Embeddings

Rogozina A.

MIPT

21.03.19

Plan

Order-embeddings

Постановка задачи

Дано

Частично упорядоченное множество $(X, \preceq_X)$, $X \in \mathcal{D}$

Задача

Восстановить частичный порядок для всего $\mathcal{D}$

Предложение

Спроектировать X в уже частично упорядоченное пространство - "embedding space"$(Y, \preceq_Y)$

"Order embeddings og images and language"by Ivan Vendrov, Ryan Kiros, Sanja Fidler, Raquel Urtasun

Выбор embedding пространства

Требования к отображению

Функция $f : (X, \preceq_X) \to (Y, \preceq_Y)$ определяет order-embedding, если $\forall (u, v) \in X : u \preceq_X v \Leftrightarrow f(u) \preceq_Y f(v)$

Предлагаемое пространство Y

В качестве $(Y, \preceq_Y)$ предлагается взять $\mathbb{R}_+^{\mathcal{N}}$ с таким отношением частичного порядка: $x \preceq y \Leftrightarrow \bigwedge_{i=1}^{n} x_i \geq y_i$

Обучение

Штрафы

Предлагается искать аппроксимацию order-embedding f, вводя штрафы для каждой пары $(x, y) \in \mathbb{R}_+^{\mathcal{N}} : E(x, y) = \| \max(0, x - y)\|^2$

Функция ошибки подбирается для каждой конкретной прикладной задачи

Hypernum prediction

Def

Hypernum pair - пара (пример, обобщение), например (woman, person), (New York, city) В работе выявляются не только direct hypernums (person, organism), но и transitive hypernymy relation (cat, organism)

Loss function

$$\sum_{(u,v)\in WordNet} E(f(u), f(v)) + \max\{0, \alpha - E(f(u'), f(v'))\}$$

Results

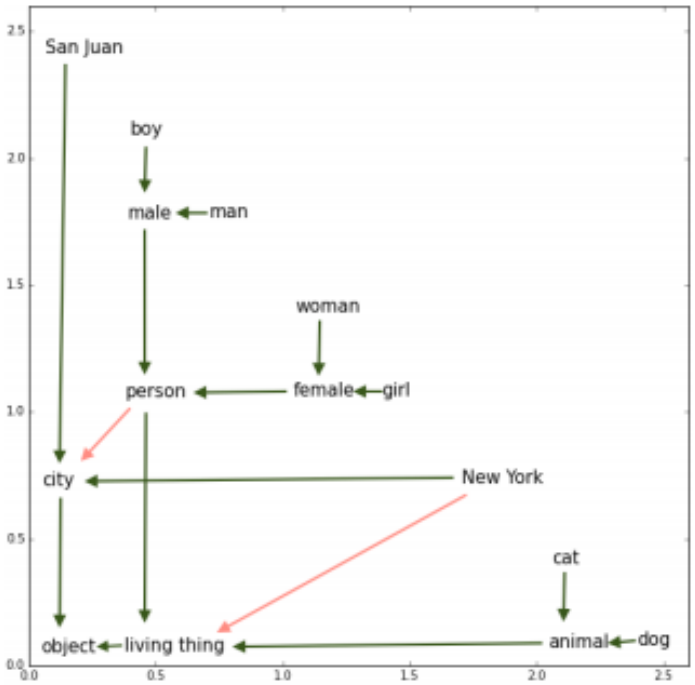


Figure 2: 2-dim order-embedding of a small subset of the WordNet hypernym relation. All the true hypernym pairs (green arrows) are correctly embedded, but two spurious pairs (pink arrows), are introduced. Only direct hypernyms are shown.

Caption-image retrieval

Хотим

Оценивать меру $S(c, i)$ соответствия описания(caption) картинке(image)

Предложение

Проектируем и картинки, и описания, в одно embedding-order пространство(функциями $f_i(i)$ и $f_c(c)$ соответственно). Тогда определим $S(c, i) = -E(f_i(i), f_c(c))$.

