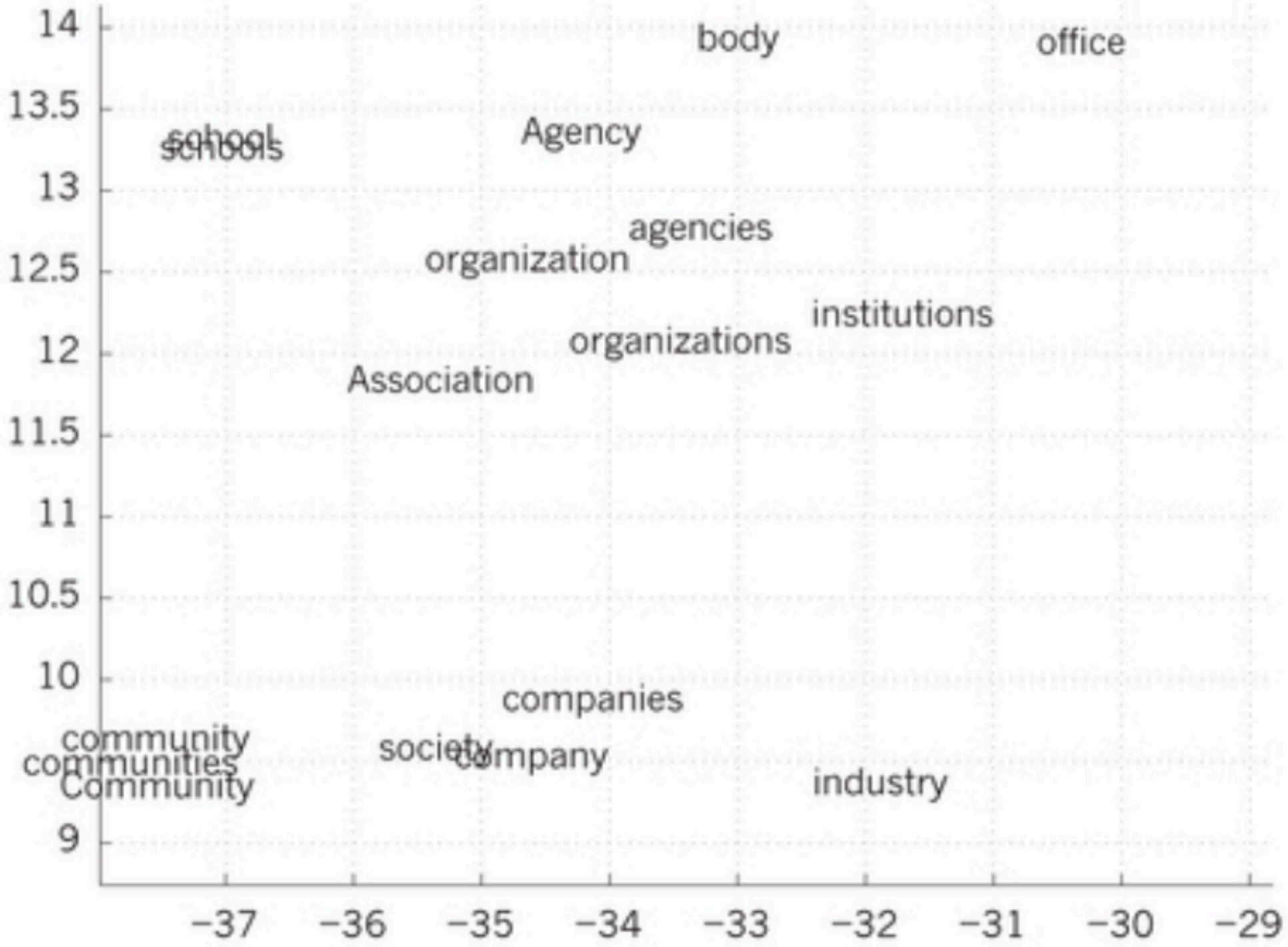
New directions in word representations

Носов Степан
НИУ ВШЭ

25 января 2019

Векторные представления слов

- Вход - коллекция текстов
- Выход - представление каждого слова из словаря вектором небольшой фиксированной размерности
- Цель - похожие семантически и синтаксически слова должны находить близко друг к другу

