АЛГОРИТМ МОРФЕМНОГО РАЗБОРА СЛОВ НА ОСНОВЕ НАИВНОГО БАЙЕСОВСКОГО ПРЕДПОЛОЖЕНИЯ И ИСПОЛЬЗОВАНИЕ ЕГО ДЛЯ ПОСТРОЕНИЯ ВЕКТОРНОЙ МОДЕЛИ ТЕКСТА

Яковлева Ю.В, бакалавр кафедры информационной безопасности компьютерных систем Университета СПбПУ Петра Великого, juli-jakovleva@yandex.ru

Коваленко Т.В., магистрант кафедры информационно-управляющих систем Университета СПбПУ Петра Великого, tanyakovalenko1994@gmail.com

Никифоров И.В., к.т.н., доцент кафедры информационно-управляющих систем Университета СПбПУ Петра Великого, igor.nikiforovv@gmail.com

Галинский Р.Б., старший программист, ООО «БигДэйтаСолюшенз», galinskyifmo@gmail.com

Аннотация

В данной статье описан подход к улучшению группы алгоритмов word2vec с учетом морфемного анализа слов корпуса используемого языка.

Подход апробирован на ряде задач, представлена оценка его результативности.

Введение

Для решения задач, связанных с обработкой и анализом естественного языка, например, семантического анализа текста, анализа тональности, тематического анализа, и извлечения ключевых слов необходимо наличие модели анализируемого языка. Одним из подходов получения такой модели является векторное представление слов, входящих в его состав. Вектора строятся в соответствии с различными алгоритмами. Одним из популярных наборов таких алгоритмов является Word2Vec [1].

На вход алгоритма Word2Vec подается корпус языка (множество текстов). Результатом работы алгоритма является набор пар слово-вектор. Основной принцип построения векторов с помощью Word2Vec базируется на предположении, что слова, появляющиеся в похожих контекстах с похожей частотой, являются семантически близкими, а значит, в векторном пространстве всех слов будут находиться рядом. Более формально,

векторные представления слов с похожей семантикой являются косинусно-близкими [1], так как для определения близости рассчитывается косинус угла между векторами: чем он ближе к единице, тем ближе слова. Например, семантически близкими являются слова «кровать» и «подушка».

В морфологически богатых языках, например, в русском, каждое слово состоит из его основной, смысловой части (корня слова) и дополнительных морфем (приставок, суффиксов и окончаний). Каждая морфема, входящая в состав слова, несет определенный смысловой оттенок. Например, возьмем слово «бездельник». Это слово состоит из корня «дель», обозначающего «дело», приставки «без», обозначающей «отсутствие» и суффикса «ник», образующего существительное мужского рода со значением «тот, кто связан с указанной в основе деятельностью, профессией». Как видно из проведенного разбора, каждая морфема вносит определенный смысл в слово. Поэтому, улучшением подхода использования векторного представления слов может быть более глубокий морфемный анализ слов, входящих в корпус языка и корректировка вектора каждого слова на основании значений входящих в его состав морфем [2].

Цели и задачи работы

Целью работы является создание библиотеки, в основе которой лежит алгоритм морфемного разбиения слов и генерация на его основе векторной модели русскоязычного текста. Для достижения цели требуется решить следующие задачи:

1. составление словарей значений морфем русского языка;
2. разработка парсеров для составленных словарей;
3. разработка алгоритма морфемного разбора слов;
4. тестирование качества морфемного разбора слов;
5. разработка модуля пересчета векторной модели текста;
6. апробация полученной модели для решения задачи классификации слов по тематикам.

Реализация

Для реализации поставленной задачи были составлены словари, представленные в таблице 1.

