Word2vec

- word2vec architecture
- subsampling
- negative sampling
- GLOVE
- FastText
- ELMO

Seminar - ranking of messages

Lecture 1. Milana Shkhanukova, 2023

Recap

- В чем был главный недостаток TF-IDF и One-Hot encoding?

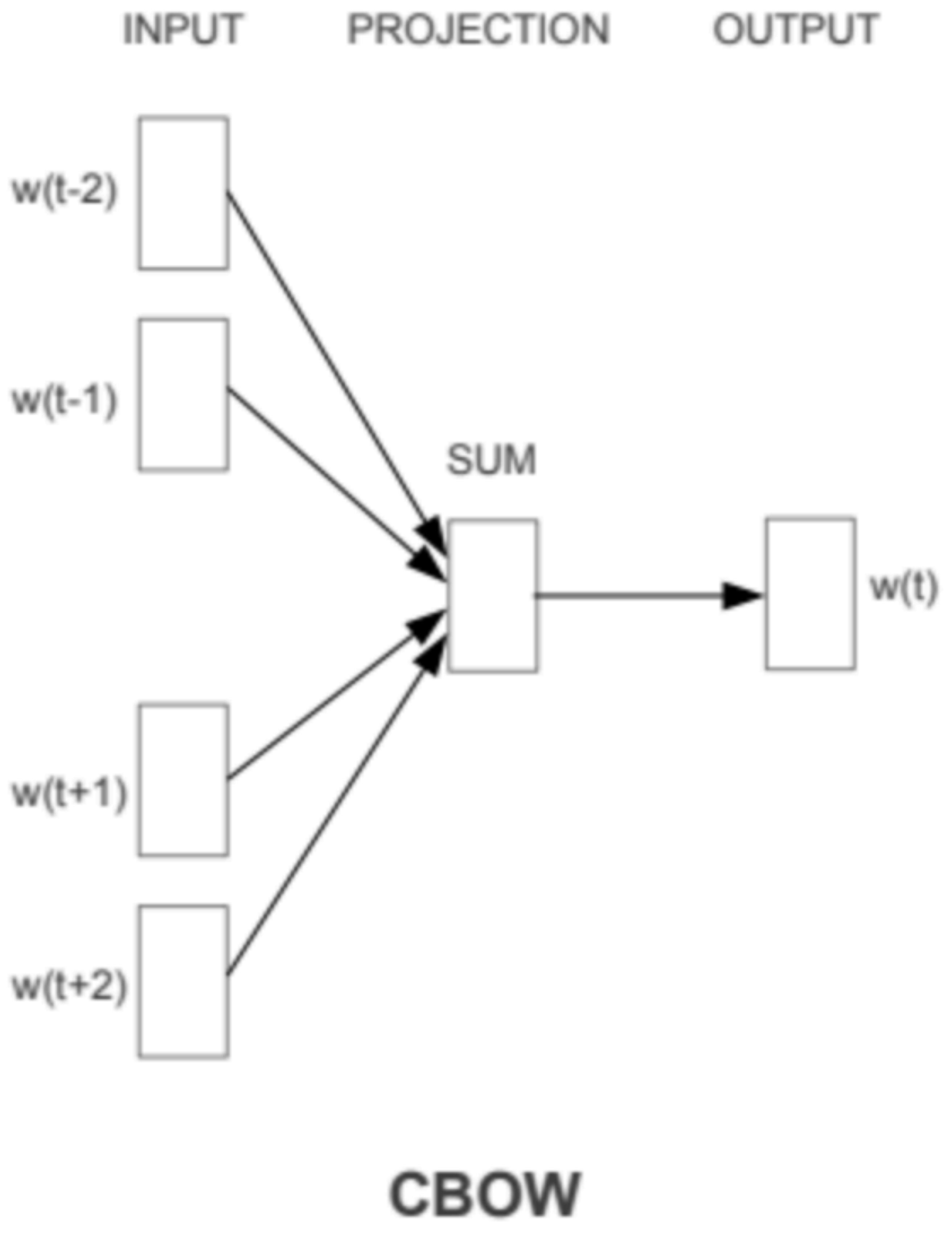
N-gram embeddings, CBOW

Problem? Our words appear together, it matters.

Guess the word by its context, CBOW – **Continuous Bag-of-Words**

Continuous - continuous vector representations to represent words.
Each word is represented as a dense vector in a continuous vector space.

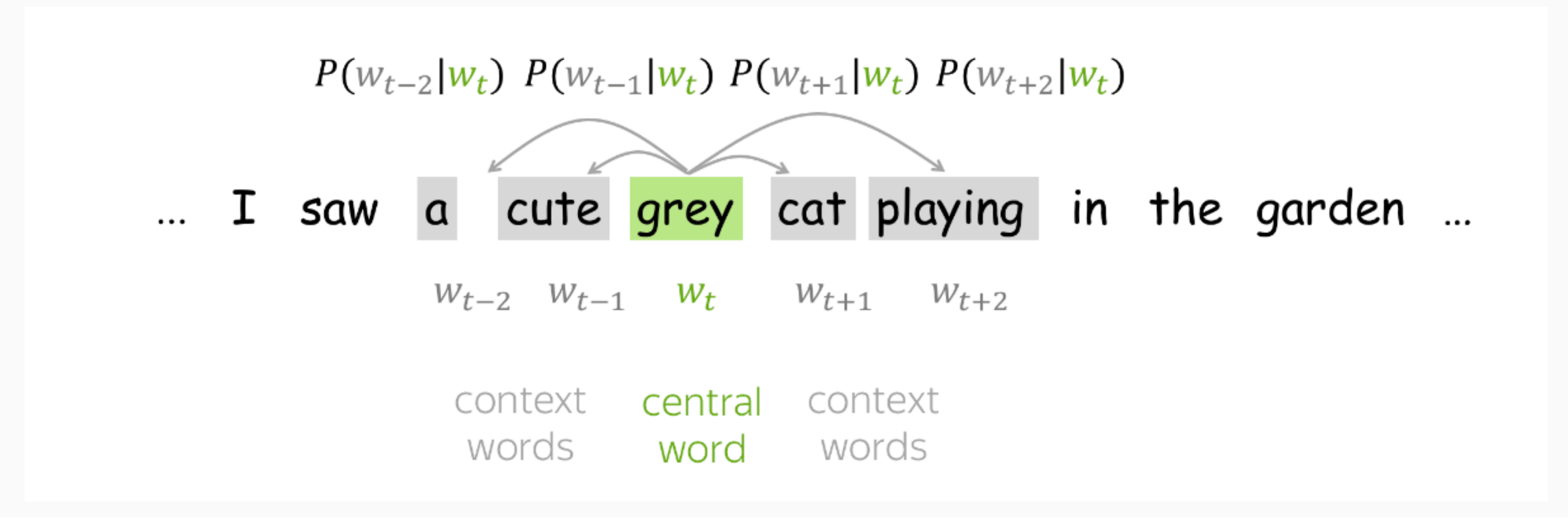
Окно - это количество токенов в одну сторону, на которое мы смотрим

