

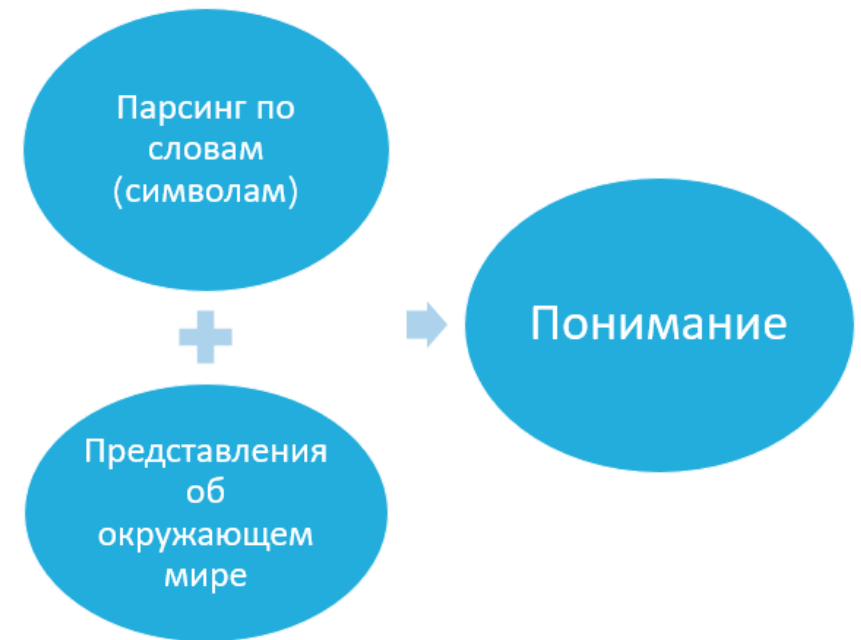
NLP и машинное обучение

Natural Language Processing (NLP)

Обработка естественного языка

Применение алгоритмов машинного обучения для обработки текста и речи:

- системы распознавания речи
- обобщения документов
- машинный перевод
- выявление спама
- распознавание именованных сущностей
- ответы на вопросы
- генерация текста и речи
- автоматический пересказ



Примеры использования NLP

Анализ тональности постов может предоставить много полезной информации о выборе клиентов и факторах, влияющих на их решения

NLP позволяет распознавать и **прогнозировать заболевания** на основе электронной медицинской документации и устной речи пациента.

Фильтрация и классификация электронных писем

Выявлении фейковых новостей

Автоматизация рутинных **судебных задач**

Интеллектуальные голосовые интерфейсы

История NLP

Обработка текста: **Eliza** - чат-бот, разработанный в 1960-х годах и пародирующий диалог с психотерапевтом.

Распознавание речи: в 1970-х годах в Университете Карнеги-Меллона была разработана **Harpy**. Первая компьютерная программа, которая понимала 1000 слов.

Синтез речи: Первым в мире устройством для синтеза речи считается **VODER** (англ. *Voice Operating Demonstrator* — модель голосового аппарата). Оно было разработано Гомером Дадли из компании Bell Labs в 1930-х годах.

Cortana



Виртуальный помощник Windows

Siri

Виртуальный помощник от Apple

Gmail

Почтовый сервис определяет спам

Dialogflow



Платформа от Google, которая позволяет создавать NLP-ботов.
Например, можно сделать бота для заказа пиццы

NLTK

Natural Language Toolkit – ведущая платформа для создания
NLP-программ на Python

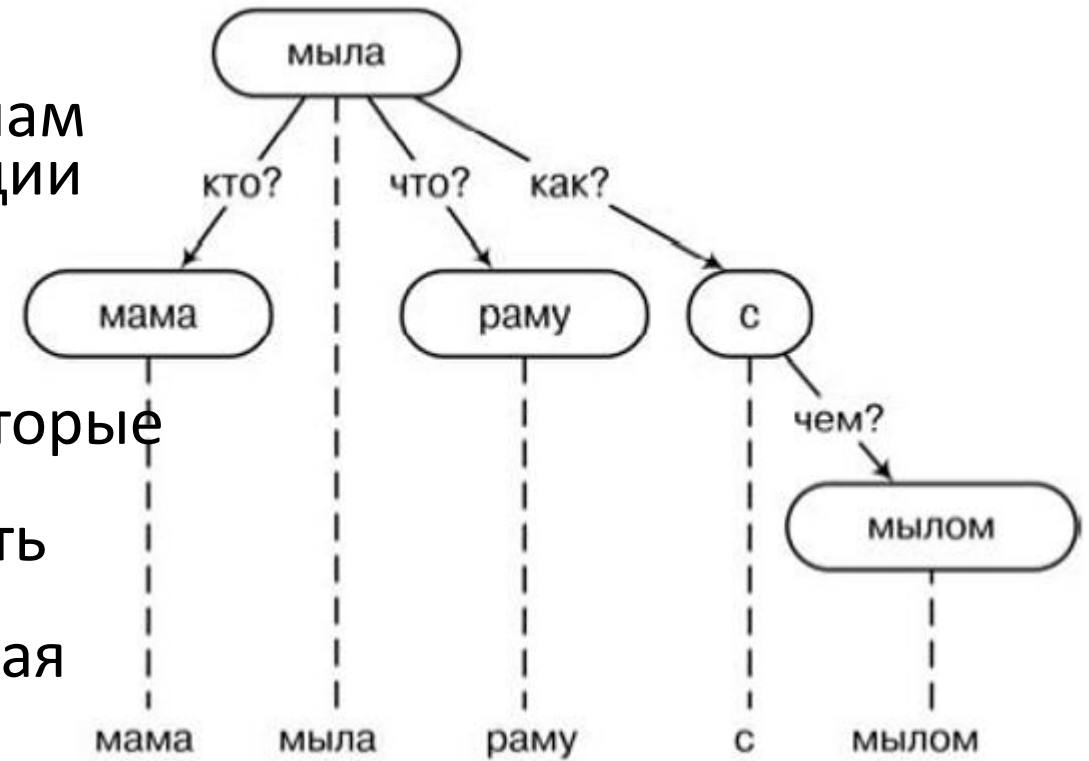
Развитие NLP

Алгоритмы, основанные на правилах и словарях, которые по заданным синтаксическим и семантическим шаблонам искали в текстах определенные конструкции или слова

Дерево синтаксического анализа

Языковые модели - это такие модели, которые для фрагмента текста (в NLP текст это последовательность слов) умеет оценивать вероятность встретить такую последовательность слов в языке. Языковая модель находит наиболее вероятное продолжение для текста.

