Выявление поляризации и нейтральности текстов в новостном потоке

A Preprint

Роман Авдеев
Студент 4 курса 417 гр.
Факультет ВМК
Кафедра ММП
МГУ имени М. В. Ломоносова
roma.avdeyev@gmail.com

Константин Вячеславович Воронцов
Профессор РАН, д.ф.-м.н., проф., зав.каф.ММП
Факультет ВМК
Кафедра ММП
МГУ имени М. В. Ломоносова

Abstract

В данной работе предлагается способ определения поляризации текстов в новостном потоке. Решение основано на методах машинного обучения без учителя, что позволяет работать как с малыми, так и с большими наборами текстов. Решается задача разделения множества новостных сообщений на кластеры-мнения, выделения отдельных кластеров нейтральных и нерелевантных документов. Предложены метрики оценивания качества отсева нерелевантных и нейтральных сообщений.
Эксперименты проводились на датасете, состоящем из 30 корпусов новостных сообщений по темам «политика» и «происшествия».

Keywords Поляризация · sentiment analysis · opinion mining · нейтральность

1 Введение

Задача кластеризации текстов, то есть разбиения множества документов на подмножества тематически близких, остается актуальной на протяжении ряда последних лет. Знания в этой области полезны не только для того, чтобы корректно идентифицировать мнения клиентов в отзывах на какой-либо продукт, но и для более сложных задач. Например, для сбора общественного мнения в рамках анализа ценообразования, конкурентной разведки, прогнозирования рынка, прогнозирования выборов и выявления рисков в банковских системах.

Решаемая мной проблема относится к областям Opinion Mining и Sentiment Analysis. В задачах первой упомянутой области происходит поиск мнений пользователей о товарах и других объектах, а также производится анализ этих мнений. Искомая совокупность мнений и называется поляризацией. Вторая область близка к классической задаче контент-анализа текстов массовой коммуникации, в ней оценивается общая тональность высказываний и текста в целом.
Существуют разные подходы к выявлению поляризации мнений в тексте. Многие из них основываются на работе нейронных сетей ((10)). Также существует подход, при котором используется обучение с учителем, что не всегда является оптимальным из-за разного размера текстов и объема обучающей выборки в целом. Работы (1) и (3) базировались на тематическом моделировании. Их авторы используют такие модальности как SPO триплеты (subject-predicate-object), семанические роли (по Филлмору), тональность слов, социально-демографические показатели. Также было показано, что включение семантической близости текстов в модель улучшает качество, а социально-демографические показатели не вносят особого вклада.

В данной работе учтен опыт предыдущих работ, проведены эксперименты со структурой модели, с разными алгоритмами. Предложены метрики оценивания различных аспектов качества модели.

2 Данные

Используемый датасет представляет собой набор из 30-ти корпусов новостных сообщений из рубрик «Политика» и «Происшествия». В каждом корпусе от 8-ми до 33-х документов. Всего 452 документа.
Разметкой будем называть набор меток $X = \{x_1, ..., x_n\}$, каждый элемент которого x_i является

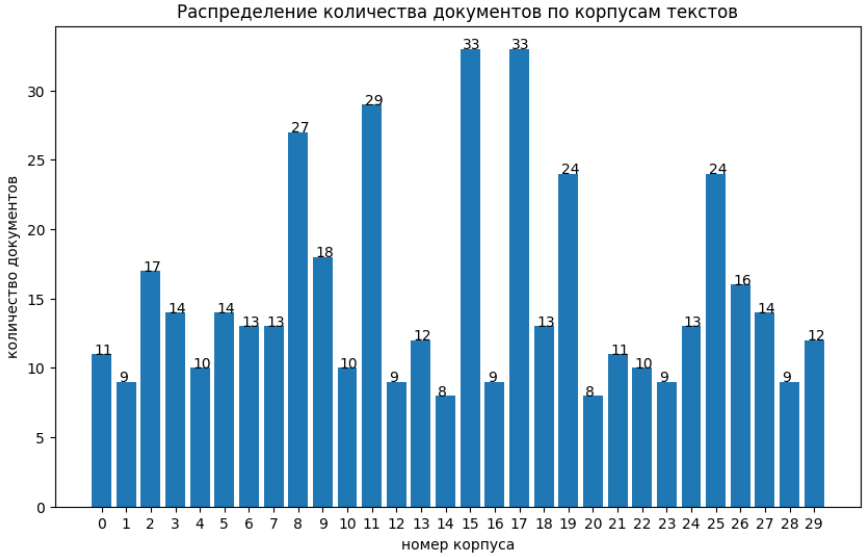

Рис. 1: Количество документов в каждом корпусе

меткой i-го сообщения в корпусе. Предполагается, что каждый корпус содержит сообщения об одном событии или теме, однако может содержать и посторонние сообщения. Метки $\{1, 2, 3, ...\}$ соответствуют кластерам-полюсам общественного мнения по данной теме. Метка <0> означает, что сообщение является нейтральным, то есть не содержит субъективного мнения. Метка <-1> означает, что документ нерелевантен, то есть не относится к общей для всех сообщений теме.

Для разметки данного датасета использовался сервис Яндекс.Толока. Маркировка осуществлялась асессорами по инструкции, разработанной Е.Милютой в работе (1).

2.1 Анализ разметки

Нерелевантные документы
Рассмотрим документы, которые были промаркированы асессорами как неотносящиеся к теме (нерелевантные):

	Разметка 1	Разметка 2	Разметка 3
Кол-во нерелевантных док-ов	31	59	38
В процентах (%) (от общего размера датасета)	6.9 %	13 %	8.4 %

В таблице ниже приведено распределение нерелевантных документов по корпусам для каждой разметки. Каждой строке соответствует один из 30-ти корпусов. Метка строки - имя корпуса в датасете.
Обозначения:

1. `docs_amount` - количество документов в корпусе

2. `amount_i` - количество нерелевантных документов в корпусе для i-ой разметки ($i \in \{1, 2, 3\}$)
3. `percent_i` - процентное содержание нерелевантных документов в корпусе для i-ой разметки

	docs_amount	amount_1	percent_1	amount_2	percent_2	amount_3	percent_3
17af2621-05bd-4008-a8f5-4afe64a4e823	11	0	0.0	0	0.0	4	36.36
1bcad992-3329-4807-8ac5-920aee3c0db8	9	0	0.0	0	0.0	0	0.0
28db5916-80dc-42f9-a822-5f14259b0e14	17	0	0.0	0	0.0	0	0.0
3975555d-2f8b-48a4-85ed-7ce390c04618	14	0	0.0	5	35.71	0	0.0
3af7ccba-dc12-4d24-8271-298204826729	10	0	0.0	0	0.0	0	0.0
3c2da57c-c42f-4897-b5e7-7b44855baae6	14	1	7.14	1	7.14	1	7.14
42f49e66-6d17-4869-adb4-3b0ebf51fc87	13	1	7.69	9	69.23	0	0.0
46509952-8146-4262-9bb0-d77d94dab5ef	13	7	53.85	7	53.85	7	53.85
48b345a1-0d3d-4df8-ae7a-862d55f16fde	27	1	3.7	1	3.7	0	0.0
4ab6d1a4-f530-451c-bcfa-852c7784e9f6	18	0	0.0	0	0.0	0	0.0
4cd6fbbd-3123-4d87-9a2e-a0ce0eae933c	10	3	30.0	3	30.0	2	20.0
50ae73b5-3dd0-4298-b04b-aa0496000eb0	29	0	0.0	0	0.0	0	0.0
5157cb93-7a0e-4485-a715-fa36d3ca94cb	9	2	22.22	1	11.11	0	0.0
51e55b2f-611c-48f3-8310-8fb12bfaab94	12	5	41.67	0	0.0	0	0.0
581ffa03-7310-460e-8ce0-99af18046b73	8	1	12.5	0	0.0	1	12.5
5c24dfbd-f3b5-4608-892f-80cd2ec82f27	33	0	0.0	0	0.0	0	0.0
5da72401-3267-44e6-a133-095988933987	9	0	0.0	1	11.11	0	0.0
6415b59f-a6cf-4789-93cb-98024289ba74	33	0	0.0	18	54.55	1	3.03
6d98268c-df5e-4700-8192-c2633f5fbd99	13	0	0.0	0	0.0	2	15.38
77085a14-c648-4207-a555-d7d3987400fd	24	0	0.0	0	0.0	0	0.0
817d0fce-bb03-4d80-8da1-190999093b2f	8	2	25.0	2	25.0	2	25.0
905d170a-569b-4ee9-ae32-86bdb33f7309	11	3	27.27	6	54.55	2	18.18
9deaeb05-f44b-4c3b-83e7-c44fbe762b4c	10	0	0.0	0	0.0	0	0.0
a2defc9b-ba3b-4f3e-94b7-f07e72cae431	9	5	55.56	5	55.56	5	55.56
bb470657-d3a9-4c90-b7f4-95beaead5ae7	13	0	0.0	0	0.0	0	0.0
c80e8aa3-ca56-44c1-8ed7-868d2c78a52c	24	0	0.0	0	0.0	0	0.0
d8e5332d-7da1-491c-bf3a-248b10a56c30	16	0	0.0	0	0.0	0	0.0
dceb2da8-3a81-4210-80b5-c6ed1b682acf	14	0	0.0	0	0.0	11	78.57
ec2e2478-8239-4a2a-9f58-a258529fdd3c	9	0	0.0	0	0.0	0	0.0
f3fcb2bf-44ec-4c68-bce2-23bde9e14ebd	12	0	0.0	0	0.0	0	0.0

Рис. 2: Распределение нерелевантных документов по корпусам

Нейтральные документы
Рассмотрим документы, которые были промаркированы асессорами как невыражающие субъективного отношения к событиям - нейтральными:

	Разметка 1	Разметка 2	Разметка 3
Кол-во нейтральных док-ов	110	55	73
В процентах (%) (от общего размера датасета)	24.3 %	12.2 %	16.2 %

В таблице ниже приведено распределение нейтральных документов по корпусам для каждой разметки. Каждой строке соответствует один из 30-ти корпусов. Метка строки - имя корпуса в датасете. Обозначения:

1. `docs_amount` - количество документов в корпусе
2. `amount_i` - количество нейтральных документов в корпусе для i-ой разметки ($i \in \{1, 2, 3\}$)

	docs_amount	amount_1	percent_1	amount_2	percent_2	amount_3	percent_3
17af2621-05bd-4008-a8f5-4afe64a4e823	11	1	9.09	11	100.0	0	0.0
1bcad992-3329-4807-8ac5-920aee3c0db8	9	9	100.0	0	0.0	2	22.22
28db5916-80dc-42f9-a822-5f14259b0e14	17	0	0.0	0	0.0	12	70.59
3975555d-2f8b-48a4-85ed-7ce390c04618	14	14	100.0	0	0.0	0	0.0
3af7ccba-dc12-4d24-8271-298204826729	10	0	0.0	0	0.0	9	90.0
3c2da57c-c42f-4897-b5e7-7b44855baae6	14	0	0.0	0	0.0	1	7.14
42f49e66-6d17-4869-adb4-3b0ebf51fc87	13	0	0.0	0	0.0	0	0.0
46509952-8146-4262-9bb0-d77d94dab5ef	13	0	0.0	0	0.0	0	0.0
48b345a1-0d3d-4df8-ae7a-862d55f16fde	27	0	0.0	0	0.0	2	7.41
4ab6d1a4-f530-451c-bcfa-852c7784e9f6	18	18	100.0	0	0.0	0	0.0
4cd6fbbd-3123-4d87-9a2e-a0ce0eae933c	10	0	0.0	0	0.0	0	0.0
50ae73b5-3dd0-4298-b04b-aa0496000eb0	29	0	0.0	0	0.0	2	6.9
5157cb93-7a0e-4485-a715-fa36d3ca94cb	9	1	11.11	0	0.0	0	0.0
51e55b2f-611c-48f3-8310-8fb12bfaab94	12	0	0.0	0	0.0	0	0.0
581ffa03-7310-460e-8ce0-99af18046b73	8	0	0.0	8	100.0	0	0.0
5c24dfbd-f3b5-4608-892f-80cd2ec82f27	33	0	0.0	1	3.03	6	18.18
5da72401-3267-44e6-a133-095988933987	9	9	100.0	0	0.0	0	0.0
6415b59f-a6cf-4789-93cb-98024289ba74	33	33	100.0	1	3.03	21	63.64
6d98268c-df5e-4700-8192-c2633f5fbd99	13	0	0.0	1	7.69	0	0.0
77085a14-c648-4207-a555-d7d3987400fd	24	0	0.0	1	4.17	3	12.5
817d0fce-bb03-4d80-8da1-190999093b2f	8	6	75.0	0	0.0	0	0.0
905d170a-569b-4ee9-ae32-86bdb33f7309	11	0	0.0	0	0.0	1	9.09
9deaeb05-f44b-4c3b-83e7-c44fbe762b4c	10	0	0.0	0	0.0	0	0.0
a2defc9b-ba3b-4f3e-94b7-f07e72cae431	9	1	11.11	0	0.0	0	0.0
bb470657-d3a9-4c90-b7f4-95beaead5ae7	13	6	46.15	2	15.38	11	84.62
c80e8aa3-ca56-44c1-8ed7-868d2c78a52c	24	1	4.17	1	4.17	3	12.5
d8e5332d-7da1-491c-bf3a-248b10a56c30	16	1	6.25	15	93.75	0	0.0
dceb2da8-3a81-4210-80b5-c6ed1b682acf	14	1	7.14	14	100.0	0	0.0
ec2e2478-8239-4a2a-9f58-a258529fdd3c	9	9	100.0	0	0.0	0	0.0
f3fcb2bf-44ec-4c68-bce2-23bde9e14ebd	12	0	0.0	0	0.0	0	0.0

