МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное образовательное
учреждение высшего образования
ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Институт математики, механики и компьютерных наук
имени И. И. Воровича

Направление подготовки
Прикладная математика и информатика

Кафедра информатики и вычислительного эксперимента

АВТОМАТИЧЕСКОЕ ФОРМИРОВАНИЕ ЗАДАНИЙ ПО ГРАММАТИКЕ АНГЛИЙСКОГО ЯЗЫКА

Выпускная квалификационная работа
на степень бакалавра

Студента 4 курса
А. А. Мезги

Научный руководитель:
кандидат физико-математических наук, доцент В. А. Нестеренко

Ростов-на-Дону
2020

Содержание

Введение 3

1. Теоретические основы задачи обработки естественного языка 5
   1.1. Word Embeddings 6
      1.1.1. One-hot encoding 6
      1.1.2. Bag of Words 7
      1.1.3. Word2vec 10

Список литературы 14

Введение

Принято считать, что история развития обработки естественного языка берёт своё начало в 1950-х годах, когда Алан Тьюринг опубликовал свою работу «Вычислительные машины и разум» (англ. Computing Machinery and Intelligence), где был представлен тест Тьюринга[].

В настоящий момент в рамках обработки естественного языка (англ. Natural Language Processing), далее NLP, стоящей на пересечении таких компьютерных наук, как машинное обучение и компьютерная лингвистика, решаются проблемы анализа, понимания и извлечения смысла из естественной человеческой речи.

К наиболее актуальным задачам данной области можно отнести автоматическую суммаризация текстов, машинный перевод, распознавание именованных объектов, извлечение отношений, анализ настроений, распознавание речи и т.д.

Анализируя рынок онлайн-образования, можно отметить, что изучение иностранных языков является приоритетным направлением для потребителей. Создатели онлайн-платформ часто называют персонализацию обучения как отличительную особенность их обучения(?).

Рассмотрим основные этапы изучения грамматических явлений, встречающихся при знакомстве с новым языком:

- Presentation
  Демонстрация примеров и структуры грамматической конструкции.

- Pattern Recognition
  Самостоятельный поиск грамматических конструкций в исходном тексте.

- Controlled Practice
  Построение изучаемых грамматических конструкций в ограниченной форме.
- Semi-Controlled Practice
  Персонализация языка с упором на изучаемые грамматические явления.
- Free Practice
  Свободное использование языка.

Однако несмотря на большую гибкость по сравнению с классическим методом изучения иностранного языка в классах, некоторые материалы могут также быть устаревшими и неактуальными. Это связано с тем, что процесс формирования упражнений является время-затратным ввиду того, что задействует человеческий труд.

В рамках данной работы будет рассмотрена задача проектирования и реализации системы, позволяющей автоматизировать процесс создания заданий видов Controlled Practice и Semi-Controlled Practice, что позволит сократить время подготовки методических материалов и даст возможность большей персонализации траектории обучения.

1. Теоретические основы задачи обработки естественного языка

Задача обработки естественного языка это целый ряд теоретико-мотивированных вычислительных методов, которые позволяют производить анализ человеческого языка. В рамках NLP решается большое количество задач, затрагивающих различные уровни- начиная от определения частей речи (англ. Parts Of Speech), далее POS, заканчивая созданием диалоговых систем.

Долгое время в задачах NLP применялись модели поверхностного обучения, такие как SVM (англ. support vector machine), обученные на разреженных данных, представленных в пространстве высокой размерности, что не позволяло достичь достаточной точности и требовало значительных вычислительных ресурсов. Однако в последние годы были разработаны новые методы векторного представления слов, что позволило избежать экспоненциального роста размерности. Это помогло сократить время обучения сетей и перейти к методам, основанным на глубоком обучении, которые обеспечивают автоматическое обучение признакам. Это существенно отличает новые архитектуры от тех, что применялись раньше, так как для них больше не требуется ручное конструирование признаков.

Одна из первых архитектур глубокого обучения в области NLP была продемонстрирована в статье “Natural Language Processing (Almost) from Scratch” [] Ронаном Коллобертом и Джейсоном Уэстоном. На тот момент такая архитектура превосходила большую часть уже существующих моделей в поиска именованных сущностей (англ. named-entity recognition, NER), семантической маркировке ролей (англ. semantic role labeling, SRL), и определении частей речи (POS). С того времени появилось большое количество алгоритмов и моделей, решающих сложные задачи обработки естественного языка.