Loss function

$$\sum_{(c,i)} \left(\sum_{c'} \max\{\alpha - S(c, i) + S(c', i)\} + \left(\sum_{i'} \max\{\alpha - S(c, i) + S(c, i')\} \right) \right.$$

Results

Captions	Image rank	
	cosine	order-emb
a sitting area with furniture and flowers makes a backdrop for a boy with headphones sitting in the foreground at one of the chairs at a dining table that holds glasses and a handbag working at a laptop	4	8
a kid is wearing headphone while on a laptop	286	24
view of top of a white building with tan speckled area an uncovered awning with a pigeon in fight below and a red umbrella behind balcony wall	3	5
a pigeon flying near white beams of a building	91	6

Figure 3: Images with captions of very different lengths, and the rank of the GT image when using each caption as a query.

Textual entailment / Natural language inference

Задача

Обобщение задачи hypernum prediction на предложения.
Пример: "woman walking her dog in a park"$\leq$ "woman walking her dog" или "dog in a park", но не "old woman" или "black dog".

Loss function

$$\sum_{(p,h)} E(f(p), f(h)) + \sum_{(p',h')} \max\{0, \alpha - E(f(p'), f(h'))\}$$

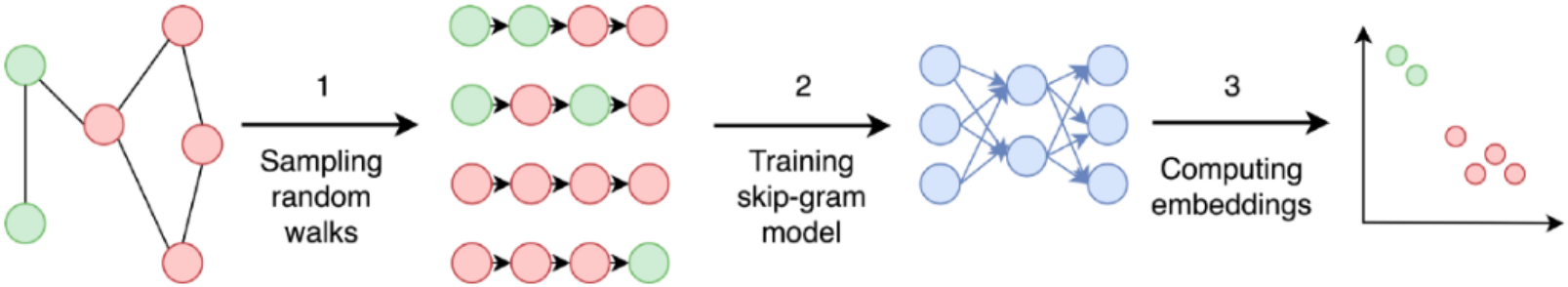
Graph embeddings

Основные подходы

- Vertex embedding
  - Deep walk
  - Node2Vec
  - Structural Deep Network Embedding (SDNE)
- Whole graph embedding
  - Graph2vec

DeepWalk
Vertex embedding

- Sampling: random walk 32 - 64 из каждой вершины на 40 вершин в глубину
- Training skip-gram: Получившаяся последовательность вершин = предложение в Skip-gram модели
- Computing embeddings



Phases of DeepWalk approach

Картинки: https://towardsdatascience.com/graph-embeddings-the-summary-cc6075aba007

Node2Vec
Vertex embedding

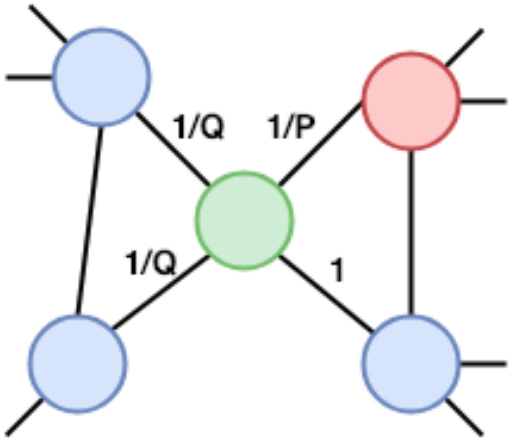
Недостаток DeepWalk

Так как "блуждания"по графу случайные, метод плохо сохраняет локальную окрестность вершины

Исправление

Добавляются параметры P и Q

- Q - вероятность что случайное блуждание перейдет из вершины в непросмотренную часть графа. Отвечает за исследование глобальной связности и сложных зависимостей.
- P - вероятность вернуться в просмотренную часть графа. Отвечает за исследование локальной окрестности вершин.



