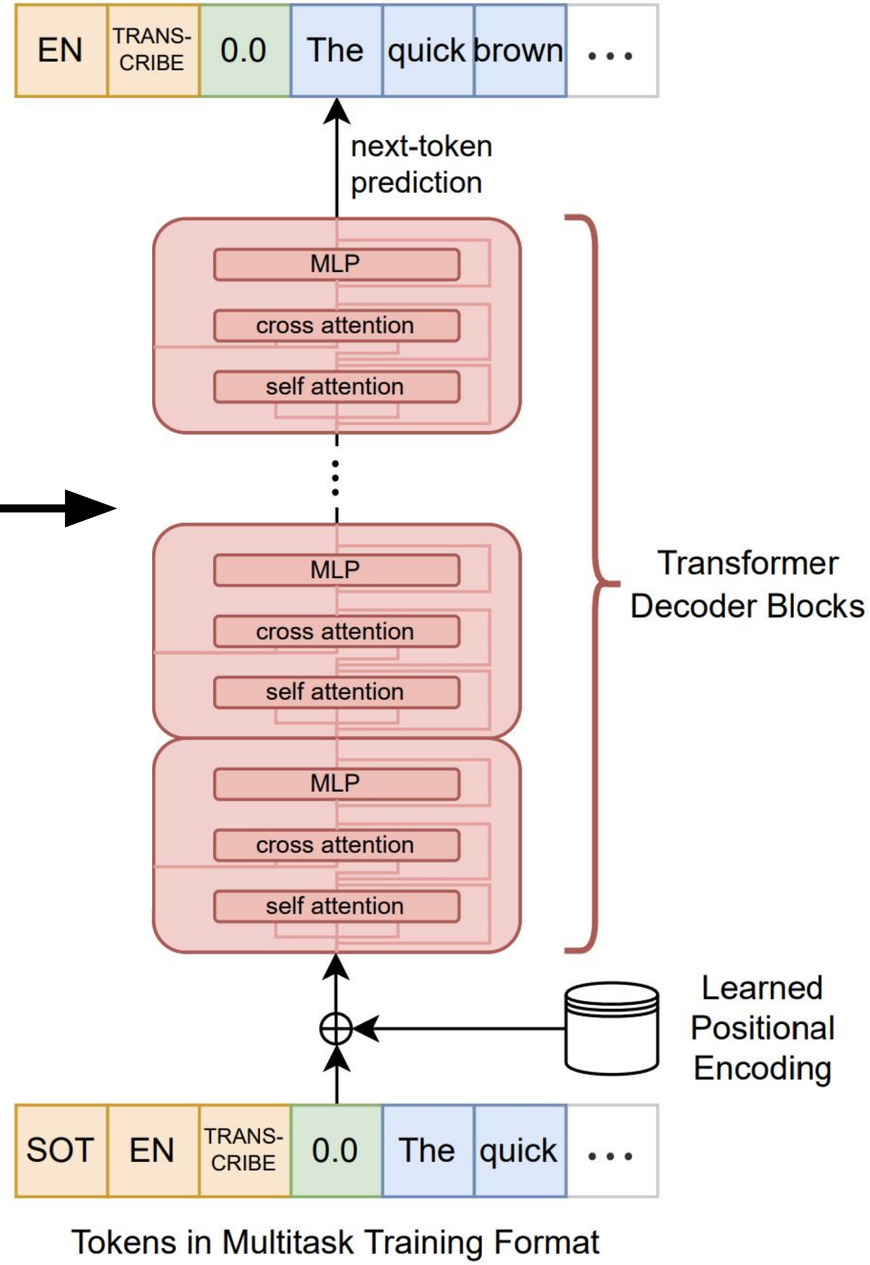




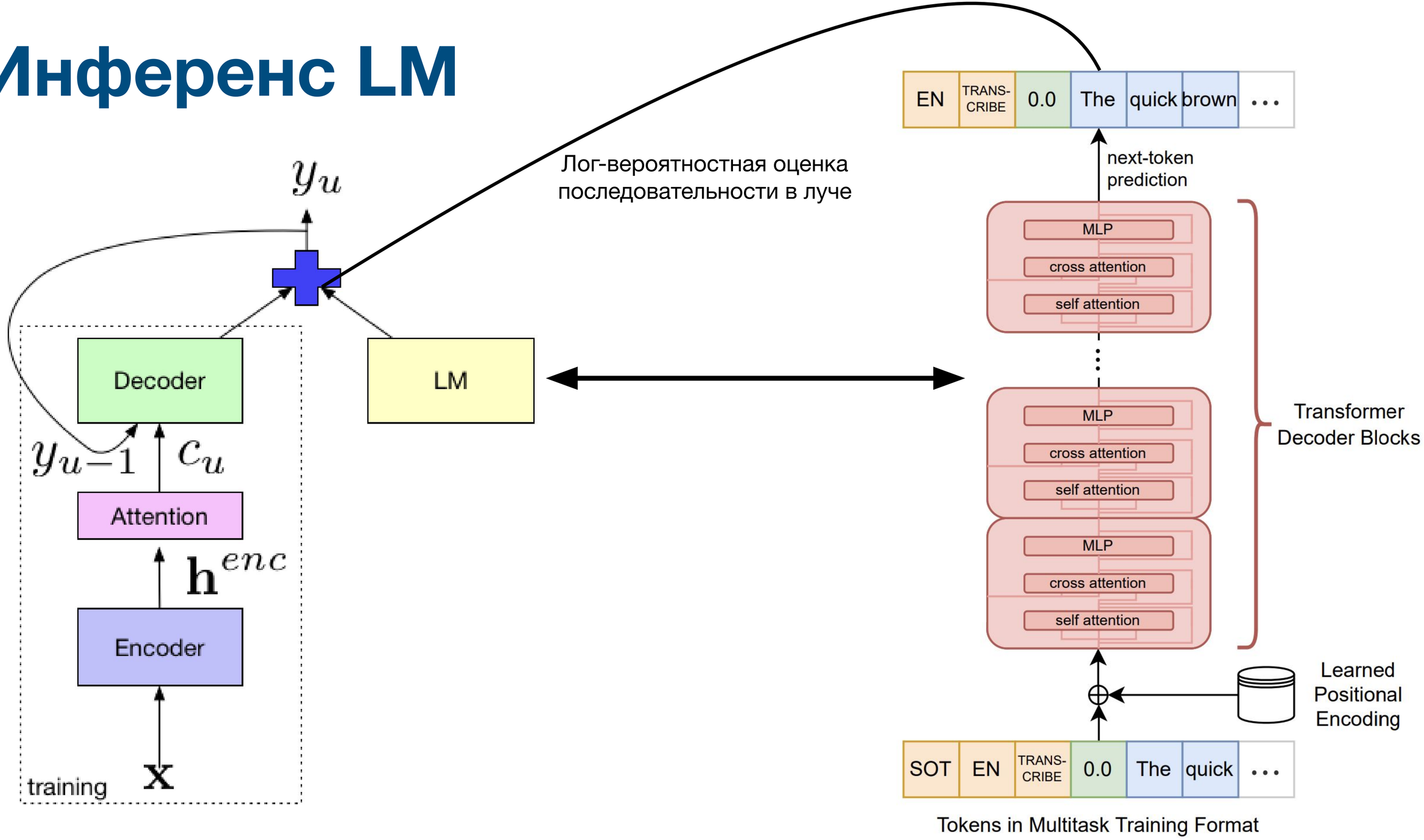
Tokens in Multitask Training Format

Инференс LM

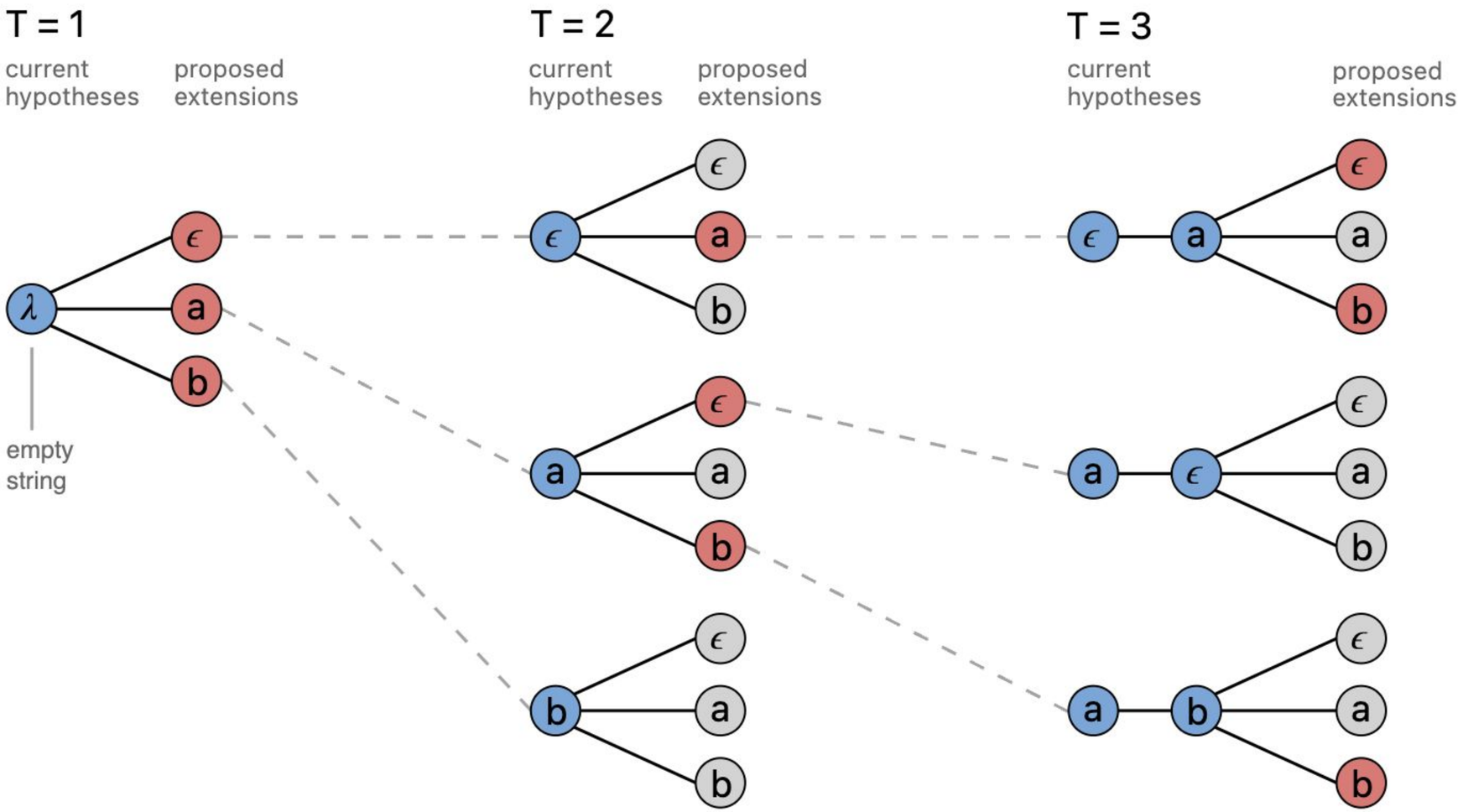

y_u
Decoder
LM
y_{u-1}
c_u
Attention
$\mathbf{h}^{enc}$
Encoder
$\mathbf{x}$
training


EN
TRANS-CRIBE
0.0
The
quick
brown
...
next-token prediction
MLP
cross attention
self attention
MLP
cross attention
self attention
MLP
cross attention
self attention
Transformer Decoder Blocks
Learned Positional Encoding
SOT
EN
TRANS-CRIBE
0.0
The
quick
...
Tokens in Multitask Training Format

Инференс LM


