Д. Н. Пайвин, А. В. Глазкова

Тюменский государственный университет, г. Тюмень

УДК 004.912

ИССЛЕДОВАНИЕ МЕТОДОВ ВЕКТОРНОГО ПРЕДСТАВЛЕНИЯ ЕСТЕСТВЕННОГО ЯЗЫКА НА ПРИМЕРЕ КЛАССИФИКАЦИИ КОРОТКИХ ТЕКСТОВ

Аннотация. В данной статье рассматриваются современные методы построения распределенного представления естественного языка. Подробно рассматриваются три основные модели: Word2Vec, Fasttext, GloVe. Сравнительный анализ выбранных алгоритмов проводится на задаче классификации текстов.

Ключевые слова: word embedding, NLP, классификация тестов, интеллектуальный анализ данных.

Введение

Обработка естественного языка (Natural Language Processing, NLP) — общее направление искусственного интеллекта и математической лингвистики. Оно изучает проблемы компьютерного анализа и синтеза естественных языков. NLP включает в себя большое число задач. В этой работе будут рассмотрены следующие задачи: построение модели языка и классификация текста. Обработку естественного языка можно представить в виде пирамиды (рис. 1), состоящей следующих частей [1]:

1. Морфология: определение основных характеристик слова, таких как: часть речи, лемматизация и т.д.

2. Синтаксис: происходит построение дерева зависимостей и синтаксического дерева.

3. Семантика: происходит работа с фактическим пониманием естественного языка.

4. Прагматика: производится анализ текста в целом: определение темы текста, выделение подтем в тексте и т.д.

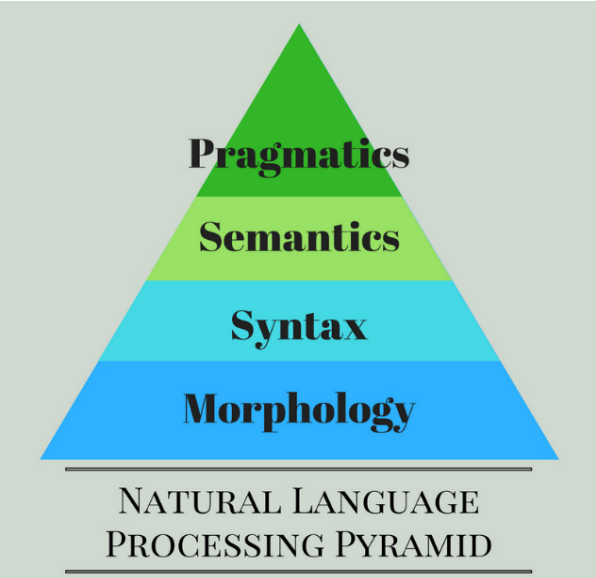

Рис. 1. Пирамида обработки естественного языка.

В данной работе подробно рассматривается третий этап пирамиды.

Распределенные представления слов

Согласно дистрибутивной гипотезе Харриса, слова с похожим смыслом будут встречаться в схожих контекстах. Именно эта гипотеза легла в основу современного подхода к распределенному представлению слов. [2]

Распределенное представление слов (word embeddings) — это класс методов, в которых отдельные слова представлены в виде вещественных векторов заданной размерности. В этой работе рассматривается 3 модели для построения распределенного представления слов: Word2Vec, Fasttext, GloVe.

Word2Vec

Идея, положенная в основу модели word2vec, была заложена в работе [3] и включает в себя 2 метода: метод непрерывного мешка слов (continuous bag of words, CBOW) и метод архитектуры skip-ngram.

Суть метода CBOW заключатся в том, чтобы по заданному контексту слова восстановить само слово. Предсказание искомого слова делается неглубокой нейронной сетью [4]. Архитектура нейронной сети,

соответствующей модели CBOW, изображена на рис. 2. Процесс построения векторного пространства можно описать следующим алгоритмом:

1. На вход подается корпус T для которого составляется словарь W, т.е. список всех уникальных слов. Из словаря удаляются наиболее редкие слова.

