УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

На правах рукописи

Крайнов Александр Юрьевич

ИЕРАРХИЧЕСКАЯ ОБРАБОТКА ПОТОКОВ ТЕКСТОВЫХ СООБЩЕНИЙ НА БАЗЕ НАИВНОГО БАЙЕСОВСКОГО КЛАССИФИКАТОРА

Специальность 05.13.18 — Математическое моделирование, численные методы и комплексы программ

Диссертация на соискание учёной степени кандидата технических наук

Научный руководитель
доктор технических наук,
профессор
Смагин А. А.

Ульяновск — 2013

Оглавление

Введение

Проблема *обработки информационных потоков* была чётко обозначена ещё в 80-е годы XX века. Как отмечено в [1, с. 11], "В середине 50-х годов резко возросшие темпы научно-технического прогресса привели к информационному взрыву — скачкообразному увеличению объемов создаваемой и используемой информации, усложнению существовавших и образованию новых информационных потоков, которые захлестнули информационные системы общества. Наиболее ярко информационный взрыв проявился в вида лавинообразного роста потоков публикаций в области технических, естественных и гуманитарных наук. Менее заметным для широкой общественности, но не менее серьезным по своим масштабам и социальным последствиям было резкое увеличение объёма документации, используемой в сфере управления, сопровождавшееся соответствующим ростом количества информационных материалов в области производства и технологии. Общий аффект информационного взрыва усиливался возрастанием информационных потоков в системах массовой информации, возможности которых необычайно расширились в связи с развитием печати, радио, телевидения и сетей связи".

Со времён выхода этой книги положение дел в области обработки потоков текстовых сообщений несколько изменилось. С одной стороны, во второй половине 90-х годов были созданы и стали развиваться эффективные системы информационного поиска глобального масштаба, так называемые поисковые машины (search engines). С другой стороны, за последние 15 лет дальнейшее развитие телекоммуникационных технологий привело к возникновению большого числа новых источников потоков текстовых сообщений. Всё больший масштаб приобретают социальные медиа-ресурсы (социальные сети, блоги, микроблоги, вики и др. [2]) и электронные средства массовой информации. В настоящее время методы и системы сбора и обработки потоков текстовых сообщений из разрозненных источников представляют особый интерес для аналитиков, работающих в самых разных сферах: бизнесе, экономике, государственном управлении и т. д. [3, с. 186]

Традиционные методы анализа текстов, основанные на *глубинной обработке естественного языка* (ОЕЯ), ориентированы на взаимодействие с хранилищами документов, изменения в которые вносятся сравнительно редко: национальными корпусами текстов, электронными библиотеки, базами научных статей или архивами веб-сайтов. В современных условиях эти методы малоприменимы на практике из-за больших объёмов текстовых коллекций. С другой стороны, большинство методов *интеллектуального анализа текстов* (ИАТ, text mining) и поверхностной ОЕЯ, исследования которых продолжаются с конца 80-х годов, не учитывают динамический характер потоков. Разработка алгоритмов ИАТ, ориентированных на обработку потоков текстовых сообщений, является одним из наиболее перспективных направлений современной информатики.

Современные задачи интеллектуального *анализа текстовых потоков* (text stream mining) включают:

- классификацию потоков — распределение сообщений по заранее заданным группам (категориям, тематикам, событиям);
- кластеризацию потоков — распределение сообщений по группам, которые должны быть определены в процессе работы алгоритма;
- обнаружение и отслеживание тематик (topic detection and tracking), обнаружение возникающих трендов (emerging trend detection), включая выявление технологических трендов — идентификация новых тем, соответствующих новым явлениям, событиям, предметам и т. д.;
- анализ эволюции потоков (evolution analysis) — исследование динамики отдельных тем, а также их взаимодействий с течением времени.

В настоящей работе под потоком текстовых сообщений понимается последовательность текстовых сообщений с определёнными для каждого сообщения моментами времени. Под обработкой потока текстовых сообщений понимается комплексная задача оперативной классификации поступающих сообщений, определения новизны сообщений и обнаружения возникающих тематик.

Объектом исследования диссертационной работы являются потоки текстовых сообщений. **Предметом исследования** выступают математические

и компьютерные модели этих потоков и методы обработки входящих в них сообщений.

Цели и задачи исследования.

Целью настоящей работы является повышение эффективности обработки потоков текстовых сообщений в системах принятия решений. Для выполнения поставленной цели в работе решаются следующие задачи:

1. анализ современных методов обработки потоков текстовых сообщений для оценки ситуации в предметной области и выявление путей повышения эффективности обработки потоков;
2. построение математической модели текстового информационного потока, которая позволяет в формальном виде отобразить и исследовать закономерности между тематиками и динамику тематических потоков;
3. разработка алгоритма обработки потока текстовых сообщений, позволяющего производить оперативную классификацию сообщений, определение новизны сообщений и обнаружение возникающих тематик;
4. выбор критериев эффективности обработки потока и разработка метода оценки эффективности итерационных алгоритмов обработки текстовых сообщений по выявленным критериям для выбора наиболее эффективного алгоритма;
5. разработка программного комплекса, поддерживающего предложенный алгоритм обработки потока, и экспериментальное исследование эффективности предложенного алгоритма.

Методы исследования.

При решении поставленных задач использовались методы системного анализа, математического и компьютерного моделирования, обработки естественного языка, теории вероятностей, прогнозирования временных рядов, искусственного интеллекта, разработки информационных систем и программирования.

Научной новизной обладают разработанные в диссертационном исследовании математические модели потока текстовых сообщений и системы обработки потоков; метод многозначной наивной байесовской классификации; предложенные оценки степени новизны сообщения и тематики; итерационный

оперативный алгоритм обработки потока текстовых сообщений с частичным обучением и методика оценки его эффективности; разработанный программный комплекс, реализующий предложенные алгоритмы.

Положения, выносимые на защиту:

1. Математическая модель системы обработки потоков текстовых сообщений, позволяющая построить эффективный алгоритм обработки потока, учитывать динамику тематик во времени и производить адаптацию алгоритма к предметной области.
2. Математическая модель потока текстовых сообщений в виде направленного ациклического графа.
3. Метод многозначной классификации на основе наивного байесовского подхода, в котором применена сбалансированная процедура вычисления степени полезности признаков.
4. Итерационный оперативный алгоритм обработки потока текстовых сообщений с частичным обучением, позволяющий производить классификацию текстовых сообщений, обнаружение возникающих тематик и вычисление степени новизны сообщений и тематик.
5. Программный комплекс, позволяющий применять разработанные методы к различным предметным областям и производить вычислительные эксперименты по оценке эффективности предложенных решений на основе скользящего контроля.

Практическая и теоретическая значимость исследований.

Результаты диссертационной работы могут найти применение в задачах принятия решений на основе анализа потоков сообщений. Разработанный программный комплекс может быть непосредственно применён для анализа потоков новостных сообщений, заявок на модификацию программной продукции, обращений граждан в государственные учреждения, контент-мониторинга социальных медиа-ресурсов.

Достоверность проведённых в диссертационной работе результатов определяется корректным использованием методов исследования, подтверждена результатами вычислительных экспериментов и эффективностью функционирования алгоритмов и программного обеспечения при внедрении.

Апробация основных положений диссертационной работы проведена на XV международной научно-практической конференции "Перспективы развития информационных технологий" (Новосибирск, 2013) и XII международной научной конференции "Актуальные вопросы современной техники и технологии" (Липецк, 2013).

Личный вклад автора. Постановка задачи исследования осуществлена совместно с научным руководителем А. А. Смагиным. Основные теоретические и практические исследования проведены автором самостоятельно.

Структура и объём работы.

Диссертационная работа состоит из введения, четырёх глав, заключения, списка литературы из 112 наименований источников отечественных, зарубежных авторов и электронных ресурсов и одного приложения. Общий объём диссертации составляет 147 страниц машинописного текста, в том числе 117 страниц основного текста и 30 страниц приложений.

Глава 1. Сравнительный анализ современных подходов и систем обработки потоков текстовых сообщений

1.1. Проблема обработки потоков текстовых сообщений

1.1.1. Современное состояние обработки естественного языка как направления искусственного интеллекта

Анализ и понимание текстов на естественном языке занимали важную нишу в исследованиях по искусственному интеллекту с момента возникновения этой науки. В течение многих десятилетий работы в этом направлении были сосредоточены в областях, называемых "вычислительной лингвистикой" и "обработкой естественного языка". Целью *вычислительной лингвистики* (ВЛ, computational linguistics, CL) является "построение логико-лингвистических моделей и соответствующих им алгоритмов и программ" [4, введение]. В отличие от ВЛ, где большое значение имеет теоретическая лингвистическая корректность и адекватность предложенных моделей, *обработка естественного языка* (ОЕЯ, natural language processing, NLP) сосредоточена "на моделировании всего того, что изучает лингвистика в целом" [4].

Первоначально обе этих области, преимущественно, стремились построить точные языковые модели, основанные на формальном синтаксическом и семантическом анализе; этот подход получил название "глубинная ОЕЯ", deep NLP, DNLP [5] (иногда применяется термин "символическая ОЕЯ" — symbolic NLP). Потребность в точных моделях была обусловлена тем, что одной из первых задач для являлся автоматический машинный перевод [6, с. 133]. Исследования в направлении глубинной ОЕЯ стимулировались развитием генеративной лингвистики, начало которой было положено с выходом в 1957 году работы Н. Хомского "Синтаксические структуры" [7; 8].

Со времени появления в 1990-х годах понятий "обнаружение знаний" (knowledge discovery) и "интеллектуальный анализ данных" (ИАД, data mining), исследователи обнаружили, что для многих практических задач, в

первую очередь информационного поиска (в том числе поисковых машин масштаба всемирной сети Интернет и поиска для социальных сервисов), удовлетворительные результаты могут быть получены более простыми и вычислительно эффективными способами — на основе статистических данных о тексте [9, с. 7]. Этот подход называется "поверхностная ОЕЯ", shallow NLP, SNLP (иногда — "эмпирическая NLP" — empirical NLP).

Современная область интеллектуального анализа текстов (ИАТ, text mining) является динамично развивающимся и практически востребованным направлением ОЕЯ, основанным на применении методов ИАД и машинного обучения (machine learning).

Ключевыми группами задач ИАТ можно назвать [10, с. xi]:

- *классификацию* (распределение текстов по заранее заданным группам) и *кластеризацию* (распределение текстов по группам, которые должны быть определены в процессе работы алгоритма);
- *извлечение информации* (information extraction, идентификация фактов и метаинформации о них в тексте) и информационный поиск;
- *обнаружение возникающих трендов* (emerging trend detection) — идентификация новых тем в коллекции текстов, соответствующих новым явлениям, событиям, предметам и т. д.[1]

Важным параметром систем ИАТ является качество *средств просмотра* (browsing) результатов, в том числе визуализации данных и навигации [11, с. 10], поэтому проектирование моделей уровня представления (presentation layer) можно считать четвёртой группой задач ИАТ.

Первые системы ИАТ, относящиеся к 1990-м годам, основывались на вычислительно несложных методах, разрабатываемых для каждой отдельной задачи. С развитием техники и появлением дешёвых вычислительных ресурсов, в том числе за счёт организации распределённых вычислений [12], в область ИАТ были привлечены более сложные методы теории вероятностей и статистики. В частности, в настоящее время зарубежные исследования по обработке текстов, преимущественно, связаны с так называемым тематическим

[1] Классическая задача обнаружения трендов касается обработки архивных данных за большой период времени, т. е. не подразумевает оперативную обработку текстовых документов по мере их поступления.

моделированием (topic modeling, TM) [13], основанным на аппарате статистического вероятностного вывода [14; 15].

В то же время нельзя утверждать, что ранние, более простые методы анализа данных и текстов не применимы в современных системах [16, с. 47]. Исследователями искусственного интеллекта было показано, что в ряде случаев применение интуитивных, нечётких и приближённых методов обладает существенными преимуществами в плане эффективности и прозрачности систем (см. напр. [17]).

1.1.2. Области применения алгоритмов обработки потоков текстовых сообщений

Поток текстовых сообщений — это последовательность текстовых сообщений с предписанными им моментами времени (создания или поступления на вход получателя).

Стремительное развитие сети Интернет и мобильных технологий в последние 15 лет привело к распространению приложений и систем, ориентированных на большие объёмы текстовых данных, поступающих в потоковом режиме. В качестве примеров источников массивных потоков текстовых сообщений можно назвать следующие [18]:

- *Службы коротких сообщений* (short message service, SMS), позволяющие обмениваться сообщениями пользователям мобильных сетей. Число пользователей SMS в 2012 году составляло примерно 3,5 миллиарда; количество отправляемых сообщений — 17,6 миллиарда в день [19].
- *Агрегаторы новостей* (онлайновые, такие как Google News, или клиентские, такие как Liferea). Новостные сообщения отличаются большей структурированностью и, как правило, более длинные, чем сообщения SMS.
- *Сканеры поисковых машин* (web crawlers). Для поддержки баз данных в актуальном состоянии поисковые машины должны периодически проверять наличие обновлений страниц доступных веб-сайтов. Для обработки

потока текстов веб-страниц должны использоваться соответствующие средства.

Особый интерес представляют сайты, ориентированные на принципы "Веб 2.0" [20] (то есть использующие содержание, которое создаётся самими пользователями таких сайтов), или *социальные медиа-ресурсы* (social media). Основными разновидностями социальных медиа являются следующие категории веб-сайтов [2]: вики, блоги ("Живой Журнал"), микроблоги (Twitter), социальные новости, сайты отзывов и обзоров, сайты ответов на вопросы, сайты обмена медиа-контентом (YouTube), социальные закладки, социальные сети.

1.1.3. Отличия методов обработки информационных потоков от традиционных методов интеллектуального анализа данных

Ключевыми особенностями, отличающими алгоритмы обработки потоков данных от алгоритмов, действующих на статических коллекциях, являются [21, с. 43]:

1. Высокая скорость поступающих данных. Скорость вычисления параметров модели должна быть выше, чем скорость поступления данных. При этом становится затруднительным использование сложных алгоритмов обработки информации.
2. Требование неограниченной памяти. Зачастую обработать все данные потока без применения распределённых вычислений невозможно. Ряд методов основываются на построении выборок сообщений из потоков.
3. Изменение содержания понятий с течением времени (*дрейф понятий*, concept drift), или *эволюция потока* (stream evolution). Модель классификатора должна учитывать изменение данных, связанных с классами, иначе результаты классификации могут быть менее точными или ложными.

Высокая скорость поступления данных, в первую очередь, приводит к необходимости компромисса между точностью и эффективностью применяемых алгоритмов. Вследствие этого в задачах обработки потоков к алгоритмам предъявляется требование *оперативности* (on-line) — способности обрабатывать данные по мере их поступления ("в масштабе реального времени").

В ряде практических приложений скорость поступления данных отражается на требованиях к аппаратным средствам. Потоковые данные и результаты их обработки часто передаются по низкоскоростным каналам связи. Эта проблема, в частности, касается применения алгоритмов в мобильных приложениях, а также в случае сжатия данных и распределения вычислений.

Ограничения на память приводят к другому требованию, предъявляемому к алгоритмам обработки потоков — *итеративности,* которое означает способность обучаться без необходимости повторной обработки всей коллекции текстов.

В условиях дефицита вычислительных мощностей и памяти становится также важным оценивать эффективность и точность алгоритма обработки потока в процессе его работы, чтобы иметь возможность подстройки его параметров под требования решаемой задачи и имеющиеся ресурсы.

Вследствие дрейфа понятий в некоторых задачах становится важным не извлечение информации из самих данных, а наблюдение и анализ тенденций изменения данных. В дополнение к задаче обнаружения возникающих трендов при обработке потоков текстов ставятся задачи *обнаружения и отслеживания тематик* (topic detection and tracking, TDD — идентификации новых тем по мере поступления сообщений) и анализа эволюции потоков (evolution analysis — исследование динамики отдельных тем, а также их взаимодействий, например, корреляций [20], с течением времени).

Задачи обработки потоков часто связаны с областями, где от пользователя требуется быстрое принятие решений, что предъявляет дополнительные требования к эргономичности пользовательского интерфейса, подсистеме визуализации промежуточных и окончательных результатов и средствам поддержки принятия решений. Ряд прикладных задач требует также возможности изменения пользователем параметров классификации данных. В условиях быстро поступающих и изменяющихся потоковых данных обеспечение подобной интерактивности становится сложной задачей.

1.2. Анализ подходов к обработке текстовых сообщений

1.2.1. Классический подход к обработке естественного языка

Классический подход к анализу текстов основывается на представлении процесса обработки в виде ряда последовательных этапов. Истоком такого разделения является учение Ч. Морриса о трёх составляющих семиотики: синтактике, изучающей отношения между знаками, семантике, изучающей отношения знаков к объектам действительности, и прагматике, изучающей отношения знаков к их интерпретаторам [22]. Для задач ОЕЯ чаще всего применяется более полная декомпозиция, включающая следующие этапы обработки текста [5; 23]:

- *Предварительная обработка текста*, заключающаяся в преобразовании текста к виду, пригодному для машинной обработки.
- *Морфологический анализ*, который заключается в приведении слов к исходной форме и установлении их лингвистических ролей в предложении.
- *Синтаксический анализ*, задачей которого является описание структуры предложения согласно некоторой формальной грамматике [24].
- *Семантический анализ* [25, с. 21–27], задачей которого является уточнение связей между лексическими единицами, которые не были выявлены при синтаксическом анализе (так как многие лингвистические роли выражаются с учётом значения слова), а также подготовка текста для нечёткой обработки (определение значений лингвистических переменных).
- *Прагматический анализ*, заключающийся, в зависимости от задачи, в а) исследовании связности текста и динамики контекста; или б) исследование целей участников общения (диалоговой ситуации). Прагматические знания о предметной области и текущей диалоговой ситуации чаще всего применяются при построении вопросно-ответных систем и обучающих сред [26].

Этап предварительной обработки текста иногда называют графематическим анализом, морфологический анализ — лексическим анализом, семантический анализ — концептуальным анализом [27], а прагматический анализ — дискурс-анализом [28]. В синтаксическом анализе иногда выделяют предсинтаксический и постсинтаксический этапы [23, с. 107].

Заметим также, что соседствующие этапы часто оказываются связанными друг с другом. Например, для русского языка синтаксический анализ тесно сопряжён с морфологической информацией о словах.

С точки зрения ИАТ и поверхностной ОЕЯ методы морфологического, синтаксического и начального семантического анализа относятся к задачам предварительной подготовки текста (за которым сразу следует этап анализа данных) [11, с. 57–63][2].

1.2.2. Базовые методы интеллектуального анализа текстов

Общие методы классификации с учителем (собственно классификации) и без учителя (кластеризации) текстов изучены были адаптированы ИАТ из области машинного обучения и изучены довольно подробно [29]. Общепринятой категоризации этих алгоритмов не существует. В соответствии с одной из схем, алгоритмы классификации можно разбить на статистические и структурные (см. Рисунок 1, адаптировано из [16, с. 242]). В свою очередь, статистические можно подразделить на регрессионные и байесовские, а структурные методы подразделяются на: алгоритмы на основе правил, алгоритмы на основе расстояний и нейронные сети.

[2] Этап предварительной обработки текста сообщения в потоке, описанный на с. 33, рассматривается именно с этой точки зрения.

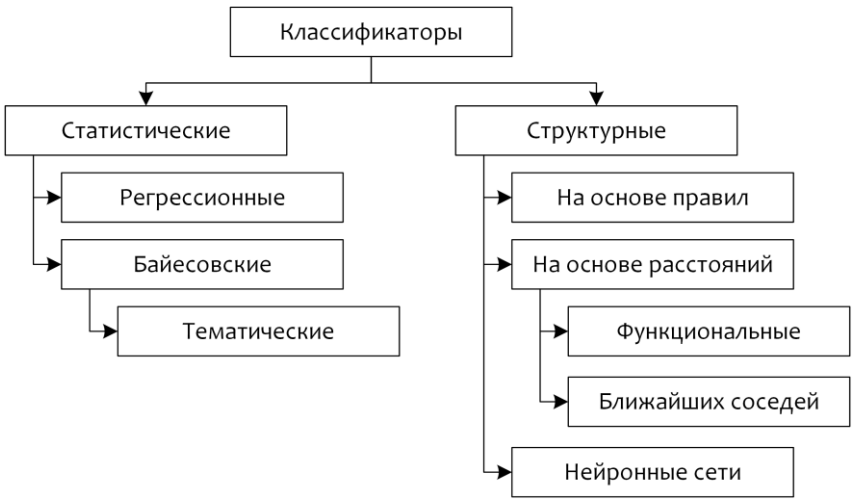


Рисунок 1. Категоризация алгоритмов классификации

Регрессионные методы классификации

Цель регрессионных методов — предсказание значения зависимой (выходной) непрерывной переменной. Регрессионные методы являются одними из наиболее сильных и универсальных методов, применяемых для решения широкого круга задач. Обучение модели заключается в нахождении коэффициентов соответствующего уравнения [30, с. 345]. Наиболее известными разновидностями данного метода являются линейная регрессия и логистическая регрессия. В то время как линейная регрессия чаще всего применяется для задач прогнозирования, логистическая регрессия находит применение и в системах обработки естественного языка [31, с. 10]. Для обучения моделей логистической регрессии используется метод максимального правдоподобия [30, с. 406].

Баейсовские методы классификации

К байесовским методам классификации относятся алгоритмы, основанные на использовании формулы Байеса для вычисления апостериорной вероятности некоторого класса c при наблюдении множества признаков F и известной априорной вероятности класса c:

$$P(c|F) = \frac{P(F|c)P(c)}{P(F)} \quad (1)$$

Наиболее простые модели вероятностных классификаторов, такие как наивный байесовский классификатор, является также наименее сложными (на

этапе обучения) с алгоритмической точки зрения. Кроме того, выходами вероятностных классификаторов являются оценки в интервале $[0,1]$, что позволяет применять их в задачах мягкой (soft) и многозначной (multi-label) классификации.

Классификация на основе правил

Продукционные правила, или правила вида "ЕСЛИ (условие) — ТО (действие)" — наиболее распространённый способ представления знаний [32, с. 22]. В случае классификации условием (антецедентом) правила является набор признаков рассматриваемого объекта, а действием (консеквентом) — результирующий класс или указатель на следующее правило.

Частными случаями продукционной модели, позволяющими эффективно организовать процесс логического вывода, являются *деревья решений* (decision trees, см. п. 1.2.3) [33] и *таблицы решений* (decision tables) [34]. Как правило, продукционные правила формируются вручную инженером-когнитологом на основе опыта эксперта, однако существуют методы автоматической генерации правил на основе данных (см. [35; 36, с. 95–97] и п. 1.2.3). В то же время формирование правил для обработки текстов сталкивается с проблемой большого количества классификационных признаков, а также их слабой связанностью.

Алгоритмы на основе расстояний

Алгоритмы классификации на основе расстояний (называемые также методами классификации в векторном пространстве) оперируют представлением документов в виде векторов действительных чисел. В основе использования такой модели лежит гипотеза компактности (contiguity hypothesis): "документы, принадлежащие одному и тому же классу, образуют компактную область, причём области, соответствующие разным классам, не пересекаются" [37, с. 295]. Векторы документов представляют собой нормализованные по длине единичные векторы из пространства $\mathbb{R}^{|V|}$, где V — лексикон языка, координаты которых лежат на поверхности гиперсферы [37, с. 297].

Наиболее распространёнными методами классификации на основе расстояний являются: метод Роккио, метод k ближайших соседей (k-nearest neighbors, kNN) и метод опорных векторов (support vector machine, SVM).

Последний из перечисленных методов является одним из наиболее точных алгоритмов, применяющихся для классификации текстов. Его недостатки связаны с относительно большой алгоритмической сложностью обучения и невозможностью применения в задачах многоклассовой классификации без существенной модификации [38].

Искусственные нейронные сети

Искусственные нейронные сети, несмотря на простоту математической модели единичного нейрона, доказали свою эффективность при решении задач, в которых формальные символьные методы и модели неприменимы или их использование нецелесообразно. В то же время в задачах ОЕЯ нейронные сети используются сравнительно редко, что объясняется популярной точкой зрения, согласно которой никакие лингвистические аспекты не могут быть в полной мере смоделированы коннекционистскими методами [39].

С другой стороны, в задачах ИАТ нейронные сети могут выступать в качестве одного из этапов обработки данных [40; 41], например, для оценки значимости фрагментов текста или классификации. В то же время подходы, основанные на использовании нейронных сетей, нуждаются в разработке обоснованной архитектуры сети и выборе методологии их обучения.

Тематическое моделирование

Помимо общих методов классификации специально для обработки больших коллекций текстовых документов были разработаны методы, основанные на многоуровневых байесовских вероятностных сетях и учитывающие скрытую структуру текста — так называемые методы тематического моделирования (topic modeling). К этой группе относятся, например, вероятностный латентно-семантический анализ (probabilistic latent semantic analysis, PLSA) [42], латентное размещение Дирихле (latent Dirichlet allocation, LDA) [43] и их расширения [13; 44].

Ансамбли классификаторов

Для повышения точности моделей ИАТ можно воспользоваться идеей агрегирования набора классификаторов [30, с. 543]. Классификаторы в этом случае объединяются параллельно (на вход подаются исходные данные, данные с выходов комбинируются друг с другом). Наиболее распространёнными методами формирования ансамблей являются бэггинг (метод обучения множества классификаторов, при котором на их вход подаются различающиеся выборки из исходных данных) и бустинг (итерационный метод, при котором на каждом новом этапе процесса обучения создаётся новый классификатор, учитывающий ошибки классификации предыдущей модели).

1.2.3. Иерархические методы машинного обучения

Баейсовские сети

Расширением простых вероятностных методов являются байесовские сети (БС). БС — это "ориентированный граф, в котором каждая вершина помечена количественной вероятностной информацией" [45, с. 661]. Вершины представляют собой дискретные или непрерывные случайные переменные, соответствующие понятиям предметной области. Дуга от вершины x_1 к вершине x_2 характеризует то, насколько сильно влияет событие x_1 на событие x_2; в формальном смысле, дуга соответствует условной вероятности $P(x_2|x_1)$. Вывод (в том числе для классификации) на байесовских сетях заключается в вычислении для каждой вершины сумм произведений "входящих" условных вероятностей.

Основное ограничение данного класса моделей заключается в том, что точный вероятностный вывод на БС является NP-полной задачей [45, с. 682]. Более того, для многих практических приложений требуется включение в БС непрерывных случайных переменных, что существенно усложняет вычисления.

Иерархическая кластеризация

Иерархическая кластеризация (hierarchical clustering) представляет собой группу методов обучения без учителя, в которых для разграничения объектов строится упорядоченное множество кластеров (классов), связанных

между собой отношениями "общее — частное". С одной стороны, такое множество кластеров является более информативным и наглядным, чем то, что получается в ходе "плоской" кластеризации; с другой стороны, многие исследователи полагают, что это способствует повышению качества (но не обязательно скорости) кластеризации [37, с. 379].

Существует два основных подхода к иерархической кластеризации.

При *нисходящей* (top-down, или разделяющей, дивизивной, divisive) кластеризации объекты изначально принадлежат одному кластеру. Затем этот кластер с помощью одного из стандартных алгоритмов "плоской" кластеризации разделяется на несколько. Этот процесс повторяется рекурсивно для каждого кластера до тех пор, пока каждому документу не будет соответствовать один кластер [37, с. 396].

При *восходящей* (агломеративной, agglomerative) кластеризации каждый документ изначально принадлежит отдельному кластеру. Затем кластеры оцениваются по некоторой заданной метрике сходства, и пары наиболее схожих кластеров объединяются. Процедура повторяется последовательно, пока все кластеры не сольются в один [37, с. 380].

Деревья решений

Деревья решений — это "способ представления правил в иерархической, последовательной структуре" [36, с. 93]. В каждой нелистовой вершине дерева в общем случае производится сравнение некоторой функции от параметров антецедента (независимой переменной) с константой, интервалом или категорией. Каждому возможному результату сравнения соответствует выходящая из вершины дуга. Листовые вершины обозначают результаты классификации (значения зависимой переменной).

Как и продукционные правила, деревья решений строятся либо на основе опыта эксперта, либо генерируются специальными алгоритмами, такими как [30, с. 437–465; 36, с. 100–112]: "разделяй и властвуй", ID3, C4.5, алгоритм покрытия, CART. В качестве примера специализированного алгоритма построения дерева на основе существующих правил можно привести алгоритм Rete [46].

Концептуальная кластеризация

Алгоритмы вероятностной и векторной кластеризации, рассмотренные выше, "не учитывают цели и базовые знания ... и не обеспечивают осмысленное семантическое обоснование сформированных категорий" [47, с. 418]. С точки зрения инженерии знаний получаемый кластер является интенсионалом понятия. С практической точки зрения это означает, что кластер может неточно описывать объект предметной области (либо вообще не соответствовать какому-либо объекту). Алгоритмы концептуальной кластеризации (conceptual clustering), разрабатывавшиеся, преимущественно, в 80-х годах, были призваны решить эту проблему за счёт создания экстенсионального [48] описания понятия (concept description) для каждого кластера. Классическими алгоритмами концептуальной кластеризации являются COBWEB [47, с. 419–424; 49] и CLUSTER/2.

1.2.4. Оценка эффективности методов классификации текстов и экспериментальные коллекции документов

Метрики эффективности классификации текстов

Чаще всего для оценки эффективности алгоритмов бинарной классификации текстов применяются три базовых показателя: аккуратность, точность и полнота [29, с. 188–189; 37, с. 168–169]. Для бинарных классификаторов они вычисляются следующим образом.

Пусть имеется бинарный классификатор, определяющий принадлежность объекта к классу A. Пусть N_{tp} — количество объектов в контрольной выборке, для которых классификатор дал правильный ответ (отнесённых и принадлежащих к классу A, истинно-положительных результатов); N_{fp} — количество объектов, отнесённых к классу A, но фактически ему не принадлежащих (ложноположительных результатов, ошибок первого рода, "ложных тревог"); N_{fn} — количество объектов, принадлежащих к классу A, но не отнесённых к нему (ложноотрицательных результатов, ошибок второго рода, "пропусков цели"); N_{tn} — количество объектов, не отнесённых к классу A и не принадлежащих к нему (истинно отрицательных результатов).

Тогда аккуратность (правильность, accuracy) — это доля правильных ответов классификатора: $T = (N_{tp} + N_{tn})/(N_{tp} + N_{fp} + N_{fn} + N_{tn})$; точность (precision) — доля правильно обнаруженных объектов среди отнесённых к классу A: $P = N_{tp}/(N_{tp} + N_{fp})$; полнота (recall) — доля правильно обнаруженных объектов среди принадлежащих к классу A: $R = N_{tp}/(N_{tp} + N_{fn})$.

Экспериментальные коллекции документов

Для оценки методов классификации текстов за годы исследований по ИАТ были созданы стандартные наборы данных, представляющие собой множества (коллекции) текстовых документов с приписанными им классами. Часто используемыми коллекциями для классификации текстов являются [37, с. 167–168]:

- "Reuters-21578" и её обновление "Reuters-RCV1";
- коллекции Конференции по оценке поиска текстов (TREC);
- корпус "20 Newsgroup".

Перечисленные выше коллекции составлены из документов на английском языке. Для большинства стандартных методов ИАТ и поверхностного ОЕЯ язык текста сообщения не играет значимой роли. В то же время некоторые модификации алгоритмов могут включать отдельные этапы, ориентированные на обработку текстов на конкретном языке.

Для оценки методов классификации текстов на русском языке могут использоваться:

- коллекции Российского семинара по оценке методов информационного поиска (РОМИП)[3];
- отдельные корпуса текстов в рамках проекта "Национальный корпус русского языка"[4] и аналогичным ему[5].

[3] http://romip.ru/ru/collections/index.html
[4] http://www.ruscorpora.ru/corpora-structure.html
[5] http://www.ruscorpora.ru/corpora-other.html

Методика оценки алгоритмов классификации

Для малочисленных наборов экспериментальных данных могут применяться методы перекрёстного контроля (скользящего контроля [50], кросс-валидации, cross-validation, CV). Классический метод перекрёстного контроля проиллюстрирован на Рисунок 2. Вся выборка объектов, для которых определены принадлежности к классам, разбиваются на n частей. Далее проводится n итераций: в k-ой итерации обучающим множеством являются части $\{1,2,\ldots,k-1,k+1,\ldots n\}$, а в качестве контрольного множества выступает часть k.

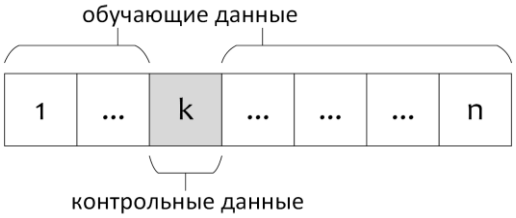


Рисунок 2. Схема перекрёстного контроля для алгоритмов классификации

1.2.5. Представление лингвистических данных

Обработка текстов непосредственно связана с представлением как самого текста как объекта обработки, так и промежуточными и конечными результатами его анализа вычислительной системой. Выбор представления данных о тексте зависит от решаемой задачи, причём очень часто для каждой задачи разрабатывается свой метод представления, позволяющий достичь определённых преимуществ при обработке текста.

Х. Каннингхэм предложил классифицировать методы представления метаинформации о тексте следующим образом [51]:

- аддитивные (additive) — метаинформация об элементах текста хранится внутри текста в виде разметки (markup);
- ссылочные (referential) — метаинформация со ссылками на исходный текст хранится отдельно от этого текста;
- абстрактные (abstraction-based) — исходный текст не сохраняется, элементы текста хранятся в некоторой определённой задачей структуре данных.

Аддитивные методы также называются методами на основе разметки (markup-based) или методами на основе встроенных аннотаций (in-line annotations), а ссылочные методы — методами на основе внешних аннотаций (stand-off annotations) [52, с. 2].

Методы на основе аннотаций

Использование встроенных аннотаций очень удобно для многих задач, связанных с последовательной обработкой текста (особенно при классическом подходе в ОЕЯ). Встроенные аннотации позволяют легко анализировать результаты обработки пользователем системы, особенно при использовании специальных методов визуализации.

Большим недостатком языков разметки является то, что они представляют древовидные структуры метаданных, в то время как для обработки текста часто требуются сетевые структуры [53, с. 145]. Для преодоления этого недостатка различными группами исследователей было предложено множество модификаций языков разметки [54].

Внешние аннотации представления хранят метаинформацию об элементах текста отдельно от него и ссылающуюся на его участки. Это позволяет описывать пересекающиеся и сложные языковые структуры без риска конфликта аннотаций. Кроме того, процесс аннотирования при этом подходе может быть разделён на несколько независимых этапов.

Семантические сети и онтологии

Онтология — "система, состоящая из набора понятий и набора утверждений об этих понятиях, на основе которых можно строить классы, объекты, отношения, функции и теории" [55, с. 9]. Методы онтологического моделирования и учёта семантической метаинформации о предметной области считаются важным направлением при построении информационных систем [56; 57]. Применение этих методов позволяет повысить эффективность обработки информационных ресурсов большого объема и предоставляет дополнительные возможности для задач, связанных с информационным поиском, поддержкой принятия решений [58] и др.

Знания о предметной области могут быть представлены на базе [59, с. 21–26]: продукционных правил, фреймов, семантических сетей или формальных логических моделей. Форма представления знаний зависит от задачи. Знания в системах поддержки принятия решений и экспертных системах удобно представлять в виде продукционных правил (см. п. 1.2.2). Лингвистическую информацию о предметной области удобно представлять в виде семантической сети — множества понятий и отношений между понятиями.

1.3. Подходы и алгоритмы обработки потоков данных

1.3.1. Анализ и прогнозирование временных рядов

Временным рядом называется совокупность наблюдений некоторой величины (как правило, обозначаемая Y), выполненных в течение некоторого промежутка времени [60, с. 191]. Типичный временной ряд можно представить четырьмя компонентами [60, с. 193]:

- тренд (T) — компонента, представляющая основное изменение (рост или спад) временного ряда;
- цикличность (C) — последовательность волнообразных флуктуаций, имеющих большой период цикличности (в приложении к экономическим задачам — несколько лет); на практике цикличность часто включается в тренд, так как эти компоненты очень сложно разделить;
- сезонность (S) — изменения с более или менее стабильной структурой, имеющие годовую цикличность;
- нерегулярность (I) — компонента, включающая непредсказуемые или случайные флуктуации, которые являются результатом множества разнообразных событий.

Согласно [61], по оценкам зарубежных и отечественных исследователей, насчитывается свыше 100 методов прогнозирования. Для прогнозирования экономических рядов на практике чаще всего применяются [62]:

- *наивные модели* — используются для построения прогноза на основании последних данных о явлении [60, с. 134–137]; данные методы полезны,

если требуется выполнить прогноз в условиях отсутствия большого количества наблюдаемых данных;

- *методы экспоненциального сглаживания* с учётом тренда: Хольта, Брауна, Винтерса, Тейла — Вейджа;
- методы на основе *декомпозации временного ряда* предполагают построение модели ряда в виде отдельных компонент T, C, S, I;
- *регрессионные методы*, включающие линейную, логистическую и многомерную регрессию (см. п. 1.2.2);
- *адаптивные методы Бокса — Дженкинса* (АРСС, АРПСС), которые позволяют ограничиться заданием общего класса модели ряда, после чего алгоритм сам подстроит внутренние параметры и выберет модель прогнозирования;
- *нейросетевые методы* (см. п. 1.2.2) являются наиболее общей моделью прогнозирования среди рассмотренных; они предоставляют "дополнительные возможности в моделировании нелинейных явлений и распознавании хаотического поведения" [63, с. 222].

В некоторых случаях, например, для упорядоченных во времени последовательностей экспертных оценок некоторой величины, можно говорить о нечётком временном ряде [64, с. 28]. Понятие *нечётких временных рядов* (fuzzy time series) было введено и разработано Ц. Суном и Б. Чиссомом [65] на основе аппарата нечётких множеств и лингвистических переменных, разработанного Л. Заде.

1.3.2. Обработка информационных потоков

Область прикладных исследований, занимающаяся вопросами анализа содержания множеств сообщений, поступающих на вход системы, называется *обработкой информационных потоков* (ОИП, information flow processing, IFP [66]). Входными моделями для систем ОИП могут являться продукционные правила или структурированные запросы (structured queries), учитывающие распределение данных во времени [67].

В последнее время ОИП тесно связывается с понятием *обработки событий* (event processing). В рамках ОИП событием называется сообщение, содержащее релевантные для получателя знания о том, что-то в наблюдаемой системе произошли изменения.

Существует 4 подхода к обработке событий в системах ОИП:

- *обработка простых событий* (simple event processing, SEP) — реакция системы формируется на основании анализа последнего сообщения;
- *управление потоками данных* (data stream management, DSM) — реакции системы основана на выполнении *постоянных запросов* (continuous queries) над базой данных, которая постоянно пополняется новыми сообщениями;
- *обработка потоков событий* (event stream processing, ESP) — подход ориентирован на анализ характеристик последовательно поступающих сообщений (потоков событий, event streams);
- *обработка сложных событий* (complex event processing, CEP [68]) — реакция системы основана на анализе *облаков событий* (event clouds) — множеств неупорядоченных сообщений, значимость которых по отдельности не определена.

В рамках ОИП к событиям могут быть применены следующие операции [69, с. 34]: сжатие, агрегация, обобщение, уточнение, фильтрация, гашение (suppression), эскалация; пороговая обработка, логические операции, темпоральные операции, ограничение по скорости поступления.

Системы ОИП часто используются совместно с *машинами исполнения бизнес-правил* (business rule engine, BRE) в системах *управления бизнес-процессами* (business process management, BPM), где событиями, в частности, могут выступать:

- изменения наблюдаемых величин (например, цен на продукты);
- данные с сенсоров (например, активация RFID-меток);
- результаты работы сторонних программ;
- действия пользователей (клиентов, управляющего персонала) и др.

Вследствие этого системы ОИП для описания правил обработки событий часто предоставляют собственные предметно-ориентированные языки

(domain-specific language, DSL), приближенные к естественным (например, английскому) или структурированным (например, SQL). Некоторые системы включают средства для графического описания процесса обработки сообщений (например, с помощью нотации BPMN).

1.3.3. Интеллектуальный анализ последовательностей

Во многих практических приложениях данные поступают в виде последовательностей — кортежей связанных элементов. Примерами последовательностей являются [70, с. 2–4]:

- цепочки ДНК (кортеж: {аминокислота, …, аминокислота});
- журналы посещений веб-страниц пользователями
  (кортеж: {*ip* адрес пользователя, страница});
- последовательности покупок
  (кортеж: {имя пользователя, сумма, город, товар}).

Типичными задачами интеллектуального анализа последовательностей (ИАП, sequence data mining) являются: классификация и кластеризация, анализ признаков и расстояний, ассоциация, коллаборативная фильтрация.

- Классификация и кластеризация [70, с. 47] последовательностей могут осуществляться на основе стандартных алгоритмов машинного обучения.
- *Анализ признаков* последовательностей [70, с. 48] связан с выявлением в кортежах тех элементов (или характеристик их совокупности, например, частоты некоторого элемента), которые существенны для решаемой задачи.
- Задача *анализа расстояний* заключается в количественном определении сходства и различия последовательностей [70, с. 52].
- Задача *ассоциации* заключается в выявлении частых последовательностей; широко распространённым частным случаем является *анализ покупательской корзины* (market basket analysis) [30, с. 281]. Целью алгоритмов ассоциации является описание закономерностей в последовательностях в явном виде с помощью множества правил. Классическим алгоритмом ассоциации является Apriori [71].

- *Коллаборативная (совместная) фильтрация* заключается в выработке прогнозов и предложений в рекомендательных системах [16, с. 113].

В ряде практических приложений возникают специальные задачи, связанные с обработкой потоков:

- В области интеграции программных приложений часто требуется обеспечить маршрутизацию потока управляющих сообщений в зависимости от информации (команд или данных), которую они содержат. Соответствующая процедура называется *маршрутизацией, основанной на содержании* (content-based routing).
- В области телекоммуникаций и управления компьютерными сетями известна задача диагностики и *локализации отказов* (fault localization) [72], включающая обработку и сопоставление данных из большого количества сигнальных сообщений.
- Обобщением методов, применяемых при локализации отказов, является процедура *корреляции событий* (event correlation), суть которой заключается в анализе множества проявлений ("симптомов") некоторого негативного явления с целью объяснения причин нарушений [73].
- Термин "корреляция событий" применяется в отношении методов автоматического поиска нарушений. Для "ручного" и автоматизированного анализа нарушений чаще всего используется другой термин — *поиск первопричин* (root cause analysis, RCA) [74].

Классификация методов анализа последовательностей формальных сообщений приведена на Рисунок 3 (адаптировано из [69]).