The figure shows probabilities of a random walk step in Node2vec. We just made a step from the red to the green node. The probability to go back to the red node is 1/P, while the probability to go the node not connected with the previous (red) node 1/Q. The probability to go to the red node's neighbor is 1.

Structural Deep Network Embedding (SDNE)

Vertex embedding

Свойства

Сохраняет близость первого и второго порядка

- Первый порядок - близость вершин, соединенных ребром
- Второй порядок - близость вершин, соединенных через общих соседей(компоненты связности)

Устройство

- Два "ванильных"автоэнкодера получают вектор "связности" вершины и пытаются восстановить его же.
- На вход эвтоэнкодерам подаются пары вершин,соединенные ребром. Расстояние между их эмбедингами включается в ошибку.

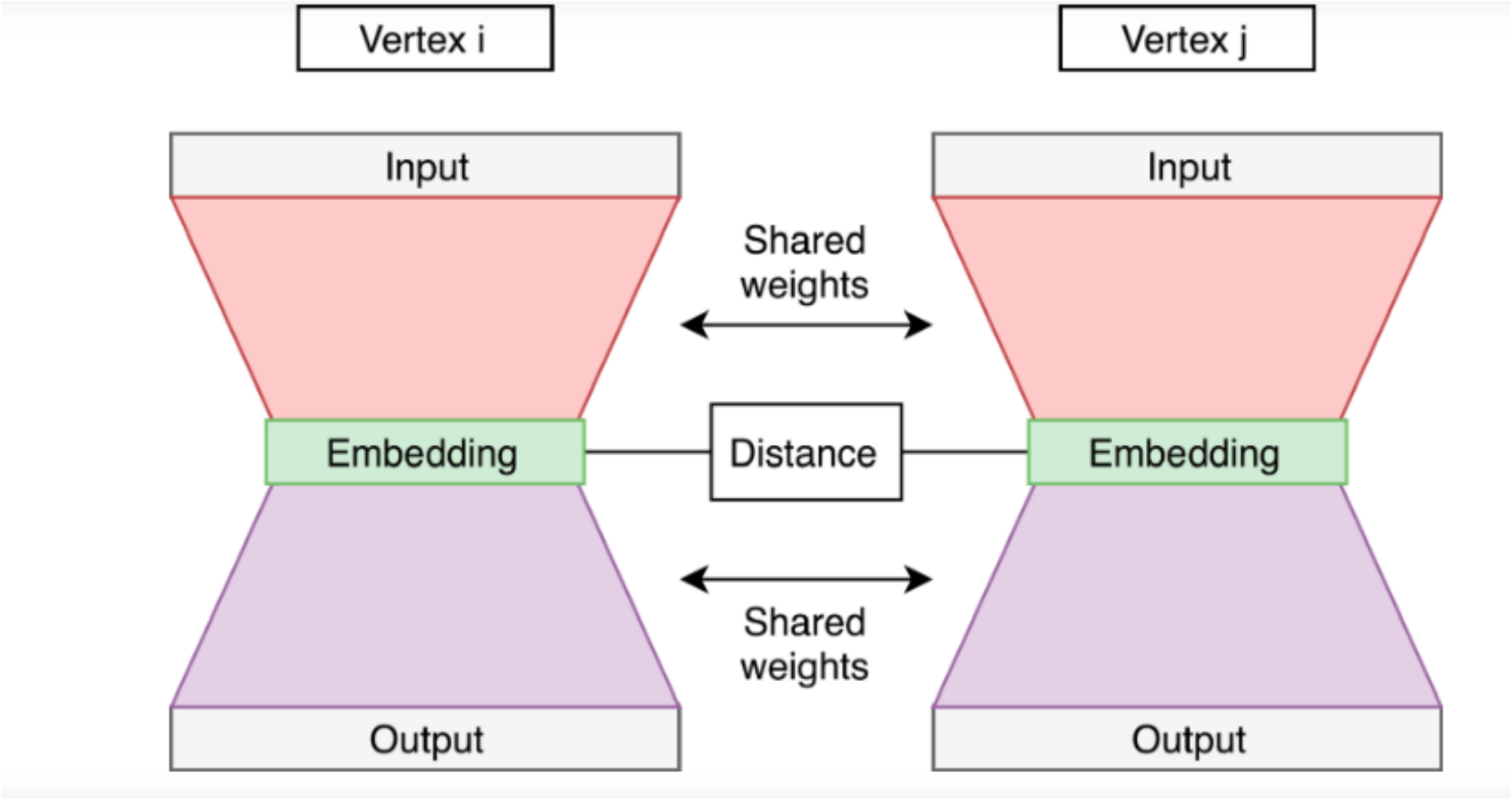


Рис.: Схема Structural Deep Network Embedding

Graph2Vec

Устройство

- Один граф - один выходной вектор
- Близкие по структуре графы оказываются близки в пространстве эмбеддингов.

- Выделяем множество подграфов в графе.
- Граф = множество подграфов, документ = множество слов, поэтому обучаем embedding как doc2vec: максимизируем вероятность предсказать "слово которое есть в "документе".
- На входе граф представляется как one-hot вектор.