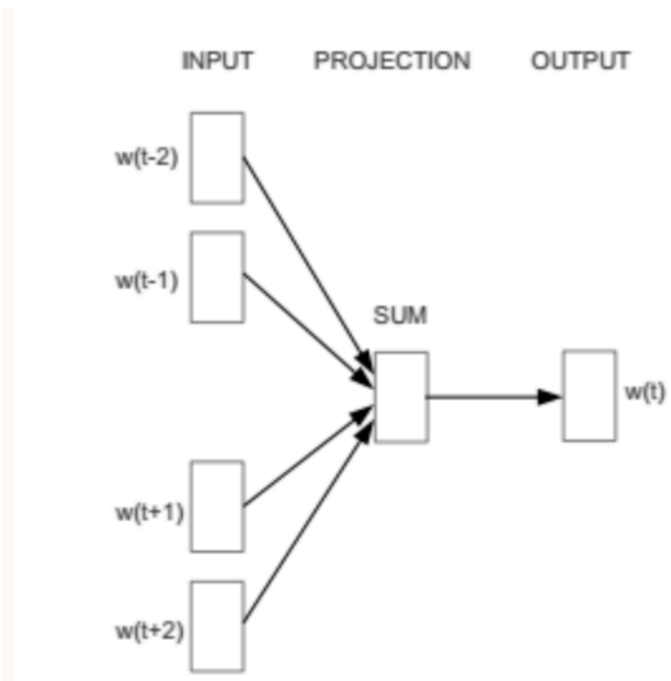

14
13.5
13
12.5
12
11.5
11
10.5
10
9
−37
−36
−35
−34
−33
−32
−31
−30
−29
body
office
school
schools
Agency
agencies
organization
institutions
organizations
Association
companies
community
communities
Community
society
company
industry

Векторные представления слов: зачем?

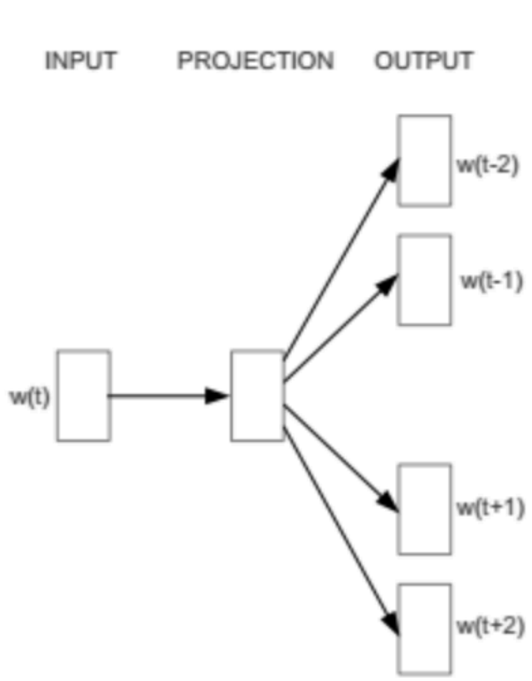
- Тэгирование частей речи
- Машинный перевод
- Ранжирование и кластеризация документов
- Анализ тональности текста

word2vec

- Инструмент для векторного представления слов
- Представлен в 2013 году
- Два основных алгоритма: CBOW(предсказывает слово по контексту) и Skip-gram(предсказывает контекст по слову)



CBOW



Skip-gram

CBOW

$$P(w_o|w_c) = \frac{e^{s(w_o,w_c)}}{\sum_{w_i \in V} e^{s(w_i,w_c)}}$$

- S - функция, считающая расстояние между векторами.
- Оптимизируем отрицательный логарифм правдоподобия
- Проблема - знаменатель посчитать сложно
- Negative Sampling: оптимизируем формулу $NegS(w_o) = \sum_{i=1,x_i \sim D}^{i=k} -log(1 + e^{s(x_i,w_o)}) + \sum_{j=1,x_j \sim D'}^{j=l} -log(1 + e^{-s(x_j,w_o)})$
- Здесь D - распределение встречаемости слов с исходным в одном контексте, D’ - распределение «невстречаемости» слов из словаря с исходным, на практике берется как равномерное по всем словам

ELMo; Deep Contextualised Word Representations

- Проблема традиционного подхода - он не учитывает, что слово может иметь разные значения в зависимости от контекста, так как получает одно представление для каждого слова

- Вместо того, чтобы хранить словарь векторов для каждого слова, ELMo создает векторы «на лету», в зависимости от текста

- Использует двухслойная bi-directional LSTM.

