Языковые модели

1. Задача предсказания следующего слова

Одной из задач обработки текстов на естественном языке выступает задача предсказания следующего слова по предшествующим. Примерами таких задач являются задача предсказания следующего слова при наборе текста в смартфоне или дополнение поисковых запросов.

Данная задача может быть сведена к оценке вероятностей встретить каждое из возможных слов после имеющегося. Соответствующая языковая модель примет вид:

$$w^* = \text{argmax}_{w_k \in V} P(w_k \mid w_{k-1}), \tag{1}$$

$$P(w_k \mid w_{k-1}) = \frac{P(w_k)\, P(w_{k-1} \mid w_k)}{P(w_{k-1})}, \tag{2}$$

где V – множество всех возможных слов, а $P(w_k \mid w_{k-1})$ – вероятность встретить слово w_k после w_{k-1}.

Для удобства полученную модель можно представить в следующем виде:

$$w^* = \text{argmax}_{w_k \in V} [\log P(w_k) + \log P(w_{k-1} \mid w_k)]. \tag{3}$$

Иногда по последнему слову оказывается невозможным определить следующее, поскольку в нем не учитывается контекст. Например, если последнее слово является союзом, то приемлемые варианты следующего слова определяет и предшествующее данному союзу слово. Оценка вероятности в данном случае проводится на основе m последних слов. Таким образом, модель (1)-(2) примет вид:

$$w^* = \text{argmax}_{w_k \in V} P(w_k \mid w_{k-1}, w_{k-2}, \ldots, w_{k-m}), \tag{4}$$

$$P(w_k \mid w_{k-1}, w_{k-2}, \ldots, w_{k-m}) = ((P(w_k)\, P(w_{k-1}, w_{k-2}, \ldots, w_{k-m} \mid w_k)) / P(w_{k-1}, w_{k-2}, \ldots, w_{k-m})). \tag{5}$$

Предположим, что текущее слово w_k зависит только от того, какие слова встретились перед ним и не зависит от того, в каком порядке они встретились. Тогда

$$w^* = \text{argmax}_{w_k \in V} \left[P(w_k) \prod_{i=1}^{m} P(w_{k-i} \mid w_k) \right] = \text{argmax}_{w_k \in V} \left[\log P(w_k) + \sum_{i=1}^{m} \log P(w_{k-i} \mid w_k) \right]. \tag{6}$$

Полученная языковая модель также используется для решения задачи классификации документов.

1.1. Наивный байесовский классификатор

Пусть стоит задача определить категорию новостей по их тексту. Тогда оценки вероятностей принадлежности новости к каждой $c \in C$ категории могут быть найдены по формуле:

$$c^* = \text{argmax}_{c \in C} \frac{P(c)\, P(d \mid c)}{P(d)}. \tag{7}$$

Здесь $P(c)$ – вероятность встретить новость данной категории, рассчитываемая по формуле:

$$P(c) = \frac{N_c}{N}, \tag{8}$$

где N_c – количество новостей в категории c, N – общее число новостей;

$$P(d \mid c) = P(w_1 \mid c)\, P(w_2 \mid c) \ldots P(w_m \mid c) = \prod_{i=1}^{m} P(w_i \mid c), \tag{9}$$

где w_1, ..., w_m – слова, встретившиеся в новости d (в предположении, что все слова независимы), а

$$P(w_i \mid c) = \frac{\nu_{i\,c} + 1}{\sum_{i' \in V} (\nu_{i'\,c} + 1)} = \frac{\nu_{i\,c} + 1}{|V| + \sum_{i' \in V} \nu_{i'\,c}} \tag{10}$$

после применения сглаживания Лапласа для устранения проблемы неизвестных слов;

$P(d)$ – оценка вероятности встретить новость, состоящую из данного набора слов. Поскольку $P(d)$ не оказывает влияния на результат, то итоговая модель классификации может быть представлена в виде:

$$c^* = \operatorname{argmax}_{c \in C} \left[\log P(c) + \sum_{i=1}^{m} \log P(w_i \mid c) \right]. \tag{11}$$

1.2. Оценка условных вероятностей в языковой модели на основе словаря

Условные вероятности в формуле (3) могут быть оценены непосредственно по имеющемуся словарю слов. Для этого необходимо рассмотреть имеющиеся в словаре биграммы, тогда:

$$P(w_{k-1} \mid w_k) = \frac{\nu_{(w_{k-1}, w_k)}}{\sum_{(w_i, w_j) \in V} \nu_{(w_i, w_j)}}, \tag{12}$$

где $\nu_{(w_{k-1}, w_k)}$ – частота встречаемости словосочетания $(w_{k-1},\ w_k)$.

