Работа с текстом

Эмбеддинги и нейросети

февраль, 2017

CNN

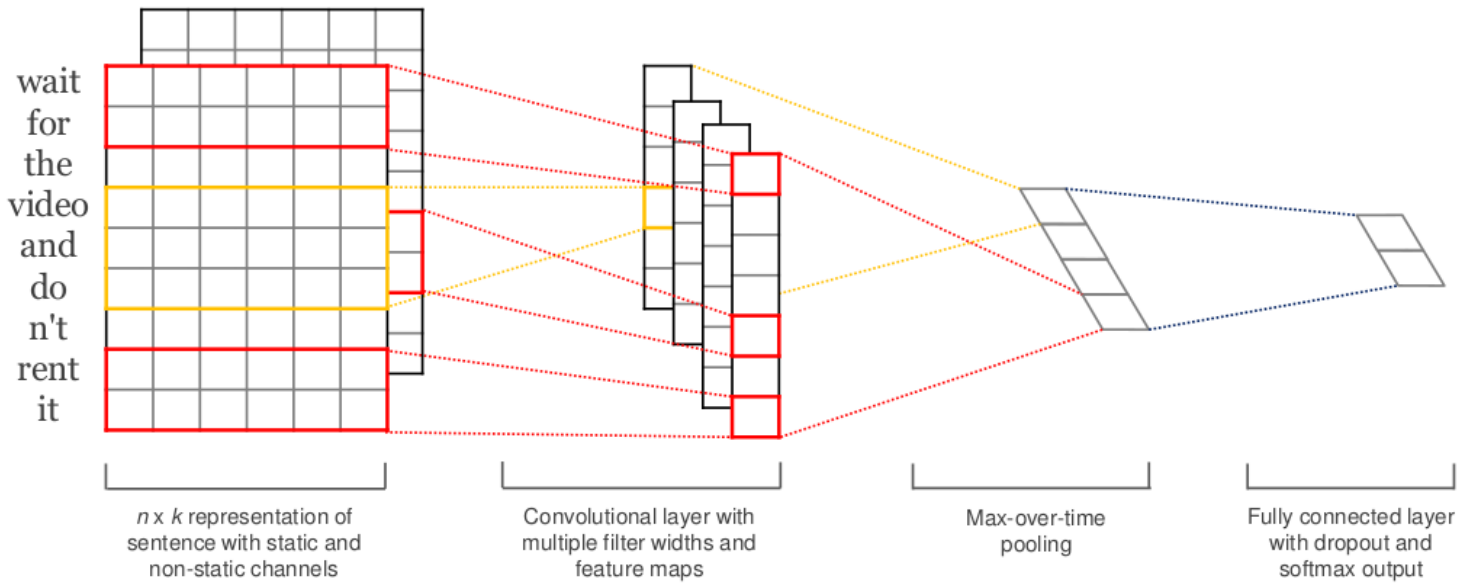
Задачи

- sentiment analysis
- question classification

Модель

- Векторные представления, обученные на 100 миллиардов слов из Google News (Mikolov et al., 2013)
- 1 сверточный слой
- 1 полносвязный слой

Модель


wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Модель

- Представление предложения
  $x_{1:n} = x_1 \bigoplus x_2 \bigoplus \cdots \bigoplus x_n$
- Свертка
  $c_i = f(w * x_{i:i+h} + b)$
  h – ширина окна,
  $b \in \mathbb{R}$ – смещение(bias)
  f $([x_{1:h}, x_{2:h+1} \ldots x_{n-h-1:n}]) \rightarrow$
  $\rightarrow [c_1, c_2 \ldots c_{n-h-1}] := c \in \mathbb{R}^{n-h-1}$
- Max pooling
  $\hat{c} = max\{c\}$

Модель

- Регуляризация
  1.$dropout(p = 0.5) \rightarrow$ отн. прирост качества 2-4%:
  $z := [\hat{c_1}, \hat{c_2} \dots \hat{c_m}]$
  $y = wz + b \rightarrow y = w(z \circ r + b)$,
  r – маска $\in \mathbb{R}^m$, $r_i \sim Ber(p = 0.5)\ \forall i = 1 \dots m$

  2. если $\|w\|_2 > s$, $w := \frac{w}{s}$

Модификации

- CNN-rand:
  инициализация слов случайными векторами
- CNN-static:
  инициализация векторами, обученными на Google News
- CNN-nonstatic:
  инициализация векторами, обученными на Google News + fine-tuning
- CNN-multichannel:
  два «канала» – static и nonstatic, оба инициализированы векторами, обученными на Google News.