Рис. 3: Распределение нейтральных документов по корпусам

3. `percent_i` - процентное содержание нейтральных документов в корпусе для i-ой разметки

Полюсы-мнения
Теперь рассмотрим количество найденных мнений в каждом документе:
Обозначения:

1. `poles_i` - количество мнений в корпусе для i-ой разметки ($i \in \{1, 2, 3\}$)
2. `neutral_i` - наличие/отсутствие нейтральных документов в корпусе для i-ой разметки
3. `non_relevant_i` - наличие/отсутствие нерелевантных документов в корпусе для i-ой разметки

	poles_1	neutral_1	non_relevant_1	poles_2	neutral_2	non_relevant_2	poles_3	neutral_3	non_relevant_3
17af2621-05bd-4008-a8f5-4afe64a4e823	2	+	-	-	+	-	1	-	+
1bcad992-3329-4807-8ac5-920aee3c0db8	-	+	-	3	-	-	1	+	-
28db5916-80dc-42f9-a822-5f14259b0e14	2	-	-	3	-	-	1	+	-
3975555d-2f8b-48a4-85ed-7ce390c04618	-	+	-	3	-	+	3	-	-
3af7ccba-dc12-4d24-8271-298204826729	2	-	-	2	-	-	1	+	-
3c2da57c-c42f-4897-b5e7-7b44855baae6	4	-	+	3	-	+	2	+	+
42f49e66-6d17-4869-adb4-3b0ebf51fc87	2	-	+	1	-	+	2	-	-
46509952-8146-4262-9bb0-d77d94dab5ef	3	-	+	3	-	+	2	-	+
48b345a1-0d3d-4df8-ae7a-862d55f16fde	5	-	+	4	-	+	3	+	-
4ab6d1a4-f530-451c-bcfa-852c7784e9f6	-	+	-	4	-	-	2	-	-
4cd6fbbd-3123-4d87-9a2e-a0ce0eae933c	3	-	+	3	-	+	2	-	+
50ae73b5-3dd0-4298-b04b-aa0496000eb0	2	-	-	6	-	-	7	+	-
5157cb93-7a0e-4485-a715-fa36d3ca94cb	2	+	+	2	-	+	2	-	-
51e55b2f-611c-48f3-8310-8fb12bfaab94	3	-	+	4	-	-	3	-	-
581ffa03-7310-460e-8ce0-99af18046b73	2	-	+	-	+	-	2	-	+
5c24dfbd-f3b5-4608-892f-80cd2ec82f27	6	-	-	9	+	-	3	+	-
5da72401-3267-44e6-a133-095988933987	-	+	-	2	-	+	2	-	-
6415b59f-a6cf-4789-93cb-98024289ba74	-	+	-	4	+	+	2	+	+
6d98268c-df5e-4700-8192-c2633f5fbd99	3	-	-	2	+	-	2	-	+
77085a14-c648-4207-a555-d7d3987400fd	4	-	-	6	+	-	3	+	-
817d0fce-bb03-4d80-8da1-190999093b2f	-	+	+	2	-	+	2	-	+
905d170a-569b-4ee9-ae32-86bdb33f7309	4	-	+	4	-	+	3	+	+
9deaeb05-f44b-4c3b-83e7-c44fbe762b4c	2	-	-	4	-	-	2	-	-
a2defc9b-ba3b-4f3e-94b7-f07e72cae431	2	+	+	2	-	+	2	-	+
bb470657-d3a9-4c90-b7f4-95beaead5ae7	2	+	-	3	+	-	-	+	-
c80e8aa3-ca56-44c1-8ed7-868d2c78a52c	3	+	-	3	+	-	2	+	-
d8e5332d-7da1-491c-bf3a-248b10a56c30	2	+	-	1	+	-	3	-	-
dceb2da8-3a81-4210-80b5-c6ed1b682acf	2	+	-	-	+	-	1	-	+
ec2e2478-8239-4a2a-9f58-a258529fdd3c	-	+	-	3	-	-	2	-	-
f3fcb2bf-44ec-4c68-bce2-23bde9e14ebd	3	-	-	3	-	-	3	-	-

Рис. 4: Распределение мнений по документам/корпусам

Видим, что в разметке присутствует несколько разных комбинаций:

1. только полюсы-мнения
2. полюсы-мнения + нейтральные
3. полюсы-мнения + нерелевантные
4. полюсы-мнения + нейтральные + нерелевантные
5. только нейтральные
6. нейтральные + нерелевантные

