Искусственный интеллект: вычисление смысла текста.

Once a particular problem is considered solved, it often is not considered AI.
Jerry Kaplan

Введение

Как известно, люди многие годы пытались создать искусственный интеллект. Они разделили его на задачи — задачи искусственного интеллекта. Одной из задач искусственного интеллекта является понимание смысла текста.

Математика и смысл текста

Как можно *вычислить* смысл слова?

Дистрибутивная гипотеза в лингвистике говорит, что смысл слова в тексте определяется контекстом, то есть теми словами, которые употребляются вместе с ним. Во многих случаях можно считать, что контекст данного слова — это слова, которые находятся в непосредственной близости от него. Слова с похожим смыслом должны быть взаимозаменяемые, то есть вероятности встретить их в одном и том же контексте должны быть близки.

Пусть, для простоты, контекст состоит из одного слова. Представим для некоторого текста слова и контексты в виде векторов некоторого многомерного пространства размерности D, и рассмотрим множество векторов слов $\{w\}$ и множество векторов контекстов $\{c\}$. Из-за симметрии ситуации каждое слово является контекстом для какого-нибудь другого слова, поэтому $|\{w\}| = |\{c\}| = N$. Мы хотим, чтобы вероятность $P(c|w)$ того, что данное слово w находится в контексте c, соотносилась с их скалярным произведением $c \cdot w$. Для этого можно использовать функцию $softmax$, которая делает вектор похожим на распределение вероятности — все его компоненты оказываются в диапазоне $[0;\ 1]$, а их сумма равна 1. Применяя $softmax$ для всех скалярных произведений слова w на контексты, получим:

$$P(c|w) = softmax_{\{c_i|i \in 1..N\}}(c \cdot w) = \frac{e^{c \cdot w}}{\sum_{i=1}^{N} e^{c_i \cdot w}}$$

Можно составить матрицу контекстов C, **строки** которой будут векторами контекстов $C = (c_1,\ c_2,\ \ldots,\ c_N)^T$, тогда вектор вероятностей нахождения слова в каждом контексте $\delta_w = (P\ (c_1|w),\ \ldots,\ P\ (c_N|w))^T$ можно выразить через произведение матрицы на вектор $\delta_w = softmax\,(Cw)$. Введем *унитарный код* — вектор $e_i = (0_0,\ 0_1,\ \ldots,\ 1_i,\ \ldots,\ 0_{N-1},\ 0_N)^T$, в котором на i-м месте стоит единица, а все остальные нули. Тогда вектор слова w_i можно представить как произведение матрицы слов W, **столбцы** которой будут векторами слов $W = (w_1,\ w_2,\ \ldots,\ w_N)$, на унитарный код e_i: $w_i = We_i$, и значит, $\delta_i = softmax\,(CWe_i)$.

Матрицы C и W можно вычислить приближенно. Для этого будем рассматривать пары слово-контекст $\left(w_i;\ c_j\right)$, встречающиеся в тексте. Для хорошо подобранных матриц C и W ожидаемо, что вектор $\delta_i = softmax\,(CWe_i)$, будет иметь большее значение вероятности в j-м компоненте, так как слово w_i встретилось в контексте c_j, и меньшее значение в других компонентах. Можно вычислить какие компоненты C и W в первую очередь нужно изменить, чтобы приблизить δ_i к e_j, то есть к унитарному коду, соответствующему контексту c_j. Повторяя эту операцию много раз на случайно выбранных парах $\left(w_i;\ c_j\right)$, мы улучшаем коэффициенты матриц C и W, в результате чего вектор δ_i все лучше представляет распределение слова w_i по контекстам.

В различных работах было получено, что даже для текстов с количеством слов более миллиарда (порядка миллиона уникальных слов, то есть $N \sim 10^6$), для хорошего представления распределения слов достаточно пространства сравнительно небольшой размерности ($D \sim 300$). Более того, оказалось, что некоторые смысловые отношения между словами могут быть выражены как линейные комбинации соответствующих этим словам векторов. Один из самых известных примеров:

$$w_{king} - w_{man} + w_{woman} \approx w_{queen}$$

Если взять разность векторов, соответствующих словам $woman$ и man, то получается вектор, как бы означающий смысловую разницу между словами женского и мужского рода. Если теперь добавить эту разницу к вектору, соответствующему слову $king$, то новое слово должно быть во всех смыслах таким же как и $king$, но женского рода. Действительно, полученный вектор оказывается ближе всего к вектору слова $queen$.