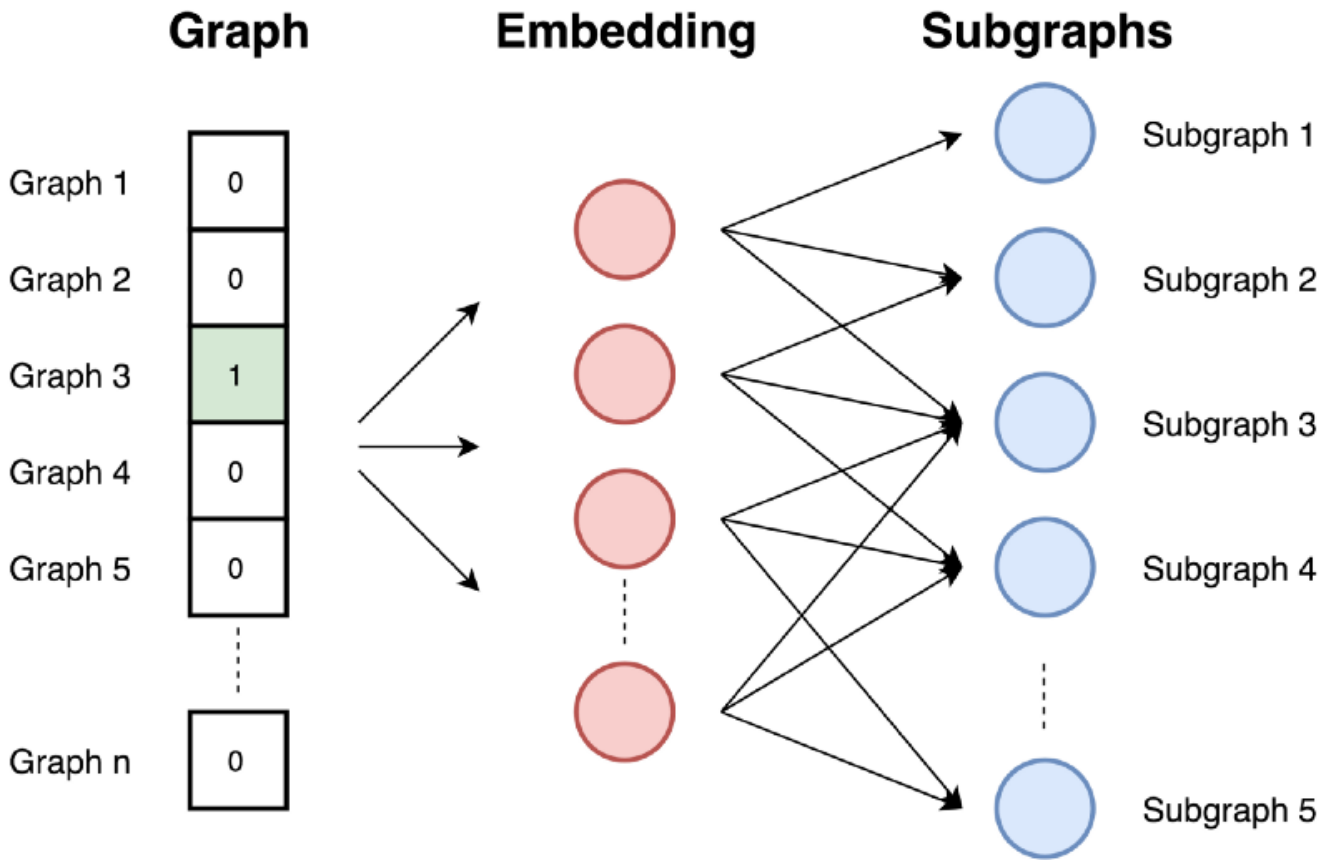