Лог-вероятностная оценка последовательности в луче
y_u
Decoder
LM
y_{u-1}
c_u
Attention
$\mathbf{h}^{enc}$
Encoder
$\mathbf{x}$
training
EN
TRANS-CRIBE
0.0
The
quick
brown
...
next-token prediction
MLP
cross attention
self attention
Transformer Decoder Blocks
Learned Positional Encoding
SOT
EN
TRANS-CRIBE
0.0
The
quick
...
Tokens in Multitask Training Format

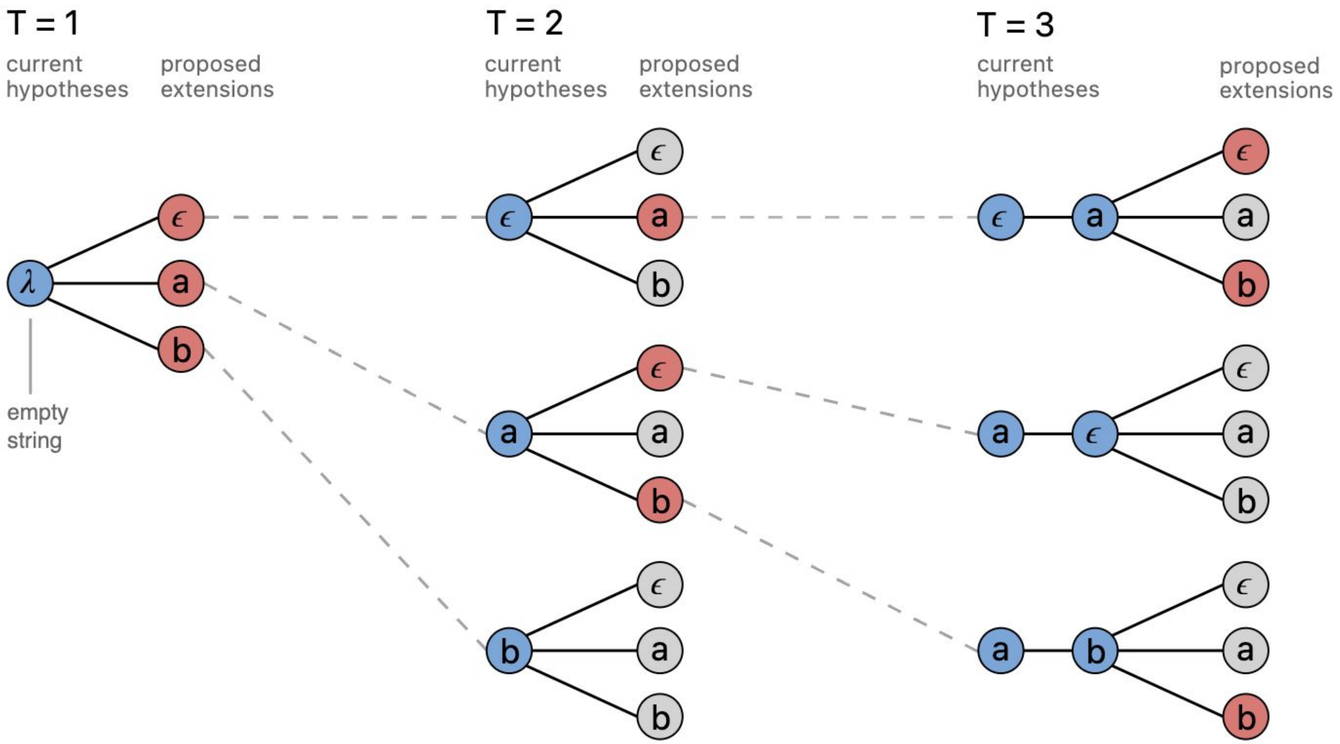
Инференс LM



A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.

Инференс LM

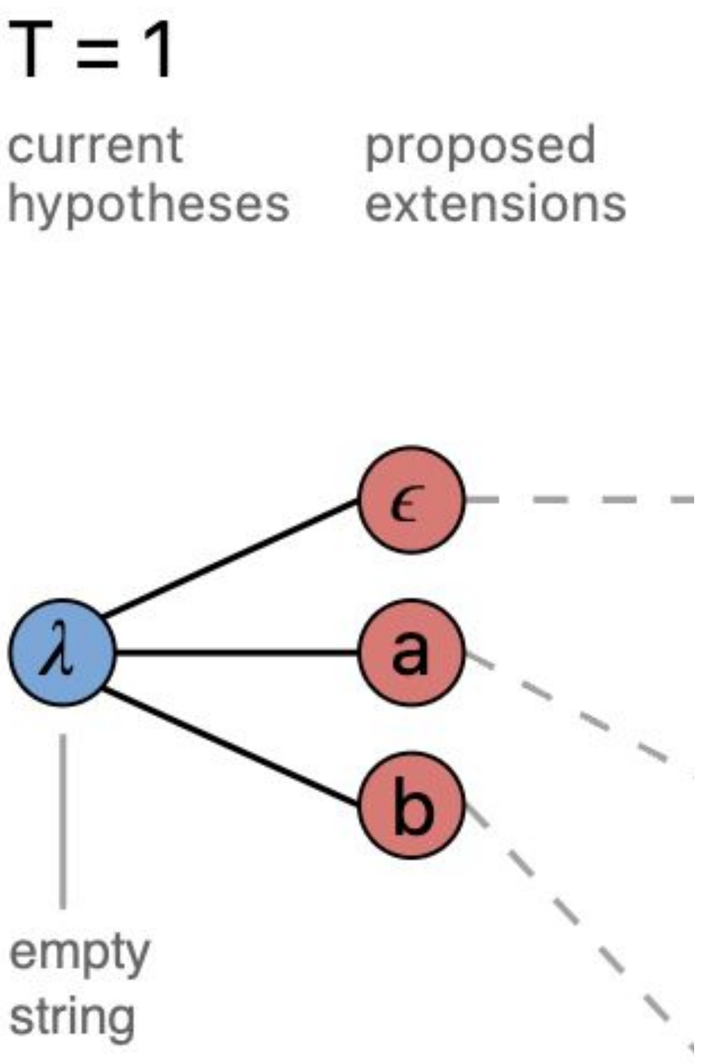
N-Gram



A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.