2. Для каждого слова $\omega_i \epsilon T$ собирается контекст, т.е. набор слов $c_i \epsilon W$, удаленных не более чем на s позиций в последовательности слов корпуса T: $c_i = \{\omega_i \epsilon T : (i - s) \leq j \leq (i + s), j \neq i\}$.

3. Выполняется унитарное кодирование [5] (one-hot encoding) словаря W, т.е. каждому слову $\omega_i \epsilon W$ ставится в соответствие вектор $u_i \epsilon U$ из нулей и одной единицы, размер вектора u_i равен размеру словаря W, позиция единицы в векторе u_i соответствует номеру i-ого слова в словаре W.

4. Для формирования распределенных векторов слов применяется нейронная сеть, состоящая из 3-х слоёв.

5. Входной слой представляет собой матрицу C x U, где C – количество слов в контексте, U – количество слов в словаре.

6. Скрытый слой представляет собой матрицу L, где n-ая – строка матрицы содержит вектор n-ого слова из словаря. При вычислении выхода берется среднее всех входных векторов из скрытого слоя. Выход представляет собой некоторую оценку k_j для каждого слова в словаре.

7. Для расчета апостериорного распределения модели применяется функция softmax: $p(i|c_1, \ldots\ldots, c_n) = \frac{\exp(k_f)}{\sum_{i=1}^{W} \exp(k_i)}$;

8. Функция потерь на одном окне выглядит следующим образом: $L = p(t|c_1, \ldots\ldots, c_n) = k_j + \log(\sum_{i=1}^{|W|} \exp(k_i))$.

Метод skip-ngram работает противоположным образом: теперь не предсказывается слово по усредненному контексту, а предсказывается каждое слово контекста по данному слову.

Ключевые отличия метода skip-ngram от CBOW можно описать следующими пунктами [6]:

1. На выходном слое нейронной сети теперь получается n мультиноминальных распределений, по одному для каждого слова контекста: $p(c_r|i) = \frac{\exp(k_{rc_r})}{\sum_{i=1}^{W} \exp(k_i)}$;

2. Функция потерь модели на одном окне выглядит:

$$L = -logp(c_1, \ldots\ldots, c_n|i) = -\sum_{r=1}^{n} k_r + n * log \sum_{i=1}^{|W|} \exp(k_i).$$

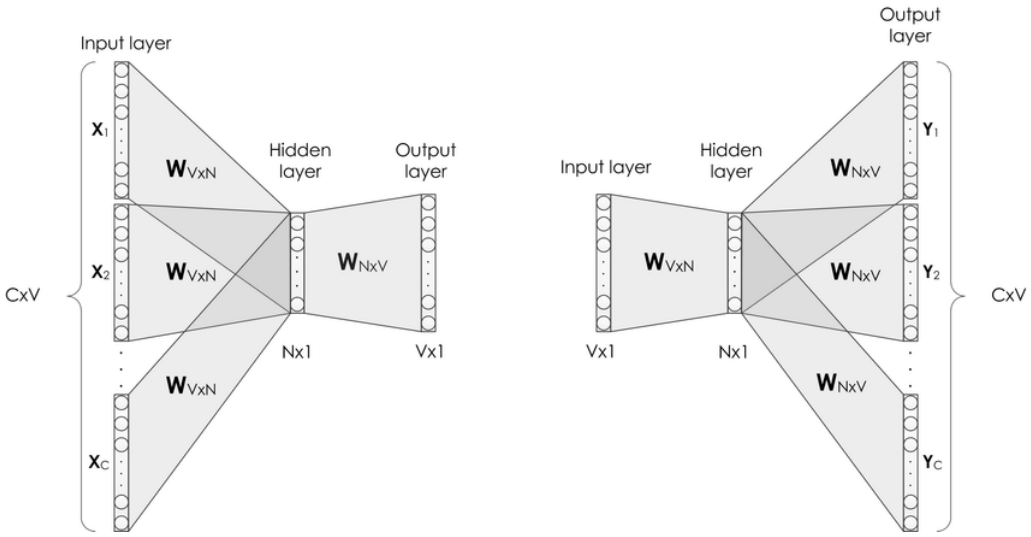

Рис. 2. Модель word2vec. Слева – CBOW, справа – Skip-Ngram.