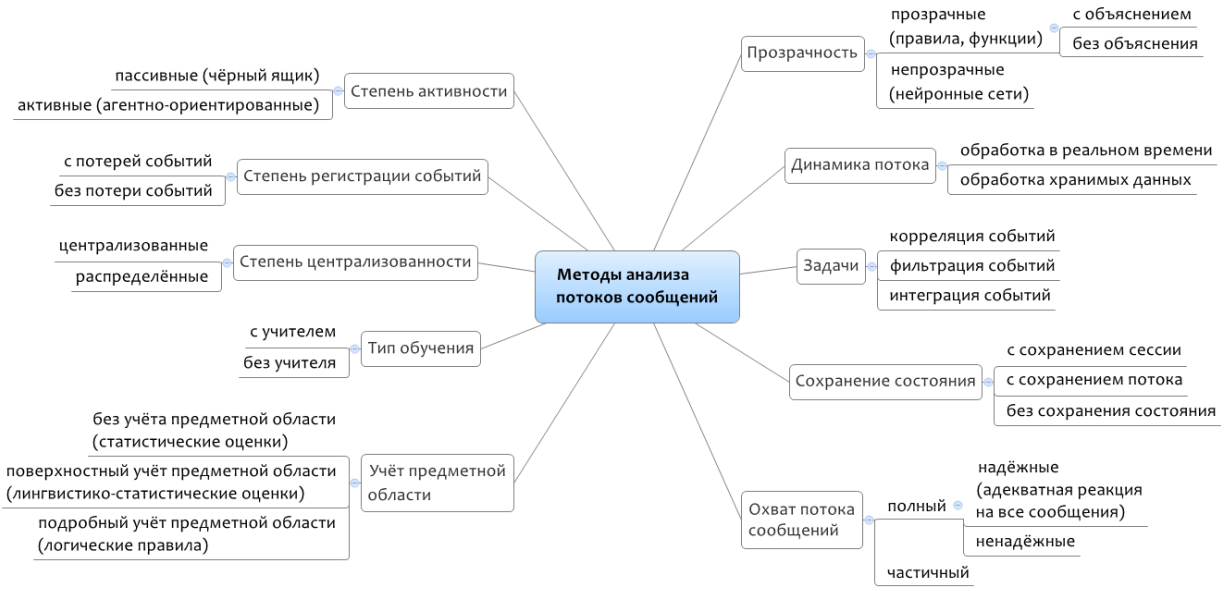


Рисунок 3. Классификация методов анализа потоков формальных сообщений

1.4. Средства и системы обработки текстовой информации

1.4.1. Системы обработки естественного языка

Классификация систем обработки естественного языка

В настоящее время существует огромное количество систем, ориентированных на обработку текстовой информации. Эти системы можно классифицировать следующим образом [75, с. 53–81]:

- системы подготовки/редактирования текста (text preparation) (автоматическая расстановка переносов — automatic hyphenation, проверка орфографии — spell checking, проверка грамматики, проверка стиля, управление ссылками — referencing);
- системы информационного поиска (information retrieval) и поисковые машины (search engines);
- системы автоматического реферирования (topical summarization);
- системы машинного перевода (automatic translation);
- подсистемы естественно-языкового интерфейса (natural language interfaces) к базам данных и другим системам, вопросно-ответные системы (question-answering systems);
- системы извлечения фактов (extraction of factual data) из научных и деловых текстов;

- системы генерации текста (text generation) на основе графических или формальных данных;
- системы оптического распознавания символов (optical character recognition, OCR) и рукописного текста (handwriting recognition);
- системы распознавания речи (speech recognition) и синтеза речи (speech synthesis);
- системы управления онтологиями;
- системы понимания естественного языка (natural language understanding).

Лингвистическая обработка текстов

Stanford NLP Toolkit[6] — открытый (open-source) комплекс библиотек для обработки текстов на естественном языке, разработанный в Стэндфордском университете. Реализует множество методов для морфологического и синтаксического анализа, преимущественно, текстов на английском языке; частично поддерживает арабский, немецкий и китайский языки. Предоставляет базовые средства классификации (байесовские и регрессионные методы).

Apache OpenNLP[7] — открытый комплекс, функционал которого аналогичен предыдущему; основное отличие заключается в возможности обучения морфологических и синтаксических анализаторов по примерам (размеченным пользовательским текстам). Поддерживает интеграцию с Apache UIMA (см. ниже).

Apache UIMA[8] (Unstructured Information Management Architecture, архитектура управления неструктурированной информацией) — открытая инфраструктура для обработки и извлечения знаний из больших массивов текстов. Предоставляет компонентный подход к построению систем контент-анализа. Основной функционал системы — аннотирование текстов.

[6] http://nlp.stanford.edu/software/index.shtml
[7] http://opennlp.apache.org/
[8] http://uima.apache.org/

GATE[9] (General Architecture for Text Engineering, обобщённая архитектура инжиниринга текстов) — открытый комплекс средств для обработки текстов. Включает средства для семантических аннотаций, извлечения знаний, работы с онтологиями, машинного обучения. Ориентирован на работу с английским языком.

Проект **AOT**[10] ("Автоматическая обработка текстов") [76] — рабочая группа, занимающаяся исследованиями по обработке текстов на русском языке. В рамках проекта было создано множество словарей и открытых модулей для морфологического, синтаксического, семантического анализа для русского и, частично, английского и немецкого языков; были разработаны системы машинного перевода ("Политекст" и "Диалинг").

Интеллектуальный анализ текстов

RapidMiner[11] — открытая интегрированная среда, предназначенная для ИАД, преобразования данных (extract-transform-load, ETL), аналитической обработки в реальном времени (online analytical processing, OLAP). Средства анализа текстов включают: базовую обработку текста, анализ с использованием векторных методов, вычисление сходства, основные методы классификации и кластеризации. Включает базовые средства анализа потоков данных, основанные на методе скользящего окна.

Apache Mahout[12] — открытая инфраструктура для построения распределённых масштабируемых систем машинного обучения. Функциональность системы в настоящий момент ограничена, но включает некоторые редко рассматриваемые методы ИАД; например, данная система предоставляет эффективную реализацию метода LDA (см. п. 1.2.2, тематическое моделирование).

WordStat[13] — проприетарная система контент-анализа больших статических массивов текстов (новостных статей, научных публикаций, коммерческих отчётов и др.). Включает средства извлечения фактов, базовые методы

[9] http://gate.ac.uk/
[10] http://www.aot.ru/
[11] http://sourceforge.net/projects/rapidminer/
[12] http://mahout.apache.org/
[13] http://provalisresearch.com/products/content-analysis-software/

классификации и кластеризации; визуализации результатов для поддержки принятия решений.

IBM Content Analytics[14] — проприетарная система анализа неструктурированной информации. Ориентирована на применение внутри корпоративных информационных систем, однако позволяет определять и внешние источники данных и текстов. Функционал системы аналогичен предыдущей.

Transinsight ESI[15] (Enterprise Semantic Intelligence) — проприетарная система семантического поиска и управления знаниями на предприятии. Включает компоненты ИАТ (преимущественно, для извлечения фактической информации), семантического индексирования, управления онтологиями, визуализации данных и знаний. Ориентирована только на внутренние массивы информации (интранет).

1.4.2. Системы обработки потоковых данных

Интеллектуальный анализ потоков

MOA[16] (Massive On-line Analysis) — инфраструктура для исследования эволюции потоков данных. Включает средства ИАП и некоторые специальные алгоритмы кластеризации (например, COBWEB), а также позволяет дополнять и развивать существующие алгоритмы.

ADWIN[17] (Adaptive Stream Mining) [77] — исследовательская классификации и кластеризации данных, поступающих в потоковом режиме и изменяющихся с течением времени. Система основана на использовании стандартных алгоритмов ИАД и концепции адаптивного скользящего временно́го окна, что позволяет установить факт изменений в потоке и соответствующим образом скорректировать получаемую статистику. Реализация выполнена на базе системы MOA.

TwitIE [78] — система извлечения фактов из сообщений, публикуемых пользователями сети микроблогов Twitter. Система использует в своей работе

[14] http://www.ibm.com/software/products/en/subcategory/SWN40
[15] http://transinsight.com/
[16] http://moa.cs.waikato.ac.nz/
[17] http://adaptive-mining.sourceforge.net/?page_id=20

инфраструктуру GATE и модули Stanford NLP Toolkit. Особенностями системы, обусловленными предметной областью, являются: работа с короткими сообщениями ("твитами"), исправление орфографических ошибок, расшифровка синонимов, сокращений и аббревиатур.

Обработка событий

JBoss Drools[18] — открытый комплекс средств ОИП, включающий систему исполнения бизнес-процессов (jBPM), машину исполнения правил (Expert), систему моделирования временны́х рассуждений (Fusion), систему автоматизированного планирования (OptaPlanner) и репозиторий знаний (Guvnor). Система позволяет описывать правила на предметно-ориентированном языке, близком к естественному. Функции стандартны для систем ОИП.

Esper[19] — открытая библиотека обработки событий. Функции стандартны для систем ОИП. Система позволяет описывать правила обработки событий на языке, близком к SQL.

Oracle CEP[20] — проприетарная система обработки событий. Функции стандартны для систем ОИП. Имеются встроенные адаптеры для считывания сообщений, описывающие события, из очередей JMS и HTTP.

Анализ трендов

Attensity[21] — проприетарная система поддержки принятия маркетинговых решений и социальной аналитики, предназначенная для извлечения фактов, зависимостей и мнений (sentiment) из неструктурированных массивов текстов, преимущественно, в социальных сетях. Ключевые функции системы включают: анализ мнений о компаниях и продуктах, оценка "успешности" появления продукта на рынке, анализ рисков и потенциальных проблем, прогнозирование трендов. Система обладает развитыми средствами визуализации результатов анализа.

[18] http://www.jboss.org/drools/drools-fusion.html
[19] http://esper.codehaus.org/
[20] http://www.oracle.com/technetwork/ru/middleware/complex-event-processing/index.html
[21] http://www.attensity.com/home/

Google Zeitgeist[22] и **Google.Тренды**[23] — проприетарные системы визуализации и ранжирования трендов в поисковых запросах, поступающих в поисковую машину Google. Позволяют категоризировать интересующие результаты по странам и тематикам; отслеживать и оценивать пользовательские поисковые тренды.

Calliope Text Mining Suite[24] — проприетарная система анализа эволюции понятий в коллекциях текстов, включающая средства извлечения терминологии, индексирования, анализа совместной встречаемости слов (co-word analysis), визуализации графов понятий и динамики терминов с течением времени. Ориентирована на работу с английским, французским, немецким и испанским языками; лингвистическая поддержка остальных языков ограничена.

ARCOMEM[25] — проект по исследованию средств развития институтов памяти (memory institution, или репозиториев общественного знания — repository of public knowledge, RPK — библиотек, архивов, музеев — преимущественно, электронных и основанных на принципах социальных медиа). Основной целью проекта является: улучшение доступности знаний категориям пользователей, на которых они ориентированы, за счёт применения средств Веб 2.0. В рамках проекта был создан ряд систем, ориентированных на анализ влияния общественных событий на социальные медиа-ресурсы (ARCOMEM Crawler Cockpit, ARCOMEM SARA, TwikiMe, TwitIE и др.).

TrendMiner[26] — исследовательская система анализа трендов, ориентированная на поддержку принятия решений в экономике и политике на основе комплексного анализа текстов из социальных медиа и данных по продажам. Включает средства многоязычной обработки текстов и интеллектуального анализа временных рядов.

ИДК-ОС ("Интерактивная дорожная карта с обратной связью") [79] — исследовательская система, построенная на основе онтологической модели технологических трендов. Включает средства глубинного ОЕЯ, извлечения терминологии, статистической обработки, прогнозирования и визуализации.

[22] http://www.google.com/zeitgeist/
[23] http://www.google.com/trends/
[24] http://www.calliope-textmining.com/index.html
[25] http://www.arcomem.eu/
[26] http://www.trendminer-project.eu/

Ориентирована на обработку текстов разных жанров, в том числе научных статей, новостей, сообщений из блогов, патентов.

1.4.3. Функциональная архитектура систем интеллектуального анализа текстов

Традиционно многие информационные системы представлялись в виде модели чёрного ящика. С этой точки зрения предполагается, что на вход системы ОЕЯ подаётся множество документов, а на выходе мы получаем результат обработки текстов: закономерности, взаимосвязи, тренды и т. д.

Современные прикладные задачи анализа данных, решаемые системами *бизнес-аналитики* (business intelligence, BI), требуют несколько другого подхода к организации систем, который можно назвать *человеко-ориентированным* (human-centered). Его суть заключается в том, что в процессе обработки данных система должна тесно взаимодействовать с пользователем, который оценивает результаты работы системы на каждом шаге и, при необходимости, уточняет запросы. При этом особенно важными становятся параметры интерактивности системы и средства визуализации данных [30, с. 173].

С функциональной точки зрения систему ИАТ можно представить в виде 4 последовательно применяемых групп методов [11, с. 14]:

- Предварительная обработка (preprocessing), заключающаяся в конвертации текстовых данных в каноничную форму, пригодную для интеллектуального анализа.
- Основные алгоритмы интеллектуального анализа (core mining), специфичные для решаемой задачи. К ним относятся поиск образов (pattern discovery, включая методы классификации, кластеризации и ассоциации), анализ трендов (trend analysis), поиск знаний (knowledge discovery).
- Представление данных и результатов, производящаяся специальным графическим пользовательским интерфейсом (graphical user interface, GUI), включая средства визуализации, интерактивного уточнения запросов и настройки параметров алгоритмов анализа.

- Конкретизация (refinement) информации. Целью этого этапа является отсеивание избыточных данных и знаний, которые не представляют интереса с точки зрения решаемой задачи. К этой группе относятся методы гашения (suppression), упорядочения (ordering), очистки (pruning), обобщения (generalization) и кластеризации информации.

В развитых системах ИАТ этапы интеллектуального анализа, визуализации и конкретизации информации применяются итерационно до получения пользователем необходимого результата.

Если система ИАТ ориентирована на определённую предметную область (domain-oriented), её архитектура несколько изменяется и дополняется несколькими компонентами:

- База фоновых знаний (background knowledge base), содержащая информацию об особенностях предметной области, включая понятия, закономерности, иерархии и т. д., которые могут быть полезны при интеллектуальном анализе или конкретизации информации.
- Процедуры синтаксической обработки (parsing) формальных знаний из внешних источников, необходимые для отсеивания нерелевантных знаний с точки зрения решаемой задачи и имеющихся данных.

Общая архитектура предметно-ориентированной системы интеллектуального анализа текстов [11, с. 17] представлена на Рисунок 4.

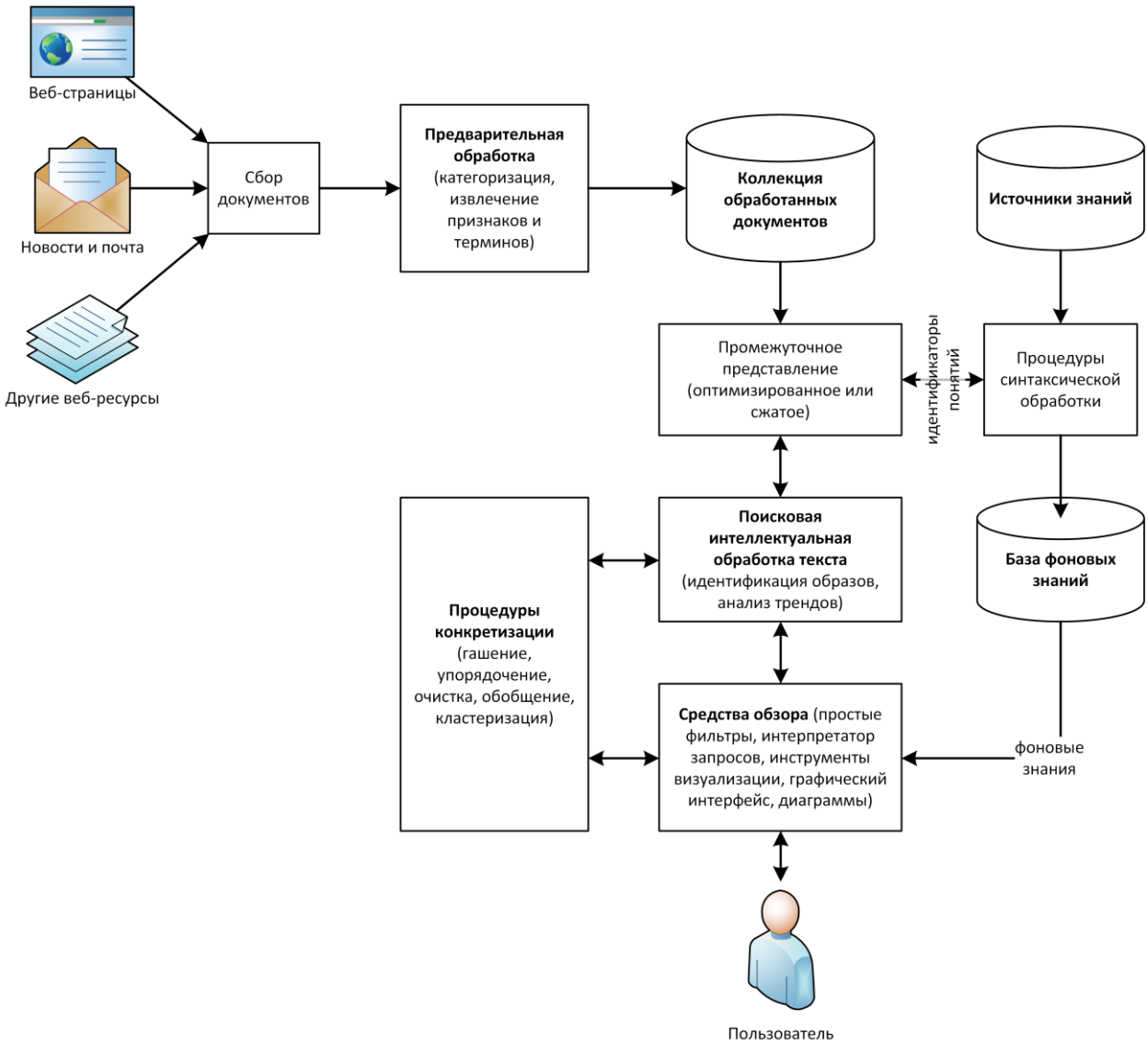


Рисунок 4. Архитектура предметно-ориентированной системы интеллектуального анализа текстов

Этап интеллектуального анализа предметно-ориентированной системы ИАТ может также, при использовании фоновых знаний, включать методы оценки "полезности" и "интересности" найденных образцов, а также алгоритмы оценки собственной эффективности.

1.5. Выводы

Выполненный обзор и анализ позволил сделать следующие выводы:

1. Классические методы ОЕЯ, ориентированные на глубинный анализ текста (морфологический, синтаксический, семантический, прагматический) становятся малоэффективными при обработке больших объёмов текстовых данных из-за больших вычислительных затрат.
2. Методы поверхностного ОЕЯ (интеллектуального анализа данных и машинного обучения, касающиеся обработки текстов), позволяют производить обработку больших объёмов текстов, однако большинство из этих методов ориентированы на работу со статическими коллекциями документов. При обработке потоков данных, и в том числе текстовых сообщений, необходимо учитывать, как минимум, три особенности: а) высокую скорость поступающих данных; б) требование неограниченной памяти; в) изменение понятий с течением времени.
3. Эффективность обработки потоков данных может быть повышена за счёт использования методов анализа и прогнозирования временных рядов, обработки сложных событий и интеллектуального анализа последовательностей.
4. К современным системам бизнес-аналитики предъявляются дополнительные требования: а) возможность учёта знаний предметной области за счёт ведения баз знаний (онтологий); б) наличие удобных, качественных и интерактивных средств визуализации промежуточных и окончательных результатов.

Глава 2. Разработка алгоритма классификации текстовых сообщений и обнаружения трендов в потоках текстовых сообщений

2.1. Математические модели предметной области

2.1.1. Математическая модель потока текстовых сообщений

Для построения математической модели потока текстовых сообщений воспользуемся аппаратом создания онтологий. В этом случае модель потока текстовых сообщений в конкретной предметной области удобно представить в виде направленного ациклического графа с одной корневой вершиной, отражающего отношения "общее—частное" (см. Рисунок 5). Каждая вершина представляет собой *тематику* — последовательность сообщений, относящихся к одному понятию, явлению, событию, предмету, персоналии и т. п.

Корневая вершина (вершина с нулевым заходом) тематического графа соответствует всему множеству поступивших сообщений. Каждый последующий ярус графа соответствует множеству тематик, более узко описывающих таксономию предметной области. Листовые вершины, образующие последний ярус, будут соответствовать тематикам, максимально узко отражающих понятия предметной области с точки зрения пользователя.

Такое представление позволит нам выполнить требования к конечной системе обработки потока сообщений, определённые в п. 5 параграфа 1.5:

- учёт знаний предметной области обеспечивается за счёт семантических связей между тематиками;
- визуализация данных предметной области обеспечивается за счёт непосредственного отображения графовой структуры тематик пользователю.

В настоящей работе простая древовидная структура расширена за счёт подразделения тематик на два вида: абстрактные и конкретные. Между связанными вершинами, соответствующими конкретным тематикам, проводятся дуги в направлении корневой вершины. Подобные семантические связи можно будет использовать в алгоритме обработки потока текстовых сообщений для уточнения результата классификации (см. пп. 2.2, 2.5).

Рисунок 5. Иерархия тематик в потоке текстовых сообщений

В качестве иллюстрации рассмотрим тематический граф потока текстовых сообщений, представленный на Рисунок 5[27]. Корневой вершиной является тематика "Все сообщения". Тематиками, описывающими абстрактные понятия (выделены штриховой границей), являются "Экономика", "Политика", "Праздники" и "Происшествия". Конкретная тематика "Президент" включает сообщения в потоке, в тексте которых фигурирует Президент РФ (как персоналия). Другими тематиками, описывающими конкретные события и явления, являются "Новый год", "Пожары", "Новогодние обращения..." и "Пожары при праздновании...".

Дуги в направлении от корневой вершины к листовым выражают возможные (вероятные) варианты отношения сообщения к тематикам. Например, если сообщение было отнесено к тематике "Политика", оно с большой степенью вероятности может также относиться и к тематике "Президент". В то же время её отношение к "Пожарам" маловероятно.

Дуги между конкретными тематиками в направлении корневой выражают отношения *наследования* (inheritance): каждая конкретная тематика относится также ко всем конкретным тематикам, являющимся её непосредственными родителями. Например, если сообщение было отнесено с тематике "Новогодние обращения Президента...", то оно должно быть отнесено также к тематикам "Президент" и "Новый год".

[27] Иллюстрация составлена на основе потока новостных сообщений с сайта "Российской газеты" (www.rg.ru) за январь 2013 г.

В общем виде модель потока, описанная выше, может быть представлена четвёркой:

$$S_{\mathrm{T}} = (\mathbb{C}, \mathbb{D}, \mathbb{F}, \mathbb{T}), \quad (2)$$

где S — поток текстовых сообщений в текущий момент времени T; $\mathbb{C}$ — множество тематик в потоке; $\mathbb{D}$ — множество сообщений в потоке; $\mathbb{F}$ — множество признаков, характеризующих тематики и сообщения, $\mathbb{T} = \{1, \dots, \mathrm{T}\}$ — множество периодов времени, для которых велись наблюдения за потоком (например, множество дней).

Каждая тематика $c \in \mathbb{C}$ представляется в виде:

$$c = (h, b, Z, Q, G, K, t, Y), \quad (3)$$

где:

$$Z = \{z\}, z = (m, P(m|c), L_m, w),$$
$$L_m = \{l\}, l = \big(f, P(f|c, m)\big),$$
$$Q = \{q\}, q = (d, P(d|c), v),$$
$$G = \{g\}, g = \big(f, P(f|c)\big),$$
$$K = \{k\},$$
$$Y = \{y_t, \dots y_{\mathrm{T}-1}\},$$
$$m \in \mathbb{C}, f \in \mathbb{F}, d \in \mathbb{D}, k \in \mathbb{C},$$
$$h \in \{1,2, \dots\}, b \in \{\mathrm{A}, \mathrm{К}\}, w \in [0,1], v \in [0,1], t \in \{1, \dots, \mathrm{T}\}, y \in \{1,2, \dots\}.$$

В приведённых формулах:

- h — уровень тематики — максимальная длина маршрута от корневой тематики до данной;
- b — тип тематики: А — абстрактная, К — конкретная;
- Z — множество связей z тематики c с дочерними тематиками $m \in \mathbb{C}$, характеризующихся условными априорными вероятностями $P(m|c)$ наблюдения, условными вероятностями появления признаков L_m и степенью новизны $w \in [0,1]$ тематики m по отношению к c;
- Q — множество связей q тематики c с текстовыми сообщениями $d \in \mathbb{D}$, характеризующихся условными вероятностями появления $P(d|c)$ в тематике c и степенью новизны $v \in [0,1]$ по отношению к c;

- G — множество отличительных признаков f тематики c, характеризующихся условными вероятностями появления в сообщениях из c;
- K — множество конкретных тематик $k \in \mathbb{C}$, являющихся родительскими для тематики c (при условии, что c — также конкретная тематика);
- t — момент времени возникновения (или начала отслеживания) тематики;
- Y — временной ряд тематики, отражающий количества сообщений, поступивших за периоды $\{t, t+1, \dots, \mathrm{T}-1\}$.

Каждое сообщение $d \in \mathbb{D}$ представляется в виде:

$$d = (s, t, F_d), \tag{4}$$

где s — текст сообщения (последовательность символов), $t \in \{1,2, \dots, \mathrm{T}\}$ — момент времени создания сообщения; $F_d = \{f\}$ — множество признаков $f \in \mathbb{F}$ сообщения.

Признаки $f \in \mathbb{F}$ являются номинальными признаками тематик и сообщений в потоке [80, p. 4], а именно, представляет собой *лексемы* — исходные формы слов, встречающихся в сообщении.

В графическом виде взаимосвязи между элементами формальной модели (ф. 2) можно выразить в виде следующей схемы (Рисунок 6):

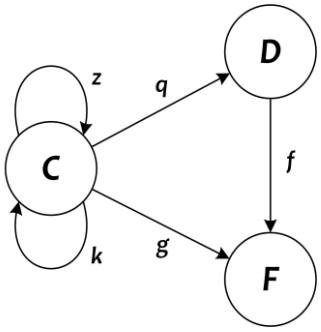


Рисунок 6. Концептуальная схема потока текстовых сообщений

2.1.2. Математическая модель системы обработки потока текстовых сообщений

В общем виде модель системы можно представить в виде (Рисунок 7):

$$Y = \Psi\left(X_{\text{внеш}}, X_{\text{внутр}}\right), \tag{5}$$

$$X_{\text{внеш}} = (s, t),$$

$$X_{\text{внутр}} = (D_t, C_t, P_t),$$

где:

- $X_{\text{внеш}}$ — внешние данные (сообщение в потоке);
- $X_{\text{внутр}}$ — внутренние данные (хранящиеся в базе);
- Ψ — алгоритм обработки поступившего сообщения;
- s — текст поступившего сообщения;
- t — момент времени, предписанный сообщению;
- D_t — множество актуальных на момент времени t сообщений;
- C_t — множество актуальных на момент времени t тематик;
- P_t — множество актуальных на момент времени t пользовательских правил.

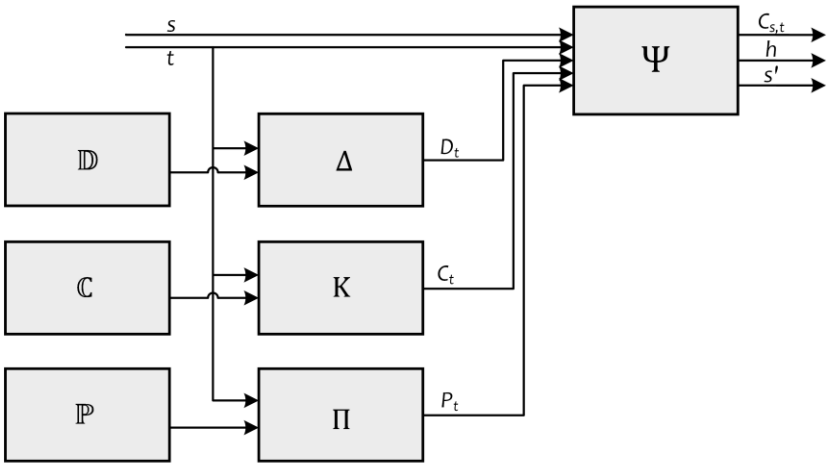


Рисунок 7. Структурно-функциональная модель системы обработки потоков сообщений

Выход Y в модели (ф. 5) представляется тройкой:

$$Y = \left(C_{s,t}, h, s'\right), \tag{6}$$

где:

- $C_{s,t}$ — множество тематик c из числа существующих $\mathbb{C}$, к которым с большой вероятностью относится сообщение s, с оценками вероятностей (b) принадлежности к ним сообщения, новизны (v) сообщения относительно тематик, новизны (w) тематик; $C_{s,t} = \{(c, b, v, w)\}, c \in \mathbb{C}, b \in [0,1], v \in [0,1], w \in [0,1]$;
- h — новая тематика, к которой относится сообщение s, с указанием родительской тематики (z) и оценкой новизны (w); $h = (z, w), z \in \mathbb{C}, w \in [0,1]$;
- s' — обработанный текст сообщения.

Множества D_t, C_t, P_t являются результатами следующих процедур поиска:

$$D_t = \Delta(\mathbb{D}, t), \quad (7)$$
$$C_t = \text{К}(\mathbb{C}, t),$$
$$P_t = \Pi(\mathbb{P}, t),$$

где:

- $\mathbb{D}$ — база всех зарегистрированных сообщений;
- $\mathbb{C}$ — база всех известных тематик;
- $\mathbb{P}$ — база всех имеющихся пользовательских правил.

К алгоритму Ψ предъявляются следующие требования:

- обеспечение точности и полноты классификации (см. п. 1.2.4);
- обработка сообщений в оперативном режиме;
- выявление новых тематик;
- определение новизны сообщения.

К алгоритмам Δ, К, П предъявляется следующее требование:

- точный и быстрый поиск в базе.

Требование оперативности означает способность обрабатывать сообщения по мере их поступления, или в *масштабе реального времени* (см. п. 1.1.3). Требование оперативности намного слабее, чем требование обработки данных в *режиме реального времени* (real-time); в частности, оно допускает игнорирование (отказ от обработки) некоторых сообщений для повышения общей производительности системы. Формально оно может быть выражено следующим образом:

$$\forall \tau \in \mathbb{T} : \Delta\tau < (1 - l) \cdot |C_\tau| \cdot \Delta t_d, \quad (8)$$

где:

- $\Delta\tau$ — длительность периода времени τ;
- l — допустимый процент отказов для сообщений;
- C_τ — множество сообщений, поступивших за период τ;
- Δt_d — средняя длительность обработки одного сообщения.

2.2. Общая структура алгоритма иерархической обработки текстового сообщения в потоке

Для обработки потоков текстовых сообщений необходимо разработать алгоритм обработки сообщения, включающий классификацию сообщений, определение степени новизны сообщения и группы сообщений, идентификацию новых событий и модификацию решающего графа. В основе разрабатываемого алгоритма лежат следующие предложения:

- определение множества тематик, к которым принадлежит сообщение, может быть достигнуто за два этапа: на этапе первичной классификации будет получено избыточное множество тематик F (но гораздо меньшее по мощности, чем множество всех известных тематик C), на этапе точной классификации из множества F остаются только те вершины, к которым принадлежит обрабатываемое сообщение;
- первичная классификация заключается в последовательной классификации сообщения на каждом ярусе тематического графа и отсечением заведомо нерелевантных вершин;
- точная классификация может быть достигнута при использовании семантической информации о предметной области, выражаемой связями между вершинами.

В общем виде предлагаемый алгоритм обработки сообщения можно описать в виде последовательности следующих этапов:

1. Предварительная обработка текста сообщения.
2. Первичная классификация.
3. Точная классификация.
4. Определение новизны тематик.
5. Применение пользовательских правил.

В качестве иллюстрации работы предлагаемого алгоритма рассмотрим иерархию тематик, изображённую на Рисунок 5. Допустим, в систему поступило сообщение, изначально относящееся к тематике "Новогодние обращения Президента...". Тогда последовательность шагов классификации на этапе первичной классификации может выглядеть следующим образом (см. Рисунок 8):

1. Классифицируем сообщение на вершине "Все сообщения". Результат — множество {Экономика, Политика, Праздники}. Тематика "Происшествия" признана нерелевантной.
2. Классифицируем сообщение на вершине "Экономика". Результат — пустое множество. Все дочерние тематики признаны нерелевантными.
3. Классифицируем сообщение на вершине "Политика". Результат — пустое множество. Все дочерние тематики признаны нерелевантными.
4. Классифицируем сообщение на вершине "Праздники". Результат — множество {Новый год}. Остальные дочерние тематики признаны нерелевантными.
5. Классифицируем сообщение на вершине "Новый год". Результат — множество {Новогодние обращения…}. Тематика "Пожары при праздновании..." признана нерелевантной.
6. Так как все вершины, на которых может производиться классификация, пройдены, останавливаем алгоритм. Результатом работы алгоритма первичной классификации является множество {Экономика, Политика, Праздники, Новый год, Новогодние обращения…}.

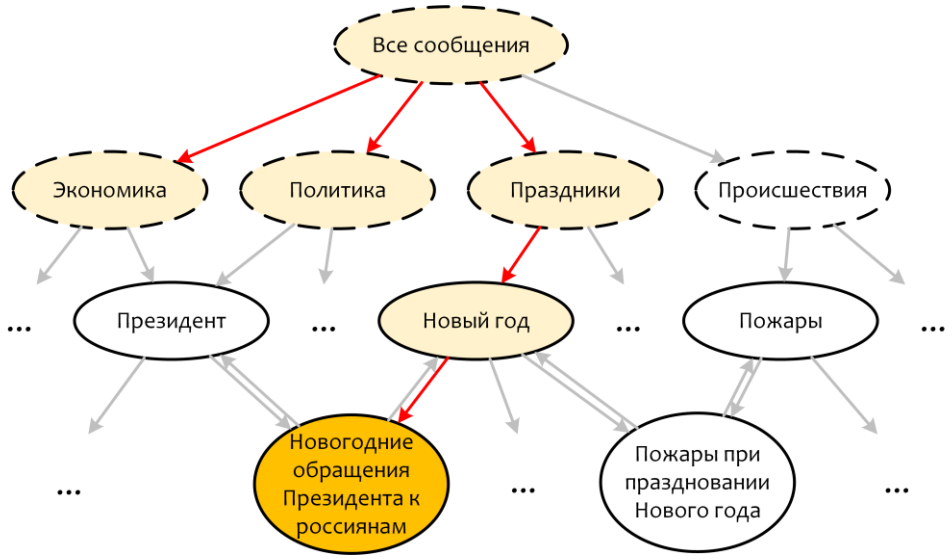


Рисунок 8. Пример первичной классификации сообщения

Последовательность шагов точной классификации может выглядеть следующим образом (Рисунок 9):

1. Наиболее "специализированная" тематика из множества, полученного на этапе первичной классификации, это "Новогодние обращения...". Добавляем её к результату.
2. Лучшая (в данном случае, единственная) оценка для тематики "Новогодние обращения..." была достигнута при классификации в вершине "Новый год". Добавляем её к результату.
3. Лучшая оценка для тематики "Новый год" была достигнута при классификации в вершине "Праздники". Добавляем её к результату.
4. Лучшая оценка для тематики "Праздники" была достигнута при классификации в вершине "Все сообщения". Добавляем её к результату.
5. Так как тематика "Все сообщения" является корневой, останавливаем рекурсию.
6. Так как тематики "Новогодние обращения" является конкретной, добавляем к результату все связанные с ней родительские конкретные тематики. Таковыми являются "Президент" и "Новый год".
7. Результатом работы алгоритма точной классификации является множество {Новогодние обращения..., Новый год, Праздники, Все сообщения, Президент}.

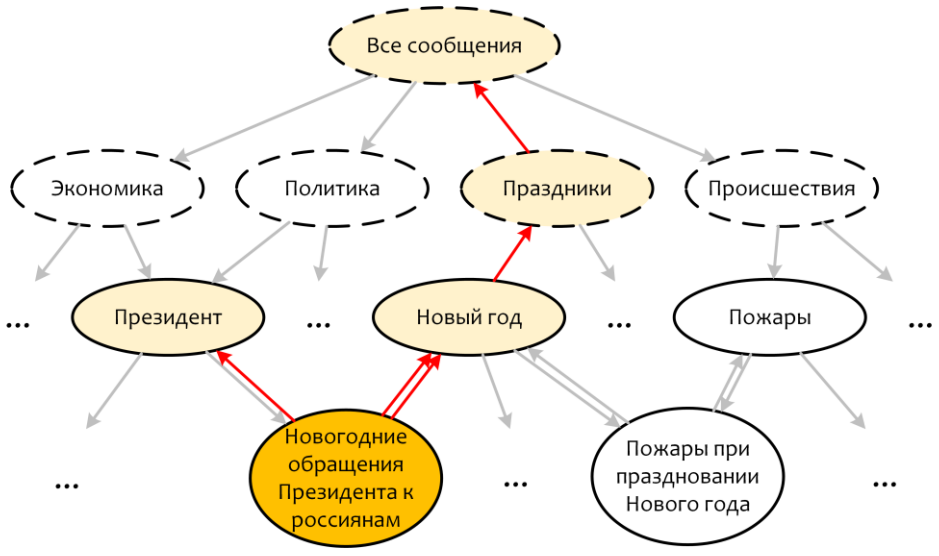


Рисунок 9. Пример точной классификации сообщения

За счёт разделения тематик на два типа (абстрактные и конкретные) удалось разрешить противоречие, возникающее при наследовании тематик: с одной стороны, поступившее сообщение было корректно отнесено к тематике

"Президент", с другой стороны, родительские для тематики "Президент" вершины "Экономика" и "Политика" не были (опять же, корректно) добавлены к результату, так как имели лишь косвенное отношение к сообщению.

2.3. Предварительная обработка текста сообщения

Цель предварительной обработки текста (text preprocessing) заключается в идентификации элементов текста, пригодных для последующей обработки. В контексте ОЕЯ предварительная обработка включает в себя три этапа:

- сортировка документов (document triage);
- сегментирование текста (text segmentation);
- нормализация слов.

Сортировка документов включает следующие подэтапы [81]:

- идентификация кодировки символов;
- идентификация языка;
- секционирование текста (text sectioning).

Последний подэтап особенно важен с условиях работы с мультимедиаконтентом, и, в первую очередь, с веб-страницами, где необходимо отделять полезный текст от гипертекстовых метаданных, ссылок, графической информации, таблиц и рекламы.

Сегментирование текста включает следующие подэтапы:

- разметка, или сегментирование слов (word segmentation), или выделение лексем (tokenization);
- сегментирование предложений (sentence segmentation), или макросинтаксический анализ [25, с. 58].

Разметка слов является одним из базовых этапов обработки текста. Для большинства распространённых искусственных и естественных языков он не представляет существенной проблемы благодаря присутствию чётко обозначенных разделителей — пробелов и знаков препинания.

Специальные методы выделения слов требуются в случаях:

- языков без разделителей (японского, китайского);

- языков с сильно развитой словообразовательной морфологией (немецкого);
- текстов с типографическими ошибками и нарушениями (пропущенными пробелами, лишними дефисами и т. д.).

В данной работе предполагается, что для текстов сообщений используется русский язык без явных ошибок типографики.

Под *нормализацией слов* мы будем понимать приведение различных форм одного слова к единому виду. Существует два основных способа нормализации: стемминг — "приближённый эвристический процесс, в ходе которого от слов отбрасываются окончания" и лемматизация — "точный процесс с использованием лексикона и морфологического анализа слов, в результате которого удаляются только флективные окончания и возвращается основная, или словарная, форма слова, называемая леммой" [37, с. 54].

Стемминг [82], как правило, применяется для языков со слабо развитой морфологией (например, английского). Наиболее распространённым методом стемминга является алгоритм Портера [83], адаптация которого существует и для русского языка. Преимуществами стемминга являются:

- одназначность результата;
- возможность обработки "незнакомых" слов;
- высокая скорость работы.

Лемматизация [84; 85, с. 87], обычно, применяется для тех языков, для которых стемминг усекает слишком большое количество информации о слове. Преимуществами этого вида нормализации являются:

- возможность одновременного проведения морфологического анализа и поиска ошибок;
- разграничение слов, имеющих одинаковые основы.

Большой проблемой при выполнении лемматизации являются омоформы (омонимы в одной из морфологических форм); эвристическое разрешение этой проблемы может осуществляться путём выбора формы с максимальной частотой термина для данной предметной области.

Наиболее простым вариантом лемматизации является получение нормальных форм на основе словаря, содержащего все пары "слово — исходная

форма". Недостатки этого подхода связаны с большим объёмом данных и невозможностью обработки незнакомых слов. Модификацией этого решения, позволяющей повысить скорость поиска и сократить объём требуемой памяти, является использование конечного автомата [86]. Третьим вариантом является использование гибридного варианта стемминга и словарного подхода [87], основанного на хранении пар "окончание слова — исходная форма".

Сравнение двух вариантов нормализации слов (стеммера Портера и лемматизации на основе гибридного подхода) приведено в Таблица 1.

Таблица 1. Сравнение вариантов нормализации слов (И — исходный текст[28], С — результат применения стеммера Портера, Л — результат применения лемматизации)

И	Горе и ужас мадам Беломут не поддаются описанию. Но, увы, и то и другое было непродолжительно. В ту же ночь, вернувшись с Анфисой с дачи, на которую Анна Францевна почему-то спешно поехала, она не застала уже гражданки Беломут в квартире. Но этого мало: двери обеих комнат, которые занимали супруги Беломут, оказались запечатанными!
С	гор и ужас мад беломут не подда описан но ув и то и друг был непродолжительн в ту же ноч вернувш с анфис с дач на котор ан францевн поч то спешн поеха он не заста уж гражданк беломут в квартир но эт мал двер обе комнат котор занима супруг беломут оказа запечата
Л	горе и ужас мадам беломут не поддаваться описание но увы и тот и другой быть непродолжительный в тот же ночь вернуться с анфиса с дача на который анна франц почему тот спешно поехать она не застать уже гражданка беломут в квартира но этот мало дверь оба комната который занимать супруга беломут оказаться запечатать

Основным языком текстов сообщения, с которыми будут работать пользователи системы, является русский язык, для которого характерна сильно развитая морфология слов. С учётом этого в качестве процедуры предварительной обработки для алгоритма обработки текстовых сообщений следует использовать процедуру лемматизации.

2.4. Этап первичной классификации

Как было отмечено в п. 2.2, задачей первичной классификации является получение множества тематик. Для реализации процедуры, описанной в п. 2.2,

[28] Фрагмент текста взят из главы 8 романа М. Булгакова "Мастер и Маргарита".

удобно использовать вероятностные методы классификации, а именно наивный байесовский классификатор (НБК, см. п. 2.4.1). Несмотря на то, что НБК в целом показывает худшие результаты по эффективности (по сравнению с методом опорных векторов — support vector machine, SVM — на основании эксперимента, проведённого в [37, с. 289], — на 10 % для классов с большим числом сообщений и на 22 % — для классов с малым числом примеров), он обладает существенно меньшей временной сложностью по сравнению с другими классификаторами. Его преимуществами также является лёгкость модификации для создания многозначного классификатора (см. п. 2.4.2), а также устойчивость к несбалансированным данным.

Алгоритм, реализующий первичную классификацию, представлен на Рисунок 10. Для выбора новой тематики, по которой необходимо произвести классификацию на новой итерации, используется структура типа очереди с приоритетом.