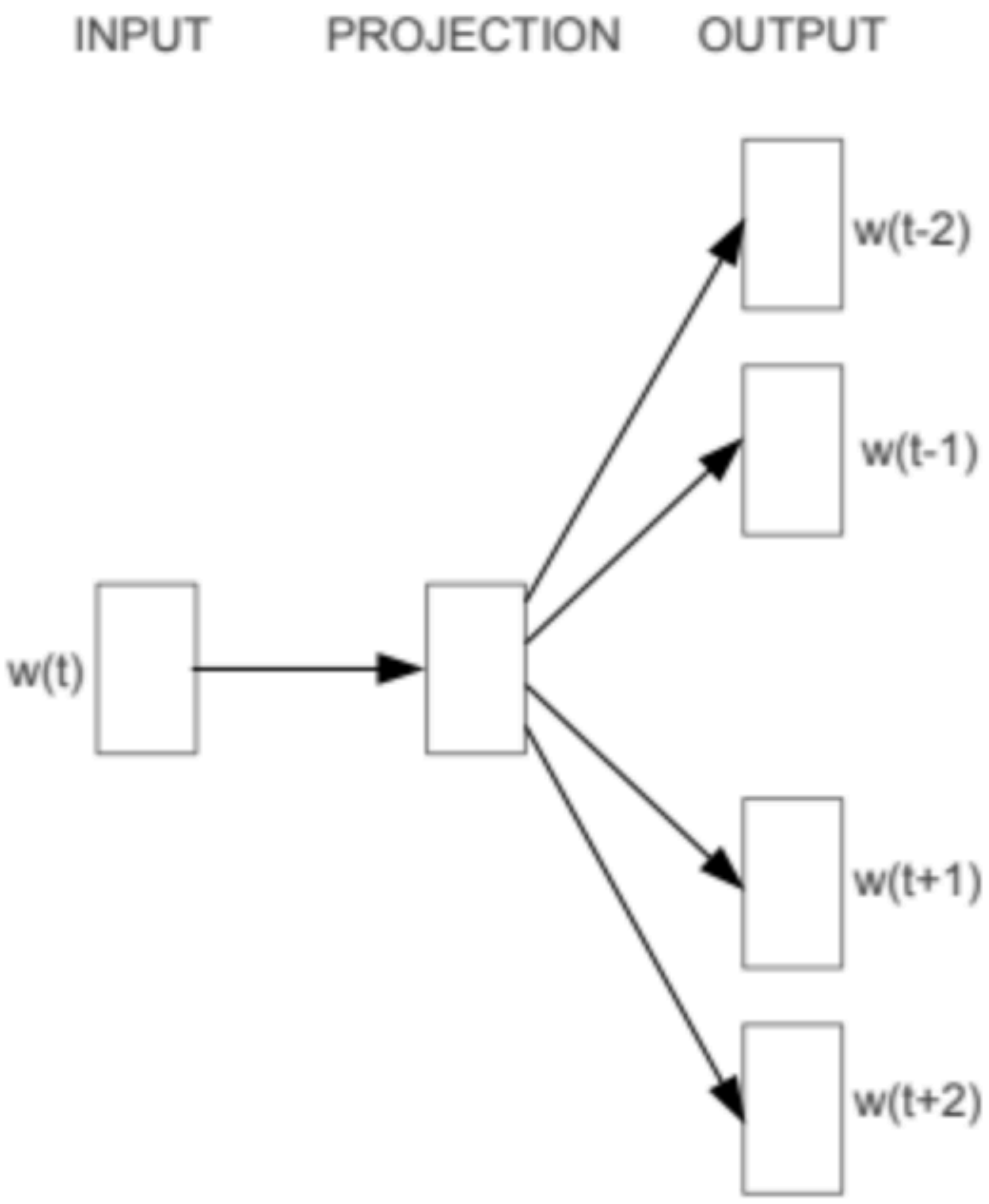


https://lena-voita.github.io/nlp_course/word_embeddings.html

N-gram embeddings, Skip-gram

Problem? Our words appear together, it matters.





Skip-gram

What is n?