ELMo

$$\sum_{k=1}^{N} (\log p(t_k \mid t_1, \ldots, t_{k-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s) + \log p(t_k \mid t_{k+1}, \ldots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s)) .$$

- Для каждого токена строится множество представлений

- Для дальнейшего использования в других моделях они объединяются в один вектор

$$\begin{aligned} R_k &= \{\mathbf{x}_k^{LM}, \overrightarrow{\mathbf{h}}_{k,j}^{LM}, \overleftarrow{\mathbf{h}}_{k,j}^{LM} \mid j = 1, \ldots, L\} \\ &= \{\mathbf{h}_{k,j}^{LM} \mid j = 0, \ldots, L\}, \end{aligned}$$

$$\mathbf{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM} .$$

ELMo

Task	Previous SOTA		Our baseline	ELMo + baseline	Increase (absolute/ relative)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

Improving Language Understanding by Generative Pre-Training

- Подход состоит из двух стадий - обучение известной модели на больших корпусах текстов, а затем тонкая адаптация к задаче с помощью помеченных данных.

- На 1 этапе(Unsupervised pre-training) максимизируем правдоподобие какой - либо известной моделью.

$$L_1(\mathcal{U}) = \sum_i \log P(u_i|u_{i-k}, \ldots, u_{i-1}; \Theta)$$

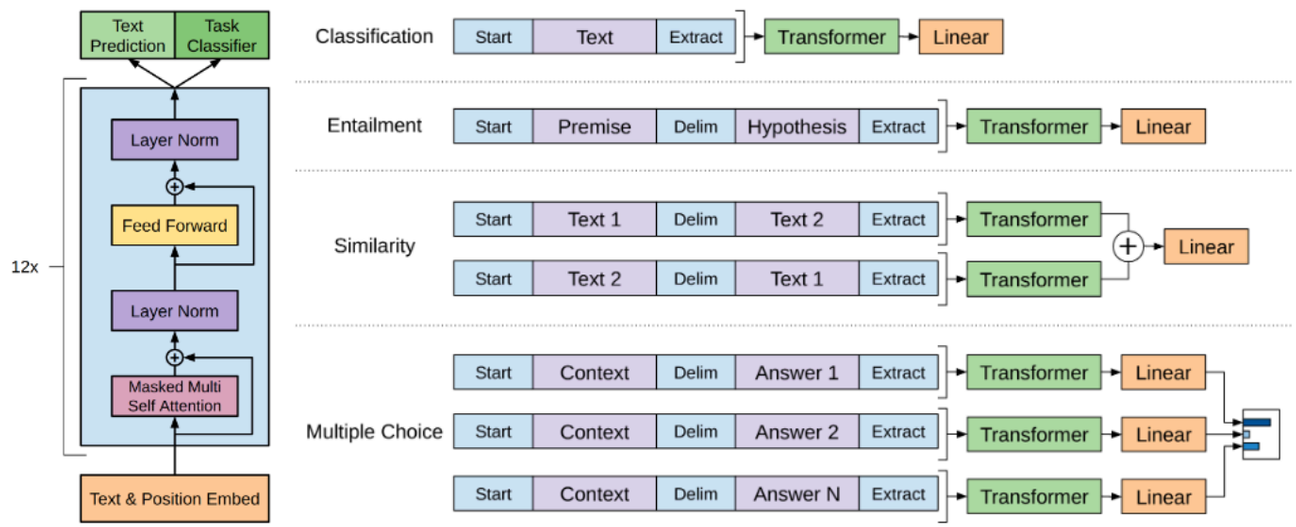
Improving Language Understanding by Generative Pre-Training

- Второй этап(Supervised fine-tuning) заключается в добавлении линейного выходного слоя для предсказания целевой переменной.

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \ldots, x^m).$$

- Оптимизация комбинированной функции правдоподобия улучшает обобщающую способность и ускоряет сходимость

$$L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C})$$



- Для некоторых задач(textual entailment, similarity…) перед второй частью данные предобрабатывают

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	89.3	-	-	-
CAFE [58] (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network [35] (3x)	80.6	80.1	-	-	-	-
CAFE [58]	78.7	77.9	88.5	83.3		
GenSen [64]	71.4	71.3	-	-	82.3	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	**61.7**
Finetuned Transformer LM (ours)	**82.1**	**81.4**	**89.9**	**88.3**	**88.1**	56.0