Датасеты

Датасет	Количество классов	Обучающая выборка	Тестовая выборка	Тип задачи
MR	2	9056	1006	sentiment analysis
SST-1	5	11855	2210	sentiment analysis
SST-2	2	9613	1821	sentiment analysis
Subj	23	9000	1000	subjectivity analysis
TREC	6	5452	500	question classification
CR	2	3398	377	sentiment analysis
MPQA	2	9545	1061	sentiment analysis

Результаты

Model	MR	SST-1	SST-2	Subj	TREC	CR	MPQA
CNN-rand	76.1	45.0	82.7	89.6	91.2	79.8	83.4
CNN-static	81.0	45.5	86.8	93.0	92.8	84.7	**89.6**
CNN-non-static	**81.5**	48.0	87.2	93.4	93.6	84.3	89.5
CNN-multichannel	81.1	47.4	**88.1**	93.2	92.2	**85.0**	89.4
RAE (Socher et al., 2011)	77.7	43.2	82.4	-	-	-	86.4
MV-RNN (Socher et al., 2012)	79.0	44.4	82.9	-	-	-	-
RNTN (Socher et al., 2013)	-	45.7	85.4	-	-	-	-
DCNN (Kalchbrenner et al., 2014)	-	48.5	86.8	-	93.0	-	-
Paragraph-Vec (Le and Mikolov, 2014)	-	**48.7**	87.8	-	-	-	-
CCAE (Hermann and Blunsom, 2013)	77.8	-	-	-	-	-	87.2
Sent-Parser (Dong et al., 2014)	79.5	-	-	-	-	-	86.3
NBSVM (Wang and Manning, 2012)	79.4	-	-	93.2	-	81.8	86.3
MNB (Wang and Manning, 2012)	79.0	-	-	**93.6**	-	80.0	86.3
G-Dropout (Wang and Manning, 2013)	79.0	-	-	93.4	-	82.1	86.1
F-Dropout (Wang and Manning, 2013)	79.1	-	-	**93.6**	-	81.9	86.3
Tree-CRF (Nakagawa et al., 2010)	77.3	-	-	-	-	81.4	86.1
CRF-PR (Yang and Cardie, 2014)	-	-	-	-	-	82.7	-
SVM S (Silva et al., 2011)	-	-	-	-	**95.0**	-	-

Вывод

- При своей простоте однослойный CNN показывает высокие результаты даже при небольшом тюнинге гиперпараметров
- Использование предобученных векторов слов дает значительный прирост в качестве

ConvNet

Задачи

- text categorization
- sentiment analysis

Идея

Чтобы эффективно обучать модели, нам необязательно располагать знаниями о структуре языка. Спустимся на самый низкий уровень – символьный.

Предобработка

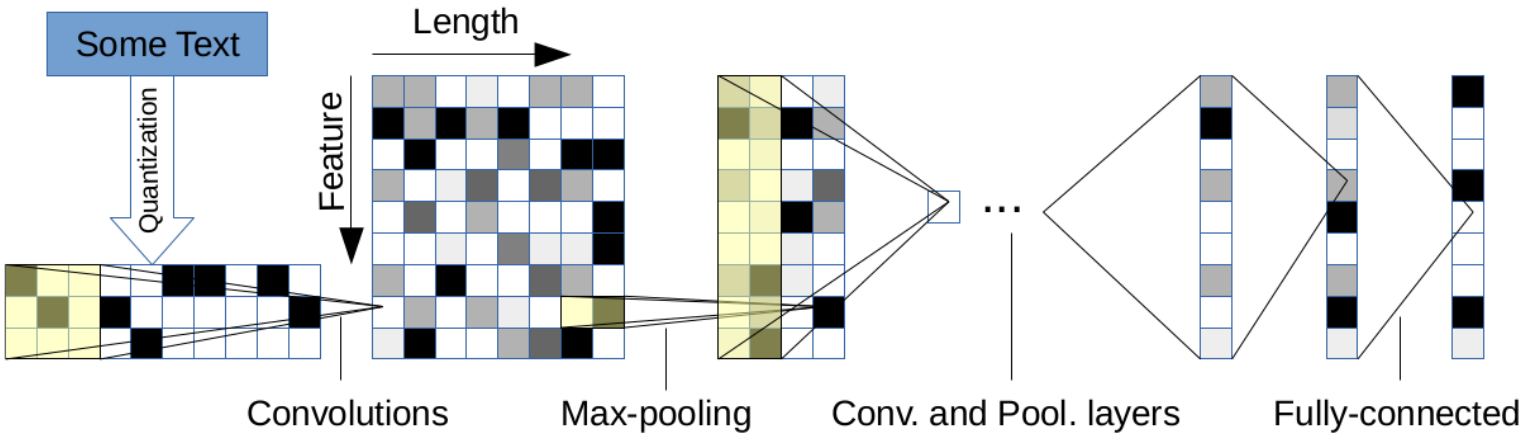
1. Замена слов синонимами из mytheas от LibreOffice: из всех имеющих синонимы слов выбираются случайные $r : P(r) = p^r$. Каждое заменяется синонимом с индексом s в отсортированном по схожести списке синонимов, $P(s) = q^s$.

2. 1-of-M («one-hot») кодирование символов алфавита, где M – размер алфавита. Регистр не учитывается. Неалфавитные символы кодируются нулевым вектором. На вход подается последовательность символов длины l_0.

