Linguistics in ML

Embeddings and Linguistic Structure

Александра Муравьёва

Национальный исследовательский университет
«Высшая школа экономики»

18 января 2019 г.

Содержание

Построение

1. Предобработка (токенизация, лемматизация, etc.)
2. Построение частотной матрицы
   - элементы — счётчики n_{uv}, которые показывают, сколько раз слово u встретилось в определённом контексте v
   - контекст — соседние слова в окне размера n
3. Частотное взвешивание
4. Понижение размерности

Word2vec

- CBOW: пытаемся предсказать слово по контексту и минимизируем $E = -\log(p(\vec{w}_t|\vec{W}_t))$
- Skipgram: предсказываем контекст по слову

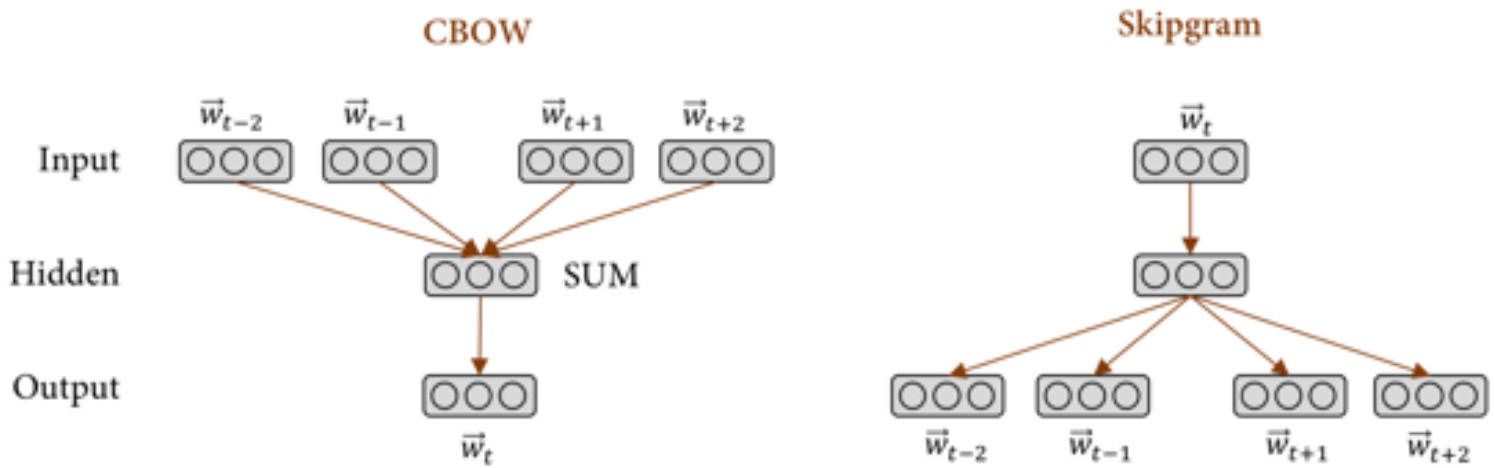


Figure 2: Learning architecture of the CBOW and Skipgram models of Word2vec (Mikolov et al., 2013a).

Лингвистический контекст

- *WE* основаны на дистрибутивной семантике — используют контекст
- Традиционно контекст — окно размера n слов
- Однако можно использовать **лингвистический контекст**
  - глаголы в конструкциях типа субъект-глагол и глагол-объект
  - одно существительное влево и одно существительное вправо для главного существительного в предложении
  - все прилагательные, зависящие от данного существительного

Проблемы дистрибутивного подхода

- Главное качество получившихся векторных представлений — близость векторов (например, по косинусной мере) для семантически близких слов
- Однако векторное представление единственно для разных значений слова: $WE(\text{ключ}_{\text{родник}}) = WE(\text{ключ}_{\text{дверной}})$
- Не зависит от принадлежности к семантическому полю: $d(WE(\text{свет}), WE(\text{сиять})) = \infty$
- Или даже от принадлежности к одному словообразовательному гнезду: $d(WE(\text{облако}), WE(\text{облачный})) = \infty$
- Не учитывает perceptual features типа "is curved"