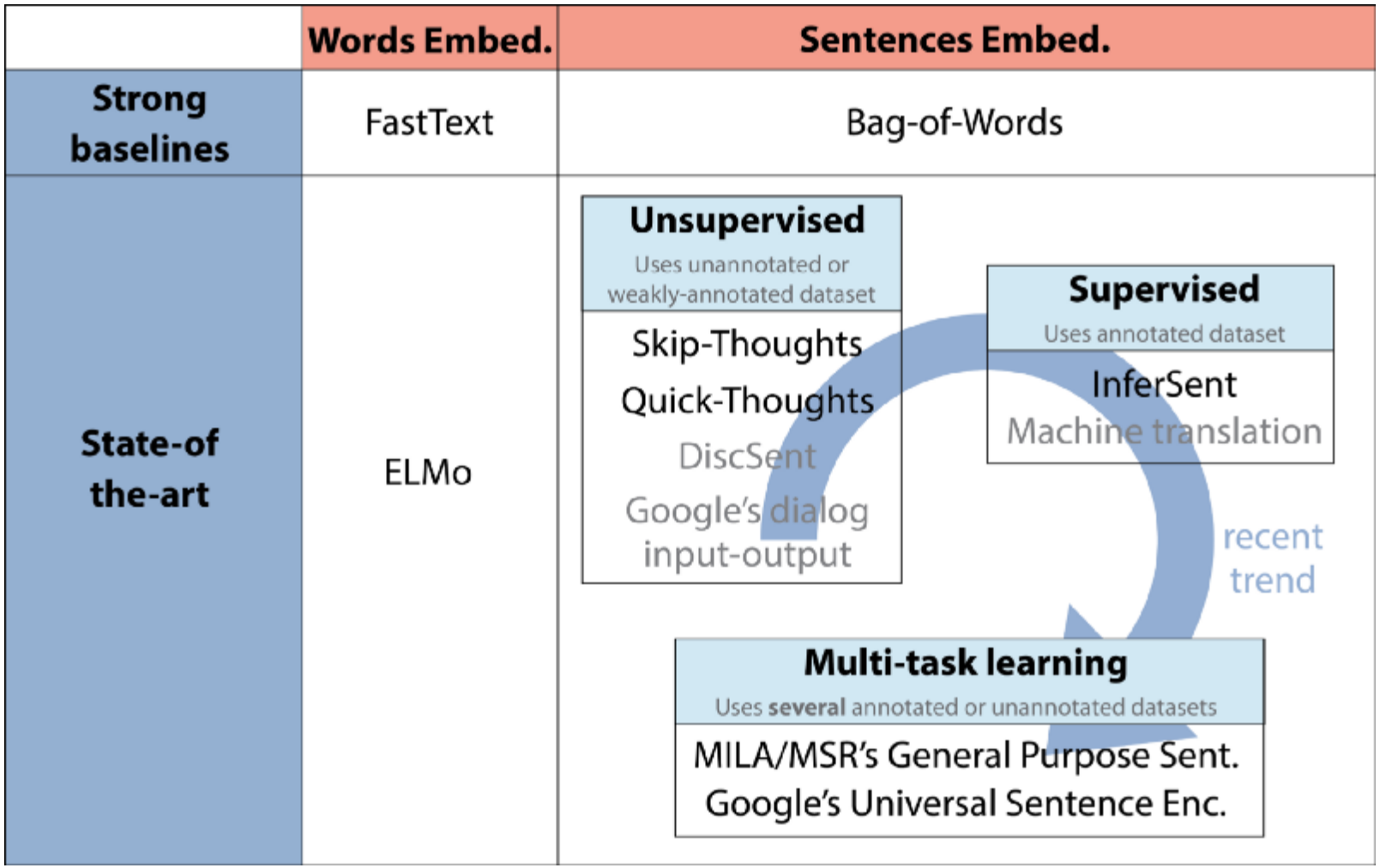