Модель

- 6 сверточных слоев
- 3 полносвязных слоя
- две модели:
  Large(1024 признака на каждом сверточном слое, 2048 выхода с каждого полносвязного, кроме последнего) и Small(256 и 1024)
- инициализация весов случайными векторами из нормального распределения со стандартным отклонением 0.02 (Large) и 0.05 (Small)

Модель


Some Text
Quantization
Length
Feature
...
Convolutions
Max-pooling
Conv. and Pool. layers
Fully-connected

Модель

$g(x) \in [1, l] \to \mathbb{R}$ – дискретная функция входа
$f(x) \in [1, k] \to \mathbb{R}$ – дискретное ядро

$h(y) \in [1, [(l - k)/d] + 1] \to \mathbb{R}$ – временная свертка (temporal convolution):

$$h(y) = \sum_{x=1}^{k} f(x) * g(y * d - x + c),$$

где $c = k - d + 1$ – смещение (offset)

temporal max-pooling:

$$h(y) = \max_{x=1}^{k} g(y * d - x + c)$$

Регуляризация – $dropout(p = 0.5)$

Датасеты и задачи

Датасет	Количество классов	Обучающая выборка	Тестовая выборка	Тип задачи
AG's News	4	120000	7600	text categorization
Sogou News	5	450000	60000	text categorization
DBPedia	14	560000	70000	text categorization
Yelp Review Polarity	2	560000	38000	sentiment analysis
Yelp Review Full	5	650000	50000	sentiment analysis
Yahoo! Answers	10	1400000	60000	text categorization
Amazon Review Full	5	3000000	650000	sentiment analysis
Amazon Review Polarity	2	3600000	400000	sentiment analysis

Результаты

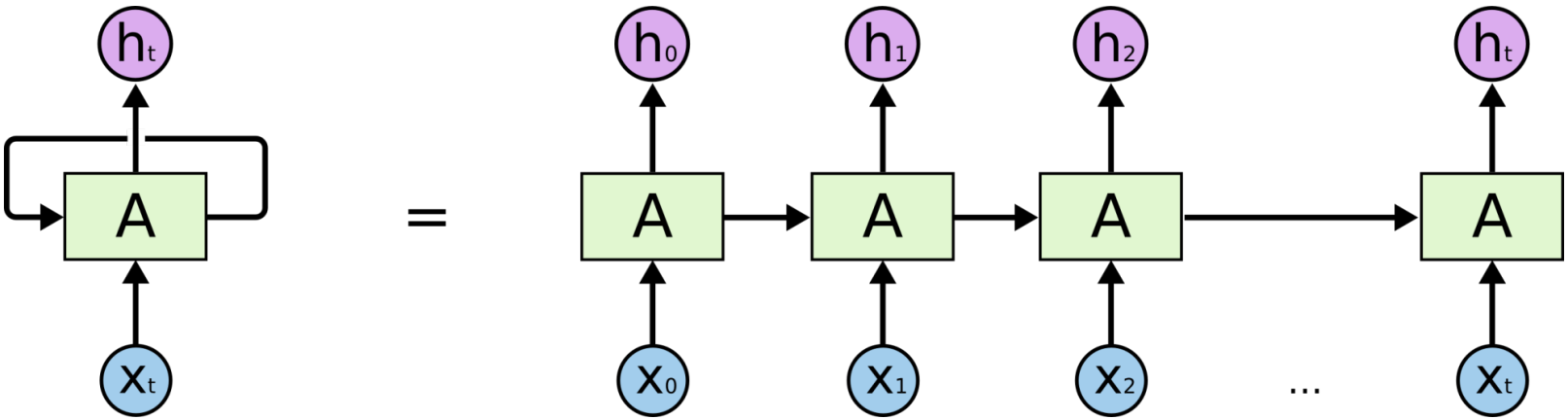
Model	AG	Sogou	DBP	Yelp P.	Yelp F.	Yah. A.	Amz. F.	Amz. P.
BoW	11.19	7.15	3.39	7.76	42.01	31.11	45.36	9.60
BoW TFIDF	10.36	6.55	2.63	6.34	40.14	28.96	44.74	9.00
ngrams	7.96	2.92	1.37	**4.36**	43.74	31.53	45.73	7.98
ngrams TFIDF	**7.64**	**2.81**	**1.31**	4.56	45.20	31.49	47.56	8.46
Bag-of-means	16.91	10.79	9.55	12.67	47.46	39.45	55.87	18.39
LSTM	13.94	4.82	1.45	5.26	41.83	29.16	40.57	6.10
Large + w2v	9.92	4.39	1.42	4.60	40.16	31.97	44.40	5.88
Small + w2v	11.35	4.54	1.71	5.56	42.13	31.50	42.59	6.00
Large + w2v + thesaurus	9.91	-	1.37	4.63	39.58	31.23	43.75	5.80
Small + w2v + thesaurus	10.88	-	1.53	5.36	41.09	29.86	42.50	5.63
Large + lookup table	8.55	4.95	1.72	4.89	40.52	29.06	45.95	5.84
Small + lookup table	10.87	4.93	1.85	5.54	41.41	30.02	43.66	5.85
Large + lookup table + thesaurus	8.93	-	1.58	5.03	40.52	28.84	42.39	5.52
Small + lookup table + thesaurus	9.12	-	1.77	5.37	41.17	28.92	43.19	5.51
Large + lower/upper	9.85	8.80	1.66	5.25	38.40	29.90	40.89	5.78
Small + lower/upper	11.59	8.95	1.89	5.67	38.82	30.01	40.88	5.78
Large + lower/upper + thesaurus	9.51	-	1.55	4.88	38.04	29.58	40.54	5.51
Small + lower/upper + thesaurus	10.89	-	1.69	5.42	**37.95**	29.90	40.53	5.66
Large	12.82	4.88	1.73	5.89	39.62	29.55	41.31	5.51
Small	15.65	8.65	1.98	6.53	40.84	29.84	40.53	5.50
Large + thesaurus	13.39	-	1.60	5.82	39.30	**28.80**	40.45	**4.93**
Small + thesaurus	14.80	-	1.85	6.49	40.16	29.84	**40.43**	5.67

