

# OCR

Capcha recognition and other problems

# Области применения OCR

- Распознавание автомобильных номеров
- Распознавание документов
- Создание электронных версий бумажных документов
- Перевод текста по фото
- ...

# Самый первый в OCR

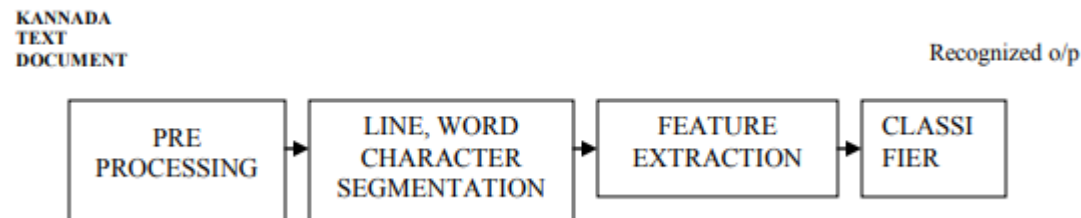
Пионер!!!!



Густав Таушек

# Эпоха до DL

- Сегментация изображения, выделение отдельных слов/символов
- Препроцессинг – бинаризация, морфологические операции, выделение признаков (любого рода)
- Классические методы МК (KNN, SVM, Bayesian)
- N-граммы вместо NLP



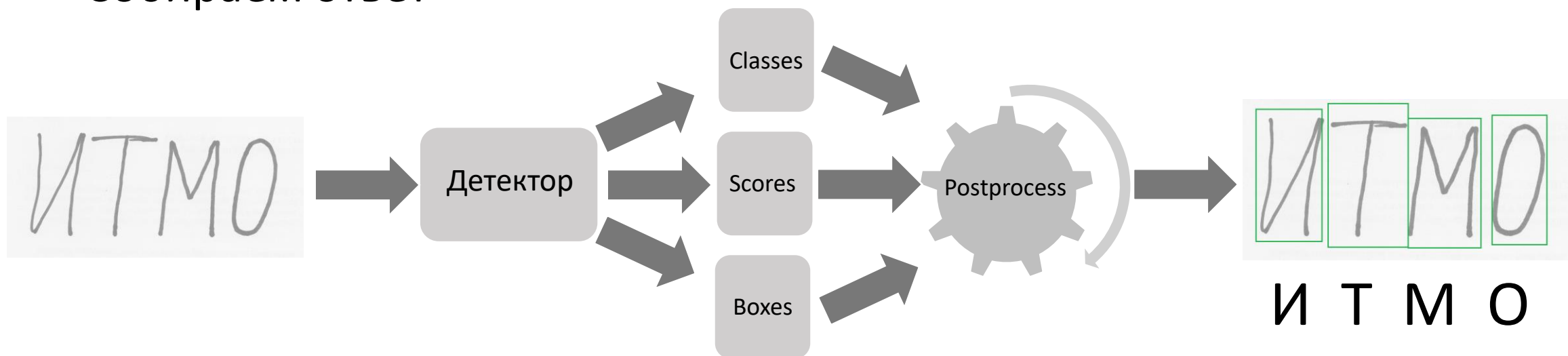
[Kannada Character Recognition System: A Review](#)

# Качество сканирования

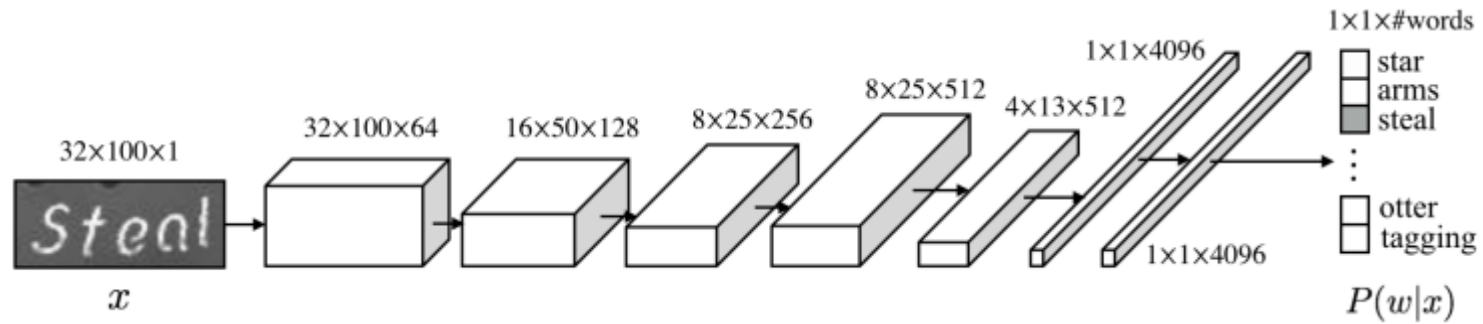
[illegible][illegible]

# Применяем DL “в лоб”

- Обучаем мультиклассовый детектор
- Сортируем боксы по оси x
- Собираем ответ



# Применяем DL “в лоб”



Распознавание 90К слов

[Reading Text in the Wild with Convolutional Neural Networks](#)

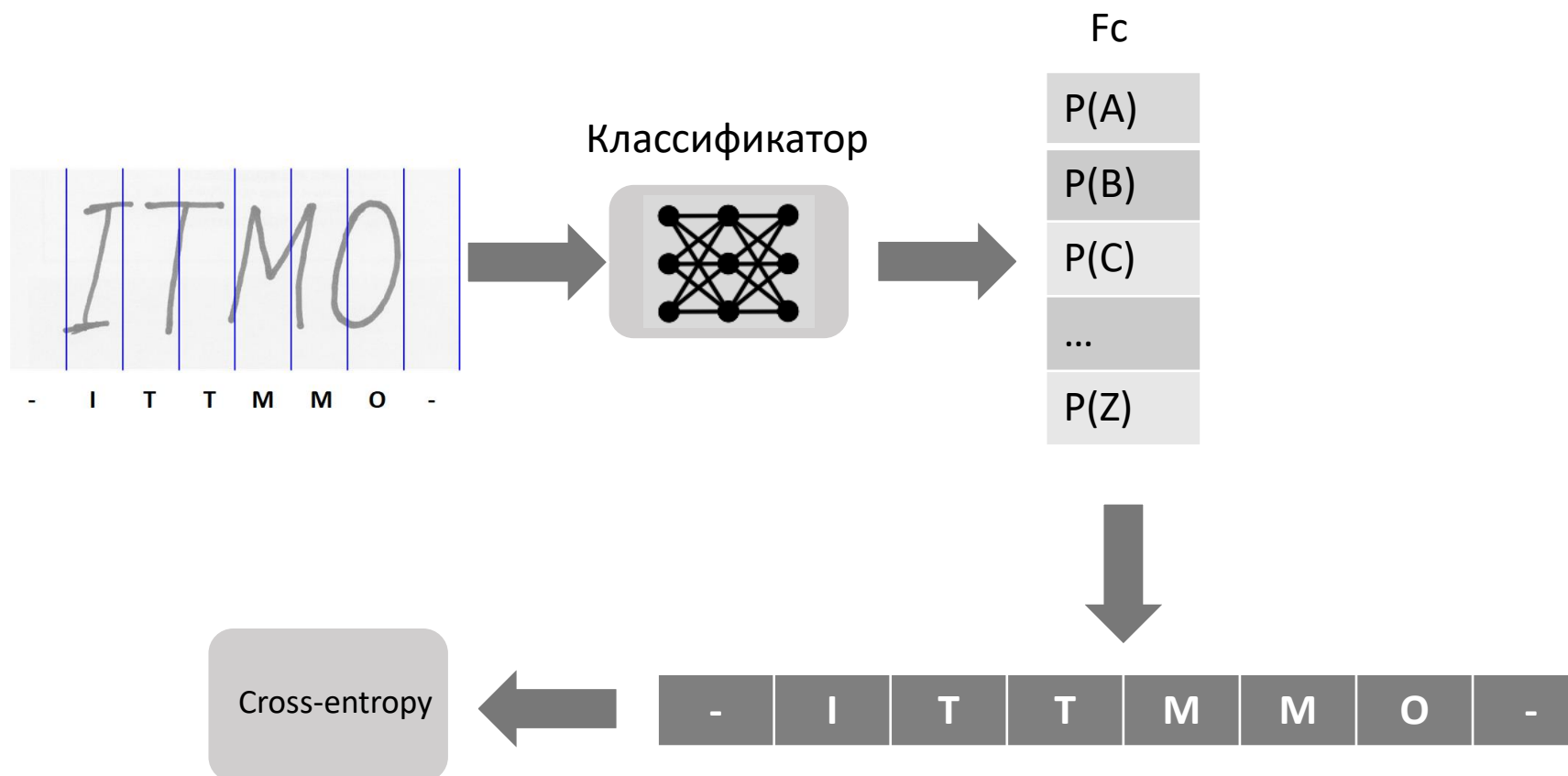
Какие еще есть идеи?



Какие еще есть идеи?