3 Постановка задачи

3.1 Метрики

Наша задача заключается в корректной кластеризации документов внутри каждой темы. Нужно не только верно выделить мнения, но и проверить, существует ли отдельная группа нейтральных/ нерелевантных документов.
Для сравнения разметки $X = \{x_1, ..., x_n\}$ с «золотым стандартом» — экспертной разметкой $Y = \{y_1, ..., y_n\}$, используются BCubed-версии точности и полноты поиска.
Для контроля качества алгоритма введем несколько критериев:

Критерий M1: точность и полнота кластеризации мнений
Точность и полнота сначала определяются относительно каждого объекта x_i, затем усредняются по

всем объектам с меткой мнения:

$$P = \underset{x_i>0}{avr} P_i; \qquad P_i = \frac{\sum_k [x_k{=}x_i \ and \ y_k{=}y_i]}{\sum_k [x_k{=}x_i]}$$

$$R = \underset{y_i>0}{avr} R_i; \qquad R_i = \frac{\sum_k [x_k{=}x_i \ and \ y_k{=}y_i]}{\sum_k [y_k{=}y_i]}$$

(если знаменатель дроби оказывается равным нулю, то считается, что дробь равна нулю).
Агрегированный критерий (F1-мера) определяется как среднее гармоническое:

$$M_1(X,Y) = \frac{2PR}{P+R}$$

Критерий M2: точность и полнота отсева нейтральных документов
Определим точность и полноту отсносительно метки с (c=0):

$$P_c = \frac{\sum_k [x_k{\neq}c \ and \ y_k{\neq}c]}{\sum_k [x_k{\neq}c]}$$

$$R_c = \frac{\sum_k [x_k{\neq}c \ and \ y_k{\neq}c]}{\sum_k [y_k{\neq}c]}$$

Глядя на формулу, можно сказать, что по сути мы оцениваем, как хорошо модель отделяет субъективные позиции от нейтральной констатации фактов. Агрегированный критерий (F1-мера) определения нейтральных сообщений:

$$M_2(X,Y) = \frac{2P_0R_0}{P_0+R_0}$$

Критерий M3: точность и полнота отсева нерелевантных документов
Здесь по аналогии с нейтральными документами мы оцениваем корректность выявления релевантных, отделяем нужные нам мнения от шума. Агрегированный критерий (F1-мера) определения релевантных сообщений (c=-1):

$$M_3(X,Y) = \frac{2P_{-1}R_{-1}}{P_{-1}+R_{-1}}$$

$$M_4(X,Y) = \frac{min\{K_x,K_y\}}{max\{K_x,K_y\}}$$

Обозначим через K_x и K_y число различных мнений в разметках X и Y соответственно.

$$M_4(X,Y) = \frac{min\{K_x,K_y\}}{max\{K_x,K_y\}}$$

3.2 Эксперименты с метриками

Описанные выше метрики являются результатом нескольких экспериментов. Критерии M1 и M4 стабильно давали понятные и хорошо интерпретируемые результаты как для модели, так и для разметчиков. Проблемы возникали с M2 и M3. Количество найденных нерелевантных документов сильно отличается от корпуса к корпусу, но, как мы видим на рисунке №2, доля согласия у разметчиков довольно высокая. Сложнее ситуация обстоит с нейтральными документами. Во-первых, из рисунка №3 следует, что весь корпус документов может быть нейтральным (в отличие от нерелевантных). Во-вторых, доля согласия между асессорами существенно ниже. Изначально точность и полнота для M2 и M3 задавались формулами:

$$P_c = \frac{\sum_k [x_k{=}y_k{=}c]}{\sum_k [x_k{=}c]}$$

$$R_c = \frac{\sum_k [x_k = y_k = c]}{\sum_k [y_k = c]}$$

При таком определении метрик возникало сразу несколько сложностей:

1. неясно, как определять P в случае нулевого знаменателя, что случалось часто. Для precision нулевой знаменатель означает, что модель не нашла нерелевантные/нейтральные документы, что автоматически обнуляет и числитель дроби. Получили неопределенность вида $\frac{0}{0}$.
2. аналогичная проблема возникает и с R. Нуль в знаменателе recall говорит о том, что в разметке нет нейтральных/нерелевантных документов. Тогда воникает вопрос, можем ли мы оценивать корпус для таких X и Y. При этом, оставив NaN и отказавшись от оценивания, мы теряем информацию о качестве модели.
3. для M2 получаем очень низкие значения из-за довольно слабого согласия между экспертами.

Результаты метрик при таком задании precision и recall представлены ниже:

	M2	M3
Разметка	0.08	0.49

Оценим качество модели и уровень согласия разметчиков с помощью каппа-статистики Коэна:
$\kappa = \frac{p_0 - p_e}{1 - p_e}$; $\kappa \in [0, 1]$, где p_0 - относительное наблюдамое согласие среди оценщиков (accuracy), p_e - гипотетическая вероятность случайного совпадения.

Значения κ носят следующий смысл ((11)):

kappa	интерпретация
<0	согласие меньше, чем случайная вероятность
0	нет согласия
0.1-0.2	незначительное согласие
0.21-0.4	"удовлетворительное"согласие
0.41-0.6	умеренное согласие
0.61-0.8	существенное согласие
0.81-0.99	почти идеальное согласие
1	полное согласие

Применив данную метрику к нерелевантным документам, получил, что средний уровень согласия разметчиков - 0.35, а качество модели - 0.2. Вызвано это тем, что каппа-статистика сильно штрафует неверные ответы для класса с меньшим количеством объектов. Но от одного корпуса к другому «меньшим классом» может являться как множество нерелевантных/нейтральных, так и множество кластеров-мнений (см. рис.2 и рис.3). Таким образом, штрафы за ошибки разных типов «смешиваются», и мы не получаем реальной оценки качества работы конкретного блока алгоритма.

Поэтому в качестве итоговых метрик были выбраны критерии, описанные в разделе 3.1.