1.3. Представление в виде задачи машинного обучения

Задачу предсказания следующего слова по предыдущим можно представить как задачу многоклассовой классификации, где $y = 1$, если данное слово следует за предыдущим, и $y = 0$ в противном случае. Обучающую выборку в данном легко получить с помощью скользящего окна.

Рассмотрим строку *«Success is the ability to go from failure to failure without losing your enthusiasm»* и пройдем по ней окном размера 3. Таким образом наберется обучающая выборка, где первые два слова являются признаками, а последний столбец – целевой переменной.

Success	is	the
is	the	ability
the	ability	to
ability	to	go
to	go	from
go	from	failure
from	failure	to
failure	to	failure
to	failure	without
failure	without	losing
without	losing	your
losing	your	enthusiasm

Представить слова обучающей выборки в виде вектора можно с помощью метода One-hot-encoding. Число уникальных слов в представленной строке равно 12. Тогда для первого объекта на вход модели будет подаваться следующий вектор:

{1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},

а соответствующее значение целевой переменной:

$$\{0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0\}.$$

Таким образом проводится оценка вероятностей каждого возможного варианта.

---

2. Векторное представление слов

Для обучения предсказательных и прогнозных моделей текстовые данные необходимо представить в виде числового вектора. Наиболее простыми подходами являются представление в виде частотного вектора и Tf-Idf. Однако, данные подходы не позволяют учесть контекст слов, поскольку при формировании «мешка слов» теряется информация о взаимном расположении слов в тексте.

2.1. Continuous Bag of Words (CBoW)

Если стоит задача восстановления пропущенного слова, то кроме слов перед целевым можно рассматривать также и последующие слова. Пусть в рассмотренном ранее примере пропущено слово *«the»*, тогда при ширине окна, равной 5, признаками будут выступать слова *«Success»*, *«is»*, *«ability»*, *«to»*

Success	is	ability	to	the
is	the	to	go	ability
the	ability	go	from	to
ability	to	from	failure	go
to	go	failure	to	from
go	from	to	failure	failure
from	failure	failure	without	to
failure	to	without	losing	failure
to	failure	losing	your	without
failure	without	your	enthusiasm	losing

Данная архитектура называется Continuous Bag of Words.

2.2. Skip-gram

Архитектура skip-gram устроена несколько иначе. Если CBoW предсказывает слово по его окружению, то в архитектуре skip-gram, наоборот, по текущему слову необходимо предсказать его окружение. Тогда обучающая выборка будет представлена в следующем виде:

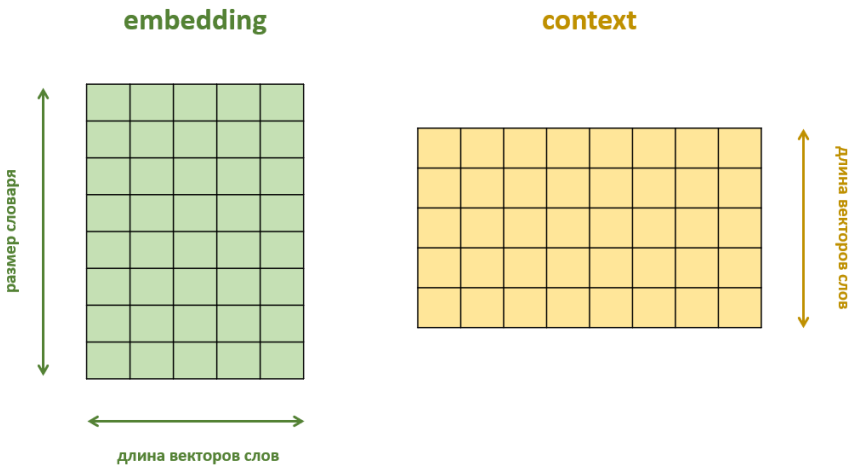
«Success is the ability to go from failure to failure without losing your enthusiasm»

the	Success
the	is
the	ability
the	to
ability	is
ability	the
ability	to
ability	go
to	the
to	ability
to	go
to	from
go	ability
go	to
go	from
go	failure
from	to
from	go
from	failure
from	to
failure	go
failure	from
failure	to
failure	failure
to	from
to	failure
to	failure
to	without
failure	failure
failure	to
failure	without
failure	losing
without	to
without	failure
without	losing
without	your
losing	failure
losing	without
losing	your
losing	enthusiasm

2.3. Модель word2vec

Идея модели word2vec состоит в том, чтобы представить каждое слово в векторном виде таким образом, чтобы косинус угла между векторами слов был тем меньше, чем ближе по смыслу данные слова.

