СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ WORD2VEC И GLOVE И НЕКОТОРЫЕ ВЫВОДЫ ИЗ НЕГО

COMPARATIVE ANALYSIS OF THE METHODS AND WORD2VEC GLOVE AND SOME CONCLUSIONS DRAWN FROM IT

M. Zakirov

Summary. In this paper, we introduce the notion of lexical parameterization with the "zero" and a comparatively considered the two most common methods word2vec [3] and GloVe. [4] In the process of incarnation in the form of practical recommendations were obtained data practices programs, based on pilot experience, backed up by theoretical calculations. Also represented in the statistics and conclusions of the author could explain some obscure properties primarily related to methods of their stability and convergence. This work is a review, the introductory part contains everything you need for the introduction into the problems.

Keywords. pointwise mutual information, shifted pointwise mutual information negative sampling

Закиров Марат Энварович
Старший разработчик в поисковом портале
ООО «Спутник» marat61@gmail.com

Аннотация. В данной работе вводится понятие лексической параметризации с "нуля" и сравнительно рассматриваются два наиболее распространенных методов word2vec [3] и GloVe.[4] В процессе воплощения в виде программ данных методов были получены практические рекомендации, на основе экспериментального опыта, подкрепленные теоретическими выкладками. Также в работе представлена статистика и выводы автора могущие объяснить некоторые неясные прежде свойства методов касающиеся их устойчивости и сходимости. Данная работа носит обзорный характер, вводная часть содержит все необходимое для введения в проблематику.

Ключевые слова: взаимная информация, сдвинутая взаимная информация, негативное семплирование.

Введение

Лексическая параметризация — это направление в вычислительной лингвистике, в рамках которого лексике ставится в соответствие набор параметров, несущий в себе "семантику". Таким образом, что одинаковым в некотором смысле словам, ставится в соответствие похожие (в соответствии с некоторой метрикой) наборы параметров. А поскольку параметры несут семантику, возможно манипулирование непосредственно параметрами, минуя отсылку к тексту как таковому.

Гипотеза о статистической аналогии

Впервые показанная в работе [1], интересная "аномалия" привела к открытию понятия статистическая аналогии. Было найдено, что простая арифметическая разность и сумма трех словарных векторов приводит к нахождению словарного вектора, наиболее близким к которому (например, по косинусной мере) оказывается словарный вектор слова аналога. К примеру, так:

$$v_{король} - v_{мужчина} + v_{женщина} \approx v_{королева} \qquad (1)$$

Причем сам алгоритм [1] находил, не только семантические, но и синтаксические аналоги, не делая разницы между ними. Отсюда ими была выведена следующая гипотеза-эвристика для аналогий:

$$\frac{p(w|a)}{p(w|b)} \approx \frac{p(w|c)}{p(w|d)} \qquad (2)$$

Данная формула утверждает, что для слов , верно, соотношение условных вероятностей двух пар слов (a,b) и (c,d) в целом более или менее одинаково (2) для всех произвольных слов w, из словаря то такие пары считаются аналогиями.

Формула (2) может быть переформулирована, используя центральное понятие данной статьи,— понятие о взаимной вероятности слов, по-английски *pointwise mutual information* [3]. Данная структура представляет собой симметричную матрицу с нулевой главной диагональю (значения на главной диагонали нас вообще не будут интересовать), элементы, которые определяются как:

$$PMI(a,b) = \log\left(\frac{p(a,b)}{p(a)p(b)}\right) \qquad (3)$$

Где $p(a,b)$ вероятность встречи слов a и b в одном контексте[1], $p(a)$ и $p(b)$ вероятности слов a и b соответственно. В некоторых работах (3) прибегают к несколько иной

[1] Под одним контекстом понимается, словарная близость слов в тексте, определяемая некоторым радиусом.