Реализация модели

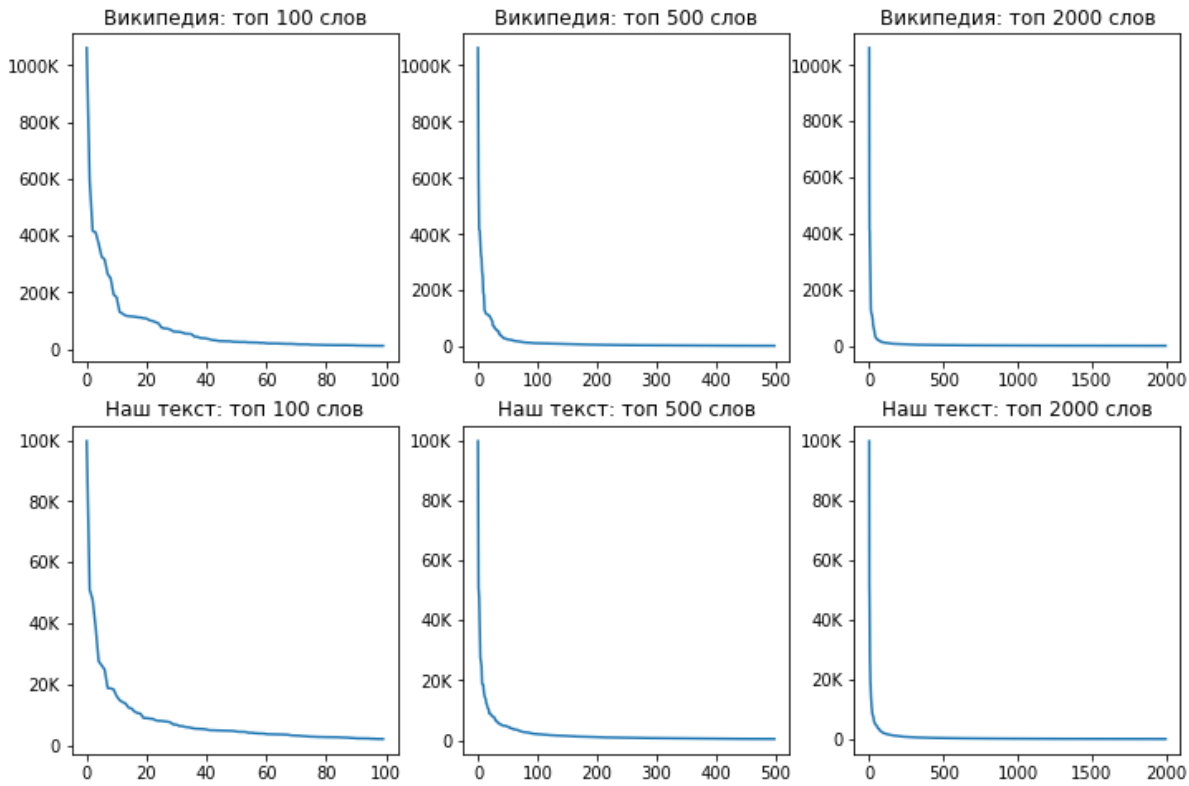
Одной из популярных библиотек для решения задач искусственного интеллекта является **`tensorfow`**. В ней есть реализация описанной модели под названием **`word2vec`** с параметрами, влияющими на качество модели: размерность пространства с векторами, скорость обучения (величина шага обучения), количество циклов

тренировки(эпох тренировки). Также в модели в качестве контекста слова используется группа из пяти слов слева и пяти слов справа от него.

Для демонстрации модели предлагается использовать: для тренировки — около 100 мегабайт склеенных статей из английской википедии, а для проверки качества модели — файл с аналогиями. Текст википедии содержит более 17 миллионов слов, из них 235 тысяч уникальных. Запущенная с настройками по умолчанию модель после тренировки на английской википедии правильно угадывает около 35% ответов.

Что мы сделали сами?

Наша задача — попытаться построить векторы для русского языка и исследовать полученное пространство. В качестве текста на вход мы взяли большие по объему тексты русских писателей: Л.Н.Толстого, Н.А.Некрасова, И.С.Тургенева. Чтобы наш текст был похож на образец текста википедии, мы удалили всю пунктуацию и цифры, и получили список слов, разделенных пробелом. Из графиков видно, что распределение слов по встречаемости такое же, как и в тексте википедии. Итоговый объем нашего текста получился около 23 мегабайт, в нем более 2 миллионов слов, из них 134 тысячи уникальных.