4 Модель

4.1 Подробнее про данные

Опишем данные с технической точки зрения. На вход поступил JSON-файл, содержащий корпуса документов. Каждый документ имеет уникальный идентификационный номер. Также в имеющимся датасете присутствует текст каждой новости и выделенные именованные сущности с промаркированной тональностью.
Пример: `{'neg': {'ТАСС': 1, 'Приморском крае': 1}, 'pos': {'Орловской области': 1}}`

В ходе работы над моделью понадобились лемматизированные предложения, поэтому текст каждого документа был соответствующим образом обработан и добавлен в датасет.

4.2 Эксперименты

Сначала была предпринята попытка решить поставленную задачу с помощью методов обучения с учителем. Использовался RandomForestClassifier, работающий на TF-IDF (term frequency - inverse document frequency) представлении лемматизированного текста. Лучшим результатом являлась F1-мера, равная 0.95 для кластеризации мнений и 0.67 для выделения нейтральных. Однако при уменьшении обучающей выборки и увеличении тестовой данные значения стремительно падали. От такого варианта реализации было решено отказаться, так как в общем случае нам неизвестен размер приходящего датасета; кроме того, в реальной задаче отсутствует разметка, поэтому supervised методы не подойдут.

Далее был сформирован общий вид модели - алгоритм, имеющий трехступенчатую структуру: на 1-ом шаге проводится отсев нерелевантных документов, на 2-ом - нейтральных, а на 3-ем шаге выполняется деление на кластеры-мнения.

4.2.1 Выделение нерелевантных документов

Выделение нерелевантных документов происходит при помощи подсчета семантической близости лемматизированных предложений в каждом документе. Для этого используется библиотека SpaCy с готовым пайплайном `ru_core_news_sm` для русского языка; данный пайплайн обучен на текстах новостей. Так как семантическая близость считается для пар документов, получаем $C^2_{\texttt{docs\_amount}}$ пар. Как будет описано ниже, мы не используем семантическую близость всех пар сообщений. Проведя несколько экспериментов, было решено брать только 8 самых близких к конкретному документу сообщений. А характеристикой документа является среднее значение семантической близости его 8-ми соседей. Подробное описание данных экспериментов представлено ниже.

OPTICS
Сначала было решено применить unsupervised метод OPTICS (Ordering Points To Identify the Clustering Structure). Можно сказать, что OPTICS является обобщением более известного DBSCAN. Есть несколько существенных отличий и особенностей:

1. OPTICS строит дендрограмму по ближайшим соседям точек
2. также внутри OPTICS строится график достижимого расстояния (расстояния до ближайшего соседа) от номера точки, а кластеры определяются как «долины» на графике

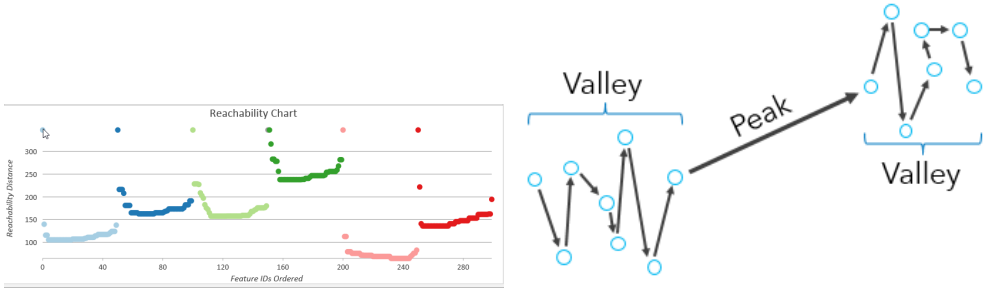

Рис. 5: Принцип работы OPTICS

3. как и DBSCAN, OPTICS выделяет шумовые точки
4. технология OPTICS требует больше памяти, так как хранит большое количество расстояний до ближайших точек для определения расстояния достижимости
5. главным достоинством метода OPTICS является то, что он не требует указания параметра epsilon, что приводит к существенному сокращению процесса настройки параметров

Данный подход дал следующие результаты:
* - метрика M3 вычислялась как агрегированный критерий поиска НЕрелевантных (раздел 3.2)

	M3*
Алгоритм	0.39
Разметка	0.49

Далее было решено протестировать методы, ориентированные именно на выявление аномалий.

IsolationForest
Попробовал применить алгоритм Isolation Forest, используя разное число деревьев (`n_estimators`).

Результаты при использовании финальной реализации метрики M3 (раздел 3.1):

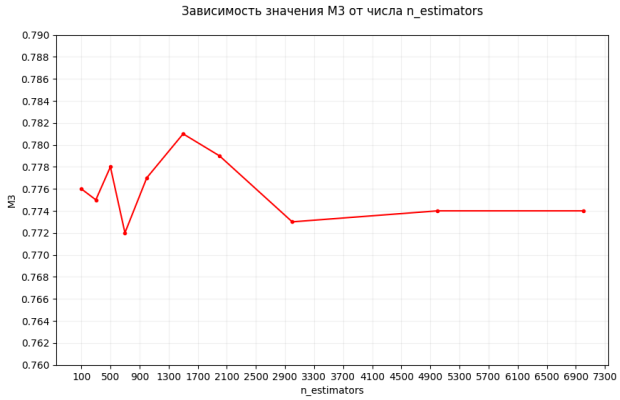

Рис. 6: Зависимость качества выделения нерелевантных от числа деревьев

Максимальное значение M3=0.781 было достигнуто при `n_estimators`=1500 (при $M3_{expert} = 0.93$).

Видим, что в сравнении с OPTICS (по «старой» метрике) качество повысилось. Однако существенным недостатком Isolation Forest является то, что он предполагает наличие лишь нескольких аномалий, а, глядя на разметку, мы видим, что в корпусе может присутствовать до 79% нерелевантных документов. Поэтому было принято решение протестировать еще одну модель.

OneClassSVM
Данный алгоритм позволяет осуществить более точную настройку выделения аномалий, также здесь по умолчанию стоит ограничение в 50% на долю выбросов среди всей выборки, но этот параметр можно указать любым.