# Как это будет работать



# CTC Loss

## Connectionist Temporal Classification

- Решение seq2seq задач
- Не требуется выравнивание данных (Weakly aligned data)
- Задачи ASR / OCR

# Что мы хотим получить?

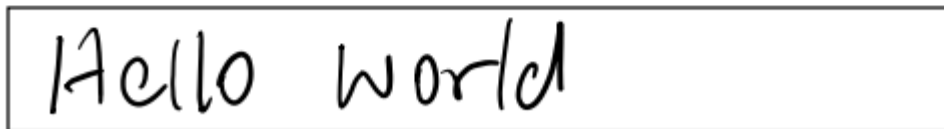
$$X \rightarrow Y$$

- Speech recognition



“Hello world”

- Optical character recognition (OCR)



“Hello world”

# Компоненты системы с CTC Loss

1. Visual feature extraction (CNN)
2. Sequential modeling based on visual feature sequence (RNN)
3. CTC layer to map input sequence (visual feature sequence) to output sequence (character sequence)

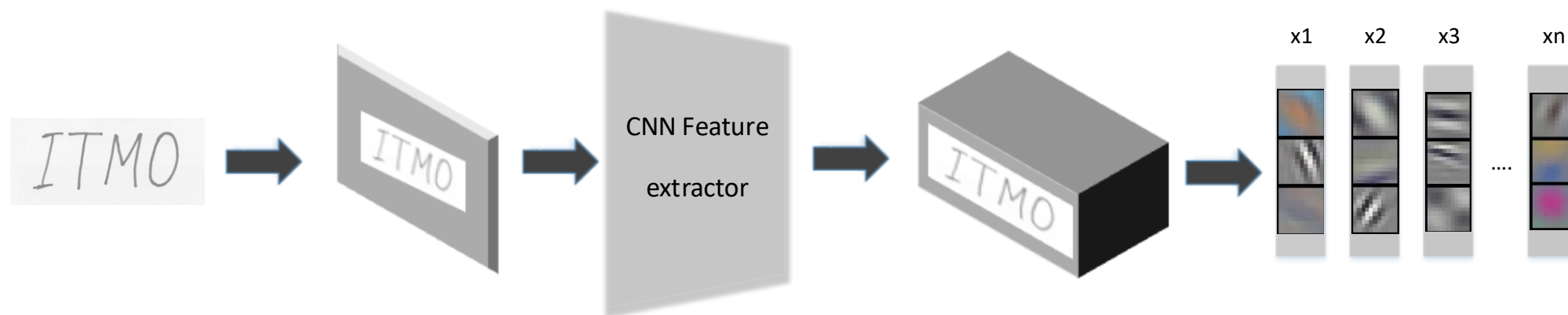
# Формальности

- Конечный алфавит  $A = \{a, b, c \dots z, -\}$
- Целевая последовательность  $Y = (a, b, c)$
- $X$  - изображение
- Выход нейронной сети  $f_{\theta}(X) = M$

$$M = \begin{pmatrix} P(a)_{t1} & \cdots & P(a)_{tn} \\ \vdots & \ddots & \vdots \\ P(-)_{t1} & \cdots & P(-)_{tn} \end{pmatrix}$$

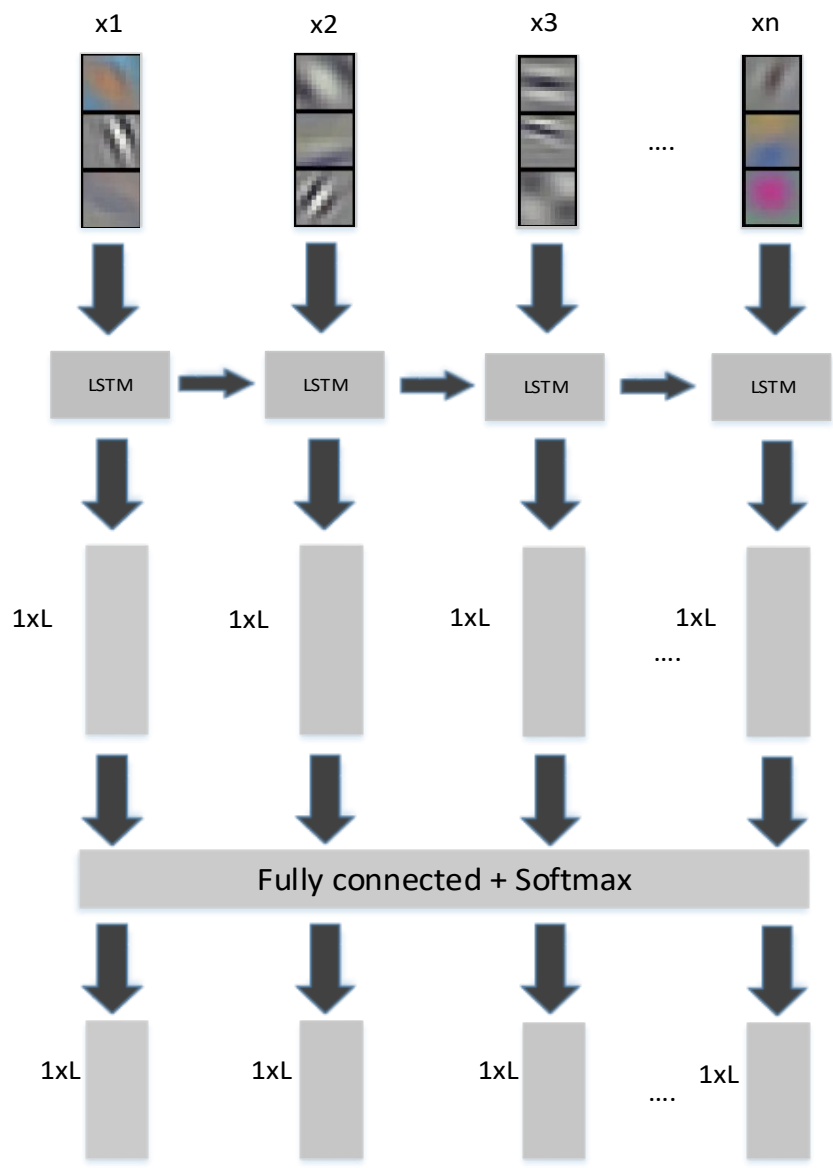
# Типовая архитектура

## Visual feature extraction



# Типовая архитектура

Sequential modeling based on visual feature sequence (RNN)

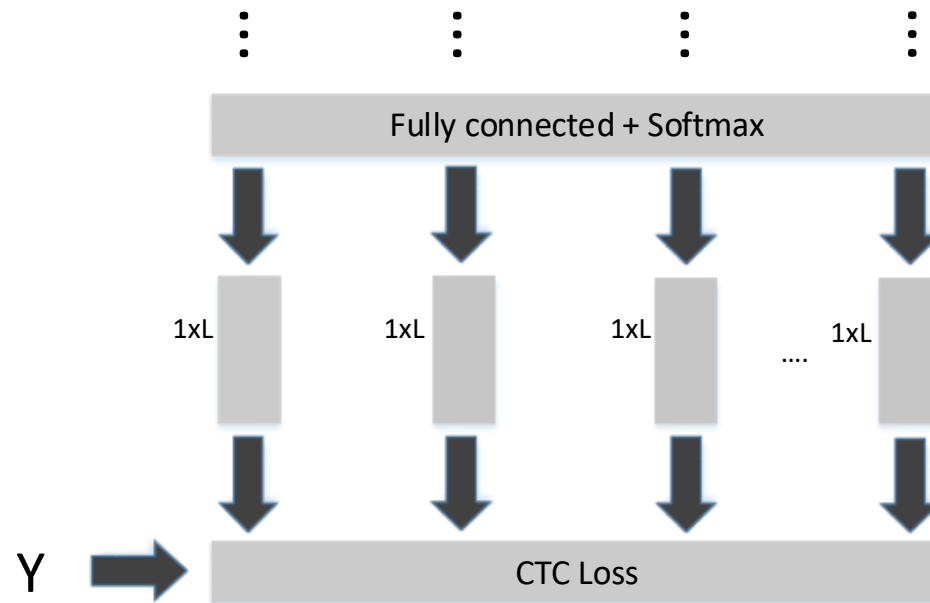


	x1	x2	x3	...	xn
P(A)	0.01	0.1	0.05	0.1	0.009
P(B)	0.2	0.05	0.07	0.005	0.01
...	0.8	0.006	0.005	0.001	0.3
P(-)	0.03	0.006	0.22	0.2	0.004

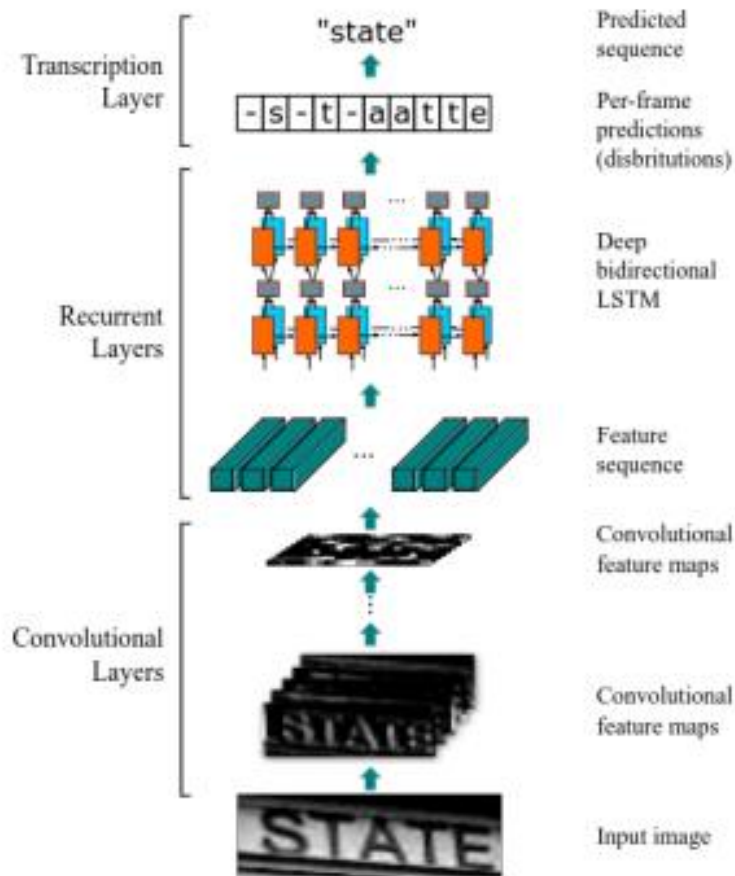


# Типовая архитектура

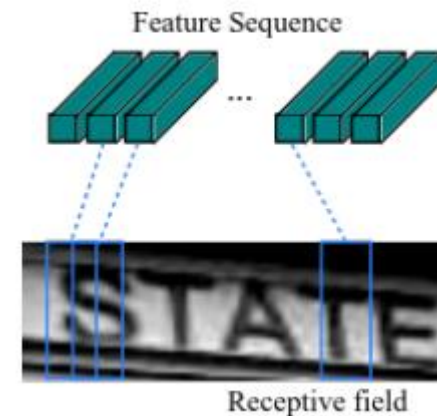
CTC layer to map input to output sequence



# An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition



Архитектура



Признаки строго соответствуют участку изображения

# Пример

	x1	x2	x3	x4	x5	x6	x7	x8
P(I)	0.8	0.7	0.1	0.2	0.0258	0.175	0.1	0.075
P(M)	0.05	0.075	0.1	0.13	0.0258	0.3	0.1	0.05
P(O)	0.05	0.075	0.1	0.13	0.0258	0.175	0.1	0.0416
P(T)	0.05	0.075	0.1	0.4	0.0258	0.175	0.1	0.0416
P(-)	0.05	0.075	0.6	0.13	0.5	0.175	0.6	0.0416



[ I I - T - M - O ]



"I-T-M-O"



"ITMO"

# Blank

- Blank – спецсимвол, для разделения букв (не пробел!)