Я люблю пить кофе по утрам одна

Пить люблю я кофе по утрам одна

N = 3

Контекст = пить кофе по

Контекст = я люблю пить + по утрам одна

N-window size

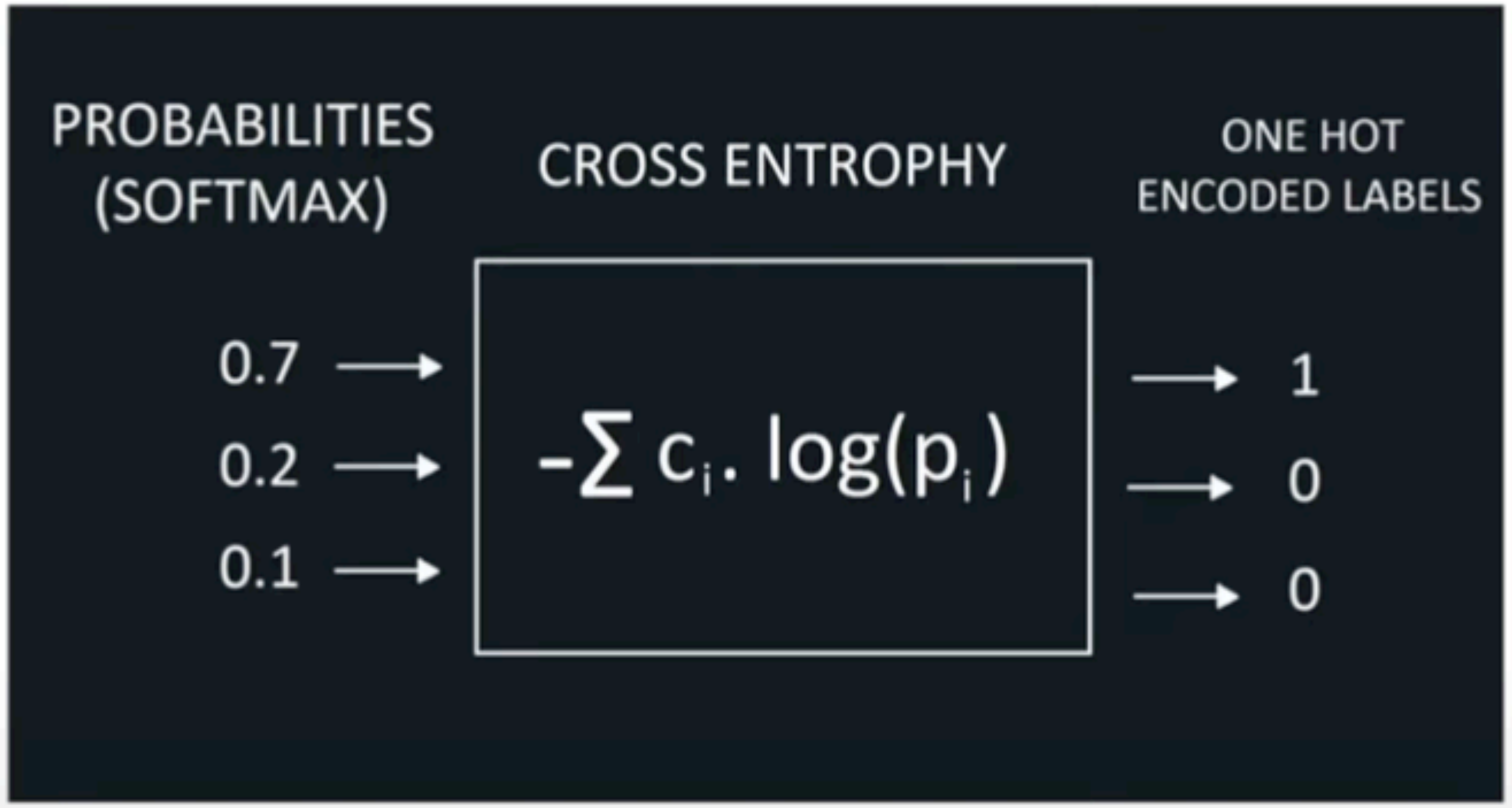
this paper notes that larger windows tend to produce more topical similarities (i.e. **dog**, **bark** and **leash** will be grouped together, as well as **walked**, **run** and **walking**), while smaller windows tend to produce more functional and syntactic similarities (i.e. **Poodle**, **Pitbull**, **Rottweiler**, or **walking**, **running**, **approaching**)

Some questions

- Are context words at different distances equally important? If not, how can we modify co-occurrence counts?
Контекст везде одинаковый?
- В какой модели у нас есть информация о порядке слов?