Рис.: Схема Graph2Vec

Word embeddings

Fast overview



Word embeddings

Most commonly used

- GloVe
- Word2Vec
- ELMo

Glove

Основная идея

Matrix factorization + local context window

- Строим матрицу $X \in \mathbb{R}^{V \times V}$, x_{ij} — количество раз, когда слово i встречается в контексте слова j (в окне ≤ 9 слов)
- $P_{ij} = \frac{x_{ij}}{\sum_j x_{ij}}$ - вероятность j в контексте i
- $F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$ - какое слово (i или j) вероятнее увидеть в контексте k

GloVe

- оценим $\tilde{F}((w_i - w_j)^{\mathrm{T}}\tilde{w}_k) = \frac{F(w_i^{\mathrm{T}}\tilde{w}_k)}{F(w_j^{\mathrm{T}}\tilde{w}_k)}$, $F(w_i^{\mathrm{T}}\tilde{w}_k) = P_{ik}$
- Возьмем $F(x) = \exp(x)$, а w_i такими, что $w_i^{\mathrm{T}}\tilde{w}_k = \log(P_{ik}) = \log(x_{ik}) - \log\left(\sum_j x_{ij}\right)$
- Оптимизируем функционал ошибки получившегося SVD - разложения: $J = \sum_{i,j=1}^{V} f(x_{ij})(w_i^{\mathrm{T}}\tilde{w}_j + b_i + b_j - \log(x_{ij}))^2$, где $b_i + \tilde{b_j} = \log\left(\sum_j x_{ij}\right)$

Word2Vec: Skip-Gram

- Максимизируем функцию
  $L = p(w_{i-h}, \dots, w_{i+h} \mid w_i) = \prod_{-h \le k \le h, k \neq 0} p(w_{i+k} \mid w_i)$
- $p(u \mid v) = Softmax(w_u, \tilde{w}_v)$
- Continuous Bag-of-Words - все аналогично, но предсказываем не контекст по слову, а слово по контексту