Инференс LM

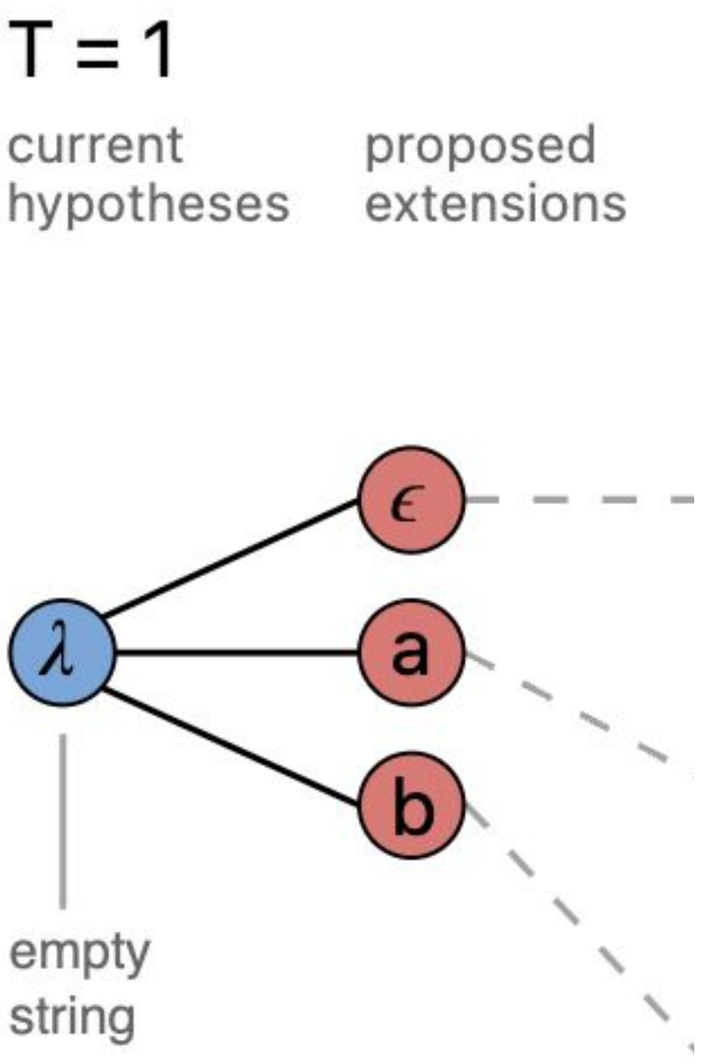
N-Gram



Инференс LM

N-Gram

A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.

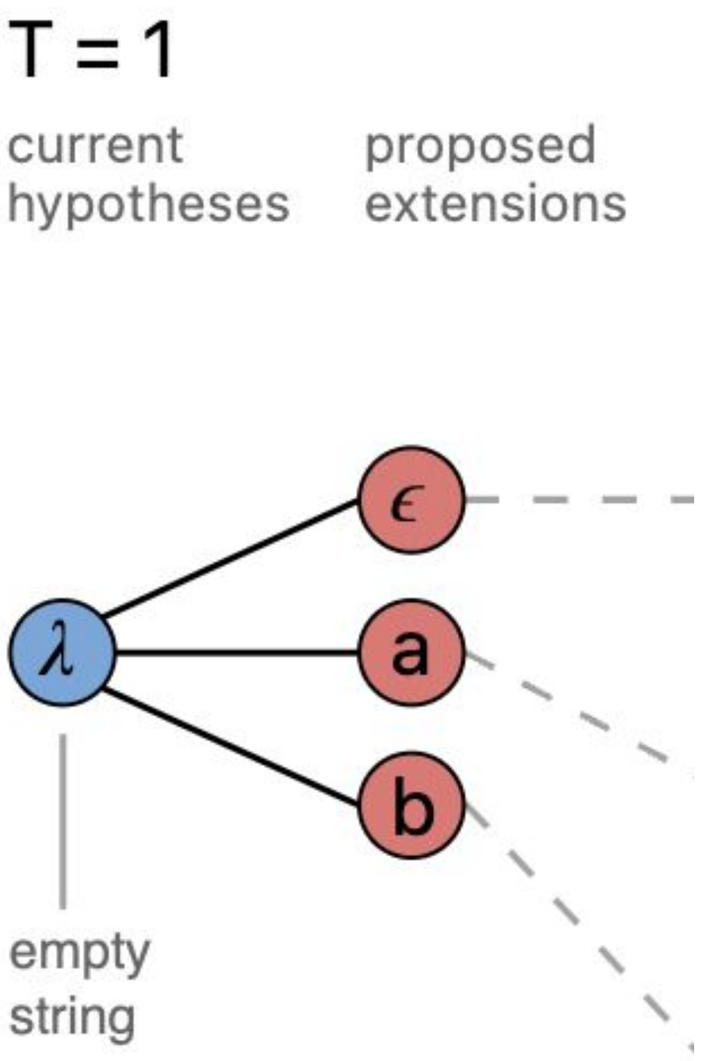


Инференс LM

N-Gram

A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.

N=3



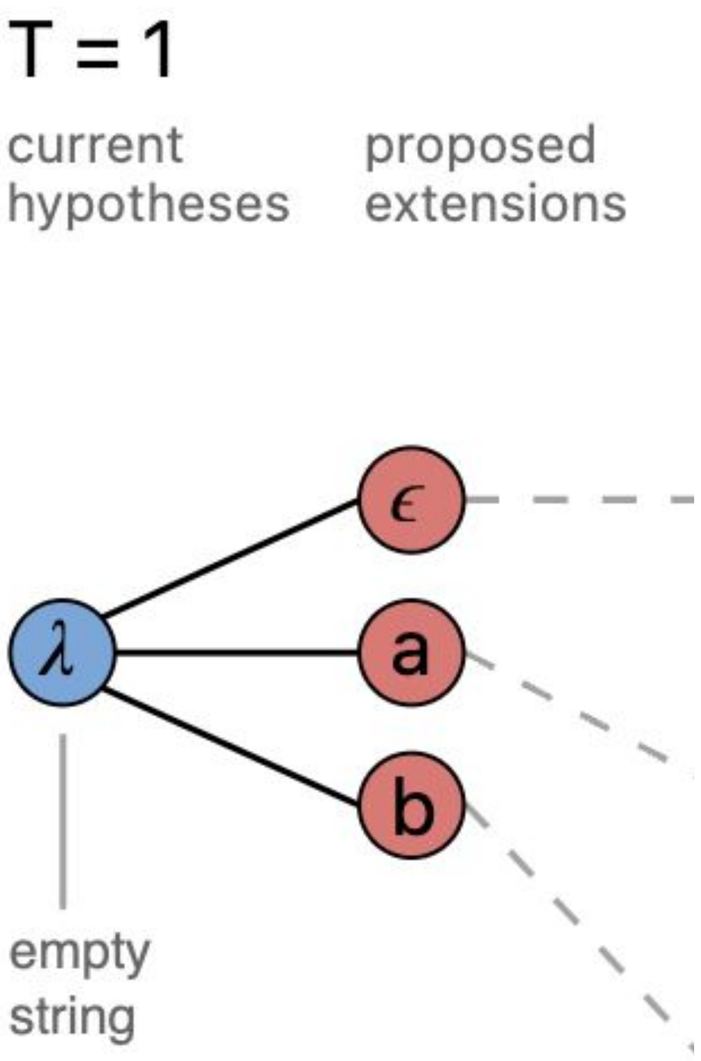
Инференс LM

N-Gram

A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.

N=3

Шаг 1: [<SOS>]
p(blank) = p(blank) * p(blank | <SOS>)



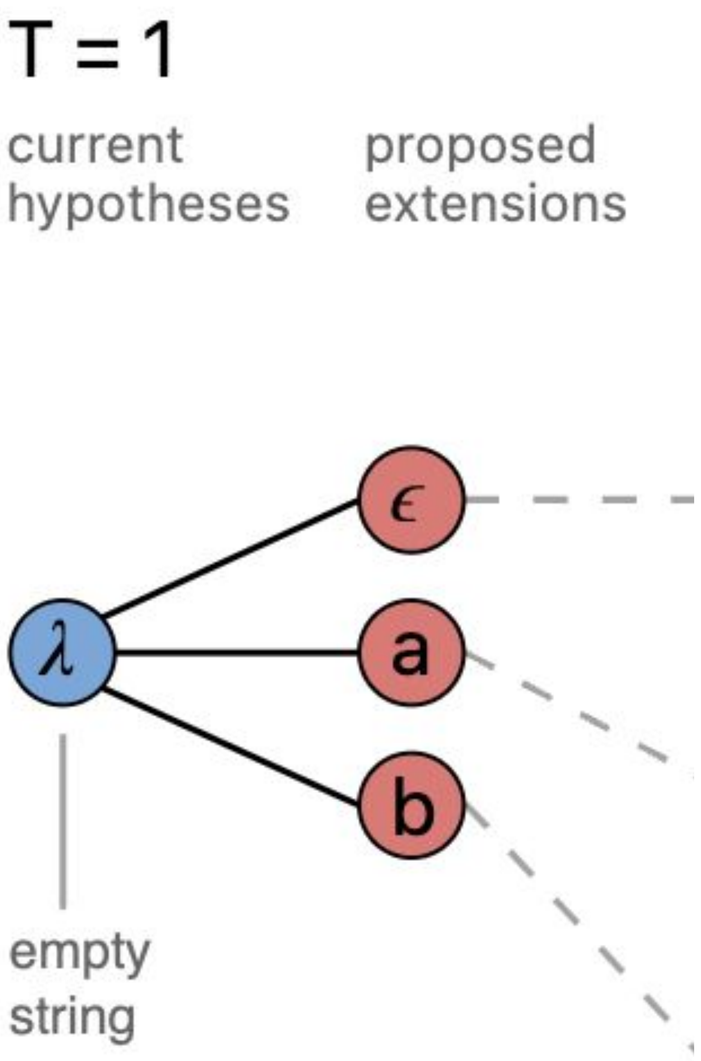
Инференс LM

N-Gram

A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.