Учёт полисемии

Несколько подходов

1. Sense per collocation
2. Using multilingual context
3. Multimodal distributions

Учёт полисемии

Несколько подходов

1. Sense per collocation
2. Using multilingual context
3. Multimodal distributions

Sense per collocation

- Значение моделируется как скрытая переменная в байесианских моделях
- Опора на эвристику *sense per collocation*, т.е. что распределение контекстов зависит от значения

Учёт полисемии

Несколько подходов

1. Sense per collocation
2. Using multilingual context
3. Multimodal distributions

Sense per collocation

- Значение моделируется как скрытая переменная в байесианских моделях
- Опора на эвристику *sense per collocation*, т.е. что распределение контекстов зависит от значения
- Для улучшения качества требуется много training data
- Увеличение размерности *WE* или использование более сложного алгоритма (LSTM) зачастую приводит к таким же результатам

Учёт полисемии

Using multilingual context

- Разные значения могут переводиться на другие языки по-разному: *bank* (англ.) $\rightarrow$ *banc* или *banque* (фр.)
- Используем параллельные корпуса и функции слово-перевод (word alignments) для обучения T векторов для T значений слов
- значение T своё для каждого слова, новое значение добавляется, если его вероятность превышает порог
- Мультилингвальный, а не билингвальный метод: нужно, чтобы вектора для разных переводов лежали в одном пространстве и близко друг к другу
- Для этого конкатенируем корпуса $EN - l_1, \ldots, EN - l_n$ и обучаем совместно (joint learning)

Учёт полисемии

Multimodal distributions

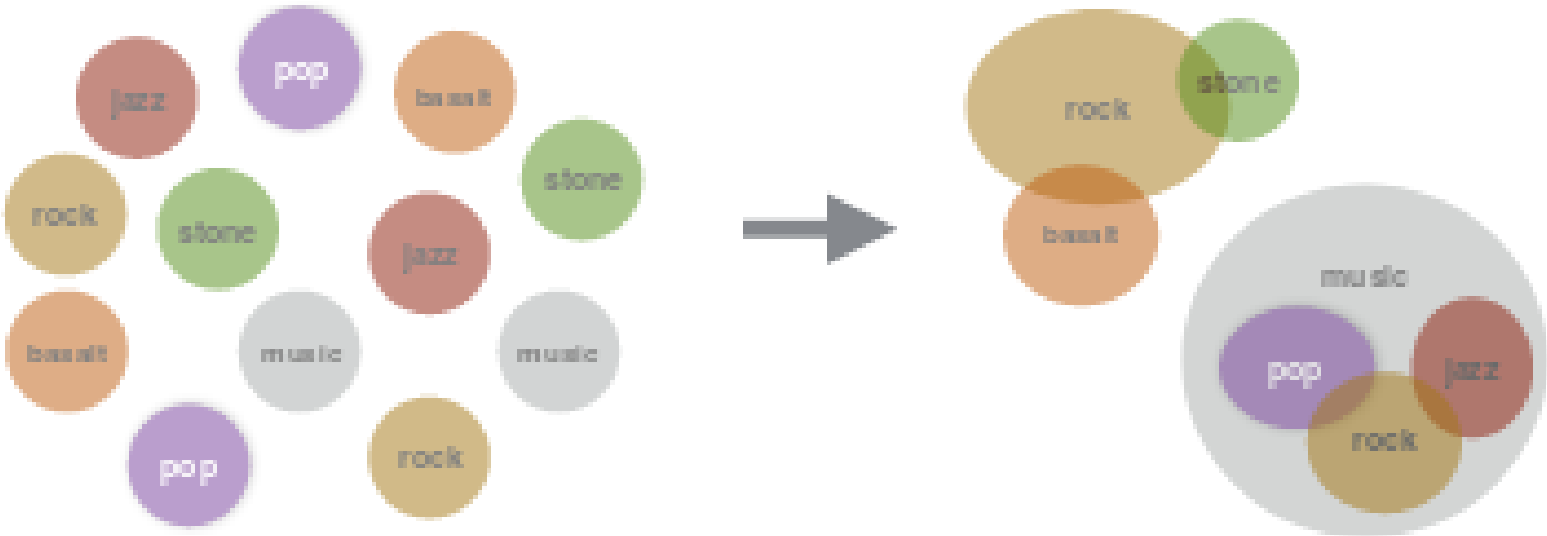
- Одно слово моделируется в виде смеси гауссиан

