Классификация +. Задачи NLP. Word2vec.

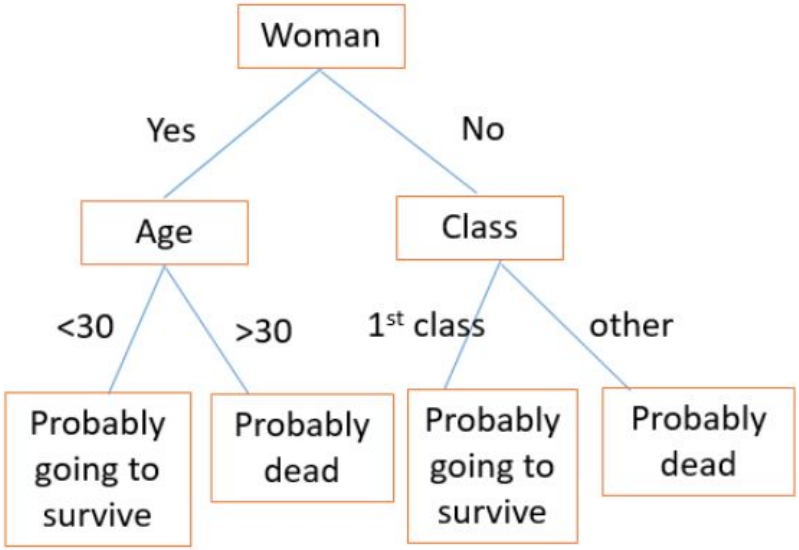
Маша Шеянова, masha.shejanova@gmail.com

План

- Больше про деревья решений
- Другие задачи NLP
- эмбеддигни и Word2vec

Больше о деревьях решений

Reminder



Источник картинки.

Дерево решений на примере датасета из титанка.

Как строятся деревья?

Сверху-вниз: сначала находим корень, потом в каждом из поддеревьев — новый корень, и так далее.

Как выбираем корень? Вводим “impurity function” — насколько плохо классифицирован датасет. Каждое разделение классифицирует датасет чуть лучше. Impurity function может быть

Алгоритмы:

- CART (Classification and Regression Trees) → Gini Index
- ID3 (Iterative Dichotomiser 3) → Entropy, Information gain

Алгоритм

1. Вычислить impurity function для изначального датасета
2. Для каждого признака:
    a. вычислить impurity function для каждого сплита
    b. вычислить, насколько текущий атрибут лучше, чем было до него
3. Выбрать атрибут с лучшей разницой в impurity function
4. Повторять, пока мы не захотим остановиться

Вспомним энтропию

Как измерить насколько распределение "разнородное", или насколько “грязный” датасет?

Взять математическое ожидание количества бит, которое понадобится, чтобы закодировать один из исходов в **оптимальной** кодировке.

Это количество бит — **информация**
1/ log p(x) = -log p(x)

Мат ожидание информации - **энтропия**

$$Entropy = -\sum p(X) \log p(X)$$

here p(x) is a fraction of examples in a given class

Энтропия

Entropy

Entropy $H(S)$ is a measure of the amount of uncertainty in the (data) set S (i.e. entropy characterizes the (data) set S).

$$H(S) = \sum_{c \in C} -p(c) \log_2 p(c)$$

Where,

- S – The current (data) set for which entropy is being calculated (changes every iteration of the ID3 algorithm)
- C – Set of classes in S C={ yes, no }
- $p(c)$ – The proportion of the number of elements in class c to the number of elements in set S

When $H(S) = 0$, the set S is perfectly classified (i.e. all elements in S are of the same class).

In ID3, entropy is calculated for each remaining attribute. The attribute with the **smallest** entropy is used to split the set S on this iteration. The higher the entropy, the higher the potential to improve the classification here.