N=3

Шаг 1: [<SOS>]
p(blank) = p(blank) * p(blank | <SOS>)



Инференс LM

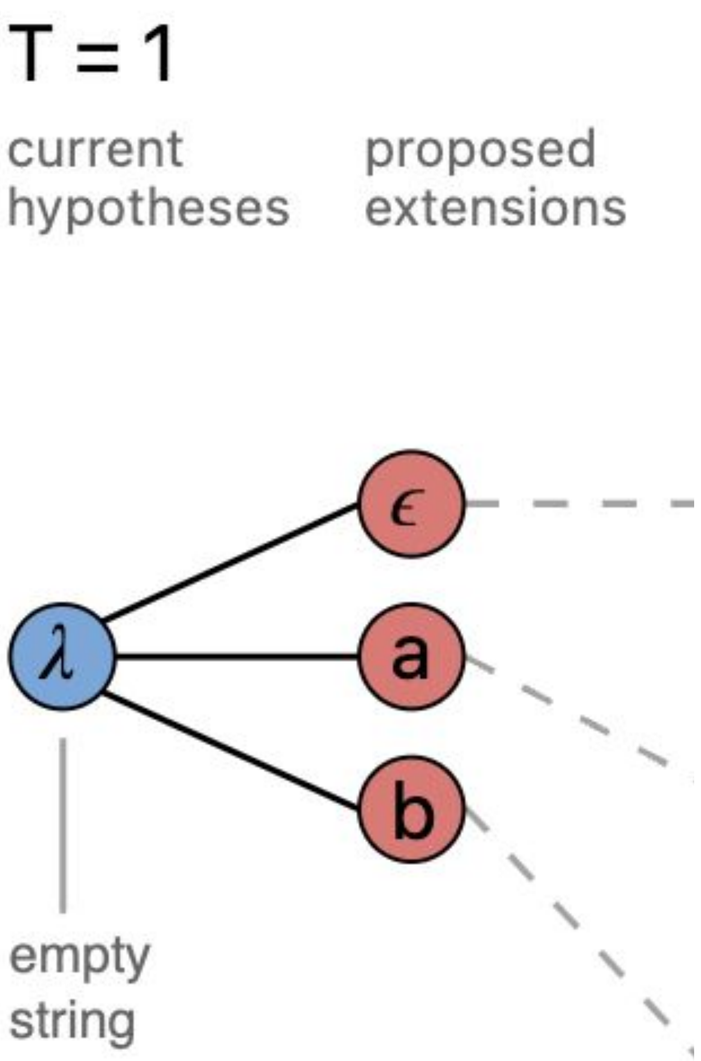
N-Gram

A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.

N=3

Шаг 1: [<SOS>]
p(blank) = p(blank) * p(blank | <SOS>)

1-gram 2-gram



Инференс LM

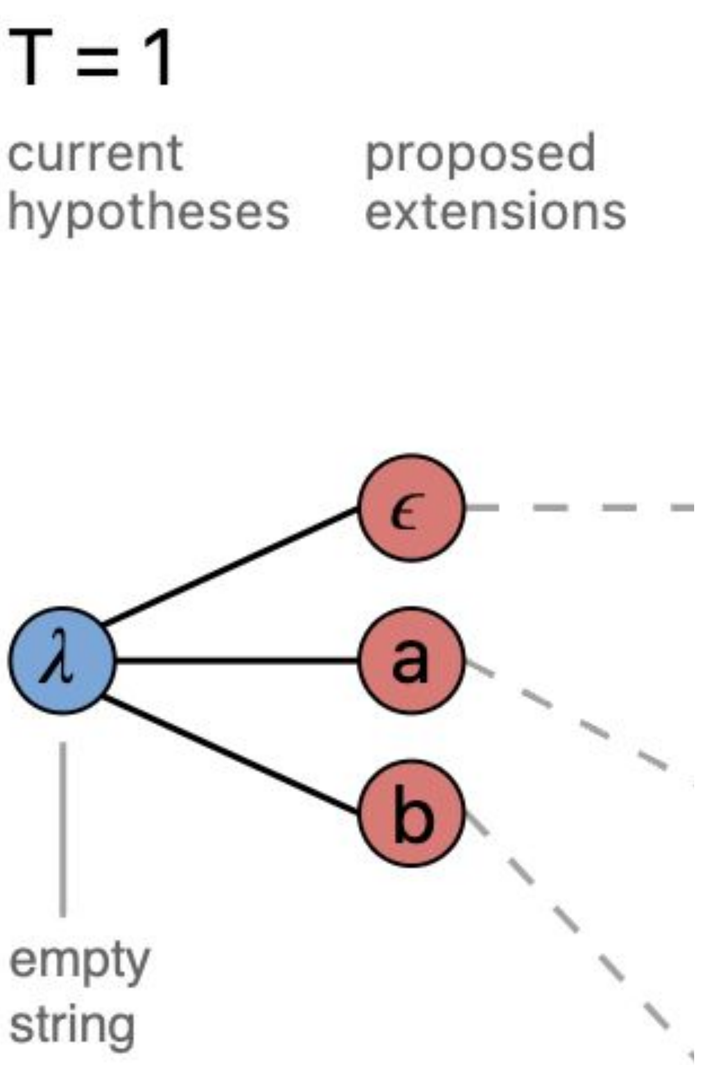
N-Gram

A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.

N=3

Шаг 1: [<SOS>]
p(blank) = p(blank) * p(blank | <SOS>)
p(a) = p(a) * p(a | <SOS>)

1-gram 2-gram



Инференс LM

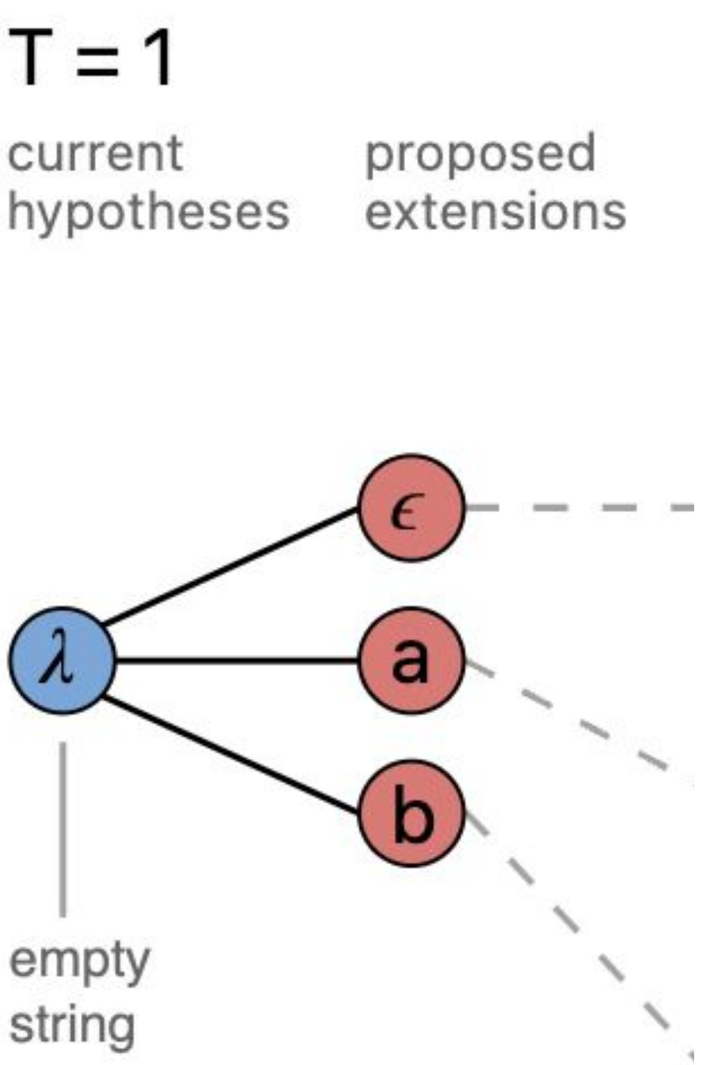
N-Gram

A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.