№	**Тип словаря**	**Размер словаря**
1	Словарь значений *приставок*	88

2	Словарь значений *суффиксов*	293
3	Словарь *корней*	1195
4	Словарь готовых *морфемных разборов* слов	137827

Таблица 1: Список словарей

«Словарь значений приставок» (1) и «словарь значений суффиксов» (2) для каждой морфемы содержат список ее значений и соответствующие примеры. Словари были составлены вручную на основании источников [3, 4, 5, 6]. Для удобной манипуляции данными словарей были реализованы парсеры на языке python. Каждый отдельный парсер представляет из себя разрабатываемой библиотеки, который может использовать любой желающий, клонировав репозиторий библиотеки с Github [7] и импортировав необходимые модули в свой код. Словарь корней был составлен на основании сайта [8], а словарь готовых разборов слов – сайтов [9, 10], создатели которых за источник брали словари А.Н. Тихонова [11] и Т.Ф. Ефремовой [12].

При составлении словарей были исправлены различные имеющиеся в них ошибки и неточности, которых было более 100. Например, в одном из источников, слово «см`ётанный» авторами было разобрана как «смет`анный», а в слове «отчисленный» авторы забыли выделить окончание как отдельную морфему. В слове «укрепляю» авторы забыли про букву «у», а в слове «скрепка» – букву «с». Все исправления выполнялись вручную при многократном просматривании результатов разборов.

Алгоритм морфемного разбора

Используемые словари содержат не все возможные слова русского языка, и при этом только в начальной форме (инфинитив, единственное число, мужской род, именительный падеж), поэтому был разработан собственный алгоритм, осуществляющий морфемный разбор слов. В качестве первого варианта алгоритма разбиения слова по составу была использована следующая методика: в слове с помощью алгоритма поиска подстроки в строке ищутся все корни из словаря корней; слева от корней производится поиск всех приставок из словаря, а справа - суффиксов. Приоритет отдается наиболее длинным морфемам.

Среди общего количества слов из тестового набора только 9% такой алгоритм разбирает абсолютно так же, как и в словаре. Наблюдаемый малый процент точности связан со сложностью в выделении в словах суффиксов и

корней (в то время как в 90% случаев приставка определяется верно) и в неоднозначности разбора.

Для улучшения результата был использован алгоритм с использованием *наивного байесовского классификатора*, где классом выступало разбиение слова на морфемы, а признаками – сами морфемы. Иными словами, алгоритм оценивал правдоподобность разбиения через произведение вероятностей появление морфем из разбиения при условии, что морфемы не зависят друг от друга. Перебирая все возможные разбиения и оценивая правдоподобность каждого, алгоритм находит самое правдоподобное разбиение. Шаги алгоритма:

1. Разбить входное слово w на непересекающиеся подстроки x_1, x_2, …, x_n, где n = 1..|w и верно $w = x_1 + x_2 + \ldots + x_n$.
2. Для каждой подстроки найти наиболее вероятный тип морфемы m (корень, суффикс, окончание, приставка): посчитать сколько раз такая подстрока была корнем, суффиксом, окончанием и приставкой; найти наиболее вероятный тип морфемы $p(m|x_i) = p(m_i)$
3. Посчитать вероятность разбиения $p(w = x_1 + x_2 + \ldots + x_n) = \Pi p(m_i)$. Если вероятность текущего разбиения больше, чем любого другого, то сохранить разбиение в качестве результата алгоритма.
4. Если существует ещё одно разбиение слова w, отличное от предыдущих разбиений, то перейти к шагу 1)

В пункте 2 выбирается максимальный тип в силу того, что алгоритм максимизирует функцию правдоподобия $p(w = x_1 + x_2 + \ldots + x_n)$. Пункт 3 верен в силу предположения о независимости морфем между собой (наивное байесовское предположение).

Предположение о независимости отражает естественную структуру слова – к примеру, корень никак не зависит от приставки, а приставка от окончания или суффикса. Однако тот факт, что окончания могут появиться только в конце слова, а приставка в начале, натолкнуло на идею, что при оценке вероятности появления морфемы $p(m|x_i)$ стоит учитывать ещё и её удаленность от начала или конца слова – $p(m|x_i s)$, где s, к примеру, удаленность x_i от конца слова. Кроме того, некоторые приставки применим только тогда, когда корень начинается с какой-нибудь определенной буквы, что привело нас к использованию в качестве признака для оценки вероятности появления морфемы следующей за морфемой буквы. Таким

образом, для оценки вероятности появление морфемы использовались два набора признаков: 1) удаленность от конца слова, сама морфема и следующая за морфемой буква 2) удаленность от конца слова, сама морфемы, следующая за морфемой буква и предшествующая морфеме буква.