1.1. Word Embeddings

Статистические модели стали основным инструментом в задачах NLP, однако в начале они страдали от т.н. проклятия размерности[]. Это повлекло развитие алгоритмов, которые позволили бы представлять слова в низкоразмерном пространстве[].

Стоит сказать, что на данный момент нет общепризнанного перевода термина embedding (от англ. ‘вложение’), поэтому будет использоваться англицизм.

Embedding представляет собой преобразование некой сущности в числовой вектор. Далее будут рассмотрены основные подходы, применяющиеся для решения данной задачи.

1.1.1. One-hot encoding

Одним из первых решений задачи векторного представления слов был так называемый унитарный код. Он представлял собой вектор длины n, которая определяется количеством слов некоторого словаря, содержащий (n-1) нулей и 1 единицу. Индекс значащей единицы соответствовал расположению слова в данном словаре —см. таблицу 1.

Несмотря на то, что такая архитектура позволяет решить проблему кодирования слов, она обладает рядом существенных недостатков:

- При добавлении нового слова в середину существующего словаря есть необходимость заново проводить нумерацию его элементов.
- Происходит быстрый рост размерности представления текстов. Например для текста из 9 уникальных слов требуется матрица 9x9
- Данный метод не предоставляет информации о семантической близости слов. Данный аспект связан с тем, что написание слов

Таблица 1 — One-Hot Encoded векторы

Vocabulary	1	2	3	4	5	6	7	8	9	10
bag	1	0	0	0	0	0	0	0	0	0
words	0	1	0	0	0	0	0	0	0	0
model	0	0	1	0	0	0	0	0	0	0
way	0	0	0	1	0	0	0	0	0	0
representing	0	0	0	0	1	0	0	0	0	0
text	0	0	0	0	0	1	0	0	0	0
data	0	0	0	0	0	0	1	0	0	0
simple	0	0	0	0	0	0	0	1	0	0
understand	0	0	0	0	0	0	0	0	1	0
implement	0	0	0	0	0	0	0	0	0	1

не имеет непосредственной связи с объектами, которые они описывают [2.12. Фердинанд де Соссюр (1827–1913). Лингвистический структурализм].

1.1.2. Bag of Words

На основе кодирования слов унитарным кодом, рассмотренным в пункте 1.1.1, был предложен более экономичный вариант представления текстов, называемый “мешком слов” (от англ. Bag of words).

Алгоритм построения такого представления содержит следующие основные этапы:

- Предварительное создание словаря методом OHE.
- Кодирование слов, содержащихся в тексте.
- Сложение всех полученных one-hot векторов.

На выходе получим числовой вектор, который описывает информацию о количестве различных слов в исходном тексте. Такая модель не сохраняет структуру входных данных, в связи с чем теряется информация о взаимном расположении слов. Тем не менее, применяя

Таблица 2 — Терм-документная таблица

Vocabulary	Documents	
	C1	C2
bag	1	1
words	1	1
model	1	1
way	1	0
representing	1	0
text	1	0
data	1	0
simple	0	1
understand	0	1
implement	0	1

данный подход можно сравнивать тексты путем сравнения выходных векторов. Примером такой метрики является косинусная мера:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}, \tag{1}$$

A_i, B_i – компоненты векторов $\mathbf{A}$ и $\mathbf{B}$ соответственно.

Рассмотрим этот процесс имплементации алгоритма.

1. Зададим экземпляры текста.
   C1 = *“The Bag of Words model is a way of representing text data.”*
   C2 = *“The Bag of Words model is simple to understand and implement.”*

2. Очистим тексты от пунктуационных символов и ‘стоп-слов’[1], которые не несут смысловой нагрузки.

3. Сформируем словарь и закодируем его методом OHE (см. таблицу 1).

[1] the, of, is, a, to, and

4. Для каждого предложения сложим One-Hot векторы слов, которые входят в их составы (см. таблицу 2).