Мы просмотрели несколько тысяч самых часто встречаемых слов, и сделали свой набор аналогий для русского языка:

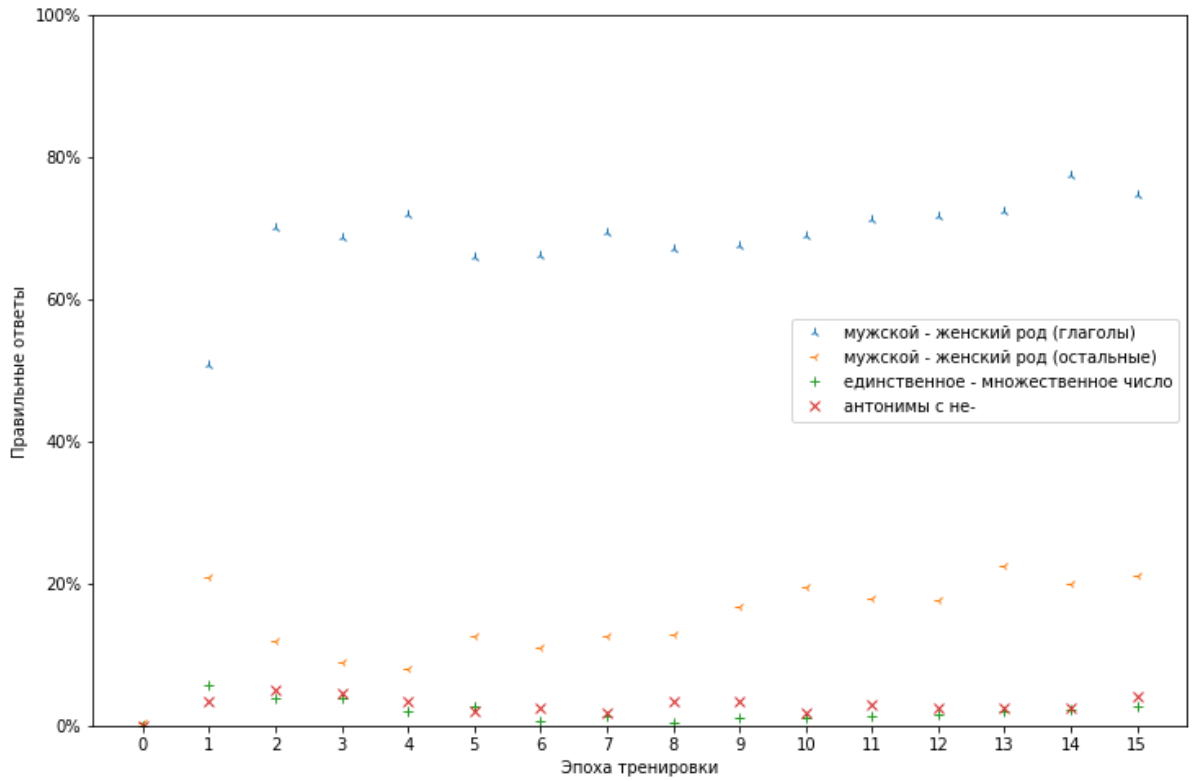
мужской - женский род (глаголы прошедшего времени)		
посмотрел-посмотрела	спросил-спросила	слышал-слышала
вышел-вышла	сказал-сказала	был-была
понял-поняла	встал-встала	подошел-подошла
продолжал-продолжала	хотел-хотела	видел-видела
подумал-подумала	знал-знала	думал-думала
стал-стала	отвечал-отвечала	говорил-говорила
жил-жила	делал-делала	мог-могла

мужской - женский род (остальные)		
он-она	его-ее	муж-жена
девушка-мальчик	добрый-добрая	мужчина-женщина
старик-старуха	один-одна	онъ-она
должен-должна	мой-моя	твой-твоя
другой-другая	князь-княгиня	князь-княжна
который-которая	сам-сама	нему-ней
ему-ей	него-нее	