$$f_w(\vec{x}) = \sum_{i=1}^{K} p_{w,i}\, \mathcal{N}\left[\vec{x}; \vec{\mu}_{w,i}, \Sigma_{w,i}\right]$$

  где $\mu_{w,i}$ – расположение i-того компонента слова w, соответствует обычным word2vec-эмбеддингам; $p_{w,i}$ – вероятность компонента (вес в смеси); $\Sigma_{w,i}$ – матрица ковариаций компонента

- далее мы выучиваем параметры $\theta = \{\mu_{w,i}, p_{w,i}, \Sigma_{w,i}\}$, максимизируя правдоподобность встретить слово при данном соседнем

Учёт полисемии



[*Athiwaratkun, Wilson. ACL 2017*]

Центр эллипсоида определяется $\mu_{w,i}$, а контур — $\Sigma_{w,i}$ и отражает тонкости значения и неопределённость

Учёт полисемии

Внизу гауссиана, вверху смесь гауссиан.

Учёт полисемии

Dataset	sg*	w2g*	w2g/mc	w2g/el	w2g/me	w2gm/mc	w2gm/el	w2gm/me
SL	29.39	**32.23**	29.35	25.44	25.43	29.31	26.02	27.59
WS	59.89	65.49	71.53	61.51	64.04	**73.47**	62.85	66.39
WS-S	69.86	76.15	76.70	70.57	72.3	**76.73**	70.08	73.3
WS-R	53.03	58.96	68.34	54.4	55.43	**71.75**	57.98	60.13
MEN	70.27	71.31	72.58	67.81	65.53	**73.55**	68.5	67.7
MC	63.96	70.41	76.48	72.70	**80.66**	79.08	76.75	80.33
RG	70.01	71	73.30	72.29	72.12	**74.51**	71.55	73.52
YP	39.34	41.5	41.96	38.38	36.41	**45.07**	39.18	38.58
MT-287	-	-	64.79	57.5	58.31	**66.60**	57.24	60.61
MT-771	-	-	**60.86**	55.89	54.12	60.82	57.26	56.43
RW	-	-	28.78	32.34	33.16	28.62	31.64	**35.27**

[*Athiwaratkun, Wilson. ACL 2017*]

Корреляция Спирмена для разных датасетов word similarity

- mc: maximum cosine similarity
- el: expected kernel likelihood
- me: minimum euclidean distance
- w2gm – мультимодальная модель

Учёт морфологии

Subword representations

1. Слово в виде множества признаков, включающих морфологическую информацию:
   <cat, noun, sg, 1,...>
2. Слово как сумма репрезентаций морфем:
   WE(по) + WE(ряд) + WE(ок)
3. Слово как сумма n-граммных векторов символов:
   WE(<wh) + WE(whe) + WE(her) + WE(ere) + WE(re>) + WE(<where>)
4. Морфологические трансформации:
   WE(create) + suffix:ϵ:s