Моделируется язык при помощи вероятности встретить одни слова рядом с другими



Задачи интеллектуальной обработки текстов

Задачи первого класса (синтаксические)

- Разметка по частям речи, по морфемам, распознавание именованных сущностей

Задачи второго класса (понимание текста)

- Языковые модели, информационный поиск, анализ тональности

Задачи третьего класса (понимание текста и порождение нового)

- Генерация текста, автоматическое реферирование (краткое содержание), диалоговые модели

Основные операции над текстом

Основные термины

Токены

- Текстовые единицы (символы, слова)

Словарь

Документ

- совокупность токенов, которые принадлежат одной смысловой единице (предложение, комментарий)

Корпус

- генеральная совокупность всех документов

Регулярные выражения

это последовательность символов, которая определяет шаблон поиска

Модуль re

- `.` – любой символ, кроме перевода строки;
- `\w` – один символ;
- `\d` – одна цифра;
- `\s` – один пробел;
- `\W` – один НЕсимвол;
- `\D` – одна НЕцифра;
- `\S` – один НЕпробел;
- `[abc]` – находит любой из указанных символов match any of a, b, or c;
- `[^abc]` – находит любой символ, кроме указанных;
- `[a-g]` – находит символ в промежутке от a до g.

Чтение данных

```
1 # Read in the raw text
2 rawData = open("SMSSpamCollection.tsv").read()
3 # Print the raw data
4 rawData[0:250]
```

"ham\tI've been searching for the right words to thank you for this breather. I promise i wont take your help for granted and will fulfil my promise. You have been wonderful and a blessing at all times.\nspam\tFree entry in 2 a wkly comp to win FA Cup final ticket. Please call 0871 230 2235. Please URGENT! 1500 prize guaranteed."

```
1 import pandas as pd
2 # Reading Tab separated Value
3 data = pd.read_csv('SMSSpamCollection.tsv', sep='\t', names=['label', 'body_text'], header=None)
4 # Print first 5 data
5 data.head()
```

	label	body_text
0	ham	I've been searching for the right words to tha...
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...
2	ham	Nah I don't think he goes to usf, he lives aro...
3	ham	Even my brother is not like to speak with me. ...
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!

Удаление пунктуации

```
1 import string
2 string.punctuation
```

```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

```
1 #Function to remove Punctuation
2 def remove_punct(text):
3     text_nopunct = "".join([char for char in text if char not in string.punctuation])# It will discard all punctuations
4     return text_nopunct
5
6 data['body_text_clean'] = data['body_text'].apply(lambda x: remove_punct(x))
7
8 data.head()
```

	label	body_text	body_text_clean
0	ham	I've been searching for the right words to tha...	Ive been searching for the right words to than...
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...
2	ham	Nah I don't think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives aroun...
3	ham	Even my brother is not like to speak with me. ...	Even my brother is not like to speak with me T...
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL

Токенизация по предложениям

процесс разделения письменного языка на предложения-компоненты

по знаку пунктуации - точка

`nltk.sent_tokenize`

Токенизация по словам

- процесс разделения предложений на слова-компоненты
- по знаку пробела
- `nltk.word_tokenize`

Токенизация

```
1 import re
2
3 # Function to Tokenize words
4 def tokenize(text):
5     tokens = re.split('\W+', text) #W+ means that either a word character (A-Za-z0-9_) or a dash (-) can go there.
6     return tokens
7
8 data['body_text_tokenized'] = data['body_text_clean'].apply(lambda x: tokenize(x.lower()))
9 #We convert to lower as Python is case-sensitive.
10
11 data.head()
```

	label	body_text	body_text_clean	body_text_tokenized
0	ham	I've been searching for the right words to tha...	Ive been searching for the right words to than...	[ive, been, searching, for, the, right, words, ...]
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...	[free, entry, in, 2, a, wkly, comp, to, win, f...
2	ham	Nah I don't think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives aroun...	[nah, i, dont, think, he, goes, to, usf, he, l...
3	ham	Even my brother is not like to speak with me. ...	Even my brother is not like to speak with me T...	[even, my, brother, is, not, like, to, speak, ...]
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL	[i, have, a, date, on, sunday, with, will]

Стоп - слова

Это слова, которые выкидываются из текста до/после обработки текста.

Такие слова могут добавить много шума, поэтому необходимо избавляться от нерелевантных слов.

Стоп-слова это обычно понимают артикли, междометия, союзы и т.д., которые не несут смысловой нагрузки.

```
nltk.download("stopwords")
```


Удаление стоп-слов

```
1 import nltk
2
3 stopword = nltk.corpus.stopwords.words('english')# All English Stopwords

1 # Function to remove Stopwords
2 def remove_stopwords(tokenized_list):
3     text = [word for word in tokenized_list if word not in stopword]# To remove all stopwords
4     return text
5
6 data['body_text_nostop'] = data['body_text_tokenized'].apply(lambda x: remove_stopwords(x))
7
8 data.head()
```

	label	body_text	body_text_clean	body_text_tokenized	body_text_nostop
0	ham	I've been searching for the right words to tha...	Ive been searching for the right words to than...	[ive, been, searching, for, the, right, words,...	[ive, searching, right, words, thank, breather...
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...	[free, entry, in, 2, a, wkly, comp, to, win, f...	[free, entry, 2, wkly, comp, win, fa, cup, fin...
2	ham	Nah I don't think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives around...	[nah, i, dont, think, he, goes, to, usf, he, l...	[nah, dont, think, goes, usf, lives, around, t...
3	ham	Even my brother is not like to speak with me. ...	Even my brother is not like to speak with me T...	[even, my, brother, is, not, like, to, speak, ...	[even, brother, like, speak, treat, like, aids...
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL	[i, have, a, date, on, sunday, with, will]	[date, sunday]

Лемматизация и стемминг текста

Привести все встречающиеся словоформы к одной, нормальной словарной форме.

Частные случаи нормализации

