Вычисление смысла текста

Исследовательская работа
ученика 10”А” класса лицеяN°281
Еникеева Дмитрия.
Научный руководитель: Абрамова А.Н.

Цели и задачи

- Рассмотреть представление слов в виде многомерных векторов.
- Проверить его применимость для русского языка.
- Исследовать свойства полученной модели.

Введение

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

Искусственный интеллект — перспективная область изучения.

Одна из задач искусственного интеллекта — понимание смысла текста.

Настройка смысла вручную — практически невозможна.

Хочется автоматическое **вычисление** смысла **неизвестных** слов.

Математика и смысл слова

Как же **вычислить** смысл слова?

“Маленькая, пушистая **канна** залезла на дерево.”

— что такое “канна”? Что-то вроде белки?

Математика и смысл слова

Как же **вычислить** смысл слова?

“Маленькая, пушистая **канна** залезла на дерево.”

— что такое “канна”? Что-то вроде белки?

Дистрибутивная гипотеза в лингвистике утверждает, что лингвистические единицы, встречающиеся в схожих контекстах, имеют близкие значения.

Смысл слова во многом определяется словами, стоящими рядом с ним — то есть его контекстом.

Математика и смысл слова

Для простоты, пусть контекст — одно слово, слева или справа.

Представляем слова и контексты в виде D-мерных векторов.

Множество векторов слов {**w**} и множество векторов контекстов {**c**}.

Нужно, чтобы вероятность **P**(**c**|**w**) того, что слово **w** находится в контексте **c**, соотносилась со скалярным произведением **c**·**w**.

Как из скалярного произведения получить вероятность?

Математика и смысл слова

$$softmax \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \frac{1}{\sum_{i=1}^{N} e^{x_i}} \begin{pmatrix} e^{x_1} \\ e^{x_2} \\ \vdots \\ e^{x_N} \end{pmatrix}$$

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

Математика и смысл слова

Матрица контекстов **C**: строки — **D**-мерные векторы контекстов

$$C = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} = \begin{pmatrix} c_1^{(1)} & c_1^{(2)} & \cdots & c_1^{(D)} \\ c_2^{(1)} & c_2^{(2)} & \cdots & c_2^{(D)} \\ \vdots & \vdots & \ddots & \vdots \\ c_N^{(1)} & c_N^{(2)} & \cdots & c_N^{(D)} \end{pmatrix}$$

Математика и смысл слова

Вектор вероятностей δ_w нахождения слова **w** в каждом контексте:

$$\delta_w = \begin{pmatrix} P(c_1|w) \\ P(c_2|w) \\ \vdots \\ P(c_N|w) \end{pmatrix} = softmax(Cw)$$

Математика и смысл слова

Унитарный код $\mathbf{e_i}$:

$$e_i = \begin{pmatrix} 0_1 \\ \vdots \\ 0_{i-1} \\ 1_i \\ 0_{i+1} \\ \vdots \\ 0_N \end{pmatrix}$$

Матрица слов **W**: столбцы — векторы слов

$$W = (w_1\ w_2\ \ldots\ w_N) = \begin{pmatrix} w_1^{(1)} & w_2^{(1)} & \ldots & w_N^{(1)} \\ w_1^{(2)} & w_2^{(2)} & \ldots & w_N^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ w_1^{(D)} & w_2^{(D)} & \ldots & w_N^{(D)} \end{pmatrix}$$

Математика и смысл слова

Вероятности для i-го слова: $\delta_i = softmax(CWe_i)$

Берем из текста случайные пары слово-контекст ($\mathbf{w}_i$; $\mathbf{c}_j$)

Для хорошей модели в δ_i **j**-й компонент большой, если слово часто встречается в контексте $\mathbf{c}_j$.

Для каждой пары стремимся чтобы δ_i было ближе к $\mathbf{e}_j$.

Пошагово улучшаем коэффициенты матриц.

Свойства модели

- Линейные операции на векторах модели имеют смысл!