Intellectual Information Technologies → IIT → [ I I T ] → "IT"

# CTC Loss

## Собираем ответ

	x1	x2	x3	x4	x5	x6	x7	x8
P(I)	0.8	0.7	0.1	0.2	0.0258	0.175	0.1	0.13
P(M)	0.05	0.075	0.1	0.13	0.0258	0.3	0.1	0.1
P(O)	0.05	0.075	0.1	0.13	0.0258	0.175	0.1	0.5
P(T)	0.05	0.075	0.1	0.4	0.0258	0.175	0.1	0.13
P(-)	0.05	0.075	0.6	0.13	0.5	0.175	0.6	0.13

$$P("II-T-M-O") = p_I^1 p_I^2 p_-^3 p_T^4 p_-^5 p_M^6 p_-^7 p_O^8$$

"II-T-M-O" – путь

# CTC Loss

## Примеры путей

$$P("I\text{--}T\text{--}M\text{--}O") = p_I^1 p_I^2 p_{-}^3 p_T^4 p_{-}^5 p_M^6 p_{-}^7 p_O^8 \rightarrow \text{"ITMO"}$$

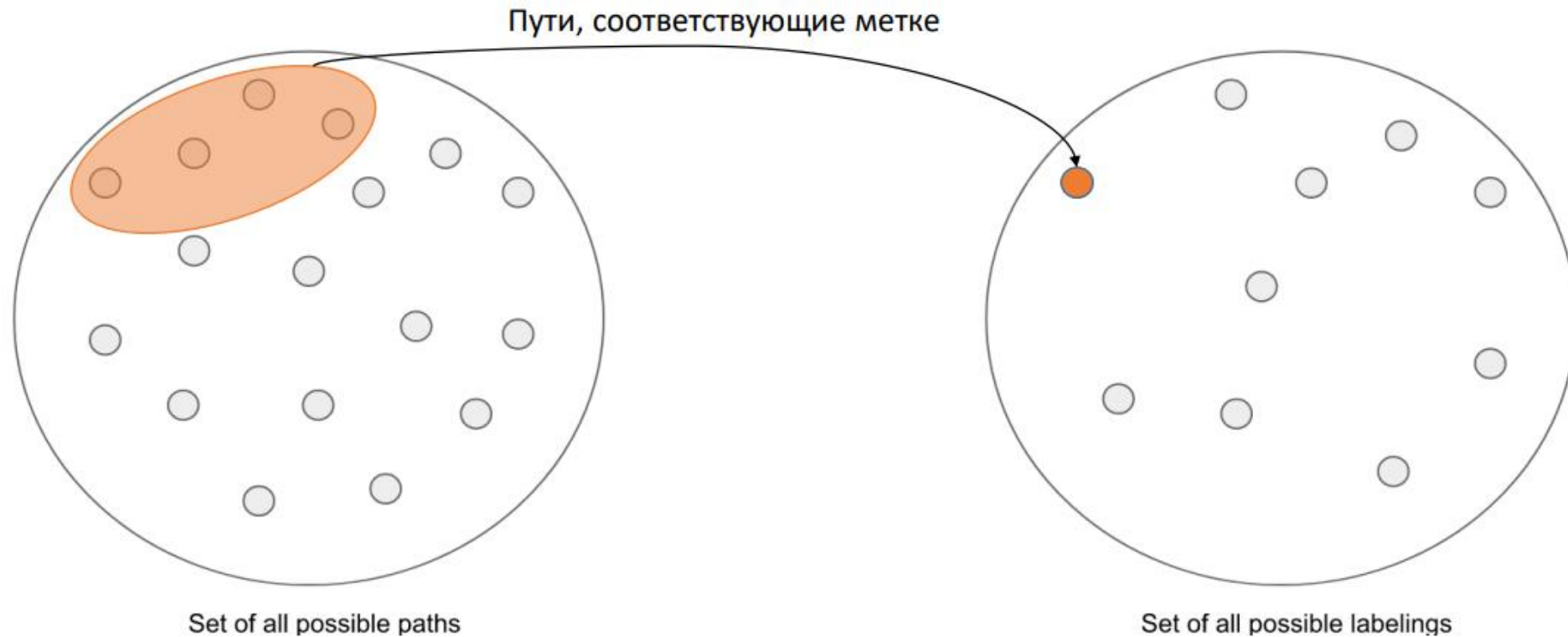
$$P("IT\text{--}MM\text{--}\text{--}O") = p_I^1 p_T^2 p_{-}^3 p_M^4 p_M^5 p_{-}^6 p_{-}^7 p_O^8 \rightarrow \text{"ITMO"}$$

# CTC Loss

## Примеры путей

$$P("I-I-T-M-O") = p_I^1 p_I^2 p_T^3 p_T^4 p_M^5 p_M^6 p_O^7 p_O^8 \rightarrow "ITMO"$$

$$P("IT-MM--O") = p_I^1 p_T^2 p_T^3 p_M^4 p_M^5 p_-^6 p_-^7 p_O^8 \rightarrow "ITMO"$$



# CTC Loss

## Вероятность лейблинга $Y$

$$p(y|x) = \sum_{\pi \in B^{-1}(y)} p(\pi|x)$$

$\pi$  — путь

$x$  — изображение

$y$  — лейбел (последовательность)

$B$  — функция сжатия

Вероятность лейблинга — сумма вероятностей всех путей, которые ведут в данный лейбел