ELMo

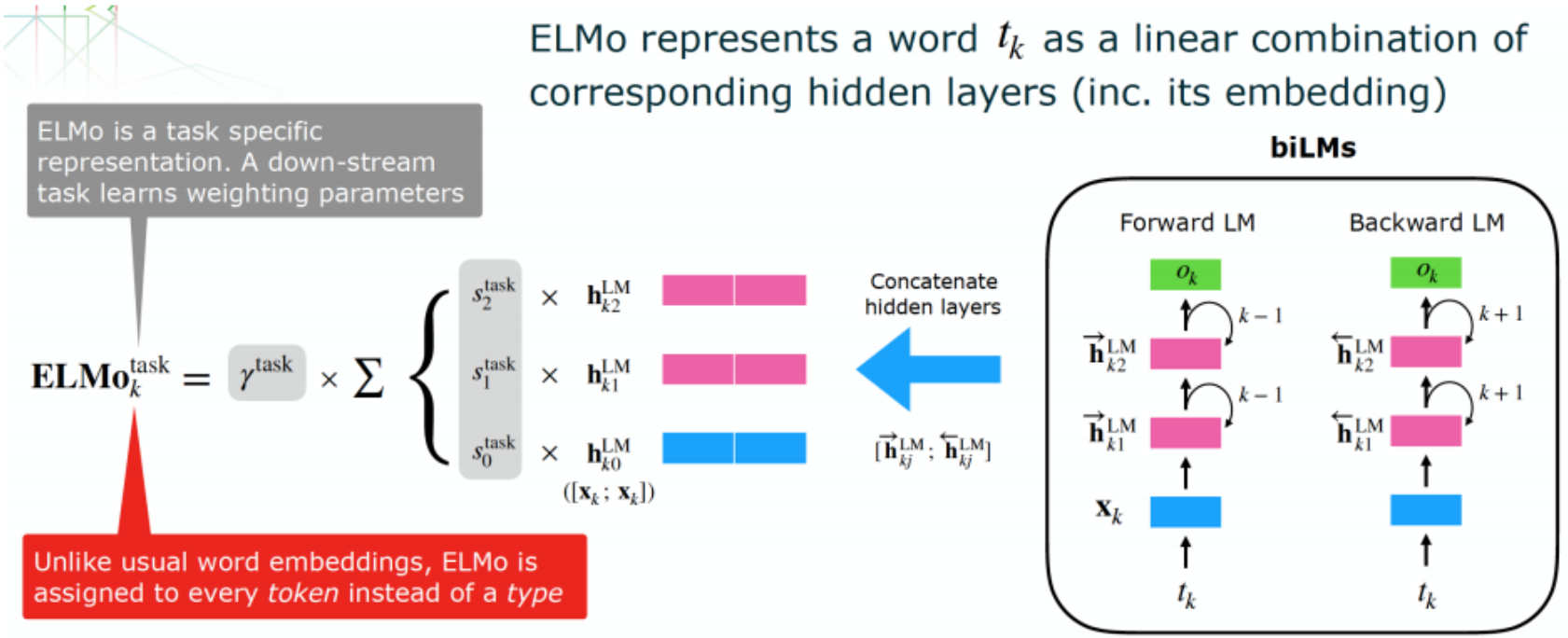


Рис.: http://www.machinelearning.ru/wiki/images/d/d7/Mmta18-sentence-encoders.pdf

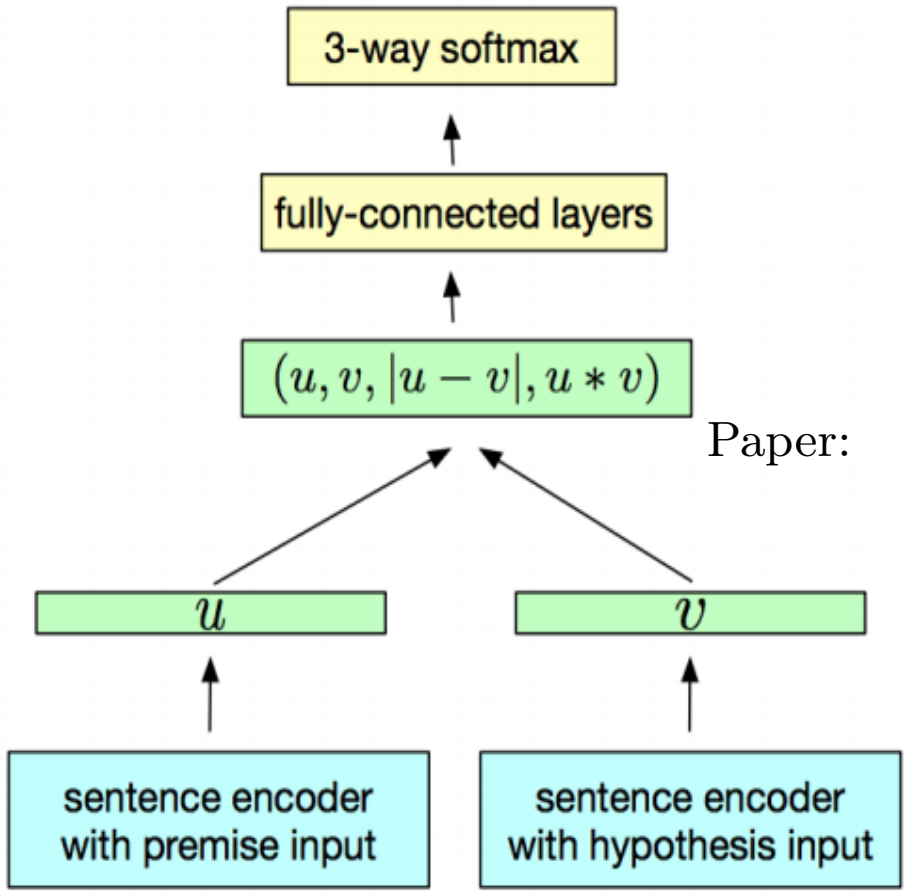
Sentence embeddings

Most commonly used

- Supervized
  - InferSent
- Unsupervized
  - Skip-thoughts vectors
  - Quick-thoughts vectors
- Multi-task learning
  - Google's Universal Sentence Encoder

Supervised: InferSent

Проводится трехклассовая классификация. Пары предложений размечены на 3 категории: neutral, contradiction и entailment



Paper:

"Supervised Learning of Universal Sentence Representations from Natural Language Inference Data" by A. Conneau et al.