Выводы

- Слова не являются атомарными носителями информации: можно эффективно работать с текстами на символьном уровне
- Преимущества ConvNet проявляются на больших датасетах (от нескольких миллионов), на небольших серьезную конкуренцию ему составляют традиционные методы – в частности, bag-of-ngrams + TFIDF
- ConvNet более эффективно работает на менее структурированных(curated) текстах (ср. отзывы на Amazon и ответы на Yahoo! Answers)

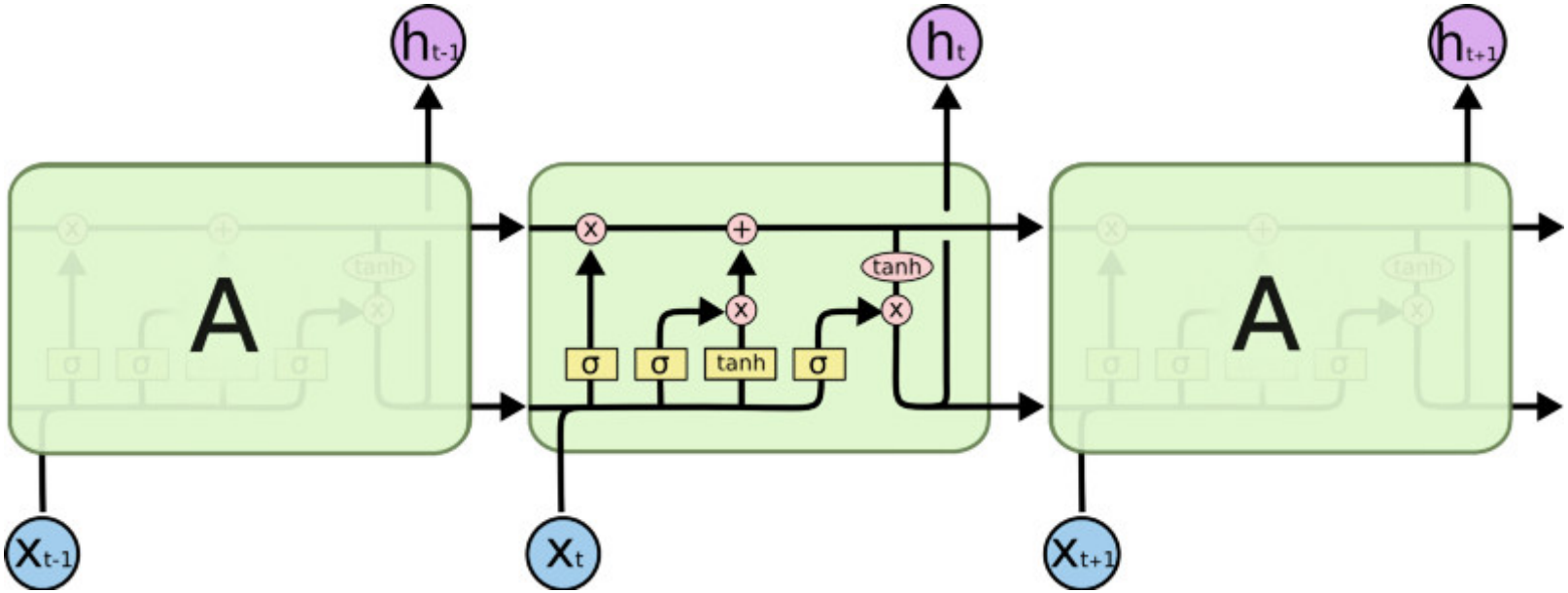
RNN

Simple RNN

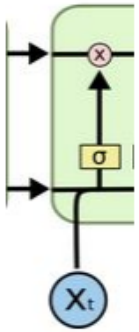


$$s_t = f_s\left(Ux_t + Ws_{t-1}\right)$$
$$o_t = f_o\left(Vs_t\right)$$

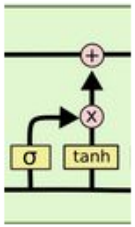
LSTM



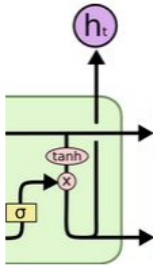
LSTM



$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right)$$

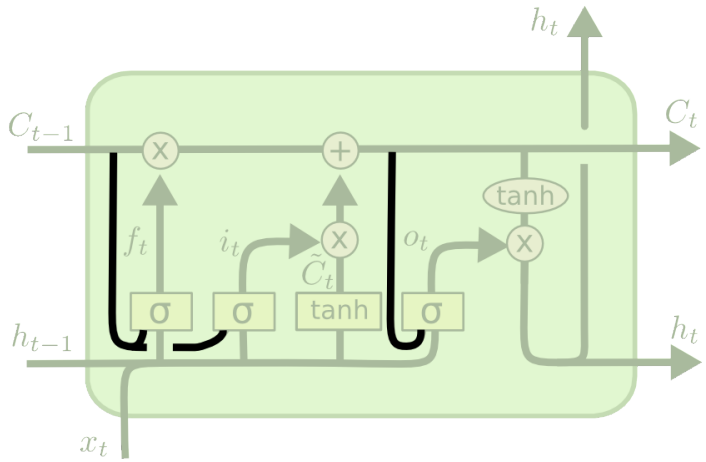


$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right)$$
$$\tilde{C}_t = \sigma\left(W_c x_t + U_c h_{t-1} + b_c\right)$$
$$C_t = i_t \bigodot \tilde{C} = f_t \bigodot C_{t-1}$$