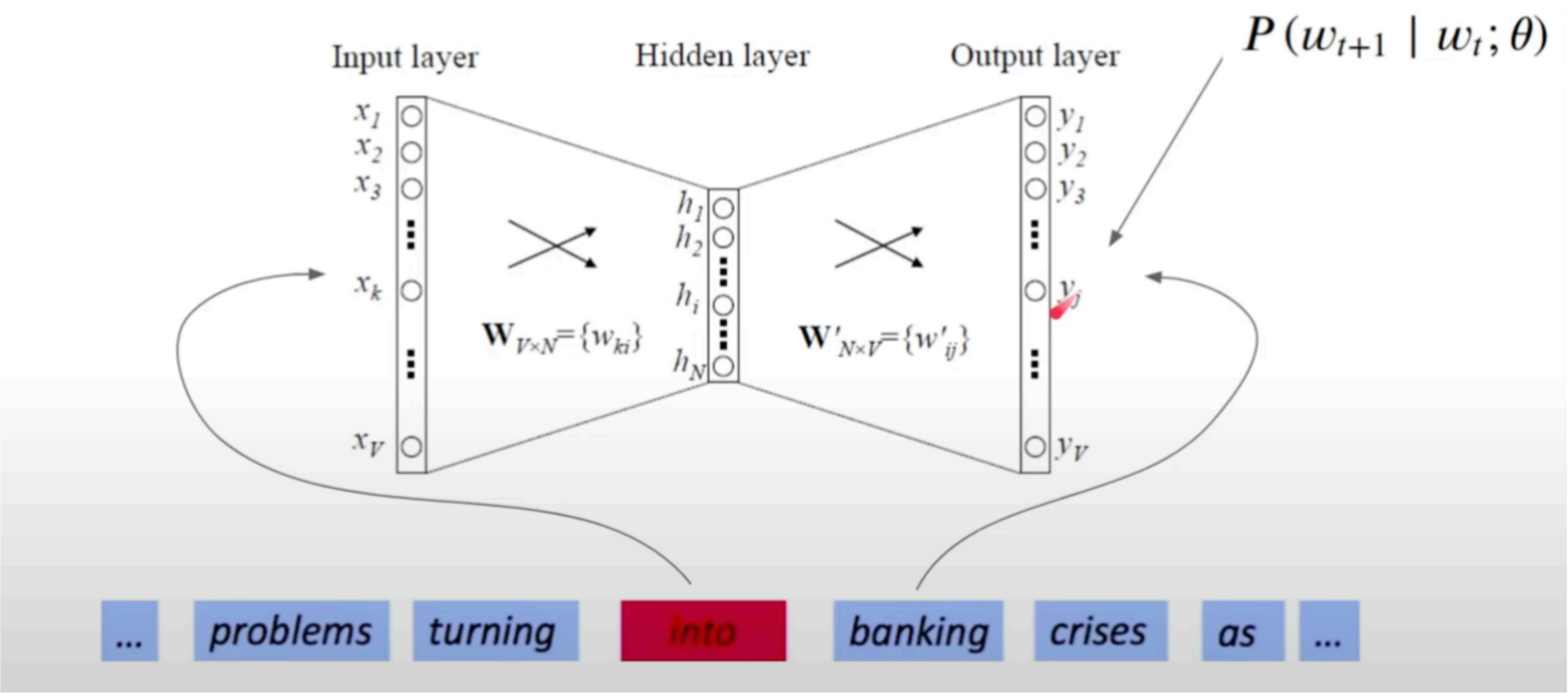
Cross entropy loss

p(x) - распределение, которое мы получили
q(x) - распределение истинное



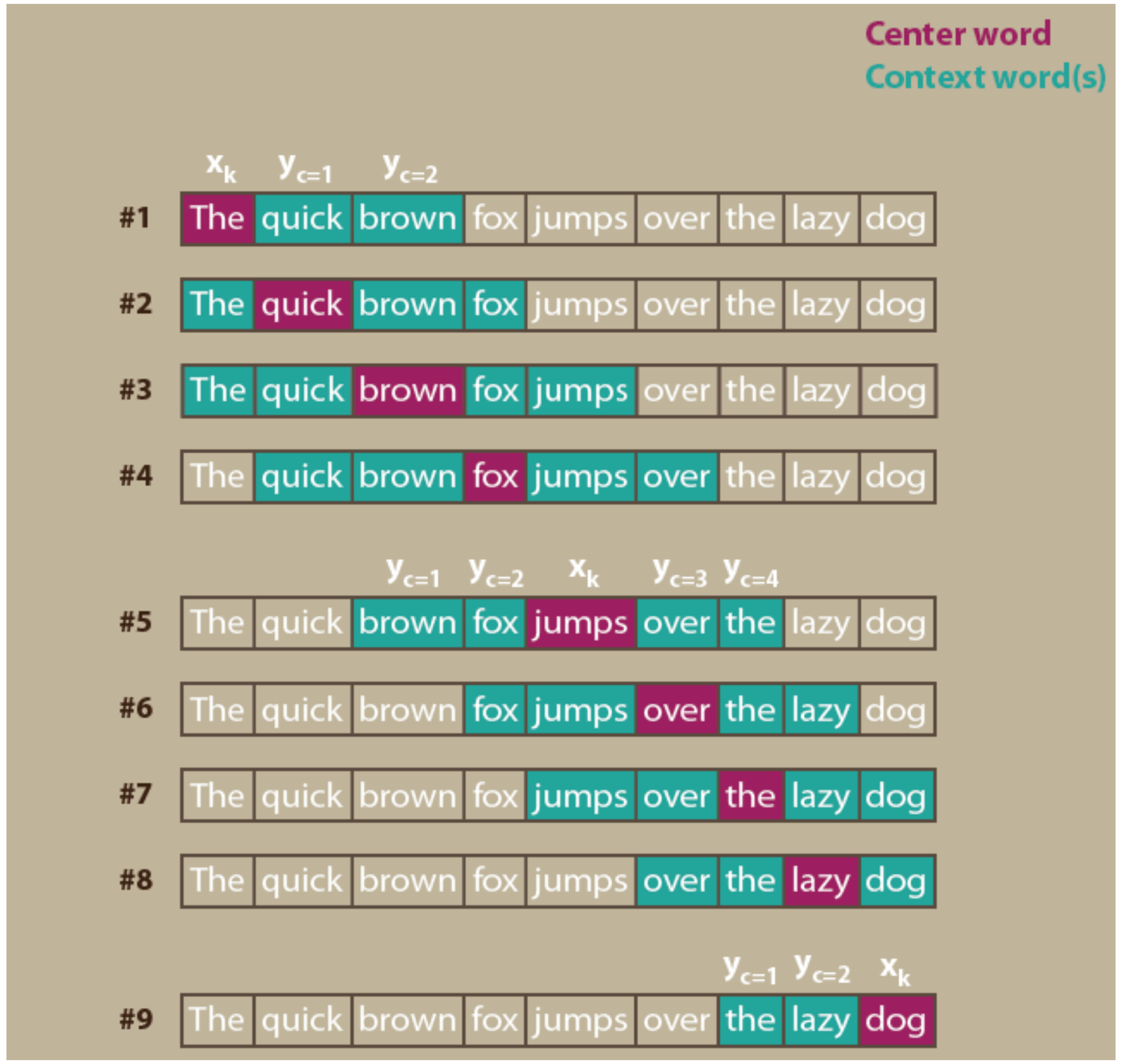
$$\mathrm{H}(q, p) = -\sum_x q(x) \log p(x)$$