```
dog, dogs, dog's, dogs' => dog
```

Стемминг – это грубый эвристический процесс, который отрезает «лишнее» от корня слов, часто это приводит к потере словообразовательных суффиксов.

Лемматизация – это более тонкий процесс, который использует словарь и морфологический анализ, чтобы в итоге привести слово к его канонической форме – лемме.

Стемминг

```
1 ps = nltk.PorterStemmer()
2
3 def stemming(tokenized_text):
4     text = [ps.stem(word) for word in tokenized_text]
5     return text
6
7 data['body_text_stemmed'] = data['body_text_nostop'].apply(lambda x: stemming(x))
8
9 data.head()
```

	label	body_text	body_text_clean	body_text_tokenized	body_text_nostop	body_text_stemmed
0	ham	I've been searching for the right words to tha...	Ive been searching for the right words to than...	[ive, been, searching, for, the, right, words,...	[ive, searching, right, words, thank, breather...	[ive, search, right, word, thank, breather, pr...
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...	[free, entry, in, 2, a, wkly, comp, to, win, f...	[free, entry, 2, wkly, comp, win, fa, cup, fin...	[free, entri, 2, wkli, comp, win, fa, cup, fin...
2	ham	Nah I don't think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives aroun...	[nah, i, dont, think, he, goes, to, usf, he, l...	[nah, dont, think, goes, usf, lives, around, t...	[nah, dont, think, goe, usf, live, around, tho...
3	ham	Even my brother is not like to speak with me. ...	Even my brother is not like to speak with me T...	[even, my, brother, is, not, like, to, speak, ...	[even, brother, like, speak, treat, like, aids...	[even, brother, like, speak, treat, like, aid,...
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL	[i, have, a, date, on, sunday, with, will]	[date, sunday]	[date, sunday]

Лемматизация

```

1 wn = nltk.WordNetLemmatizer()
2
3 def lemmatizing(tokenized_text):
4     text = [wn.lemmatize(word) for word in tokenized_text]
5     return text
6
7 data['body_text_lemmatized'] = data['body_text_nostop'].apply(lambda x: lemmatizing(x))
8
9 data.head(10)

```

	label	body_text	body_text_clean	body_text_tokenized	body_text_nostop	body_text_stemmed	body_text_lemmatized
0	ham	I've been searching for the right words to tha...	Ive been searching for the right words to than...	[ive, been, searching, for, the, right, words, ...	[ive, searching, right, words, thank, breather...	[ive, search, right, word, thank, breather, pr...	[ive, searching, right, word, thank, breather, ...
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...	[free, entry, in, 2, a, wkly, comp, to, win, f...	[free, entry, 2, wkly, comp, win, fa, cup, fin...	[free, entri, 2, wkli, comp, win, fa, cup, fin...	[free, entry, 2, wkly, comp, win, fa, cup, fin...
2	ham	Nah I don't think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives aroun...	[nah, i, dont, think, he, goes, to, usf, he, l...	[nah, dont, think, goes, usf, lives, around, t...	[nah, dont, think, goe, usf, live, around, tho...	[nah, dont, think, go, usf, life, around, though]
3	ham	Even my brother is not like to speak with me. ...	Even my brother is not like to speak with me T...	[even, my, brother, is, not, like, to, speak, ...	[even, brother, like, speak, treat, like, aids...	[even, brother, like, speak, treat, like, aid,...	[even, brother, like, speak, treat, like, aid,...
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL	[i, have, a, date, on, sunday, with, will]	[date, sunday]	[date, sunday]	[date, sunday]
5	ham	As per your request 'Melle Melle (Oru Minnamin...	As per your request Melle Melle Oru Minnaminun...	[as, per, your, request, melle, melle, oru, mi...	[per, request, melle, melle, oru, minnaminungi...	[per, request, mell, mell, oru, minnaminungint...	[per, request, melle, melle, oru, minnaminungi...

Векторизация текста

Прямое кодирование (one-hot encoding)

- 1.каждый токен представляет бинарный вектор (значения 0 или 1);
- 2.единица ставится тому элементу, который соответствует номеру токена в словаре.

```
{Пес, Кот, ель, на, сел, пень} # Словарь
# Первый документ
[[1, 0, 0, 0, 0, 0] # Пес
[0, 0, 0, 0, 0, 0] # Кот (нет в предложении)
[0, 0, 0, 0, 0, 0] # ель (нет в предложении)
[0, 0, 0, 1, 0, 0] # на
[0, 0, 0, 0, 1, 0] # сел
[0, 0, 0, 0, 0, 1]] # пень
# Второй документ
[[0, 0, 0, 0, 0, 0] # Пес (нет в предложении)
[0, 1, 0, 0, 0, 0] # Кот
[0, 0, 1, 0, 0, 0] # ель
[0, 0, 0, 1, 0, 0] # на
[0, 0, 0, 0, 1, 0] # сел
[0, 0, 0, 0, 0, 0]] # пень (нет в предложении)
```

```
{Пес, Кот, ель, на, сел, пень} # Словарь
```

```
[Пес, сел, на, пень] # Первый документ
[Кот, сел, на, ель] # Второй документ
[[Пес, сел, на, пень], [Кот, сел, на, ель]] # Корпус
```

Мешок слов (bag of words)

Выделяет вектору весь документ, и каждый элемент кодируется 1 по порядку следования слов в словаре

Метод не учитывает важность того или иного токена, ведь одно слово может повторяться по несколько раз.