5. Применим косинусную меру (1) к полученным векторам, которые количественно описывают исходные тексты.

$$\text{similarity} = \cos(\theta) = \frac{3}{\sqrt{7} \cdot \sqrt{6}} = 0.46291$$

Кроме того, полученную Таблицу 2 с зависимостью “слово-документ” можно представить в виде произведения матриц “слово-тема” и “тема-документ”. Для этого можно использовать SVD-разложение:

$$(\mathbf{t}_i^T) \rightarrow \begin{matrix} & (\mathbf{d}_j) \\ & \downarrow \end{matrix} \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \\ \mathbf{u}_1 \\ \\ \end{bmatrix} & \dots & \begin{bmatrix} \\ \mathbf{u}_l \\ \\ \end{bmatrix} \end{bmatrix} \cdot \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{bmatrix} \cdot \begin{bmatrix} \begin{bmatrix} & \mathbf{v}_1 & \end{bmatrix} \\ \vdots \\ \begin{bmatrix} & \mathbf{v}_l & \end{bmatrix} \end{bmatrix} \tag{2}$$

$\mathbf{t}_i$ – слово, $\mathbf{d}_i$ – документ.

Применим такое разложение для текстов из примера и визуализируем (см. рис. 1). Как можно видеть, получаемые результаты имеют сильную зависимость от корпуса, к которому применяется разложение.

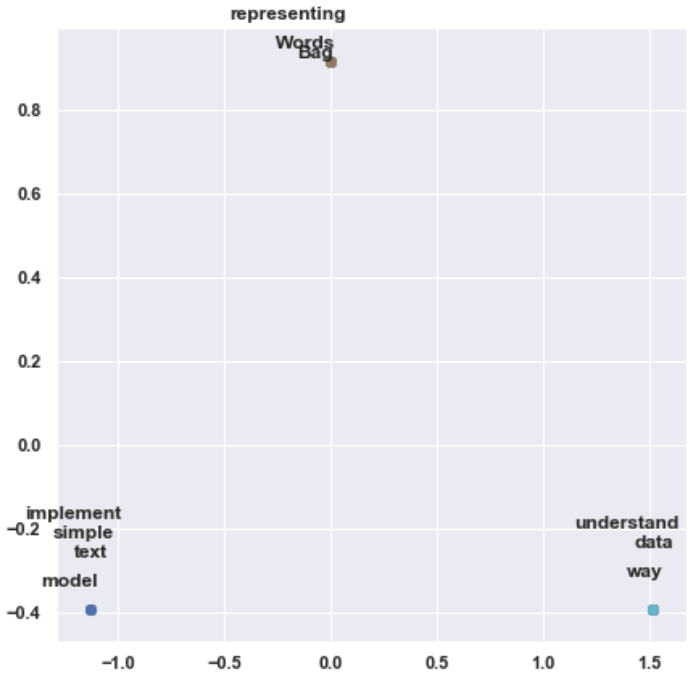

Рисунок 1 — SVD разложение для C1 и C2

1.1.3. Word2vec

Из-за ряда недостатков, которыми обладали традиционные алгоритмы, исследования в области представлений слов продолжились. В 2013 году Томаш Миколов представил[] подход к кодированию слов, который решал целый ряд проблем, присущих другим моделям, в том числе рост размерности векторного пространства и невозможность сохранять семантическую близость.

Им было представлено 2 подхода, которые он назвал continuous bag-of-words (CBOW) и skip-gram. Они позволяют строить высококачественные распределенные векторные представлений.

Между представленными моделями есть существенное отличие. CBOW вычисляет условную вероятность появления целевого слова исходя из его контекста, который лежит в окне размера k. В свою очередь Skip-gram предсказывает слова контекста по центральному сло-

ву. Предполагается, что контекстные слова расположены симметрично целевым словам на расстоянии, равном размеру окна в обоих направлениях. Схематичное представление данных методов показано на рис. 2