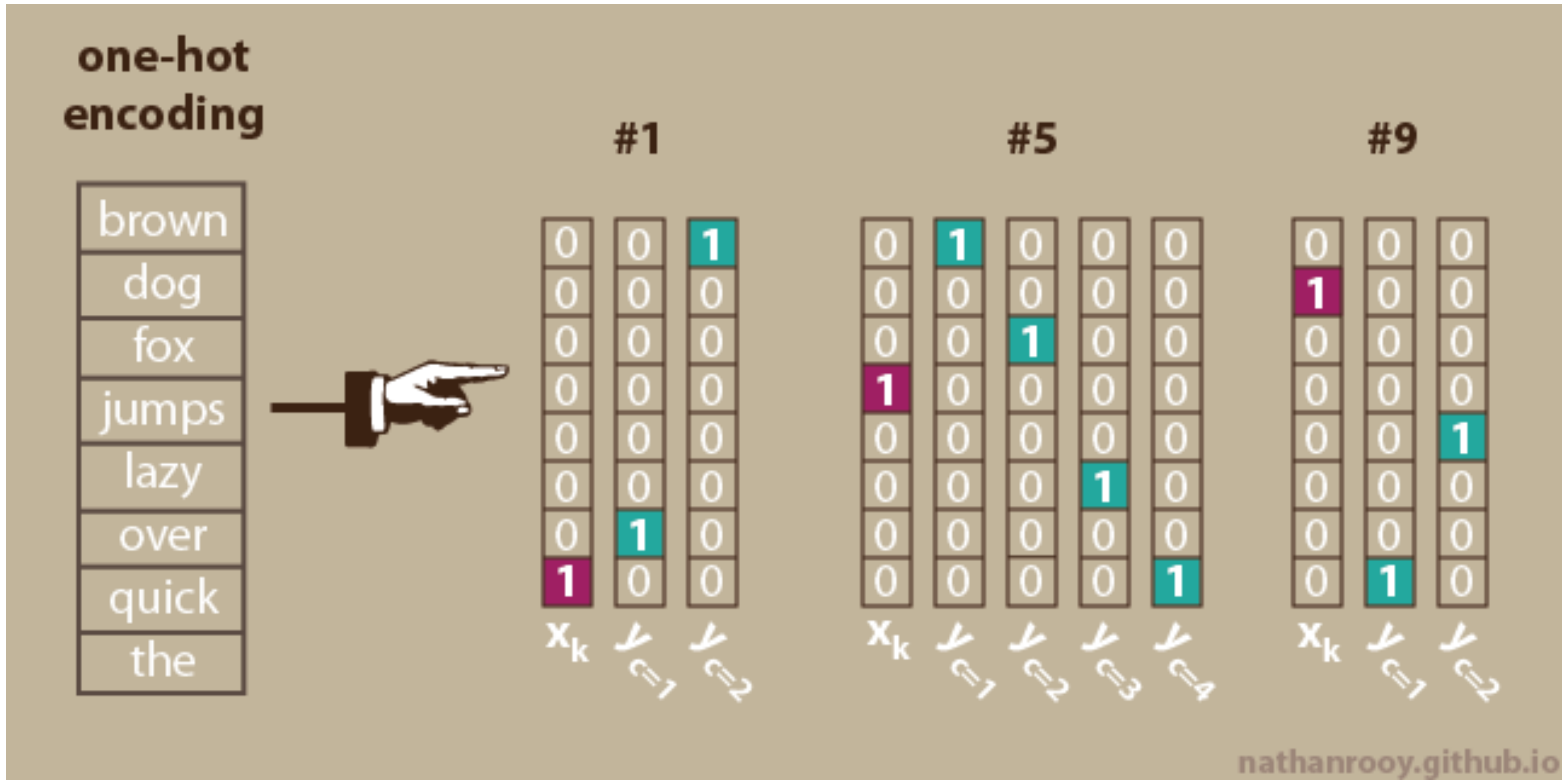
Word2vec



https://youtu.be/eXQAOTsbhLQ?t=554

Процесс обучения





Процесс обучения

V - размер словаря
N - размер эмбеддинга (200)

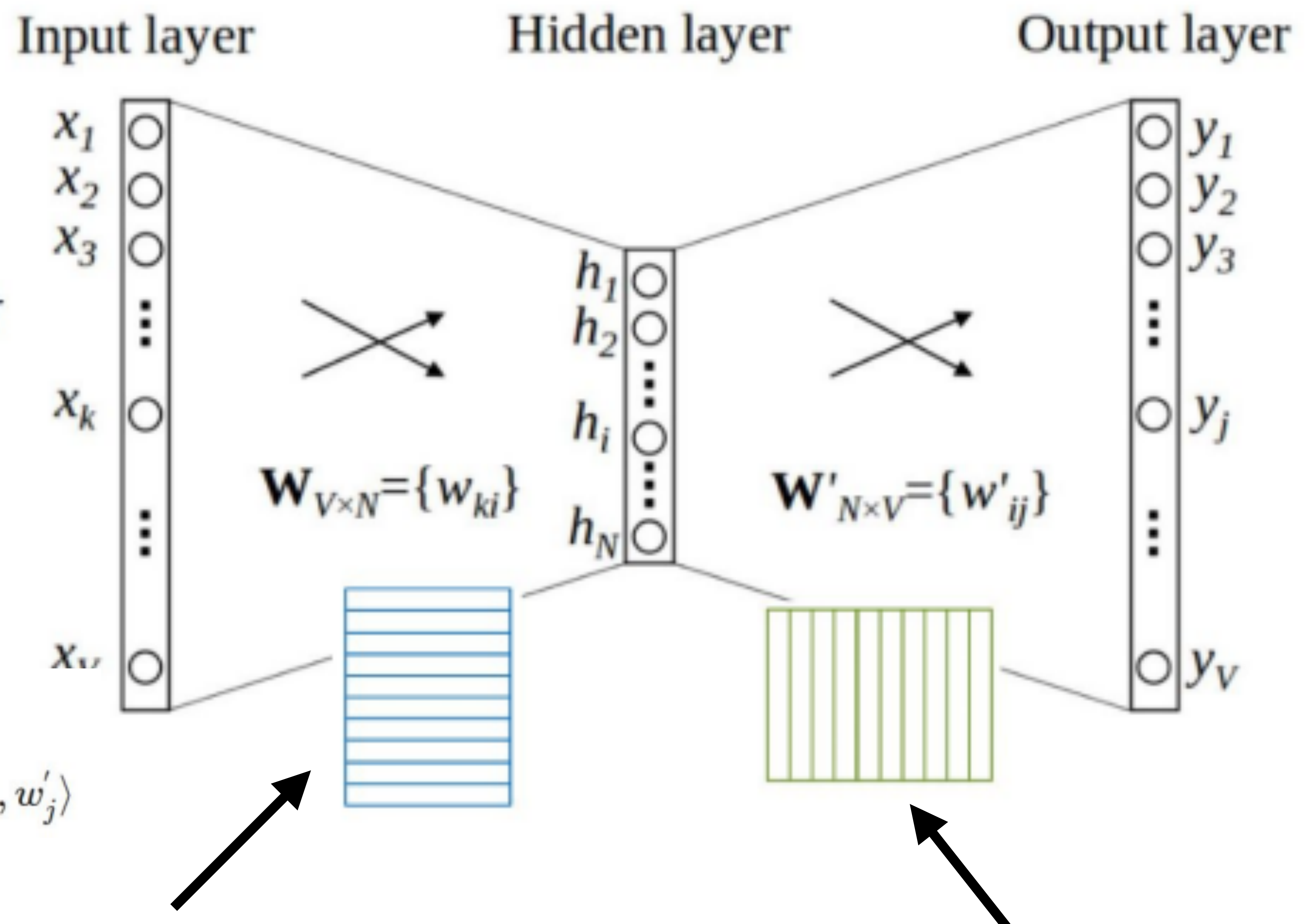
$$\mathbf{W} - V \times N$$
$$\mathbf{W}' - N \times V$$

$$\mathbf{W}^T \cdot \mathbf{x} = h \Longrightarrow (N \times V) \cdot (V \times 1) = N \times 1$$

$$\mathbf{W}^T = [w_1^T w_2^T \cdots w_V^T] \Rightarrow \sum_{i=1}^{V} w_i^T \mathbf{x}_i = w_k^T = h$$

$$\mathbf{W}'^T \cdot h = y \Longrightarrow (V \times N) \cdot (N \times 1) = V \times 1$$