```
{Пес, Кот, ель, на, сел, пенъ} # Словарь  
# Корпус:  
[[1, 0, 0, 1, 1, 1] # Первый документ  
 [0, 1, 1, 1, 1, 0]] # Второй документ
```

Теряется вся информация о взаимном расположении слов внутри текста

```
[Пес, сел, на, пенъ] # Первый документ  
[Кот, сел, на, ель] # Второй документ  
[[Пес, сел, на, пенъ], [Кот, сел, на, ель]] # Корпус
```

TF-IDF

Мера оценки важности слова в контексте документа

Term Frequency (частотность слова в документе)

Inverse Document Frequency (инверсия частоты документа)

$$TF_{token_i} = \frac{n_i}{N_i},$$

$$TF-IDF = TF \times IDF$$

$$IDF_{token} = \log \frac{P}{p},$$

где n_i — сколько раз встречается токен в i -ом документе,

N_i — общее количество токенов в i -ом документе,

p — количество документов, в которых встречается токен,

P — общее количество документов.

Токен	TF		IDF	TF-IDF	
	Док. №1	Док. №2		Док. №1	Док. №2
Кот	0	$\frac{1}{4}$	$\log \frac{2}{1}$	0	0.17
Пес	$\frac{1}{4}$	0	$\log \frac{2}{1}$	0.17	0
ель	0	$\frac{1}{4}$	$\log \frac{2}{1}$	0	0.17
на	$\frac{1}{4}$	$\frac{1}{4}$	$\log \frac{2}{2} = 0$	0	0
сел	$\frac{1}{4}$	$\frac{1}{4}$	$\log \frac{2}{2} = 0$	0	0
пень	$\frac{1}{4}$	0	$\log \frac{2}{1}$	0.17	0

редкие слова и слова, которые встречаются во всех документах, несут мало информации

Проблемы