Учёт морфологии

Morphological transformations. Soricut and Och

- Извлечь кандидатов правил суффиксального/префиксального словообразования
- Вне зависимости от этого обучить обычные эмбеддинги
- Выбрать из правил лучшие, основываясь на частоте их использования
- Для оценивания близости слов использовать эмбеддинги и правила совокупно: косинусная мера и ранг с пороговыми значениями >0.5 и <15 в исследовании
- Hit rate – число попаданий Ev выше $t_0 = 100$, где Ev

$$Ev^F((w_1, w_2), (w, w')) = F_E(w_2, w_1 + \uparrow d_w) \quad (1)$$
$$(w_1, w_2), (w, w') \in S_r, \quad \uparrow d_w = w' - w$$

и F – cosine similarity rank function

Учёт морфологии

Morphological transformations

rule	hit rate	Example $\uparrow d_w$
suffix:er:o	0.8	$\uparrow d_{\text{Voter}}$
suffix:ton:ϵ	1.1	$\uparrow d_{\text{Galeton}}$
prefix:S:ϵ	1.6	$\uparrow d_{\text{SDK}}$
prefix:ϵ:in	28.8	$\uparrow d_{\text{competent}}$
suffix:ly:ϵ	32.1	$\uparrow d_{\text{officially}}$
prefix:ϵ:re	37.0	$\uparrow d_{\text{sited}}$
prefix:un:re	39.0	$\uparrow d_{\text{unmade}}$
suffix:st:sm	52.5	$\uparrow d_{\text{egoist}}$
suffix:ted:te	54.9	$\uparrow d_{\text{imitated}}$

w_1	w_2	rank	cosine
create	created	0	0.58
create	creates	0	0.65
create	creates	1	0.62
created	create	0	0.65
creation	create	0	0.52
creation	created	0	0.54
recreations	recreate	2	0.59
recreations	recreating	1	0.53
recreations	Recreations	81	0.64

[*Soricut and Och. NAACL 2015*]

Примеры правил и сравнение близости слов

Учёт морфологии

Достоинства и недостатки

- Близость векторных представлений для морфологически связанных слов (create - creation)
- Особенное улучшение качества для *синтетических языков*: тех, в которых словоизменение происходит с помощью внутрисловных морфем, а не отдельных слов (делает - is doing)
- Качество плохо улучшается для *высокофлективных языков*: тех, у которых на стыке морфем происходят изменения (мести - метёшь)
- Недостаток 1 и 2: необходимость морфологического парсера

Учёт морфологии

Сравнение разных морфологических моделей

	DE		EN		ES	FR
	GUR350	ZG222	WS353	RW	WS353	RG65
Luong et al. (2013)	-	-	64	34	-	-
Qiu et al. (2014)	-	-	65	33	-	-
Soricut and Och (2015)	64	22	71	42	47	67
sisg	73	43	73	48	54	69

[*Bojanowski et al. ACL 2017*]

- Luong et al. – сумма репрезентаций морфем
- Qui et al. – skipgram-модель
- Soricut and Och – морфологические трансформации
- sisg – сумма n-граммных векторов символов

Учёт морфологии

Сравнение лучшей с неморфологическими моделями

		sg	cbow	sisg-	sisg
AR	WS353	51	52	54	**55**
DE	GUR350	61	62	64	**70**
	GUR65	78	78	**81**	**81**
	ZG222	35	38	41	**44**
EN	RW	43	43	46	**47**
	WS353	72	**73**	71	71
ES	WS353	57	58	58	**59**
FR	RG65	70	69	**75**	**75**
RO	WS353	48	52	51	**54**
RU	HJ	59	60	60	**66**

Table 1: Correlation between human judgement and similarity scores on word similarity datasets. We

- sg – skipgram-модель
- sisg- – сумма n-граммных векторов символов с нулевыми векторами для новых слов
- sisg– сумма n-граммных векторов символов с предсказанными векторами для новых слов