Вероятности обучались из исходных словарей, оценка качества происходила на этих же словах. Точность алгоритма при использовании 3 признаков на наборе из 10 тысяч слов составила 81.2% - это точность полного совпадения разбора; при использовании 4 признаков на том же наборе точность разбора составила 89.5%. Во втором случае результат ожидаемо оказался лучше, так как использование большего количества признаков более точно моделирует распределение $p(m_i)$. Сходимость точности алгоритма показана на рис. 1.

Некоторые ошибки разбора оказались очень интересными. Было замечено, что некоторые слова разбирались с помощью алгоритма *лучше*, чем в словаре, так как, скорее всего, в словаре были допущены ошибки. К примеру, слово «скоростном», в исходных словарях, имеет следующий разбор: с – приставка, корост – корень, н - суффикс, ом – суффикс. Разработанный алгоритм предложил следующий разбор: скор – корень, ост – суффикс, н – суффикс и ом – суффикс. Не трудно заметить, что оригинальный разбор не верный, потому что корня «корост» не существует, а предложенный алгоритм определил корень верно.

Полученный алгоритм может быть применен для слов, для которых морфемный разбор не зафиксирован в словарях. Однако у алгоритма есть недостаток: если предложенное для разбора слово не имеет однокоренных слов в исходной обучающей выборке, то алгоритм может отказаться его разбивать, не предложив ни одного варианта разбора.

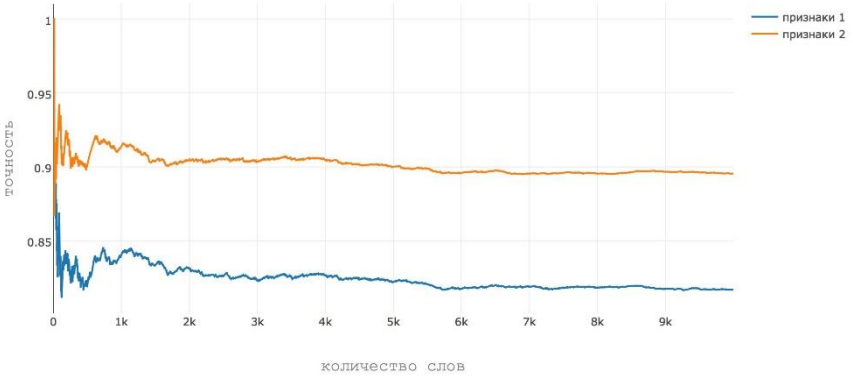

Рисунок 1. Точность алгоритма разбиения слова на морфемы на тестовом наборе из 10 тыс. слов. «признаки 1» соответствует набору из 3 признаков – морфема, предшествующая морфеме буква, удаленность от конца; «признаки 2» - набору из 4 признаков – морфема, предшествующая морфеме буква, последующая за морфемой буква, удаленность от конца.

Применение алгоритма морфемного разбора

На сегодняшний день общедоступных алгоритмов морфемного разбора слов русского языка не существует. Поэтому, отличительной особенностью и новизной данной работы является создание общедоступной библиотеки построения векторной модели текста русского языка на основе морфемного разбора слов.

Был реализован модуль на языке python, осуществляющий формирование новой модели для некоторого заранее выбранного корпуса языка. Скрипт осуществляет загрузку модели, полученной после обработки некоторого текста алгоритмами Word2Vec, и пересчет векторов для всех слов в модели. Готовые модели для проверки были взяты из источника RusVectōrēs [13]. Для каждой отдельно взятой модели размерность векторов и количество слов разное.