$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right)$$
$$h_t = o_t \bigodot tahn(C_t)$$

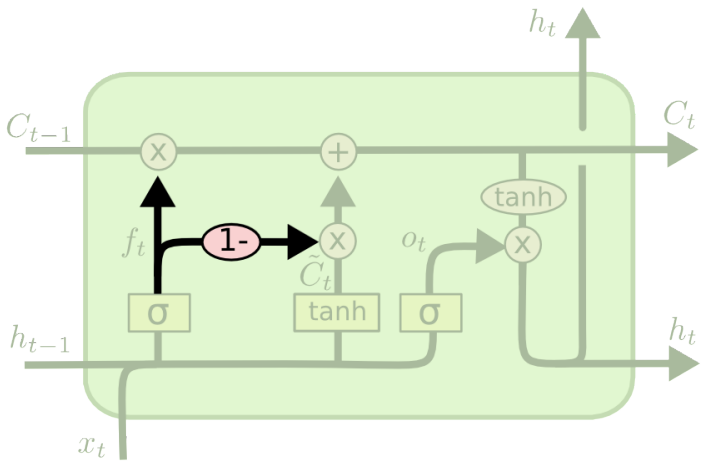
Вариации LSTM



$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] \; + \; b_o\right)$$

Figure: LSTM with «Peephole Connections»

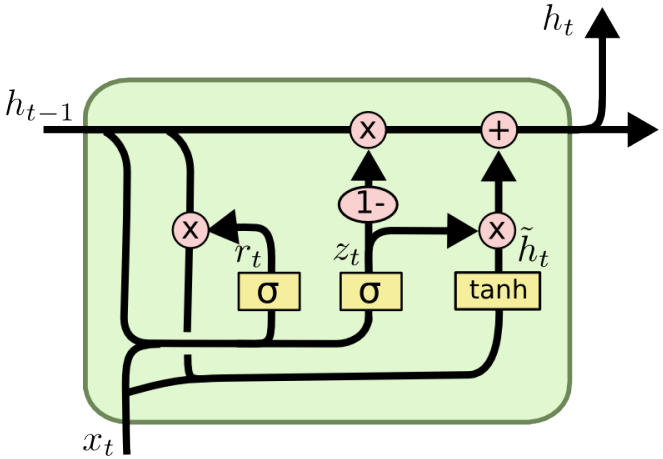
Вариации LSTM



$$C_t = f_t * C_{t-1} + (1 - \boldsymbol{f_t}) * \tilde{C}_t$$

Figure: LSTM with coupled forget and input gates

Вариации LSTM



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

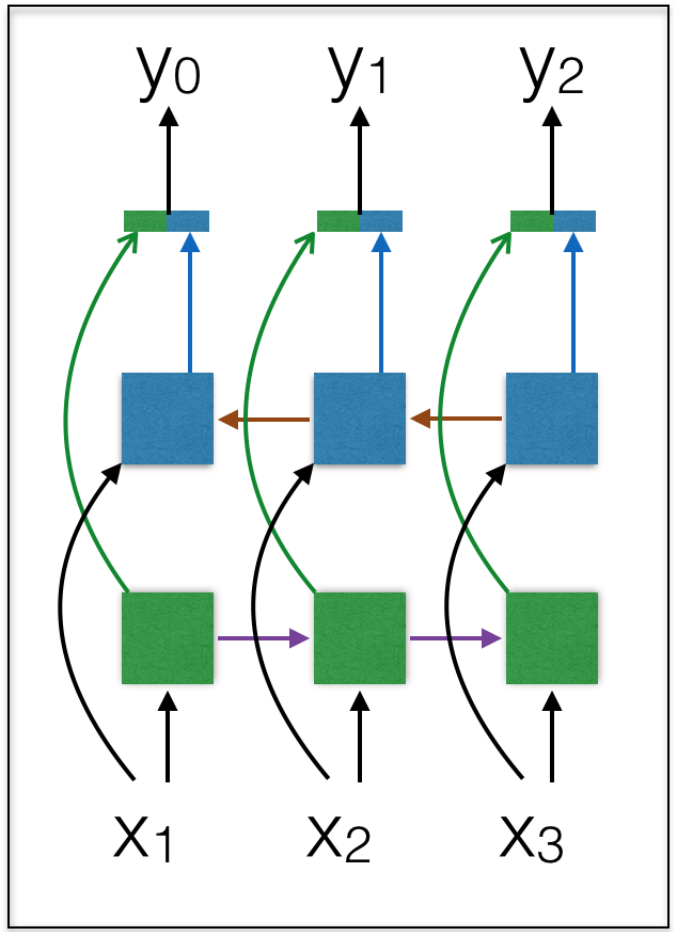
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure: Gated Recurrent Unit(GRU)