N=3

Шаг 1: [<SOS>]
p(blank) = p(blank) * p(blank | <SOS>)
p(a) = p(a) * p(a | <SOS>)
p(b) = p(b) * p(b | <SOS>)

1-gram 2-gram



Инференс LM

N-Gram

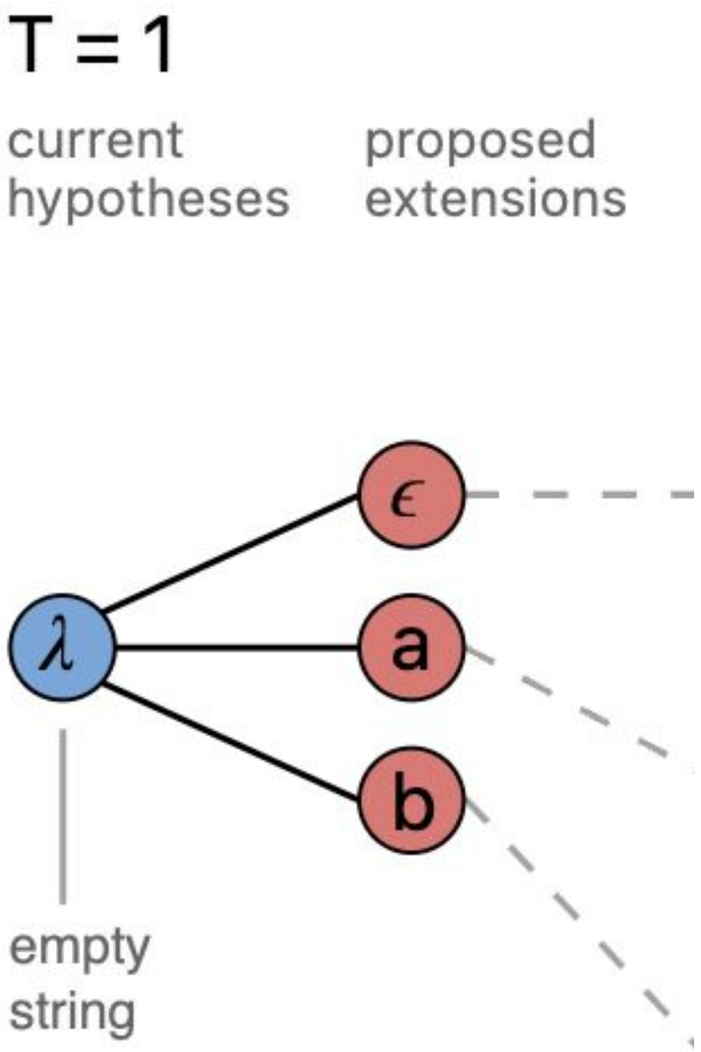
A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.

N=3

Шаг 1: [<SOS>]
p(blank) = p(blank) * p(blank | <SOS>)
p(a) = p(a) * p(a | <SOS>)
p(b) = p(b) * p(b | <SOS>)

1-gram 2-gram

Шаг 1: [<SOS>]
log(p(blank)) = log(p(blank)) + log(p(blank | <SOS>))



Инференс LM

N-Gram

A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.

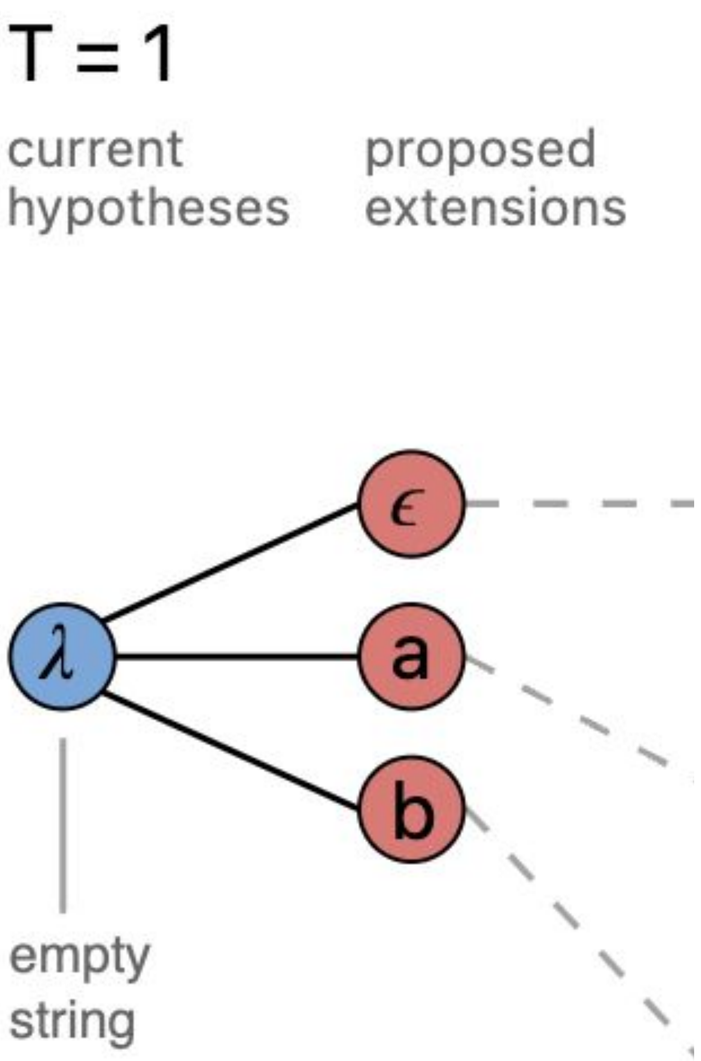
N=3

Шаг 1: [<SOS>]
p(blank) = p(blank) * p(blank | <SOS>)
p(a) = p(a) * p(a | <SOS>)
p(b) = p(b) * p(b | <SOS>)

1-gram 2-gram

Шаг 1: [<SOS>]
log(p(blank)) = log(p(blank)) + log(p(blank | <SOS>))

Общий вид:
log(p(t)) = Sum(log(p(t | t-1, t-2, …, t-N+1)))



Инференс LM

N-Gram

A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.

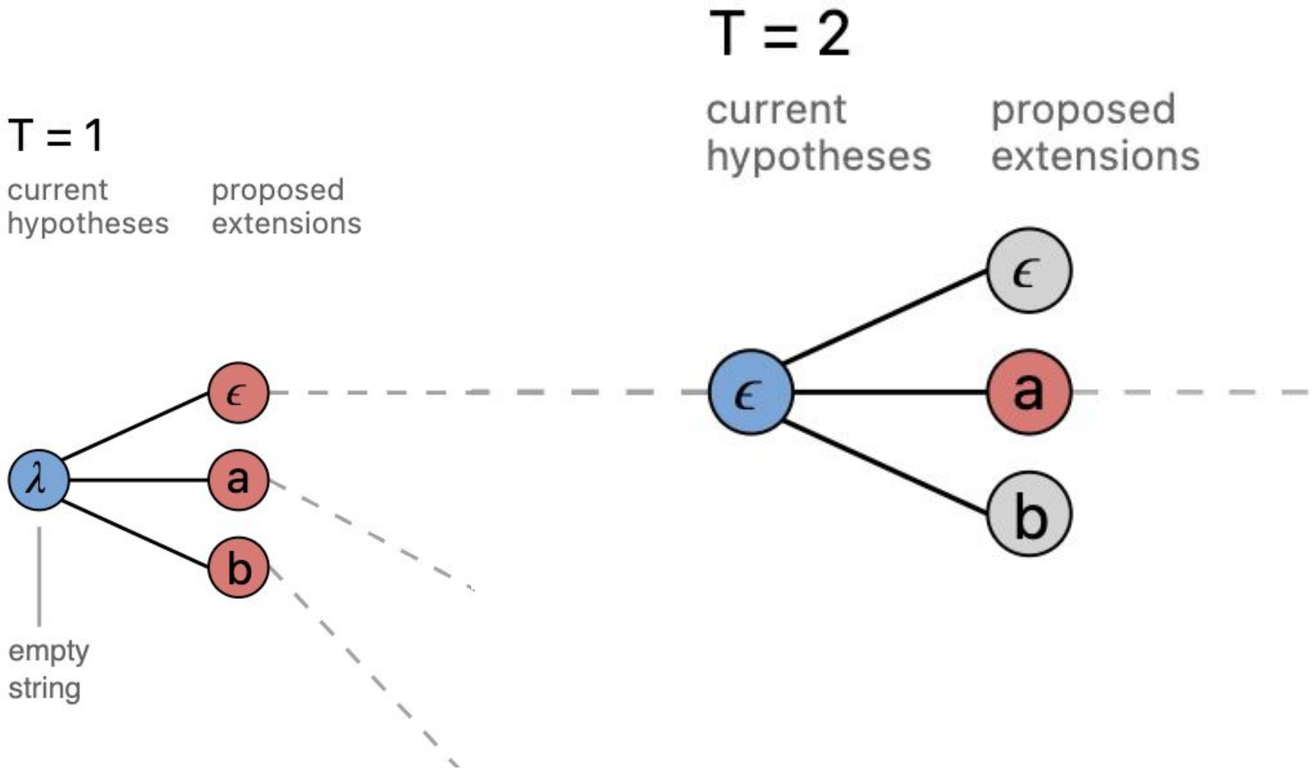
N=3

Шаг 2: [<SOS>, blank]
p(blank) = p(blank) * p(blank | blank) * p(blank | blank, <SOS>)
p(a) = p(a) * p(a | blank) * p(a | blank, <SOS>)
p(b) = p(b) * p(b | blank) * p(b | blank, <SOS>)