Пересчет векторов осуществляется по следующей формуле:

$$v'_{t_i} = \frac{1}{2}(v_{t_i} + \sum_{w \in M_i} sim(v_{t_i}, v_w) * v_w),$$

где t_i – i-тое слово в словаре в корпусе;

v'_{t_i} – новое векторное представление i-ого слова в корпусе;

v_{t_i} – векторное представление i-ого слова в корпусе;

w – морфема i-ого слова в корпусе;

M_i – множество всех морфем i-ого слова в корпусе;

$sim(v_{t_i}, v_w)$ – косинусная близость между i-ым словом и морфемой (лежит в пределах от 0 до 1).

Как видно из представленной формулы, каждый вектор слова корректируется в зависимости от векторов значений входящих в его состав морфем. Например, для слова «бездельник», значениями морфем являются слова «отсутствие» (значение приставки), «дело» (значение корня) и «человек» (значение суффикса). Окончание в строимой модели не учитываются из-за отсутствие значимого вклада в смысл слова. В нашем алгоритме отдельным параметром можно задать ожидаемую косинусную близость между словом и значениями входящих в его состав морфем. В случае, если косинусная близость между значением некоторой морфемы и словом меньше этого коэффициента, в итоговую формулу для пересчета модели попадало само слово, а не значение ее морфемы. Коэффициент ½ выбран из-за того, что новый вектор слова формируется как среднее значение между самим вектором слова и суммой векторов значений морфем.

Полученная векторная модель языка была протестирована на задаче классификации слов по тематикам. На рис. 2 изображено разбиение на три кластера слов из таких областей, как финансовая сфера (зеленый цвет), медицинские термины (красный цвет) и автомобильные термины (синий цвет). Для того, чтобы визуализировать кластеры, был использован алгоритм t-SNE [14], осуществляющий понижение размерности векторов. На рис. 3 представлена аналогичная кластеризация, но за модель языка была взята модель, полученная с помощью Word2Vec.

Сравнить результаты можно при помощи визуального метода, а также сравнивая значения определителей матриц ковариации (обобщенных определителей), для модели гауссовых смесей, построенной для каждой из областей слов. Определитель матрицы ковариаций определяет степень случайного разброса элементов системы [15]. Для новой модели значения определителей равны 299760.70, 798422.11, 569422.18 для каждой из тематик, соответственно. Для Word2Vec модели – 775773.20, 722424.83, 585058,09. Для двух из трех областей слов, определители матриц ковариации оказались меньше, а для одной – почти такой же, как и оригинальный, что означает, что новая векторная модель лучше подходит

для использования в решении задачи классификации, чем модель Word2Vec, так как близкие по смыслу слова оказываются более плотно сгруппированными.

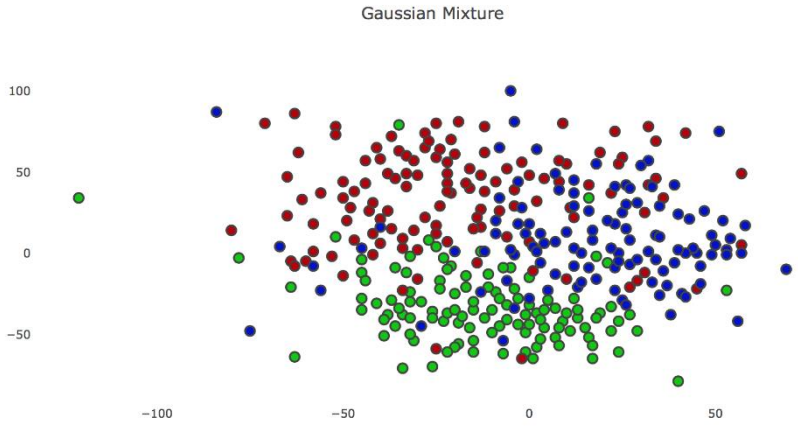

Рисунок 2. Кластеры слов из области финансов, медицины и автомобилей, полученные с помощью векторной модели языка на основе морфемного разбиения слов.