формулировке, в которой все отрицательные значения заменены на нулевые, утверждая при этом получение лучшего качества, однако оставляя это без теоретических объяснений, тем не менее, в этом есть определенный смысл, который станет понятен ниже.

$$SPMI(a,b) = PMI(a,b) if PMI(a,b) > 0 \; else 0 \quad (4)$$

Называется эта характеристика сдвинутая взаимная вероятность, по англ. *shifted pointwise mutual information*. Поскольку матрица $S_{n\times n}$ [2] сформированная из этих значений, как и матрица P состоящая из значений взаимных вероятностей слов, является симметричный и имеет только положительные значения, то такая матрица может быть сколь угодно идеально разложена с использованием всего одной матрицы. В литературе эта трансформация получила название разложение холецкого [6]:

$$P_{n\times n} \approx W_{n\times m} W^T_{n\times m} \quad (5)$$

Где матрица $W_{n\times m}$ где каждому слову соответствует вектор V_m — размер вектора параметров для каждого слова из словаря D. Как видно из приведенной формулы (5) для разложения матрицы $P_{n\times n}$ используется всего лишь одна матрица $W_{n\times m}$. Причем чем больше размерность m тем лучше соответствие воспроизведенной матрице своему оригиналу $P_{n\times n}$. С точки зрения линейной алгебры матрица $P_{n\times n}$ как уже было сказано должна обладать такими свойствами как симметричность и положительная полуопределенность, но на практике даже разложение, "невозможное" с точки зрения теории, дает неплохие результаты по качеству, а потому разложение одной матрицей может применятся даже в ситуация когда $P_{n\times n}$ не является положительно полуопределенной. По всей видимости, у сдвинутой матрицы P больше шансов на то, чтобы быть положительно полуопределенной, отсюда некоторый рост качества.

Теперь переформулируем формулу (2) в терминах взаимной информации двух слов:

$$\frac{\frac{p(w,a)}{p(w)p(a)}}{\frac{p(w,b)}{p(w)p(b)}} \approx \frac{\frac{p(w,c)}{p(w)p(c)}}{\frac{p(w,d)}{p(w)p(d)}} \Rightarrow \frac{PMI(w,a)}{PMI(w,b)} \approx \frac{PMI(w,c)}{PMI(w,d)} \quad (6)$$

Имея матрицу взаимной информации, и используй формулу (6) мы можем проверять гипотезу об аналогичности пар слов[3]. Поскольку соотношение (6) далеко не для всех слов соблюдается одинаково, то необходимо сформулировать степень аналогичности пар, например через квадратичную ошибку, чем ошибка меньше, тем пары в большей степени аналогичны.

$$E_{(a,b)(c,d)} = \sum_{\forall w\in D} \left[\frac{PMI(w,a)}{PMI(w,b)} - \frac{PMI(w,c)}{PMI(w,d)} \right]^2 \quad (7)$$

Где $E_{(a,b)(c,d)}$ есть мера аналогичности для каждой возможной пары пар слов *(a,b)* и *(c,d)* а w — произвольное слово из словаря. Теперь может быть сформулирована задача поиска неизвестного слова одной из пар аналогий. Предположим, есть две пары слов *(a,b)* и *(c,x)*, где x — неизвестное слово. Это задача поиска слова x такого, что:

$$argmin_{x\in D} \sum_{\forall w\in D} \left[\frac{PMI(w,a)}{PMI(w,b)} - \frac{PMI(w,c)}{PMI(w,d)} \right]^2 \quad (8)$$

Эффект семантического сглаживания

На практике напрямую, без параметризации не получается вычислять синонимичность слов и аналогии из-за того, что хоть слова и синонимичны статистики их словоупотребления отличаются даже для больших текстовых корпусов. Потому синонимы часто несравнимы. К примеру, есть два слова ***отель*** и ***мотель***, с точки матрицы взаимной информации эти слова разные так как употребляются в совершенно разных контекстах.

	собака	кошка	машина	авто
отель	0	0.2	0	0.5
мотель	0.1	0	0.3	0

Можно попытаться путем транзитивного замыкания, матрицы взаимных вероятностей с некоторым коэффициентом транслируя словарные замыкания.

$$Если PMI(a,b) = 0 то PMI(a,b) = k \sum_{\forall c\in D} PMI(a,c) \cdot PMI(c,b) \quad (9)$$

Где $0 < k \ll 1$. За несколько итераций можно добиться ситуации, что не останется ни одной пары слов с нулевым значением в матрице взаимной информации P, непосредственное хранение которой не такая уж проблема учитывая ее сильную разреженность (об этом ниже). И такой метод (9) будет сглаживать, но гораздо хуже, чем методы факторизации описанные ниже.