Сначала проверим три бейзлайна с разными ядрами:

1. `'linear'` - линейное ядро
2. `'rbf'` - радиальная базисная функция
3. `'poly'` - полиномиальное ядро

	linear	rbf	poly
M3 (old)	0.562	0.45	0.58
M3 (new)	0.68	0.632	0.682

Видим, что два из трех бейзлайнов превышают «старую» экспертную метрику M3 (0.49), а при повышении точности (tol) разница становится еще больше. Рассмотрев подробнее метки объектов, был заметен очень низкий precision при высоком recall. Такие показатели доказывают, что данная метрика действительно некорректно отражает качество выделения аномалий. Таким образом, подтвердились проблемы, указанные в разделе 3.2, поэтому с текущего момента будем оценивать все модели только по

новой реализации критерия М3 (раздел 3.1).

Что касается новой версии критерия М3, то лучшее значение достигается при полиномиальном ядре. Постараемся подобрать некоторые другие гиперпараметры для улучшение качества.

Попробуем найти оптимальную степень полинома:

	1	2	3	4	5	6	7	8
М3	0.666	0.666	0.682	0.672	0.671	0.684	0.68	0.669

Заметим, что значения практически не отличаются(различие в пределах шума).

Теперь обратим внимание на другой важный гиперпараметр OneClassSVM. Параметр nu отвечает за верхний порог ожидаемой доли аномалий в датасете. По умолчанию он равен 0.5. Посмотрим на значение метрики М3 при разных nu:

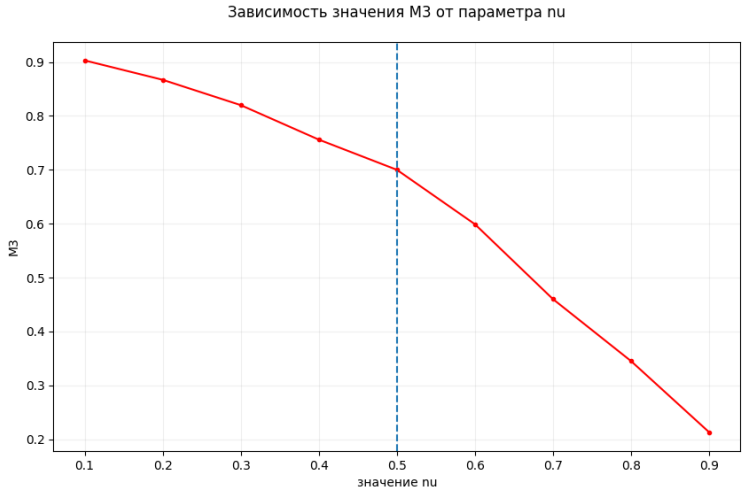

Рис. 7: Зависимость значения М3 от верхнего порога доли аномалий

Качество ожидаемо снижается при увеличении верхнего порога. В разделе 2.1 было показано, что в данном датасете асессоры нашли около 10% нерелевантных документов. Поэтому наиболее высокое качество достигается именно при таком пороге. Но, используя данную информацию, мы фактически пользуемся разметкой, поэтому введем модифицированный алгоритм выделения нерелевантных сообщений.

Что касается остальных гиперпараметров OneClassSVM, их подбор сильно зависит от конкретного датасета, и, так как мы решаем поставленую задачу в общем виде, их перебирать не будем.

Модифицированный алгоритм
Сначала коротко опишем подготовку данных в предыдущих экспериментах. Как было сказано в начале текущего раздела, мы не используем значения семантической близости всех пар документов, хотя на начальных этапах был вариант алгоритма, где для каждого документа считалась сумма расстояний до всех остальных сообщений. Такая модель достигла значения М3=0.685 при полиномиальном ядре с degree=6 и nu=default. Такой подход можно существенно улучшить. Теперь будем брать некоторое число «соседей» (наиболее близких сообщений) и усреднять эти значения.

Схематично алгоритм выглядит так:

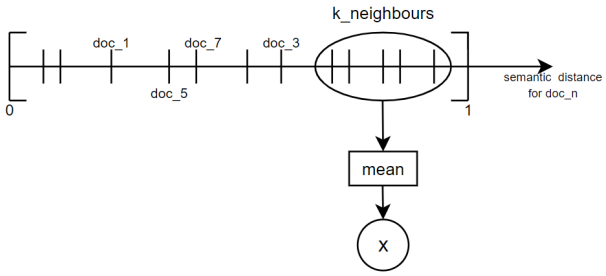

Рис. 8: Обработка семантической близости пар документов

Теперь попробуем разбить данные на две части и анализировать их отдельно. Мы можем сразу выделить 50% документов с наибольшими усредненными значениями семантической близости. Эти сообщения будем считать заведомо релевантными, так как если в корпусе находится более половины шумовых объектов, то, возможно, они и являются «чистыми» данными, а меньшая часть - это аномалии. Далее к 50% документов с меньшими показателями применяем OneClassSVM/Isolation Forest. Здесь использование Isolation Forest уже более корректно, так как поиск нерелевантных происходит в подвыборке, и проблема, описанная в 4.2.1 (IsolationForest), становится не такой серьезной.

Посмотрим на значения M3 при разном числе `"соседей"`:

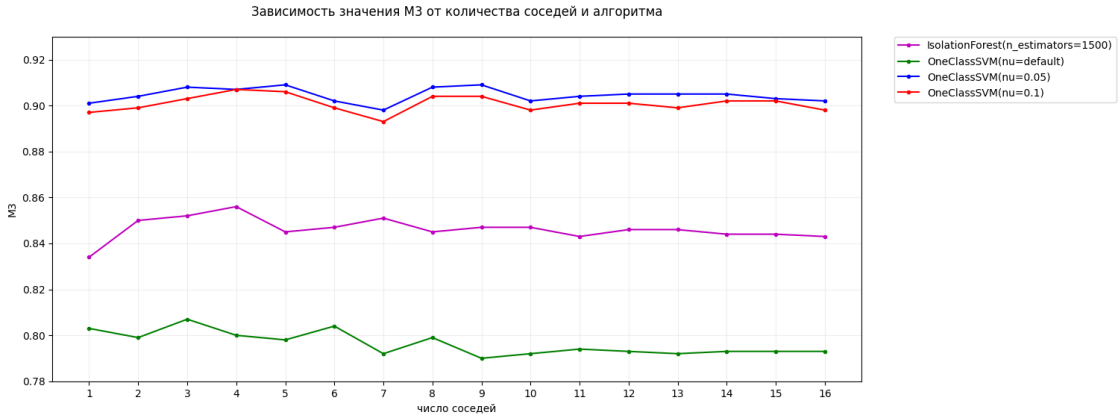

Рис. 9: Зависимость значения M3 от числа соседей

Из четырех рассмотренных моделей IsloationForest и OneClassSVM(nu=default) являются общими решениями, в то время как другие две вариации OneClassSVM используют информацию из разметки, но при этом демонстрируют более высокое качество.