1-gram 2-gram 3-gram

Шаг 2: [<SOS>, blank]
log(p(a)) = log(p(a)) + log(p(a | blank)) + log(p(a | blank, <SOS>))

Общий вид:
log(p(t)) = Sum(log(p(t | t-1, t-2, …, t-N+1)))

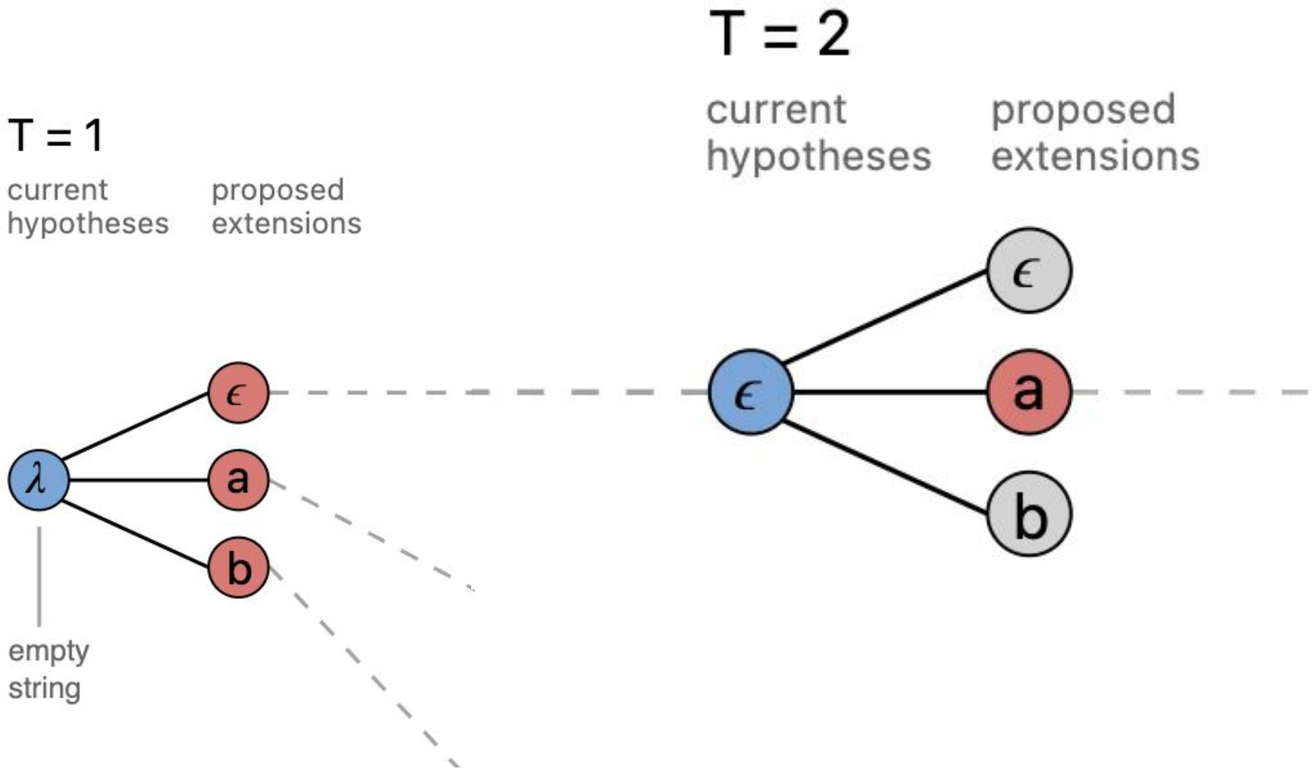


Инференс LM

RNN

A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.

Шаг 1: [<SOS>]
hidden = h_0 (zeros)
log-score-1, h_1 = log_softmax(RNN(<SOS>, h_0))

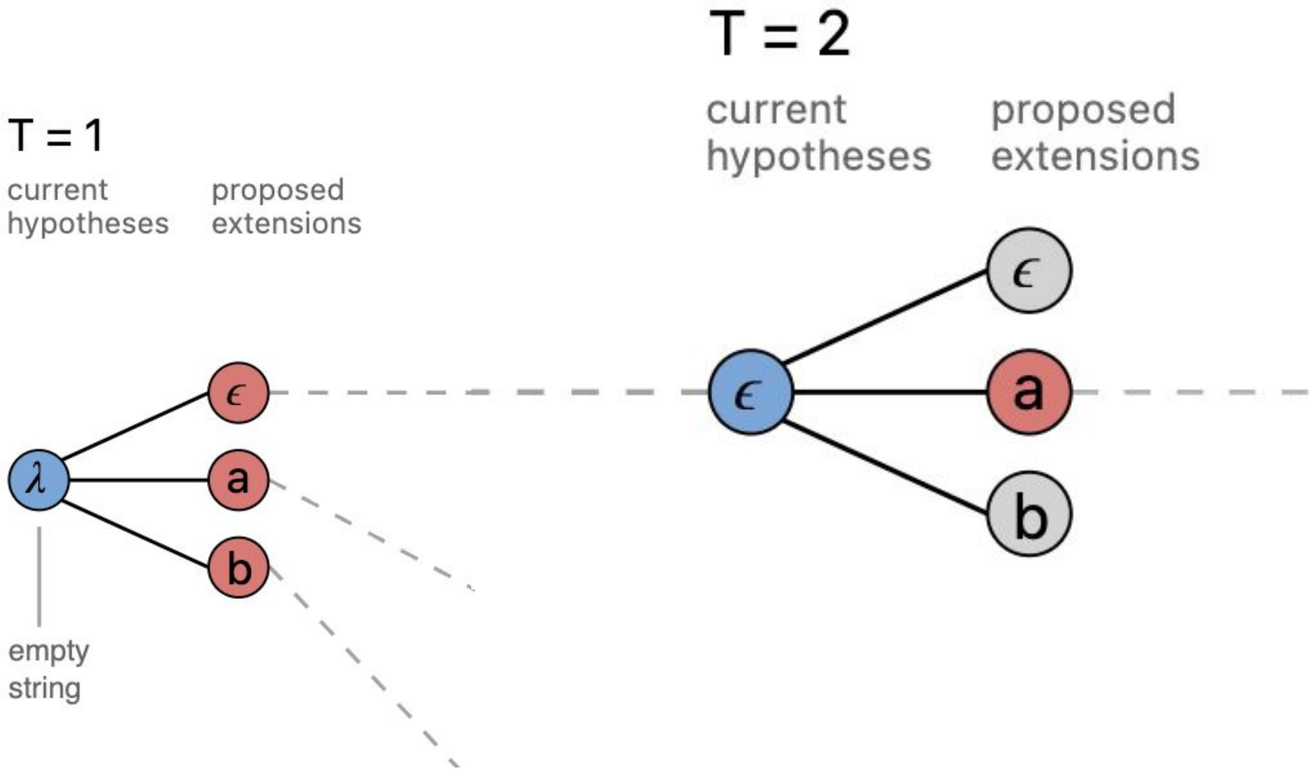


Инференс LM

RNN

A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.