Bidirectional LSTM



Задача

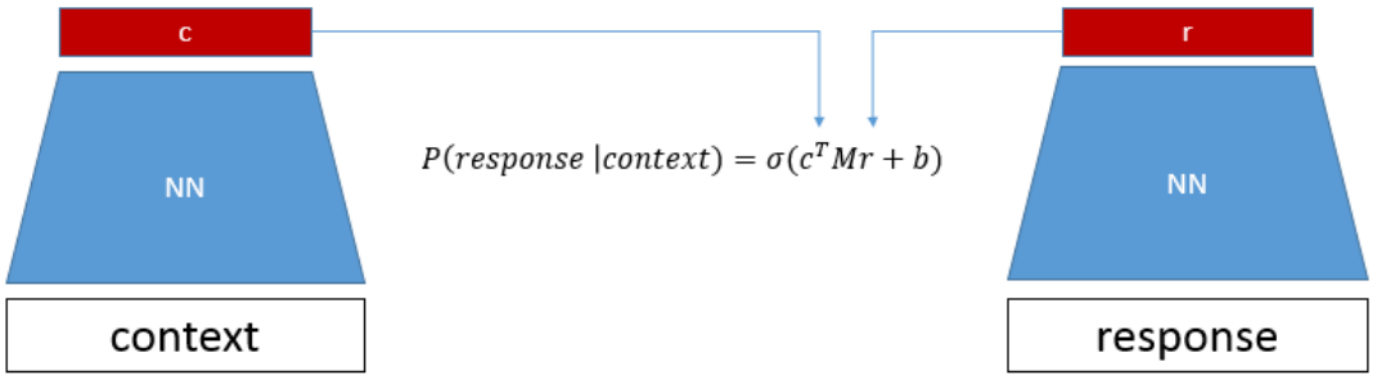
Предсказание следующей реплики в диалоге на основе данных из The Ubuntu Dialog Corpus

Предобработка

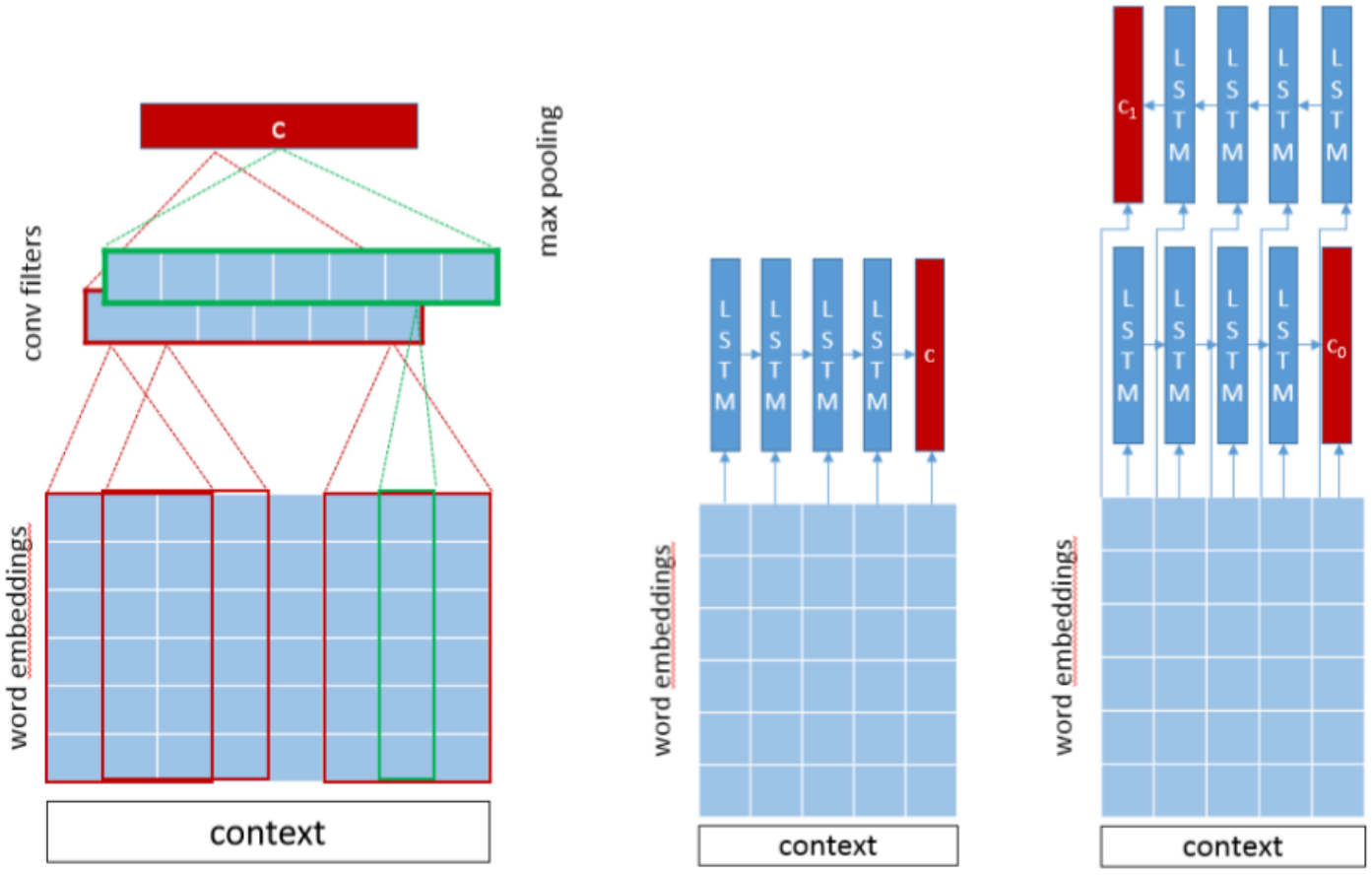
Имена собственные заменены тэгами(имя, место, организация, url etc.), данные были приведены к виду (контекст, ответ, [ответ правильный])

Использованы векторные представления слов GloVe.