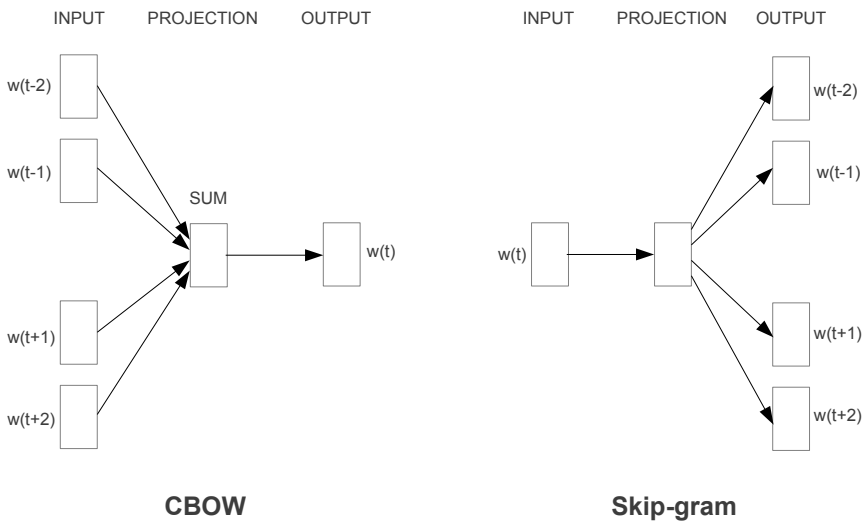

Рисунок 2 — Модели CBOW и Skip-gram (Источник рисунка: Mikolov[])

Continuous Bag-of-words Рассмотрим упрощенную модель CBOW с контекстом из одного слова более детально (см. рис. 3). Она представляет собой полносвязную нейронную сеть со скрытым слоем. Входной слой, принимающий one-hot вектор, состоит из V нейронов. Скрытый слой в свою очередь имеет N нейронов. На выходном слое применяется операция Softmax (4), давая на выходе распределение вероятностей по всем словам в словаре. Слои связаны матрицами весов $\mathbf{W} \in \mathcal{R}^{V \times N}$ и $\mathbf{W}^{'} \in \mathcal{R}^{H \times V}$ соответственно.

Каждое слово из словаря в конечном итоге представляется в виде двух выученных векторов $\mathbf{v}_c$ и $\mathbf{v}_w$, которые соответствуют контекстному и целевому слову соответственно. Таким образом, k-е слово в словаре будет представлено следующим образом:

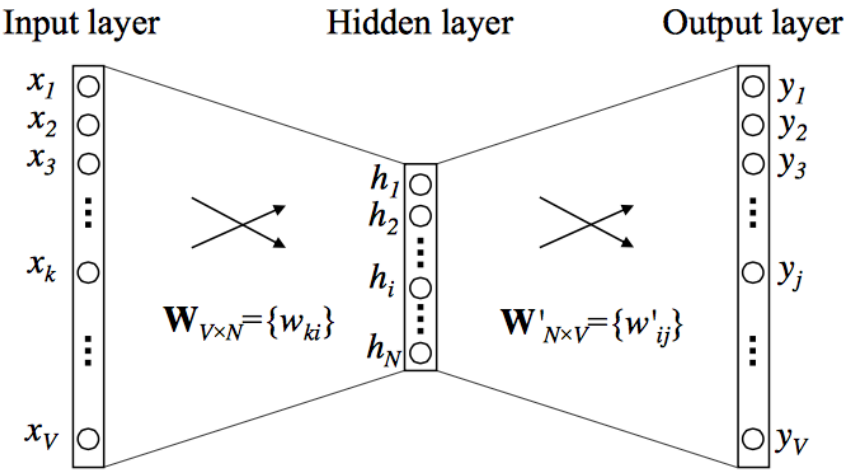

Рисунок 3 — Модель CBOW (Источник рисунка: Rong [1])

$$\mathbf{v}_c = \mathbf{W}_{(k,.)} \qquad \mathbf{v}_w = \mathbf{W}^{'}_{(.,k)} \tag{3}$$

Тогда для любого слова w_i с контекстным словом c в качестве входных данных получим:

$$p\left(w_i|c\right) = \mathbf{y}_i = \frac{e^{u_i}}{\sum_{i=1}^{V} e^{u_i}} \quad , \quad u_i = \mathbf{v}^{\mathbf{T}}_{\mathbf{w_i}}.\mathbf{v_c} \tag{4}$$