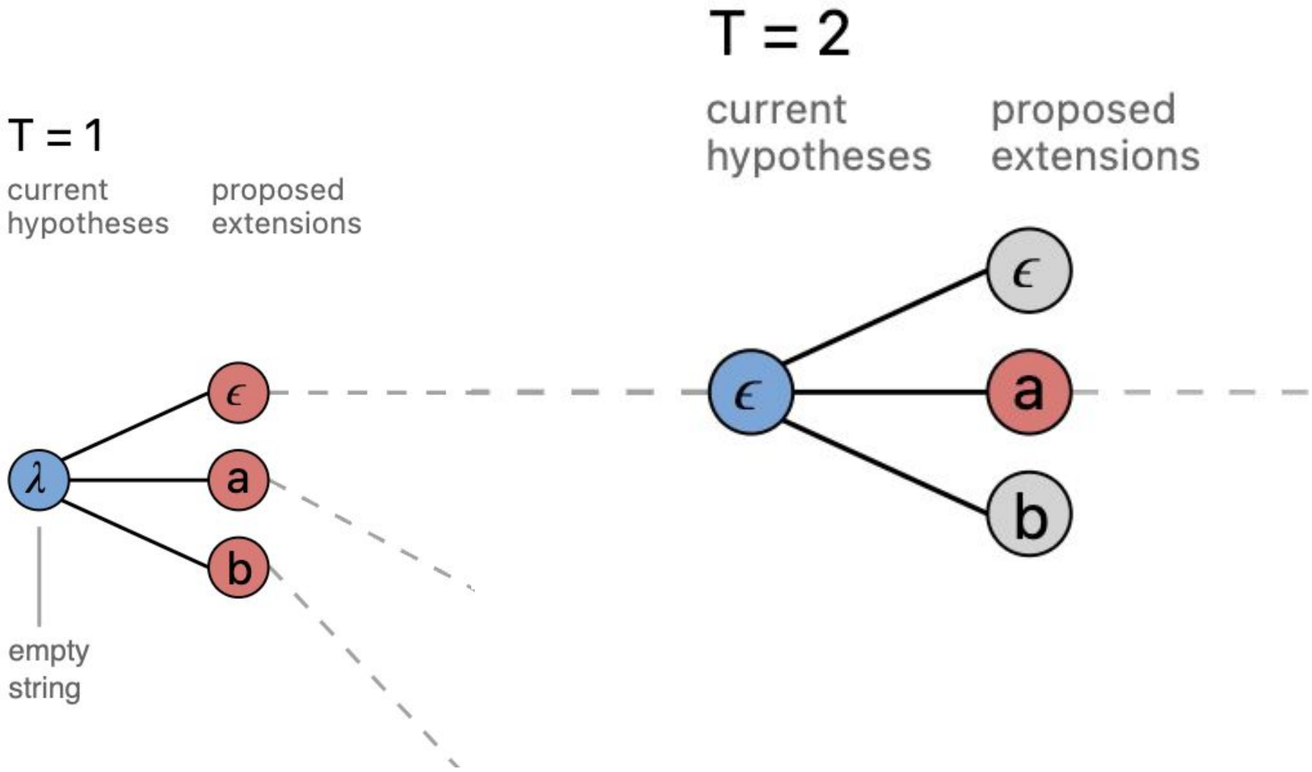
Шаг 2: [<SOS>, blank]
hidden = h_1
log-score-2, h_2 = log_softmax(RNN(blank, h_1))



Инференс LM

RNN

A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.



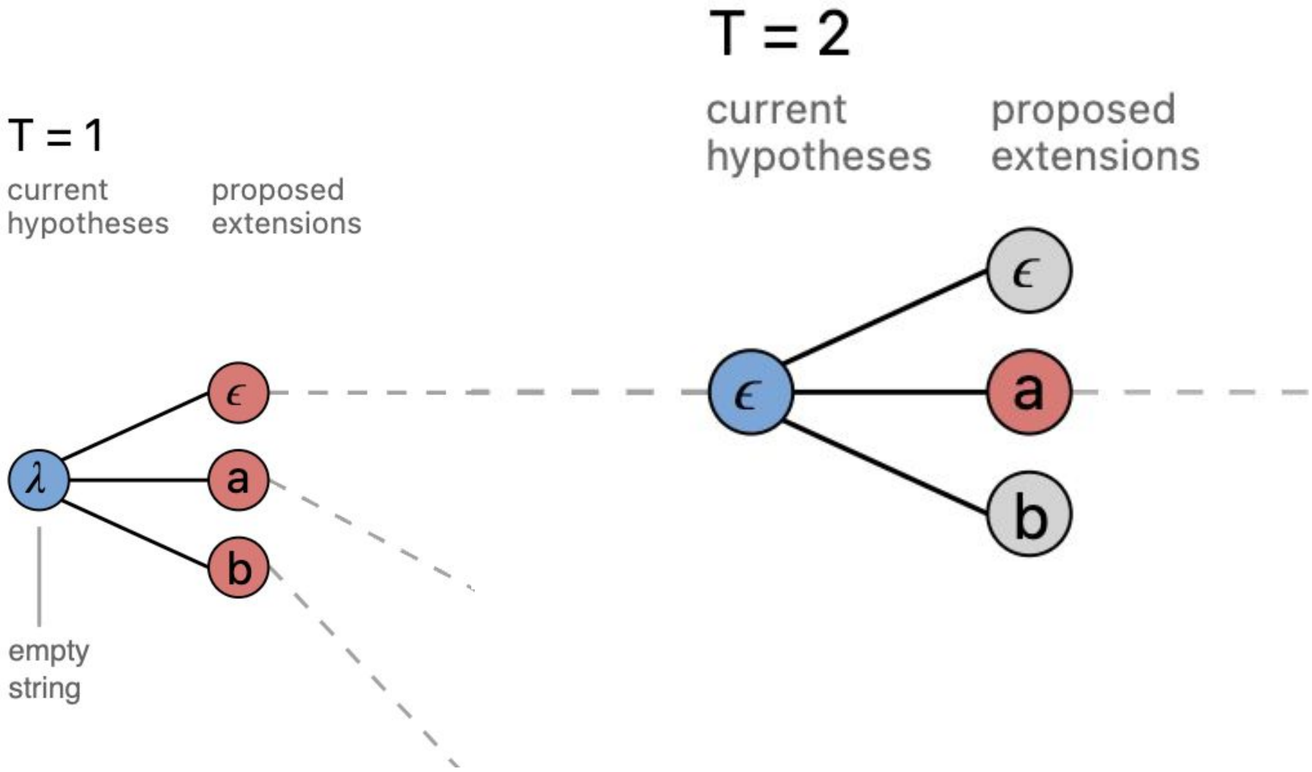
Шаг 2: [<SOS>, blank]
hidden = h_1
log-score-2, h_2 = log_softmax(RNN(blank, h_1))

log-score-2 - вектор log-вероятностей размера словаря

Инференс LM

RNN

A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.



Шаг 2: [<SOS>, blank]
hidden = h_1
log-score-2, h_2 = log_softmax(RNN(blank, h_1))

log-score-2 - вектор log-вероятностей размера словаря

для каждого луча хранится свой hidden

Keywords / Hotwords

hotword - DLS

Keywords / Hotwords

hotword - DLS

словарь = {a, b, c, d, l, s}

Keywords / Hotwords

hotword - DLS

словарь = {a, b, c, d, l, s}

N-beams = 3

Keywords / Hotwords

hotword - DLS

словарь = {a, b, c, d, l, s}

N-beams = 3

гипотеза 1:	score 1:
гипотеза 2:	score 2:
гипотеза 3:	score 3:

Keywords / Hotwords

hotword - DLS

словарь = {a, b, c, d, l, s}

N-beams = 3

гипотеза 1: a	score 1: s1_1
гипотеза 2: b	score 2: s2_1
гипотеза 3: c	score 3: s3_1

Keywords / Hotwords

hotword - DLS

словарь = {a, b, c, d, l, s}

N-beams = 3

гипотеза 1: ab	score 1: s1_2
гипотеза 2: bc	score 2: s2_2
гипотеза 3: ca	score 3: s3_2

Keywords / Hotwords

hotword - DLS

словарь = {a, b, c, d, l, s}

N-beams = 3

гипотеза 1: ab_	score 1: s1_3
гипотеза 2: bc_	score 2: s2_3
гипотеза 3: ca_	score 3: s3_3

Keywords / Hotwords

hotword - DLS

словарь = {a, b, c, d, l, s}

N-beams = 3

гипотеза 1: ab_d	score 1: s1_4 + boost_score * 1
гипотеза 2: bc_d	score 2: s2_4 + boost_score * 1
гипотеза 3: ca_d	score 3: s3_4 + boost_score * 1

Keywords / Hotwords

hotword - DLS

словарь = {a, b, c, d, l, s}

N-beams = 3

гипотеза 1: ab_dl	score 1: s1_5 + boost_score * 2
гипотеза 2: bc_dl	score 2: s2_5 + boost_score * 2
гипотеза 3: ca_dl	score 3: s3_5 + boost_score * 2

Keywords / Hotwords

hotword - DLS

словарь = {a, b, c, d, l, s}

N-beams = 3

гипотеза 1: ab_dls	score 1: s1_6 + boost_score * 3
гипотеза 2: bc_dls	score 2: s2_6 + boost_score * 3
гипотеза 3: ca_dla	score 3: s3_6 + 0

Keywords / Hotwords

hotword - DLS

словарь = {a, b, c, d, l, s}

N-beams = 3

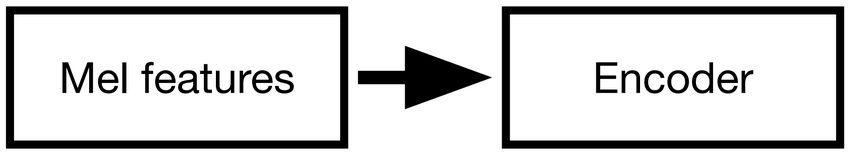
гипотеза 1: ab_dls_	score 1: s1_7 + boost_score * 3
гипотеза 2: bc_dlss	score 2: s2_7 + boost_score * 3
гипотеза 3: ca_dlas	score 3: s3_7