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

  Ближайший вектор к (**king** − **man** + **woman**) — вектор **queen**.

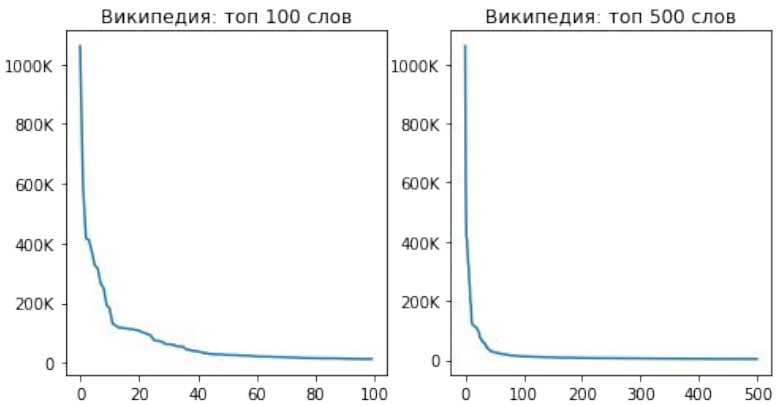
- Многие другие аналогии устроены схожим образом.
- Перевод на другой язык — умножение на матрицу!
- Модель активно исследуется (статьи начали выходить в 2013).

Что мы сделали

Стандартная модель:

- Английская Википедия 100 Мб
- Вся пунктуация удалена
- 17 000 000 слов, 235 000 уникальных

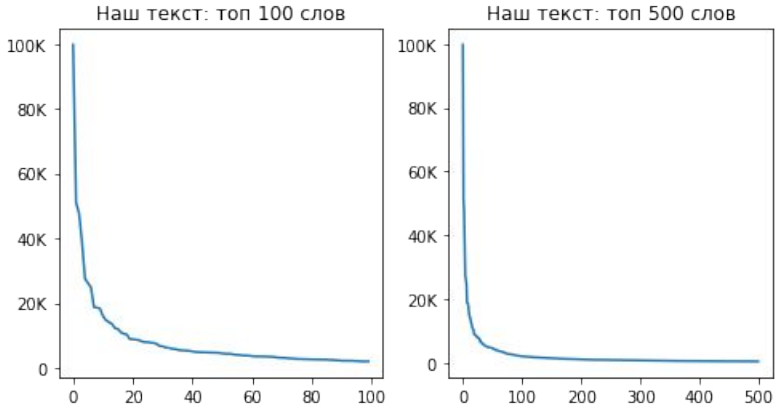
Распределение слов по частоте:



Наша модель:

- Тексты русских писателей 23 Мб
- Вся пунктуация удалена
- 2 000 000 слов, 134 000 уникальных

Распределение слов по частоте:



Что мы сделали

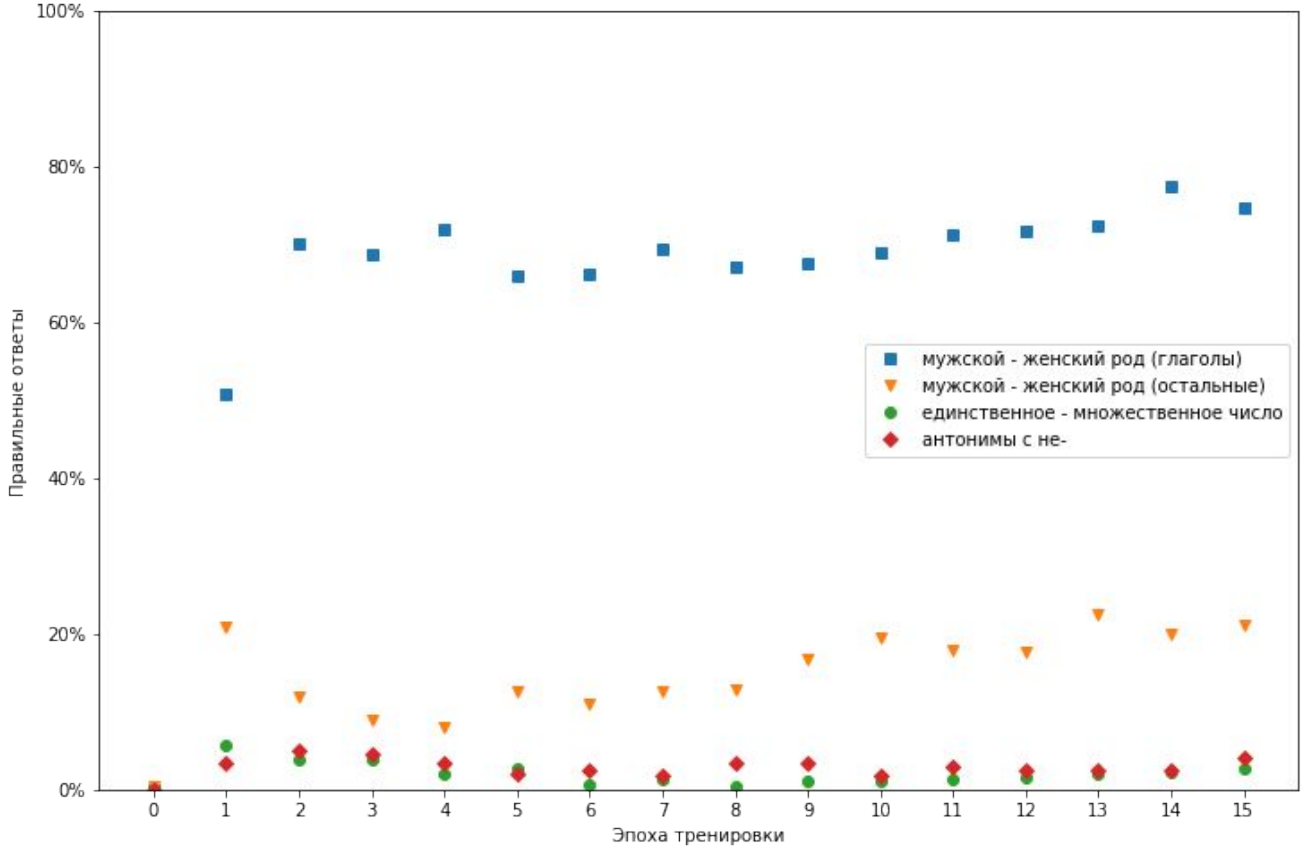
Проверка качества модели:

- Род слов
  - man-woman, king-queen, ...
- Столица - страна
  - Athens-Greece, Cairo-Egypt, ...
- Время глаголов
  - flying-flew, saying-said, ...
- Страна - валюта
  - USA-dollar, Poland-zloty, ...

Проверка качества модели:

- Род глаголов
  - жил-жила, мог-могла, ...
- Род остальных слов
  - муж-жена, твой-твоя, ...
- Единственное/множественное число
  - человек-люди, я-мы, ...
- Антонимы с “не-”
  - можно-нельзя, всегда-никогда, ...

Что мы сделали



Что мы сделали

Как можно улучшить результат?