- If all examples are of the same class or all are negative then entropy will be 0.
- If there is 50/50 distribution, it’s 1 (high)

Information gain

Разница между энтропией до и после разделения. Иными словами, насколько “чище”, “определённее” стали данные.

$$IG(A, S) = H(S) - \sum_{t \in T} p(t) H(t)$$

Where,

- $H(S)$ – Entropy of set S
- T – The subsets created from splitting set S by attribute A such that $S = \bigcup_{t \in T} t$
- $p(t)$ – The proportion of the number of elements in t to the number of elements in set S
- $H(t)$ – Entropy of subset t

Gini

Используется в CART. Это ещё один способ задать impurity function.

$$1 - \sum_{t=0}^{t=k} P_t^2$$

Maximum value of Gini Index could be when all target values are **equally distributed**.

Minimum value of Gini Index will be 0 when all observations belong **to one label**.

Деревья решений в sklearn

sklearn.tree.DecisionTreeClassifier

Параметры:

- criterion (gini или entropy)
- min_samples_split — сколько должно быть точек данных, чтобы мы продолжили делить
- min_impurity_decrease
- max_depth — максимально возможная высота дерева
- max_leaf_nodes — максимально возможная “ширина” дерева

Задачи NLP в МО

О чём уже говорили

Класификация текста:

- spam detection
- жанры
- sentiment analysis (тональность)
- предсказание темы

Кластеризация текстов:

- новости
- topic modelling

Что ещё?

- на уровне текстов
- на уровне предложений
- на уровне слов
- speech, OCR, image captioning

Предложения

- Paraphrase
- Textual entailment
- QA systems
- machine translation

Слова

- POS-tagging
- named entity recognition

Эмбеддинги и Word2vec

Дистрибутивная семантика

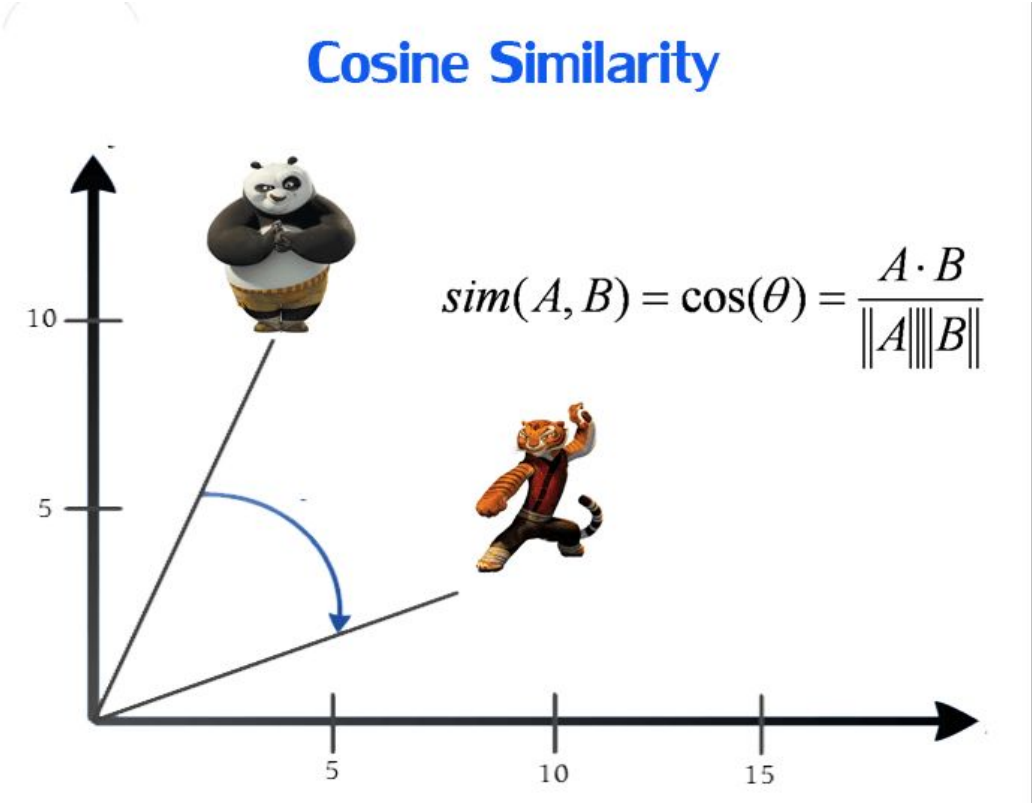
Что мы хотим:

- формальный способ считать лексическую близость
- глобально: научить компьютер извлекать смыслы из текста

Как делать это автоматически?