# CTC Loss

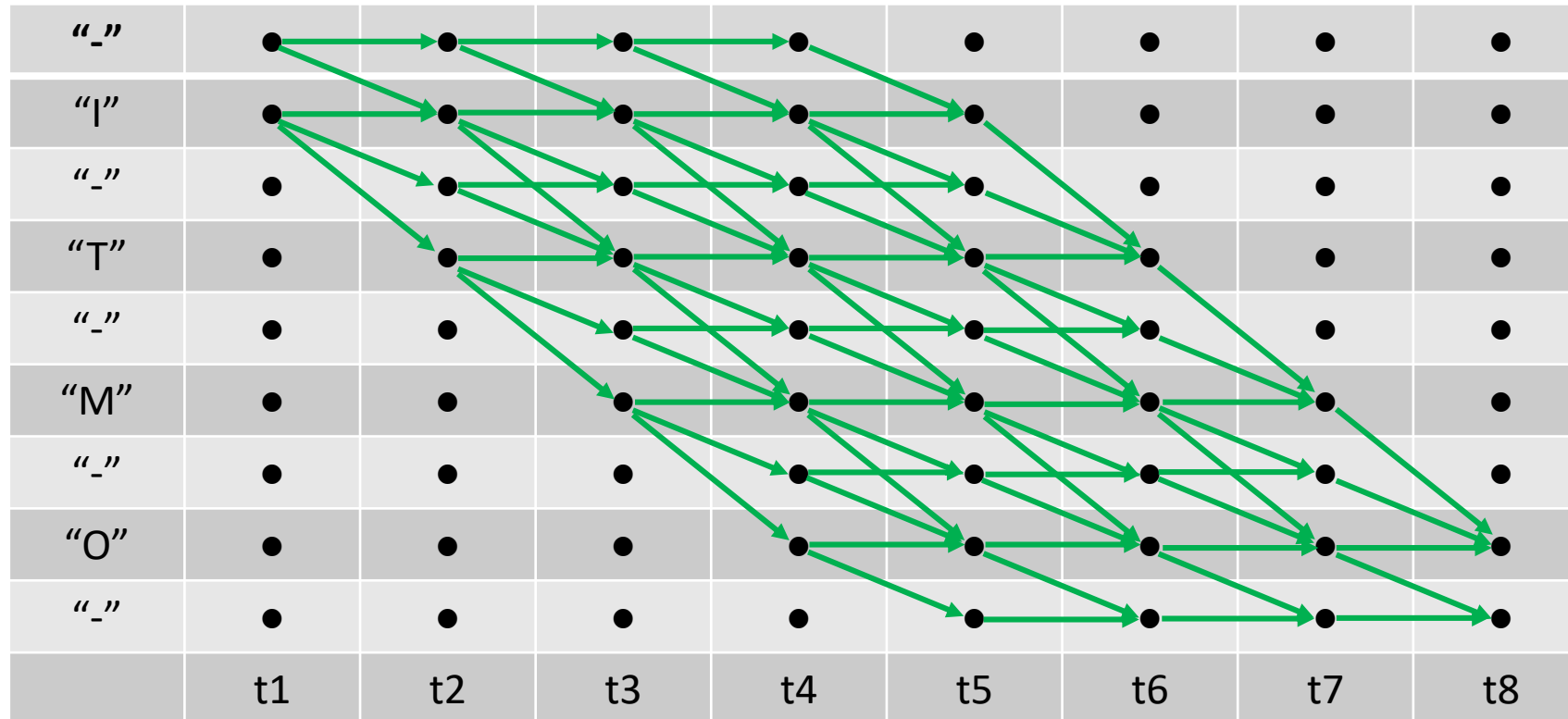
$$CTC\ Loss = -\ln P("ITMO")$$

$$N_{paths} = |A|^k$$

$k$  — количество шагов  $RNN$

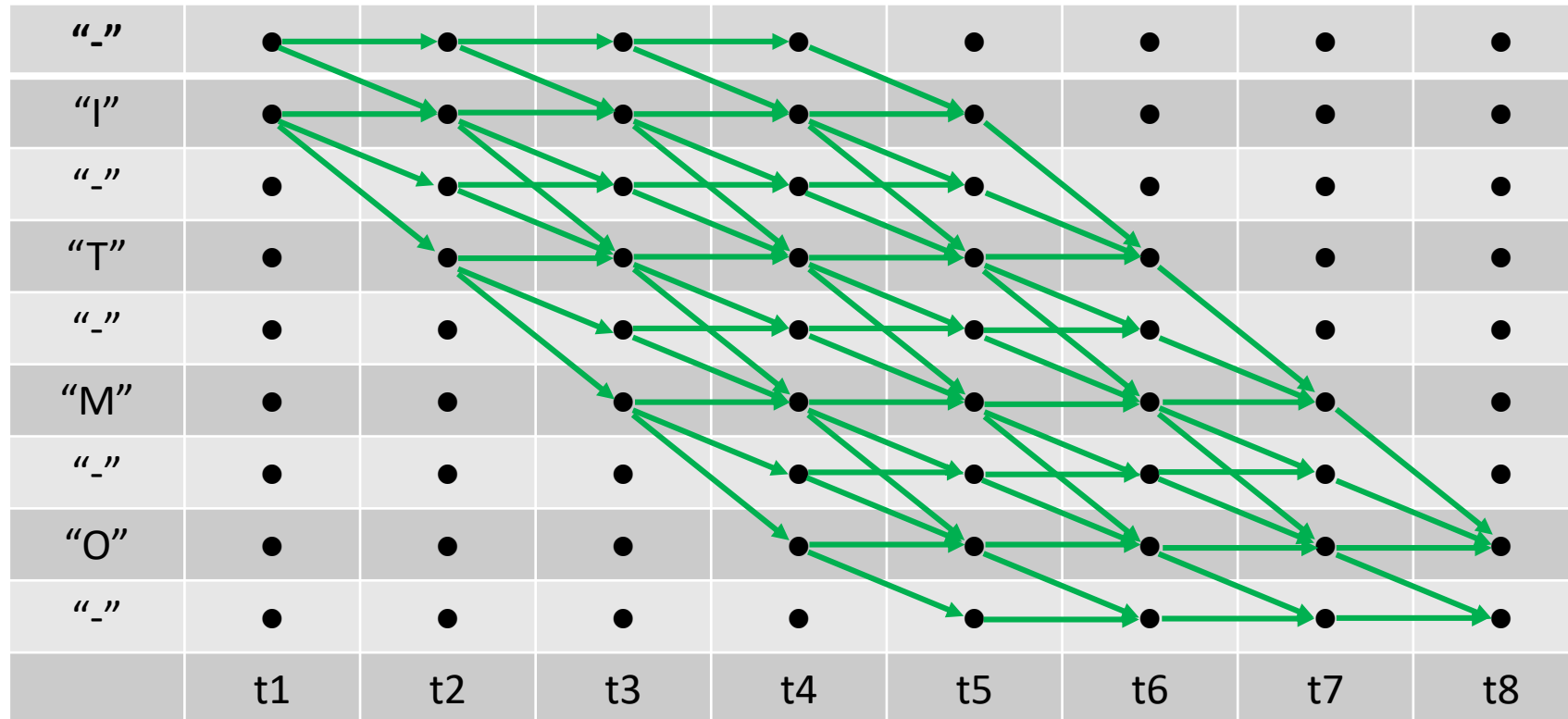
# CTC Loss

## Как считать



# CTC Loss

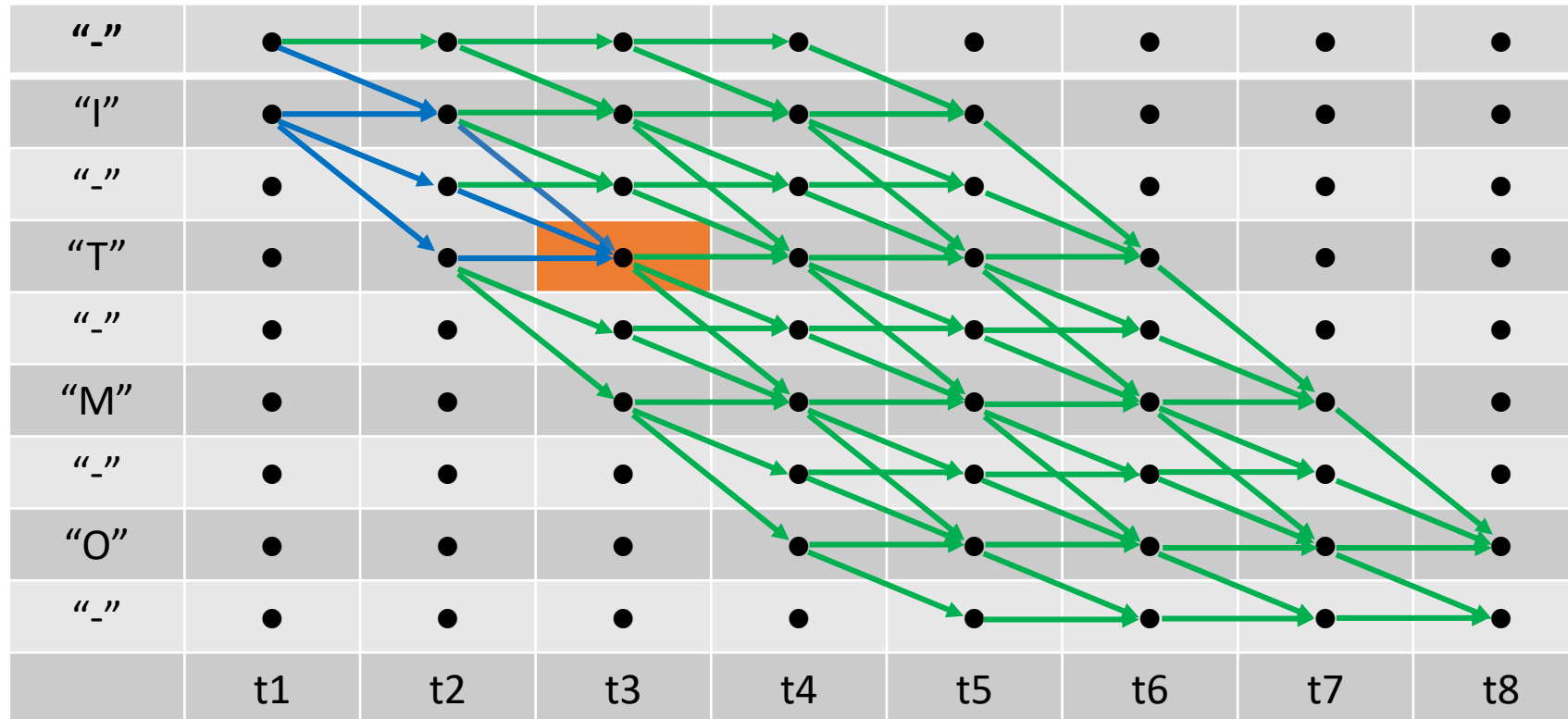
## Как считать



$\alpha_t(s)$  — суммарная вероятность всех подпутей для символа с индексом  $s$  в момент времени  $t$