Общая схема



Сравнение архитектур



(a) CNN (b) LSTM (c) Bi-Directional

Результаты

Model	TF-IDF	RNN	LSTM	CNN	LSTM	Bi-LSTM	Ensemble
1 in 2 R@1	65.9	76.8	87.8	84.8	90.1	89.5	91.5
1 in 10 R@1	41.0	40.3	60.4	54.9	63.8	63.0	68.3
1 in 10 R@2	54.5	54.7	74.5	68.4	78.4	78.0	81.8
1 in 10 R@5	70.8	81.9	92.6	89.6	94.9	94.4	95.7

Выводы

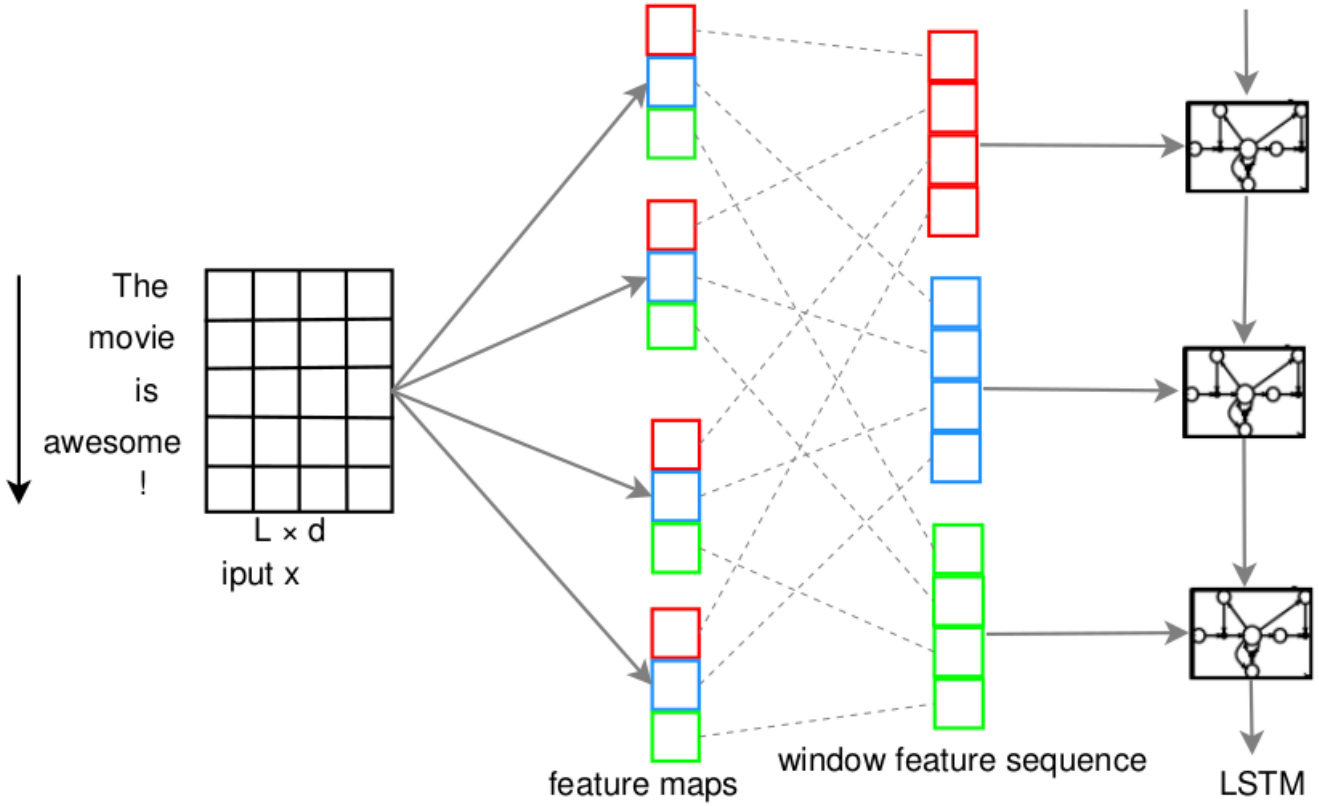
- LSTM и Bi-LSTM имеют почти одинаковые результаты: либо слова в конце предложений действительно важнее, либо обычная LSTM смогла уловить всю нужную информацию
- Ансамбль превосходит каждую модель в отдельности
- присутствие CNN значительно повышает результаты ансамбля, следовательно, представление от CNN дополняет представление от LSTM