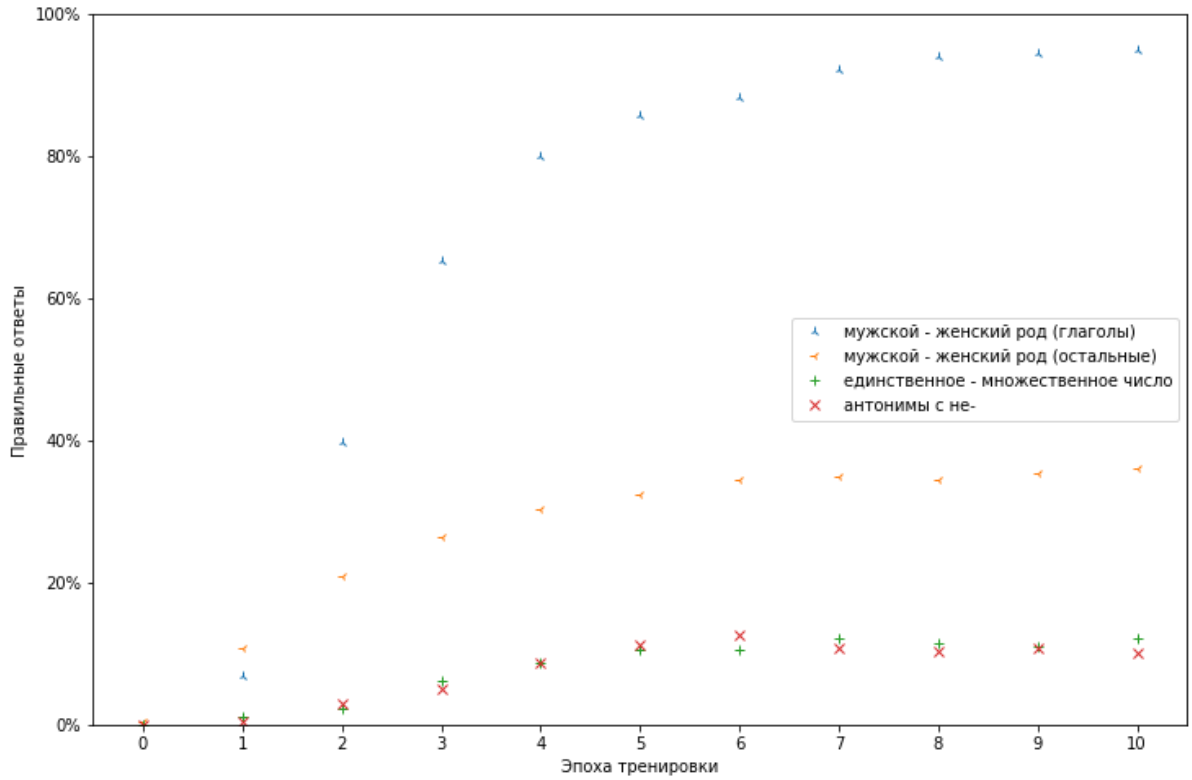
единственное число - множественное число		
человек-люди	я-мы	голова-головы
лицо-лица	окно-окна	грех-грехи
закон-законы	глаз-глаза	желание-желания
дорога-дороги	мужчина-мужчины	война-войны
мужик-мужики	этот-эти	народ-народы
офицер-офицеры	солдат-солдаты	женщина-женщины

антонимы с не-		
известно-неизвестно	возможно-невозможно	охотно-неохотно
счастье-несчастье	терпеливо-нетерпеливо	долго-недолго
заметно-незаметно	справедливо-несправедливо	правда-неправда
давно-недавно	всех-никого	всегда-никогда
хорошо-нехорошо	далеко-недалеко	можно-нельзя
приятно-неприятно		

При использовании параметров по умолчанию, достаточно хороший результат получился только для группы: **мужской - женский род(глаголы)**. Из графика видно, что количество правильных ответов колеблется вокруг нескольких установившихся значений. Предположительно это вызвано слишком большим шагом обучения, из-за которого мы “проскакиваем” оптимальные значения. Также, очевидно, можно уменьшить количество эпох до десяти без потери качества модели.



Постепенные уменьшения начального шага тренировки(до 20% от исходного) позволили получить более качественную модель, заметно увеличив процент правильных ответов:



Удивительно, но эксперименты показали, что сокращения размерности пространство векторов с 200 до 100 практически не влияет на качество модели. Также на качество модели не влияет увеличение размерности пространство до 300.

Выводы

Модель, разработанная для английского языка, оказалась также применима и для русского языка. При этом полное удаление пунктуации и склеивание предложений вместе не помешало построить модель, способную обнаруживать смысловые аналогии.
Размерность пространства слов в большом диапазоне не влияет на качество модели.
Различные аналогии по-разному обнаруживаются в этой модели.
Авторы исходной статьи утверждают, что качество модели растет с количеством обработанного ей текста, и приводят свои результаты для массива из миллиарда слов Google News англоязычных выпусков новостей. К несчастью, у нас столько слов нет :)