# CTC Loss

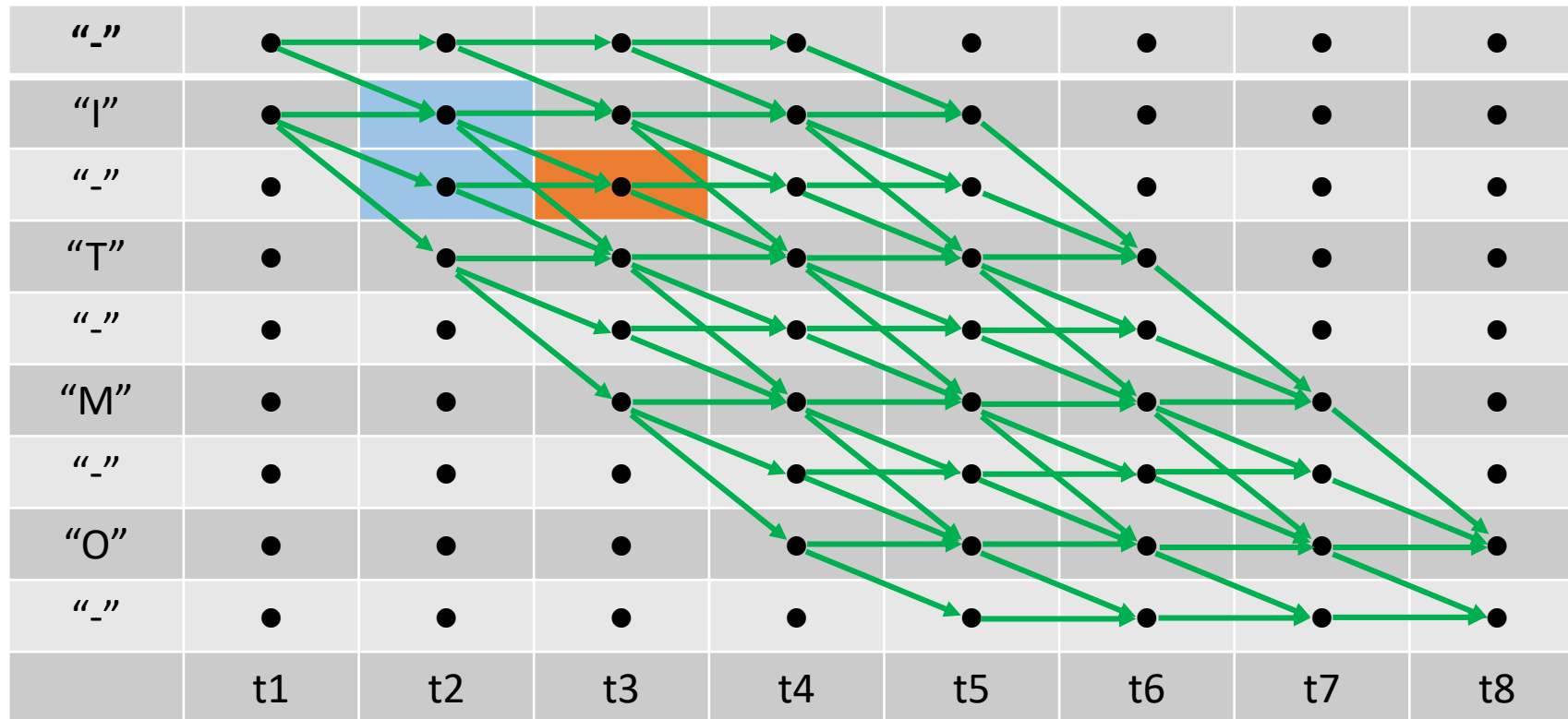
## Как считать



$$\alpha_3(4) = p(" - IT") + p("IIT") + p("I - T") + p("ITT")$$

# Подсчет вероятностей для произвольной ячейки

1. В текущий момент времени прогнозируем blank

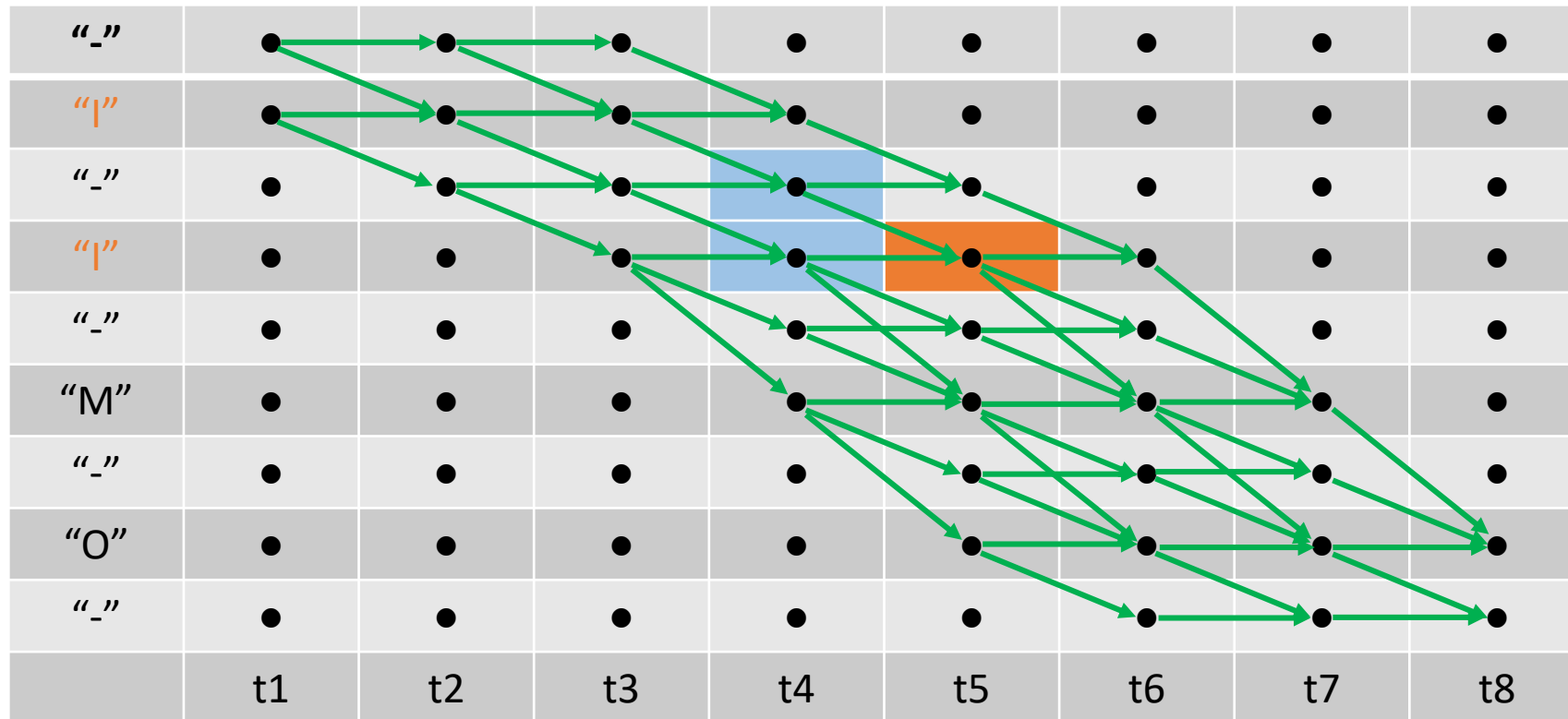


$$\alpha_t(s) = (\alpha_{t-1}(s) + \alpha_{t-1}(s-1)) \cdot p_{seq(s)}^t$$

$$\alpha_3(3) = (\alpha_2(3) + \alpha_2(2))p_-^3$$

# Подсчет вероятностей для произвольной ячейки

2. Символ, аналогичный тому, который был два шага назад

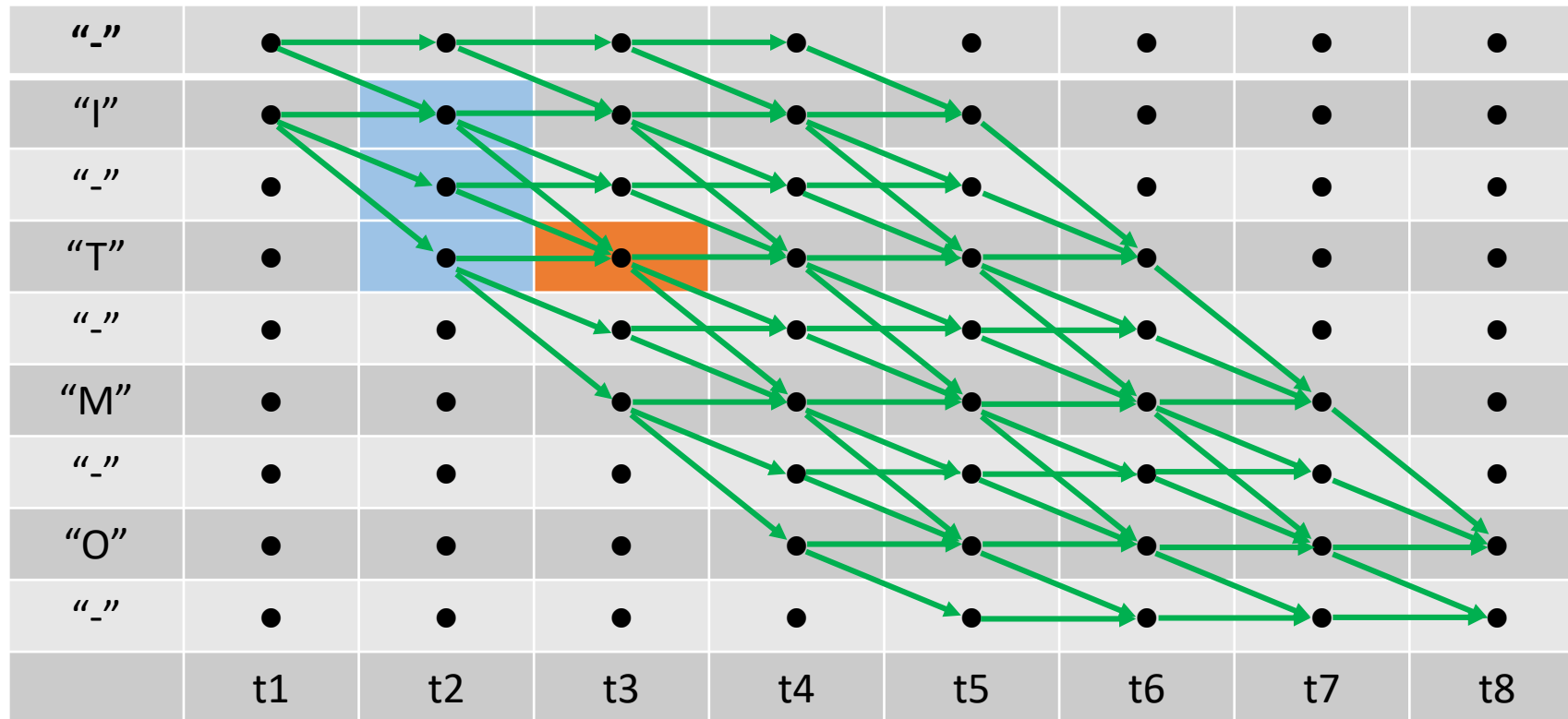


$$\alpha_t(s) = (\alpha_{t-1}(s) + \alpha_{t-1}(s-1)) \cdot p_{seq(s)}^t$$

$$\alpha_5(4) = (\alpha_4(4) + \alpha_4(3)) \cdot p_I^5$$