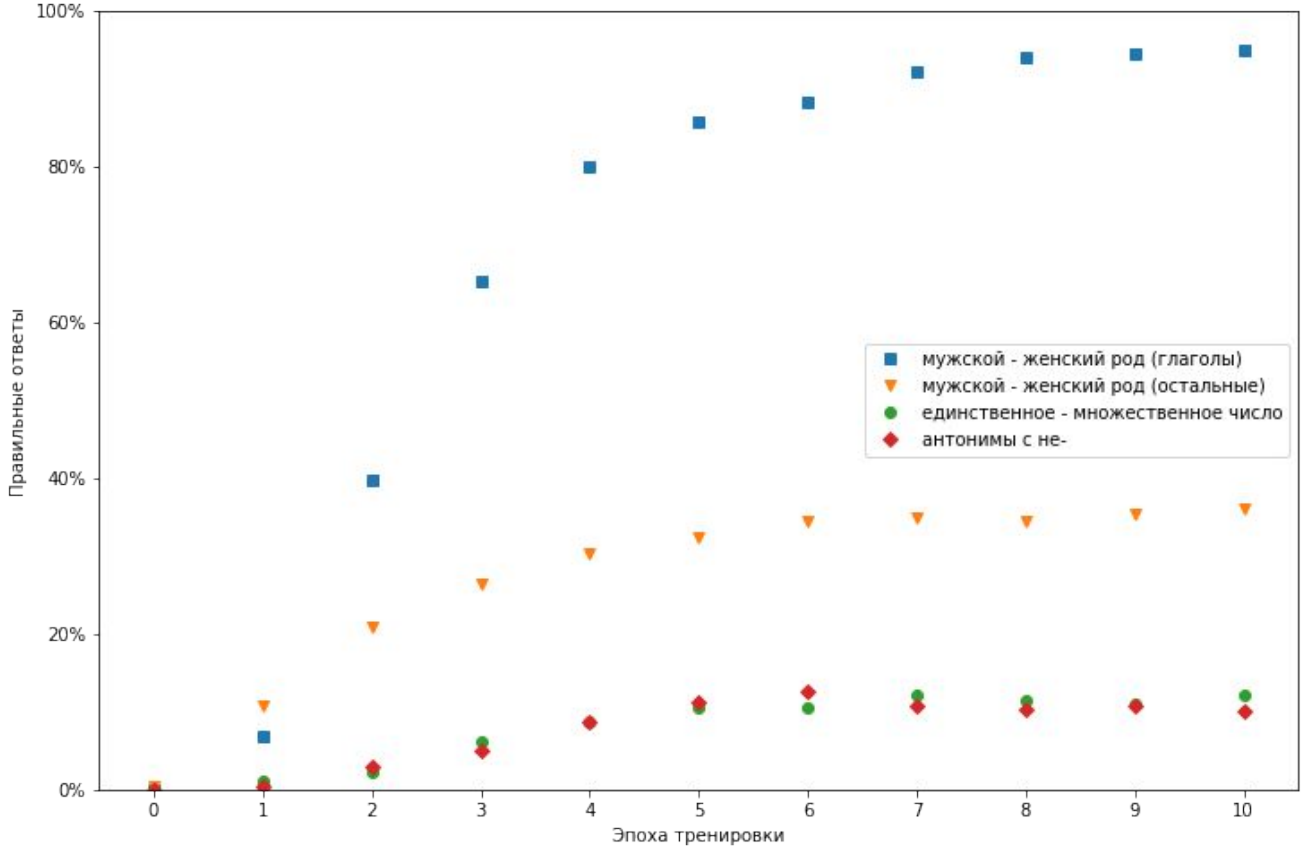
- Увеличить размерность пространства.
- Больше слов в контексте.
- Уменьшить шаг обучения.
- Изменить количество эпох.

Что мы сделали

Как можно улучшить результат?

- Увеличить размерность пространства — **не влияет.**
- Больше слов в контексте — **небольшое улучшение.**
- Уменьшить шаг обучения — **сильно помогло!**
- Изменить количество эпох — **не влияет.**

Что мы сделали



Выводы

- Модель применима также к русскому языку.
- ...несмотря на удаление пунктуации!
- Размерность пространства мало влияет на качество модели.
- Некоторые аналогии очень хорошо.
- Больше текста — лучше результат.
    - (по мнению авторов английской статьи)

Источники

- Mikolov et al., Efficient Estimation of Word Representations in Vector Space. (arXiv:1301.3781v3)
- A. Mnih, K. Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation.
- Y. Goldberg, O. Levy. word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method. arXiv 1402.3722v1
- C. Moody. A Word is Worth a Thousand Vectors.
- Mikolov et al., Exploiting Similarities among Languages for Machine Translation. arXiv:1309.4168v1
- Сайт https://www.tensorflow.org/tutorials/word2vec.

Спасибо за внимание!