- не зависят от контекста – например, оба анализируемых предложения отражают примерно одно и то же: “что-то куда-то село”;
- не учитывают порядок слов в предложении;
- обладают высокой размерностью в случае большого словаря, что может снизить производительность модели глубокого обучения.

Word embeddings – векторное представление слов

Word2vec - 2013

Слова, которые встречаются в одинаковых окружениях, имеют близкие значения.

Предсказывать вероятность слова по его окружению (контексту).

Вероятность, присваиваемая моделью слову близка к вероятности встретить это слово в этом окружении в реальном тексте.

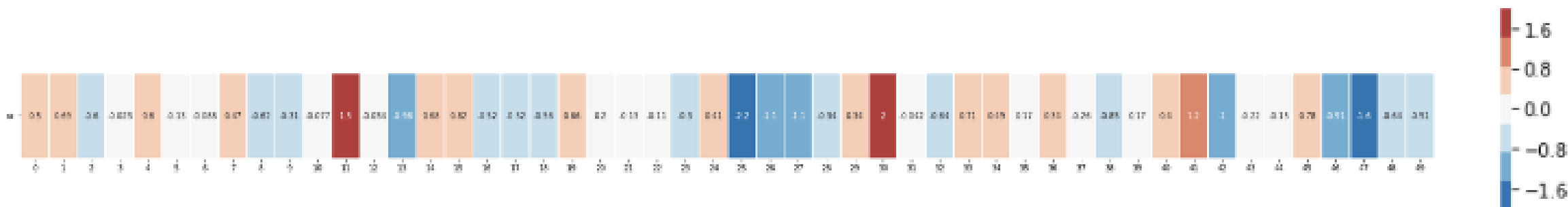


Модель обработки естественного языка на основе вложений

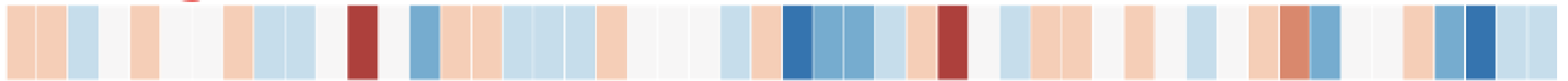
Слово «Король»

вектор GloVe, обученный на Википедии

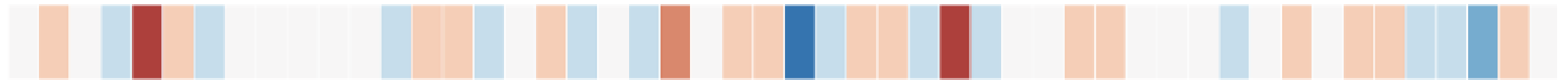
```
[ 0.50451 , 0.68607 , -0.59517 , -0.022801, 0.60046 , -0.13498 , -0.08813 , 0.47377  
 , -0.61798 , -0.31012 , -0.076666, 1.493 , -0.034189, -0.98173 , 0.68229 , 0.81722 ,  
 -0.51874 , -0.31503 , -0.55809 , 0.66421 , 0.1961 , -0.13495 , -0.11476 , -0.30344 ,  
 0.41177 , -2.223 , -1.0756 , -1.0783 , -0.34354 , 0.33505 , 1.9927 , -0.04234 ,  
 -0.64319 , 0.71125 , 0.49159 , 0.16754 , 0.34344 , -0.25663 , -0.8523 , 0.1661 ,  
 0.40102 , 1.1685 , -1.0137 , -0.21585 , -0.15155 , 0.78321 , -0.91241 , -1.6106 ,  