BERT

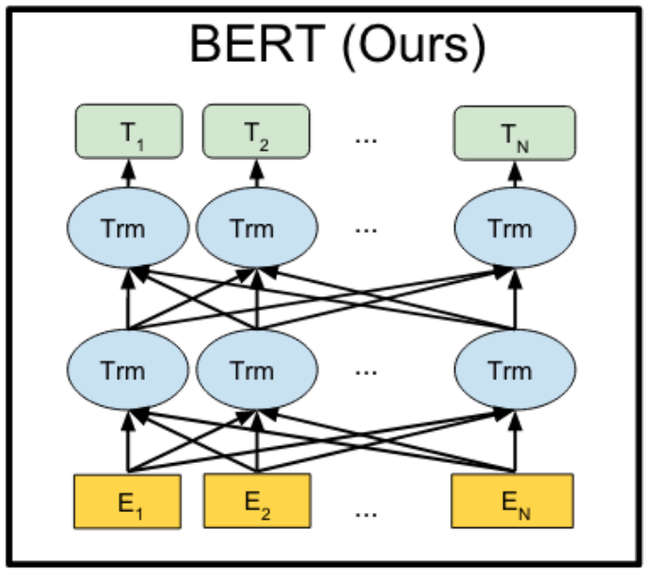
- Bidirectional Encoder Representations from Transoformers

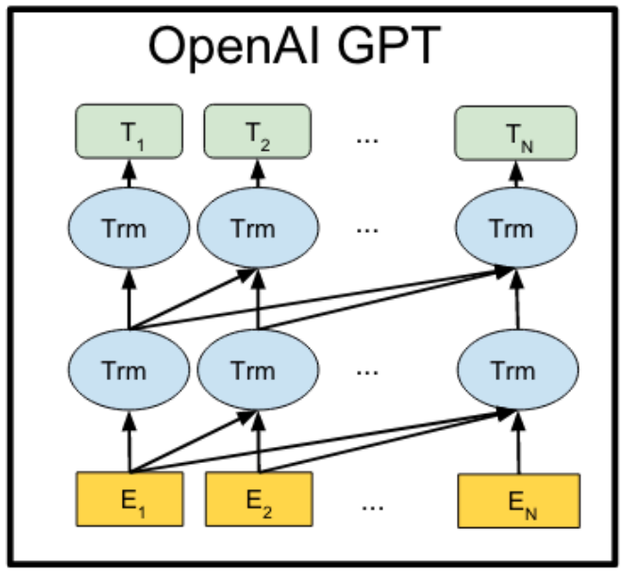
- Существует два подхода для pre-trained language representations: feature-based(ELMo), fine-tuning(Generative Pre-trained Transformer)

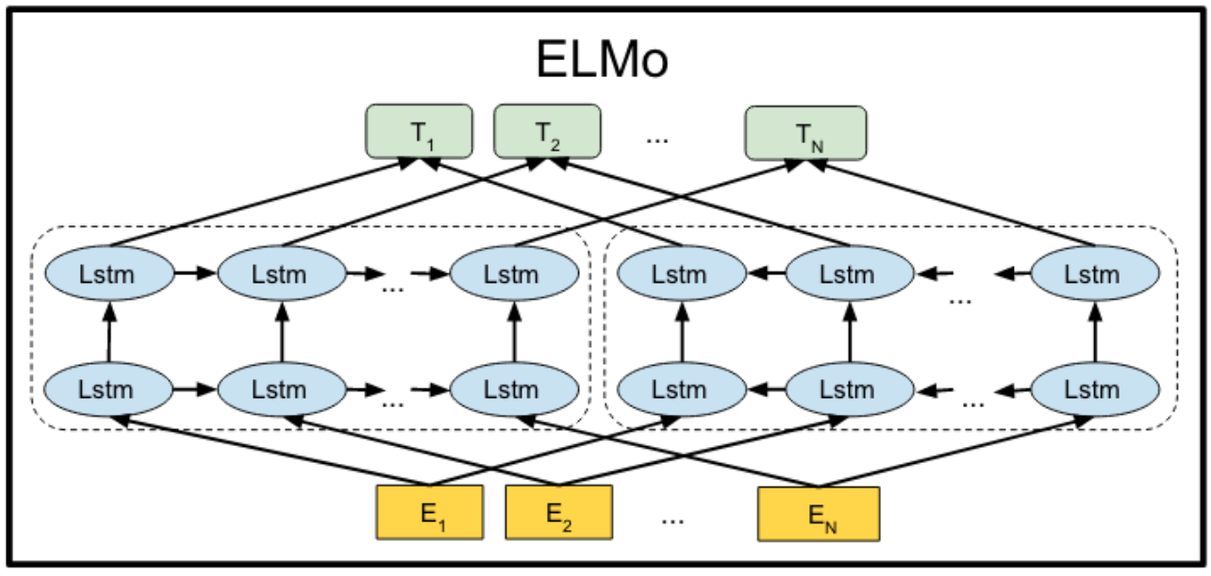
- Первый подход - использовать уникальные архитектуры для разных языковых задач, а полученное представление считать еще одним признаком.