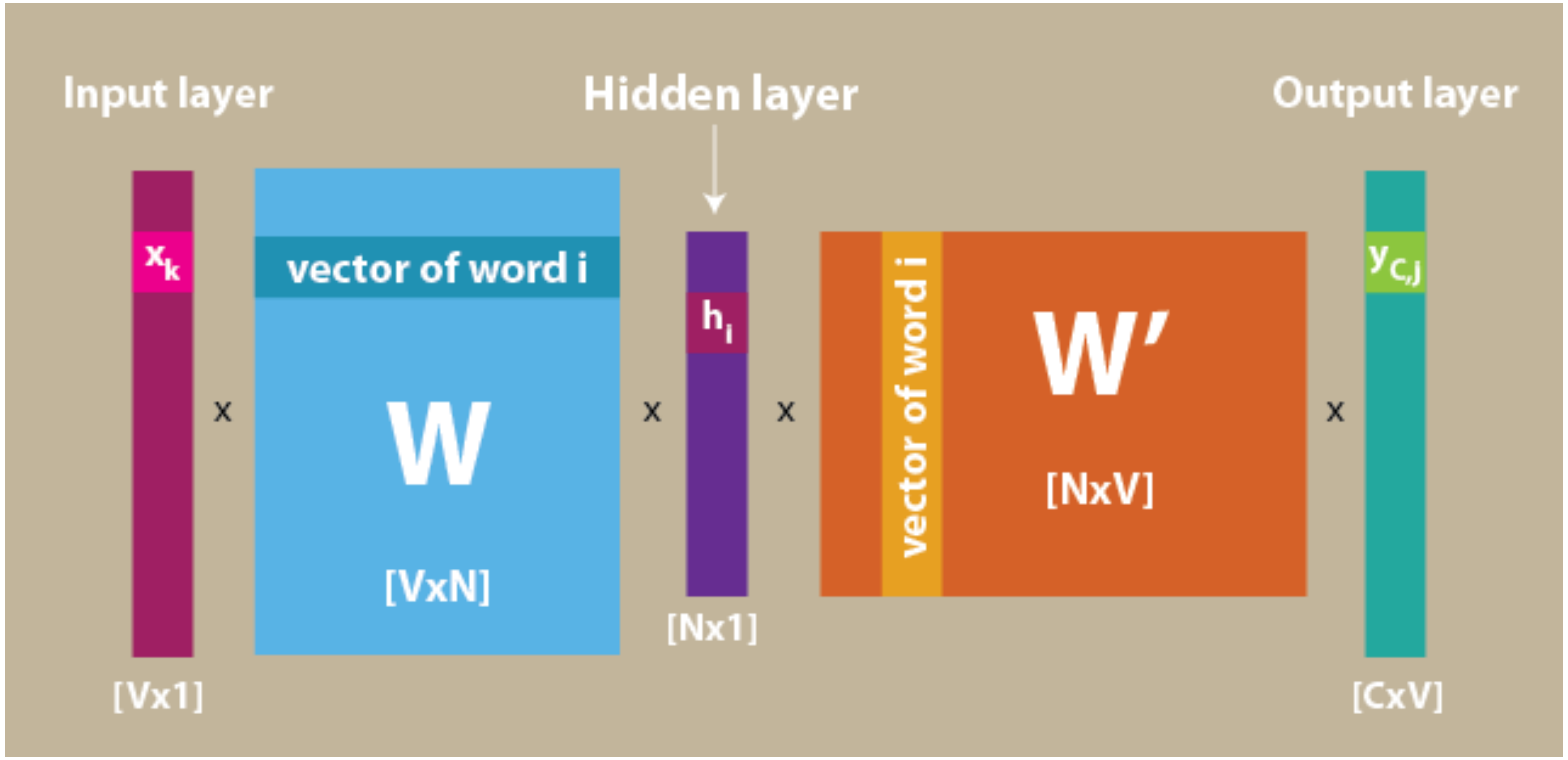
$$\mathbf{W}' = [w_1' w_2' \cdots w_V'] \Rightarrow y_j = (\mathbf{W}'^T \cdot h)_j = (w_j')^T w_k^T = \langle w_k, w_j' \rangle$$



Матрица с представлением слова как центрального

Матрица с представлением слова как контекстного

Процесс обучения



+ softmax = вероятности

Я люблю пить кофе по утрам одна V = 100

K = 23, позиция слова кофе в словарь, j = 45 = пить, [0, 0, 0, 0, 0, 0, ... 1, ...]

1) вытаскиваем 23 эмбеддинг, умножаем на матрицу контекстных эмбеддингов и получаем распределение на все контекст слова

Questions

- Какая модель работает быстрее?
- Какую проблему видите?

Subsampling

Problem? Some words are not meaningful.

Each word w in the training set is discarded with the probability computed by the formula

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$$

t- chosen threshold, words with a frequency greater than t are discarded
f(w_i) - frequency of w_i

She drinks neither a cup of coffee nor a cup of tea for breakfast.