Skip-thoughts
Unsupervized

- RNN - based encoder - decoder предсказывает "окружающие"данное предложения
- Vocabulary expansion scheme - линейное преобразование: выученные эмбеддинги $\longrightarrow$ word2vec(например).

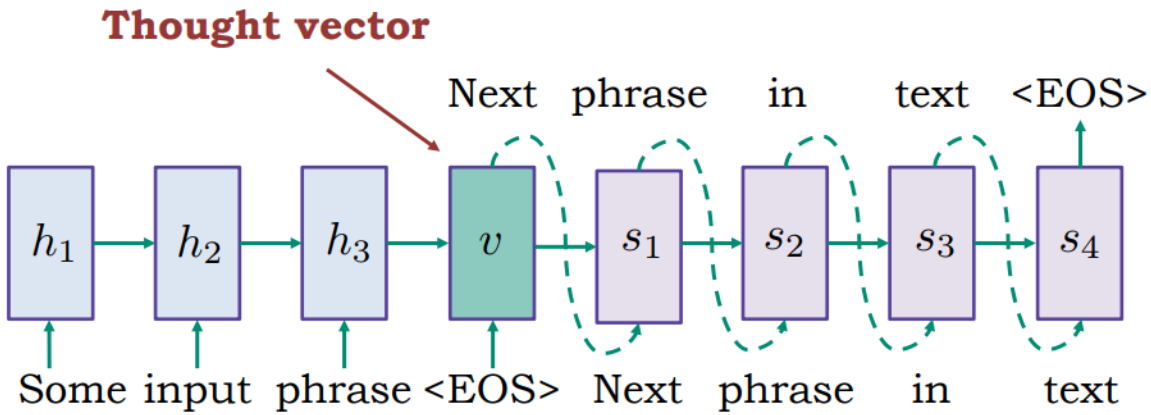
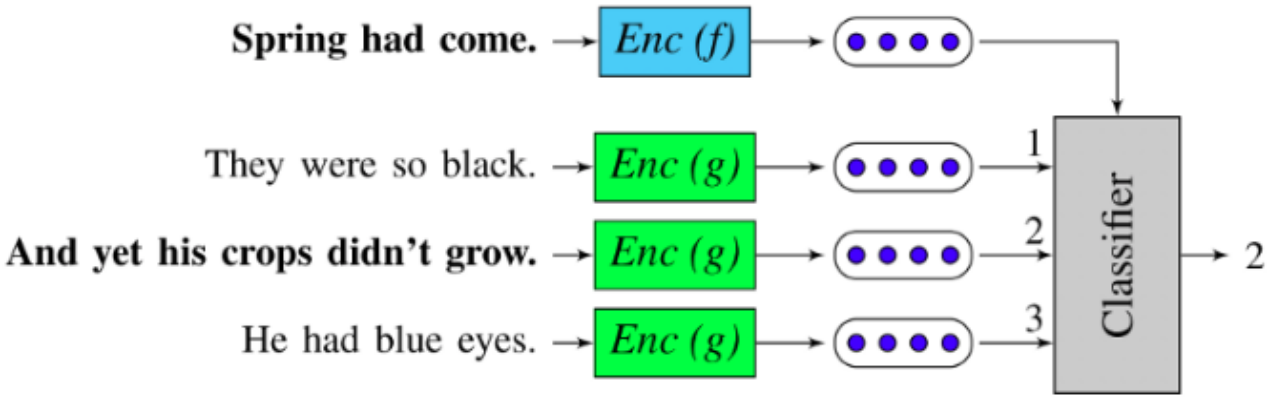


Рис.: Kiros et. al. Skip-Thought Vectors, 2015

Quick-thoughts
Unsupervized

- Предсказывать следующее предложение по предыдущему $\longrightarrow$ классификация
- Гораздо быстрее.



Quick-thoughts classification task. The classifier has to chose the following sentence from a set of sentence embeddings. Source: "An efficient framework for learning sentence representations" by Logeswaran et al.