Fasttext

В последние годы было предложено много методов для включения морфологической информации в представления слов, чтобы лучше моделировать редкие слова. Однако ближе всего к этой цели подобралась исследовательская группа из Facebook AI Research, которая предложила свой метод, для построения векторного пространства языка, под названием Fasttex.

Основная концепция Fasttext'а была позаимствована из модели word2vec. Fasttext имеет два метода для построения векторного пространства: CBOW и skip-ngram.

Модель Fasttext исправляет один из самых важных недостатков word2vec, а именно – игнорирование внутренней структуры слов. В модели Fasttext каждое слово представлено как "мешок" n-грамм, т.е. из слова

выделяются всевозможные последовательности букв длины n, также каждому слову добавлены специальные граничные символы "<"и ">", для разделения префиксов и суффиксов от других символов последовательности. Само слово также включается в набор его n-грамм, чтобы иметь представление для него [7].

В связи с представлением слова как мешка n-грамм, создателям Fasttext'а потребовалось изменить функцию правдоподобия между векторами. Предположим, что дан словарь n-грамм размера G. Через $G_w \subset \{1, \dots, G\}$ обозначим множество n – грамм входящих в слово w. Для каждой n-граммы g_i сопоставим вектор z_g. Представим слова в виде суммы его n-грамм и получим следующую функцию оценки правдоподобия:

$s(w, c) = \sum_{g \in G_w} z_g^t v_c$, где w и слова, а v_c – вектор слова c.

GloVe

Самой популярной альтернативой word2vec являются модели GloVe (global vectors) [8]. Для описания модели введем следующие обозначения: пусть дан словарь размера V, $X \in R^{V \times V}$ – матрица совместной встречаемости слов, где X_{ij} показывает, сколько раз слово i встречается вместе со словом j, $X_i = \sum_j X_{ij}$ – общее число раз, которое все слова из словаря встречались со словом i. Пусть $P_{ij} = P(j|i) = \frac{X_{ij}}{X_i}$ – вероятность того, что слово j появится в контексте слова i. Отправной точкой для идеи GloVe является отношение вероятностей совместного появления слов в корпусе: $\frac{P_{ik}}{P_{jk}}$, т.е. отношение вероятностей встречи слов i и j в контексте слова k. В общем, модель GloVe можно выразить в следующем виде: $F(w_i, w_j, \overline{w_k}) = \frac{P_{ik}}{P_{jk}}$, где $w \in R^d$ – вектор слова, $\overline{w} \in R^d$ – вектор контекста слова.

В качестве функции F модель GloVe использует экспоненту и тогда итоговую функцию можно записать в виде:

$$w_i^T \overline{w_k} = \log(p_{ik}) = \log(X_{ik}) - \log(X_i).$$

Чтобы векторы слов и векторы контекстов стали симметричными достаточно добавить по свободному члену b_i и $\bar{b}_j$. Тогда модель примет окончательный вид: $\log(X_{ik}) = w_i^T \bar{w}_k + b_i + \bar{b}_k$.