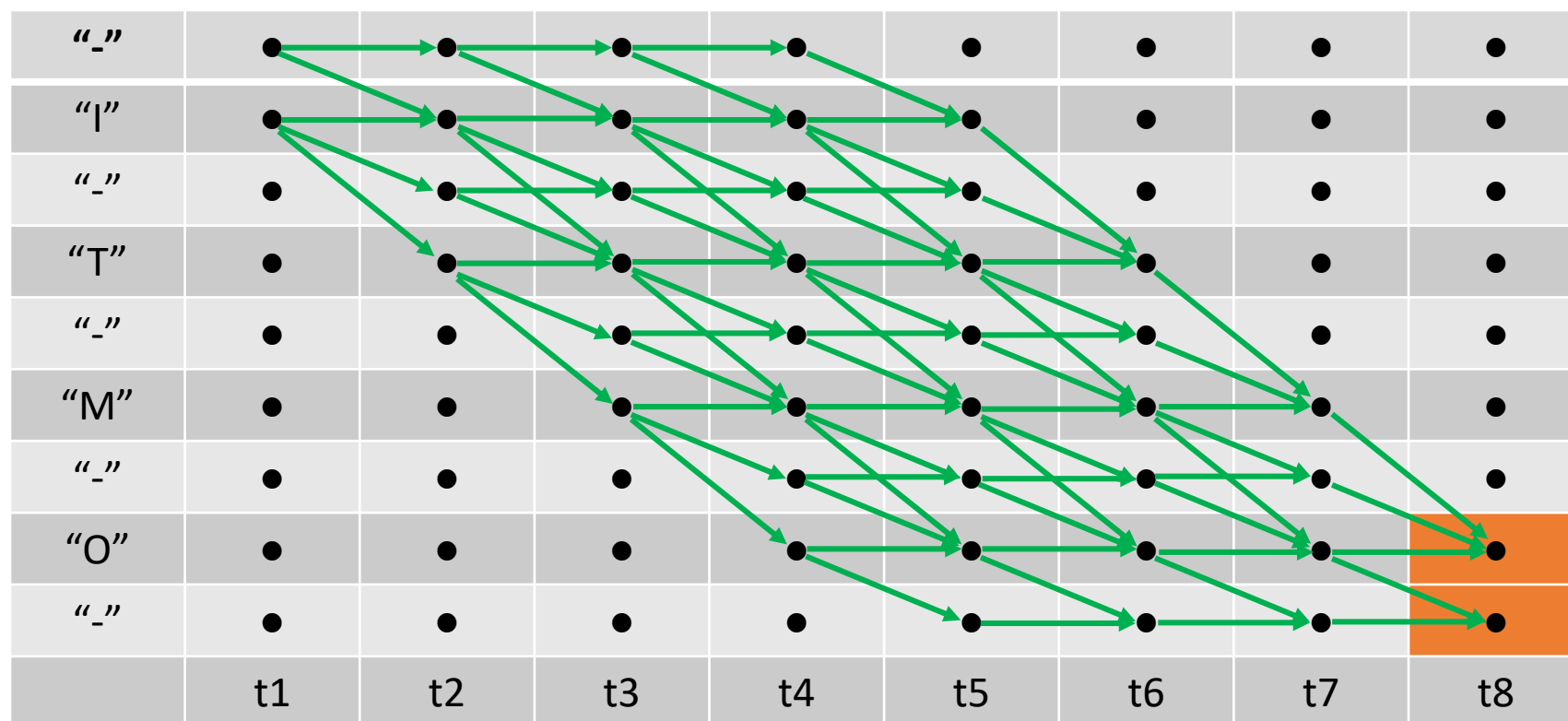
# Подсчет вероятностей для произвольной ячейки

## 3. Все остальные случаи



$$\alpha_t(s) = (\alpha_{t-1}(s) + \alpha_{t-1}(s-1) + \alpha_{t-1}(s-2)) \cdot p_{seq(s)}^t \quad \alpha_3(4) = (\alpha_2(4) + \alpha_2(3) + \alpha_2(2)) \cdot p_T^3$$

# Итоговая вероятность



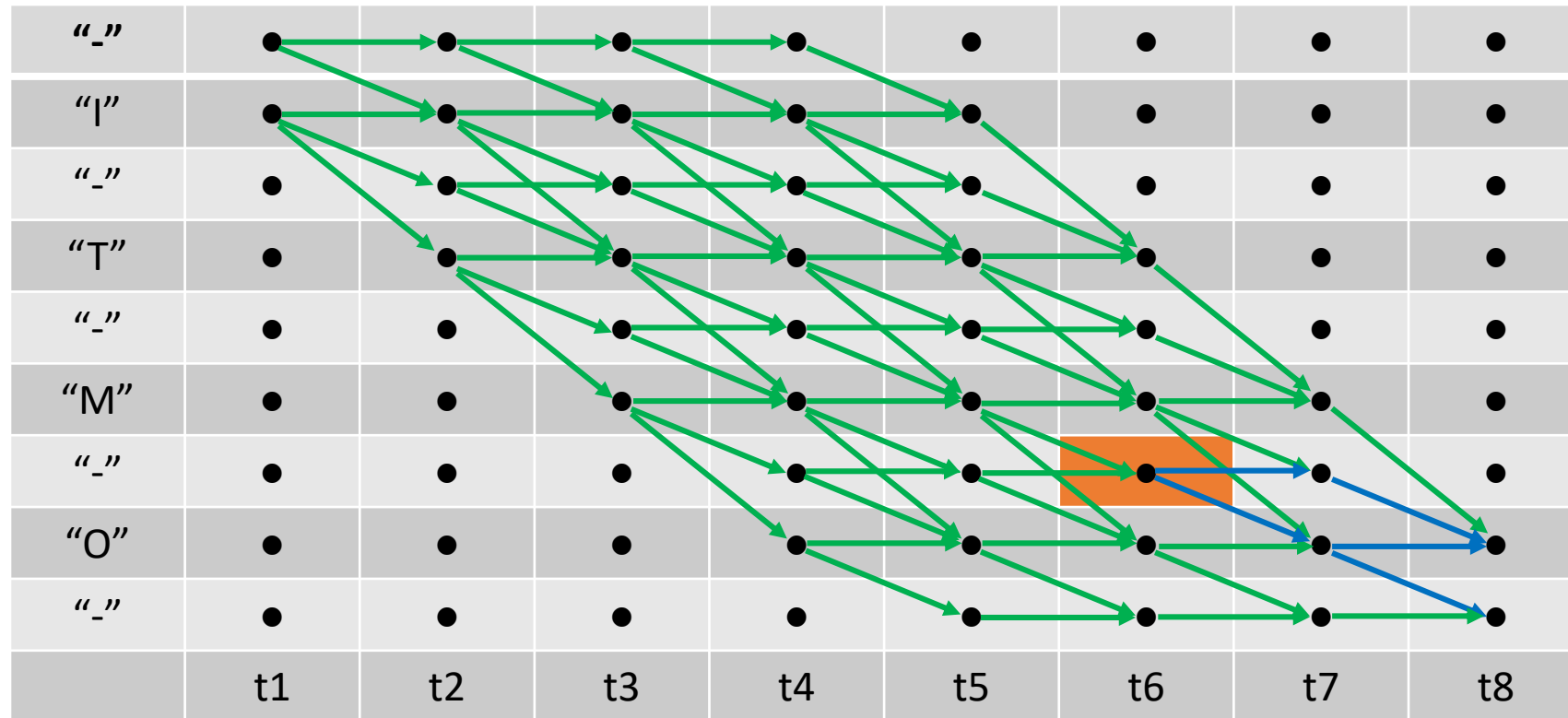
Forward pass done!

$$p("ITMO") = \alpha_8(8) + \alpha_8(9)$$

$$CTC\ loss = -\ln p("ITMO") = -\ln(\alpha_8(8) + \alpha_8(9))$$



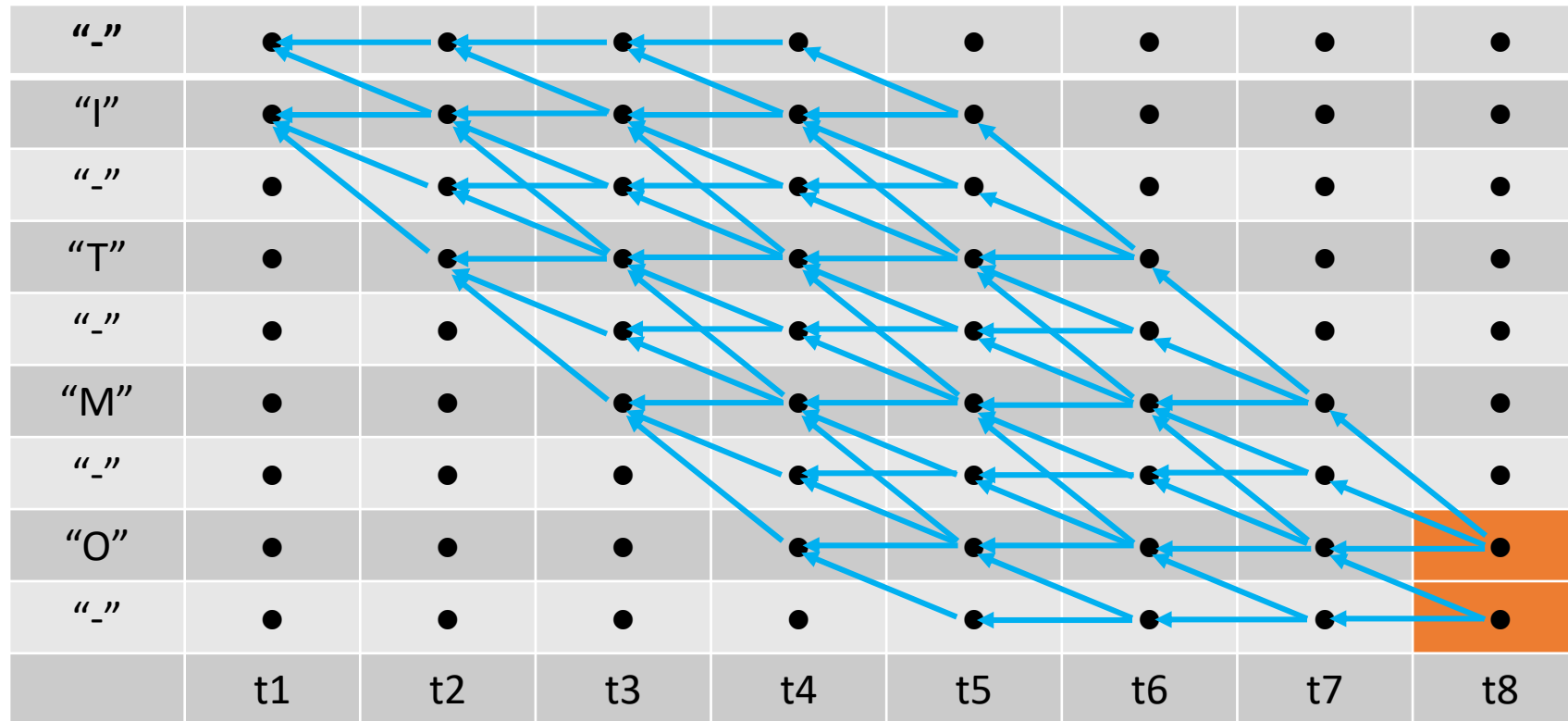
# Backward pass



$\beta_t(s)$  – вероятность всех подпутей, которые приводят к *ground true* лейбел