Синтаксис

- Большинство современных моделей использует модель *language is sequences of words*
- Recurrent NNs лучше учитывают зависимости, которые зависят от количества слов в последовательности между ними
- Однако часто стоит учитывать long-term зависимости:
  У *человека*, которого я видел вчера утром после завтрака, когда сидел в кресле и читал интересную книгу, *не было волос*.
- Syntactic recency часто работает лучше, чем sequential recency
- Используем модели, учитывающие синтаксис, и прежде всего Constituency Trees и Dependency Trees

Sentence embeddings

Зачем

- Использование информации о соседних словах в виде эмбеддинга предложения
- Снижение размерности признаковых описаний предложений и параграфов
- Оценка близости более сложных языковых единиц: "decline to comment" $\approx$ "would not disclose the terms"

Sentence embeddings

An early attempt

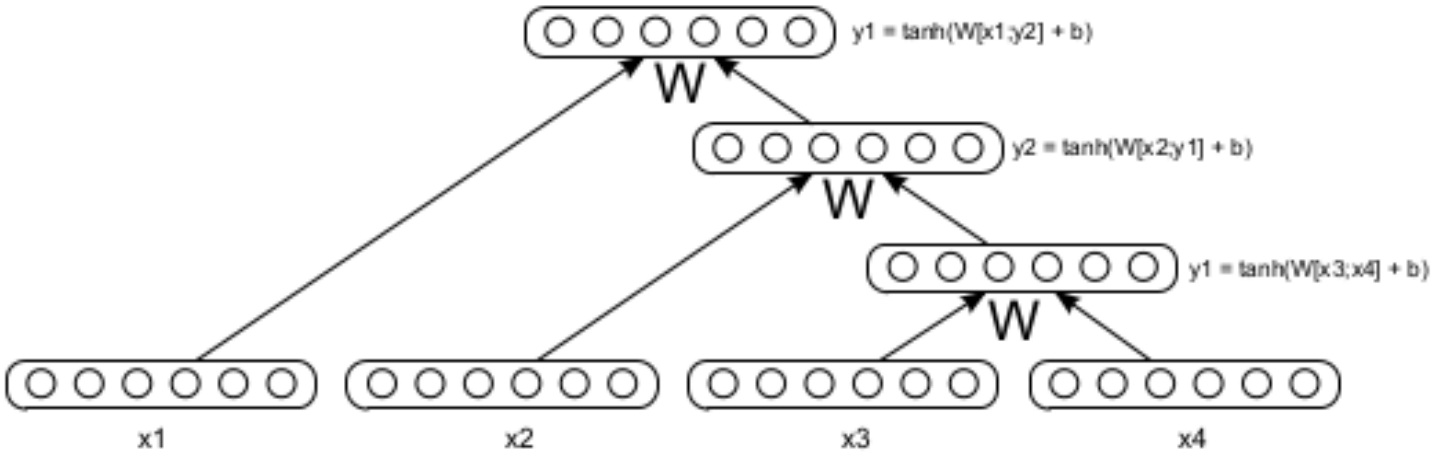


Figure 1: An example tree with a simple Recursive Neural Network: The same weight matrix is replicated and used to compute all non-leaf node representations. Leaf nodes are n-dimensional vector representations of words.

[*Socher*, *Manning*, *Ng*. *NIPS 2010*]

Sentence embeddings

Method	F1
Model 1 (Greedy RNN)	76.55
Model 2 (Greedy, context-sensitive RNN)	83.36
Model 3 (Greedy, context-sensitive RNN + category classifier)	87.05
Model 4 (Global, context-sensitive RNN + category classifier)	92.06
Left Corner PCFG, [MC97]	90.64
Current Implementation of the Stanford Parser, [KM03]	**93.98**

Table 1: Unlabeled Bracketing F-measure computed by evalb on section 22 of the WSJ dataset. Maximum sentence length is 15.