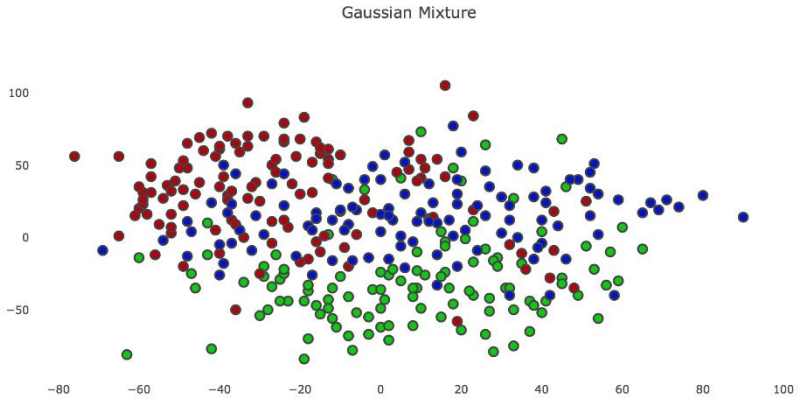

Рисунок 3. Кластеры слов из областей финансов, медицины и автомобилей, полученные с помощью word2vec модели языка.

Заключение

Результатами проделанной работы являются:

1. составленные морфемные словари со значениями и примерами;
2. python-модули для удобной работы с данными словарей;
3. алгоритм разбиения слов на морфемы с точностью 81.2%
4. модуль формирования новой векторной модели языка, протестированный на размеченных данных

Все результаты находятся в свободном доступе в репозитории Github [7].

Будущие планы. Планируется улучшать точность словарей, добавляя новые морфемы и исправляя имеющиеся ошибки. Требуется улучшить модуль разбиения слов на морфемы с помощью добавление новых признаков. Протестировать формирование новой векторной модели для большего количества тем.

Литература

1. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. // Proceedings of Workshop at ICLR, 2013
2. Xu Y., Liu J. Implicitly Incorporating Morphological Information into Word Embedding //arXiv preprint arXiv:1701.02481. – 2017.
3. Русский древослов. Историко-словообразовалельный словарь русского языка. Словарь морфем. [Электронный ресурс]. 2016. Дата обращения 20.02.2017. http://www.drevoslov.ru/wordcreation/morphem/
4. Учимся припеваючи. Значение приставок. [Электронный ресурс]. 2009-2016. Дата обращения 20.02.2017. http://spelling.siteedit.ru/page51/
5. Учимся припеваючи. Значение суффиксов. [Электронный ресурс]. 2009-2016. Дата обращения 20.02.2017. http://spelling.siteedit.ru/page50/
6. Значение латинских морфем. [Электронный ресурс]. 2016. Дата обращения 20.02.2017. https://grammatika-rus.ru/znachenie-latinskih-morfem

7. GitHub репозиторий. [Электронный ресурс]. 2017. Дата обращения 20.02.2017. https://github.com/TanyaKovalenko/Morpheme
8. Славянские корни в русском языке. [Электронный ресурс]. 2012. Дата обращения 20.02.2017. http://www.slovorod.ru/slavic-roots/
9. Разбор слов по составу. [Электронный ресурс]. 2012. Дата обращения 20.02.2017. http://sostavslova.ru/
10. Разбор слов по составу. [Электронный ресурс]. Дата обращения 20.02.2017. http://morphemeonline.ru/
11. А. Н. Тихонов. Морфемно-орфографический словарь русского языка, — М.: АСТ: Астрель, 2002. — 704 с.
12. Кузнецова А. И., Ефремова Т. Ф. Словарь морфем русского языка, — М.: Рус. яз., 1986. — 1132 с.
13. RusVectōrēs: дистрибутивные семантические модели для русского языка. [Электронный ресурс]. 2012. Дата обращения 20.02.2017. http://rusvectores.org/ru/
14. Van Der Maaten L., Hinton G. Visualizing high-dimensional data using t-sne. journal of machine learning research //J Mach Learn Res. – 2008. – Т. 9. – С. 26.
15. Айвазян С. А. и др. Прикладная статистика: Основы моделирования и первичная обработка данных. Справочное изд. / С. А. Айвазян, И. С. Енюков, Л. Д. Мешалкин. — М.: Финансы и статистика, 1983. — 471 с.