$$\beta_6(7) = p(" - - O") + p(" - O O") + p(" - O - ")$$

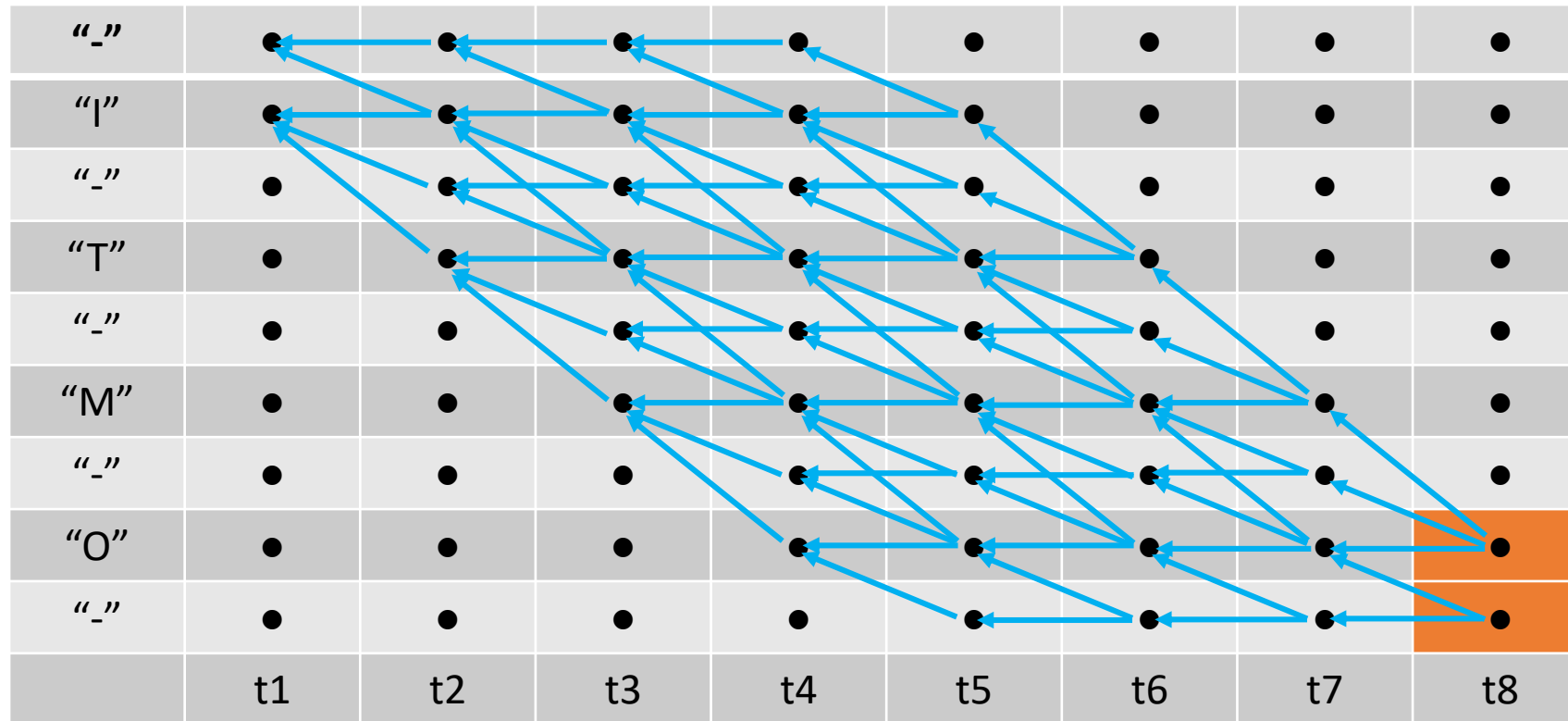
# Backward pass



Для вычисления  $\beta_t(s)$ :

1. Изменяем связи на противоположные
2. Применяем ту же логику, что и для подсчета  $\alpha_t(s)$

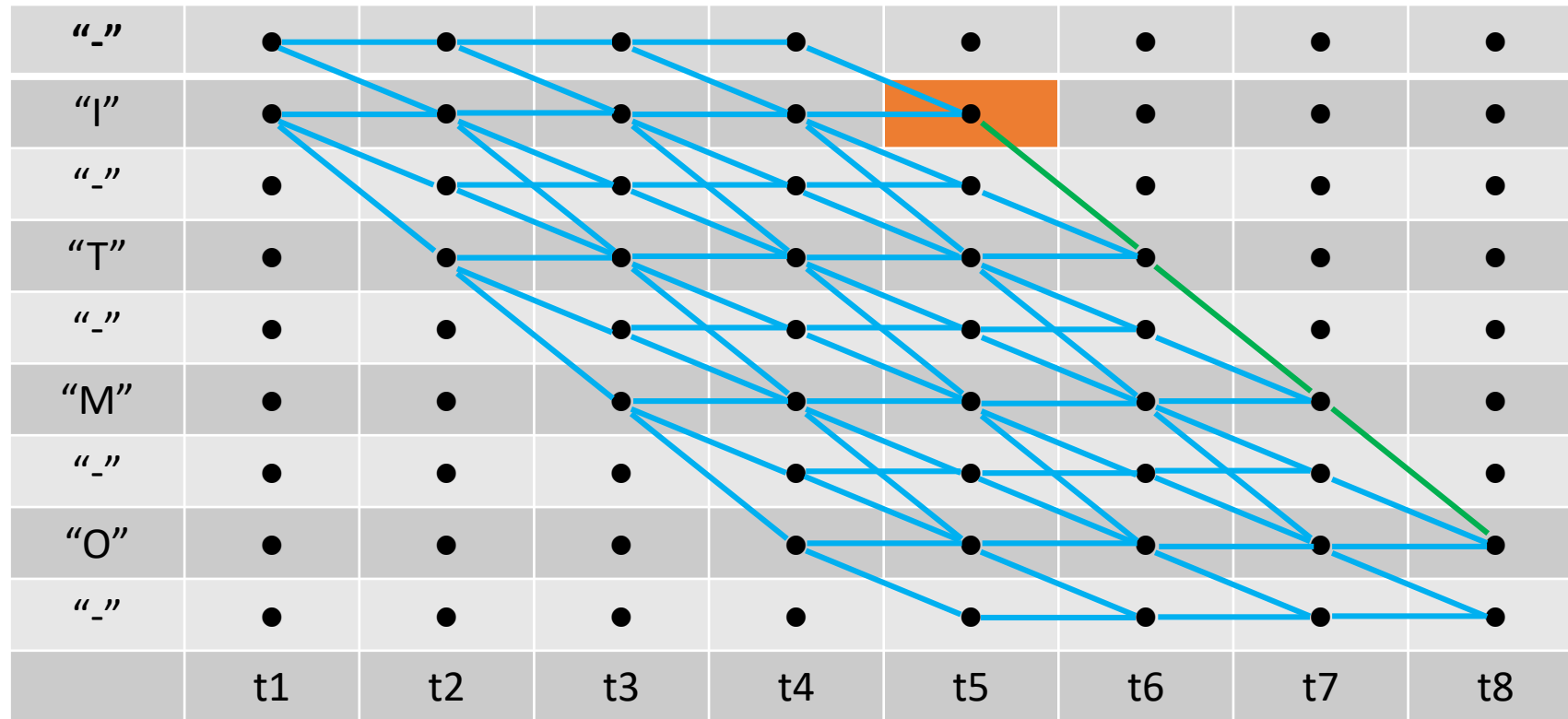
# Backward pass



Для вычисления  $\beta_t(s)$ :

1. Изменяем связи на противоположные
2. Применяем ту же логику, что и для подсчета  $\alpha_t(s)$