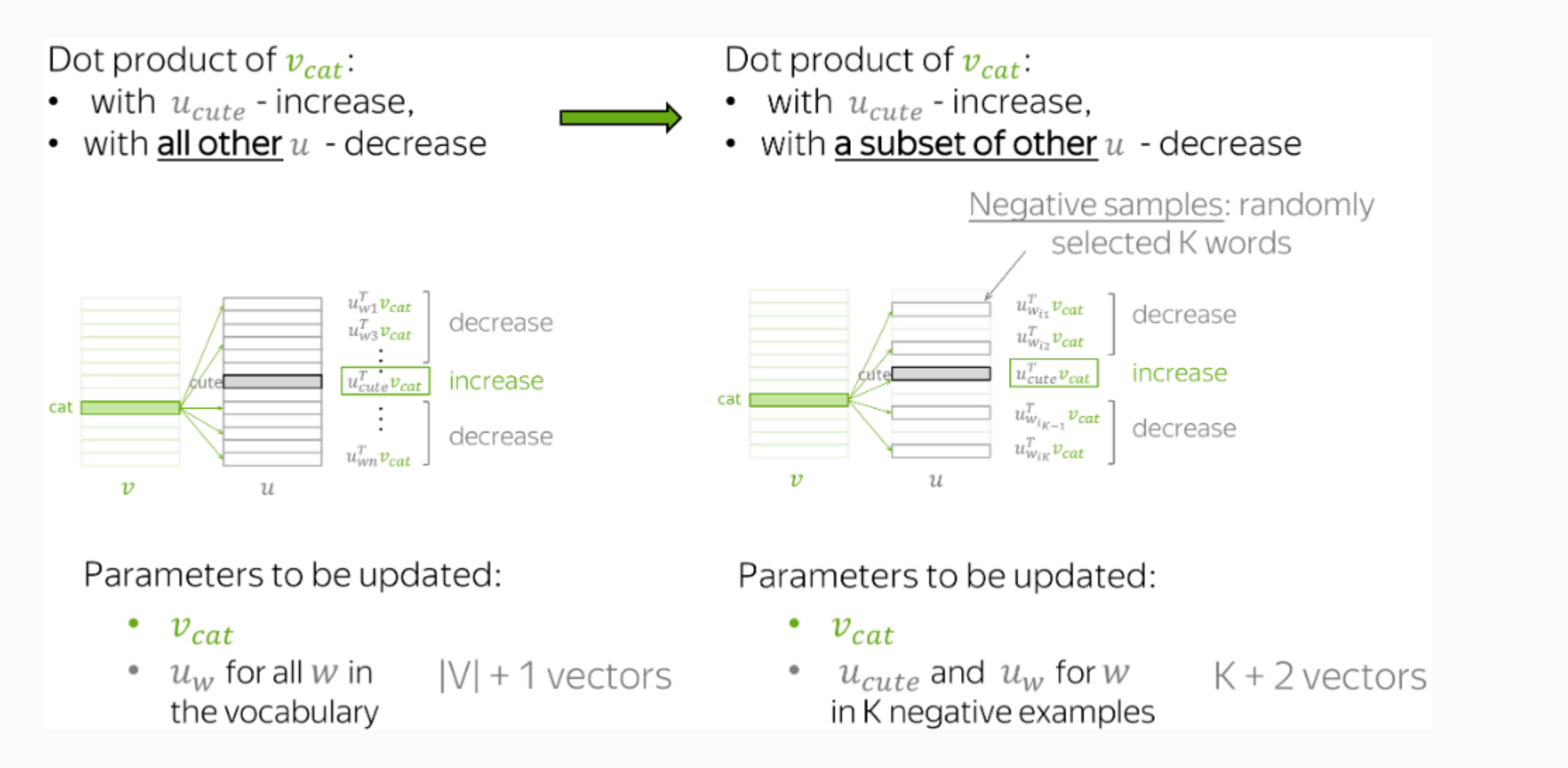
(a cup of + coffee + nor a cup)

(drinks neither cup + coffee + cup tea breakfast)

Negative sampling

1. Пить и кофе - это близкие слова = контекстные 1
2. Кофе и машина - это разные далекие = не контекстные 0

Problem? Computationally intense to train over the whole vocab.



Recipe

Somewhat standard setting is:

- Model: Skip-Gram with negative sampling;
- Number of negative examples: for smaller datasets, 15-20; for huge datasets (which are usually used) it can be 2-5.
- Embedding dimensionality: frequently used value is 300, but other variants (e.g., 100 or 50) are also possible.

Я люблю пить **кофе**
Кофе - это лучший напиток
Моя мама готовит **кофе** по утрам
Есть **кофе** - есть бодрость

{кофе: [1, 2,4 23,424 ,],
пить: [1,3452423, 4236 ,}

Types

Glove

Problem? Need global info.

Global information from corpus to **learn vectors**

Уменьшаем лосс слов, которые редко встречаются

Before training count occurrences of pairs [word$_i$, word$_j$] in corpus

Compute probabilities: $P_{ij} = \frac{Count(v_i, v_j)}{Count(v_i)}, Count(v_i) = \sum_k Count(v_i, v_k)$

Objective function:

Я люблю пить **кофе**
Кофе - это лучший напиток
Моя мама готовит **кофе** по утрам
Есть **кофе** - есть бодрость

На самолете кофе никогда не наливают

$$J(\theta) = \frac{1}{2} \sum_{i,j=1}^{W} \boxed{f(P_{ij})} (u_i^T v_j - \log P_{ij})^2$$