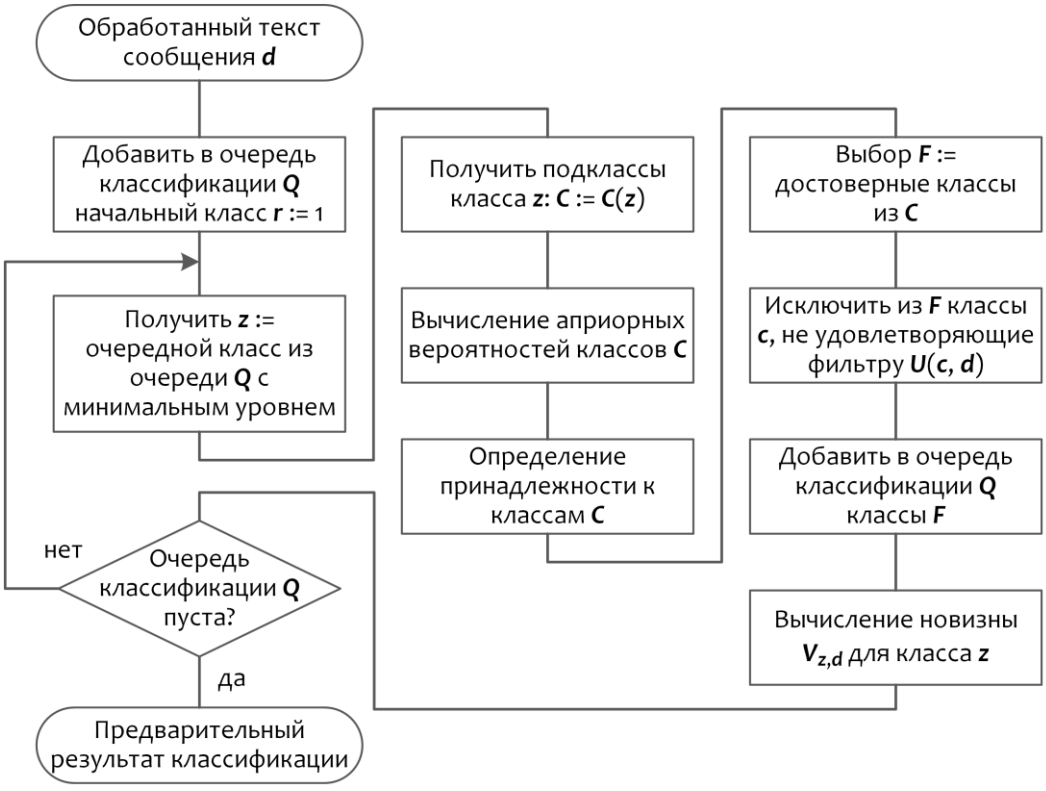


Рисунок 10. Алгоритм первичной классификации сообщения

Для реализации предложенного алгоритма обработки необходимо решить следующие подзадачи:

1. Разработать метод многозначной байесовской классификации сообщения, включая оценку априорных вероятностей классов и выбор признаков.
2. Разработать метод фильтрации тематик $U(c, d)$.
3. Разработать метод определения степени новизны сообщения $V_{z,d}$.

2.4.1. Вероятностная классификация текстов

Как было отмечено в п. 1.2.2, в основе вероятностных методов классификации лежит формула Байеса:

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)} \quad (9)$$

Класс, к которому принадлежит сообщение d, чаще всего на практике определяется как класс, имеющий максимальную апостериорную вероятность (maximum a posteriori probability, MAP) [30, с. 423]:

$$c_{res} = \arg\max_{c \in C} P(c|d) = \arg\max_{c \in C} P(c)P(d|c) \quad (10)$$

(Знаменатель формулы убирается, так как значения $P(d)$ фиксированы для заданного сообщения d, а результат функции максимума не зависит от постоянного коэффициента.)

В основе наиболее простого и вычислительно эффективного подхода к байесовской классификации текстов лежат также два упрощения:

- предположение о взаимной независимости признаков ("наивная" модель);
- предположение о позиционной независимости признаков (модель "мешка слов");

В этом случае решающее правило можно записать в виде:

$$c_{res} = \arg\max_{c \in C} P(c) \prod_{f \in F(d)} P(f|c) \quad (11)$$

Эта модификация называется наивным байесовским классификатором (НБК, naive Bayes classifier) [29, с. 175] и является одной из самых часто используемых при классификации текстов. Обучение НБК состоит в нахождении априорных вероятностей классов $P(c)$ и условных вероятностей признаков $P(f|c)$ для всех классов c.

С целью снижения размерности вычисляемых величин (что важно при использовании ЭВМ) производится логарифмирование функции под максимумом:

$$c_{res} = \arg\max[\log P(c) + \log P(f_1|c) + \log P(f_2|c) + \cdots + \log P(f_n|c)]. \quad (12)$$

В качестве классификационных признаков для НБК выступают числовые характеристики терминов (слов или групп слов) в тексте сообщения. Метод определения числовых характеристик терминов в НБК зависит от модели представления сообщения d. Существует две основных модели представления документов [88]:

- как вектора номинальных значений признаков: $d = (f_1, \dots f_n), f_i \in \mathbb{F}$;
- как вектора бинарных индикаторов, указывающих на наличие признака в документе: $d = ([f_i \in F(d)])$.

В первом случае модель НБК называется *мультиномиальной* [37, p. 267], и оценка условной вероятности появления признака f в классе c равна относительной частоте признака f в сообщениях из класса c:

$$\hat{P}(f|c) = \frac{F_{c,f} + k}{\sum_{f' \in \mathbb{F}} F_{c,f'} + k|\mathbb{F}|} \quad (13)$$

где $k > 0$ — параметр сглаживания, обусловленный возможностью встретить в тексте новый признак[29] [89], $F_{c,f}$ — количество появлений признака f в обучающих сообщениях из класса c, $\mathbb{F}$ — множество всех признаков.

Во втором случае модель НБК называется *многомерной моделью Бернулли* [37, с. 272], и оценка условной вероятности признаков $P(f|c)$ равна доле сообщений из класса c, которые содержат признак f.

2.4.2. Разработка многозначного наивного байесовского классификатора

Классификатор, описываемый формулой (11), является *однозначным* (однотемным, single-label): для каждого объекта d он определяют только один, наиболее подходящий класс c_{res} из множества доступных. Так как текстовое сообщение может относиться к нескольким классам, необходимо разработать

[29] В общем случае подобная оценка вероятности является оценкой Линдстоуна (Lindstone); для k = 1 она называется оценкой Лапласа; для k = 0,5 — оценкой ожидаемого правдоподобия (expected likelihood estimation, ELE).

многозначный (многотемный, multi-label) классификатор, который способен определить множество классов $C_{res} = \{c_1, c_2, \dots, c_n\}$, к которым относится предъявленный объект, причём $n \geq 0$.

Подходы к организации многозначных классификаторов

Существующие методы модификации алгоритмов классификации можно разделить на две большие группы [90]:

- *трансформационные методы* (problem transformation methods), которые сводят задачу многозначной классификации к одной или нескольким задачам однозначной классификации;
- *адаптационные методы* (algorithm adaptation methods), заключающиеся в изменении модели классификатора или метода выбора признаков (см. например, [91]);

Трансформационные методы, в свою очередь, можно разделить на [92]:

- методы *бинарного соответствия* (binary relevance, BR), в основе которых лежит преобразование исходного многозначного классификатора в M бинарных классификаторов (M — число классов);
- методы *попарного соответствия* (pairwise classification, PW), в которых бинарные классификаторы строятся для каждой пары классов (см., например, [38]);
- методы *комбинации тем* (label combination, LC), в которых строится новый классификатор с количеством классов $M' = 2^M - 1$ для всех возможных сочетаний исходных классов;
- *пороговые методы* (ranking and threshold, RT), заключающиеся в оценке выходных значений R_c исходного многоклассового классификатора и выборе некоторого порога π. Тогда результатом классификации будет множество $C_{res} = \{c \in C : R_c \geq \pi\}$;

(Другая классификация подразделяет трансформационные методы по порядку корреляции классифицируемых объектов [93]. При этом методы BR соответствуют методам корреляции первого порядка, PW — второго порядка, LC — высшего порядка.)

Основной недостаток методов BR, PW и LC заключается в вычислительной сложности получаемых моделей (Таблица 2), что затрудняет их применение для задач с большим числом классов.

Таблица 2. Сравнение сложности подходов к организации многозначной классификации (M — количество исходных классов, N — количество документов в обучающем множестве)

Название группы методов	Количество		
	классификаторов	классов в одном классификаторе	документов в одном классификаторе (ср.)
Бинарного соответствия	M	2	N
Попарного соответствия	$\frac{M(M-1)}{2}$	2	$\frac{4N}{M(M-1)}$
Комбинации тем	1	$2^M - 1$	$\frac{N}{2^M - 1}$
Пороговые	1	M	$\frac{N}{M}$

Несмотря на очевидность и простоту, подход на основе пороговой оценки редко рассматривается в специальной литературе из-за ненадёжности статических порогов (т. е. постоянных порогов, задаваемых извне) [92, с. 57].

В настоящей работе предлагается и исследуется ряд методов динамического вычисления пороговых значений.

Разработка многозначного классификатора на основе динамической пороговой оценки

Рассмотрим вектор выходных значений документа R_i. В общем случае, в зависимости от модели классификатора он не является нормированным, т. е. значения его компонентов могут быть больше 1. Для задач бинарной классификации этот вектор имеет длину 2 и рассматривается как вектор значений случайной величины, поэтому нормированные значения будут равны: $R_i^n = R_i/(R_0 + R_1)$, при этом: $\sum R_i^n = R_0^n + R_1^n = 1$.

Способ нормирования результатов многозначного классификатора по отношению к их сумме будем называть *нормированием по общей сумме* (далее — NSum):

$$R_i^n = \frac{R_i}{\sum R_i} \tag{14}$$

Для определения результирующих классов многозначного классификатора вводится порог π, тогда: $C_{res} = \{c \in C : R_c^n \geq \pi\}$.

С увеличением количества классов среднее значение R_i^n становится меньше, поэтому использование статического порога становится затруднительным.

Для решения этой проблемы можно либо использовать другие подходы к нормировке R_i. Мы рассмотрим два метода:

1. *Нормированием по минимальному и максимальному значениям* (далее — NMinMax) назовём способ, основанный на формуле:

$$R_i^n = \frac{R_i}{\min R_i + \max R_i} \tag{15}$$

2. *Нормированием по медиане* (далее — NMean) назовём способ, основанный на формуле:

$$R_i^n = \frac{R_i}{\max R_i + \mathrm{mean}(R_i)}, \tag{16}$$

где $\mathrm{mean}(R_i)$ — медианное значение R_i.

Вместо использования фиксированного значения порога π и нормирования результатов классификатора с помощью рассмотренных способов, можно вычислять значение π во время работы алгоритма.

Для этого выстроим значения R_i по возрастанию, полученный ряд обозначим R^s: $R_0^s < R_1^s < \cdots < R_M^s$ (M — количество классов). Определим индекс компоненты вектора, соответствующей максимуму разности между соседними значениями:

$$U = \arg\max_i (R_{i+1}^s - R_i^s) \tag{17}$$

Тогда динамический порог может быть вычислен как среднее значение:

$$\pi = \frac{R_{U+1}^s + R_U^s}{2} \tag{18}$$

Такой способ вычисления порога назовём *методом максимального подъёма* (далее — DMaxUp).

Оценка эффективности динамического порогового классификатора

Для определения наиболее эффективного подхода к построению многозначного классификатора был проведён вычислительный эксперимент [94]. В эксперименте оценивались предложенные модификации по сравнению с методом бинарного соответствия (BR). Другие трансформационные методы не рассматривались из-за их вычислительной сложности для задач с большим числом классов.

В качестве экспериментальных данных была использована выборка новостных сообщений с сайта "Российской газеты" (www.rg.ru) за период 1 января по 30 апреля 2013 года, собранная и обработанная специально созданным программным комплексом. Каждое сообщение было предписано одной или нескольким тематикам. После первоначальной фильтрации выборка состояла из 2289 сообщений и 33 тематики; 60 % сообщений использовались для обучения классификаторов, 40 % — для тестирования.

В качестве исходного классификатора выступал наивный байесовский классификатор, обучаемый согласно мультиномиальной модели. В классификации участвовали 500 наиболее релевантных признаков, оцененных по методу взаимной информации (см. п. 2.4.3).

Существует как минимум 16 стандартных методов оценки эффективности многозначных классификаторов [95]. Из них для оценки результатов эксперимента были выбраны следующие (Таблица 3).

Таблица 3. Оценки эффективности многозначных классификаторов

Название оценки	**Формула**
Функция потерь Хэмминга (Hamming loss)	$HL(h) = \frac{1}{N}\sum_{i=1}^{N}\frac{\lvert T_i \vartriangle R_i\rvert}{M}$
Точность (precision)	$Pr(h) = \frac{1}{N}\sum_{i=1}^{N}\frac{\lvert T_i \cap R_i\rvert}{\lvert R_i\rvert}$
Полнота (recall)	$Re(h) = \frac{1}{N}\sum_{i=1}^{N}\frac{\lvert T_i \cap R_i\rvert}{\lvert T_i\rvert}$
F-мера (F-measure)	$F(h) = \frac{1}{N}\sum_{i=1}^{N}\frac{2\lvert T_i \cap R_i\rvert}{\lvert T_i\rvert + \lvert R_i\rvert}$

Аккуратность (accuracy)	$A(h) = \frac{1}{N}\sum_{i=1}^{N}\frac{\lvert T_i \cap R_i\rvert}{\lvert T_i \cup R_i\rvert}$
Точное соответствие (exact match)	$Ex(h) = \frac{1}{N}\sum_{i=1}^{N} I(T_i, R_i)$

В приведённых формулах T_i — множество действительных классов, к которым относится документ i; R_i — множество результирующих классов, приписанных документу классификатором; N — число документов в тестовой выборке, M — общее число классов, $I(X, Y)$ — функция, возвращающая 1, если множества X и Y равны, и 0 в противном случае.

Результаты эксперимента представлены в Таблица 4.

Таблица 4. Результаты оценки эффективности многозначной классификации (лучшие результаты выделены)

	HL	Pr	Re	F	A	Ex
BR	0,0429	0,3430	0,4461	0,3593	0,3205	0,2238
NSum	0,0565	0,5130	0,6383	**0,5327**	0,4733	0,3242
NMinMax	0,0603	0,5105	**0,6452**	0,5312	0,4718	0,3253
NMean	0,0603	0,5105	**0,6452**	0,5312	0,4718	0,3253
DMaxUp	**0,0346**	**0,5691**	0,5210	0,5303	**0,5028**	**0,4247**

Результаты показывают, что метод бинарного соответствия (BR) обладает наименьшей эффективностью из исследованных подходов. Использование нормировок на основе общей суммы (NSum), минимального и максимального значений (NMinMax) и медианы при применении статического порога дают примерно одинаковые результаты. Оценка на основе наибольшего подъёма (DMaxUp) в целом даёт лучшие результаты из исследованных алгоритмов, существенный выигрыш по количеству точных соответствий (Ex), а также обладает лучшей сбалансированностью.

2.4.3. Выбор классификационных признаков сообщений

В методах классификации текстов классификационными признаками, как правило, являются термины текста сообщения. Задачей процедуры выбора

признаков является исключение из множества признаков тех терминов, которые не повышают качество классификации. Целями выбора признаков являются [37, с. 279; 96]:

- снижение затрат памяти на хранение множества признаков;
- снижение вычислительных затрат за счёт уменьшения числа множителей в формуле (11);
- повышение качества классификации за счёт удаления признаков, не несущих семантической нагрузки;
- повышение качества классификации за счёт удаления признаков, приводящих к излишней подстройке классификатора под заданные классы (переобучению).

В общем случае задача выбора признаков является NP-полной, то есть требует полного перебора всех сочетаний признаков. Для специализированных алгоритмов процедура выбора признаков может осуществляться на основе методов снижения размерности признакового пространства (например, с помощью метода главных компонент — principal component analysis, PCA [97]) или генетического алгоритма [98; 99]. В случае НБК эти процедуры являются слишком затратными с вычислительной точки зрения.

Эвристическая процедура выбора признаков

Процедура эвристического "жадного" выбора признаков, наиболее часто применяемая на практике для НБК, может быть представлена следующей последовательностью шагов (см. Рисунок 11) [37, с. 280; 100]:

1. Начальное множество признаков F полагается равным множеству F_0 всех терминов из всех сообщений.
2. Удаление из F стоп-слов по заданному словарю (как правило, в него входят служебные слова: предлоги, союзы, частицы и др.), а также редких терминов, которые могут привести к переобучению.
3. Вычисление меры полезности u_t термина t как максимального значения заданной метрики полезности $U(t,c)$ термина t для всех классов c, участвующих в классификации: $u_t = \max\limits_{c \in C} U(t,c)$.

4. Множество признаков F формируется из k терминов с максимальной мерой u_t.

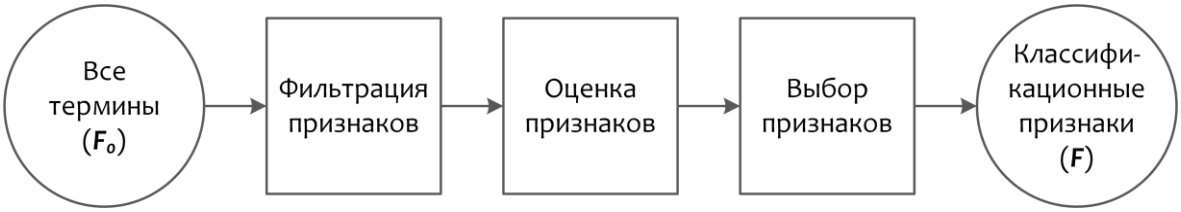


Рисунок 11. Процедура эвристического выбора признаков

Метрика полезности и значение $k = |F|$, приводящие к оптимальному выбору признаков, зависят конкретных задач и определяются на основании эмпирических исследований [101].

Метрики полезности выбора признаков

Из наиболее распространённых метрик полезности для настоящего исследования были выбраны следующие: взаимная информация (MI), знаковая взаимная информация (SMI), параметр хи-квадрат (CHI), коэффициент корреляции (CC), отношение шансов (OR), квадратичное отношение шансов (ORS) и документная частота термина (DF).

Для формального определения этих метрик рассмотрим следующие вероятности:

- $P(t, c)$ — вероятность появления сообщения с термином t и относящегося к классу c;
- $P(t, \bar{c})$ — вероятность появления сообщения с термином t и не относящегося к классу c;
- $P(\bar{t}, c)$ — вероятность появления сообщения без термина t и относящегося к классу c;
- $P(\bar{t}, \bar{c})$ — вероятность появления без термина t и не относящегося к классу c.

Первая и последняя вероятности отражают положительную зависимость между термином и классом, вторая и третья — отрицательную.

Ожидаемая взаимная информация (expected mutual information, MI) [37, с. 280–283], эквивалентная приросту информации (information gain) [101], показывает количество информации о принадлежности сообщения к классу c, получаемое при знании о наличии или отсутствии термина t в сообщении:

$$MI(t,c) = \sum_{c' \in \{c.\bar{c}\}} \sum_{t' \in \{t,\bar{t}\}} P(t',c') \cdot \log \frac{P(t',c')}{P(t')P(c')}. \quad (19)$$

Взаимная информация является *двусторонней* (two-sided) метрикой, так как она не различает положительную и отрицательную зависимость между терминами и классами, и, таким образом, оценивает полезность признака для классификатора в целом.

Знаковая взаимная информация (signed mutual information, SMI) [101], является *односторонним* (one-sided, оценивающими полезность признака для отдельного класса) аналогом MI:

$$SMI(t,c) = \text{sign}[P(t,c)P(\bar{t},\bar{c}) - P(\bar{t},c)P(t,\bar{c})] \cdot MI(t,c). \quad (20)$$

Параметр хи-квадрат (CHI) [37, p. 283–285] показывает отсутствие независимости между термином t и классом c:

$$\chi^2(t,c) = \frac{N[P(t,c)P(\bar{t},\bar{c}) - P(\bar{t},c)P(t,\bar{c})]^2}{P(t)P(\bar{t})P(c)P(\bar{c})}, \quad (21)$$

где N — общее число сообщений в обучающей выборке.

Коэффициент корреляции (CC) является односторонним аналогом CHI:

$$CC(t,c) = \frac{\sqrt{N} \cdot [P(t,c)P(\bar{t},\bar{c}) - P(\bar{t},c)P(t,\bar{c})]}{\sqrt{P(t)P(\bar{t})P(c)P(\bar{c})}}. \quad (22)$$

Отношение шансов (odds ratio, OR) [101] характеризует шансы того, что признак встретится в рассматриваемом классе, по отношению к тому, что признак встретится в других классах:

$$OR(t,c) = \log \frac{P(t|c)[1 - P(t|\bar{c})]}{[1 - P(t|c)]P(t|\bar{c})}. \quad (23)$$

Квадратичное отношение шансов (OR-square, ORS) [101] представляет собой двусторонний аналог OR:

$$ORS(t,c) = OR^2(t,c). \quad (24)$$

Документная частота терминов (document frequency, DF) [37, с. 285] соответствует вероятности встретить сообщение, содержащее термин t в классе c:

$$DF(t, c) = P(t, c). \quad (25)$$

Документная частота является односторонней метрикой.

Оценка эвристической процедуры выбора признаков

Для оценки наилучшего метода выбора признаков для разработанного многозначного классификатора был проведён вычислительный эксперимент. В качестве экспериментальных данных были использованы новостные сообщения с сайта "Российской газеты" (www.rg.ru) за период с 1 января по 30 апреля 2013 года; данные были разделены на 5 наборов с различными характеристиками (Таблица 5).

Таблица 5. Экспериментальные наборы данных

Название набора данных	**Классы**	**Число сообщений**		
		на класс	**всего**	**в обучающей выборке, %**
A33	Внутренняя политика, Законотворчество, Жилье, Московская область, Председатель Правительства, Макроэкономика, Кипр, Северная Корея, Аварии и катастрофы, Венесуэла, Экономика, Президент, Волгоград, Уфа, Космос, Интернет, Образование, Налоги, США, ТВ, Криминальная хроника, Средняя Волга, Госуправление, Общество, Религия, Праздники, Социология, Индия, Блоги и соцсети, Мировое кино, Литература, Госдума, Биатлон (всего: 33)	14 – 171	2289	70
B9	Происшествия, Светская хроника, Внешняя политика, Здоровье, Интернет, Космос, Госуправление, ТВ, Крупные компании (всего: 9)	98 – 422	1528	70
C16	Макроэкономика, Кипр, Наука, Фигурное катание, ЖКХ, Налоги, Товары и цены, Технологии, Архитектура, Легкая атлетика, Авиатранспорт, Литература, Мировое кино, Демография, Жилье, Челябинск (всего: 16)	40–68	820	60

D11	Нефть и газ, Европейский союз, Автомобили, ДТП, Европейские чемпионаты, Армия, Зимний спорт, Социальная защита, Погода, Facebook, Apple (всего: 11)	25–39	329	60
E14	Баскетбол, Гимнастика, Семья, Занятость, Охрана труда, Миграция, Транспорт, Трафик, Ульяновск, Туризм, Финансы, Калининград, Дания, Ярославль (всего: 14)	6–19	178	50

Фильтрация (шаг 2 из вышеприведённой схемы) не производилась. Результаты для бернуллиевского НБК и набора данных A33 приведены на рис. Рисунок 12.

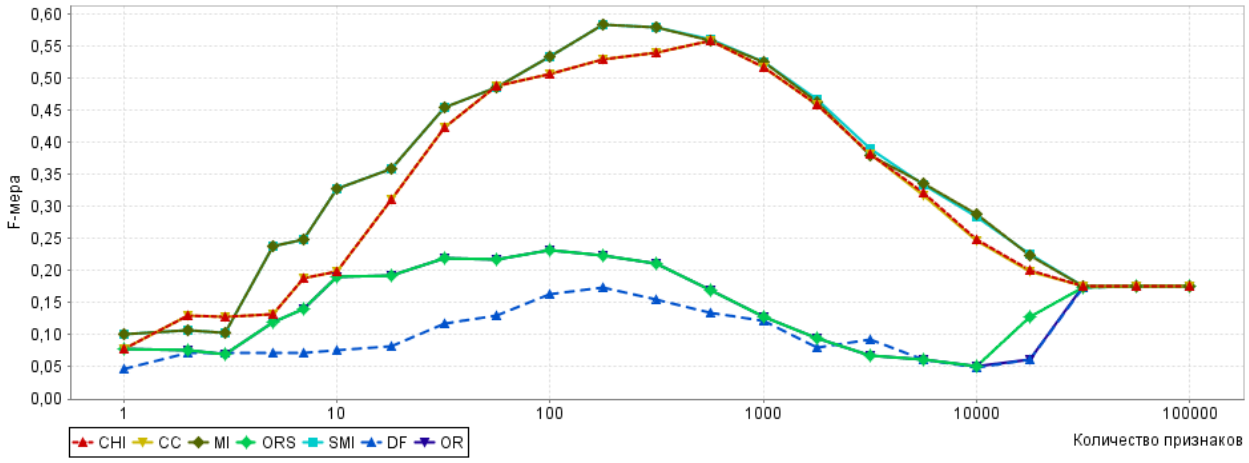


Рисунок 12. Эффективность метрик выбора признаков для эвристической процедуры, бернуллиевской модели НБК и набора данных A33

Результаты, в частности, показывают, что для набора данных A33 для $|F| < 1000$ оптимальными метриками выбора признаков являются MI и SMI при $|F| \approx 178$. Результаты для остальных наборов приведены в приложении А.

Разработка сбалансированной эвристической процедуры выбора признаков

Анализ списков терминов также показал, что для классификаторов с большим числом классов характерна несбалансированность множества признаков, получаемого с помощью стандартной процедуры выбора, заключающаяся в существенно различающихся долях наилучших признаков для отдельных классов.

Для оценки возможности устранения несбалансированности предлагается модификация стандартной процедуры, заключающаяся в том, что на шаге

3 формируются множества $U_c = \{u_{t,c}\}$ наилучших признаков для всех отдельных классов c, а на шаге 4 итоговое множество признаков формируется объединением l наилучших признаков из каждого класса c. (Заметим, что при этом $|F| \leq lM$, где M — количество классов).

Оценка сбалансированной процедуры выбора признаков

Предложенная схема была испытана на тех же наборах данных. Аналогичные результаты для набора A33 показаны на Рисунок 13.

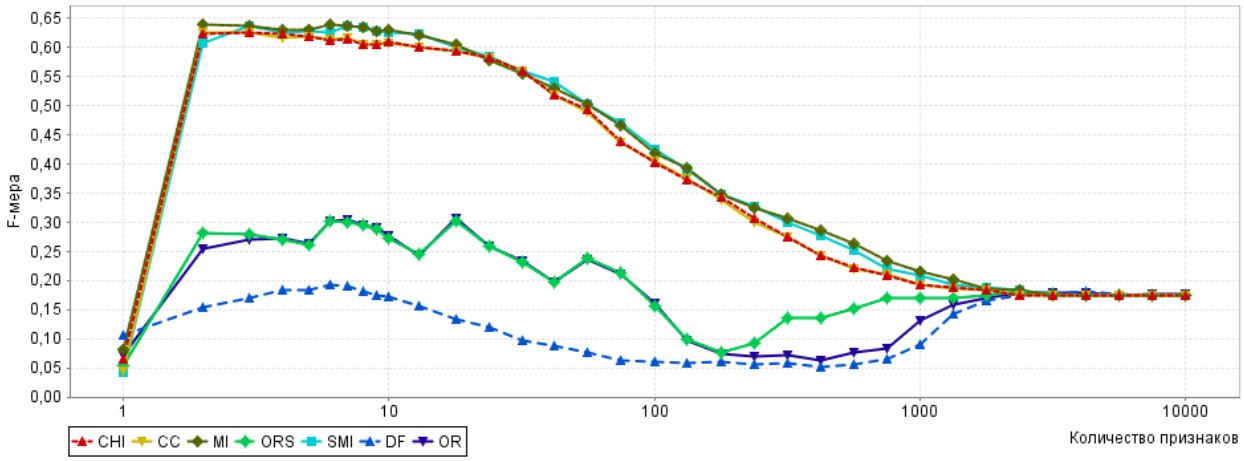


Рисунок 13. Эффективность метрик выбора признаков для модифицированной процедуры, бернуллиевской модели НБК и набора данных A33

Результаты показывают, что количество терминов удалось сократить до 2 для каждого класса ($|F| \approx 66$). Лучшей метрикой стала MI, эффективность классификации возросла с 0,5841 до 0,6406. Результаты для остальных наборов приведены в приложении Б.

Подобное исследование процедур выбора признаков не всегда удобно производить в процессе работы системы обработки потоков сообщений. Исследование результатов позволяет разработать следующие эвристические рекомендации, которые могут быть применены на практике:

1. Использование сбалансированной процедуры позволяет сократить количество признаков и улучшить качество классификации.
2. Для классификаторов со средним числом сообщений на класс, большим 100, следует использовать бернуллиевский НБК; меньшим 100 — мультиномиальный НБК.
3. Общее количество признаков должно быть порядка 100.

2.4.4. Оценка априорных вероятностей тематик

Для вероятностных классификаторов, помимо условных вероятностей признаков, необходимо определить также априорные вероятности классов, описываемые значениями $P(c)$ в формуле (2). Для статических коллекций текстов, при условии репрезентативности обучающей выборки, априорные вероятности можно оценить отношениями количеств документов, принадлежащих определённым классам, к общему числу документов в обучающей выборке. Для потоков текстовых сообщений необходима такая оценка, которая будет учитывать, что априорная вероятность тематики изменяется с течением времени.

В диссертационной работе для оценки априорных вероятностей тематик предлагается использовать методы краткосрочного прогнозирования интервальных временных рядов, соответствующих этим тематикам.

Временным рядом тематики c назовём последовательность $Y_c = \{y_{c,\tau}\}$, $\tau = 1, \ldots, \mathrm{T}-1$, где $y_{c,\tau}$ — количество сообщений, поступивших за период τ с момента начала наблюдений и принадлежащих тематике c. Введём функцию прогнозирования ряда на текущий период времени T:

$$\hat{y}_{c,\mathrm{T}} = S(Y_c) \tag{26}$$

Тогда в качестве априорной вероятности класса c на период T возьмём отношение прогнозируемого уровня временного ряда c за период T к сумме прогнозируемых уровней рядов всех классов, по которым производится классификация:

$$\hat{P}(c, \mathrm{T}) = \frac{\hat{y}_{c,\mathrm{T}}}{\sum_{c' \in C} \hat{y}_{c',\mathrm{T}}} \tag{27}$$

Для исследования характера ряда (а именно, его периодичности), как правило, применяется *автокорреляционный анализ* — исследование корреляций между величиной и её *сдвигом* (запаздыванием, lag) на некоторое количество периодов времени. Для вычисления коэффициента автокорреляции r_k со сдвигом k применяется формула [60, с. 87–88]:

$$r_k = \frac{\sum_{t=k+1}^{n}(y_t - \bar{Y})(y_{t-k} - \bar{Y})}{\sum_{t=1}^{n}(y_t - \bar{Y})^2}, \quad (28)$$

где $\bar{Y}$ — среднее значение ряда, y_x — уровень ряда в момент времени x. Расчёт коэффициентов автокорреляции часто производится не для исходного ряда, а для сглаженных и прологарифмированных его значений.

Коррелограммой, или *автокорреляционной функцией* (АКФ), называется график коэффициентов автокорреляции для различных сдвигов для данного временного ряда.

Анализ экспериментальных данных (новостных сообщений) показал, что характер временных рядов для различных тематик могут существенно различаться (Рисунок 14). С учётом этих различий функция прогнозирования должна различать ряды, имеющие периодическую составляющую, и ряды, не имеющие таковой.

В данной работе при исследовании экспериментальных данных не удалось построить надёжную автоматическую модель определения периодичности рядов. Вследствие этого тип рядов тематик и периоды прогнозирования предлагается задавать в качестве параметров алгоритма обработки сообщений. Соответственно, в системе обработки потоков сообщений должны быть предусмотрены средства визуального анализа временных рядов.

Прогнозирование периодических рядов

Как отмечено в [60, с. 106], обладающих периодичностью (см. например, Рисунок 14, а), можно использовать "классическое разложение, метод Census X-12, экспоненциальное сглаживание Винтера, многомерную регрессию временного ряда и методы Бокса — Дженкинса". Для более полного контроля над параметрами прогнозирования в алгоритме обработки потоков сообщений предлагается использовать метод прогнозирования на основе классической мультипликативной декомпозиции временного ряда (см., например, [102]).

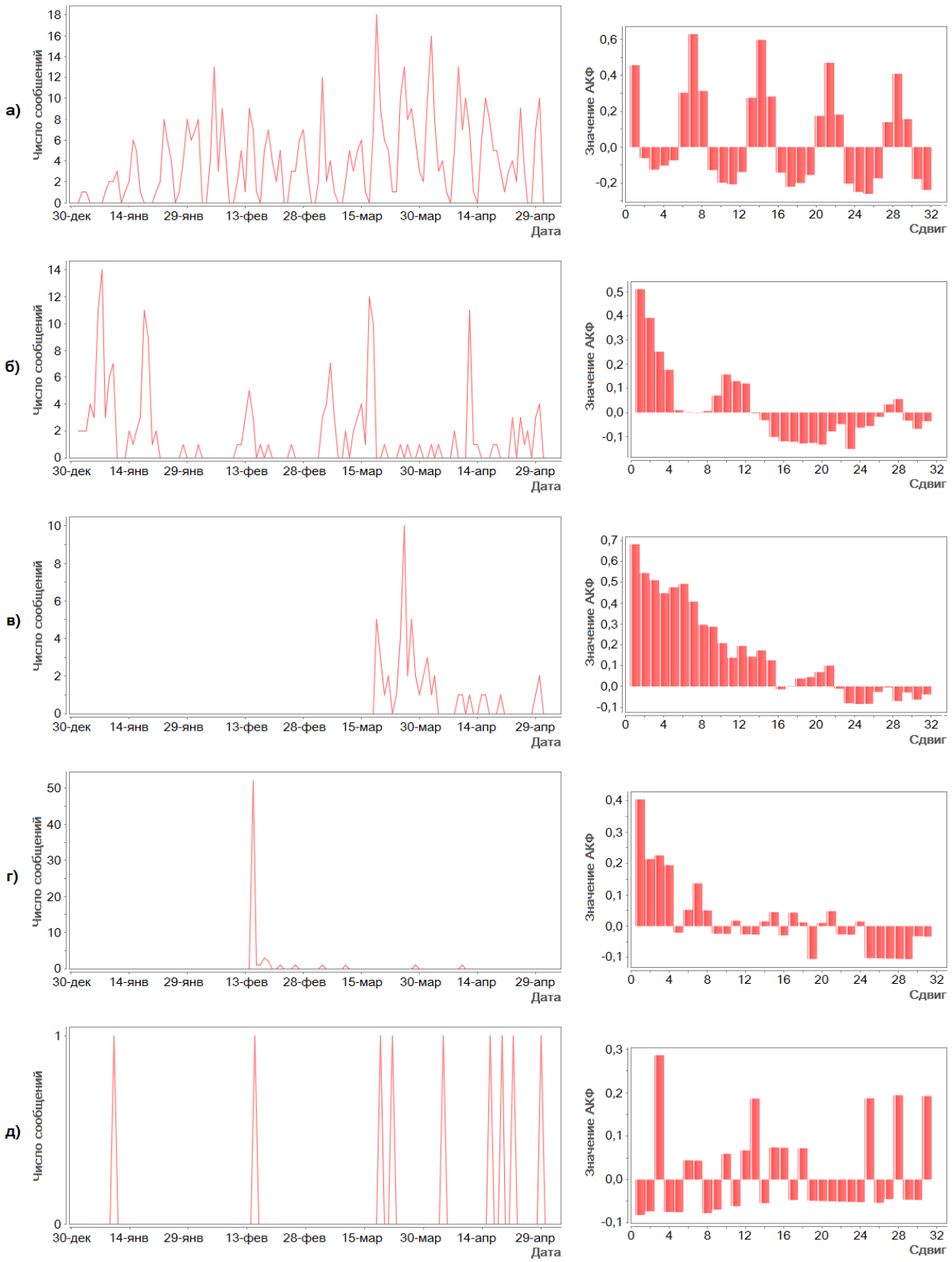


Рисунок 14. Временные ряды различных тематик и их автокорреляционные функции: а) "Экономика"; б) "Праздники"; в) "Кризис на Кипре"; г) "Челябинский метеорит"; д) "Лесные пожары"

Для построения модели прогнозирования по выбранному методу будем использовать следующую последовательность шагов:

1. Вычислить среднее $\bar{Y}$ по значениям ряда, отличным от нуля.
2. Произвести сглаживание ряда (удаление выбросов): если $y_t > 2\bar{Y}$, то принимаем $y_t = (y_{t-1} + y_{t+1})/2$.
3. Рассчитываются сезонные индексы ряда S_i ($0 \le i < k$, k — период прогнозирования) по методу отнесения к скользящему среднему [60, с. 202–209].
4. Вычисляются данные с устранёнными сезонными колебаниями, которые включают трендовую и нерегулярную компоненты ряда ($T_t \cdot I_t$).
5. На основании линейного сглаживания вычисляются параметры уравнения тренда $T_t = a + bt$.

   Для прогнозирования по данной модели используем формулу:

$$S(Y) = (a + b\mathrm{T}) \cdot S_{(\mathrm{T}+1) \bmod k}, \qquad (29)$$

где T — текущий период времени.

Прогнозирование непериодических рядов

Для остальных рядов (см. рРисунок 14, б–д) в качестве прогнозируемого уровня предлагается использовать скользящее среднее значение по ненулевым значениям за $2k$ прошлых периода, где k — период ряда ближайшей родительской тематики, имеющего периодический характер.

2.4.5. Фильтрация нерелевантных тематик

Результатами работы байесовского классификатора, разработанного в п. 2.4.2, являются относительные оценки вероятностей классов. При этом на каждой итерации этапа первичной классификации результатом будет множество тематик, содержащее по крайней мере один элемент. Иными словами, разработанный многозначный НБК сам по себе не способен распознать ситуацию, когда поступившее сообщение не относится ни к одной из рассматриваемых тематик.

Для устранения этой проблемы в алгоритме обработки потока сообщений на этапе первичной классификации (см. Рисунок 10) вводится функция $U(c, d)$, которая для заданной пары (тематика, обрабатываемое сообщение) определяет, относится ли сообщение к рассматриваемой тематике. Выходом функции является булево значение {И, Л}.

В зависимости от реализации этой функции система обработки потоков текстовых сообщений будет по-разному реагировать на поступающие сообщения. В частности, если $U(c,d) \equiv$ И (все тематики на выходе классификатора считаются релевантными по отношению к поступившему сообщению), то алгоритм будет определять принадлежность сообщения только к имеющимся тематикам.

В настоящей работе в основу реализации функции $U(c,d)$ положены два принципа.

- Поведение системы должно соответствовать специфике предметной области. Следовательно, форма реализации функции $U(c,d)$ должна определяться пользователем. Это может быть достигнуто при применении правил предметной области, выраженных с помощью продукций (см. п. 2.7).
- Даже при применении правил, относительных оценок вероятностей на выходе многозначного порогового НБК будет недостаточно для определения релевантности тематик. Для вычисления последней следует использовать абсолютные оценки, основанные на векторных мерах сходства текстов.

Меры сходства текстовых документов

В информационном поиске используются следующие основные способы определения сходства текстов [103] (Таблица 6).

Таблица 6. Меры сходства текстов

Название	**Формула**
Простой коэффициент соответствия (simple matching coefficient, SMC)	$M(X,Y) = \lvert X \cap Y \rvert$
Коэффициент Дайса (Dice coefficient)	$D(X,Y) = \frac{\lvert X \cap Y \rvert}{\lvert X \rvert + \lvert Y \rvert}$
Коэффициент Джаккарда (Jaccard coefficient)	$J(X,Y) = \frac{\lvert X \cap Y \rvert}{\lvert X \cup Y \rvert}$
Косинусный коэффициент (cosine similarity)	$S(X,Y) = \frac{\lvert X \cap Y \rvert}{\sqrt{\lvert X \rvert} \cdot \sqrt{\lvert Y \rvert}}$
Коэффициент перекрытия (overlap coefficient)	$O(X,Y) = \frac{\lvert X \cap Y \rvert}{\min(\lvert X \rvert, \lvert Y \rvert)}$

в приведённых выше формулах X и Y — множество признаков первого и второго текстов, соответственно.

Из перечисленных методов для задач классификации чаще всего применяется косинусный коэффициент, называемый также косинусной мерой сходства. Если сообщение определяется бинарным вектором, компоненты которого являются индикаторами присутствия определённого признака в тексте сообщения, то косинусную меру можно вычислить следующим образом:

$$S(d_1, d_2, F_0) = \frac{\sum_{f \in F_0}[f \in F(d_1)][f \in F(d_2)]}{\sqrt{\sum_{f \in F_0}[f \in F(d_1)]} \cdot \sqrt{\sum_{f \in F_0}[f \in F(d_2)]}}, \quad (30)$$

где $F(x)$ — функция, возвращающая множество отличительных признаков документа x, $[x]$ — индикатор булевого значения (0, если x ложно, и 1 — если истинно).

2.4.6. Новизна сообщения

Под *новизной сообщения* в настоящей диссертационной работе понимается набор количественных и качественных параметров, не позволяющих отнести сообщение ни к одной известной тематике. Для количественной оценки новизны введём метрику *степени новизны сообщения* d по отношению к тематике z, которая формально определяется как:

$$v_{d,z} = 1 - E(d, c, z); \quad (31)$$

где c — наиболее вероятный класс из потомков z; $E(d, c, z)$ – максимальная косинусная мера сходства (ф. 30):

$$E(d, c, z) = \max_{g \in D(c)} S\big(d, g, F(z)\big); \quad (32)$$

$D(c)$ — функция, возвращающая множество документов класса c; $F(c)$ — функция, возвращающая множество отличительных признаков класса c. Под отличительными признаками $F(c)$ будем понимать подмножество классификационных признаков таких, которые являются общими для отдельной тематики c.

2.5. Этап точной классификации

Процедура точной классификации, описанная в п. 2.2, может быть реализована с помощью следующего алгоритма (Рисунок 15).

На первых двух подэтапах точной классификации выбирается наиболее "узкая" тематика, к которой принадлежит рассматриваемое сообщение.

Как упоминалось ранее, результатами работы байесовских классификаторов, использующихся в итерациях на этапе предварительной классификации, являются относительные оценки вероятностей классов. Так как для разных родительских тематик применяются разные классификаторы, то их результаты не подлежат непосредственному сравнению друг с другом. Соответственно, для определения того, какая из листовых тематик является наилучшей, оценок вероятностей классов будет недостаточно.