Дистрибутивная гипотеза: значения слов полностью определяются их контекстами. Слова с похожими типичными контекстами имеют схожее значение.

Как найти, насколько близки слова?



- надо найти способ превратить слова в вектора так, чтобы они отражали **контекст**
- найти расстояние между этими векторами одним из способов

Источник картинки.

Как сделать из слов вектора?

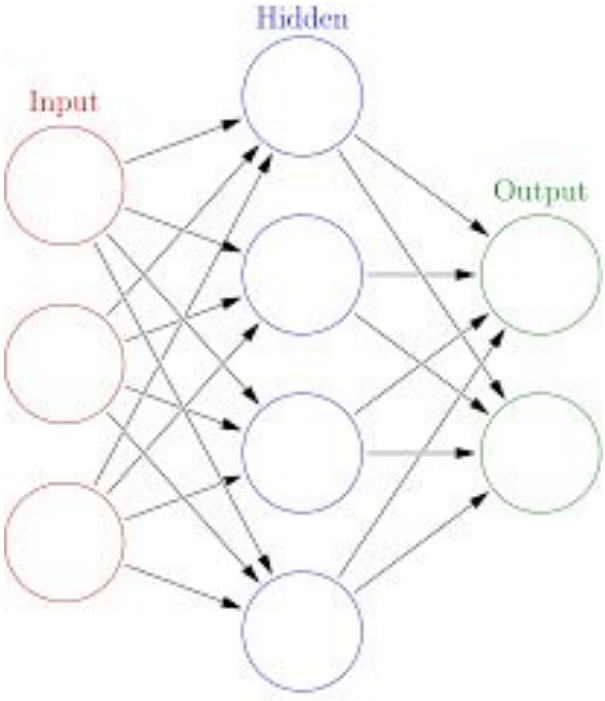
Итак, основная идея — **учитывать контекст**. Но как? А вот про это есть большая наука.

Самый простой-наивный метод — **счётный**. Идея: для каждого слова возьмём ближайшие в некотором окне (например, -5 +5). Сделаем такой же мешок слов, как делали для документов (CountVectorizer, TfidfVectorizer). Можно делать “скользящее окно”.

Плюсы: легко и быстро.

Минусы: для большого корпуса — очень большие вектора.

нейросеть in a nutshell



На входе — вектор признаков.

На каждой стрелочке — какие-то коэффициенты.

На выходе — вектор вероятностей того или иного класса.

“Нейрон” == один кружочек.

Word2vec

In general, Word2Vec — это метод строить гораздо более компактные эмбеддинги с помощью нейросетей.

Методы:

- CBOW (Common Bag Of Words)
- skipgram

CBOW

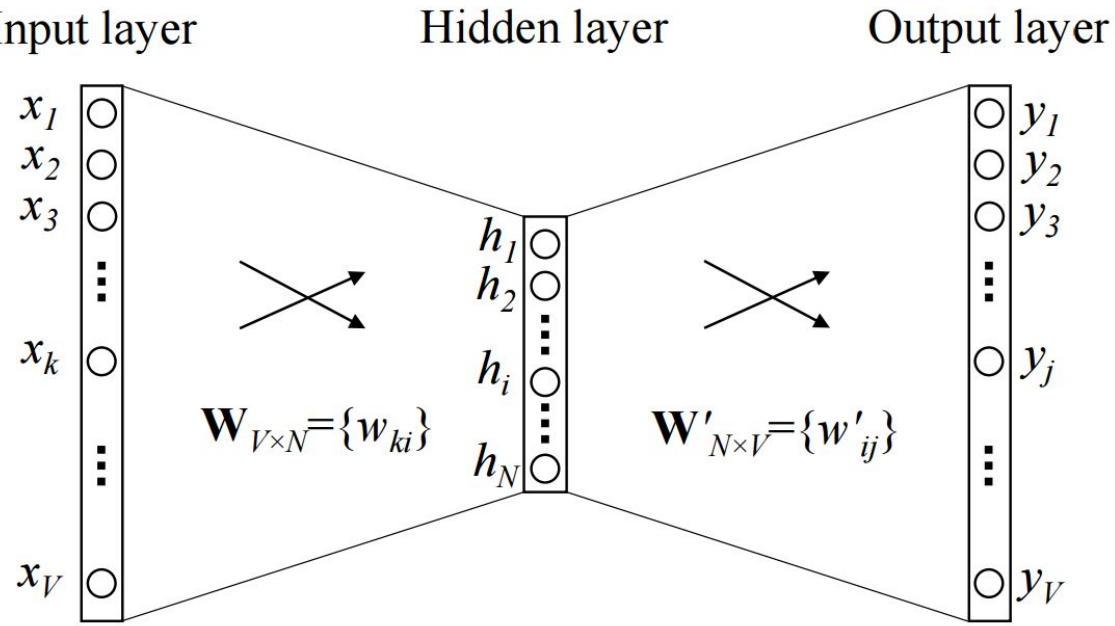


Figure 1: A simple CBOW model with only one word in the context

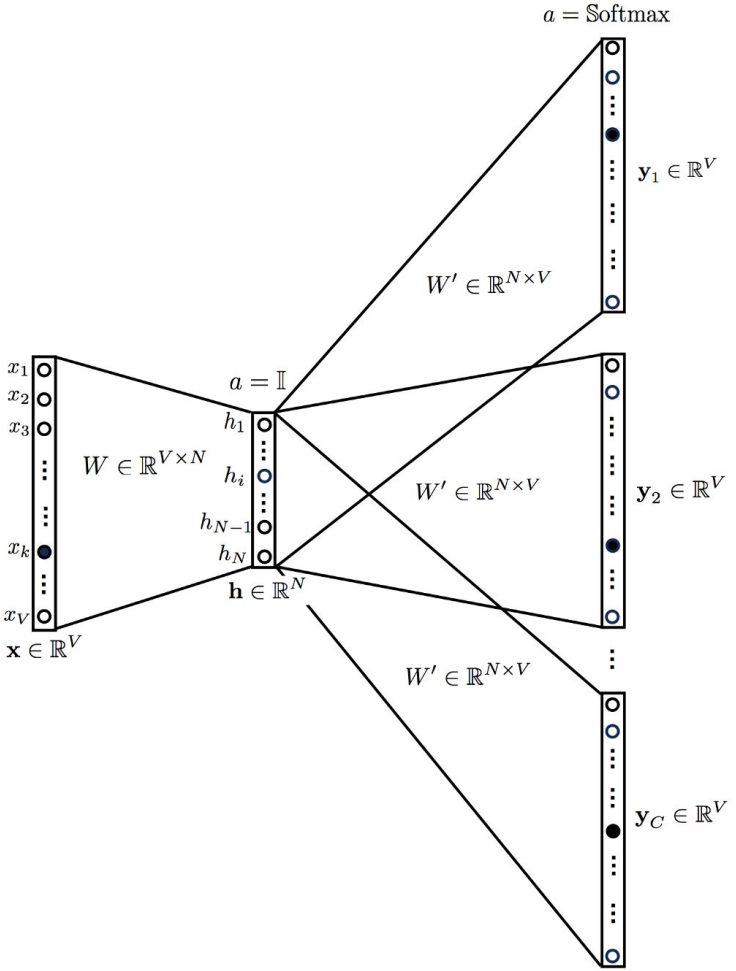
Источник

Takes the context of each word as the input and tries to predict the word.