 -0.64426 , -0.51042 ]
```



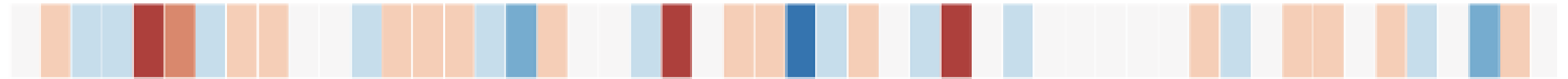
“king”



“Man”



“Woman”



queen

woman

girl

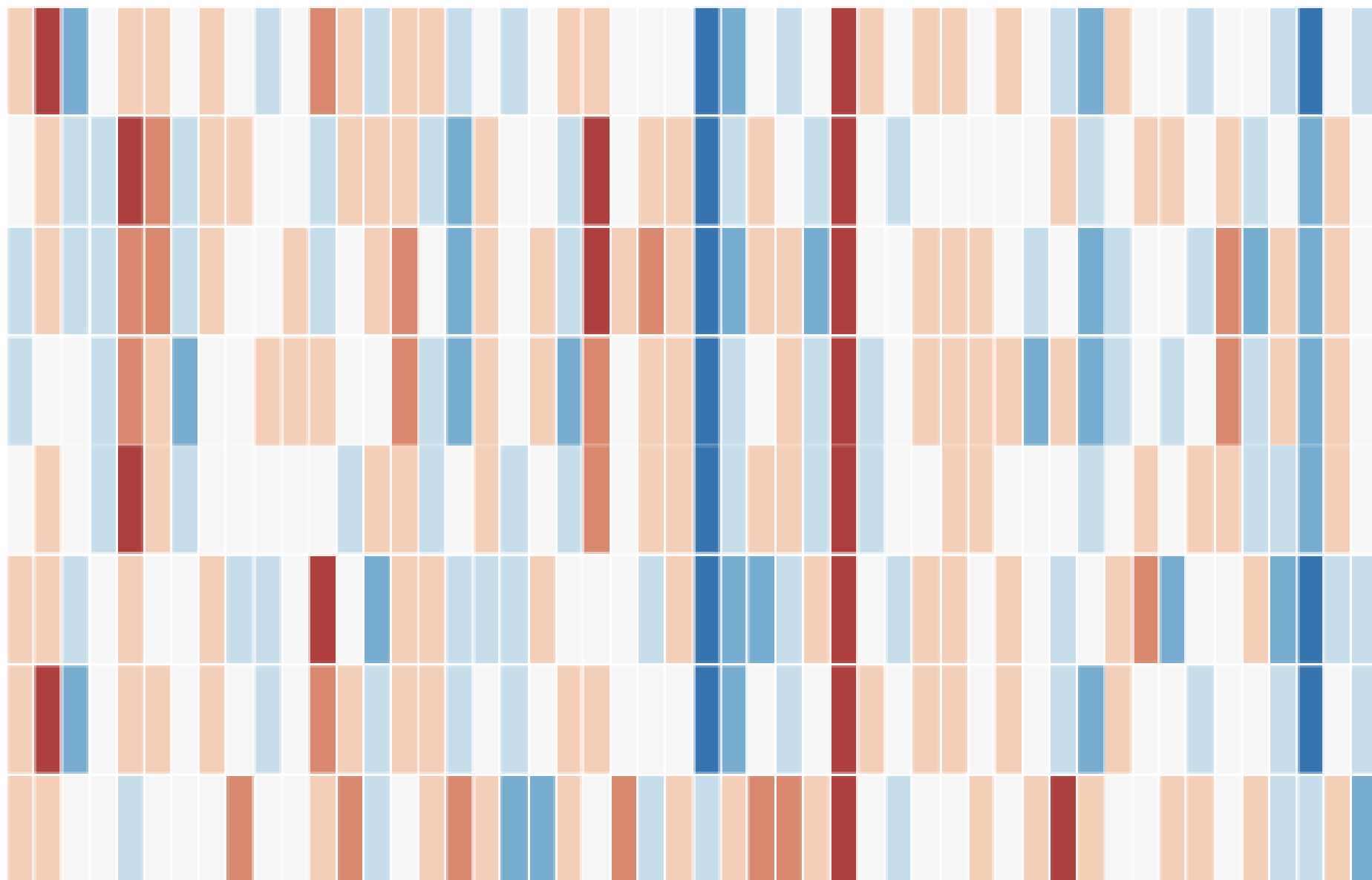
boy

man

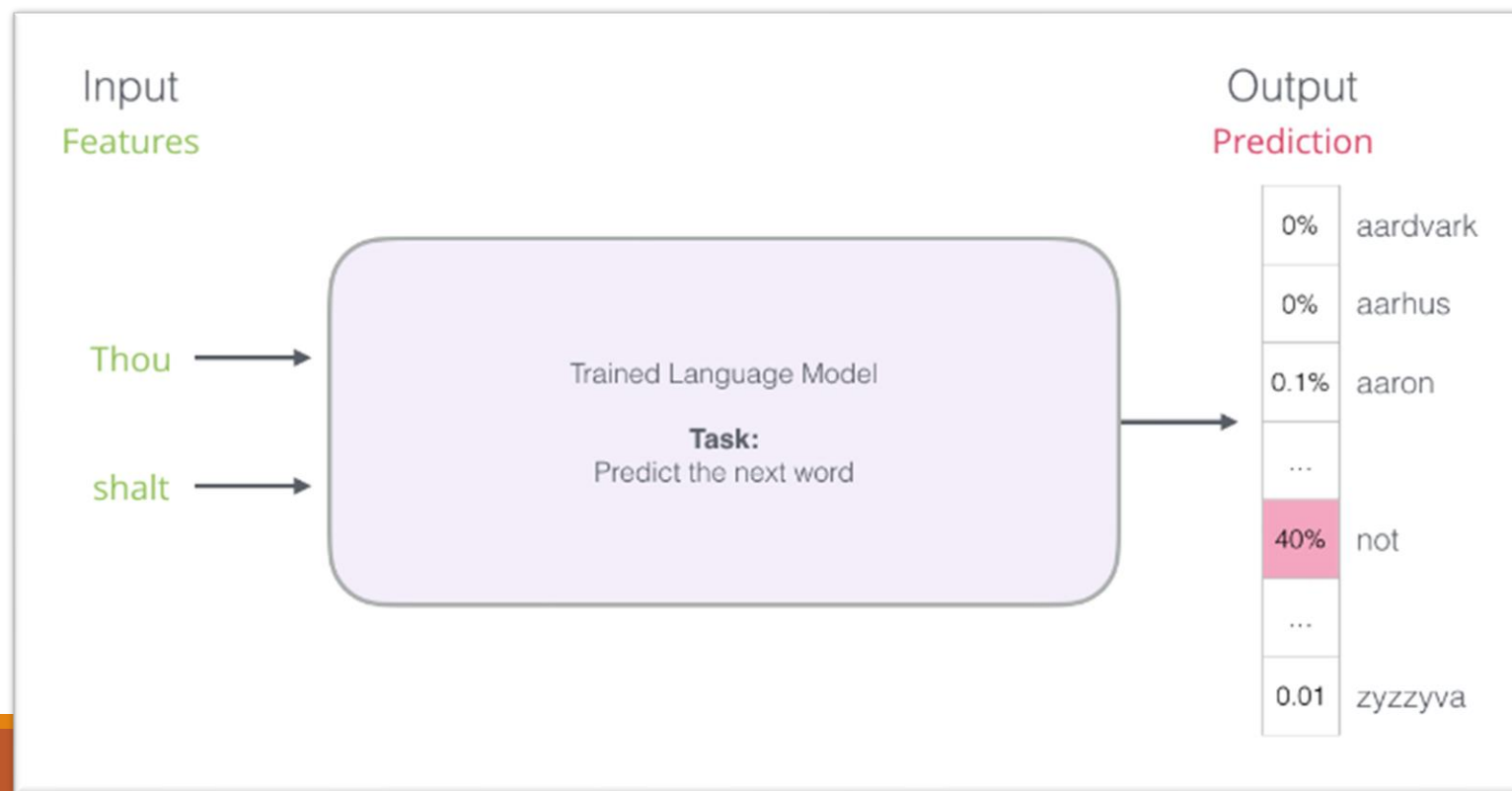
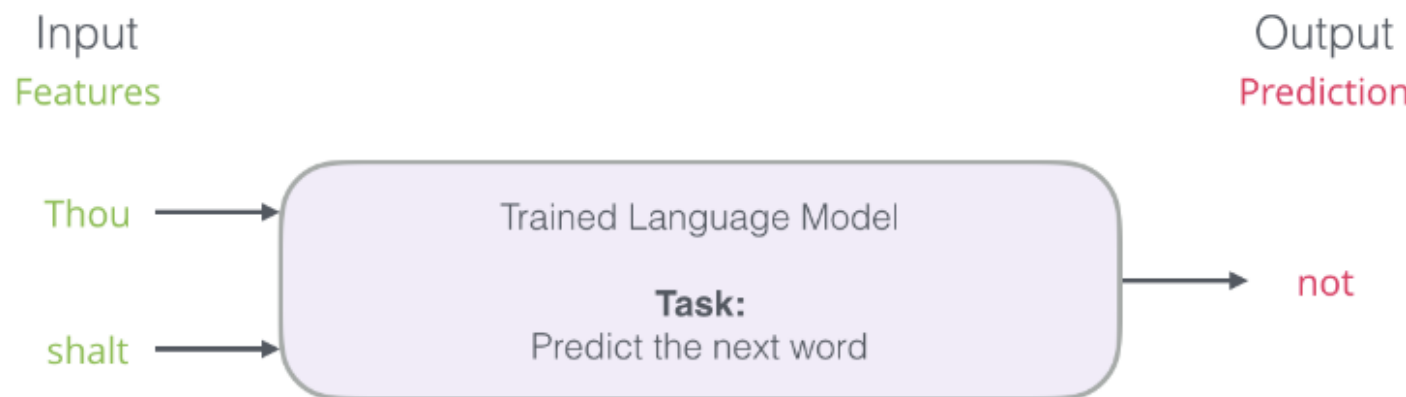
king

queen

water



Модель языка: предсказание следующего слова



Обучение модели языка: скользящее окно

Thou shalt not make a machine in the likeness of a human mind

Sliding window across running text

thou	shalt	not	make	a	machine	in	the	...
------	-------	-----	------	---	---------	----	-----	-----

Dataset

input 1	input 2	output
thou	shalt	not

Thou shalt not make a machine in the likeness of a human mind

Sliding window across running text

thou	shalt	not	make	a	machine	in	the	...
thou	shalt	not	make	a	machine	in	the	

Dataset

input 1	input 2	output
thou	shalt	not
shalt	not	make

CBOW (Continuous Bag of Words)

Jay was hit by a _____ bus in...

by	a	red	bus	in
----	---	-----	-----	----

input 1	input 2	input 3	input 4	output
by	a	bus	in	red

Skip-gram

Thou shalt not make a machine in the likeness of a human mind

thou	shalt	not	make	a	machine	in	the	...
------	-------	-----	------	---	---------	----	-----	-----

Thou shalt not make a machine in the likeness of a human mind

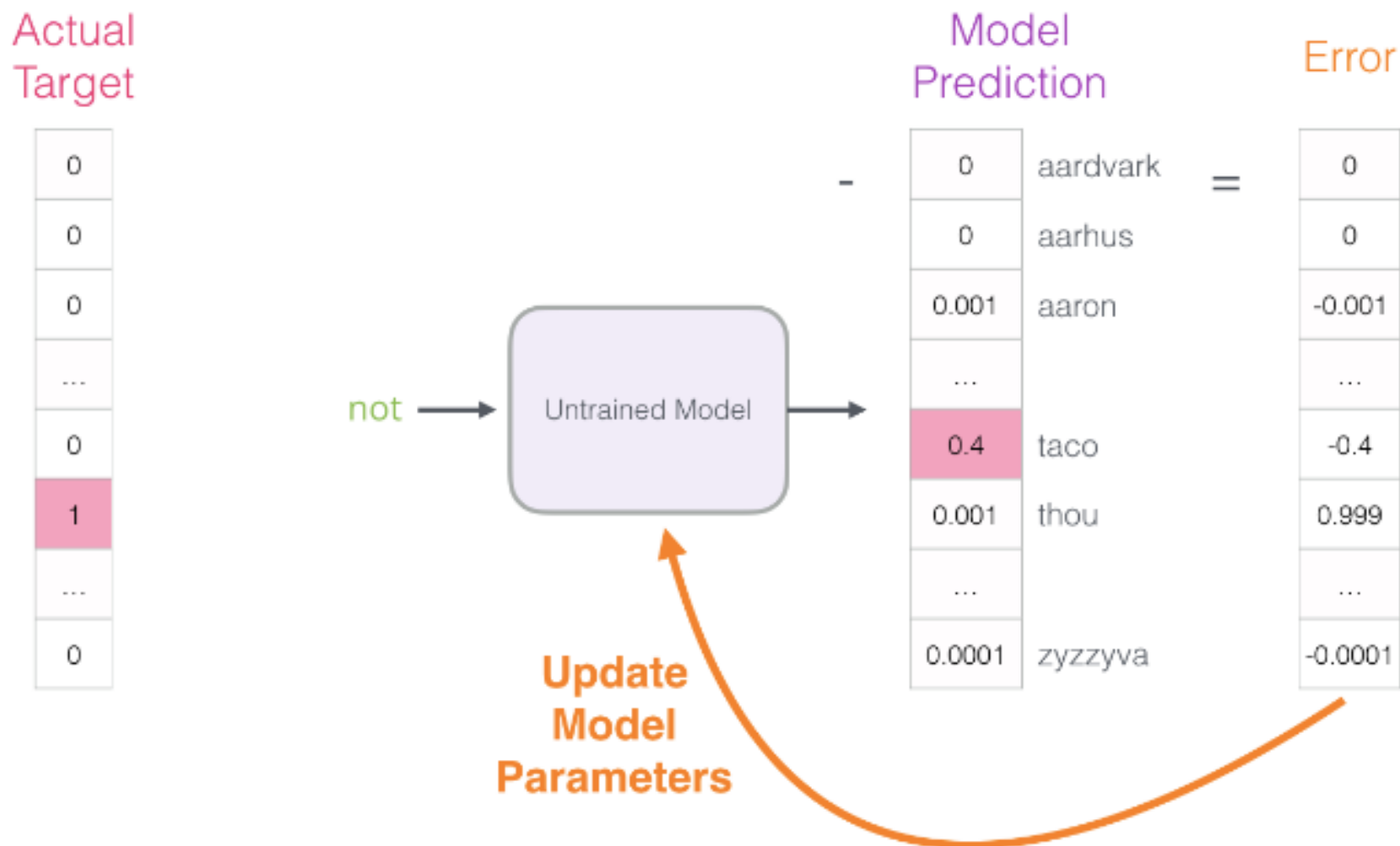