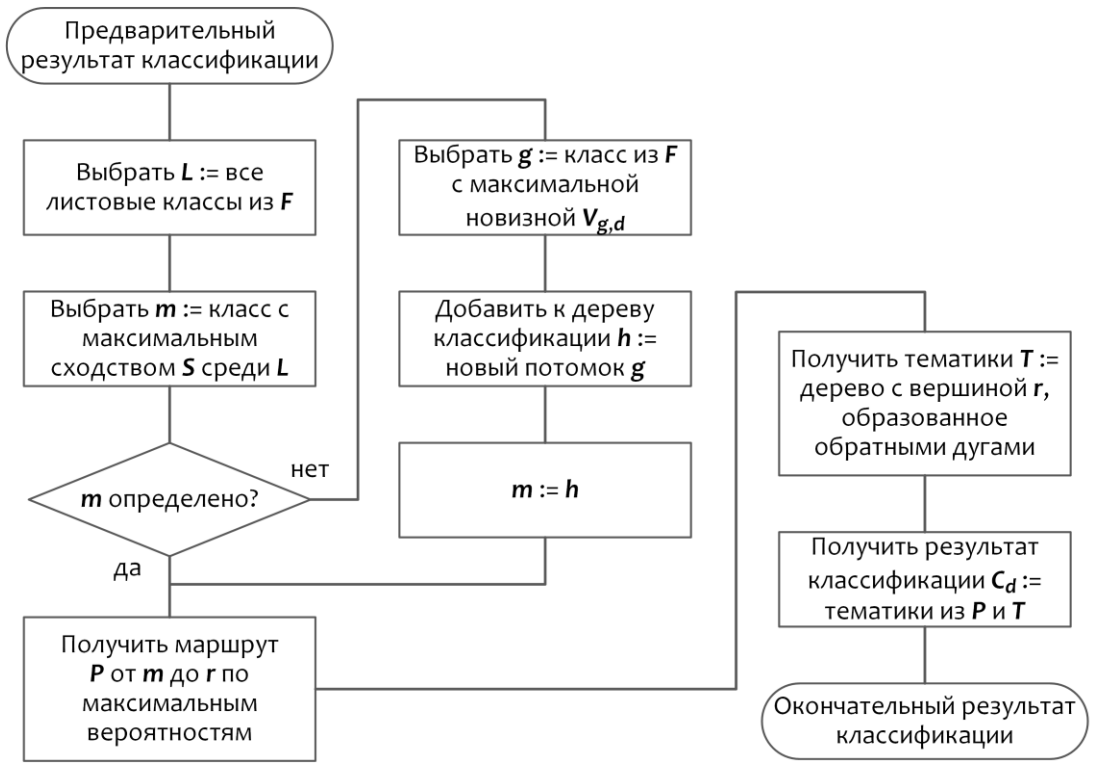


Рисунок 15. Алгоритм уточнения результатов классификации

Вместо этого в алгоритме уточнения результатов используется подход на основе векторного представления текстов сообщений. В качестве наиболее подходящей тематики m выбирается та тематика, для которой новизна сообщения (см. п. 2.4.6) минимальна.

При использовании фильтра тематик $U(c,d)$ (п. 2.4.5) множество релевантных листовых тематик для рассматриваемого сообщения может быть пустым. В этом случае предполагается, что алгоритм обработки потока должен создать для рассматриваемого сообщения новую тематику.

2.6. Этап определения новизны тематик

Следующей подзадачей, которую необходимо выполнить в рамках разработки алгоритма обработки потока текстовых сообщений, является определение новизны тематик W_c.

Под *новизной тематики* в настоящей работе понимается набор количественных и качественных параметров, не позволяющих отнести тематику в качестве дочерней ни к одной из известных. Для количественной оценки новизны введена метрика *степени новизны тематики* c по отношению к тематике z, которая формально определяется как:

$$w(c,z) = \frac{|D(c)|}{\frac{|D(z)|}{|C(z)|} + |D(c)|}. \quad (33)$$

Абсолютная степень новизны W_c при этом может быть вычислена как степень новизны тематики c по отношению к её родительской тематике:

$$W_c = w\big(c, \text{parent}(c)\big). \quad (34)$$

В качестве иллюстрации практического смысла введённой метрики рассмотрим фрагмент тематического графа потока, изображённый на Рисунок 16.

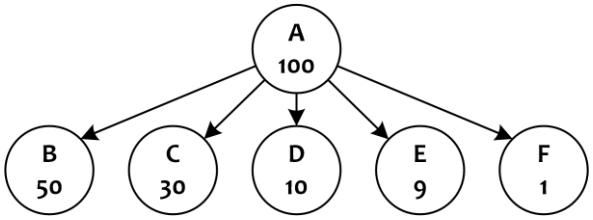


Рисунок 16. Пример определения новизны тематик (числа означают количества сообщений в тематиках, поступивших за последний период)

Степень новизны тематики B согласно формуле () будет равна:

$$W_B = w(B,A) = \frac{|D(B)|}{\frac{|D(A)|}{|C(A)|} + |D(B)|} = \frac{50}{\frac{100}{5} + 50} = \frac{5}{7} \approx 0{,}714$$

Аналогичным образом для тематик C, D, E, F будут получены следующие значения: $W_C = 0{,}5$, $W_D \approx 0{,}333$, $W_E \approx 0{,}31$, $W_F \approx 0{,}048$.

2.7. Применение пользовательских правил

В зависимости от предметной области от системы обработки потока текстовых сообщений может потребоваться выполнение специальных операций над потоком или текстом сообщения, например:

- расширение тематического графа (добавление новых тематик или связей между существующими тематиками);
- ограничение количества тематик, к которым может принадлежать сообщение на определённом ярусе;
- обнуление результатов автоматической классификации;
- присвоение тематик на основе ключевых слов, встречающихся в тексте сообщения;
- удаление сообщения из системы и др.

Эта функциональность может быть реализована на основе методов обработки сложных событий (см. п. 1.3.2) с помощью пользовательских правил, выраженных продукциями. Параметрами антецедентов в правилах, определяемых пользователем, могут являться:

- результат классификации C_{res} (множество тематик на выходе классификатора);
- темпоральные отношения между тематиками и сообщениями;
- оценка новизны сообщения по отношению к тематикам из C_{res};
- оценка новизны тематик из C_{res}.

2.8. Разработка метода оценки алгоритмов оперативной классификации потоков текстовых сообщений

Разработать метод оценки эффективности алгоритмов итерационной классификации потоков.

При разработке алгоритмов классификации потоков текстовых сообщений возникает вопрос оценки их эффективности. Для потоковых данных было бы не совсем корректно применять классическую схему разбиения имеющейся экспериментальных данных на обучающую и контрольную выборки случайным образом.

Более удачным подходом было бы "сымитировать" действия пользователя системы обработки потоков, который проверяет и корректирует действия системы через определённые периоды времени. Для реализации такого поведения удобно адаптировать схему перекрёстного контроля, рассмотренную в п. 1.2.4 на Рисунок 2.

Упорядочим имеющиеся данные по времени и разобьём их на n частей. Далее проведём $n-1$ итерацию: на k-ой итерации ($2 \leq k \leq n$) в качестве обучающей выборки возьмём часть $k-1$, в качестве контрольной — часть k. Схема данного процесса приведена на Рисунок 17, а алгоритм, реализующий данный подход — на Рисунок 18.

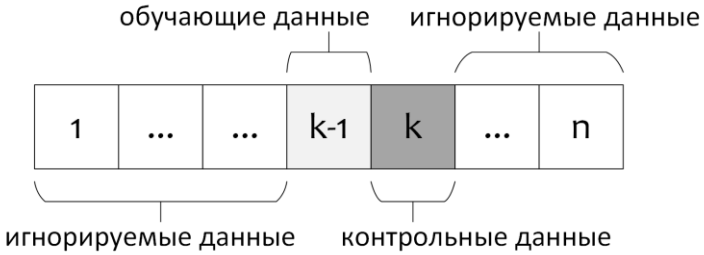


Рисунок 17. Схема перекрёстного контроля для методов обработки потоков текстовых сообщений

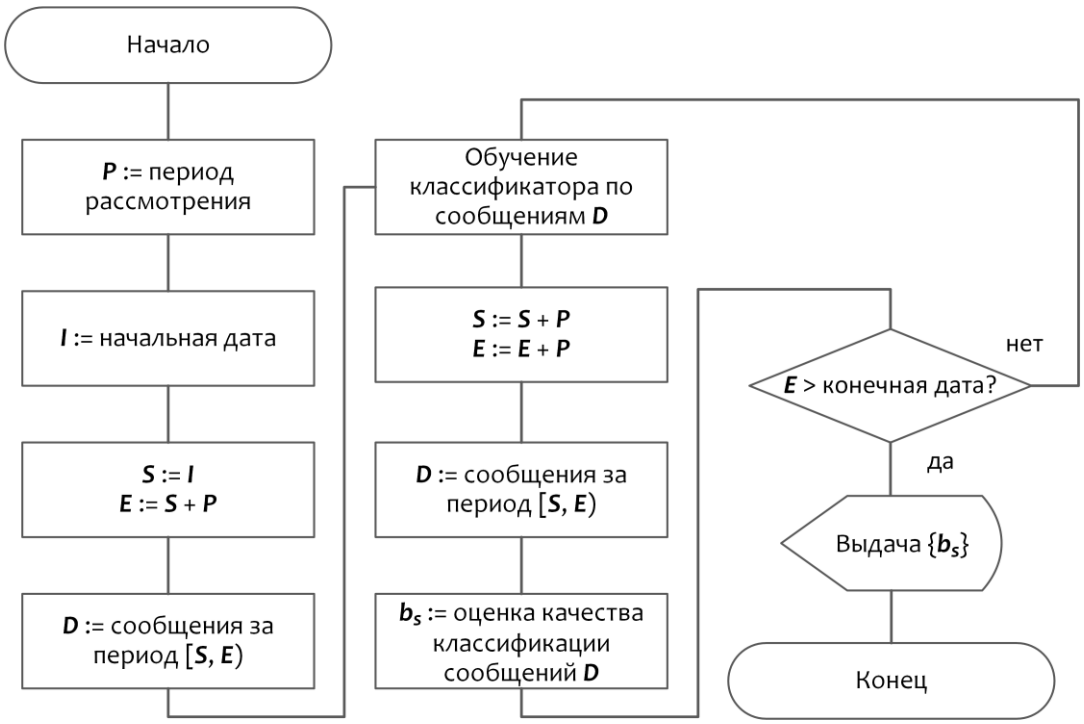


Рисунок 18. Алгоритм перекрёстного контроля для методов обработки потоков текстовых сообщений

2.9. Выводы

1. На основе направленного ациклического графа была разработана модель потока текстовых сообщений, учитывающая семантические связи между тематиками. Предложено разделение тематик на конкретные и абстрактные. За счёт включения в граф дуг между конкретными тематиками удалось разрешить противоречие, возникающие при наследовании тематик. Разработанная модель позволяет в формальном виде отобразить и исследовать закономерности между тематиками.
2. Разработана формальная модель системы обработки потока текстовых сообщений, позволившая определить требования к алгоритму обработки потока сообщений.
3. Разработан алгоритм обработки потока текстовых сообщений с частичным обучением.
4. Разработан пороговый многозначный наивный байесовский классификатор на основе динамического порога.
5. С целью повышения эффективности работы предложенного многозначного классификатора модифицирована процедура выбора признаков текста. Разработаны рекомендации по выбору признаков и модели обучения для тематик различной мощности.
6. Предложена оценка априорных вероятностей на основе методов прогнозирования временного ряда тематики. Для периодических рядов применяется классическая процедура декомпозиции временного ряда, для непериодических — метод на основе скользящего среднего.
7. Предложены подходы к оценке степеней новизны. Оценка новизны сообщения производится на основе косинусной меры сходства сообщения и наиболее вероятной тематикой, оценка новизны тематики — на основе соотношения количества сообщений в смежных тематиках.
8. Определена процедура применения пользовательских продукционных правил, позволяющих адаптировать алгоритм обработки потока к новым предметным областям.
9. На основе перекрёстного контроля разработан метод оценки эффективности итерационных алгоритмов классификации потоков сообщений.

Глава 3. Создание программного комплекса обработки потоков текстовых сообщений

3.1. Определение требований к системе обработки потоков текстовых сообщений

3.1.1. Определение требований к корпоративным системам принятия решений

Основным назначением системы обработки потоков текстовых сообщений являются их использование в качестве средств аналитики и поддержки принятия решения на предприятиях. В настоящее время всё больше предприятий заинтересованы в переходе от использования разнородного программного обеспечения, действующего в рамках отдельных подразделений, и создании единой корпоративной информационной системы (КИС), которой мог бы пользоваться каждый сотрудник предприятия [104]. Этот процесс называется интеграцией корпоративных приложений (интеграции приложений предприятия [105], enterprise application integration, EAI).

В рамках концепции единой КИС система обработки потоков текстовых сообщений должна представлять собой модуль, входом которой будут служить сообщения, а выходом — отчёты о проведённой обработке последовательностей сообщений за определённый период времени.

Требования к современным КИС включают:

- модульность — организация архитектуры в виде отдельных компонентов, связываемых промежуточным программным обеспечением (middleware);
- слабое связывание [106, с. 50] — уменьшение числа допущений, которые делают компоненты друг о друге;
- гибкость — возможность замены компонента с минимальными затратами;
- использование открытых стандартов — использование стандартных протоколов и интерфейсов для связи компонентов друг с другом;

- переносимость — независимость от аппаратно-программной платформы;
- повторное использование компонентов — возможность использования во вновь создаваемой системе некоторого компонента, разработанного ранее в рамках другой системы.

3.1.2. Определение требований к процессу разработки

При проектировании современных систем, предназначенных для корпоративной среды, использование строгой традиционной *каскадной модели разработки*, заключающейся в последовательном и полном выполнении этапов определения требований, проектирования, реализации и тестирования, становится нерациональным.

Для разработки системы обработки потоков текстовых сообщений применена *итеративная модель разработки*, заключающаяся в разбиении процесса на ряд итераций — краткосрочных проектов фиксированной длительности [107, p. 44]. Это позволило более точно определить функциональность системы и степень детализации при проектировании и реализации отдельных компонентов.

3.1.3. Определение требований к подсистеме визуализации

Как и для всех систем извлечения знаний из данных, ориентированных на человека, для систем ИАТ необходимо наличие хорошего уровня представления (presentation layer) для доступа пользователя к промежуточным данным и результатам. В составе качественной системы ИАТ должны присутствовать [11, с. 177]:

- средства для уточнения запроса (query refinement) посредством его параметризации через высокоуровневый и эргономичный графический интерфейс;
- доступ к низкоуровневому представлению запросов и данных с возможностью их модификации;
- административные средства для создания, модификации и управления иерархиями понятий, кластерами понятий и сущностями (предметной области).

Для систем, занимающихся обработкой больших объёмов текстовых данных, важно также обеспечение визуализации (visualization) и обзора (browsing) данных — высокоуровневых средств отображения пользователю полезной в данный момент информации в максимально удобном виде. В отличие от обычного графического интерфейса, эти средства обеспечивают [11, с. 177]:

- краткость (concision) — способность одновременного отображения большого числа разнотипных данных;
- относительность (relativity) и близость (proximity) — способность демонстрировать в результатах запроса кластеры, относительные размеры групп, схожесть и различие групп, выпадающие значения (outliers);
- концентрацию на контексте (focus with context) — взаимодействие в некоторым выбранным объектом с возможностью просмотра его положения и связей с контекстом;
- масштабируемость (zoomability) — способность легко и быстро перемещаться между микро- и макропредставлением;
- ориентацию на "правое полушарие" — предоставление пользователю не только заранее установленных методов работы с данными (обеспечивающими его намеренные и спланированные подходы к поиску нужной информации), но и поддержка его интуитивных, импровизационных когнитивных процессов идентификации закономерностей.

Часто применяемые средства визуализации включают [11, с. 194–224]: простые концептуальные графы (simple concept graphs); гистограммы; линейные диаграммы; круговые диаграммы (в первую очередь, круговые диаграммы ассоциаций — association-oriented circle graph); самоорганизующиеся карты; гиперболические деревья; трёхмерные эффекты; гибридные подходы.

3.1.4. Определение основных требований к функциональности системы обработки потока текстовых сообщений

В рамках итеративной модели разработки функциональные требования к системе рекомендуется выполнять в виде прецедентов — описания последо-

вательности взаимодействий с системой в процессе решения поставленных задач пользователями [107, с. 74]. Основным пользователем системы является аналитик, работающий на предприятии и занимающийся исследованием динамики текстовых потоков из различных источников с целью выработки управленческих решений. В терминологии разработки интеллектуальных систем этот пользователь именуется *лицом, принимающим решения*, или ЛПР.

С позиции ЛПР система обработки потоков текстовых сообщений должна обеспечивать выполнение следующих прецедентов:

1. *Добавление источников текстовых сообщений.* ЛПР определяет предметную область. ЛПР добавляет источники сообщений и настраивает их.
2. *Управление источниками сообщений.* ЛПР проводит мониторинг поступления сообщений из источников. ЛПР приостанавливает или запускает нужные источники.
3. *Создание тематик.* ЛПР просматривает тематический граф. ЛПР создаёт новую вершину в тематическом графе. ЛПР определяет свойства новой тематики и её связи с существующими тематиками.
4. *Просмотр информации о тематике.* ЛПР даёт команду на создание отчёта о тематике. Система создаёт и выводит на экран отчёт о тематике, её типе, типе применяемого классификатора, связанных тематиках, и классификационных признаках, временном ряде тематики.
5. *Просмотр информации о сообщении*. ЛПР даёт команду на создание отчёта о классификации сообщения. Система создаёт и выводит на экран отчёт о свойствах сообщения, этапах классификации, применённых правилах, результатах обработки сообщения.

3.2. Проектирование системы обработки потоков текстовых сообщений

3.2.1. Разработка архитектуры системы обработки потоков текстовых сообщений

Организация архитектуры компонентов влияет на модульность приложения, способ и трудоёмкость его разработки и сопровождения. В силу того,

что обработка потока текстовых сообщений является по своей природе событийным процессом, для разработки системы удобнее всего будет выбрать *событийно-ориентированную архитектуру* (event-driven architecture, EDA) организации компонентов.

В рамках EDA основные компоненты представляются в виде работающих в фоновом режиме модулей, наблюдающих за поступлением в систему сообщений согласно некоторым правилам (т. е. обладающих свойствами систем ОИП, см. п. 1.3.2).

На основании определённых в пп. 1.4.3 и 3.1 требований, математической модели системы обработки потоков текстовых сообщений (п. 2.1.2), а также предложенного алгоритма, предлагается следующая общая архитектура системы (Рисунок 19).

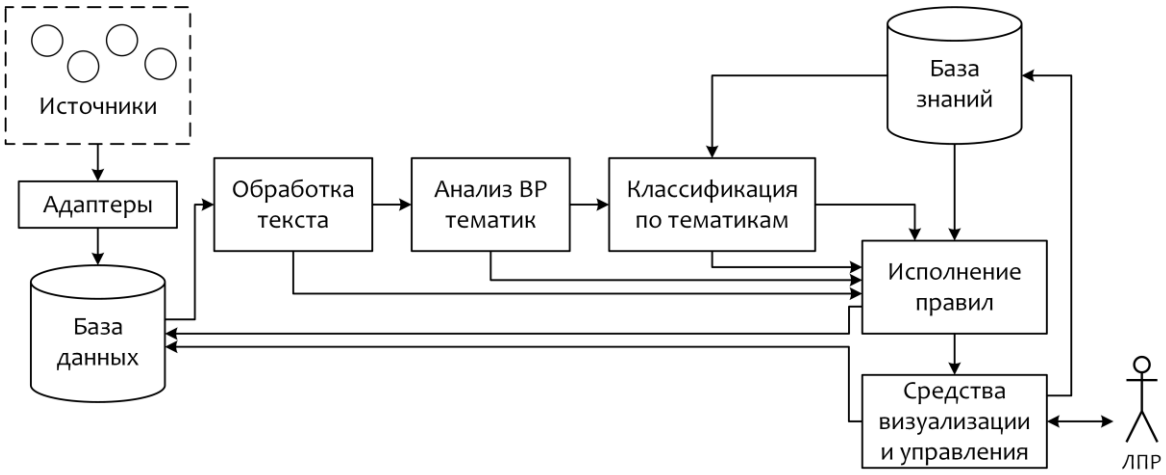


Рисунок 19. Архитектура системы обработки потоков текстовых сообщений

На приведённой схеме "источники" — это один или несколько источников, генерирующих потоки текстовых сообщений (могут быть виртуальными, т. е. созданными искусственно для целей тестирования). Состав и структура данных источников описаны в п. 3.5 и главе 4.

Компонент "*Адаптеры*" предназначен для считывания данных о наличии новых сообщений в источниках, а также самих текстов сообщений. Каждый адаптер активируется через определённые промежутки времени. При появлении нового сообщения адаптер извлекает из него нужную информацию, которую сохраняет в базе данных.

База данных обеспечивает хранение данных об источниках, поступивших сообщениях, а также обработанных (аннотированных и классифицированных) сообщениях.

База знаний предназначена для хранения информации о предметной области, включая структуру потока (иерархию тематик), структуру классификаторов и пользовательские правила.

Компонент "*Обработка текста*" предназначен для предварительной обработки текста сообщения (разметки и нормализации слов) и перевод текста в форму, пригодную для дальнейшего анализа.

Задачей компонента "*Анализ BP тематик*" является прогнозирование уровней всех тематик предметной области. Компонент активируется при поступлении нового сообщения; обновление модели прогнозирования происходит при наступлении нового периода.

Компонент "*Классификация по тематикам*" производит иерархическую классификацию поступившего сообщения (первичную и точную) и определение новизны тематик и сообщений.

Компонент "*Исполнение правил*" осуществляет принятие решений по результатам этапов обработки текста, анализа рядов тематик и классификации. Компонент активируется после каждого этапа обработки сообщения (при поступлении, после каждой итерации первичной классификации, после точной классификации).

Средства визуализации и управления осуществляют доступ пользователя к промежуточным и окончательным результатам обработки потока текстовых сообщений и позволяют ему производить настройку системы и процесса обработки потока.

3.2.3. Структура данных предметной области

Основными объектами предметной области являются сообщение и тематика. Взаимосвязь этих объектов, представленную ранее в виде математической модели (см. п. 2.1.1), можно представить в виде следующей диаграммы классов UML (Рисунок 20).

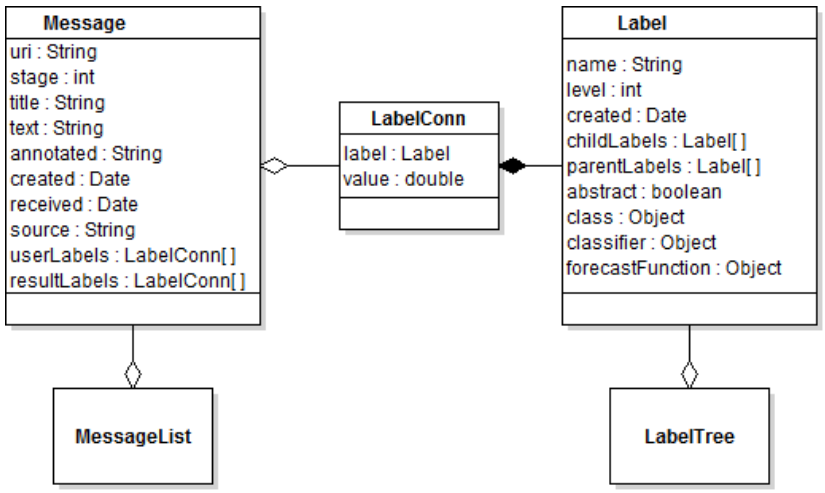


Рисунок 20. Диаграмма классов модели предметной области

Класс "Message" предназначен для хранения информации о сообщении: идентификаторе (адресе веб-страницы сообщения, uri), этапе обработки в системе (stage), заголовке страницы (title), тексте (text), обработанном тексте (annotated), времени создания (created), времени поступления в систему (received), названии источника (source), тематиках, определённым пользователем (userLabels) и тематиках, определённых классификатором (resultLabels).

Класс "LabelConn" содержит данные о вероятности принадлежности (выражаемой числом в интервале [0,1], value) сообщения тематике (label).

Класс "Label" содержит информацию о тематике: названии (name), расстоянии до корневой вершины (level), времени создания (created), дочерних тематиках (childLabels), родительских тематиках (parentLabels), типе тематики (абстрактной или конкретной, abstract), тематике как классе (class), тематике как классификаторе (classifier), модели прогнозирования (forecastFunction). Последние три объекта представляются соответствующими классами (см. п. 3.3.3).

Классы "MessageList" и "LabelTree" представляют собой, соответственно, массив всех поступивших в систему сообщений и массив определённых пользователем тематик.

3.3. Программная реализация системы обработки потоков текстовых сообщений

3.3.1. Выбор средств реализации системы обработки потоков текстовых сообщений

Для программной реализации проектных решений были использованы следующие средства (Таблица 7).

Таблица 7. Средства реализации системы обработки потоков текстовых сообщений

Задача	Средство	Версия
Операционная система	Microsoft Windows	6.1
Язык/платформа программирования	Java HotSpot	1.7.0_10
Среда разработки	Eclipse	4.3.1 (x64)
Система сборки	Apache Maven	3.0.4
Интерфейсная подсистема	Eclipse RCP 4	4.3.1
	SWT	3.102.1
Подсистема маршрутизации сообщений	Apache Camel	2.11.1
Средство протоколирования	SLF4J	1.6.6
	Log4J	1.2.17
Анализ текстов веб-страниц	Jsoup	1.7.2
Система управления базой данных	MongoDB	2.4.1 (x64)
	Mongo Java Driver	2.11.2
Объектно-документное отображение	Jongo[30]	0.4
Система исполнения правил	JBoss Drools Expert	5.5.0
Система обработки сложных событий	JBoss Drools Fusion	5.5.0
Визуализация графов	Eclipse Zest 2	0.1
Визуализация диаграмм	Highcharts	3.0.5
	jQuery	1.10.2
Создание отчётов	JMTE[31]	3.1
Лемматизация	Lucene Russian Morphology[32]	1.0
Аннотация текстов	XML	1.0

Выбор языка программирования Java обусловлен наличием большого числа готовых и открытых компонентов для создания систем бизнес-аналитики и корпоративных информационных систем, разработанных для этого языка, а также кроссплатформенностью получаемых приложений.

[30] http://jsoup.org/
[31] https://code.google.com/p/jmte/
[32] https://code.google.com/p/russianmorphology/

Разработка интерфейса на базе расширенной клиентской платформы (rich client platform) Eclipse RCP 4 позволила предоставить унифицированные, гибкие и удобные средства организации взаимодействия с пользователем.

Использование системы маршрутизации сообщений Apache Camel обусловлено выбором в п. 3.2.1 событийно-ориентированной архитектуры организации компонентов.

Для синтаксического разбора текстов веб-страниц на языке HTML использовалась библиотека Jsoup, что позволяет извлекать информацию с помощью селекторов CSS (каскадных таблиц стилей).

В качестве системы управления базой данных (СУБД) выбрана нереляционная документно-ориентированная система MongoDB, что позволяет естественным образом представлять хранимые сообщения и тематики. Взаимодействие с СУБД осуществляется посредством системы объектно-документного отображения (object-document mapping, ODM) Jongo, что существенно упрощает написание и модификацию кода, ответственного за чтение и запись данных в базу.

Подсистема, ответственная за исполнение правил предметной области, реализована с помощью компонентов Expert и Fusion комплекса JBoss Drools; последний из них позволяет использовать в правилах темпоральные отношения между текстовыми сообщениями в потоке.

За визуализацию направленного ациклического графа, представляющего иерархию тематик в потоке, отвечает система Eclipse Zest, являющаяся частью платформы Eclipse RCP. Интерактивная визуализация графиков производится с помощью языка JavaScript и библиотек jQuery и Highcharts.

Отчёты о классификации генерируются на языке HTML, подстановка данных в заготовленные шаблоны производится с использованием библиотеки JMTE (Java Minimal Template Engine).

Для морфологического анализа и лемматизации текстов сообщений на русском языке применяется библиотека Lucene Russian Morphology, реализующая гибридный подход к анализу текстов (см. п. 2.3). Морфологическая информация кодируется на основе внутренних аннотаций (см. п. 1.2.5) в формате XML в тексте сообщения.

3.3.2. Описание программных компонентов системы обработки потоков текстовых сообщений

По назначению программные компоненты (пакеты классов) системы были объединены в две большие группы: компоненты, реализующие базовые (существующие) методы интеллектуального анализа текстов, прогнозирования рядов и вспомогательные функции (общее внутреннее название — nlpml); и компоненты, реализующие предложенный алгоритм иерархической обработки и вспомогательные модули системы обработки (workbench). Состав этих двух групп компонентов приведён, соответственно, в Таблица 8 и 9.

Таблица 8. Состав программной библиотеки базового интеллектуального анализа (nlpml)

Имя пакета	Описание
annotator	Классы для предварительной обработки текста сообщения и аннотирования текста с помощью языка разметки XML.
classifier	Реализация мультиномиального и бернуллиевского НБК, словарей терминов, методов оценки и выбора признаков.
forecast	Классы для анализа и прогнозирования временных рядов.
similarity	Классы для оценки сходства сообщений на основе косинусной меры.
xml	Вспомогательные классы для работы с XML.

Таблица 9. Состав системы сбора и обработки потоков текстовых сообщений (workbench)

Имя пакета	Описание
application	Базовые классы приложения.
adapters	
camel	Классы, обеспечивающие маршрутизацию сообщений в пределах приложения.
classifier	Реализация многозначных и итеративных классификаторов, а также вспомогательные методы для обработки результатов классификации.
classifier.results	Классы для хранения и обработки результатов на отдельных итерациях иерархического классификатора.
database	Организация взаимодействия с базой данных.
dialogs	Диалоговые экранные формы.
drools	Компиляция и исполнение пользовательских правил.
model	Основные модели данных приложения (сообщение, тематика, класс).
model.graph	Вспомогательные модели данных для визуализации графиков.
model.jmte	Вспомогательные модели данных для отображения отчётов.
parts	Основные экранные формы приложения.

parts.handlers	Обработка событий графического интерфейса приложения (меню, панели инструментов).
parts.providers	Вспомогательные адаптеры данных для экранных форм.
settings	Модели для хранения информации о текущей сессии приложения.
source	Классы, представляющие предметную область, включая источники и интерфейсы для адаптеров.
source.adapters	Адаптеры для различных источников потоков текстовых сообщений.
test	Классы для экспериментальной оценки разработанного алгоритма.

3.3.3. Структура компонентов обработки сообщений

Компонент аннотирования текстов

Компонент аннотирования текстов (nlpml.annotator) состоит из двух классов (Рисунок 21).

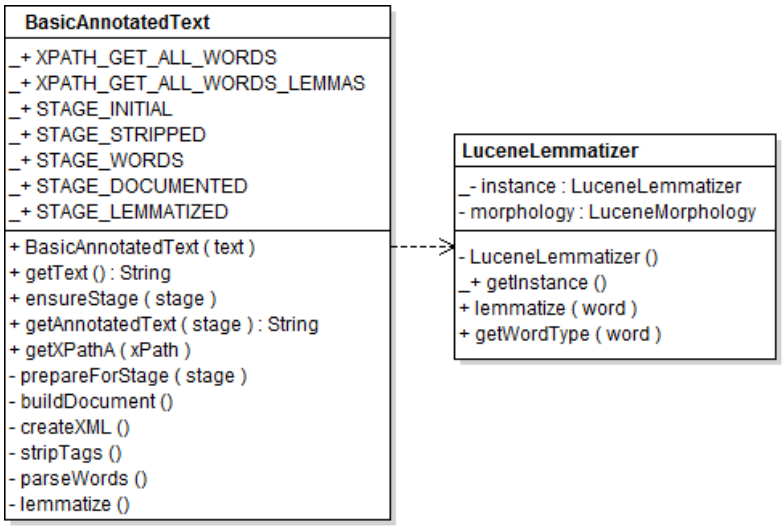


Рисунок 21. Диаграмма классов компонента аннотирования текстов

Класс LuceneLemmatizer обеспечивает взаимодействие с библиотекой лемматизации Lucene Russian Morphology. Класс BasicAnnotatedText обеспечивает аннотирование и поэтапную предварительную обработку текста. Для обработки текста необходимо указать требуемую стадию в методе getAnnotatedText с помощью одной из констант:

- STAGE_INITIAL — обработка не производится;
- STAGE_STRIPPED — из текста удаляются специальные символы, недопустимые для языка XML, буква "ё" заменяется на "е";
- STAGE_WORDS — производится базовая разметка слов и чисел;
- STAGE_DOCUMENTED — текст преобразуется в формат XML;

- STAGE_LEMMATIZED — производится морфологический анализ и лемматизация слов текста.

Пример работы функции аннотирования текста приведён в Таблица 10.

Для получения списков терминов текста или их лемм предназначен метод getXPathA, параметром которого является выражение XPath, соответствующее требуемому запросу[33].

Таблица 10. Пример последовательного аннотирования текста[34]

Стадия	Текст
INITIAL	И тут же на карте я нашёл под 32°40' северной широты и 167°50' западной долготы этот островок, открытый в 1801 году капитаном Креспо и значившийся на старинных испанских картах под названием Rocca de la Plata, что по-французски означает "Серебряный утёс".
WORDS	<annotated><text><v>И</v> <v>тут</v> <v>же</v> <v>на</v> <v>карте</v> <v>я</v> <v>нашел</v> <v>под</v> <n>32</n>°<n>40</n>' <v>северной</v> <v>широты</v> <v>и</v> <n>167</n>°<n>50</n>' <v>западной</v> <v>долготы</v> <v>этот</v> <v>островок</v>, <v>открытый</v> <v>в</v> <n>1801</n> <v>году</v> <v>капитаном</v> <v>Креспо</v> <v>и</v> <v>значившийся</v> <v>на</v> <v>старинных</v> <v>испанских</v> <v>картах</v> <v>под</v> <v>названием</v> <w>Rocca</w> <w>de</w> <w>la</w> <w>Plata</w>, <v>что</v> <v>по</v>-<v>французски</v> <v>означает</v> "<v>Серебряный</v> <v>утес</v>"<d>.</d></text></annotated>
LEMMATIZED	<?xml version="1.0" encoding="UTF-8" standalone="no"?> <annotated><text> <v lemma="и" type="R">И</v> <v lemma="тут" type="N">тут</v> <v lemma="же" type="R">же</v> <v lemma="на" type="R">на</v> <v lemma="карта" type="N">карте</v> <v lemma="я" type="P">я</v> <v lemma="найти" type="V">нашел</v> <v lemma="под" type="R">под</v> <n lemma="32" type="Z">32</n>°<n lemma="40" type="Z">40</n>' <v lemma="северный" type="A">северной</v> <v lemma="широта" type="N">широты</v> <v lemma="и" type="R">и</v> <n lemma="167" type="Z">167</n>°<n lemma="50" type="Z">50</n>' <v lemma="западный" type="A">западной</v> <v lemma="долгота" type="N">долготы</v> <v lemma="этот" type="P">этот</v> <v lemma="островок" type="N">островок</v>, <v lemma="открытый" type="A">открытый</v> <v lemma="в" type="R">в</v> <n lemma="1801" type="Z">1801</n> <v lemma="год" type="N">году</v> <v lemma="капитан" type="N">капитаном</v> <v lemma="креспо" type="I">Креспо</v> <v lemma="и" type="R">и</v> <v lemma="значиться" type="V">значившийся</v> <v lemma="на" type="R">на</v> <v lemma="старинный" type="A">старинных</v> <v lemma="испанский" type="A">испанских</v> <v lemma="карта" type="N">картах</v> <v lemma="под" type="R">под</v> <v lemma="название" type="N">названием</v> <w lemma="rocca" type="F">Rocca</w> <w lemma="de" type="F">de</w> <w lemma="la" type="F">la</w> <w lemma="plata" type="F">Plata</w>, <v lemma="что" type="R">что</v> <v lemma="по" type="R">по</v>-<v lemma="французский" type="A">французски</v> <v lemma="означать" type="V">означает</v> "<v lemma="серебряный" type="A">Серебряный</v> <v lemma="утес" type="N">утес</v>"<d>.</d> </text></annotated>

<u>Компонент прогнозирования временных рядов</u>

Компонент прогнозирования временных рядов (nlpml.forecast) включает 1 интерфейс и 3 класса (Рисунок 22).

[33] Например, для получения списка лемм слов русского языка используется выражение /annotated/text/v/@lemma

[34] Фрагмент текста взят из главы 15 романа Ж. Верна "20 тысяч лье под водой".

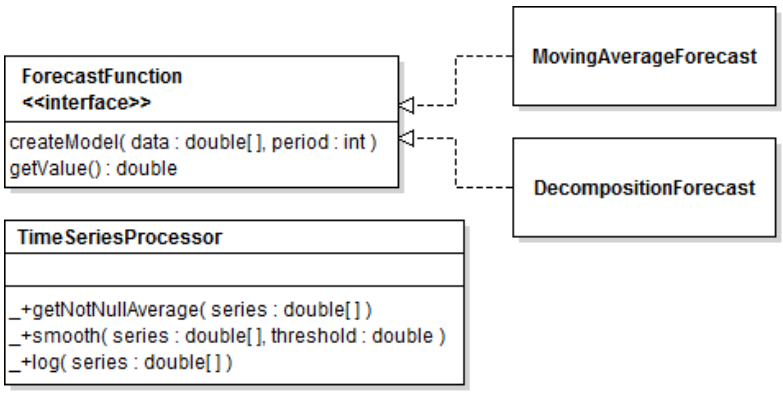


Рисунок 22. Диаграмма классов компонента прогнозирования временных рядов

Интерфейс ForecastFunction, представляющий абстрактную модель прогнозирования, описывает метод получения прогнозируемого на текущий период значения временного ряда тематики (getValue), а также фабричный метод для динамического создания модели (createModel). Классы MovingAverageForecast и DecompositionForecast реализуют, соответственно, способы прогнозирования на основе скользящего среднего и на основе декомпозиции ряда.

Класс TimeSeriesProcessor содержит общие методы обработки рядов: получение среднего по ненулевым значениям, сглаживания и логарифмирования.

Компонент байесовской классификации

Компонент байесовской классификации (nlpml.classifier) включает 2 интерфейса и 9 классов (Рисунок 23).

Ключевым интерфейсом компонента является NaiveBayesClassifier, который определяет общие методы для создания и исполнения классификатора. Процедура работы с классами данного компонента в общем случае включает следующие шаги:

1. на основе класса NaiveBayesMultinomialClassifier или NaiveBayesBernoulliClassifier создаётся экземпляр классификатора;
2. создаются экземпляры тематик (класс Clazz), которые добавляются к классификатору вызовом метода addClazz;
3. создаётся загрузчик документов (класс BatSimpleDocumentLoader);
4. создаются экземпляры документов путём вызова метода createUsingDocumentLoader класса Document; в качестве параметров передаются загруз-

чик документов и аннотированные тексты — экземпляры класса BasicAnnotatedText; полученные документы добавляются в качестве обучающих к классификатору путём вызова метода addDocument;

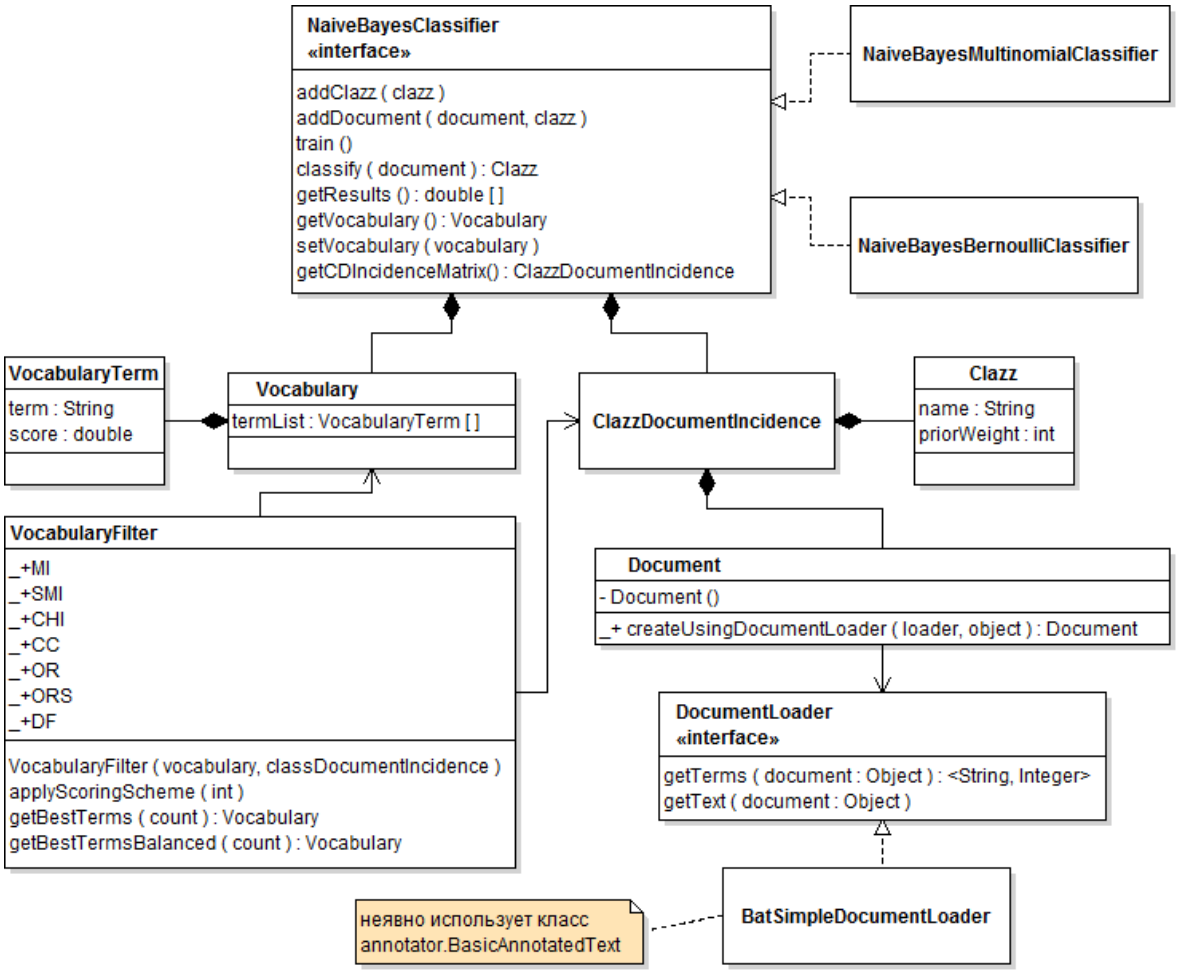


Рисунок 23. Диаграмма классов компонента баейсовской классификации

5. вызовом метода getVocabulary у классификатора получаем словарь (множество всех признаков) классификатора;
6. вызовом метода getCDIncidenceMatrix у классификатора получаем матрицу смежности, содержащую информацию о принадлежности признаков к документам и классам;
7. на основе словаря и матрицы смежности создаётся экземпляр фильтра признаков (класс VocabularyFilter);
8. вызовом метода applyScoringScheme у экземпляра фильтра признаков применяем метрику оценки признаков; вид метрики (MI, SMI, CHI, CC, OR, ORS, DF) задаётся в качестве параметра;

9. вызовом метода getBestTerms или getBestTermsBalanced получаем новый словарь признаков, который назначается классификатору;
10. вызовом метода train классификатор обучается;
11. вызовом метода classify производится классификация документа;
12. вызовом метода getResults получаем вероятности принадлежности последнего классифицированного документа к классам.

Компоненты иерархической классификации

Компоненты иерархической классификации представлены пакетами workbench.classifier и workbench.classifier.results. Их структура приведена на Рисунок 24 и Рисунок 25, соответственно.