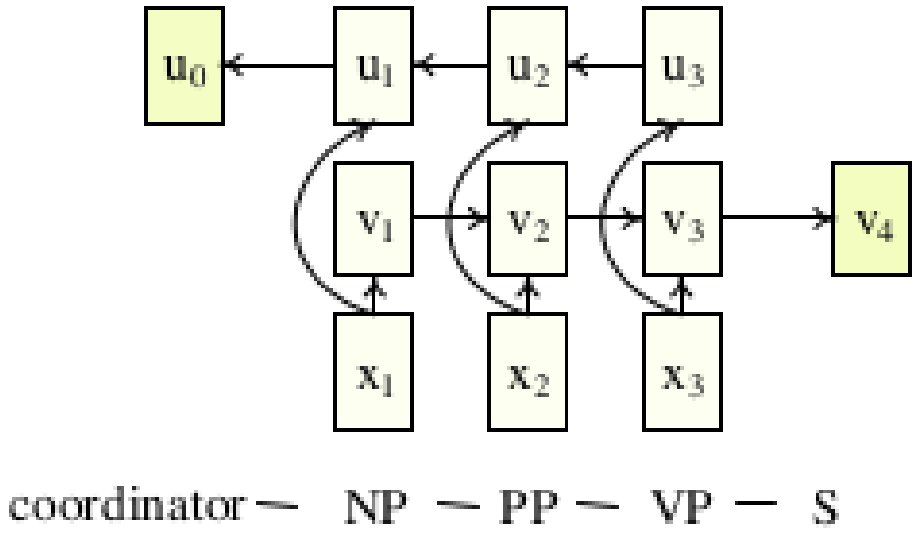
[*Socher*, *Manning*, *Ng*. *NIPS 2010*]

Эта модель была лучше практически всех других на тот момент, за исключением Stanford Parser

Structural Embeddings

- Применить синтаксический парсер (например, StanfordCoreNLP)
- Для каждого слова определить *синтаксическую последовательность* – синтаксические теги вплоть до вершины дерева
- Закодировать синтаксические последовательности в эмбеддинги с помощью RNN или CNN

Structural Embeddings



[*Liu et al. CL 2017*]

Результирующее представление слова $[Ew, u_0, v_4]$, где Ew - это заранее заданный эмбеддинг для *coordinator*, например, word2vec

Structural Embeddings

Method	Single		Ensemble	
	EM	F1	EM	F1
BiDAF	67.69	77.07	72.33	80.33
SECT-LSTM	68.12	77.21	72.83	80.58
SEDT-LSTM	**68.48**	**77.97**	**73.02**	**80.84**

Table 2: Performance comparison on the official blind test set. Ensemble models are trained over the five single runs with the identical network and hyper-parameters.

[*Liu et al. CL 2017*]

Улучшение результатов в задаче Machine Comprehension на датасете SQuAD

Итоги

- В получении эмбеддингов для слов оказывается полезным учитывать полисемию и морфологию
- Для учёта полисемии можно обучать свой вектор для каждого значения или использовать мультимодальное распределение
- Для учёта морфологии можно представлять слово в виде суммы репрезентаций морфем или n-грамм
- Учёт морфологии работает достаточно хорошо для богатых морфологией языков
- Хорошо бы учитывать синтаксис, т.к. во многих задачах имеет значение не sequential recency, а syntactic recency
- Для этого в эмбеддинги нужно включать информацию из деревьев

References

- http://www.abigailsee.com/2017/08/30/four-deep-learning-trends-from-acl-2017-part-1.html
- Li, Jurafsky. ACL 2015. Do Multi-Sense Embeddings Improve Natural Language Understanding?
- Athiwaratkun, Wilson. ACL 2017. Multimodal Word Distributions
- Soricut and Och. NAACL 2015. Unsupervised Morphology Induction Using Word Embeddings
- Bojanowski et al. ACL 2017. Enriching Word Vectors with Subword Information
- Liu et al. CL 2017. Structural Embedding of Syntactic Trees for Machine Comprehension
- Socher, Manning, Ng. NIPS 2010. Learning Continuous Phrase Representations and Syntactic Parsing with Recursive Neural Networks