Архитектура модели word2vec основывается на модели skip-gram и состоит из двух матриц: векторных представлений слов (embedding) и матрица контекста (context):



Здесь первая матрица представляет собой веса линейной предсказательной модели и после обучения может использоваться для векторного описания слов.

2.4. Добавление отрицательных примеров

Важно отметить, что обучение на всем имеющемся словаре приводит к необходимости вычислять вероятности для каждого слова (функция softmax). Чтобы этого избежать от задачи многоклассовой классификации переходят к задаче бинарной классификации, где на вход подается слово, для которого необходимо определить соседние к нему слова, а выход – метка, являются ли слова матрицы контекста соседними к нему. Таким образом, задача сводится к задаче классификации на K пересекающихся классов $y \in \{0, 1\}^K$.

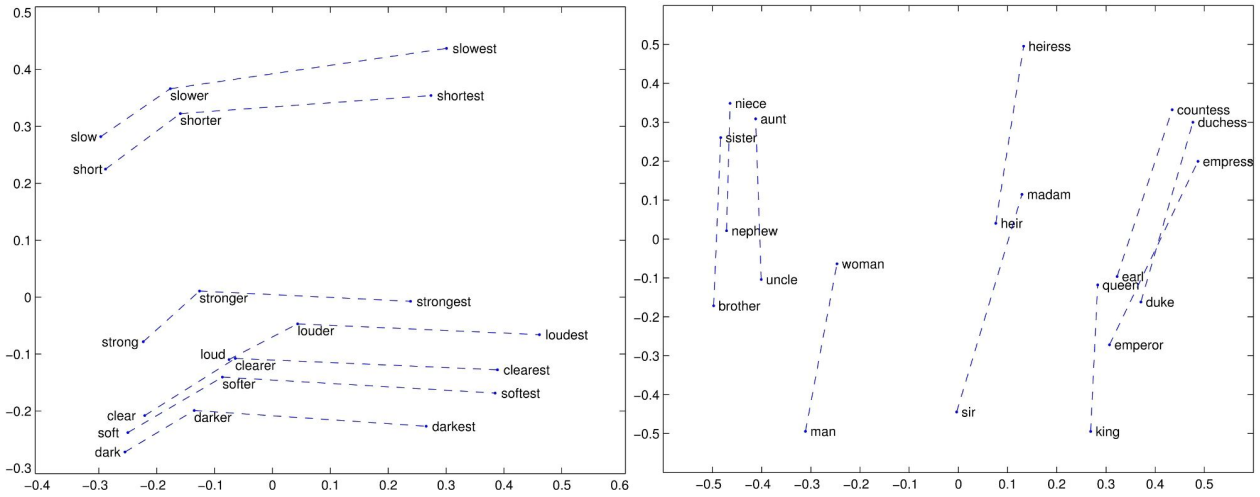
Обучающая выборка содержит исключительно положительные примеры:

input	output	target
the	Success	1
the	is	1
the	ability	1
the	to	1
ability	is	1
ability	the	1
ability	to	1
ability	go	1
to	the	1
to	ability	1
to	go	1
to	from	1
go	ability	1
go	to	1
go	from	1

Чтобы обучить модель классификации необходимо добавить отрицательные примеры, то есть те слова, которые не являются соседними.

input	output	target
the	Success	1
the	is	1
the	ability	1
the		0
the	to	1
ability	is	1
ability	the	1
ability	to	1
ability		0
ability	go	1
to	the	1
to	ability	1
to	go	1
to		0
to	from	1
go	ability	1
go	to	1
go	from	1

В результате обучения модели word2vec каждому из слов будут соответствовать вектора матриц эмбеддингов и контекста, которые оказываются расположенными в пространстве таким образом, чтобы близкие по смыслу слова находились рядом друг с другом.



2.5. Модель fastText

В модели word2vec возможно получить векторное представление слова или предложения, содержащего данное слово, только в том случае, если оно встречалось в обучающей выборке. Модель fastText от Facebook является развитием идеи word2vec и позволяет избежать проблемы отсутствия слов в словаре. Помимо эмбеддингов для каждого слова fastText строит векторное представление для n-грамм символов. Например, для слова *<where>* при $n = 3$ будут рассматриваться следующие n-граммы: *<wh, whe, her, ere, re>*, что позволит при появлении новых слов рассматривать их на символьном уровне. В таком случае векторное представление для нового слова – усредненный вектор n-грамм символов, из которых состоит данное слово.