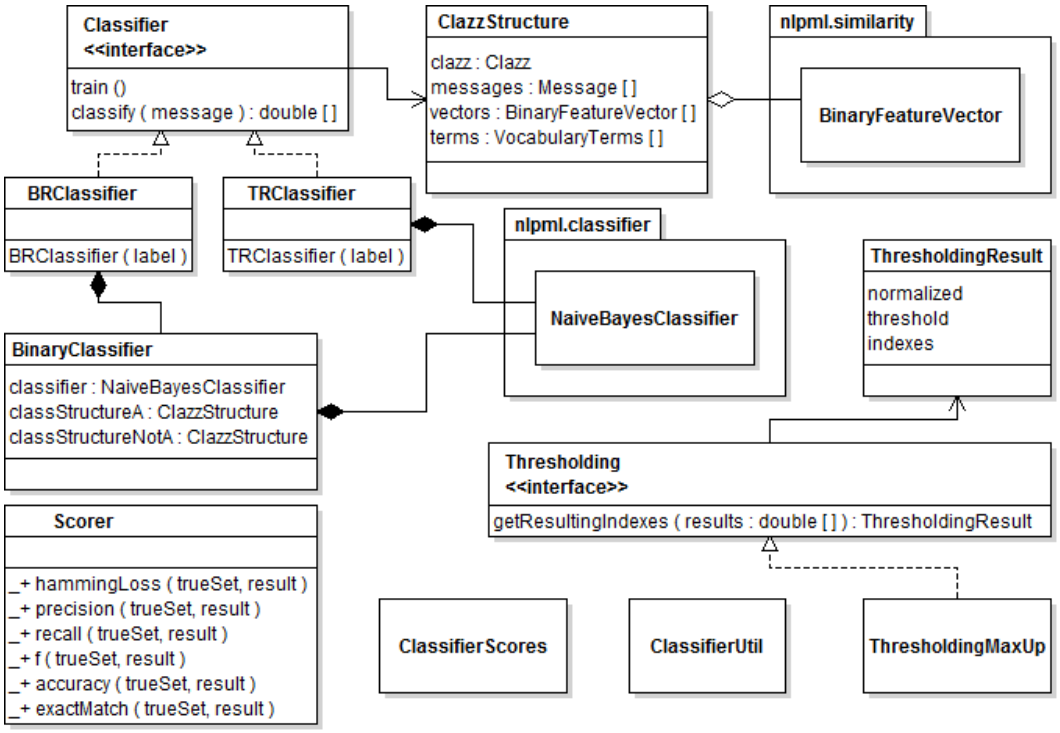


Рисунок 24. Структура пакета workbench.classifier

Основным классом первого из указанных компонентов является TRClassifier, реализующий создание, обучение и исполнение многозначного порогового НБК для заданной тематики. Каждый класс (подтематика) в классификаторе представляется классом ClazzStructure. За вычисление динамического порога отвечает класс ThresholdingMaxUp.

Основным классом второго компонента является ClassificationTask, реализующий первичную и точную классификацию для заданного сообщения. Результаты первичной классификации на каждой итерации представляются

классом ForwardIterationResult, результаты выбора наиболее подходящей листовой тематики — LeafIterationResult, результаты точной классификации — классом BackwordIterationResult.

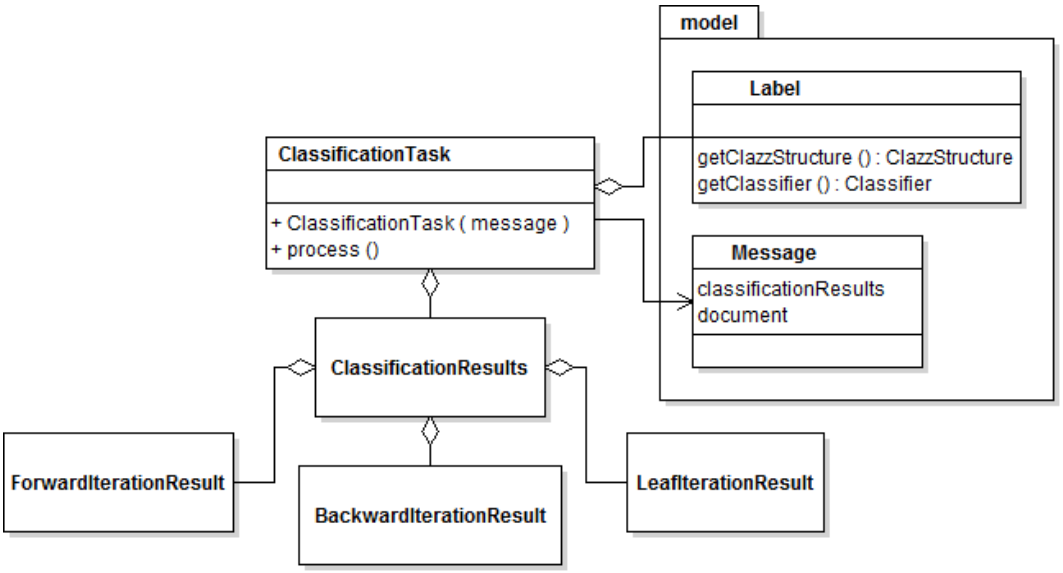


Рисунок 25. Структура пакета workbench.classifier.results

Компонент исполнения правил предметной области

Компонент исполнения правил (workbench.drools) включает основных 3 класса (Рисунок 26)

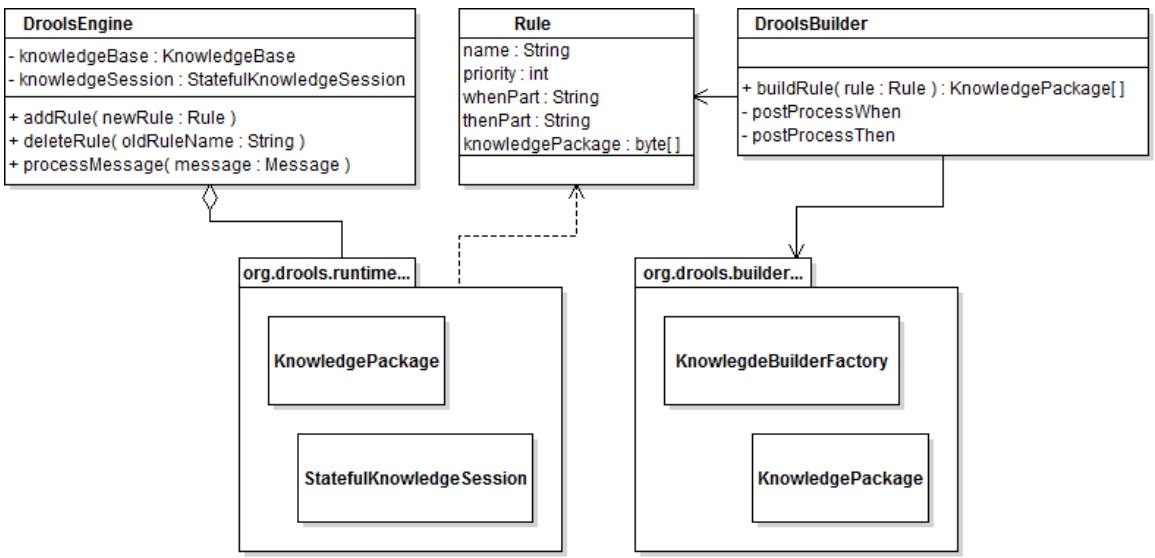


Рисунок 26. Диаграмма классов компонента исполнения правил

Класс Rule представляет пользовательское правило, определённое четырьмя параметрами: названием (name), приоритетом (priority), антецедентом (whenPart) и консеквентом (thenPart).

Класс DroolsBuilder отвечает за компиляцию правил, т. е. преобразование их в двоичный код, который выполняется машиной Drools. Результат компиляции помещается в поле knowledgePackage объекта класса Rule (с целью сохранения в базе знаний).

Класс DroolsEngine осуществляет загрузку и удаление правил в рабочую память машины исполнения Drools. При вызове метода processMessage правила, находящиеся в рабочей памяти, применяются к переданному в качестве параметра сообщению.

3.3.4. Адаптеры для источников сообщений

Для чтения сообщений из различных источников (файлов, баз данных, веб-сайтов) в системе предусмотрен механизм адаптеров. Структура классов, отвечающих за работу адаптеров, приведена на Рисунок 27.

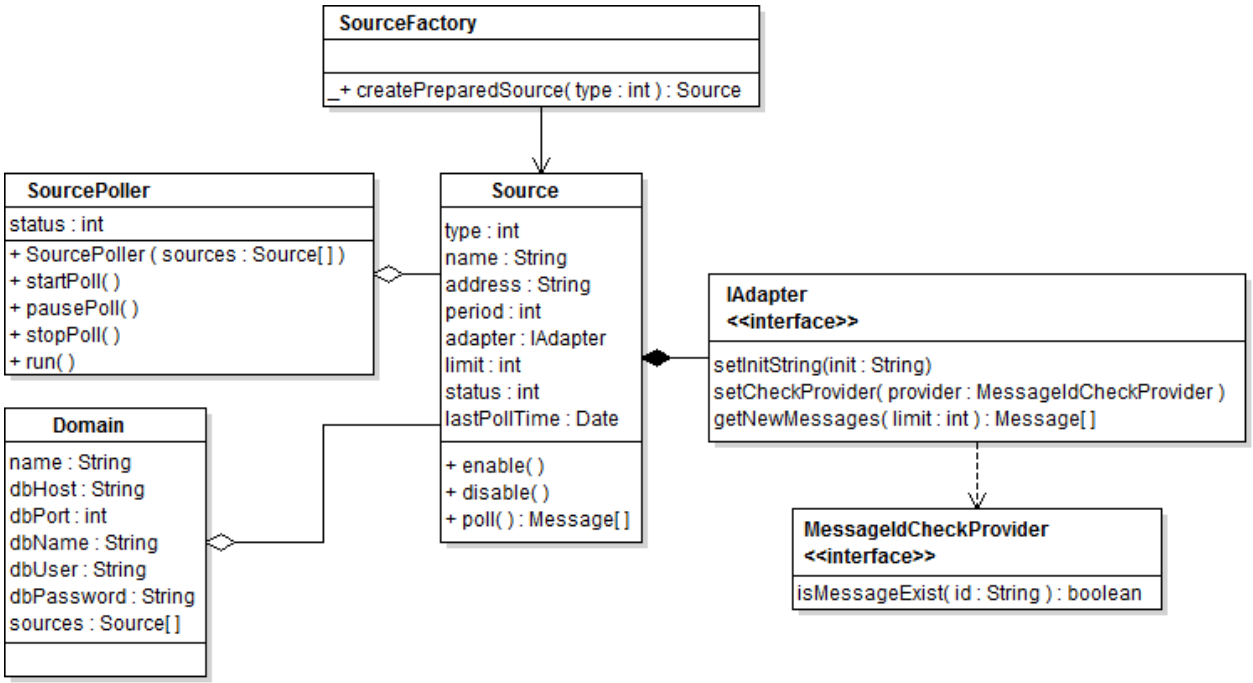


Рисунок 27. Диаграмма классов для адаптеров к источникам сообщений

Класс Domain представляет собой предметную область, с которой работает пользователь в течение одного сеанса. Предметная область определяется названием (name), параметрами связи с базой данных (для сохранения сообщений, тематик и правил) и множеством источников (sources).

Класс Source представляет информацию об источнике данных и способы взаимодействия с ним. Поле adapter содержит ссылку на объект-адаптер, применяемый для чтения и анализа сообщений из данного источника. Метод poll

опрашивает источник и возвращает множество сообщений, которые появились в источнике с момента предыдущего опроса.

Интерфейс IAdapter определяет базовые методы, которые должен реализовывать каждый адаптер. Метод setInitString устанавливает пользовательские параметры адаптера (например, имя файла), метод setCheckProvider устанавливает объект базы данных сообщений для проверки того, не было ли новое сообщение уже загружено в базу, метод getNewMessages возвращает массив новых сообщений.

Класс SourcePoller отвечает за опрос источников сообщений; процедуру опроса можно приостановить или отменить методами pausePoll и stopPoll.

Класс SourceFactory реализует метод, возвращающий предзаготовленные в системе источники для сообщений.

3.4. Описание пользовательского интерфейса системы обработки потоков текстовых сообщений

Внешний вид основного окна разработанного программного комплекса обработки потоков текстовых сообщений показан на Рисунок 28.

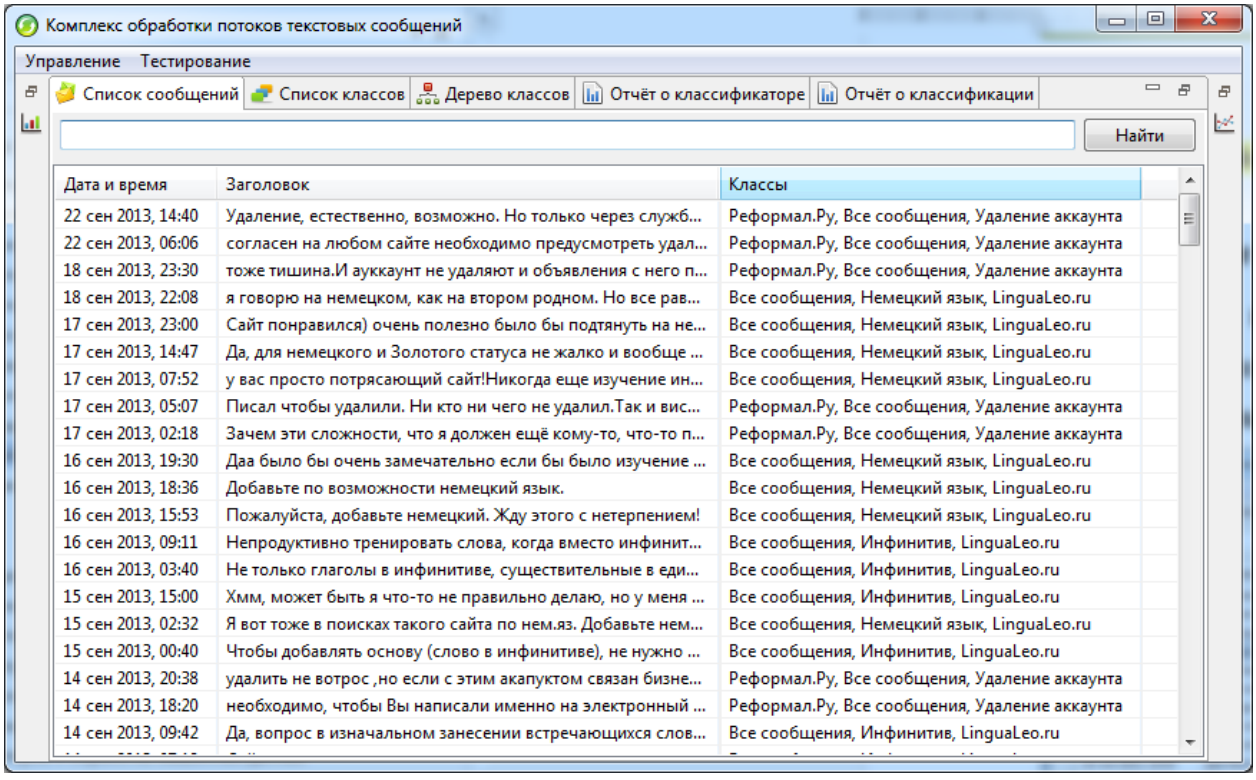


Рисунок 28. Внешний вид основного окна программного комплекса обработки потоков текстовых сообщений

В меню "Управление" можно переключиться на работу с другой предметной областью. Меню "Тестирование" создано с целью экспериментальной оценки эффективности разработанного алгоритма (п. 3.5). В основном окне пользователю доступны следующие вкладки:

- "Список сообщений" — открывается при запуске программы; отображает список последних сообщений, поступивших в систему.
- "Список классов" — отображает список тематик предметной области с указанием их родительских и дочерних тематик.
- "Дерево классов" — отображает иерархию тематик предметной области в виде направленного ациклического графа. При выборе тематик с помощью мыши на специальной панели показывается график количества сообщений по дням (см. Рисунок 29).

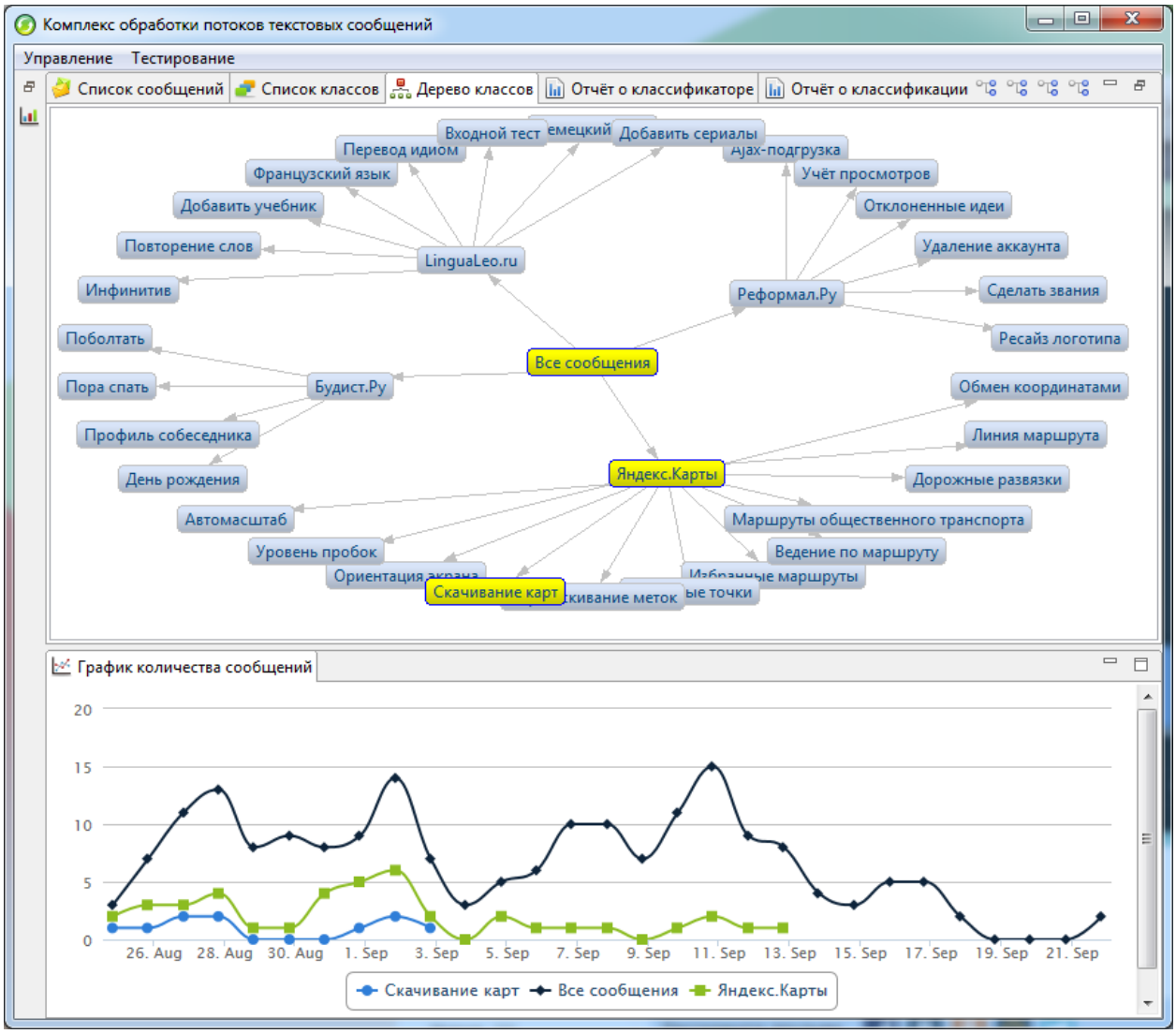


Рисунок 29. Средства визуализации комплекса обработки потоков

- "Отчёт о классификаторе" — отображает статистические данные о выбранной тематике, включая списки классификационных признаков с указанием их полезности.
- "Отчёт о классификации" — отображает данные, полученные на каждой итерации в ходе первичной и точной классификации последнего сообщения.

На вкладке "Дерево классов" пользователь может добавлять, редактировать и удалять тематики, а также определять и удалять связи между тематиками. Внешний вид экранной формы редактирования тематики представлен на Рисунок 30.

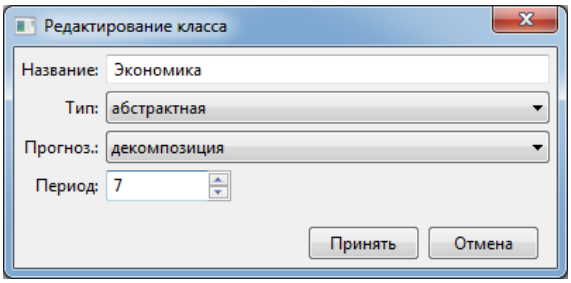


Рисунок 30. Форма редактирования параметров тематики

Информация о текущей предметной области в виде списка адаптеров и их свойств представлена на специальной панели (Рисунок 31). При вызове контекстного меню пользователь может добавить или приостановить новый источник сообщений.

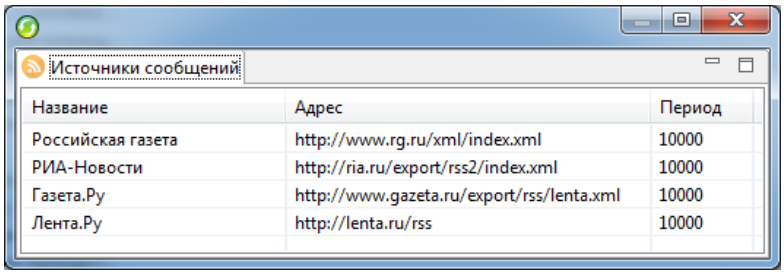


Рисунок 31. Список действующих источников сообщений

3.5. Экспериментальная оценка эффективности алгоритма обработки потоков текстовых сообщений

Тестовые коллекции текстовых сообщений (см. п. 1.2.4), используемые в современных исследованиях эффективности алгоритмов ИАТ, в большинстве своём не подходят для оценки алгоритмов обработки потоков сообщений по ряду причин:

- англоязычные коллекции не отражают специфику текстов на русском языке (в частности, особенности морфологии);
- в статических коллекциях не указывается дата создания текста.

Для экспериментальной оценки предложенного алгоритма (и разработанной системы) на основе новостных сообщений с сайта "Российской газеты" была сформирована коллекция текстовых сообщений, учитывающая дату их создания. В тестовую базу данных были загружены сообщения в период с 1 января 2013 по 30 апреля 2013 гг. В качестве контрольных тематик сообщений принимались значения поля "тематики" с соответствующей веб-страницы новости.[35] После загрузки сообщений из коллекции были удалены непредметные[36] тематики и тематики, имеющие меньше 6 сообщений. Полученная коллекция включала 3416 сообщений и 55 тематик. Множество тематик вручную было выстроено в следующую иерархию (Рисунок 32).

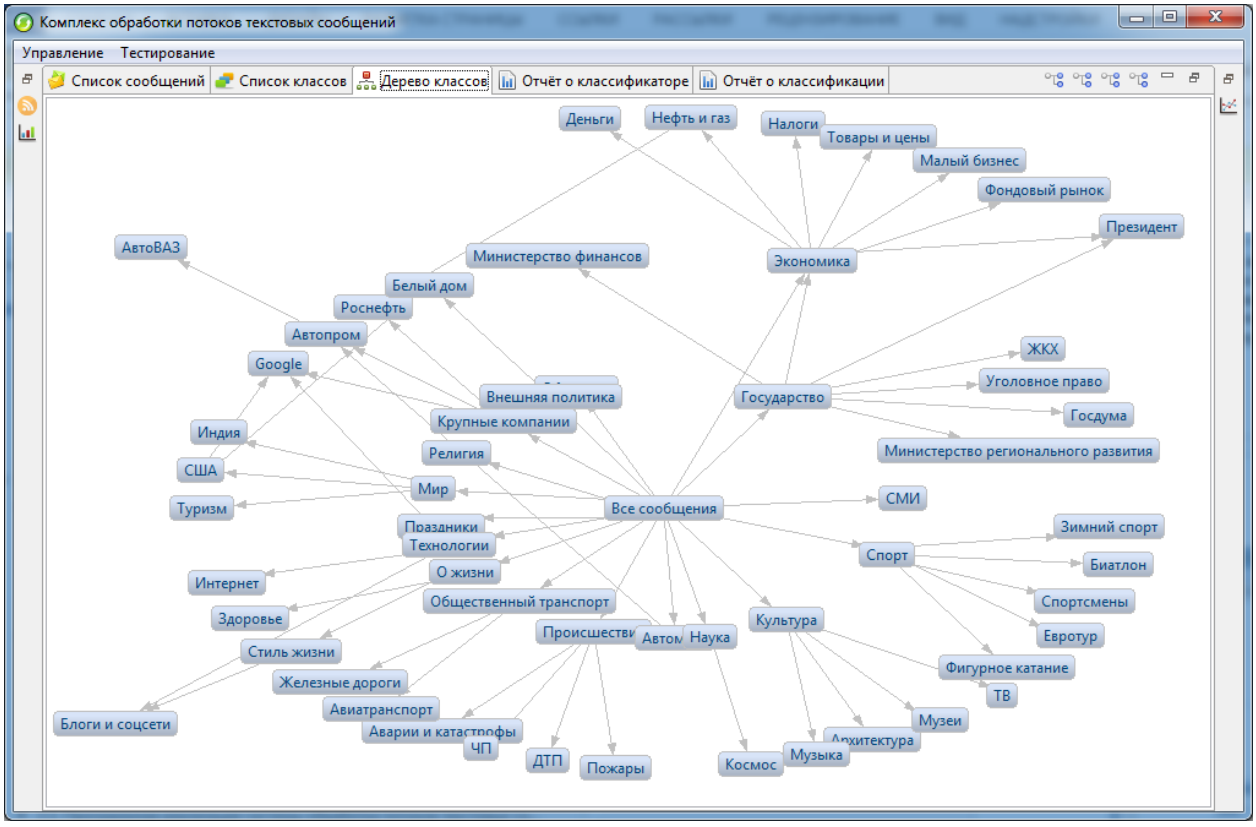


Рисунок 32. Иерархия тематик для экспериментальной оценки

[35] Каждая новость, публикуемая на сайте "Российской газеты", уже отнесена к одной или нескольким тематикам самим автором этой новости. Стоит отметить, что для некоторых статей множества тематик были указаны не полностью. В целях чистоты эксперимента корректировка отнесения сообщений к тематикам не производилась.

[36] Например, тематики, отображающие тип содержания новости или её автора.

Оценка эффективности производилась по методу, предложенному в п. 2.8. В эксперименте оценивалась способность предложенного алгоритма классифицировать более новые сообщения на основании известной классификации более старых сообщений. За начальную дату *I* было взято 1 января 2013 года, период рассмотрения *P* равнялся 14 дням. Результаты тестирования разработанного алгоритма по сравнению с методом бинарного соответствия[37] показаны на рРисунок 33 и Рисунок 34.

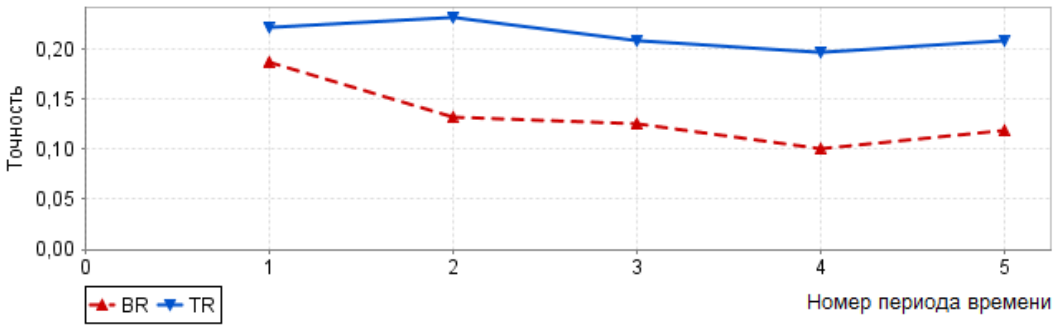


Рисунок 33. Оценка эффективности алгоритма бинарного соответствия (BR) и разработанного алгоритма (TR) по метрике точности с периодом 14 дней

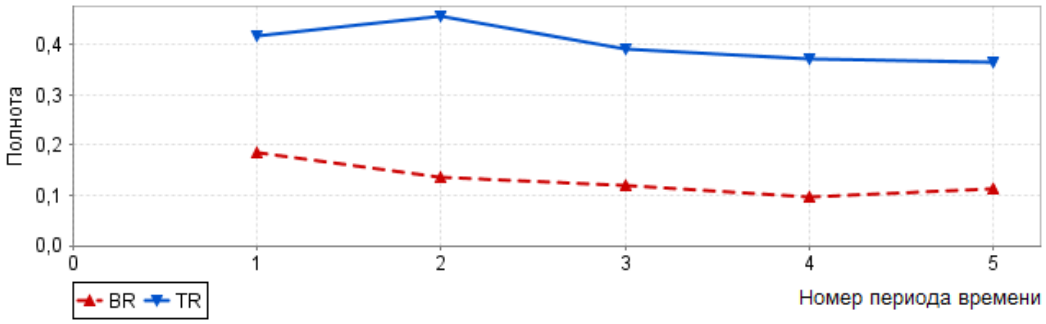


Рисунок 34. Оценка эффективности алгоритма бинарного соответствия (BR) и разработанного алгоритма (TR) по метрике полноты с периодом 14 дней

Результаты показывают, что разработанный метод большей точностью (примерно в 2 раза) и полнотой (примерно в 3 раза), чем метод бинарного соответствия. Лучшая оценка по полноте объясняется тем, что результатом разработанного алгоритма почти всегда является множество тематик (характеризующих сообщение на разных уровнях классификации).

[37] Тестирование остальных алгоритмов многозначной классификации (п. 2.4.2), а именно, методов попарного соответствия и комбинации тем, не представлялось возможным по причине большого количества участвующих в них классификаторов (как следствие большого количества тематик).

3.6. Выводы

1. Определены требования, предъявляемые к архитектуре систем бизнес-аналитики, действующим в рамках корпоративных информационных систем, процессу их разработки и интерфейсу пользователя. В качестве модели разработки программного комплекса обработки потоков текстовых сообщений выбран итеративный процесс.
2. Выработаны базовые проектные решения по архитектуре системы, пользовательскому интерфейсу, представлению данных предметной области.
3. Приведено описание общей структуры и программной реализации основных компонентов системы, позволяющей обрабатывать потоки текстовых сообщений, поступающих в оперативном режиме из различных источников. Разработанный программный комплекс позволяет учитывать знания и особенности предметной области, предоставляет оператору средства гибкой настройки параметров используемых алгоритмов в процессе работы и обладает современными средствами визуализации промежуточных и окончательных результатов обработки сообщений в потоке. Функционал системы позволяет, в частности, производить классификацию сообщений, просматривать и редактировать тематический граф, просматривать динамику тематик.
4. Произведена экспериментальная оценка эффективности разработанного в главе 2 алгоритма с алгоритмом многозначной классификации на основе бинарного соответствия. Показано, точность классификации для предложенного алгоритма выше, чем у метода бинарного соответствия, примерно в 2 раза; полнота — в 3 раза.

Глава 4. Применение системы в практических задачах обработки потоков

4.1. Обработка потока новостных сообщений

В условиях быстрого изменения тенденций экономики, общества и политики, происходящего в наши дни, множество предприятий и организаций заинтересованы в мониторинге ситуаций в этих областях по сообщениям, публикуемым в средствах массовой информации.

Для исследования возможностей применения системы обработки потоков текстовых сообщений были выбраны следующие новостные сайты:

- "Российская газета" (rg.ru) — электронная версия официального издания Правительства Российской Федерации. Преимуществами ресурса, с точки зрения обработки информации, являются: удобная структура текста статьи, наличие меток, относящих статьи к определённым сюжетам и тематикам.
- "РИА Новости" (ria.ru) — одно из крупнейших информационных агентств в мире. Тексты статей содержат наборы меток ("тэгов", tags), которые позволяют отнести статьи к определённым персоналиям, событиям, явлениям, местам и т. п.
- "Газета.Ру" (gazeta.ru) — одно из наиболее посещаемых информационных изданий в российском сегменте Интернета. Тексты статей хорошо структурированы и снабжены списком меток; адреса страниц отражают тематики и дату публикации новостной статьи.
- "Лента.Ру" (lenta.ru) — одно из наиболее цитируемых электронных изданий, и одно из наиболее часто посещаемых в Европе. К преимуществам ресурса стоит отнести удобные URL-адреса страниц, отражающие дату публикации, тематику и сюжет статьи.

Для извлечения информации из потока статей с указанных новостных сайтов на основе интерфейса IAdapter были создан адаптеры RgOnlineAdapter, RiaOnlineAdapter, GazetaOnlineAdapter, LentaOnlineAdapter. Списки извлекаемых элементов для этих адаптеров перечислены в табл. 11–14.

Таблица 11. Извлечение информации адаптером RgOnlineAdapter

Адрес страницы, формат	Семантика элемента	Извлекаемый элемент (CSS-селектор или регулярное выражение)	Извлекаемый атрибут
`http://www.rg.ru/xml/index.xml` (RSS)	список новостей	`item > guid`	текст
`http://www.rg.ru/<год>/<месяц>/<день>/<название>.html` (HTML)	текст	`div.main-text`	текст
	заголовок	`div.head-ar h1`	текст
	дата	`div.article_info span.tik`	текст
		`(\d{2}).(\d{2}).(\d{4}),\s*(\d{2}):(\d{2})`	—

Таблица 12. Извлечение информации адаптером RiaOnlineAdapter

Адрес страницы, формат	Семантика элемента	Извлекаемый элемент (CSS-селектор или регулярное выражение)	Извлекаемый атрибут
`http://ria.ru/export/rss2/index.xml` (RSS)	список новостей	`item > guid`	текст
`http://ria.ru/<тематика>/<год><месяц><день>/<номер>.html` (HTML)	текст	`div.article_full_content p`	текст
	заголовок	`h1.article_header_title`	текст
	дата	`time.article_header_date`	атрибут datetime
		`(\d{4})-(\d{2})-(\d{2})T(\d{2}):(\d{2})`	—

Таблица 13. Извлечение информации адаптером GazetaOnlineAdapter

Адрес страницы, формат	Семантика элемента	Извлекаемый элемент (CSS-селектор или регулярное выражение)	Извлекаемый атрибут
`http://www.gazeta.ru/export/rss/lenta.xml` (RSS)	список новостей	`item > guid`	текст
`http://www.gazeta.ru/<категория>/<раздел>/<год>/<месяц>/<число>/n_<номер>.shtml` (HTML)	текст	`.b-article div.text`	текст
	заголовок	`h1.article_news_name` или `h2.article_name`	текст
	дата	`.b-article time`	атрибут pubdate
		`(\d{4})-(\d{2})-(\d{2})T(\d{2}):(\d{2})`	—

Таблица 14. Извлечение информации адаптером LentaOnlineAdapter

Адрес страницы, формат	Семантика элемента	Извлекаемый элемент (CSS-селектор или регулярное выражение)	Извлекаемый атрибут
`http://lenta.ru/rss` (RSS)	список новостей	`item > guid`	текст
`http://lenta.ru/news/<год>/<месяц>/<число>/<тематика>/` (HTML)	текст	`div.b-text`	текст
	заголовок	`h1.b-topic__title`	текст
	дата	`div.b-topic__info time`	текст
		`(\d{1,2}):(\d{1,2})\W*(\d{1,2})\s*([а-яА-Я]+)\s*(\d{4})`	—

На основании работы системы обработки потоков текстовых сообщений было показано, что систему можно использовать для исследования появления новых важных событий, анализа взаимовлияния новостных тематик, классификации и фильтрации сообщений, наблюдения за развитием интересующих тем.

Разработанный программный комплекс не производит анализ метаинформации, содержащийся в сообщении (предполагается, что сообщение представляется только текстом, заголовком и датой). В частности, игнорируется "авторское" отнесение сообщения к тем или иным категориям. С одной стороны, это может быть полезно в тех случаях, когда имеются сомнения в полноте и непротиворечивости обозначенных автором категорий, тегов, сюжетов и т. д. С другой стороны, учёт метаинформации является перспективным направлением дальнейших исследований.

4.2. Обработка потока заявок на модификацию программных продуктов

В процессе сопровождения программного обеспечения (ПО) возникает задача обработки сообщений, поступаемых по каналам обратной связи от пользователей. Подобные сообщения, в частности, могут содержать запросы на расширение функциональности ПО или отчёты об ошибках, возникающих в процессе его работы [108]. Для сбора запросов на модификацию ПО широко применяются два подхода:

- Прикладные системы отслеживания ошибок (bug tracking system), ориентированные на сбор текстовых запросов на исправление ошибок через общедоступный ресурс (часто — веб-сайт).
- Средства поддержания обратной связи, интегрированные в конкретный программный продукт или линейку ПО от одного производителя.

Для исследования возможностей применения разработанной системы для обработки заявок на модификацию программных продуктов была сделана выборка сообщений пользователей с сайта "Реформал.Ру" (reformal.ru), содержащих пожелания относительно ряда продуктов и услуг. Экспериментальные данные были сохранены в текстовом файле формата CSV. В связи с малочисленностью (и, как следствием, разреженностью во времени) выборки сообщениям были предписаны искусственные моменты времени создания. С помощью разработанной системы был сформирован тематический граф пожеланий пользователей (см. Рисунок 29).

В процессе работы было показано, что систему можно использовать для классификации поступающих заявок по продукту и проблеме, а также выявлять возникающие проблемы по множеству сообщений.

В некоторых случаях сообщение пользователя о проблеме в ПО в текстовом виде может дополняться выдержкой из программного протокола работы ПО. В работе [109] описан опыт создания макета системы, который производит анализ автоматически сгенерированных записей протокола для идентификации и предупреждения ошибок. Макет состоит из программного агента (осуществляющего сбор программных протоколов) и системы управления бизнес-правилами (которая используется для классификации новых ошибок и управления обратной связью с пользователем), связанной с базой прецедентов ошибок.

Перспективным направлением исследований является повышение качества сопровождения ПО за счёт комплексного анализа текстов сообщений и автоматически сгенерированных программных протоколов [110]. Это становится особенно важным, если сообщения пользователей содержат ошибки (в том числе орфографические), неполную или неточную информацию.

4.3. Обработка потока обращений пользователей юридических форумов

Частью федеральной программы "Информационное общество (2011—2020 годы)" является проект *электронного правительства*. Одной из целей проекта является "реализация возможностей влияния граждан на деятельность государственных структур и общественного контроля над работой органов власти" [111, p. 63]. Одной из наиболее востребованных услуг данного проекта является рассмотрение электронных обращений граждан в официальные органы. Разработанный комплекс можно применить для анализа потока обращений граждан, а именно классификации по предмету вопроса, определения важности и приоритетов обращений, анализа динамики появления и закрытия вопросов, выявления социальных проблем.

Исследование возможностей системы производилось на максимально близкой к условиям электронного правительства задаче анализа потока обращений пользователей в электронные сервисы юридических консультаций.

Для эксперимента был выбран сайт "Правовед.Ру" (pravoved.ru), на котором пользователи имеют возможность задать вопрос юристу. С точки зрения исследования преимуществами ресурса являются хорошая структурированность сайта и кода веб-страниц, а также наличие категорий вопросов.

Для обработки сообщений с данного сайта был создан адаптер (PravovedOnlineAdapter), структура извлекаемых данных для данного адаптера представлена в Таблица 15.

Таблица 15. Извлечение информации адаптером PravovedOnlineAdapter

Адрес страницы, формат	Семантика элемента	Извлекаемый элемент (CSS-селектор или регулярное выражение)	Извлекаемый атрибут
`http://pravoved.ru/questions/` (HTML)	новые вопросы	`.questionBody > h3 > a[href]`	атрибут href
`http://pravoved.ru/question/<номер>/` (HTML)	текст	`.questionBody > p`	текст
	заголовок	`.questionBody > h1`	текст
	дата	`.questionBody > .questionInfo > .questionDate`	текст
		`(\d{1,2}) (\W+) (\d{4}), (\d{2}):(\d{2})`	—

Результаты применения адаптера в системе обработки потоков текстовых сообщений показали, что систему можно использовать для классификации поступающих сообщений по интересующим посетителей вопросам. Особенностью предметной области являлась слабая динамика тем (так как темы на юридических форумах соответствуют категориям законов, которые изменяются чрезвычайно редко).

В то же время сообщения в рамках электронного правительства тематика сообщений может изменяться достаточно сильно, и быть сравнимой с динамикой тем новостей (см. п. 4.1). При анализе текстов пользователей становится необходимым также учитывать орфографические и стилистические ошибки, которые могут содержаться в текстах сообщений.

4.4. Обработка потока статей в социальных медиа-ресурсах

Википедия — интернет-энциклопедия, статьи которой создаются и редактируются по принципу "вики", то есть самими пользователями. Согласно данным информационного портала "Alexa"[38] по состоянию на октябрь 2013 года Википедия занимает 9-е место по посещаемости пользователей из России, являясь наиболее популярным справочником информации в Интернете. Информация о новых событиях вносится в Википедию оперативно, поэтому для ряда тематик (особенно касающихся политики и общественных явлений) она выполняет также функции сетевого СМИ.

В среднем в проекте "Википедия" за один день создаются 400 новых статей, и производится 500 000 редактирований существующих [112]. Средний размер создаваемой статьи — 2000 символов. Как создаваемые, так и редактируемые статьи можно рассматривать в виде потока текстовых сообщений.

Для обработки потока новых статей в проекте "Википедия" на основе интерфейса IAdapter был создан адаптер WikipediaOnlineAdapter. Структура извлекаемых данных представлена в Таблица 16.

[38] http://www.alexa.com/topsites/countries/RU

Таблица 16. Извлечение информации адаптером WikipediaOnlineAdapter

Адрес страницы, формат	Семантика элемента	Извлекаемый элемент (CSS-селектор или регулярное выражение)	Извлекаемый атрибут
`http://ru.wikipedia.org/w/index.php?title=Special:NewPages&feed=atom` (Atom)	новые статьи	`feed > entry > link`	атрибут href
`http://ru.wikipedia.org/wiki/<название>`	текст	`#mw-content-text`	текст
	заголовок	`h1`	текст
	дата	`#footer-info-lastmod`	текст
		`(\d{2}):(\d{2}), (\d{1,2}) (\W+) (\d{4})`	—

Результаты исследования показывают, что разработанная система может найти применение при анализе контента социальных медиа-ресурсов, например, статей в онлайн-энциклопедиях или сообщений в блогах: классификации создаваемых пользователями заметок и статей по интересующему предмету, обнаружении спама, выявлении интересов групп пользователей.

Вместе с тем для качественной обработки сообщений специального типа, таких как сообщений в системах микроблогов, требуется включение в систему дополнительных модулей исправления ошибок, извлечения информации, поиска синонимов и т. д. [78] В условиях растущей популярности социальных сервисов данное направление исследований является наиболее перспективным из представленных.