Так как модель GloVe использует метод обучения без учителя, то требуется определить целевую функцию. Целевую функцию для модели GloVe можно записать в следующем в виде:

$$J = \sum_{i,j=1}^{V} f(X_{ik})(w_i^T \bar{w}_k + b_i + \bar{b}_k - \log(X_{ik})),$$

где функция f – выступает в качестве регуляризации и имеет вид:

$$f(x) = \left(\frac{x}{x_{max}}\right), \text{если } x < x_{max}, \text{иначе } f(x) = 1.$$

Классификация

Для сравнения моделей распределенного представления слов была выбрана задача мультиклассовой классификации. Классификатор построен на основе рекуррентной нейронной сети (Recurrent neural network; RNN), как было показано в работе [9] данный тип сетей даёт хороший результат в задачах мультиклассовой классификации текстов. В качестве датасета был выбран архив новостей Lenta.Ru-News-Dataset [10].

Входными данными для классификатора являются заголовки статей. Для входного предложения строится матрица размером N x M, где N – максимальное количество слов в предложении, а M – длина word embedding, i-ая строка данной матрицы, представляет собой i-ое слово в исходном предложении. Получившаяся матрица подаётся на вход классификатору. Задача классификатора сгенерировать правильную метку темы для входного заголовка.

Описание датасета

Датасет Lenta.Ru-News-Dataset имеет объем 1,8 гб, в котором более 800000 новостных заголовков и их кратких описаний, разбитых на 23 темы. Для классификации было выбрано 12 тем, так как остальные темы были представлены в малом объеме (менее 1000 примеров).

Параметры обучения моделей

Модели распределенного представления слов были обучены на всем исходном датасете. Параметры обучения моделей представлены в таблице 1.

Таблица 1. Параметры обучения моделей

Модель	***Важные параметры обучения***	***Размерность пространства***
Word2vec(cbow/skip-ngram)	*window = 5*	*100,200,300,500*
	min_count = 5	
Fasttext(cbow/skip-ngram)	*min_n = 3*	
	max_n = 6	
GloVe	*window = 10*	

Модели построения распределённого представления слов имеют большое число параметров обучения. В работе использовались рекомендованные параметры, указанные в соответствующих работах [4], [7], [8], описывающих данные модели. Наиболее важные параметры, указанные в таблице 1, имеют следующую расшифровку. Window – определяет количество слов, взятых слева и справа для контекста. Min_count – определяет минимальное количество вхождения слова в корпус. Min_n/max_n – минимальная/максимальная длина буквенных сочетаний (n-грамм).

Описание эксперимента

Для проведения эксперимента исходный датасет был разбит на 3 части: тренировочная, валидационная и тестовая; в соотношении 70%-20%-10% сохраняя пропорциональность классов. Тестирование модели проводилось в двух вариантах. Первый заключался в прогонке “чистого” тестового файла через обученную модель, т.е. слова в датасете не подвергались каким-либо трансформациям. Во втором варианте эксперимента в исходную тестовую

выборку был добавлен шум, в качестве которого выступают орфографические ошибки и опечатки.

Для генерации шума использовался датасет “Датасет: орфографические ошибки и опечатки” [11], который содержит 22 тыс. слов русского языка и варианты их неправильного написания. Итоговый тестовый набор содержал ~ 30% слов с опечатками.

Анализ результата

Качество модели оценивалось по Accuracy – метрики на тестовом наборе. Результаты моделей классификатора представлены в таблице 2. Тестовые наборы T1 и T2 – это тестовые наборы без опечаток и с опечатками. Жирным выделен лучший результат модели на наборе T1, красным цветом выделен лучший результат модели на наборе T2.

Таблица 2. Таблица результатов классификации с различными word embeddings

Размерность пространства	*100*		*200*		*300*		*500*	
Набор / *Метод*	*T1, acc*	*T2, acc*	*T1, acc*	*T2, acc*	*T1, acc*	*T2, acc*	*T1, acc*	*T2, acc*
Word2vec								
Cbow	*0.79*	*0.73*	*0.80*	***0.74***	***0.81***	***0.74***	***0.81***	*0.72*
skip-ngarm	*0.81*	*0.76*	*0.80*	***0.77***	***0.83***	*0.76*	***0.83***	*0.76*
Fasttext								
Cbow	*0.78*	*0.77*	*0.81*	***0.79***	***0.82***	***0.79***	***0.82***	*0.78*
skip-ngarm	*0.82*	*0.79*	*0.83*	***0.81***	***0.84***	*0.80*	***0.84***	*0.80*
GloVe								
-	*0.72*	*0.69*	*0.74*	*0.71*	*0.77*	*0.70*	***0.79***	***0.72***