Теперь рассмотрим случай, когда число соседей зависит от размера корпуса. На приведенном ниже графике `n_docs` обозначает количество документов в корпусе.

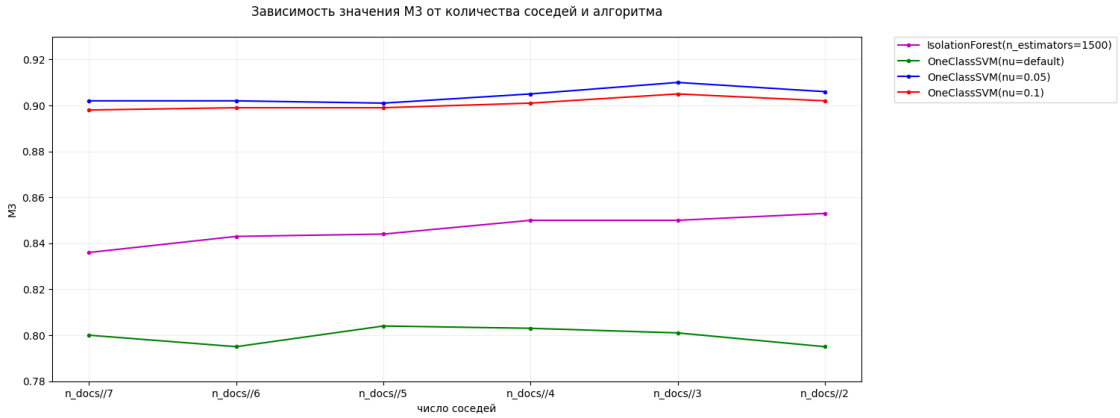

Рис. 10: Зависимость значения M3 от числа соседей

Заметим, что данный график фактически представляет собой усредненную и сглаженную версию предыдущего. Оптимальным по суммарным результатам четырех графиков является `n_docs//3`. Наверное, для небольших корпусов сообщений такой подход будет работать хорошо, но для многочисленных блоков документов лучше брать фиксированное число соседей.

4.2.2 Выделение нейтральных документов

Нейтральный документ представляет собой текст без какой-либо субъективной позиции и/или эмоциональной окраски. Поэтому в качестве основного инструмента для определения сообщений такого вида было решено взять эмоционально окрашенные именованные сущности, имеющиеся в датасете (раздел 4).

Первый подход: для каждого документа проверялось наличие/отсутствие эмоционально окрашенных слов. Если таковых не было обнаружено, то документ считался нейтральным. При данной стратегии M2=0.64.

Второй подход: теперь положительно и негативно окрашенные именованные сущности разделяются. Проверяется гипотеза, что сентименты разной эмоциональной окраски могут друг друга компенсировать. Тогда документ считается нейтральным, если такие слова отсутствуют или если их одинаковое количество на каждой полярности. В таком случае M2=0.65.

4.2.3 Кластеризация мнений

После предыдущих двух этапов (выделение нерелевантных и нейтральных) остались только документы, которые отражают одно или несколько разных мнений. Для разделения их на кластеры будет использоваться метод KMeans++. Также возможен случай, когда в корпусе были только нерелевантные и нейтральные сообщения. Тогда текущий шаг пропускается.

Архитектура алгоритма имеет следующий вид:

1. строим Tf-Idf представление эмоционально окрашенных именованных сущностей
2. перебираем число кластеров от 1 до 10, вычисляя Sum of Squared Errors (SSE)
3. применяем Elbow technique для нахождения искомого числа мнений
4. применяем KMeans++

В качестве Elbow technique возьмем готовый KneeLocator. Продемонстрируем несколько графиков-корпусов с найденным оптимальным числом кластеров:

Обратим внимание на метод KMeans++, его основное отличие от классического KMeans заключается в более удачных начальных значениях центроидов кластеров. В оригинальном KMeans начальное положение центроидов задается случайным образом, это может повлечь нестабильность алгоритма. Сравним эти две инициализации:

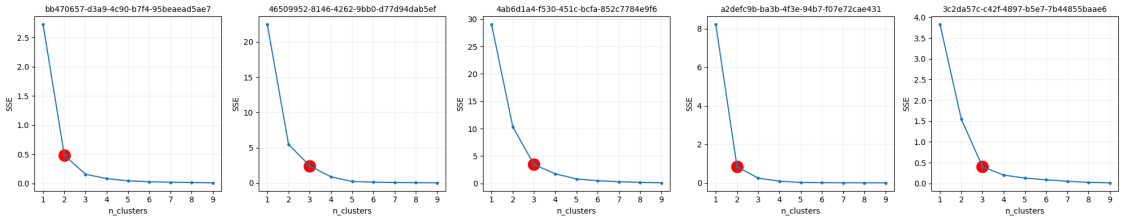

Рис. 11: Оптимальное число кластеров-мнений для некоторых корпусов

	M1	M4
KMeans++	0.65	0.696
Kmeans	0.639	0.685
Разметка	0.642	0.694

Отметим, что сравнение проводилось при `algorithm='auto'`, `max_iter=300`. Смена алгоритма и/или увеличение числа итераций либо оставляют значения метрик такими же, либо уменьшают их.

Также данный алгоритм имел еще одну модификацию, где кластеризация проводилась на Tf-Idf представлении лемматизированных предложений. Такой подход имеет сразу несколько недостатков. Во-первых, значения метрик существенно снижаются (M1=0.6, M=0.61). Во-вторых, использование исключительно лексики даст скорее разбиение на темы, а не на мнения внутри темы.