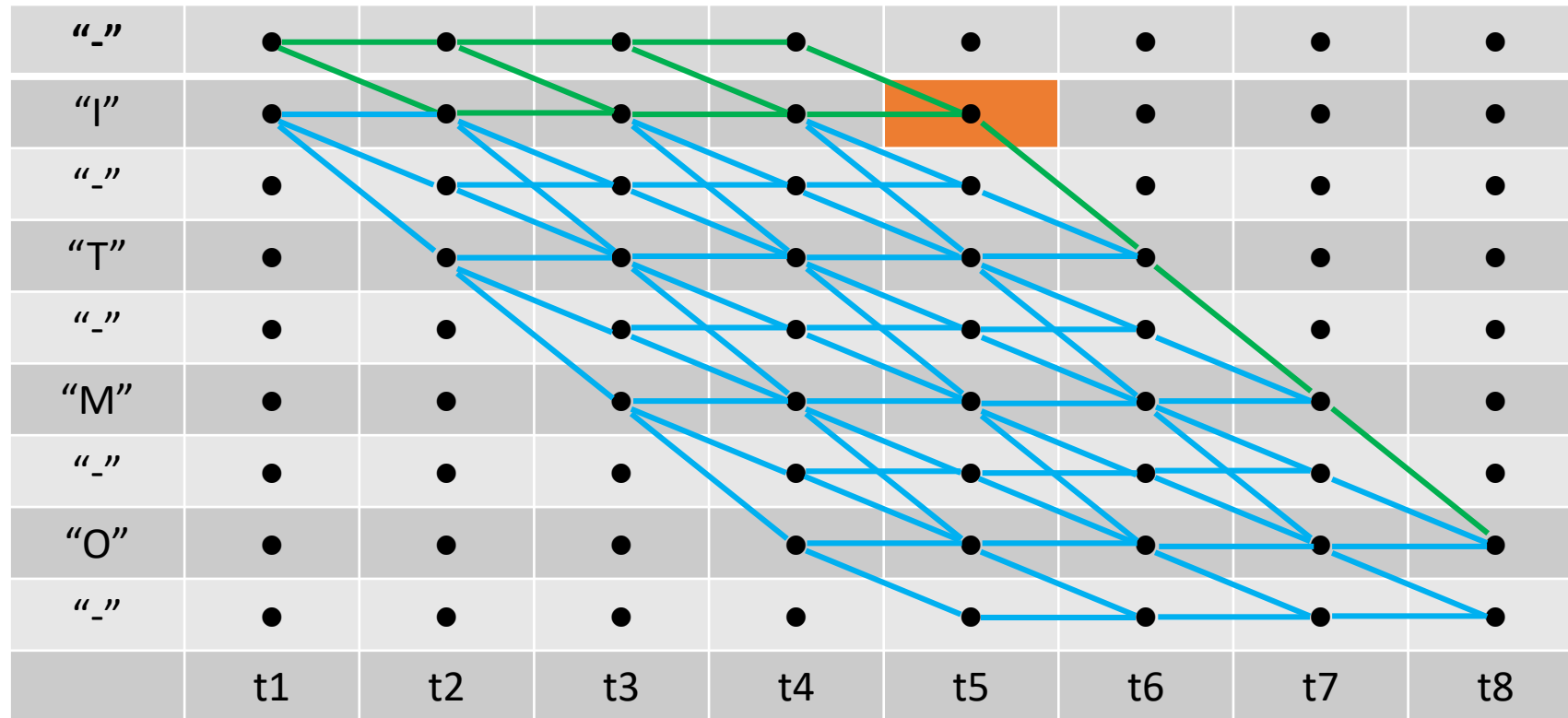
# Вероятность пути через ячейку



$$\alpha_5(2) = p(\text{" - - - - I"}) + p(\text{" - - - I I"}) + p(\text{" - - I I I"}) + p(\text{" - I I I I"}) + p(\text{" I I I I I"})$$

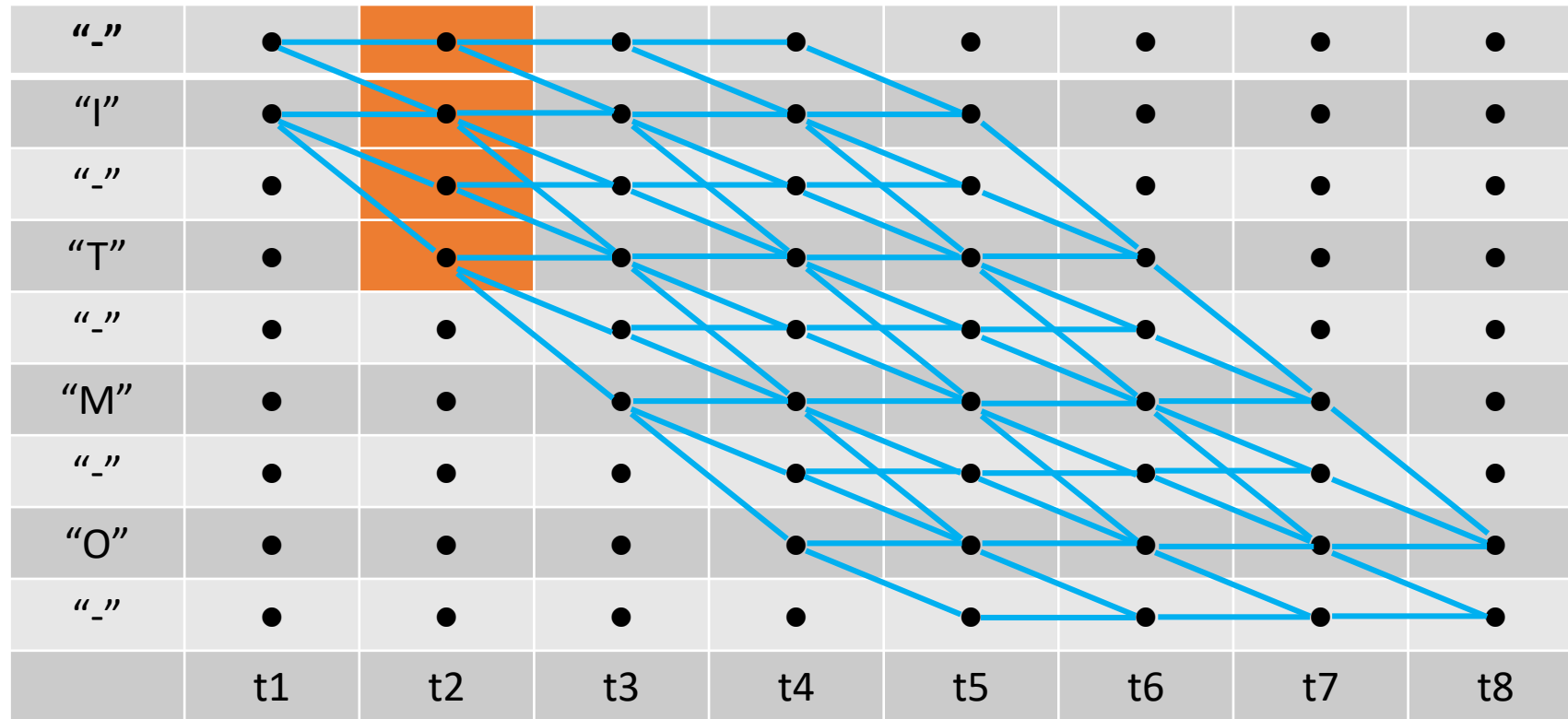
$$\beta_5(2) = p(\text{"ITMO"}) = p_I^5 \cdot p_T^6 \cdot p_M^7 \cdot p_O^8$$

# Вероятность пути через ячейку



$\frac{\alpha_5(2) \cdot \beta_5(2)}{p_I^5}$  — сумма вероятностей всех путей, проходящих через ячейку

# Вероятность лейблинга в момент времени



$$p("ITMO") = \sum_{s=1}^4 \frac{\alpha_2(s) \cdot \beta_2(s)}{p_{seq(s)}^2}$$

# Backprop

$$\frac{\partial(-\ln(p(\text{"ITMO"})))}{\partial y_k^t} = -\frac{1}{p(\text{"ITMO"})} \cdot \frac{\partial p(\text{"ITMO"})}{\partial y_k^t}$$

$$\frac{\partial(-\ln(p(\text{"ITMO"})))}{\partial y_k^t} = -\frac{1}{y_k^{t^2}} \cdot \sum_{s:seq(s)=k} \alpha_t(s) \cdot \beta_t(s)$$

# Заключение

Вычисляем  $\alpha$  для вычисления `cts loss`

*Вычисляем  $\beta$  – можно посчитать градиенты*



# Возможные улучшения

- Накрутить поверх распознавания Encoder-Decoder

Хороший пример по данной теме – соревнование 2020 года “**Digital Peter: recognition of Peter the Great's manuscripts**”

Baseline – модель с CTC loss, улучшение – приделать трансформер

[Ссылка на гитхаб соревнования](#)

# Еще более новые подходы

## AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Encoder – Transformer-based vision model  
Decoder – NLP model

Библиотека с реализацией  
[https://huggingface.co/docs/transformers/model\\_doc/vision-encoder-decoder](https://huggingface.co/docs/transformers/model_doc/vision-encoder-decoder)



# Полезные ссылки

- <https://www.youtube.com/watch?v=SAfJ6nP2rrI> (How to build end-to-end recognition system (Part 1): best practices [RU])
- <https://www.youtube.com/watch?v=eYIL4TMAeRI> (How to build end-to-end recognition system (Part 2): CTC Loss [RU])
- <https://www.youtube.com/watch?v=ZPNsYTs2Zx4> (Как нейронная сеть распознает текст? Лекция 1 по Advanced Computer Vision)
- <https://www.youtube.com/watch?v=F2ERzhLFeuo> (Распознавание текста. Обратное распространение через CTC Loss. Лекция 2 по Advanced Computer Vision)
- [https://cseweb.ucsd.edu/classes/wi19/cse291-g/student\\_presentations/CTC\\_OCR.pdf](https://cseweb.ucsd.edu/classes/wi19/cse291-g/student_presentations/CTC_OCR.pdf) (Connectionist Temporal Classification (CTC) with application to Optical Character Recognition)
- [https://www.audiolabs-erlangen.de/content/05-fau/professor/00-mueller/02-teaching/2021s\\_dla/2021\\_DLA-09\\_ZalkowMueller\\_CTC.pdf](https://www.audiolabs-erlangen.de/content/05-fau/professor/00-mueller/02-teaching/2021s_dla/2021_DLA-09_ZalkowMueller_CTC.pdf) (Connectionist Temporal Classification (CTC) Loss)
- [http://www.machinelearning.ru/wiki/images/c/c6/Digital\\_Signal\\_Processing%2C\\_lecture\\_7.pdf](http://www.machinelearning.ru/wiki/images/c/c6/Digital_Signal_Processing%2C_lecture_7.pdf) (Распознавание речи. Современные подходы)
- <https://ogunlao.github.io/blog/2020/07/17/breaking-down-ctc-loss.html> (Breaking down the CTC Loss)

Спасибо за внимание!