4.5. Выводы

1. Описывается применение разработанного программного комплекса для обработки потоков новостных сообщений, агрегируемых в реальном времени с сайтов "Российская газета", "РИА Новости", "Газета.Ру", "Лента.Ру".
2. Приводится пример работы комплекса с потоком пожеланий пользователей относительно программных продуктов с сайта "Реформал.Ру". Показано, что система может быть применена для обработки заявок пользователей в системах сопровождения программной продукции.
3. Описывается применение комплекса для оперативной обработки запросов пользователей в открытый сервис юридических консультаций "Правовед.Ру". Показано, что система может быть использована для обработки потока обращений граждан в рамках системы электронного правительства.
4. Приводится пример работы комплекса для оперативной обработки текстов научно-популярных статей, создаваемых в рамках проекта "Википедия".
5. Перспективным направлением исследований является дальнейшее повышение эффективности обработки сообщений специального типа (текстов с орфографическими ошибками; сообщений, содержащих информацию о возможных категориях; сообщений, содержащих машиночитаемые данные; коротких сообщений в сервисах микроблогов) за счёт применения методов глубинной ОЕЯ.

Заключение

Анализ современных методов обработки текстов на естественном языке показал, что большинство из них ориентированы на работу с коллекциями документов, изменения в которые вносятся относительно редко. Для повышения эффективности обработки текстовых сообщений, поступающих в потоковом режиме, в дополнение к классическим методам ОЕЯ и методам обработки текстов на основе машинного обучения были привлечены:

- методы моделирования онтологий;
- методы анализа временных рядов;
- методы обработки сложных событий;
- методы интеллектуального анализа последовательностей.

К основным результатам работы можно отнести следующие:

1. Предложены математические модели потока текстовых сообщений в виде направленного ациклического графа, позволяющего отображать и исследовать зависимости между тематиками, и системы обработки потоков текстовых сообщений, являющуюся основой для разработки алгоритма.
2. Разработан метод многозначной классификации на основе наивного байесовского подхода, вычисления динамической пороговой оценки и сбалансированной процедуры выбора признаков.
3. Предложены оценки степени новизны сообщения по отношению к тематике и новизны тематики, позволяющие автоматизировать процедуру модификации тематического графа и предоставляющие пользователю дополнительную информацию о потоке сообщений.
4. Разработан итерационный оперативный алгоритм обработки потока текстовых сообщений с частичным обучением, позволяющий производить классификацию текстовых сообщений, обнаружение возникающих тематик и вычисление степени новизны.

5. На основе предложенных решений создан программный комплекс обработки потоков текстовых сообщений с возможностью адаптации к различным предметным областям.

Результаты диссертационной работы могут найти применение в задачах принятия решений на основе анализа потоков сообщений. Разработанный программный комплекс может быть непосредственно применён для анализа потоков новостных сообщений, заявок на модификацию программной продукции, обращений граждан в государственные учреждения, контент-мониторинга социальных медиа-ресурсов.

Библиография

1. Лингвистические вопросы алгоритмической обработки сообщений / Под ред. Котова Р. Г., Курбакова К. И. М.: Изд-во “Наука”, Институт языкознания (Академия наук СССР), 1983. 254 с.
2. Hu X., Liu H. Text Analytics in Social Media // Mining Text Data / Ed. by Aggarwal C. C., Zhai C. Springer US, 2012. P. 385–414.
3. Kontostathis A. et al. A Survey of Emerging Trend Detection in Textual Data Mining // Survey of Text Mining I: Clustering, Classification, and Retrieval. 2004th ed. Springer, 2003. P. 185–224.
4. Шемякин Ю. И. Начала компьютерной лингвистики. М.: Изд-во МГОУ, Росвузнаука, 1992. 81 с.
5. Dale R. Classical Approaches to Natural Language Processing // Handbook of Natural Language Processing, Second Edition. 2nd ed. / Ed. by Indurkhya N., Damerau F. J. Chapman and Hall/CRC, 2010. P. 3–7.
6. Encyclopedia of Artificial Intelligence / Ed. by Shapiro S.C. Wiley, 1987. 1219 p.
7. Хомский Н. Синтаксические структуры // Новое в лингвистике / Под ред. Звегинцева В. А., Успенского В. А.; перев. Бабицкий К. И. М.: Изд-во иностранной литературы, 1962. Т. 2. С. 412–527.
8. Бейлин Д. Краткая история генеративной грамматики // Современная американская лингвистика. Фундаментальные направления. 2-е изд. / Под ред. Кибрика А., Кобозевой И., Секериной И. М.: Едиториал УРСС, 2002. С. 13–57.
9. Jackson D. P., Moulinier M. I. Natural Language Processing for Online Applications: Text retrieval, extraction and categorization. John Benjamins Publishing Company, 2007. 247 p.
10. Survey of Text Mining I: Clustering, Classification, and Retrieval. 2004th ed. / Ed. by Berry M. W. Springer, 2003. 261 p.
11. Feldman R., Sanger J. The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press, 2006. 422 p.
12. Deriviere J., Hamon T., Nazarenko A. A Scalable and Distributed NLP Architecture for Web Document Annotation // Advances in Natural Language Processing / Ed. by Salakoski T. et al. Springer Berlin Heidelberg, 2006. P. 56–67.
13. Воронцов К. В. Анализ текстов и вероятностные тематические модели [Электронный ресурс]: конспект лекций ВМиК МГУ. 2013. URL: http://www.machinelearning.ru/wiki/images/2/22/Voron-2013-ptm.pdf (дата доступа: 11.10.2013)
14. Bishop C. M. Pattern Recognition and Machine Learning. Springer, 2007. 738 p.
15. Murphy K. P. Machine Learning: A Probabilistic Perspective. The MIT Press, 2012. 1104 p.
16. Марманис Х., Бабенко Д. Алгоритмы интеллектуального Интернета. Передовые методики сбора, анализа и обработки данных / М.: Символ-Плюс, 2011. 480 с.
17. Holte R. C. Very Simple Classification Rules Perform Well on Most Commonly Used Datasets // Machine Learning. 1993. Vol. 11, № 1. P. 63–90.
18. Aggarwal C. C. Mining Text Streams // Mining Text Data / Ed. by Aggarwal C. C., Zhai C. Springer US, 2012. P. 297–321.
19. Ankeny J. Forecast: OTT messaging traffic will double SMS volume by year’s end [Electronic resource] // FierceMobileIT. URL: http://www.fiercemobileit.com/story/forecast-ott-messaging-traffic-will-double-sms-volume-years-end/2013-04-29 (accessed: 25.10.2013).
20. Тим О’Рейли. Что такое Веб 2.0 [Электронный ресурс] // Компьютерра. 2005. URL: http://www.computerra.ru/think/234100/ (дата доступа: 12.09.2013).

21. Gaber M. M., Zaslavsky A., Krishnaswamy S. A Survey of Classification Methods in Data Streams // Data Streams / Ed. by Aggarwal C. C. Springer US, 2007. P. 39–59.
22. Падучева Е. В., Арутюнова Н. Д. Истоки, проблемы и категории прагматики // Лингвистическая прагматика. М.: Прогресс, 1985. Т. 16. С. 3–42.
23. Клышинский Э. С. Начальные этапы анализа текста // Автоматическая обработка текстов на естественном языке и компьютерная лингвистика. М.: МИЭМ (Московский государственный институт электроники и математики), 2011. С. 106–140.
24. Ljunglof P., Wiren M. Syntactic Parsing // Handbook of Natural Language Processing, Second Edition. 2nd ed. / Ed. by Indurkhya N., Damerau F.J. Chapman and Hall/CRC, 2010. P. 59–92.
25. Леонтьева Н. Н. Автоматическое понимание текстов. Системы, модели, ресурсы. Академия, 2006. 304 с.
26. Mcroy S. W., Ali S. S., Haller S. Mixed Depth Representations for Dialog Processing // Proceedings of the Cognitive Science Society. 1998. Vol. 98. P. 687–692.
27. Шенк Р. Обработка концептуальной информации. М.: Энергия, 1980. 360 с.
28. Касавин И. Т. Дискурс-анализ // Энциклопедия эпистемологии и философии науки / Под ред. Касавина И. Т. и др. Канон+РООИ “Реабилитация,” 2009. С. 199–202.
29. Пескова О. В. Алгоритмы классификации полнотекстовых документов // Автоматическая обработка текстов на естественном языке и компьютерная лингвистика. М.: МИЭМ (Московский государственный институт электроники и математики), 2011. С. 170–212.
30. Паклин Н. Б., Орешков В. И. Бизнес-аналитика. От данных к знаниям. 2-е изд. Питер, 2013. 704 с.
31. Daume H. C. Practical Structured Learning Techniques for Natural Language Processing. University of Southern California, 2006. 134 p.
32. Частиков А. П., Белов Д. Л., Гаврилова Т. А. Разработка экспертных систем. Среда CLIPS. БХВ-Петербург, 2003. 608 с.
33. Quinlan J. R. Decision trees and decision-making // IEEE Transactions on Systems, Man and Cybernetics. 1990. Vol. 20, № 2. P. 339–346.
34. Kohavi R. The power of decision tables // Machine Learning: ECML-95 / Ed. by Lavrac N., Wrobel S. Springer Berlin Heidelberg, 1995. P. 174–189.
35. Murthy S. K. Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey // Data Mining and Knowledge Discovery. 1998. Vol. 2, № 4. P. 345–389.
36. Барсегян А. А. и др. Технологии анализа данных. Data Mining, Visual Mining, Text Mining, OLAP. 2-е изд. БХВ-Петербург, 2007. 384 с.
37. Маннинг К. Д., Рагхаван П., Шютце Х. Введение в информационный поиск / Перев. Клюшин Д. Вильямс, 2011. 528 с.
38. Глазкова В. В. Исследование и разработка методов построения программных средств классификации многотемных гипертекстовых документов: Дисс. на соиск. уч. ст. канд. техн. наук. М.: МГУ, 2008. 103 с.
39. Christiansen M. H., Chater N. Connectionist Natural Language Processing: The State of the Art // Cognitive Science. 1999. Vol. 23, № 4. P. 417–437.
40. Крайнов А. Ю. Об одном варианте применения искусственных нейронных сетей для обработки текстов на естественном языке // Учёные записки Ульяновского государственного университета. 2011. С. 225–230.
41. Smolensky P. Connectionist Approaches to Language // The MIT encyclopedia of the cognitive sciences / Ed. by Wilson R. A., Keil F. C. Cambridge: MIT Press, 1999. P. 188–190.

42. Hofmann T. Probabilistic latent semantic indexing // Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. 1999. P. 50–57.
43. Blei D. M., Ng A. Y., Jordan M. I. Latent dirichlet allocation // J. Mach. Learn. Res. 2003. Vol. 3. P. 993–1022.
44. Daud A. et al. Knowledge discovery through directed probabilistic topic models: a survey // Frontiers of Computer Science in China. 2010. Vol. 4, № 2. P. 280–301.
45. Рассел С., Норвиг П. Искусственный интеллект. Современный подход. 2-е изд. / Перев. Птицын К. Вильямс, 2006. 1408 с.
46. Forgy C. L. On the Efficient Implementation of Production Systems. Pittsburgh, PA, USA: Carnegie Mellon University, 1979. 210 p.
47. Люгер Д. Ф. Искусственный интеллект. Стратегии и методы решения сложных проблем / Перев. Протасова К. Вильямс, 2005. 864 с.
48. Fisher D., Pazzani M. Computational models of concept learning // Concept formation knowledge and experience in unsupervised learning. Morgan Kaufmann Publishers Inc., 1991. P. 3–43.
49. Fisher D. H. Knowledge acquisition via incremental conceptual clustering // Machine Learning. 1987. Vol. 2, № 2. P. 139–172.
50. Воронцов К. В. Обзор современных исследований по проблеме качества обучения алгоритмов // Таврический вестник информатики и математики. 2004. № 1. С. 5.
51. Cunningham H. et al. Software infrastructure for natural language processing // Proceedings of the fifth conference on Applied natural language processing. Stroudsburg, PA, USA: Association for Computational Linguistics, 1997. P. 237–244.
52. Wilcock G. Introduction to Linguistic Annotation and Text Analytics. 1st ed. Morgan & Claypool Publishers, 2009. 160 p.
53. Носков А. А. Инструментальные системы разработки приложений по автоматической обработке текстов на естественном языке // Автоматическая обработка текстов на естественном языке и компьютерная лингвистика. М.: МИЭМ (Московский государственный институт электроники и математики), 2011. С. 141–169.
54. DeRose S. Markup Overlap: A Review and a Horse. Montréal, Québec, 2004.
55. Соловьев В. Д. и др. Онтологии и тезаурусы. Казань, Москва, 2006. 157 с.
56. Крижановский А. А. Математическое и программное обеспечение построения списков семантически близких слов на основе рейтинга вики-текстов: Дисс. на соиск. уч. ст. канд. техн. наук. СПб.: Санкт-Петербургский институт информатики и автоматизации РАН, 2008. 188 с.
57. Estival D., Nowak C., Zschorn A. Towards Ontology-based Natural Language Processing // Proceeedings of the Workshop on NLP and XML (NLPXML-2004): RDF/RDFS and OWL in Language Technology. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004. P. 59–66.
58. Кулинич А. А. Концептуальные «каркасы» онтологий в поддержке принятия решений в условиях неопределенности // Доклады 13-й национальной конференции по искусственному интеллекту с международным участием КИИ-2012. Белгород, 2012.
59. Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем. Питер, 2000. 384 с.
60. Ханк Д. Э., Уичерн Д. У., Райтс А. Д. Бизнес-прогнозирование. Вильямс, 2003. 656 с.
61. Э. Е. Тихонов. Методы прогнозирования в условиях рынка. Невинномысск: Северо-Кавказский государственный технический университет, 2006. 221 с.

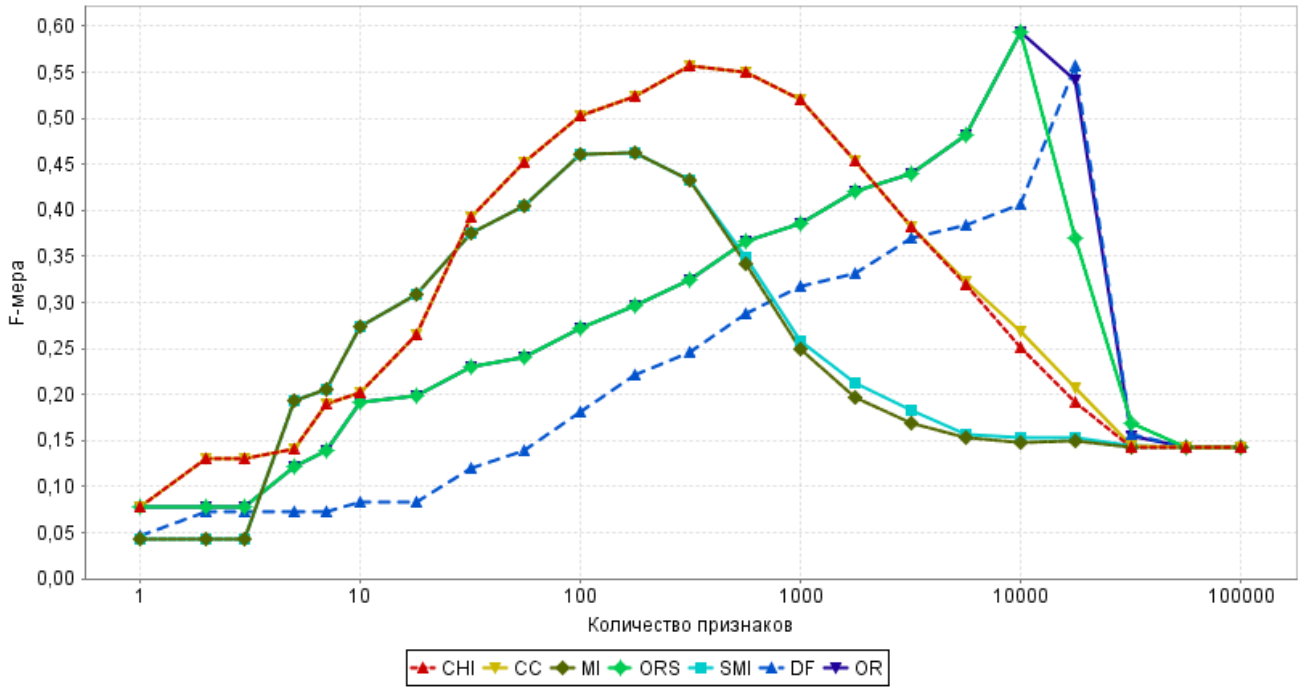
62. Аналитические технологии для прогнозирования и анализа данных. Методы прогнозирования [Электронный ресурс] // NeuroProject. 2005. URL: http://www.neuroproject.ru/forecasting_tutorial.php (дата доступа: 04.04.2013).
63. Антипов О. И., Неганов В. А. Анализ и прогнозирование поведения временных рядов: бифуркации, катастрофы, синергетика, фракталы и нейронные сети. М.: Радиотехника, 2011. 350 с.
64. Ярушкина Н. Г., Афанасьева Т. В., Перфильева И. Г. Интеллектуальный анализ временных рядов. Ульяновск: УлГТУ, 2010. 320 с.
65. Song Q., Chissom B.S. Fuzzy time series and its models // Fuzzy Sets and Systems. 1993. Vol. 54, № 3. P. 269–277.
66. Cugola G., Margara A. Processing flows of information: From data stream to complex event processing // ACM Comput. Surv. 2012. Vol. 44, № 3. P. 15:1–15:62.
67. Eckert M. et al. A CEP Babelfish: Languages for Complex Event Processing and Querying Surveyed // Reasoning in Event-Based Distributed Systems / Ed. by Poulovassilis A., Xhafa F. Springer, 2011.
68. Luckham D. The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. 1st ed. Addison-Wesley Professional, 2002. 400 p.
69. Muller A. Event Correlation Engine: Master's Thesis. Swiss Federal Institute of Technology Zurich, 2009. 175 p.
70. Dong G., Pei J. Sequence Data Mining. Springer Science+Business Media, 2007. 160 p.
71. Agrawal R., Srikant R. Fast Algorithms for Mining Association Rules in Large Databases // Proceedings of the 20th International Conference on Very Large Data Bases. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994. P. 487–499.
72. Steinder M. łgorzata, Sethi A.S. A survey of fault localization techniques in computer networks // Science of Computer Programming. 2004. Vol. 53, № 2. P. 165–194.
73. Hanemann A., Marcu P. Algorithm design and application of service-oriented event correlation // 3rd IEEE/IFIP International Workshop on Business-driven IT Management, 2008. BDIM 2008. 2008. P. 61–70.
74. Yuniarto H. A. The Shortcomings of Existing Root Cause Analysis Tools // Proceedings of the World Congress on Engineering 2012 / Ed. by S. I. Ao et al. UK: London: Newswood Limited, 2012. Vol. 3.
75. Bolshakov I. A., Gelbukh A. Computational Linguistics: Models, Resources, Applications. Mexico: Instituto Politécnico Nacional, 2004. 186 p.
76. Сокирко А. В. Семантические словари в автоматической обработке текста. По материалам системы ДИАЛИНГ: Дисс. на соиск. уч. ст. канд. техн. наук. М.: РГГУ, 2001. 120 с.
77. Bifet A. Adaptive Stream Mining: Pattern Learning and Mining from Evolving Data Streams. IOS Press, 2010. Vol. 207. 224 p.
78. Bontcheva K. et al. TwitIE: An Open-Source Information Extraction Pipeline for Microblog Text // Proceedings of the International Conference on Recent Advances in Natural Language Processing. Association for Computational Linguistics, 2013.
79. Хорошевский В. Ф. Выявление новых технологических трендов: проблемы и перспективы // Доклады XIII национальной конференции по искусственному интеллекту с международным участием КИИ-2012. Белгород: Российская ассоциация искусственного интеллекта, 2012. Т. 1. С. 252–258.
80. Воронцов К. В. Математические методы обучения по прецедентам (теория обучения машин). 2013. 141 с.
81. Palmer D. D. Text Preprocessing // Handbook of Natural Language Processing, Second Edition. 2nd ed. / Ed. by Indurkhya N., Damerau F.J. Chapman and Hall/CRC, 2010. P. 9–30.

82. Dolamic L., Savoy J. Stemming Approaches for East European Languages // Advances in Multilingual and Multimodal Information Retrieval / Ed. by Peters C. et al. Springer Berlin Heidelberg, 2008. P. 37–44.
83. Porter M. F. An algorithm for suffix stripping // Program: Electronic Library and Information Systems. 1980. Vol. 14, № 3. P. 130–137.
84. Загоровская О. В. Становление диалектной компьютерной лексикографии в отечественной лингвистике // Вестник Воронежского государственного университета. 2012. № 1. С. 10–16.
85. Марчук Ю. Н. Компьютерная лингвистика. АСТ, Восток-Запад, 2007. 320 с.
86. Daciuk J. et al. Incremental Construction of Minimal Acyclic Finite-state Automata // Comput. Linguist. 2000. Vol. 26, № 1. P. 3–16.
87. Кузнецов А. Гибридная реализация русской морфологии [Электронный ресурс] // Habrahabr.ru. 2009. URL: http://habrahabr.ru/post/66560/ (дата доступа: 30.10.2013).
88. McCallum A., Nigam K. A comparison of event models for Naive Bayes text classification // AAAI Workshop on Learning for Text Categorization. 1998.
89. Allen J. Estimating Probability Distributions. CSC 248/448: Speech Recognition and Statistical Language Models [Electronic resource] // The University of Rochester Department of Computer Science. URL: http://www.cs.rochester.edu/u/james/CSC248/Lec3.pdf (accessed: 10.10.2013).
90. Tsoumakas G., Katakis I. Multi-label classification: An overview // Int J Data Warehousing and Mining. 2007. Vol. 2007. P. 1–13.
91. McCallum A. K. Multi-label text classification with a mixture model trained by EM // AAAI 99 Workshop on Text Learning. 1999.
92. Read J. Scalable Multi-label Classification: Thesis. University of Waikato, 2010.
93. Zhang M.-L., Zhang K. Multi-label learning by exploiting label dependency // Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. New York, NY, USA: ACM, 2010. P. 999–1008.
94. Крайнов А. Ю. Разработка многозначного наивного байесовского классификатора на основе пороговой оценки // Научно-технический вестник Поволжья. 2013. № 5. С. 225–228.
95. Madjarov G. et al. An Extensive Experimental Comparison of Methods for Multi-label Learning // Pattern Recognition. 2012. Vol. 45, № 9. P. 3084–3104.
96. Li S. et al. A Framework of Feature Selection Methods for Text Categorization // Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009. P. 692–700.
97. Janecek A. et al. On the relationship between feature selection and classification accuracy // Journal of Machine Learning Research Workshop and Conference Proceedings 4. 2008. P. 90–105.
98. Vafaie H., Imam I. F. Feature selection methods: genetic algorithms vs. greedy-like search // In: Proceedings of the 3rd International Conference on Fuzzy and Intelligent Control Systems. USA: Louisville, 1994.
99. Jourdan L., Dhaenens C., Talbi E.-G. A genetic algorithm for feature selection in data-mining for genetics // Proceedings of the 4th Metaheuristics International ConferencePorto (MIC'2001). 2001. P. 29–34.
100. Blum A. L., Langley P. Selection of relevant features and examples in machine learning // Artificial Intelligence. 1997. Vol. 97. P. 245–271.
101. Zheng Z., Wu X., Srihari R. Feature selection for text categorization on imbalanced data // ACM SIGKDD Explorations Newsletter. 2004. Vol. 6, № 1. P. 80–89.

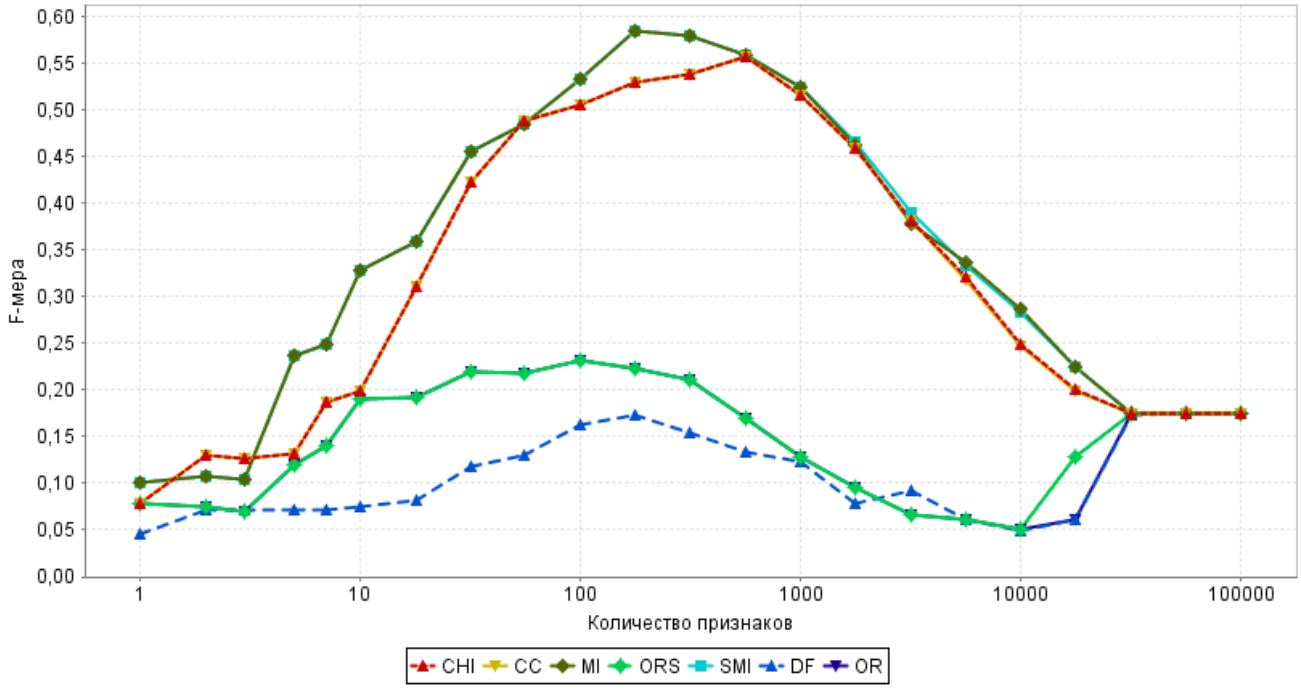
102. Никишов А. А., Сорознишвили Л. Т. Использование метода декомпозиции при анализе временных рядов для прогнозирования объемов и структуры авиаперевозок // Вестник международного славянского университета. Харьков. 2008. Т. 11. С. 37–41.
103. Семёнов Ю. А. Современные поисковые системы [Электронный ресурс] // Телекоммуникационные технологии. 2013. URL: http://book.itep.ru/4/45/retr4514.htm (дата доступа: 01.10.2013).
104. Куприянов А. А., Мельниченко А. С., Крайнов А. Ю. Подход к созданию виртуальной организации проектирования и изготовления программных изделий ИАСУ // Автоматизация процессов управления. 2009. № 3. С. 33–43.
105. Наталья Дубова. Интеграция приложений и бизнес-процессы // Открытые системы. 2009. № 10.
106. Хоп Г., Вульф Б. Шаблоны интеграции корпоративных приложений / Перев. Журавлев А., Селина Н. Вильямс, 2007. 672 с.
107. Ларман К. Применение UML и шаблонов проектирования. Вильямс, 2002. 624 с.
108. Захаров В. Г., Крайнов А. Ю., Липатова С. В., Смагин А. А. Построение системы доставки обновлений программных продуктов // Учёные записки Ульяновского государственного университета / Под ред. проф. А. А. Смагина. 2012. № 1 (4). С. 161–174.
109. Крайнов А. Ю., Смагин А. А. Автоматизация сбора и обработки протоколов в системе сопровождения программного обеспечения на основе обработки сложных событий // Автоматизация процессов управления. 2013. Т. 4 (34). [В печати]
110. Крайнов А. Ю., Смагин А. А. Разработка комплекса анализа ошибок в корпоративных информационных системах // Известия Самарского научного центра Российской академии наук. 2013. Т. 15, № 4 (3). С. 688–692.
111. Гущина Н. М. Интернет в России. Состояние, тенденции и перспективы развития. Отраслевой доклад. М.: Федеральное агентство по печати и массовым коммуникациям, 2013. 97 с.
112. Статистика Википедии — Русский раздел [Электронный ресурс] // WikiMedia.org. URL: http://stats.wikimedia.org/RU/TablesWikipediaRU.htm (дата доступа: 02.10.2013).

Приложения

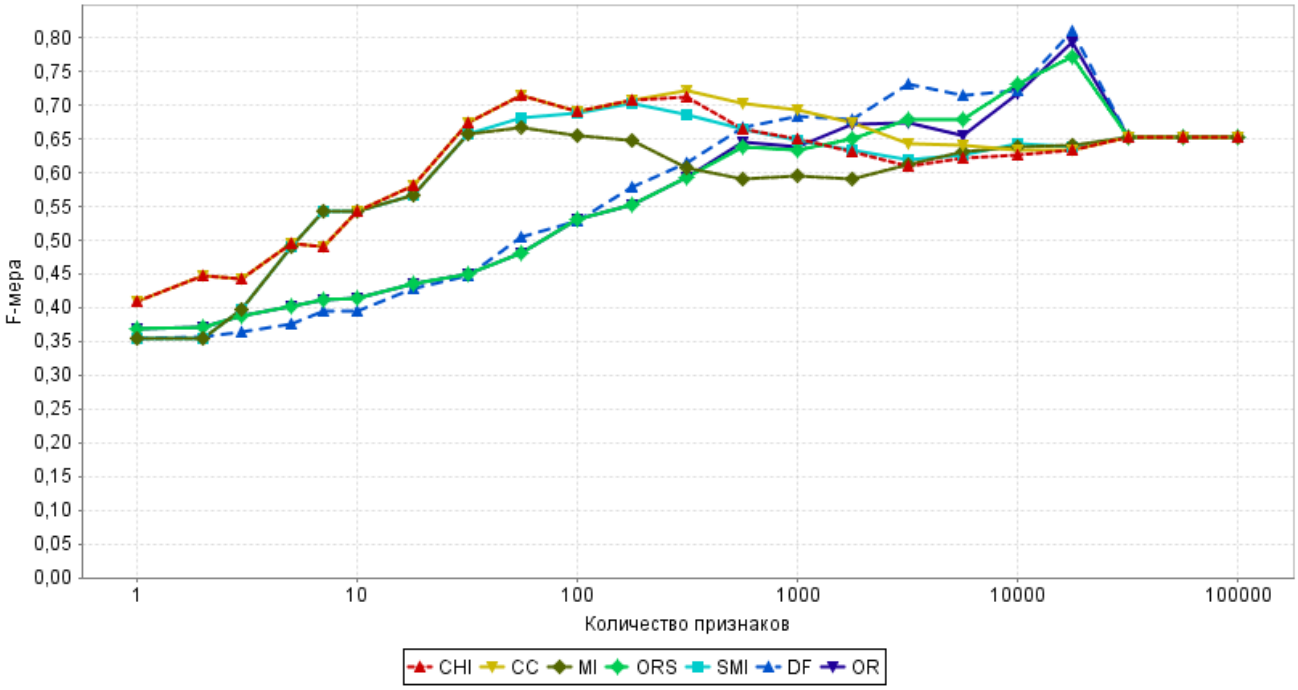
Приложение А. Эффективность классификации при применении эвристической процедуры выбора признаков