Параметры $\theta = \{\mathbf{v_w}, \mathbf{v_c}\}_{w,c\in \text{Vocab}}$ обучаются путем оптимизации целевой функции, в качестве которой выступает логарифмическая функция правдоподобия:

$$\mathbf{l}(\theta) = \sum_{\mathbf{w}\in\mathbf{Vocab}} \mathbf{log}\left(\mathbf{p}\left(\mathbf{w}|\mathbf{c}\right)\right) \tag{5}$$

$$\frac{\partial \mathbf{l}(\theta)}{\partial \mathbf{v_w}} = \mathbf{v_c}\left(\mathbf{1} - \mathbf{p}\left(\mathbf{wc}\right)\right) \tag{6}$$

В общем случае модели CBOW (см. рис. 4) все One-hot векторы берутся одновременно в качестве входных данных:

$$\mathbf{h} = \mathbf{W}^{\mathbf{T}}(\mathbf{x_1} + \mathbf{x_2} + ... + \mathbf{x_c}) \tag{7}$$

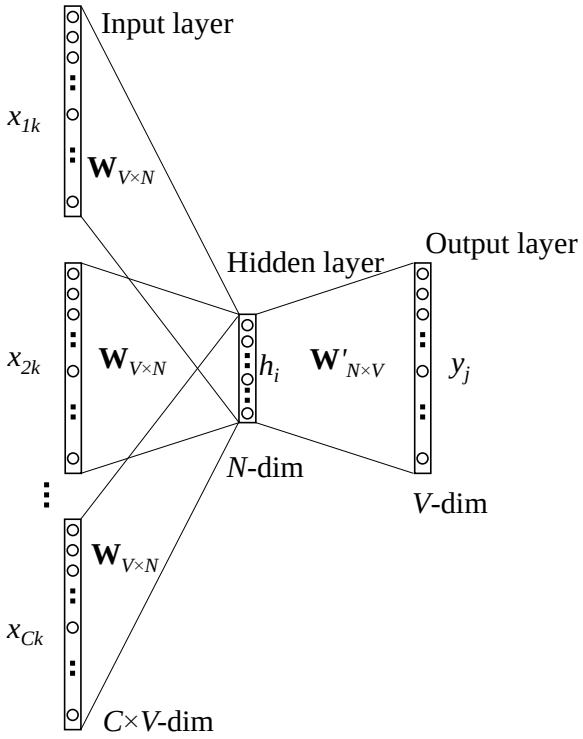

Рисунок 4 — Модель CBOW в общем случае

Несмотря на то, что в такой архитектуре не учитывается порядок слов, так как входные векторы суммируются, она сохраняет некоторые семантические характеристики корпусов текста, на которых проводится обучение. В связи с этим схожие по смыслу слова имеют схожие векторные представления. Наиболее популярный пример сохранения семантической близости представлен на рис. 5.

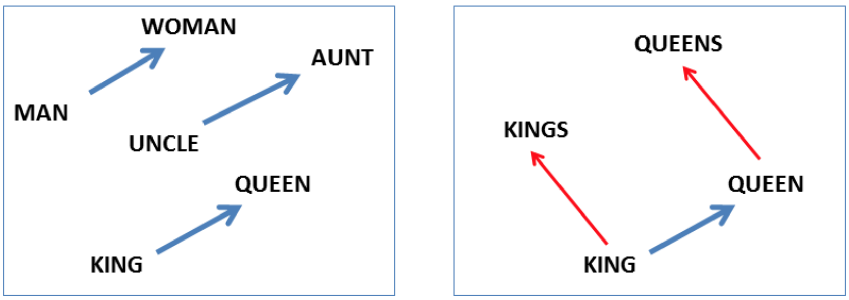

Рисунок 5 — Иллюстрация отношений векторного представления слов (Источник рисунка: Mikolov [2])

Список литературы

1. *Rong X.* word2vec Parameter Learning Explained // ArXiv. — 2014. — T. abs/1411.2738.

2. *Mikolov T.*, *Yih W.-t.*, *Zweig G.* Linguistic Regularities in Continuous Space Word Representations // Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. — Atlanta, Georgia : Association for Computational Linguistics, 06.2013. — С. 746—751. — URL: `https://www.aclweb.org/anthology/N13-1090`.