thou	shalt	not	make	a	machine	in	the	...
------	-------	-----	------	---	---------	----	-----	-----

thou	shalt	not	make	a	machine	in	the	...
------	-------	-----	------	---	---------	----	-----	-----

input word	target word
not	thou
not	shalt
not	make
not	a
make	shalt
make	not
make	a
make	machine

input word	target word
not	thou
not	shalt
not	make
not	a

Обучение модели языка



Skip-gram с отрицательной выборкой (SGNS)

Skipgram

shalt	not	make	a	machine
input		output		
make		shalt		
make		not		
make		a		
make		machine		

Negative Sampling

input word	output word	target
make	shalt	1
make	aaron	0
make	taco	0

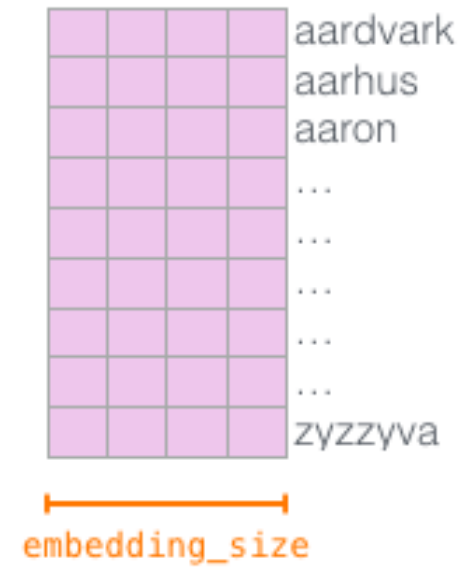
вероятность их соседства
(от 0 до 1)

word2vec

Embedding



Context



word2vec

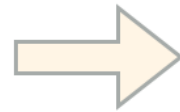
Embedding

				aardvark
				aarhus
				aaron
				...
				not
				...
				...
				...
				zyzzyva

Context

				aardvark
				aarhus
				aaron
				...
				taco
				...
				thou
				...
				zyzzyva

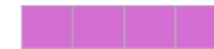
Look up
embeddings



not



aaron



taco

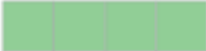


thou



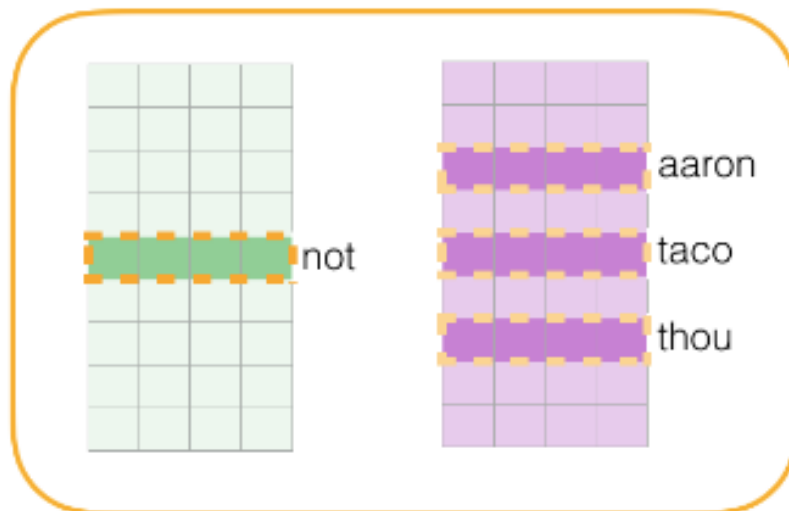
input word	output word	target	input • output	sigmoid()
not 	thou 	1	0.2	0.55
not 	aaron 	0	-1.11	0.25
not 	taco 	0	0.74	0.68

word2vec