Методы факторизации матрицы взаимной информации

Можно выделить следующие подзадачи. Сбор статистики (матрицы $P_{n\times n}$ взаимной информации), и метод её

[2] В дальнейшем будем использовать старое как более общее, оговаривая отдельно, является ли матрица сдвинутой (т.е. все ли ее элементы положительны).

[3] Могли бы если бы матрица взаимной вероятности, была бы полной, но на практике такого и близко нет. Как выйти из этого затруднения читайте в разделе **семантическое сглаживание**.

Рис. 1. Рост числа уникальных слов

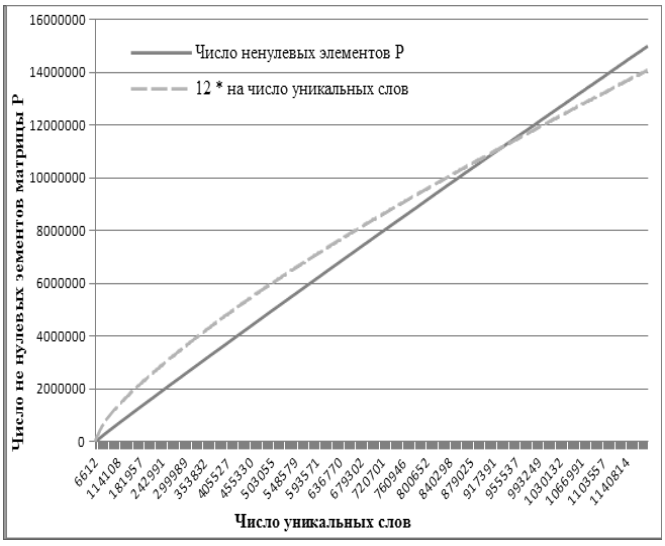

Рис. 2. Рост числа ненулевых элементов матрицы P

факторизации. В ***GloVe*** Они явно разделены, в варианте ***word2vec*** находятся внутри одного алгоритма. Независимо от того с какими параметрами и как собирается матрица P факторизовать её можно совершенно разными способами, с использованием разного количества свободных параметров, и разными алгоритмами. В рассматриваемых в данной статье методах ***word2vec*** [1] ***GloVe*** [4] используются несколько разные типы факторизации. В данных оригинальных работах ***GloVe*** стремится получить факторизацию (5) а word2vec факторизацию (10)

$$P_{n\times n} \approx W_{n\times m} C^T_{n\times m} \quad (10)$$

Где C — матрица с одинаковой с W размерностью. Эта матрица получила в работе [1] матрица векторов контекстных слов. В литературе [2] встречается такое объяснение второй матрице, что поскольку слово ***собака*** нечасто можно встретить в контексте слова ***собака***, то они как бы должны получить разные параметры, что приводит к некоторому парадоксу, и парадокс это как раз и разрешает матрица $C_{n\times m}$ контекстных векторов, где слову собака соответствует свой вектор отличный от вектора в матрице слов $W_{n\times m}$. Но автору данной работы представляется несколько иная причина, дело в том, что как следует из линейной алгебры, параметризация произвольной[4] матрицы $P_{n\times n}$ возможна только с использованием как минимум двух матриц.

Для факторизации матрицы P используют так же метод сингулярного разложения [6] Интересующий нас вариант сингулярного разложения имеет форму (11).

$$P_{n\times n} \approx U_{n\times m} \Sigma_{m\times m} V^T_{n\times m} \quad (11)$$

Где матрица $\Sigma_{m\times m}$ — это диагональная матрица с m сингулярных чисел матрицы $P_{n\times n}$ расположенный в порядке убывания. Матрицы $U_{n\times m}$ и $V_{n\times m}$ — это матрицы левых и правых сингулярных векторов. Данный метод дает невысокое качество в текстах на аналогиях. Причиной чему, как отмечается работе [3], является то, что векторное пространство распределяется равномерно для всех слов словаря D. В отличие от представлений ***word2vec*** и ***GloVe***, которые заточены на наиболее частотные слова. Хотя это еще спорный вопрос, что лучше, тем не менее, данный метод не будет описан в данной работе, так как его применение достаточно прямолинейно и не требует каких то специфических рекомендаций[5].