Исходя из данных, представленных в таблице 2, можно сделать следующие выводы:

1. Метод skip-ngram более предпочтителен для классификации коротких текстов, чем модель cbow.
2. На “чистом” тестовом наборе, классификаторы на основе fasttext и word2vec выдают близкую точность (разница 0.01 в среднем).
3. При добавлении шума в тестовый набор, классификатор на основе fasttext показывает лучший результат за счет построения вектора по n-грамм для ранее не встречаемого слова.
4. Модель GloVe хотя и применима к задаче классификации, показывает заметно хуже результат, чем модели word2vec и fasttext.

Заключение

Современные подходы построения векторного пространства естественного языка ушли далеко вперед от простого унитарного кодирования. Современные модели учитывают совместную встречаемость слов в корпусе, моделирую тем самым семантические связи естественного языка. Среди рассмотренных моделей, особенно хотелось бы выделить модель fasttext, которая является логическим продолжением модели word2vec приспособленной для работы с морфологически богатыми языками (русский, финский, турецкий) или текстовыми опечатками, что позволяет обучать модель на корпусах меньшей размерности, требуемых для моделей word2vec/GloVe, с сохранением тех же результатов и иногда улучшением их.

СПИСОК ЛИТЕРАТУРЫ

1. Diksha Khurana, Aditya Koli. Natural Language Processing: State of The Art, Current Trends and Challenges //arXiv 2017. [Электронный ресурс. https://arxiv.org/abs/1708.05148 (дата обращения: 05.05.2019).
2. Николенко С., Кадурин А., Архангельская Е. Глубокое обучение – СПб.: Питер, 2018 – 480 с.

3. Yoshua Bengio, Réjean Ducharme, Pascal Vincent A Neural Probabilistic Language Model. Journal of Machine Learning Research
4. Tomas Mikolov, Ilya Sutskever, Kai Chen Distributed Representations of Words and Phrases and their Compositionality.
5. Xilinx. "HDL Synthesis for FPGAs Design Guide". section 3.13: "Encoding State Machines". Appendix A: "Accelerate FPGA Macros with One-Hot Approach". 1995.
6. Yoav Goldberg, Omer Levy Word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method // arXiv,2014. [Электронный ресурс. https://arxiv.org/abs/1402.3722] (дата обращения: 05.05.2019).
7. Piotr Bojanowski, Edouard Grave etc. Enriching Word Vectors with Subword Information // arXiv ,2017. https://arxiv.org/pdf/1607.04606] (дата обращения: 05.05.2019).
8. Jeffrey Pennington, Richard Socher etc. GloVe: Global Vectors for Word Representation //nlp.stanford.edu. [Электронный ресурс. https://nlp.stanford.edu/pubs] (дата обращения: 05.05.2019).
9. Pengfei Liu Xipeng Qiu↽ Xuanjing Huang. Recurrent Neural Network for Text Classification with Multi-Task Learning // arXiv,2016. [Электронный ресурс. https://arxiv.org/abs/1605.05101] (дата обращения: 05.05.2019).
10. Датасет Lenta.Ru-News-Dataset. //GitHub. [Электронный ресурс. https://github.com/yutkin/Lenta.Ru-News-Dataset] (дата обращения: 05.05.2019).
11. Датасет “Датасет: орфографические ошибки и опечатки”. //GitHub. [Электронный ресурс. https://github.com/dkulagin/kartaslov/tree/master/dataset/orfo_and_typos] (дата обращения: 05.05.2019).