4.3 Финальный алгоритм

Теперь, когда были рассмотрены все этапы обработки и моделирования, можно описать финальный алгоритм. Модель имеет трехступенчатую структуру:

1. выделение нерелевантных сообщений: если заказчик/автор датасета знает потенциальную долю зашумления, можно использовать OneClassSVM с выставленным порогом nu (это даст лучшее качество из-за более точной настройки модели); в противном случае можно использовать Isolation Forest.
2. выделение нейтральных, используя предоставленную информацию об эмоциональной окраске именованных сущностей.
3. кластеризация мнений при помощи Elbow Technique и KMeans++

При независимом применении каждого из трех этапов имеем следующие значения метрик:

	M1	M2	M3	M4
Алгоритм	0.65	0.65	0.856/0.907	0.695
Разметка	0.64	0.62	0.93	0.69

(первое значение M3 для алгоритма - при использовании IsolationForest, второе - при OneClassSVM)

Теперь применим все шаги последовательно.

Если при обнаружении нерелевантных использовать Isolation Forest:

	M1	M2	M3	M4
Алгоритм	0.68	0.63	0.85	0.73

Получили хорошие значения метрик. Обратим внимание, что присутствуют некоторые отличия в сравнении с независимым применением этапов модели:

1. M1 возросла на 0.03

2. M2 снизилась на 0.02
3. M4 возросла на 0.04, что соответствует 106% относительно разметки (возможна погрешность из-за малого размера датасета)

Теперь для обнаружения нерелевантных документов будем пользоваться OneClassSVM, учитывая, что примерная доля шума равна 0.1:

	M1	M2	M3	M4
Алгоритм	0.675	0.645	0.9	0.78

1. M1 практически не изменилась в сравнении с Isolation Forest
2. M2 возросла на 0.025 относительно разметки и на 0.05 относительно Isolation Forest
3. M4 возросла на 0.09 относительно разметки (это соответствует 112% качества) и на 0.05 относительно Isolation Forest

Видно, что метрики M1 и M4 в обоих случаях увеличились в сравнении с независимым применением алгоритмов. Думаю, это связно с тем, что на вход KMeans подаются `"чистые"` данные, это исключает вероятность ошибочного выделения нерелевантных документов в дополнительный кластер и/или перемешивания нужных документов с нейтральными.

Посмотрим, есть ли зависимость между числом документов в корпусе и итоговыми значениями метрик M1 и M4, отвечающих за точность/полноту и количество мнений:

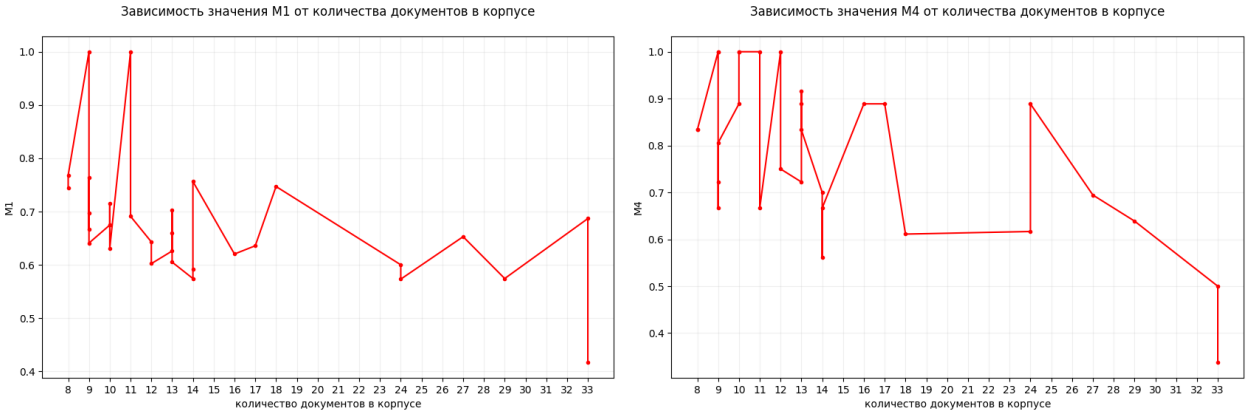

Рис. 12: Зависимость M1 и M4 от числа документов в корпусе

Видим, что явная зависимость отсутствует, однако на обоих графиках есть выброс на корпусе из 33-ех сообщений.

5 Выводы

В данной работе решается задача выявления поляризации в корпусе новостных текстов. Разработана трехступенчатая модель, последовательно выделяющая нерелевантные, нейтральные документы и разделяющая релевантные документы на кластеры-мнения. Было показано, что

1. существует несколько рабочих стратегий выделения нерелевантных (в зависимости от наличия информации о датасете)
2. использование эмоционально окрашенных именованных сущностей является более корректным и полезным методом, чем применение всей лексики
3. отсутствует явная зависимость качества работы модели от размера корпуса документов

Список литературы

[1] Е.К.Милюта. Языковые модели для обнаружения поляризации общественного мнения в новостоном потоке, 2022.

[2] Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Клышинский Э.С., Лукашевич Н.В., Сапин А.С. Автоматическая обработка текстов на естественном языке и анализ данных, 2017.

[3] Д.Г.Фельдман. Комбинирование фактов, семантических ролей и тональных слов в генеративной модели для поиска мнений, 2020.

[4] Kajal Yadav. Text Clustering using K-means, 2021.

[5] Francis Ndiritu. How to Build an NLP Based Emotion Detection Model using Neattext and Scikit-learn, 2021.

[6] Yasir Ali Solangi, Zulfiqar Ali Solangi. Review on Natural Language Processing (NLP) and Its Toolkits for Opinion Mining and Sentiment Analysis, 2018.

[7] Federico Pascual. Getting Started with Sentiment Analysis using Python, 2022.

[8] CY Yam. Emotion Detection and Recognition from Text Using Deep Learning, 2015.

[9] Enrique Amigo, Julio Gonzalo, Javier Artiles, Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints, 2008.

[10] Estela Saquete, David Tomás, Paloma Moreda, Patricio Martínez-Barco, Manuel Palomar. Fighting post-truth using natural language processing: A review and open challenges, 2019.

[11] Ajitesh Kumar. Cohen Kappa Score Python Example: Machine Learning, 2022.