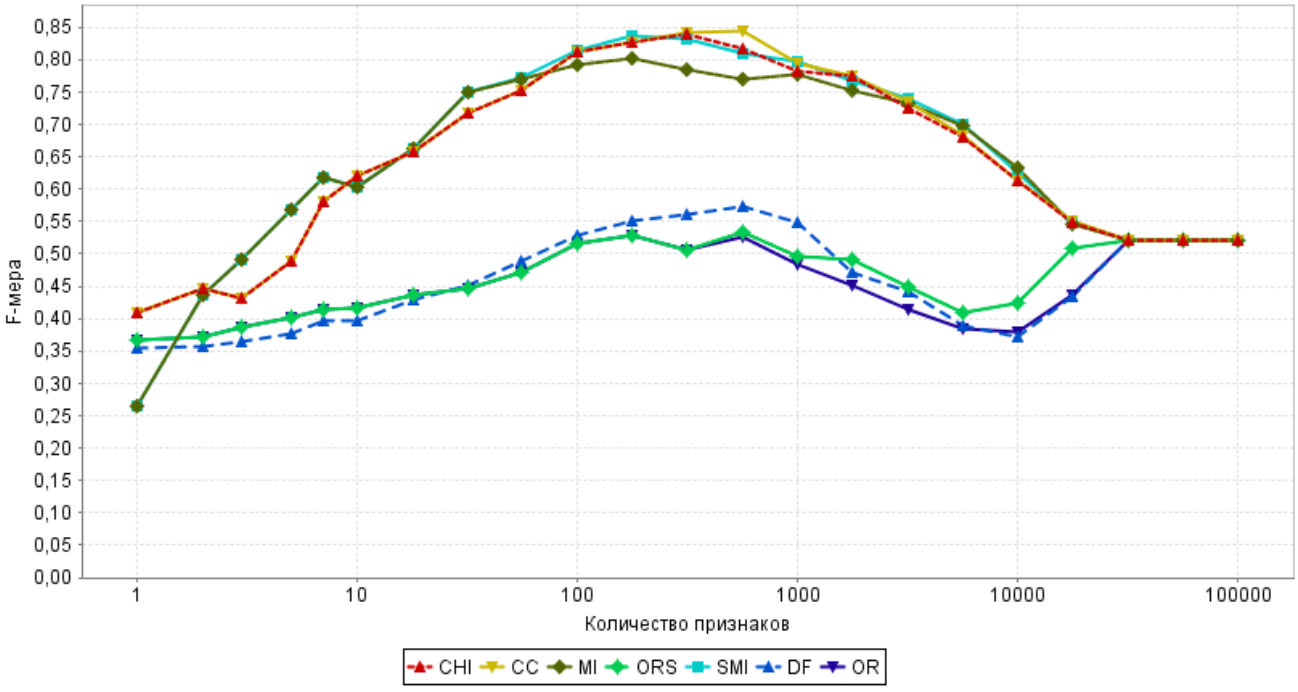
Мультиномиальный НБК, набор A33



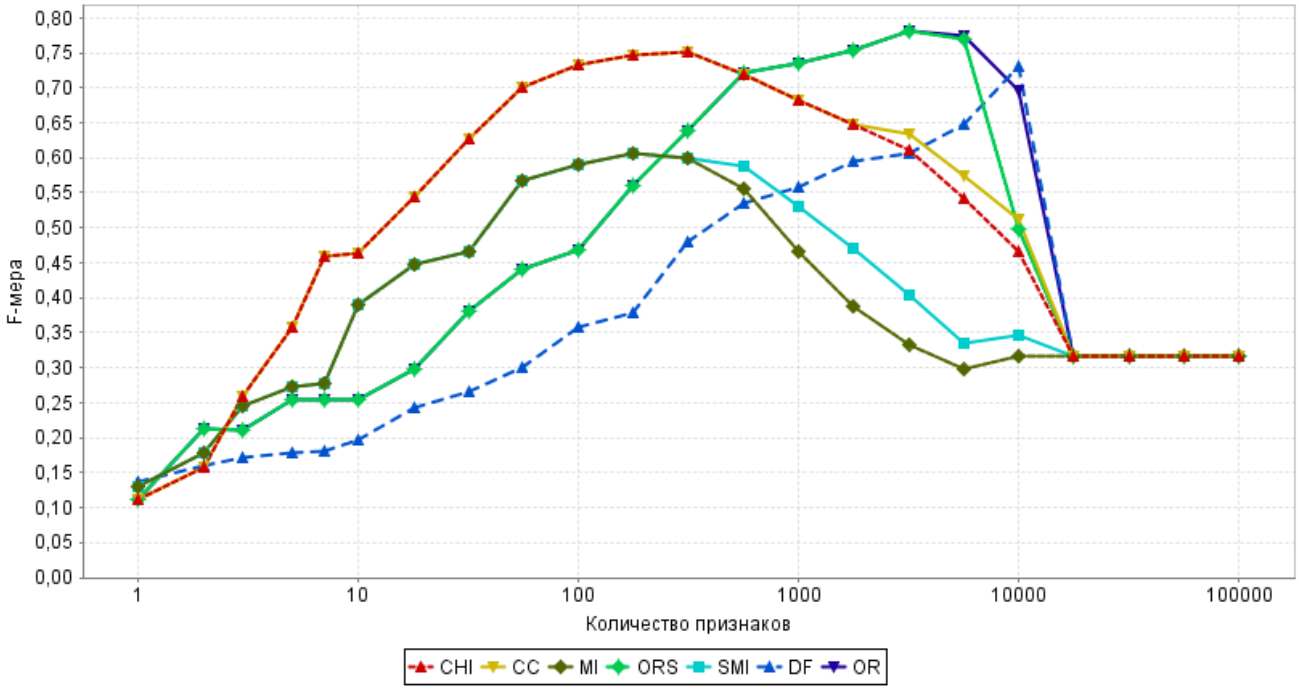
Бернуллиевский НБК, набор A33



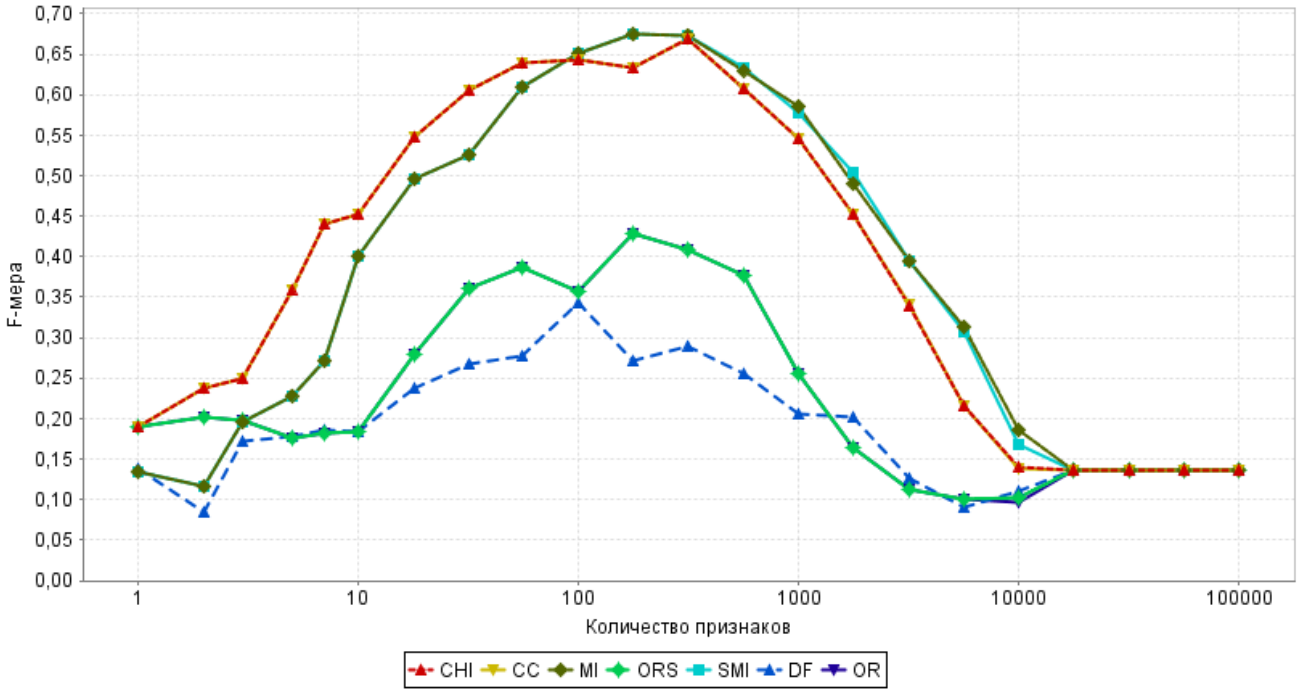
Мультиномиальный НБК, набор B9



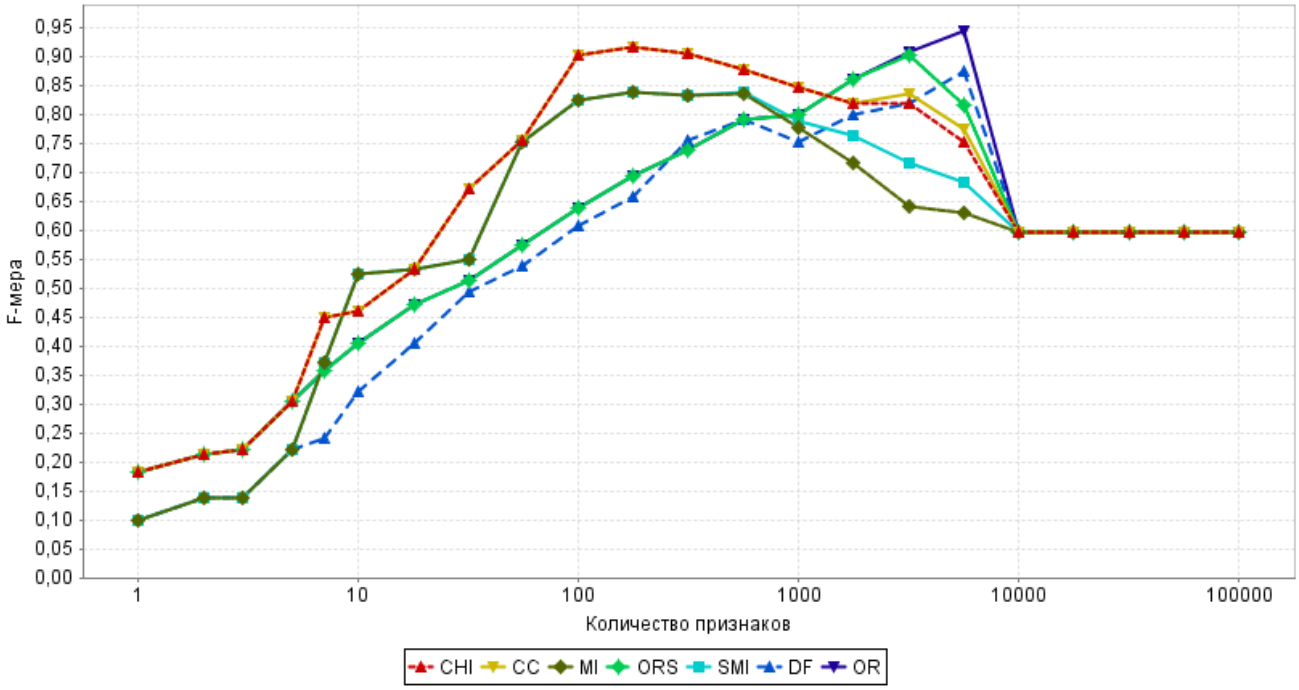
Бернуллиевский НБК, набор B9



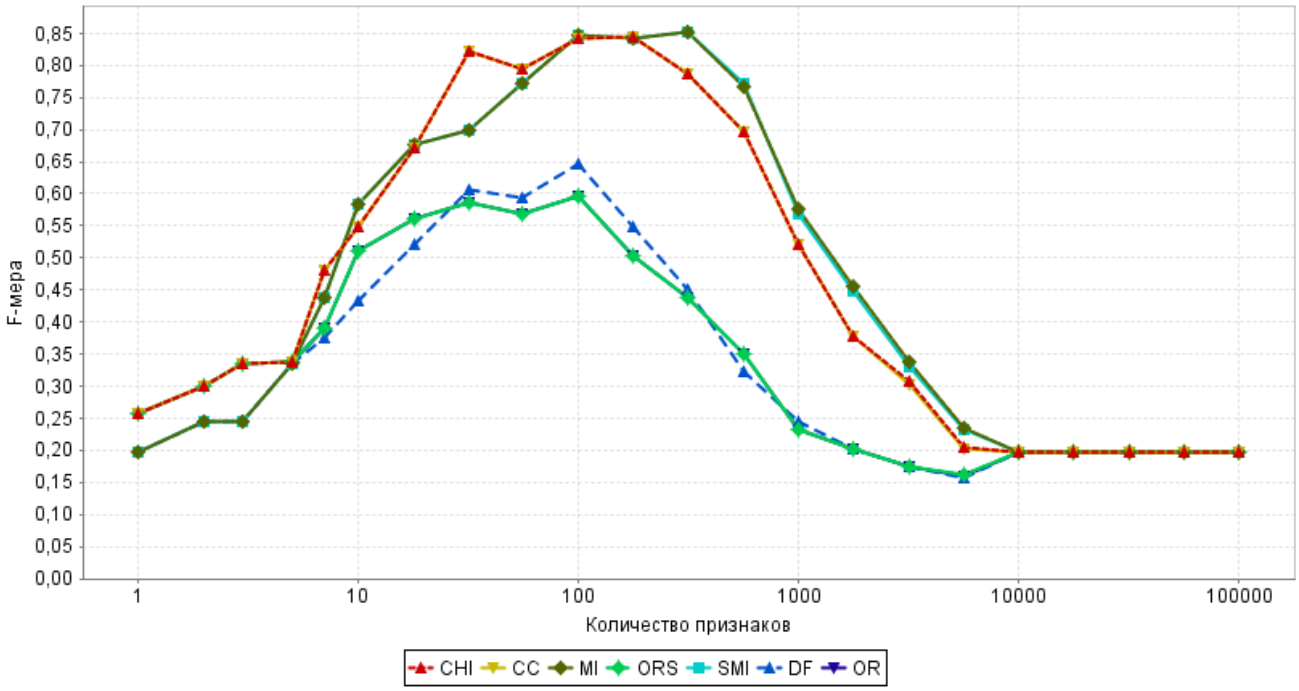
Мультиномиальный НБК, набор C16



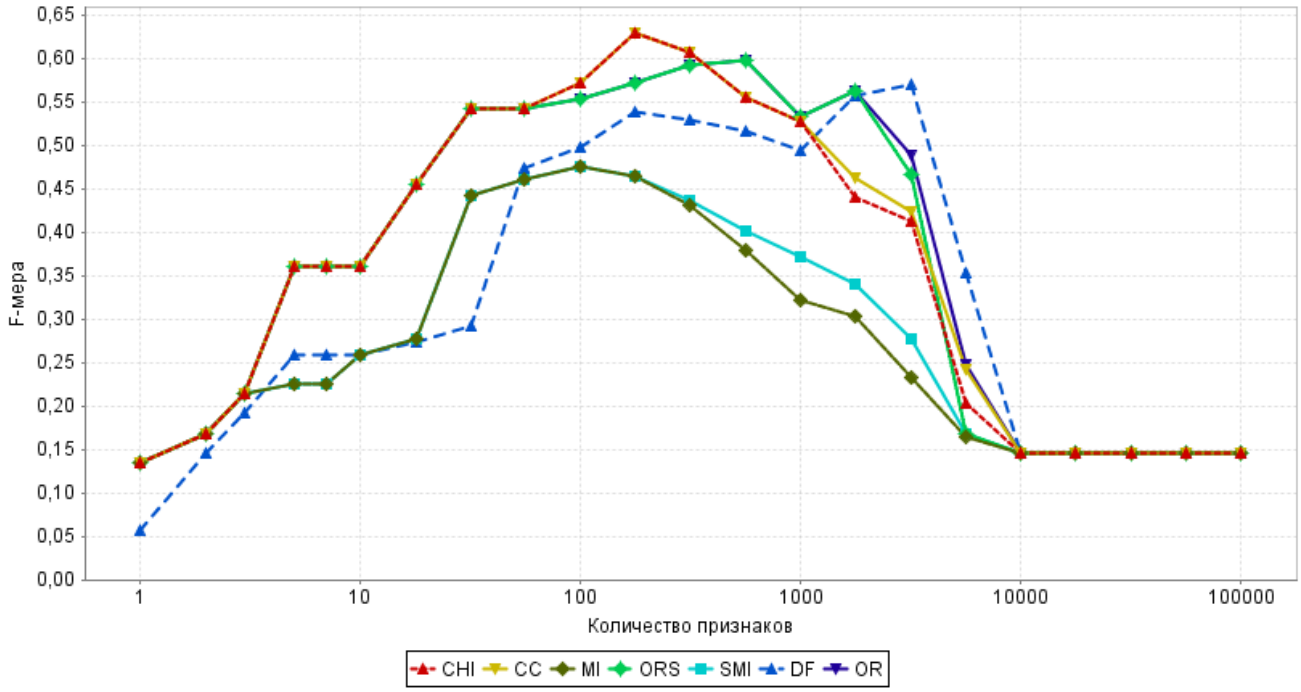
Бернуллиевский НБК, набор C16



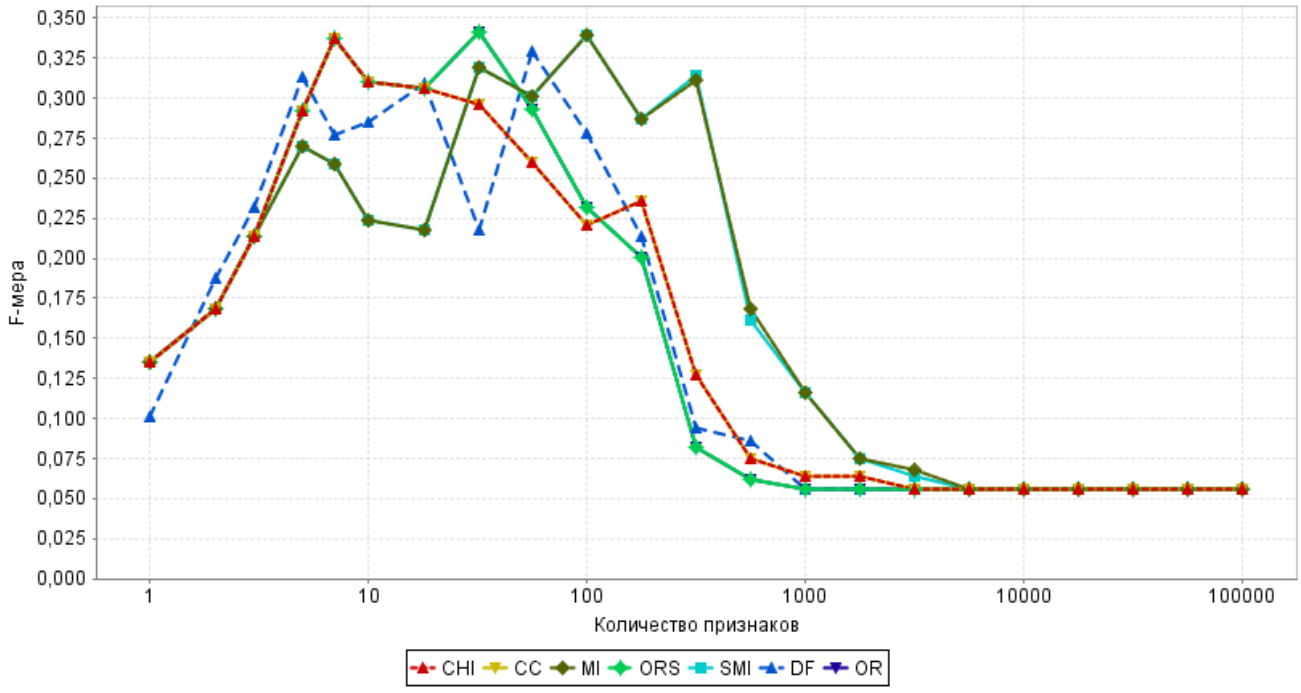
Мультиномиальный НБК, набор D11



Бернуллиевский НБК, набор D11

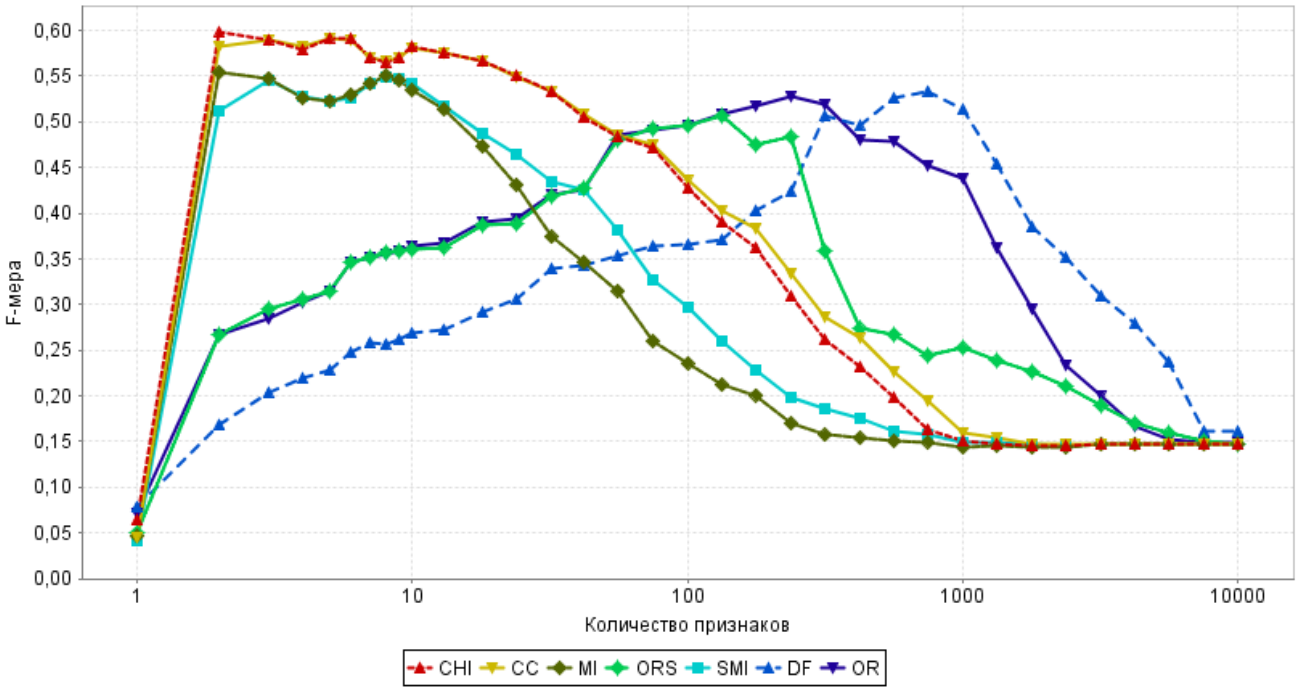


Мультиномиальный НБК, набор E14

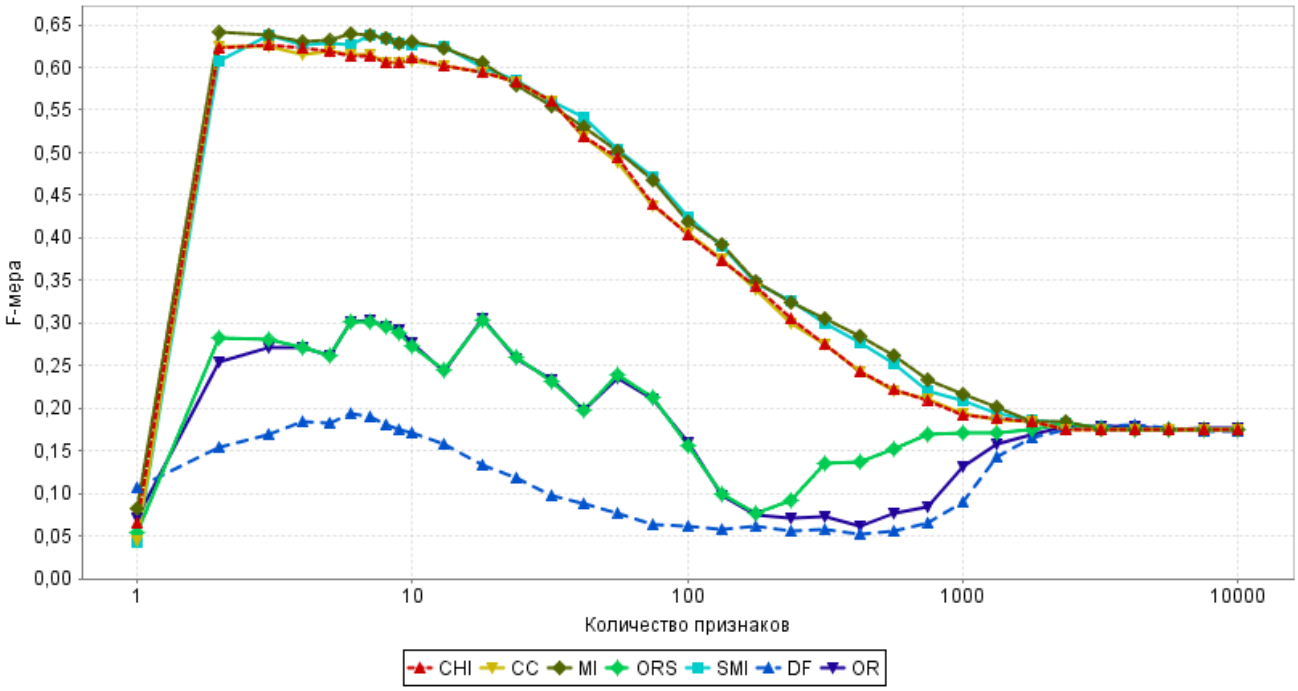


Бернуллиевский НБК, набор E14

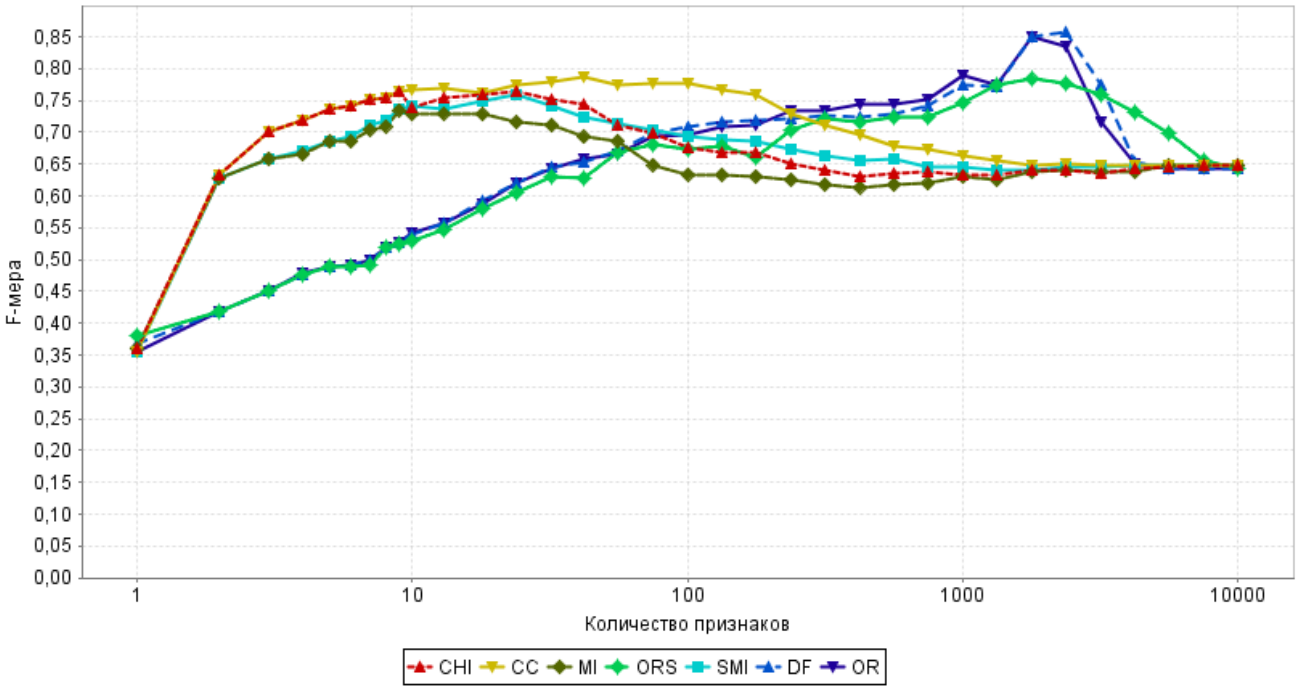
Приложение Б. Эффективность классификации при применении сбалансированной процедуры выбора признаков



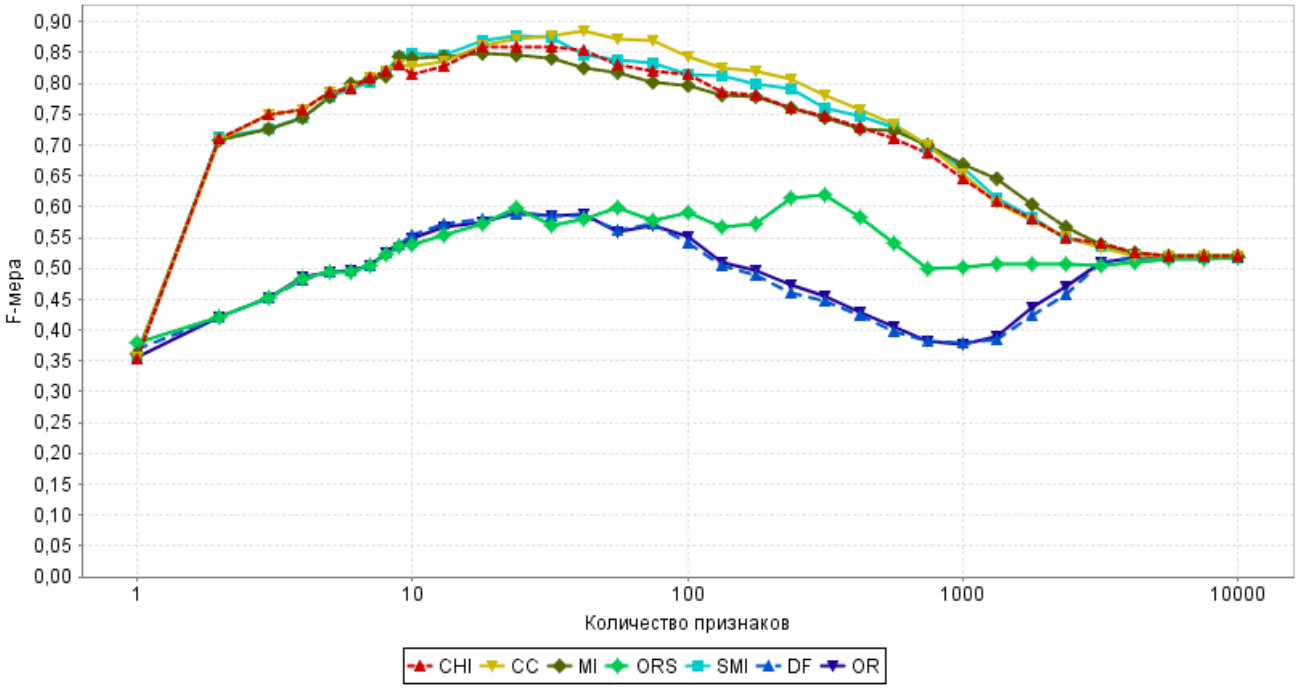
Мультиномиальный НБК, набор A33



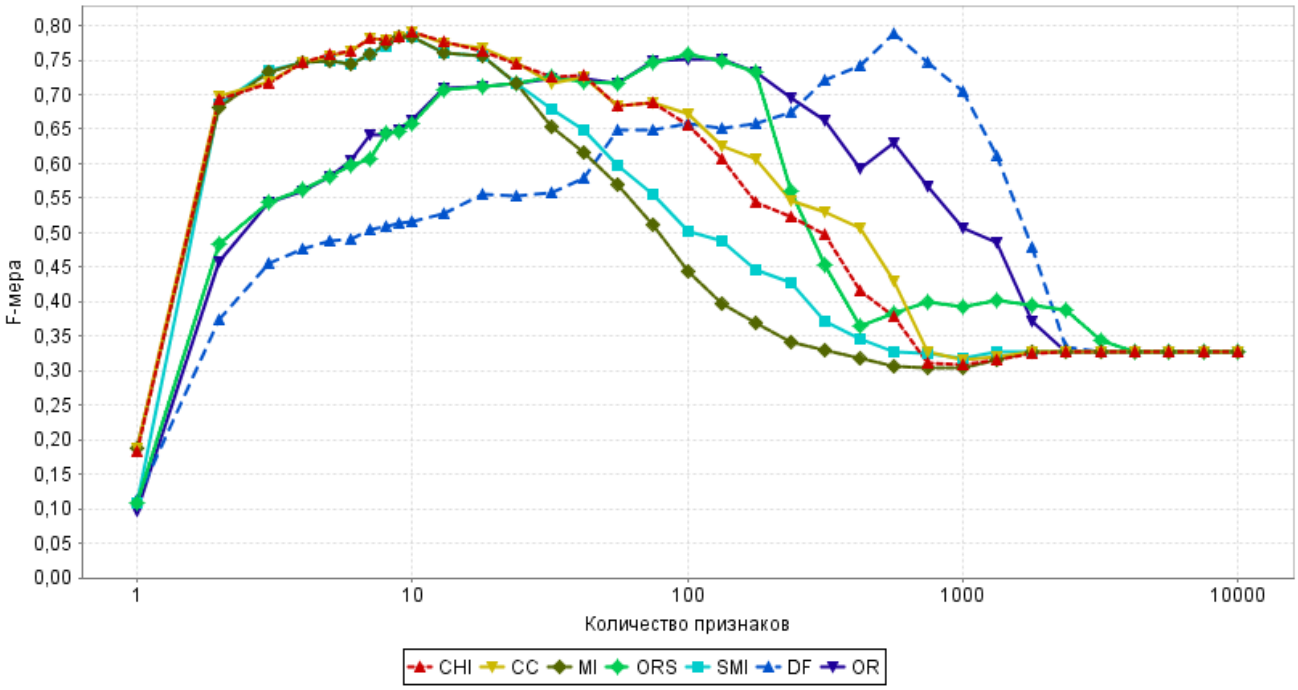
Бернуллиевский НБК, набор A33



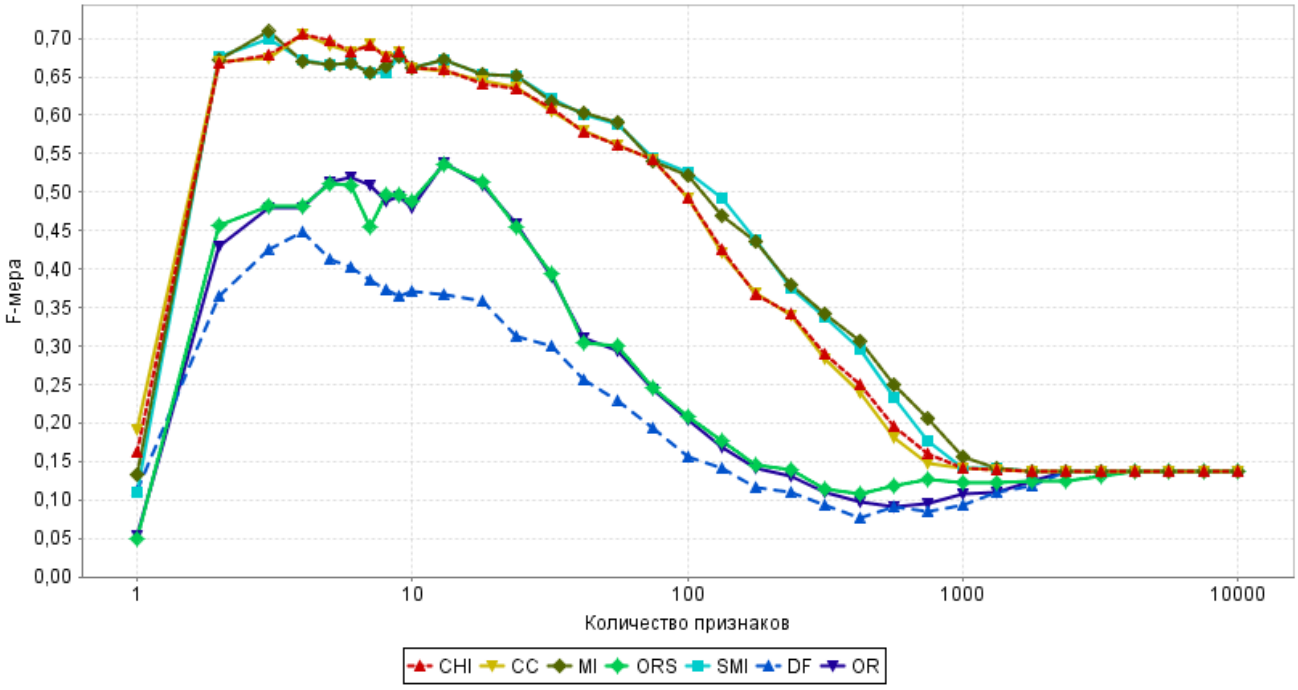
Мультиномиальный НБК, набор B9



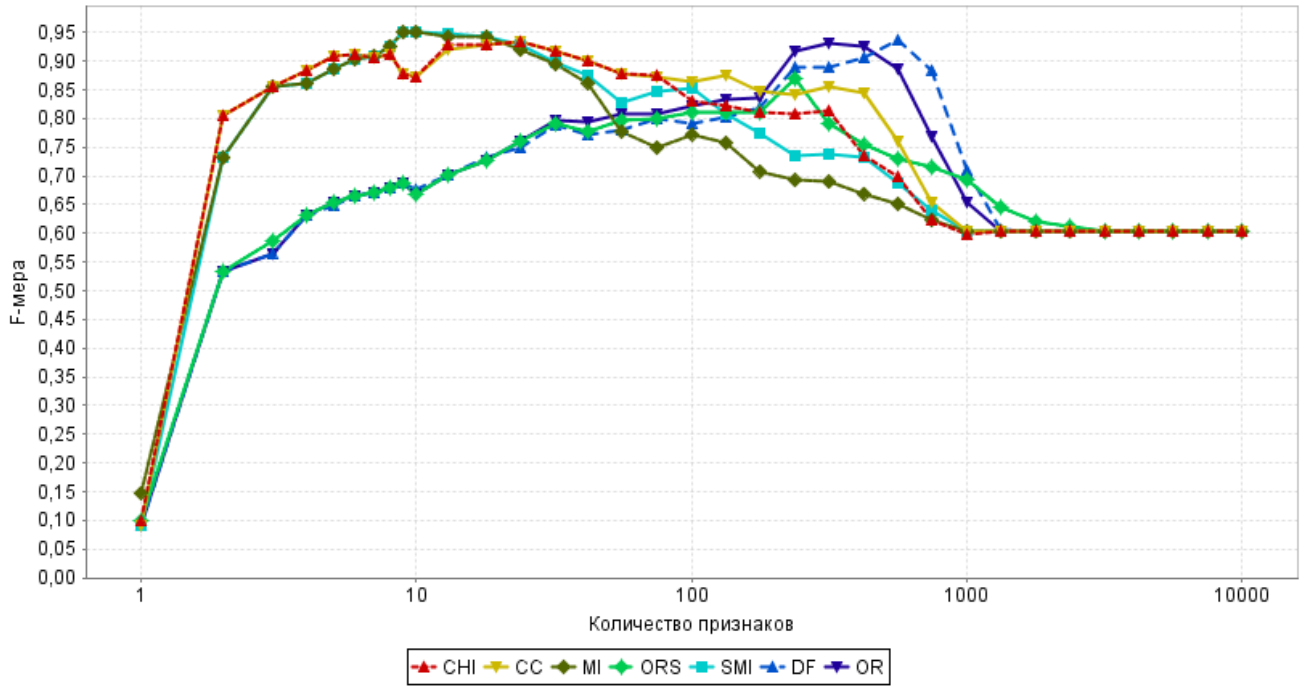
Бернуллиевский НБК, набор B9



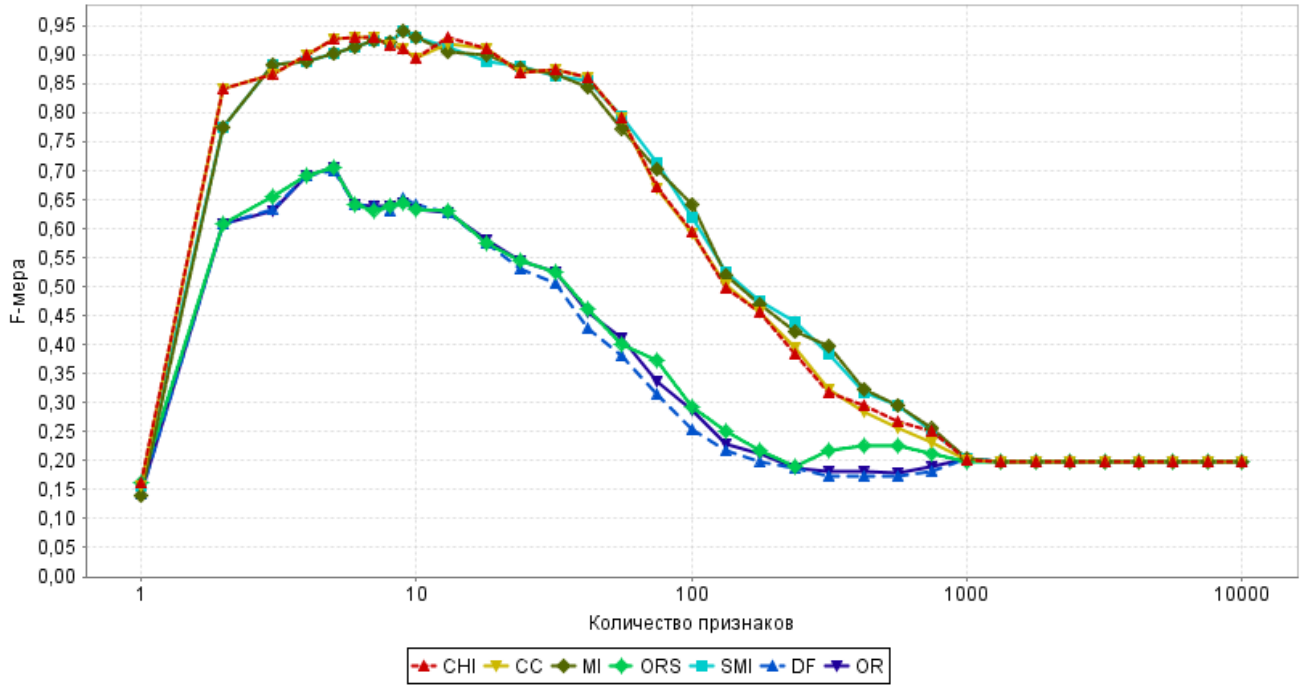
Мультиномиальный НБК, набор C16



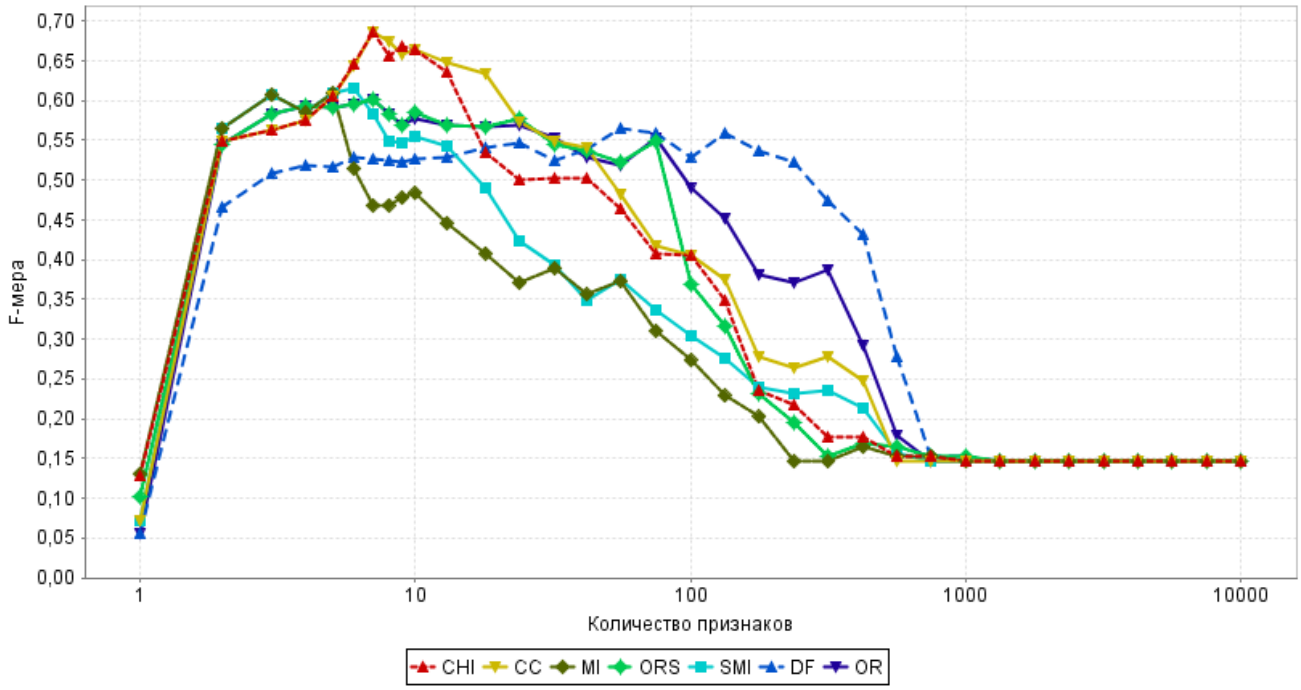
Бернуллиевский НБК, набор C16



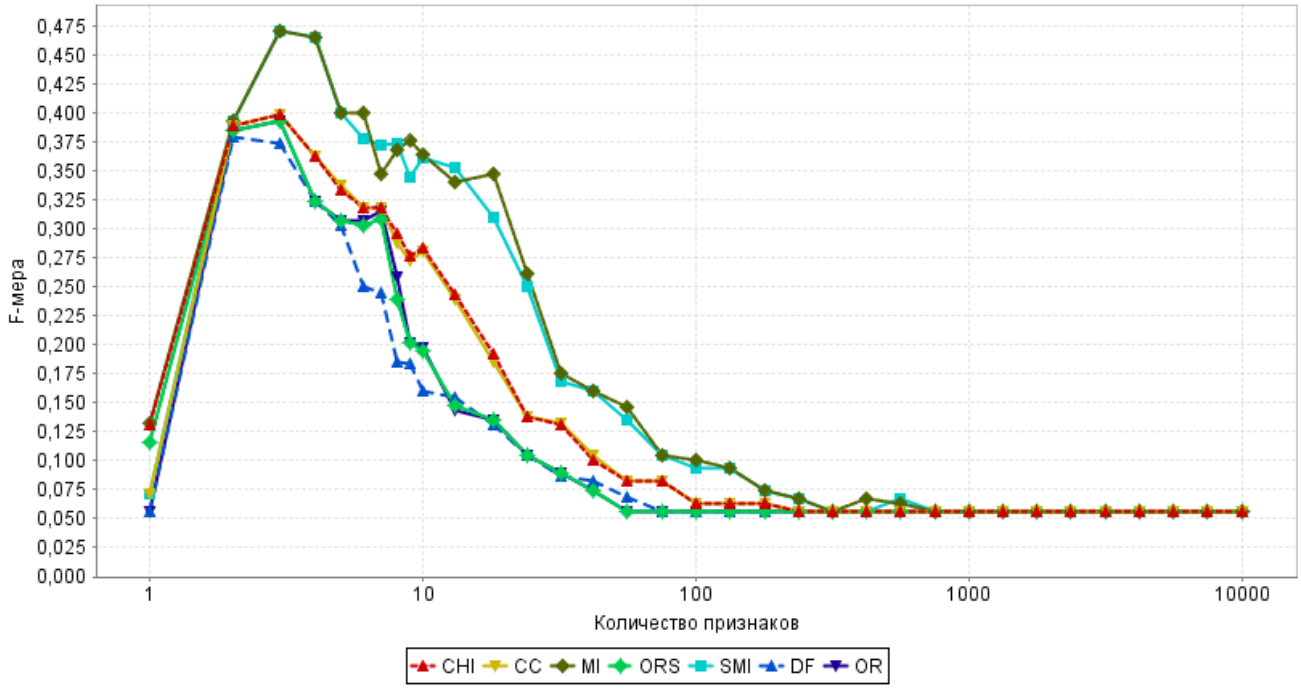
Мультиномиальный НБК, набор D11



Бернуллиевский НБК, набор D11

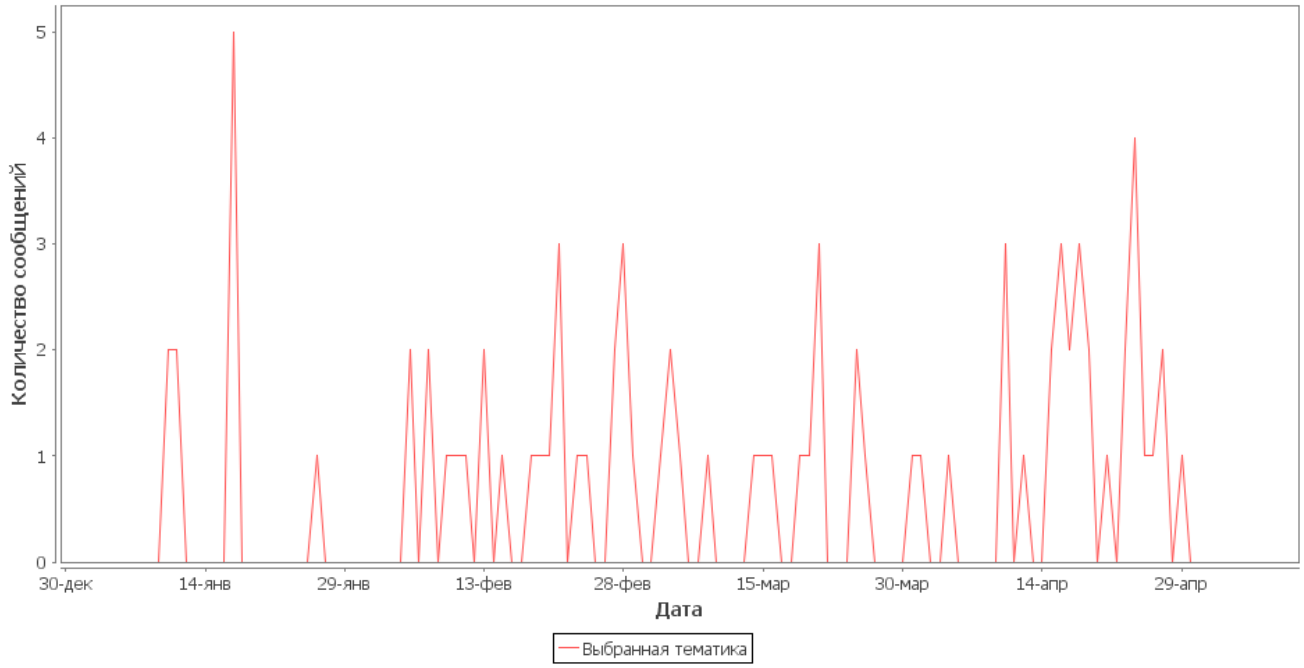


Мультиномиальный НБК, набор E14

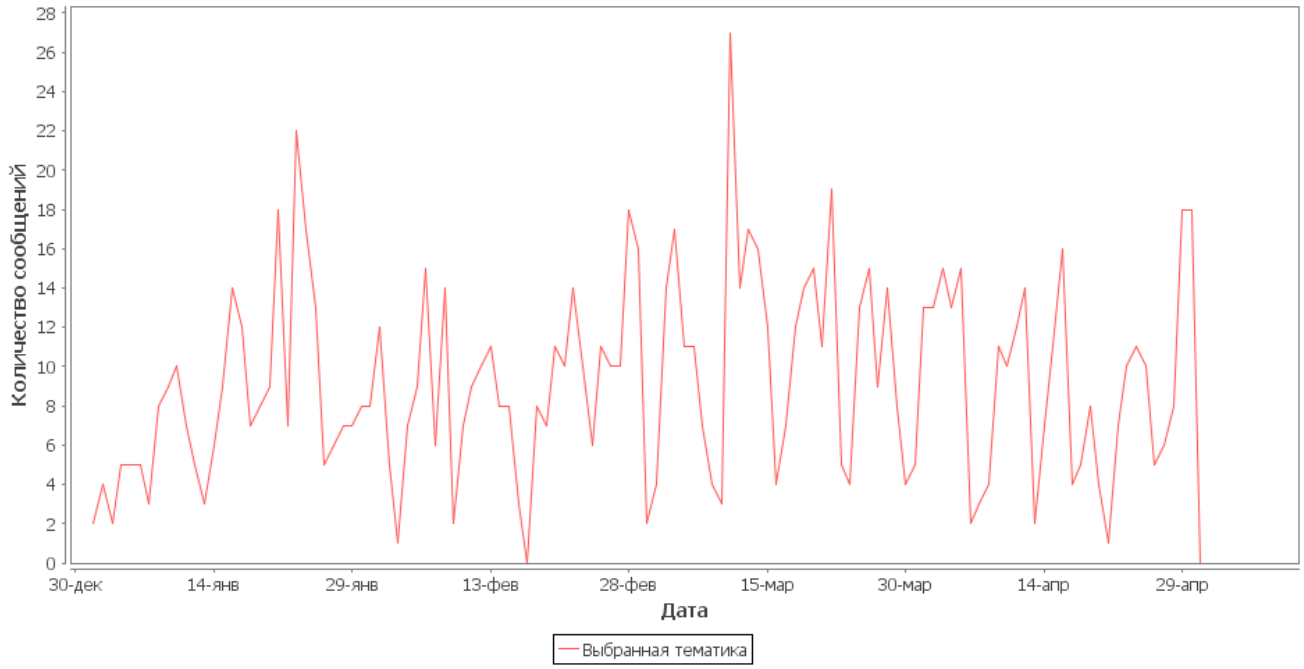


Бернуллиевский НБК, набор E14

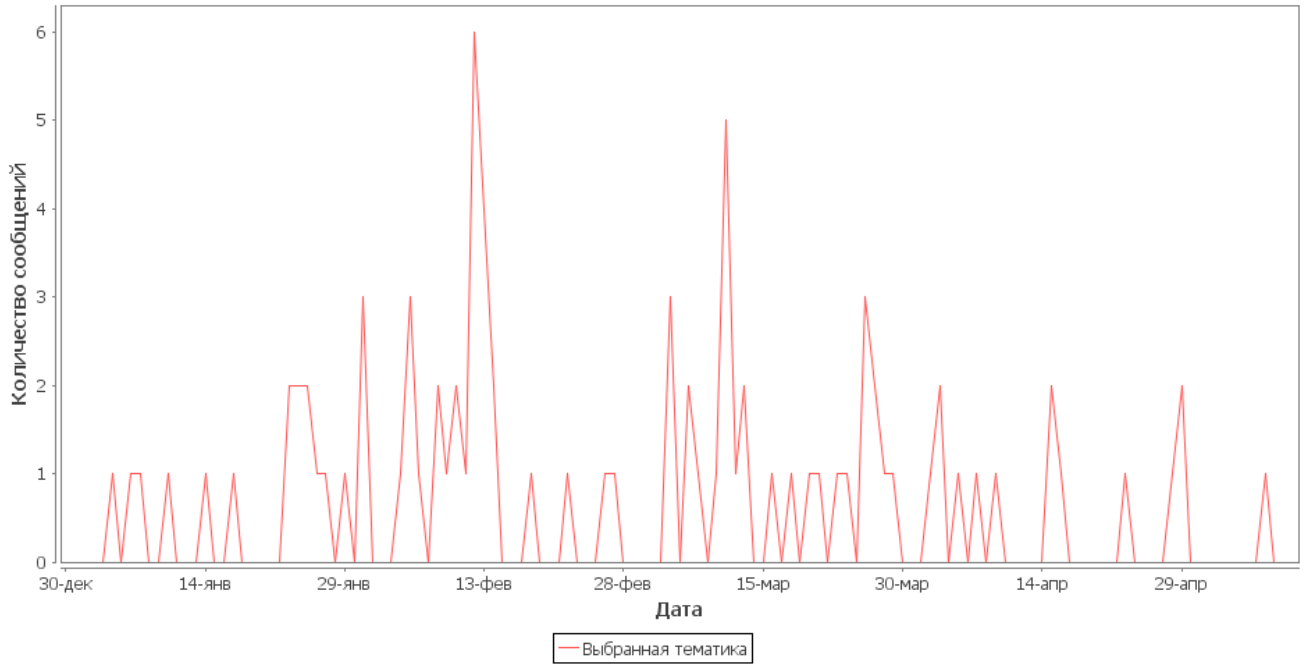
Приложение В. Временные ряды тематик новостных сообщений