Свойства данных

В целях понимания работы методов стоит показать результаты статистического исследования. Например, как растет словарь в зависимости от числа просмотренных слов? Найдено экспериментальным путем, что число уникальных слов в запросах растет примерно как степень 0.8 от числа просмотренных слов[6] (рис. 1).

Также необходимо было понять, как быстро будет расти матрица взаимной информации в зависимости от числа *уникальных* слов, эксперимент на (рис. 2). пока-

[4] Не являющейся положительно полуопределенной, или даже симметричной.

[5] Почти все современные математические пакеты, так или иначе содержат в себе метод SVD.

[6] Данный график, как и все прочие, был получен с помощью итеративного измерения размера структуры по мере извлечения слов текстового корпуса.

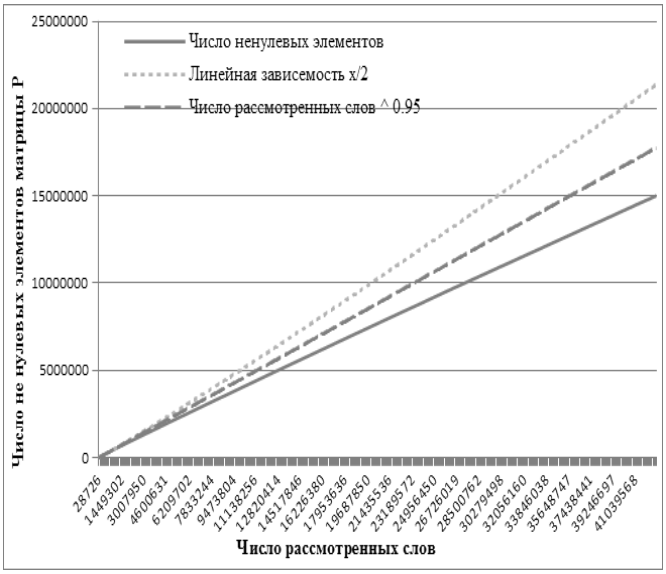

Рис 3. Рост числа ненулевых элементов матрицы P

зал, что плотность матрицы на довольно большом текстовом корпусе[7] соответствует линейному закону с коэффициентом соответствия $k \approx 12$. Это говорит о том, что лексика имеет свойство замкнутости.

На итоговом графике (рис. 3). можно увидеть, что рост числа ненулевых элементов в матрице взаимной информации в зависимости от числа слов в текстовом корпусе, нельзя даже назвать линейным.

Эти два графика имеют ряд весьма важных следствий, о которых необходим сказать. Во-первых, алгоритм онлайн факторизации ***word2vec*** при достаточно большом текстовом корпусе, имеет возможность спокойно адаптироваться под статистику, так как будто его запустили на очень большом числе эпох, и это собственно тот результат который рапортуют сами авторы метода [1]. Теперь имея на руках эту картин, стало ясно, почему это так.

Второе важное следствие, по сути, такое же как и для ***word2vec*** состоит в следующем, что поскольку рост плотности матрицы взаимной информации в зависимости от сила слов в текстовом корпусе даже слабее линейного, то сам алгоритм ***GloVe***, который непосредственно работает с хранимой в памяти матрицей взаимной информации в теории должен иметь сложность меньше линейной, а значит теоретически даже обгонять word2vec на больших данных, если процедура сбора и записи статистики работает быстро[8].

Метод GloVe

Метод, запечатленный в работе, [4] реализует выражение (5), чего он достигает за счет минимизации выражения (12) методом градиентного спуска [5][9]

$$E = \sum_{i=1,j=1}^{n,n} \left(W_{n\times m} C^T_{n\times m} - P_{n\times n}\right)^2 \qquad (12)$$

Весомым преимуществом данного метода является его хорошая сходимость, а также качество получаемого результата, которое [4] превосходит ***word2vec***.