Финальная схема

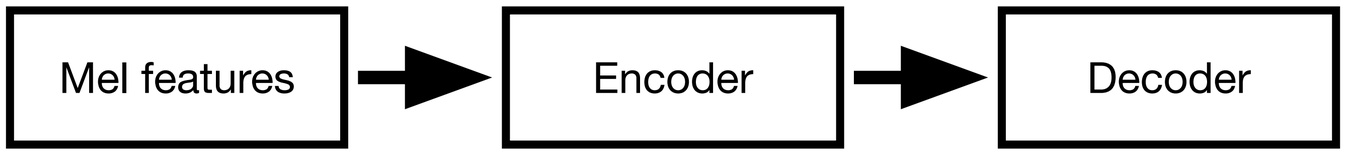
Полный пайплайн

Полный пайплайн

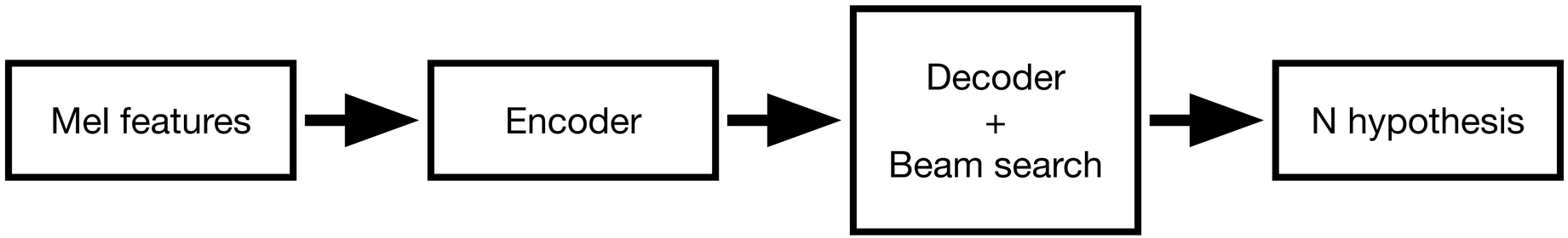
Полный пайплайн



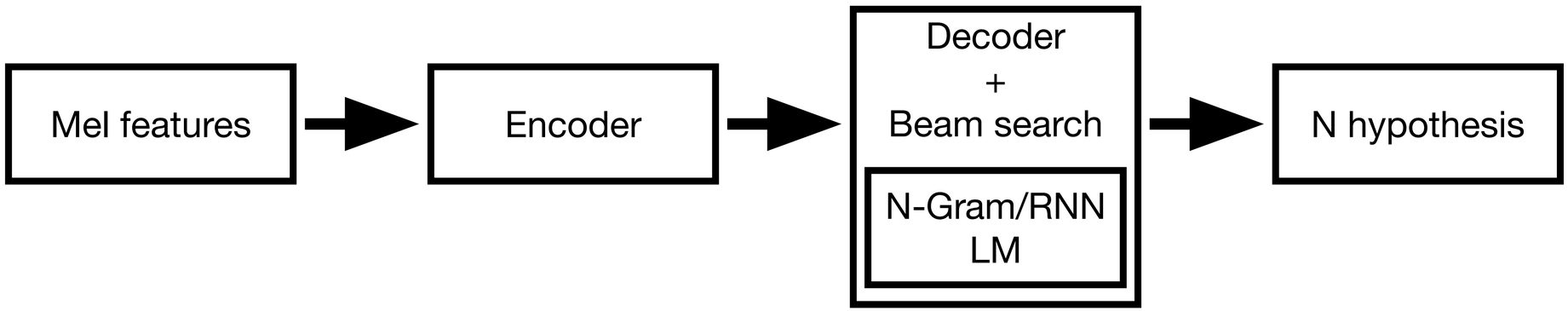
Полный пайплайн



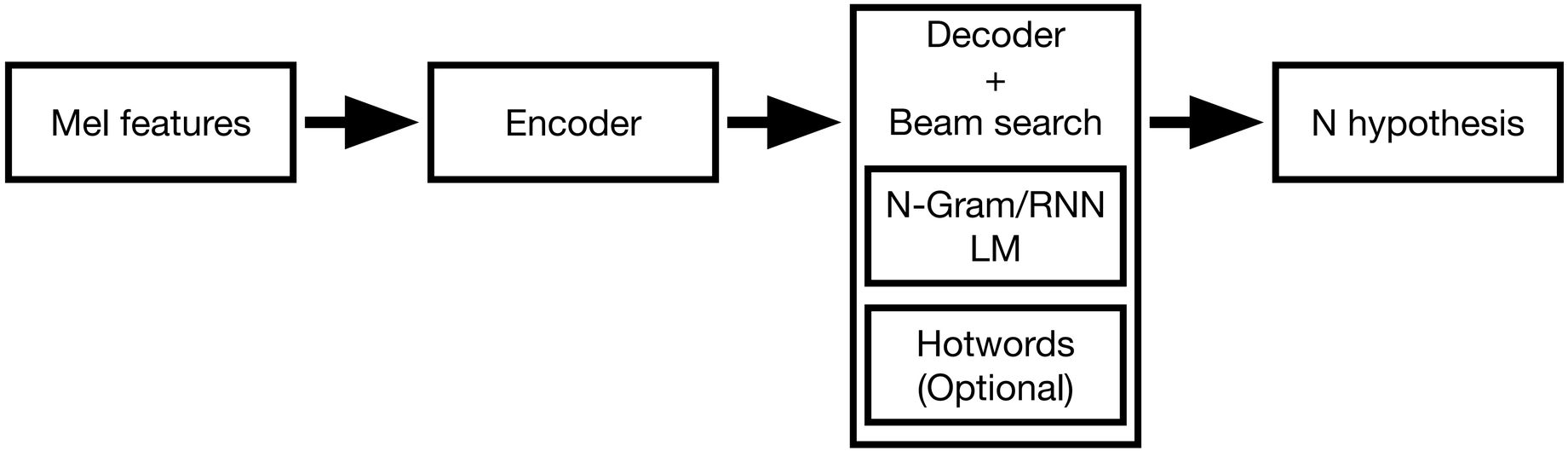
Полный пайплайн



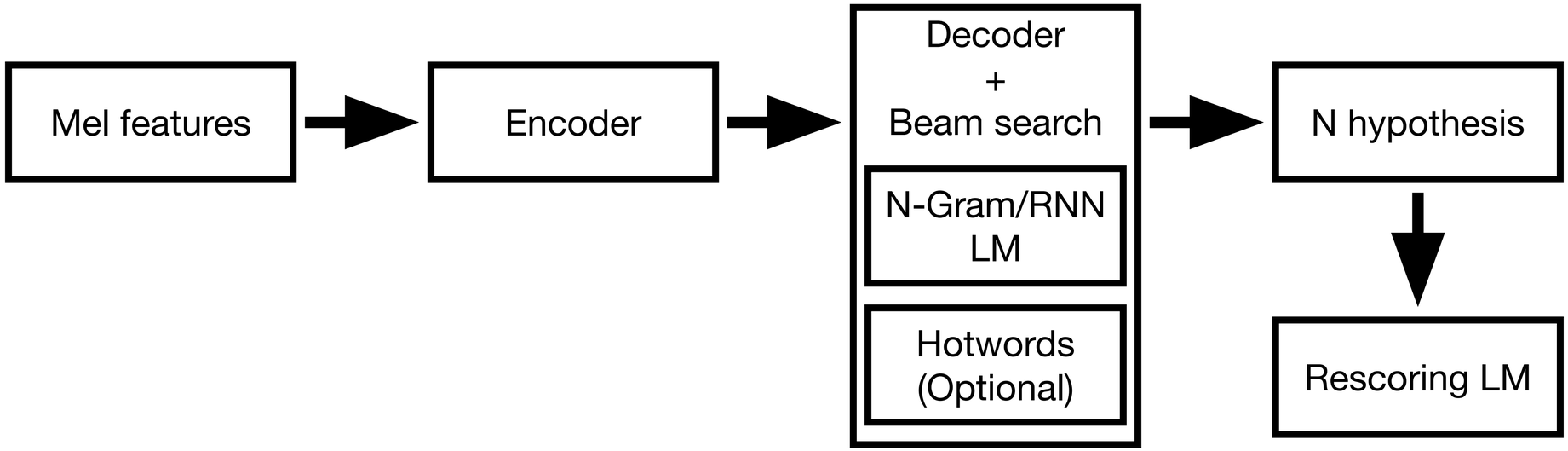
Полный пайплайн



Полный пайплайн



Полный пайплайн



Полный пайплайн