input word	output word	target	input • output	sigmoid()	Error
not 	thou 	1	0.2	0.55	0.45
not 	aaron 	0	-1.11	0.25	-0.25
not 	taco 	0	0.74	0.68	-0.68

Размер окна

Количество
отрицательных образцов



Update
Model
Parameters

Сети на базе N-грамм и символов

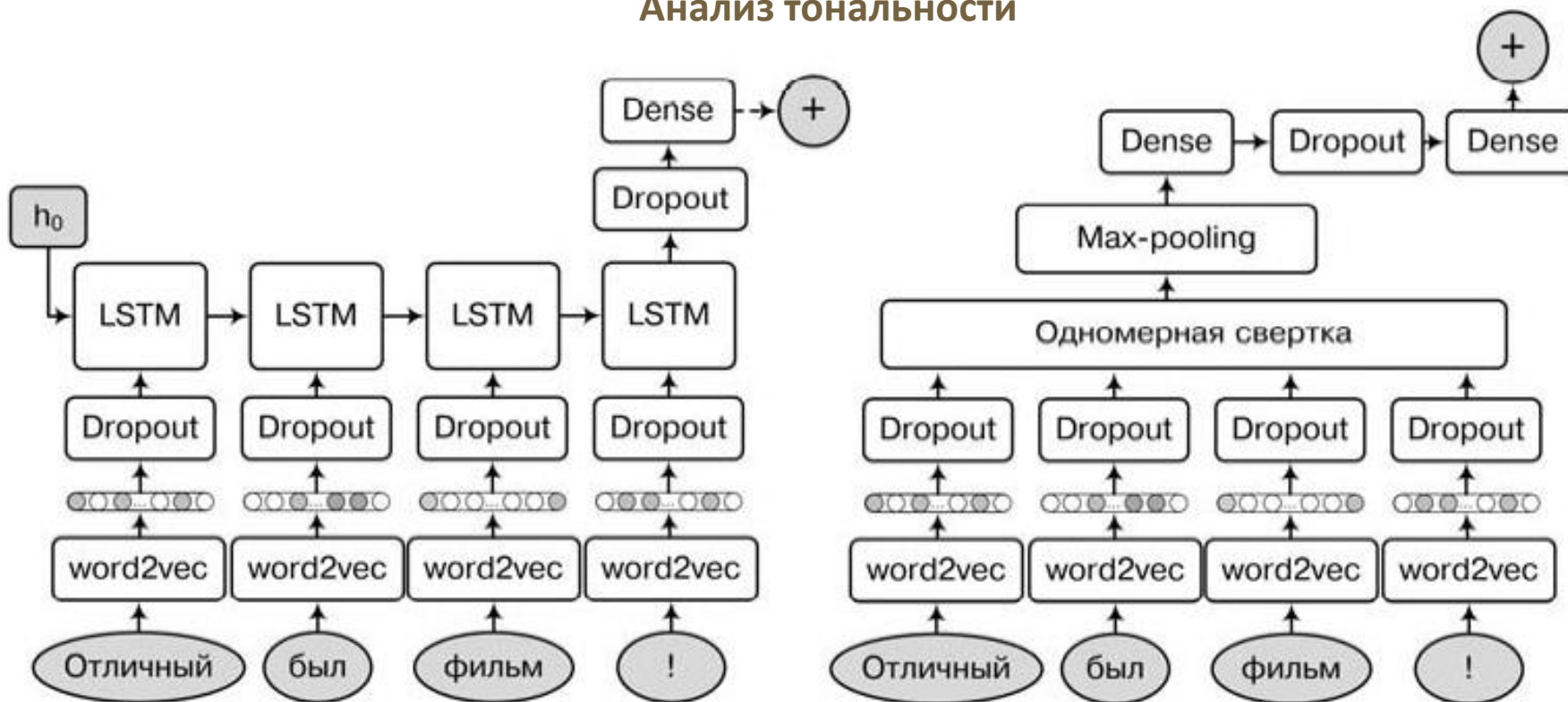
Построение входа нейронной сети на базе слов — слов может быть много, в них порой содержатся ошибки и опечатки. **Векторные** представления **слов-зашумлены**.

N-граммы (последовательностями из нескольких, как правило, трёх символов).

Нейронные сети

В какой-то момент НС будут представлять короткие участки текста в виде векторов фиксированной размерности.

Анализ тональности



Определение смысла крупных участков текста: предложений, абзацев, статей

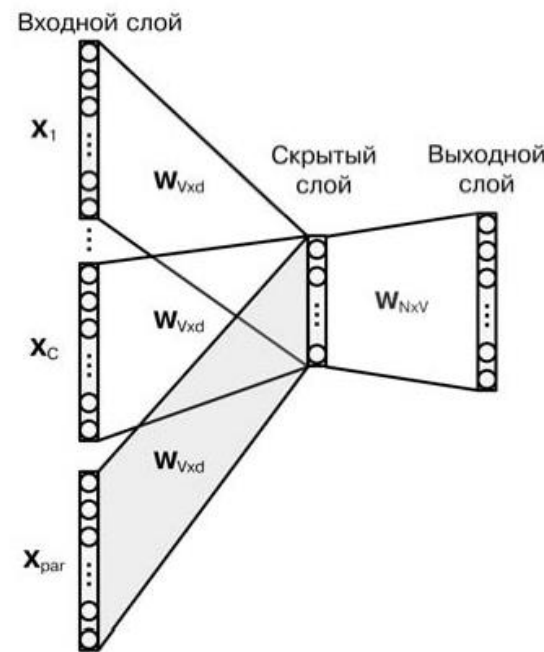
Объединение векторов отдельных слов в векторы нескольких слов:

Сумма или среднее (резюмирование документов)

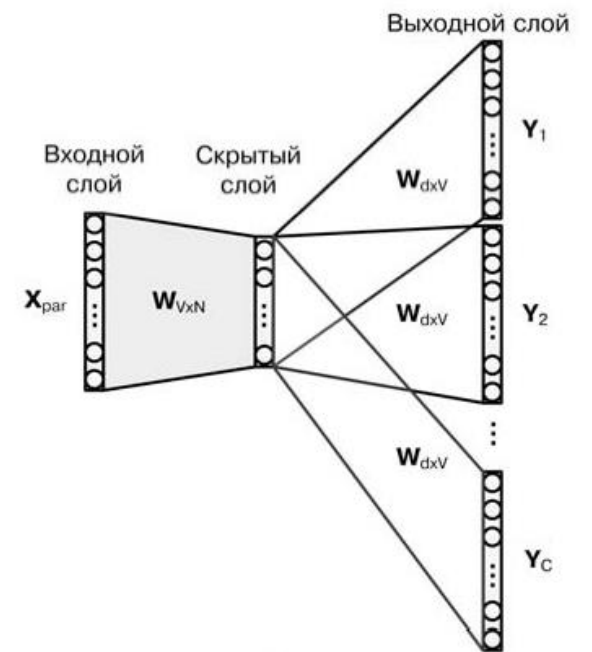
Обучение векторов слов

Distributed Memory Model of Paragraph Vectors (вектор, представляющий абзац или предложение, определяется как еще один дополнительный вектор весов)

Distributed Bag of Words Model of Paragraph Vectors (слова контекста просто полностью игнорируются, и модель использует вектор абзаца, чтобы предсказать слова из окна в том же абзаце)



а



б