Данный метод двухфазный, на первой фазе происходит накопление статистики, а именно вероятности слов и матрицы взаимных вероятностей. На второй фазе происходит собственно сам процесс факторизации. Факторизация использует тот же причем, что и ***word2vec*** а именно негативное семплирование (*negative sampling)*. Негативное сэмплирование — это процесс, при котором, параметры слов, не употребляемых в одном контексте, делаются различными. Вероятность слов собранная на первом этапе используется для такого негативного сэмплирования. Поскольку, матрица взаимной информации сильно разряжена (как kN, где $k \approx 12$), то получается так, что можно сэмплировать слова не обращая внимания на то $PMI(w,c) \neq 0$ (где c — негативно семплированнае слово) в некоторых ситуациях, поскольку в силу вышеуказанной причины это статистически не значимо. Практика показывает, что число негативно семплированых слов на одно положительное должно быть где-то 5[10]. Алгоритм факторизации ***GloVe*** итеративный[11], на каждом эпохе происходит оптимизация всех параметров, хотя градиент рассчитывается для каждого слова отдельно. Имею в виду необходимость минимизации функционала (12) можно записать следующее выражение для одной эпохи оптимизации. Где для каждого слова выделяется словарный вектор W_i из матрицы $W_{n\times n}$ и словарные вектора из контекстных слов C_j и C_k из матрицы $C_{n\times n}$ и к ним применяется операция градиента (14) с целью минимизировать функционал (13)

$$E = \sum_{\forall P_{ij}\neq 0} \left(W_i C^T_j - P_{ij}\right)^2 + \sum_{\forall i,k\in fsample} \left(W_i C^T_k\right)^2 \qquad (13)$$

Здесь показано, что хотя выражение (12) требует вычисления полной ошибки, в реальности вычисляется только та часть для которой $P_{ij} \neq 0$, и складывается с той частью, которая получается в результате негативного

[7] Порядка 10 млн. запросов.

[8] Надо понимать, что процедура сбора статистики, хотя и работает быстро, при качественной реализации, не может иметь сложность меньшую, чем O(n).

[9] Что был получить оригинальную формулу GloVe надо контекстную матрицу заменить на обычную, в св этой же работе далее будет использоваться эта формула.

[10] Чем больше текстовый корпус, тем меньше, но не меньше 3.

[11] Хватает 100, дальше либо останавливать оптимизацию, либо уменьшать параметр скорости обучения.

сэмплирования. Грубо говоря, так, ошибка есть сумма квадратов разности, где разность — это воспроизведенная взаимная вероятность[12] слов минус действительная, плюс то же самое, но только от случайных слов, для которых воспроизведенная взаимная вероятность должна быть равной нулю. От данной ошибки берется градиент, умножается на скорость обучения и вычитается из параметров для слова и для контекстных слов, как позитивных, так и негативных.

Упомянем вкратце, фазу сбора статистики. Если сбор вероятности слов текстового корпуса представляется очевидным[13], то сбор матрицы взаимной информации требует некоторых пояснений. Матрицу взаимной информации можно определить поэлементно как (15).

$$P_{ind(h),ind(k)} = \sum_{h\in T,k\in T,\vee h-k\vee\rho} f(h,k)\ ,$$

где как правило $f(h,k)=1$ (15)

Где $f(h,k)$ некая формула, которая обычно, и в данной работе тоже, является просто положительной константой (единица), но вообще говоря, может принимать большие значения для более близких слов. Соответственно T — это массив номеров слов, а *ind(h)* это функция получения номера в матрице $P_{n\times n}$ по номеру слова в текстовом корпусе, а p — это расстояние, в рамках которого слова считаются принадлежащими одному контексту.

Метод word2vec

В отличие от предыдущего метода, данный метод не содержит в себе фазы предварительного сбора статистики. Если упростить все до предела, то метод можно описать так: предварительно создается матрица (одна или две, обычная и для контекстных слов) заполненная случайными значениями (как и в предыдущем методе). Дальше алгоритм пробегает по всем словам и вектора соответствующие близким словам, делает чуть более одинаковыми, а для далеких, чуть более разными[14]. Вот и вся суть алгоритма.