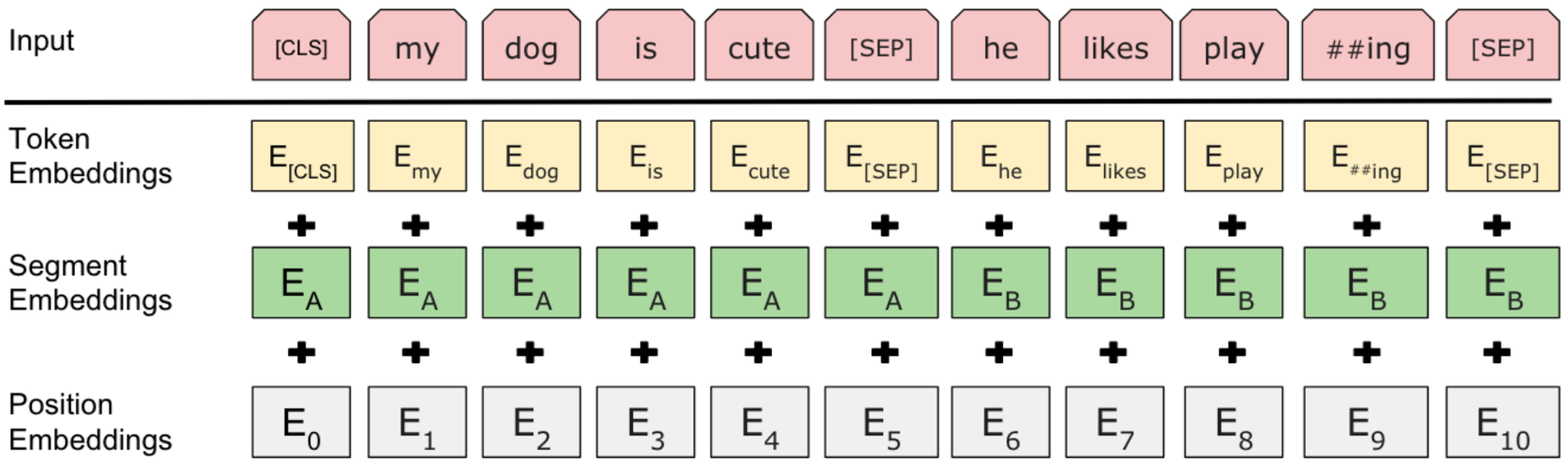
- Второй подход - использовать общую архитектуру, в которой настраивать параметры для соответствующих задач

- В отличие от описанных выше подходов левый и правый контекст анализируются совместно в BERT


BERT (Ours)
T1
T2
...
TN
Trm
Trm
...
Trm
Trm
Trm
...
Trm
E1
E2
...
EN


OpenAI GPT
T1
T2
...
TN
Trm
Trm
...
Trm
Trm
Trm
...
Trm
E1
E2
...
EN


ELMo
T1
T2
...
TN
Lstm
Lstm
...
Lstm
Lstm
Lstm
...
Lstm
Lstm
Lstm
...
Lstm
Lstm
Lstm
...
Lstm
E1
E2
...
EN


Input
[CLS] my dog is cute [SEP] he likes play ##ing [SEP]
Token Embeddings
$E_{[CLS]}$ E_{my} E_{dog} E_{is} E_{cute} $E_{[SEP]}$ E_{he} E_{likes} E_{play} $E_{\#\#ing}$ $E_{[SEP]}$
Segment Embeddings
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Position Embeddings
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

Masked LM

- Помечаем [MASK] некоторые случайно выбранные(в исследовании брали 15%) токены, затем предсказываем только их

- Проблема - модель может плохо строить представления непомеченных слов

- Решение - в 80% случаев заменяем на [MASK], в 10% - на рандомное слово, в 10% - не меняем

Next Sentence Prediction

- Некоторые важные задачи требуют понимания взаимоотношения между двумя предложениями

- Тренируем бинарную модель, которая отвечает - является ли одно предложение следующим для другого

Pre-training

- Для генерации каждой входной последовательности данных берем два отрезка текста и называем их «предложениями». В половине случаев они действительно последовательны, в половине второе выбирается случайно.

- Максимизируем сумму правдоподобий masked LM и Next Sentence Prediction

Fine-tuning

- Для задач классификации используем тонкую настройку параметров - преобразовываем размеченные данные, забирая представления, полученные на последнем скрытом слое и подавая их на вход аналогичной модели с дополнительным классифицирующем слоем в конце. Обучаем, максимизируя log-probability классов.

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**