C-LSTM

Задачи

- sentiment analysis
- question classification

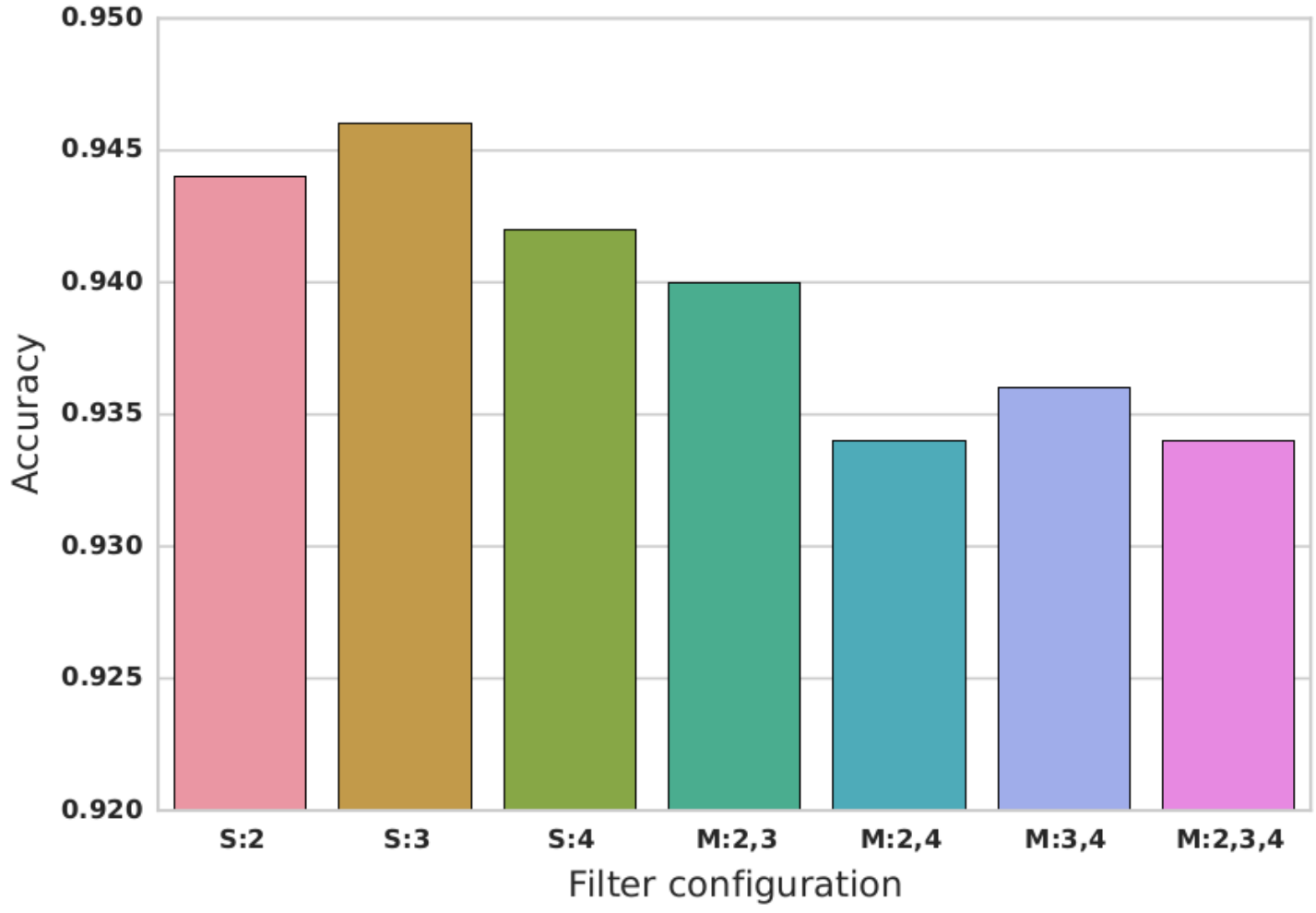
C-LSTM


The
movie
is
awesome
!
L × d
iput x
feature maps
window feature sequence
LSTM

Датасеты

Датасет	Количество классов	Обучающая выборка	Тестовая выборка	Тип задачи
SST-1	5	11855	2210	sentiment analysis
SST-2	2	9613	1821	sentiment analysis
TREC	6	5452	500	question classification

Сравнение сверточных фильтров



Результаты

Model	TREC	SST-1	SST-2
SVM	95.0	40.7	79.4
NBoW	-	42.4	80.5
Paragraph Vector	91.8	48.7	87.8
RAE	-	43.2	82.4
MV-RNN	-	44.4	82.9
RNTN	-	45.7	85.4
DRNN	-	49.8	86.6
Ada-CNN	92.4	-	-
CNN-nonstatic	93.6	48.0	87.2
CNN-multichannel	92.2	47.4	88.1
DCNN	93.0	48.5	86.8
Molding CNN	-	51.2	88.6
Dependency Tree-LSTM	-	48.4	85.7
Constituency Tree-LSTM	-	51.0	88.0
LSTM	93.2	46.6	86.6
Bi-LSTM	93.0	47.8	87.9
C-LSTM	94.6	49.2	87.8

Выводы

- в задачах sentiment analysis C-LSTM уступает только моделям, которые использовали лингвистические признаки и алгоритмы
- в задаче question classification C-LSTM превосходит почти все опубликованные базовые модели, проигрывая только SVM поверх свыше 60 вручную полученных признаков

Статьи

- CNN – Convolutional Neural Networks for Sentence Classification, Kim [1408.5882]
- ConvNet – Character-level Convolutional Networks for Text Classification, Zhang et al., [1509.01626]
- RNN – Improved Deep Learning Baselines for Ubuntu Corpus Dialogs, Kadlec et al., [1510.03753]
- C-LSTM – A C-LSTM Neural Network for Text Classification, Zhou et al., [1511.08630]
- http://colah.github.io/posts/2015-08-Understanding-LSTMs/