Если говорить конкретно о методе word2vec, то он стремится к выражению (10), путем максимизации следующего функционала (16).

$$J = \sum_{\forall(i,j)\in T} \ln\sigma(V_i C_j) + \sum_{\forall(i,k)\notin T} \ln\sigma(-V_i C_k) \qquad (16)$$

Где V_i, C_j, C_k это словарные вектора слов с номерами i, j, k принадлежащие матрицам $W_{n\times n}$ и $C_{n\times n}$ соответственно. Но на практике же применяется несколько иной функционал (15).

$$J = \sum_{\forall(i,j)\in T} \ln\sigma(V_i C_j) + \sum_{\forall(i,k)\in fsample} \ln\sigma(-V_i C_k) \qquad (17)$$

Разница в том, что как и для алгоритма ***GloVe***, вместо пар не принадлежащих текстовому корпусу, рассматриваются пары полученные в результате негативного сэмплирования[15], где как и в случае ***GloVe*** они совершенно не обязательно будут именно негативными, в том смысле что их взаимная информация не будет равной нулю, хотя как и ***GloVe*** это не влияет на сходимость, в силу тех же причин[16]. Далее для каждого слова и его контекстных (а также негативных) применяется градиент (16) в режиме онлайн (по мере прохождения текстового корпуса).

Данный метод неработоспособен если в текстовом корпусе, будут слова (или замкнутой группы), которые не встречаются ни с какими словами (одно слово в предложении), так как это приводит к несходимости. В процессе негативного сэмплирования, словарный вектор данного слова, отдаляется от словарных векторов всех слов (поскольку ни с одним из слов не используется в одном контексте). Таким образом, достаточно быстро словарный вектор это одинокого слова растет мо модулю. В то же время, от самого такого одинокого слова, словарные вектора других слов так же "отдаляются" путем вычитания, но поскольку словарный вектор *одинокого* слова становится велик, сами эти словарные вектора тоже неограниченно растут. В результате работы метода, словарные вектора слова неограниченно растут, что приводит к ошибке. При наличии избыточного числа элементов словарного вектора по отношению к размеру словаря уникальных слова, описанные проблемы усугубляются[17].

Выводы

Описанные выше методы лексической параметризации, со всей очевидностью могут быть усовершенствованы с ценою потери простой интерпретации получаемых параметрических моделей. К примеру, если описанную, или подобную, но более подробную статистику попытаться факторизовать нейросетью, то не ясно, что будет с феноменом статистической аналогии.

[12] Которая есть просто скалярное произведение о вектора слова на вектор контекстного слова.

[13] За исключением, того что в литературе [4] упоминается желательность нормализации вероятностей степенью 0.75. Практика показал, что результаты действительно значительно улучшаются, причем в независимости от метода использующего негативное сэмплирование.

[14] Под близкими понимаются слова, в рамках некоторого окна/радиуса друг от друга, как и в случае ***GloVe***. Под далекими понимаются, слова полученные с помощью негативного семплирования.

[15] По заранее собранной вероятности слов.

[16] Добавим также, что по-другому для метода word2vec и быть не может, поскольку в явном виде он матрицу взаимной информации не собирает.

[17] так же автору данной работы так и не удалось стабилизировать работу с использованием двух матриц (основной и контекстной).

ЛИТЕРАТУРА

1. Distributed Representations of Words and Phrases and their Compositionality авторы: Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean
2. word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method авторы: Yoav Goldberg, Omer Levy
3. Neural Word Embedding as Implicit Matrix Factorization авторы: Yoav Goldberg, Omer Levy
4. GloVe: Global Vectors for Word Representation авторы Jeffrey Pennington, Richard Socher, Christopher D. Manning
5. Методы оптимизации в примерах и задачах. Пантелеев А. В., Летова Т. А. (2005, 544с.)
6. Себер Дж. Линейный регрессионный анализ. М.: Мир, 1980.— 456 с.