Discount factor for rare words

FastText

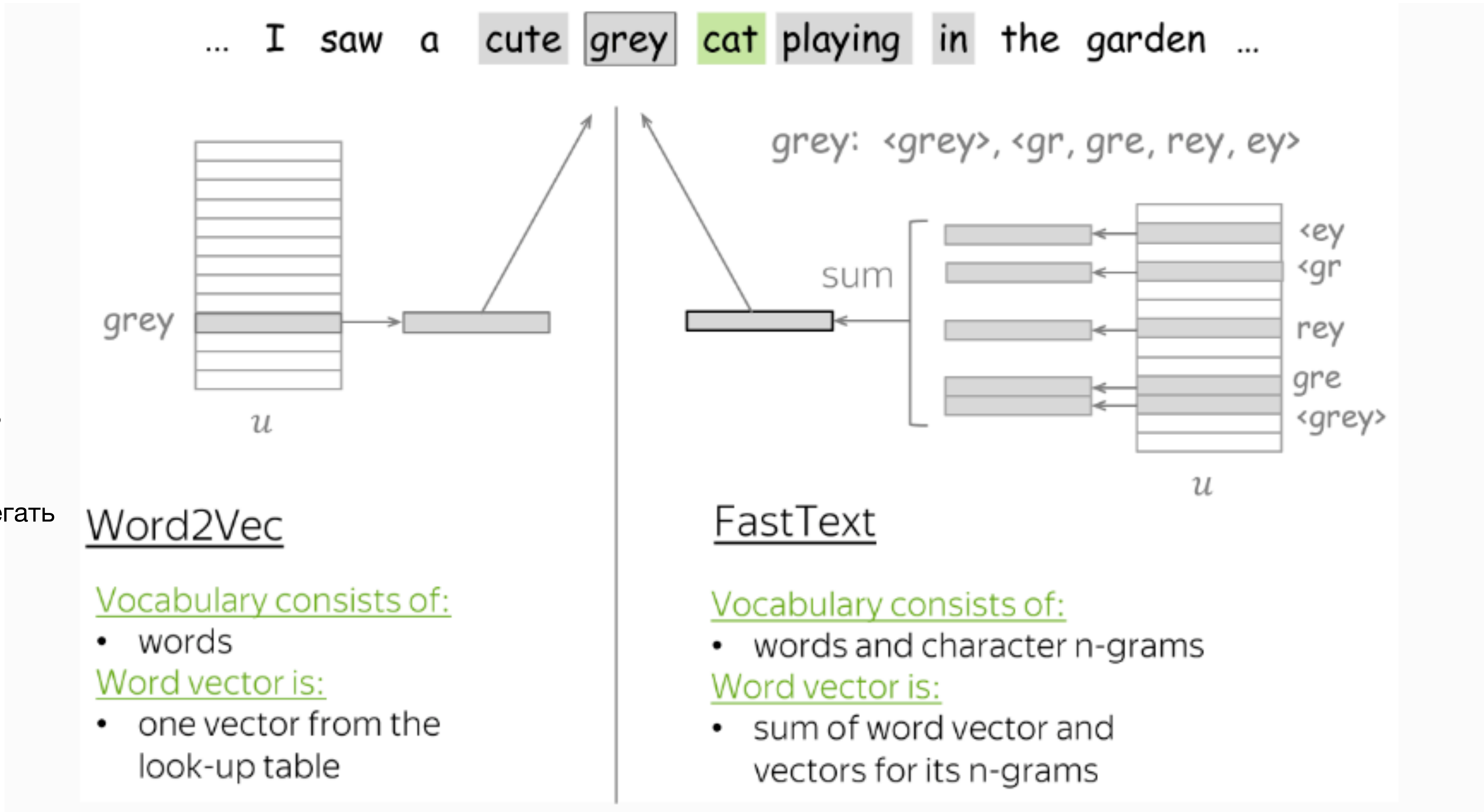
Problem? Need morph info

Убежал 2
Прибежал 1
Сбегал 3
Бегать 5
Перебежал 10

Убежал = у + бегать
Прибежал = При + бегать
Сбегал = с + бегать

21 обновим эмбеддинг для бегать

Out of vocabulary



FastText

Какие есть в этом плюсы?

- better understanding of morphology
  By assigning a distinct vector to each word, we ignore morphology.
  Giving information about subwords can let the model know that different tokens can be forms of the same word.
- representations for unknown words
  Usually, we can represent only those words, which are present in the vocabulary.
  Giving information about subwords can help to represent out-of-vocabulary words relying of their spelling.
- handling misspellings
  Even if one character in a word is wrong, this is another token, and, therefore, a completely different
  embedding (or even unknown word). With information about subwords, misspelled word would still be similar to the original one.

FastText

Какие есть в этом плюсы?

ELMO

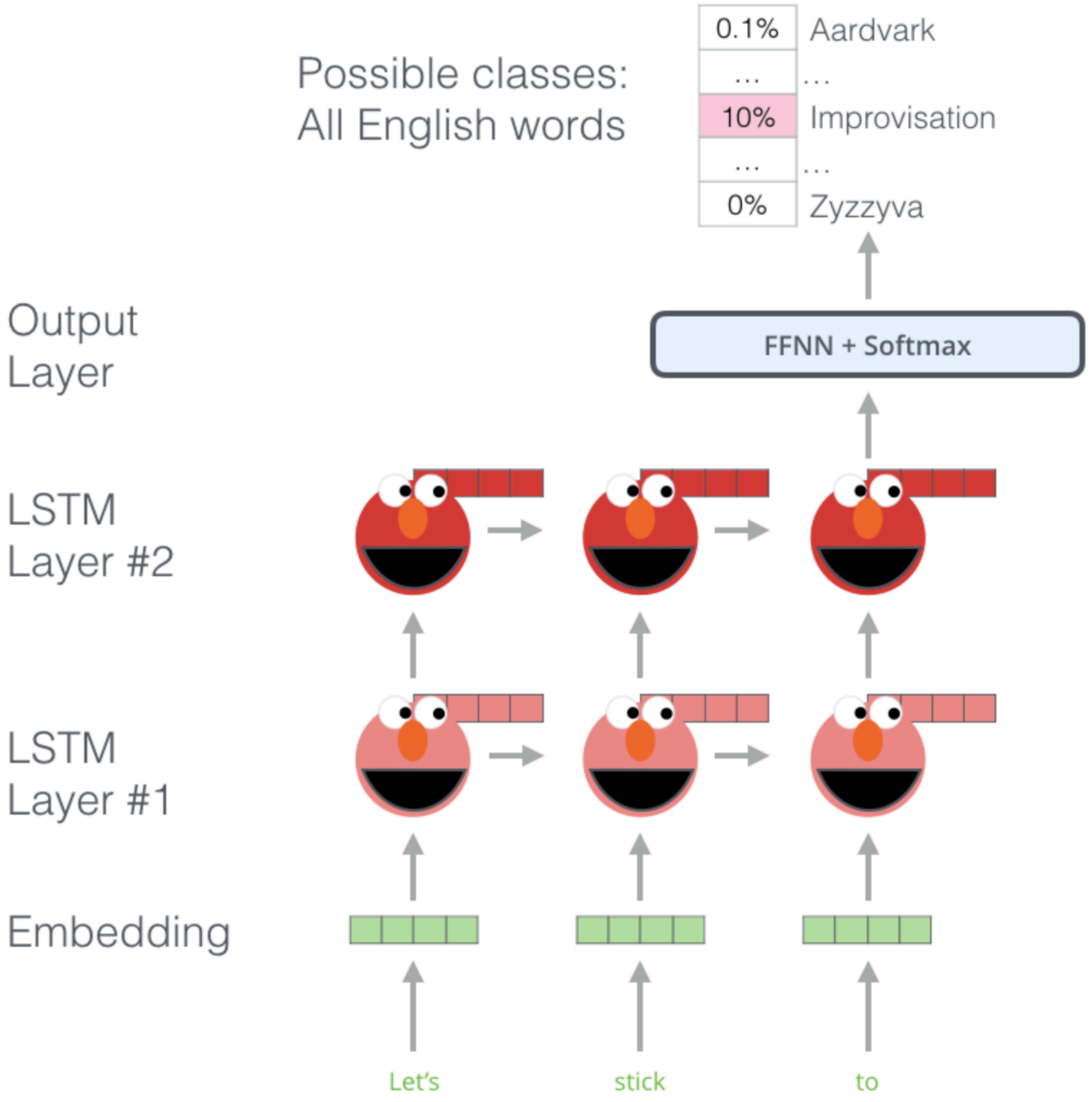
Problem? We need context

https://arxiv.org/pdf/1802.05365.pdf

ELMO

Embeddings from Language Models

Language Modeling task - predict the next word



ELMO

Problem? We need context

* Language Modeling task - predict the next word
* Bidirectional LSTM
* Token embeddings

$$ELMo_k^{task} = \gamma_k \cdot (s_0^{task} \cdot x_k + s_1^{task} \cdot h_{1,k} + s_2^{task} \cdot h_{2,k})$$

ELMo comes up with the contextualized embedding through grouping together the hidden states (and initial embedding) in a certain way (concatenation followed by weighted summation).