skipgram

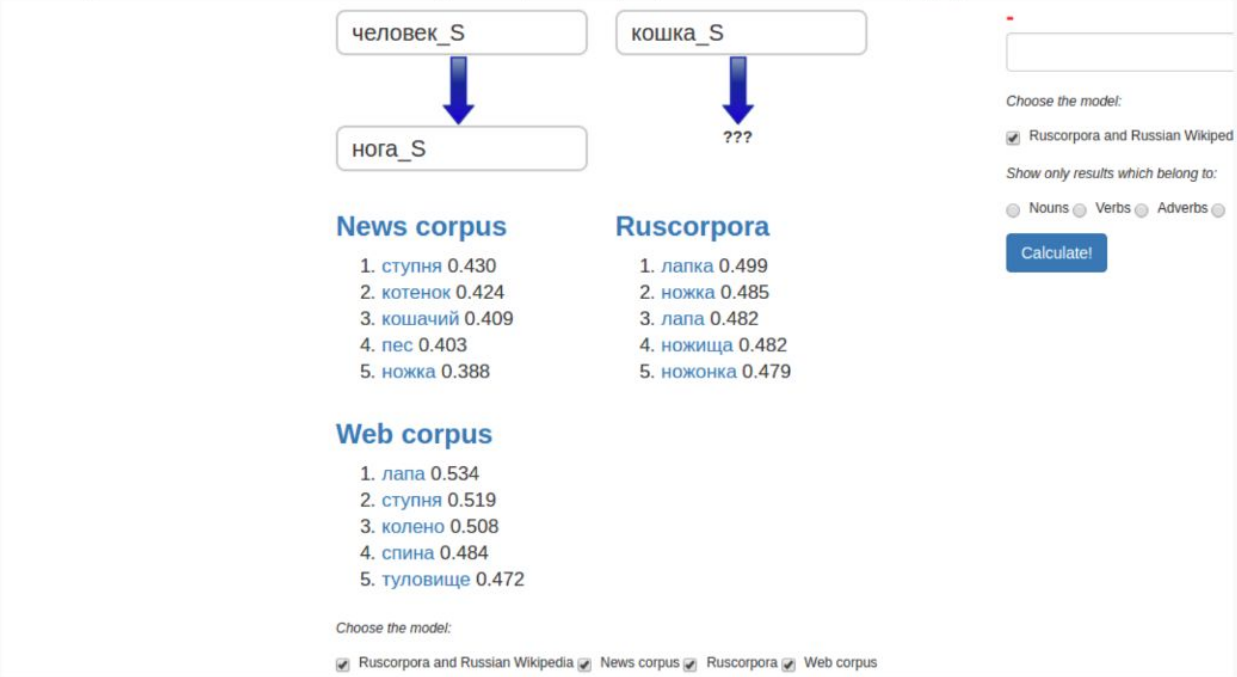
skipgram, в отличие от CBOW, пытается предсказывать контекст по слову.

- **Skip Gram** works well with small amount of data and is found to **represent rare words** well
- On the other hand, **CBOW** is faster and has **better representations for more frequent words**



Rusvectores, word2vec для русского

На rusvectores можно найти слова, наиболее близкие к данному, построить семантическую пропорцию и многое другое.



Где взять готовые эмбеддинги

Я рассказала, как обучить свои эмбеддинги. Но это долго, заморочно и не всегда нужно. Есть ли уже обученные эмбеддинги? Конечно!

Rusvectores! (для русских слов)

Ресурсы

Почитать

- про деревья решений
- про энтропию и information gain для деревьев решений
- Introduction to Word Embedding and Word2Vec
- Word2Vec and FastText Word Embedding with Gensim