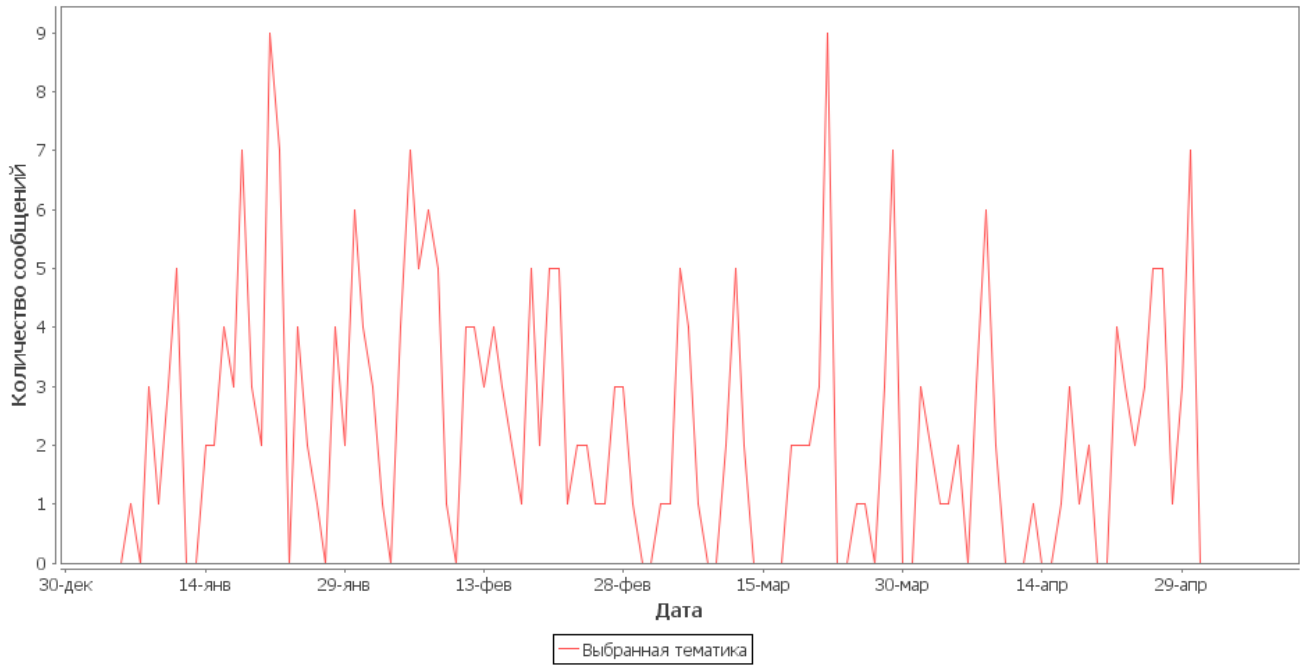
Армия



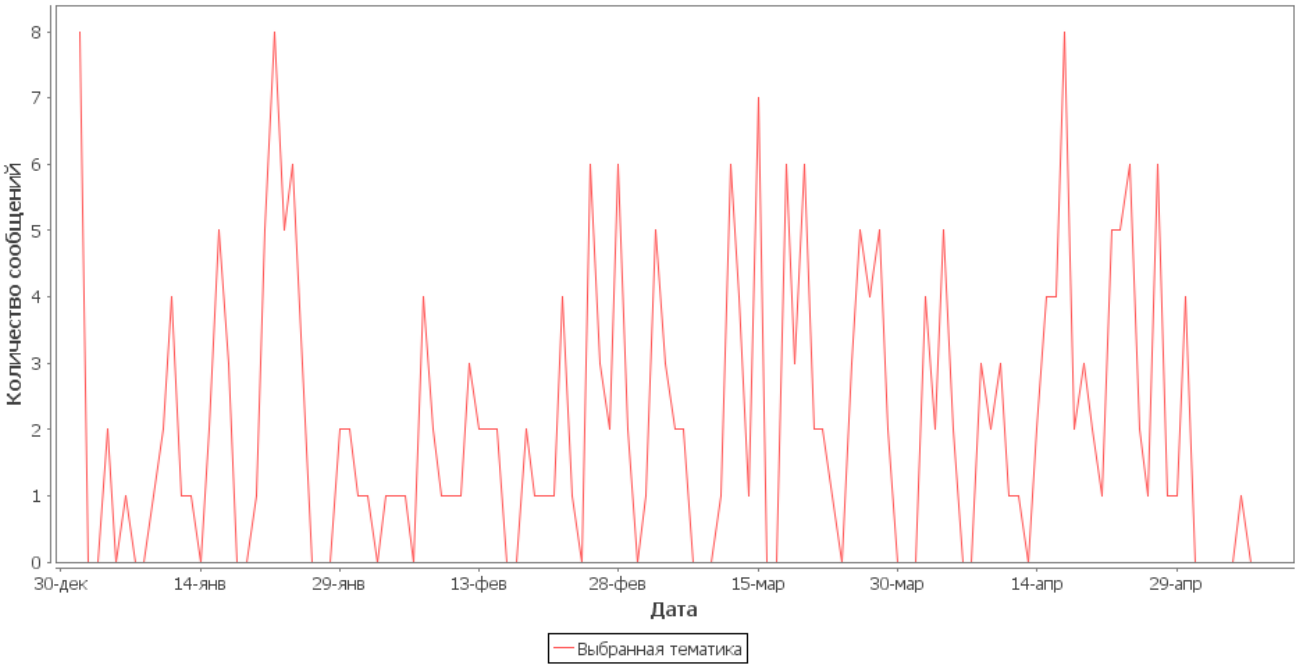
Внешняя политика



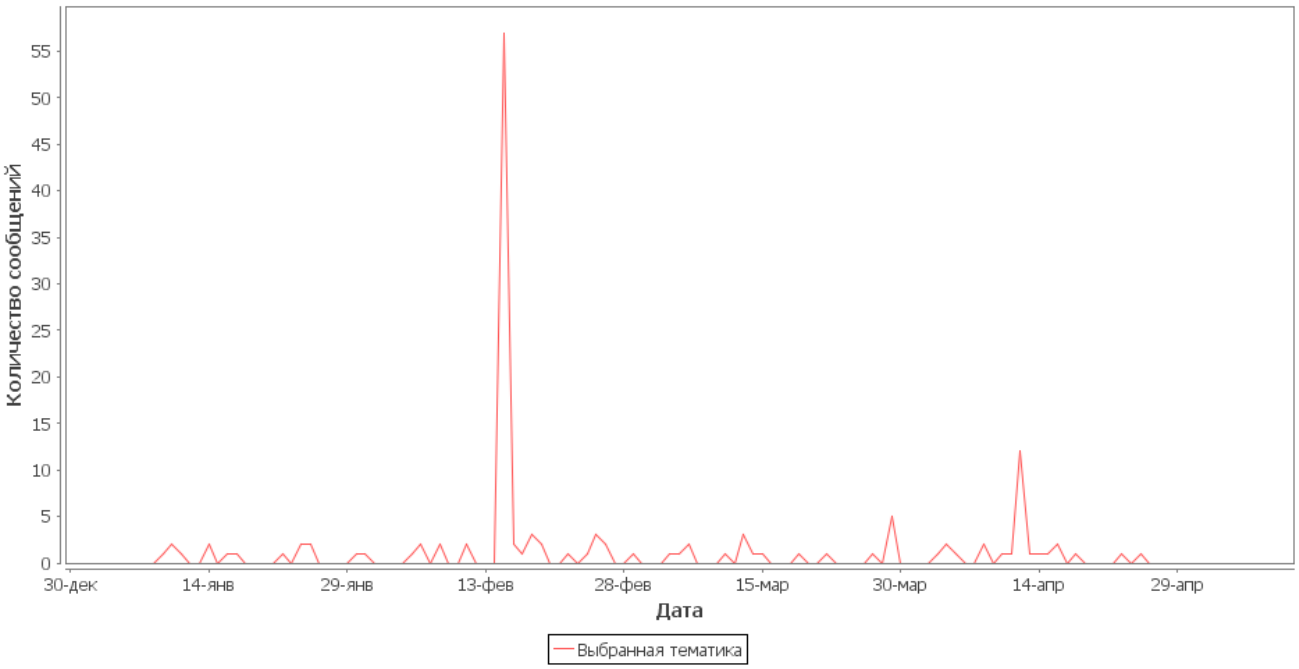
Война



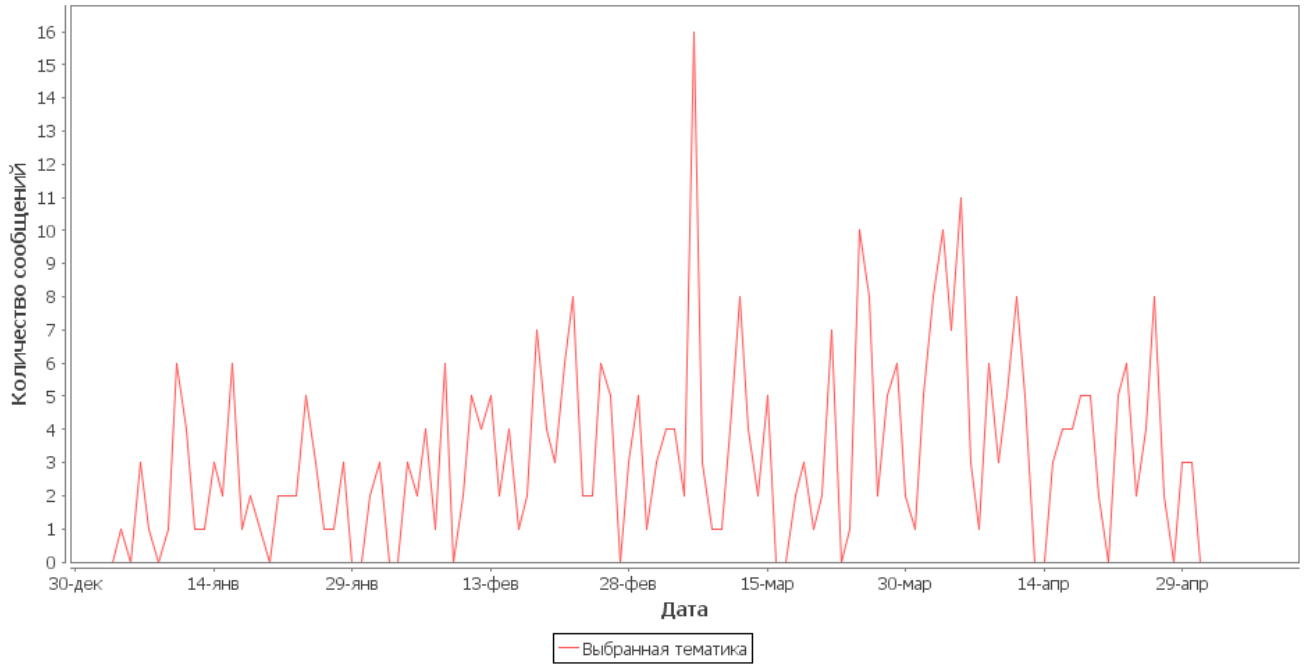
Дела, суды



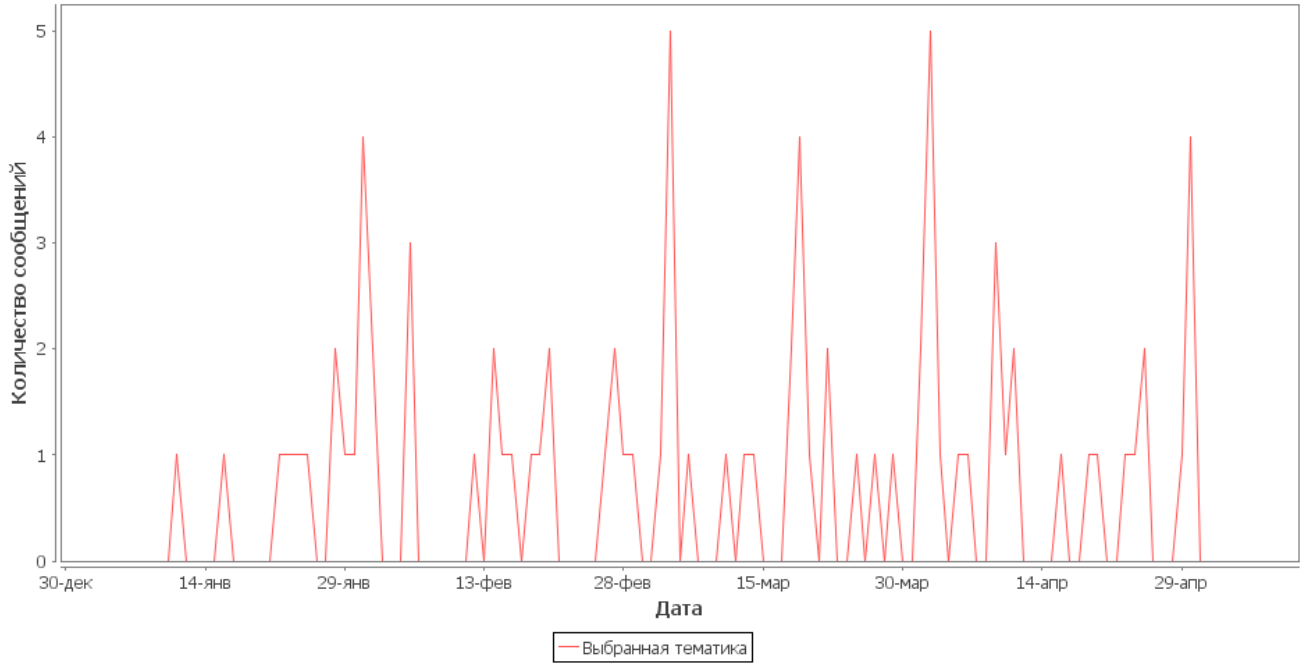
Здравоохранение



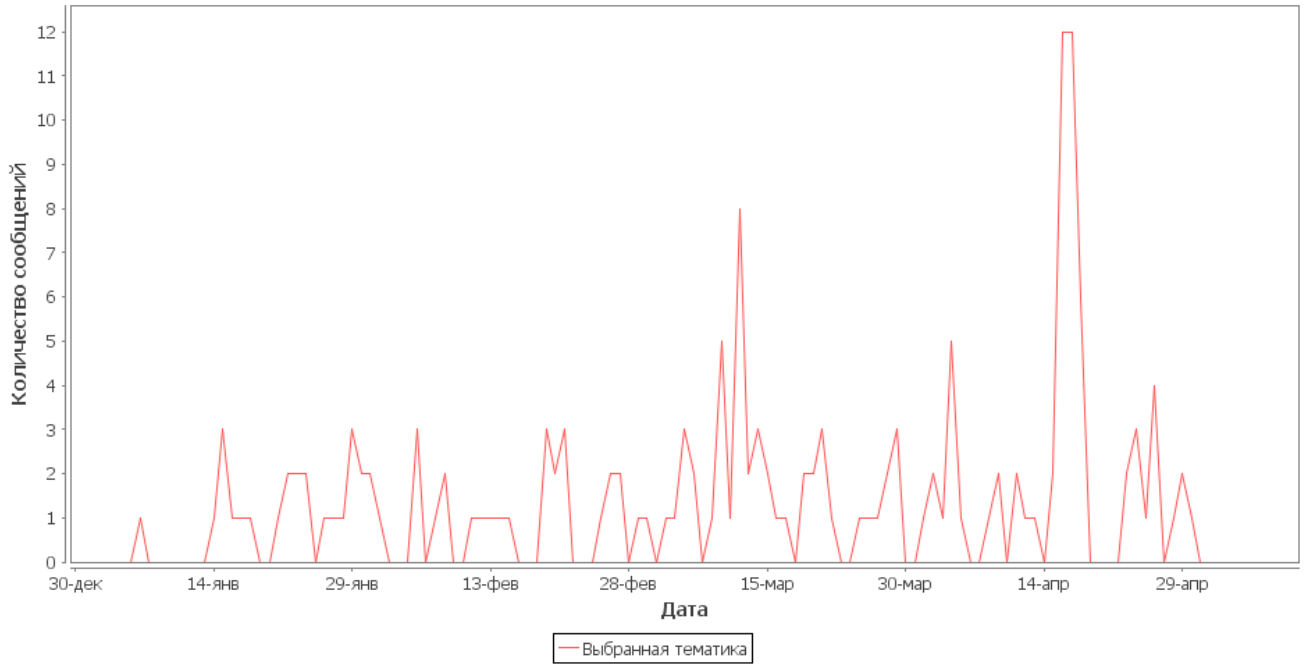
Космос



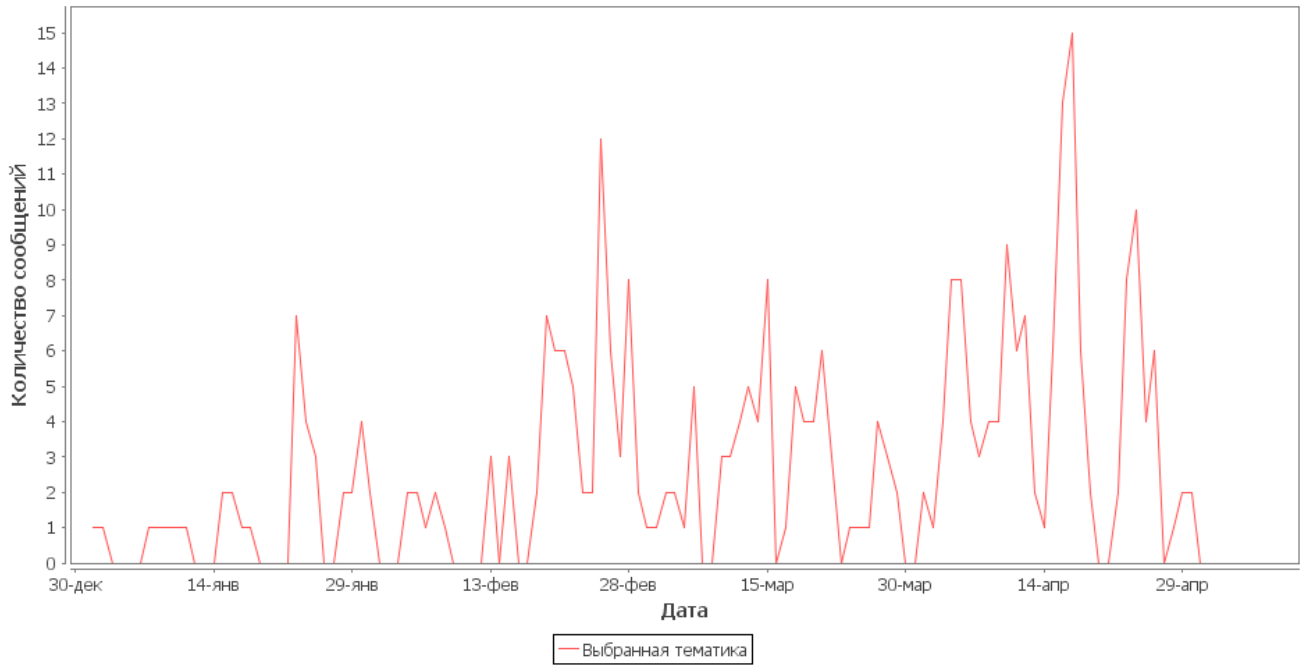
Культура



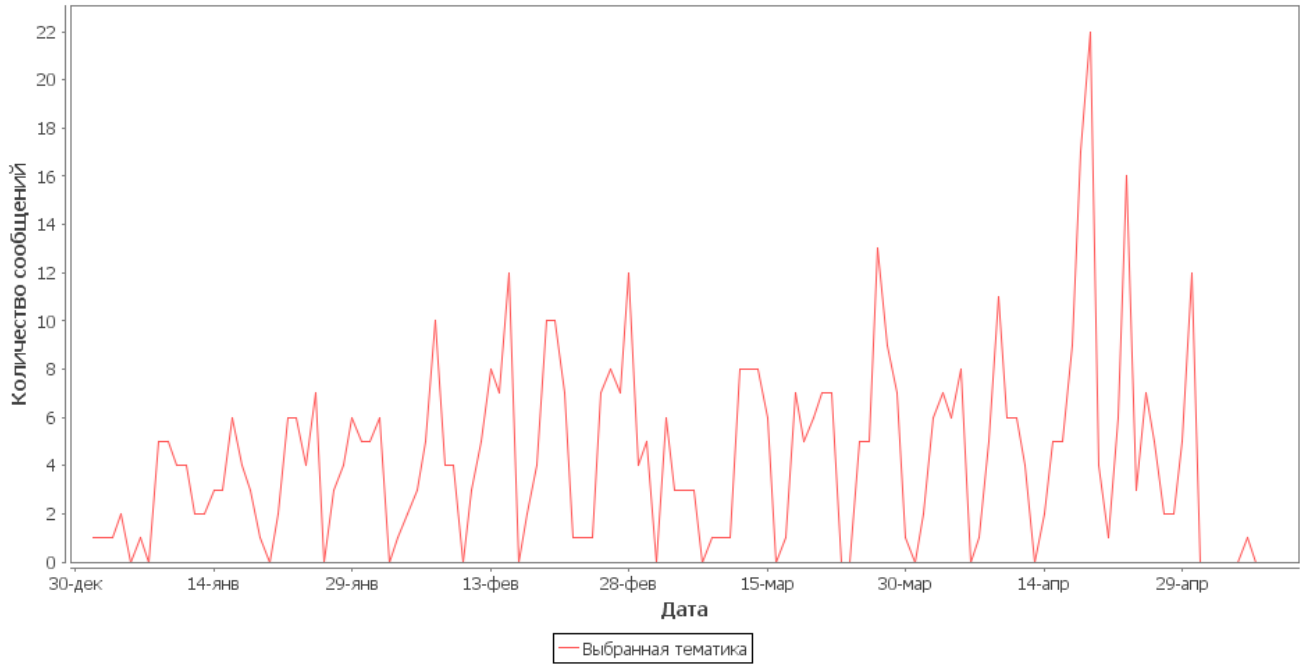
Налоги



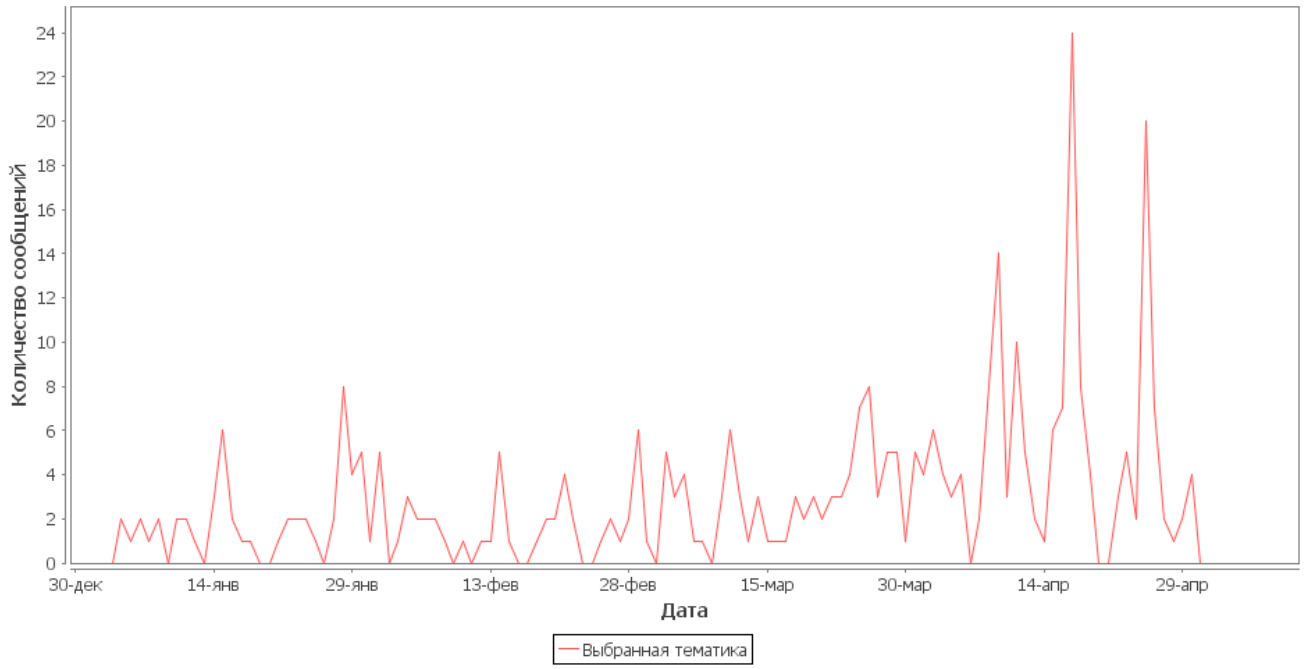
Новые технологии



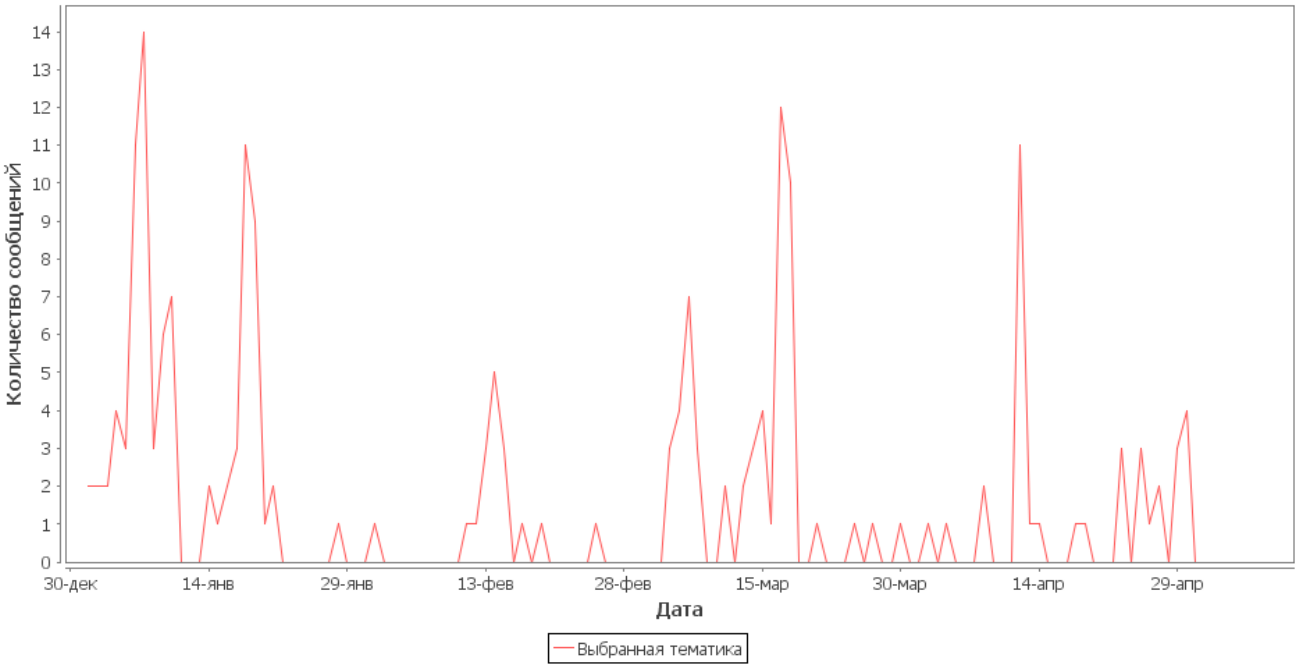
Образование



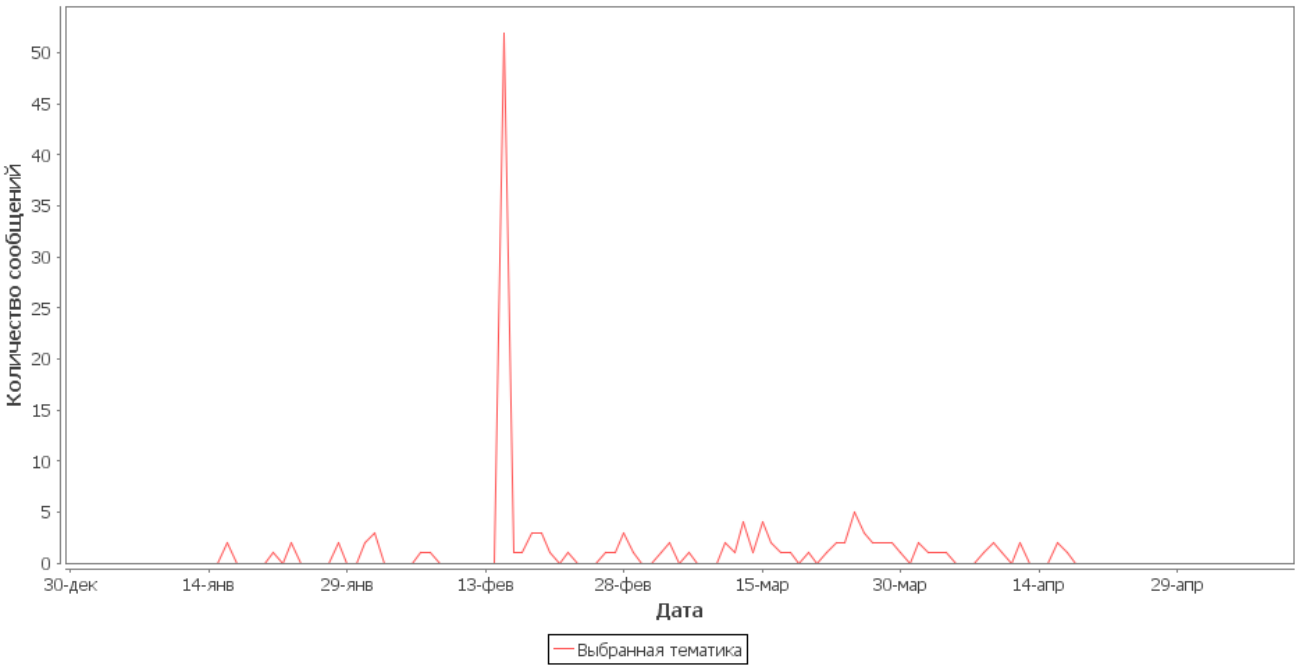
Общество



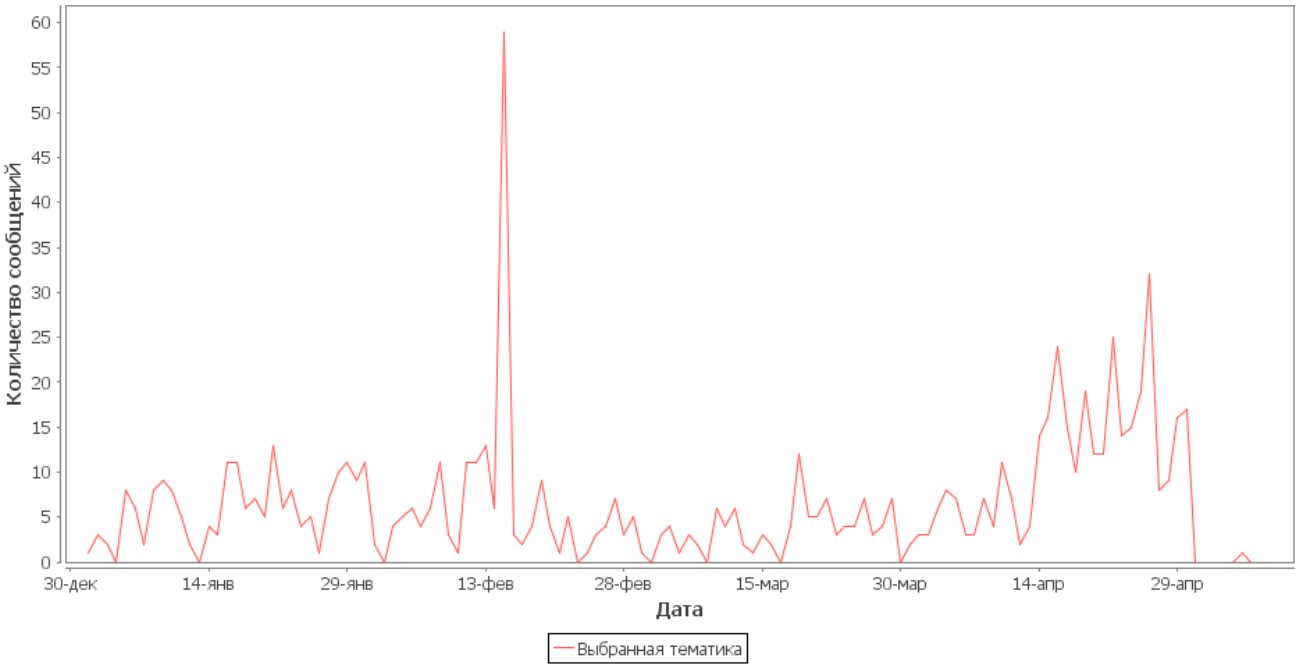
Политика



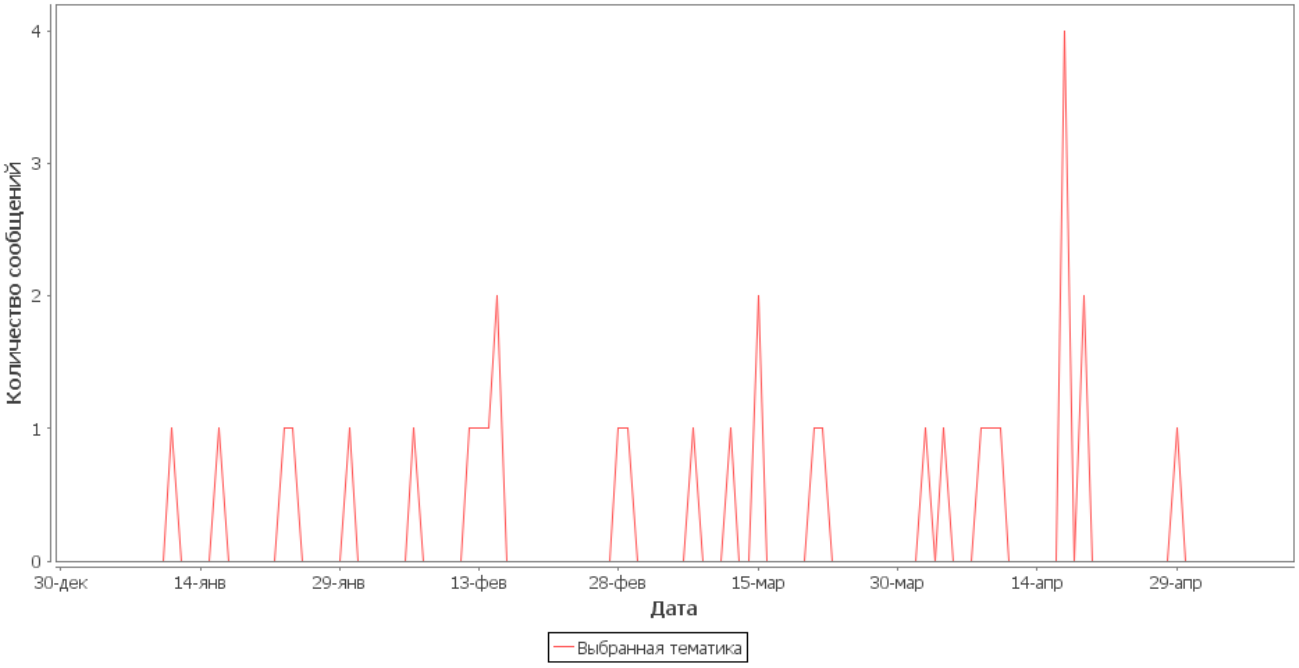
Праздники



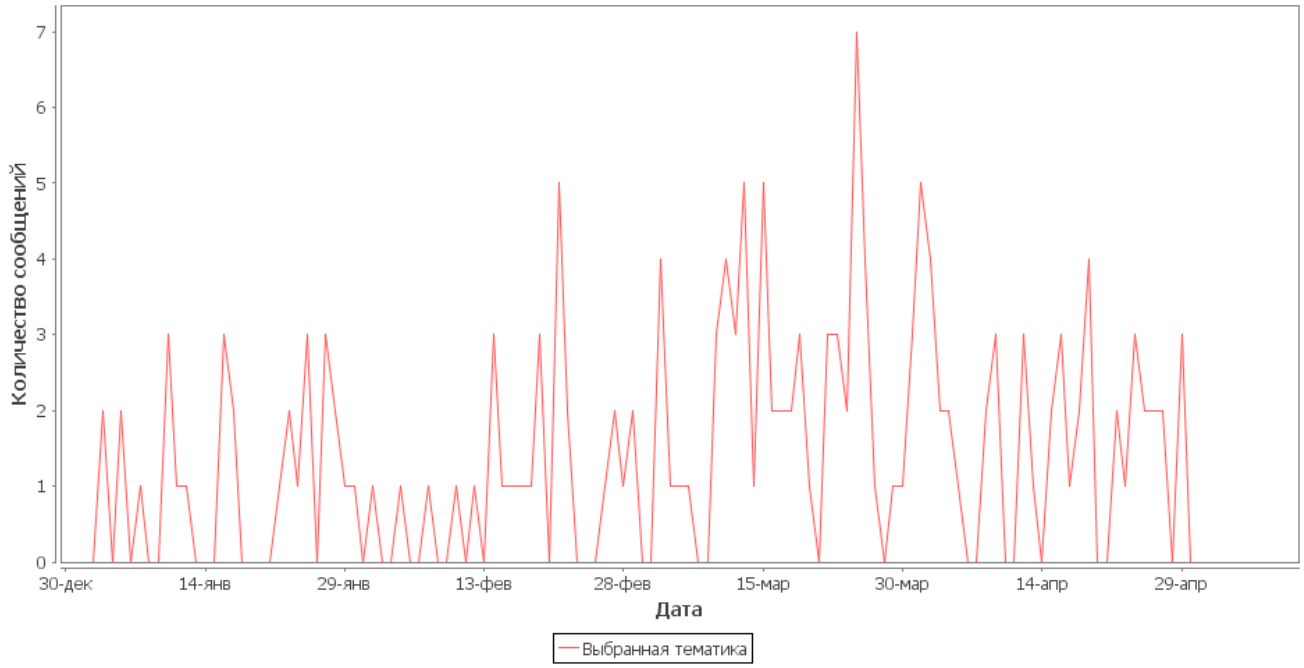
Природа



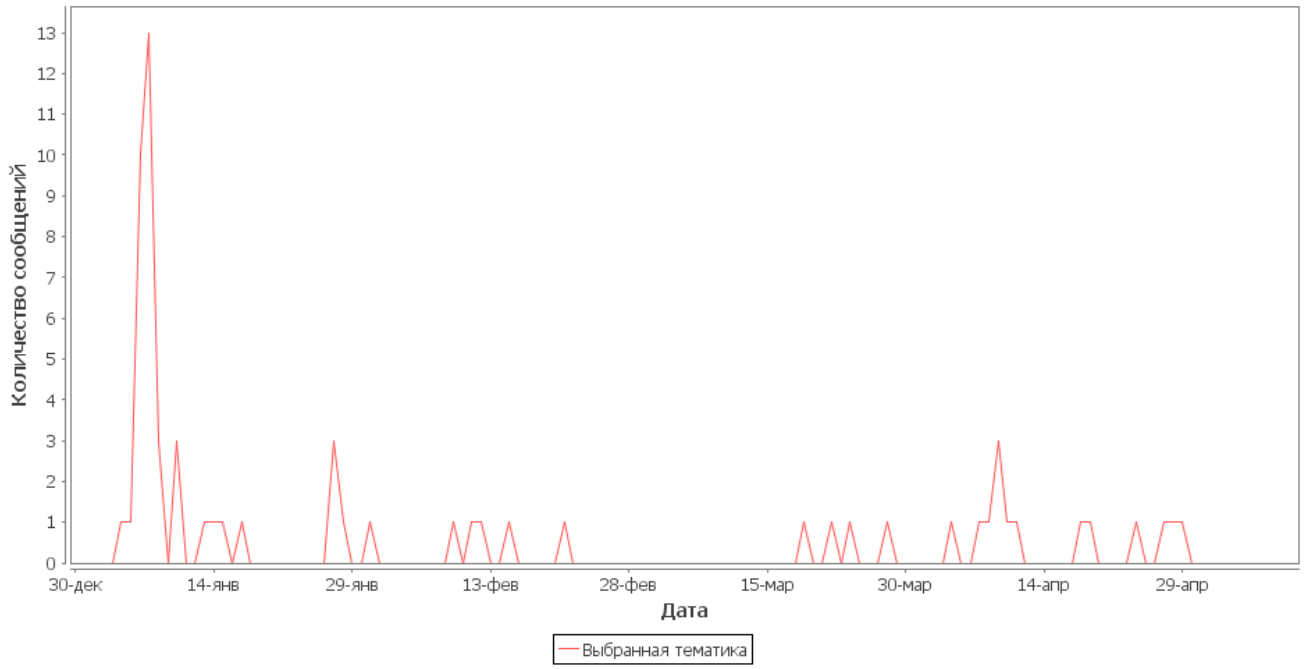
Происшествия



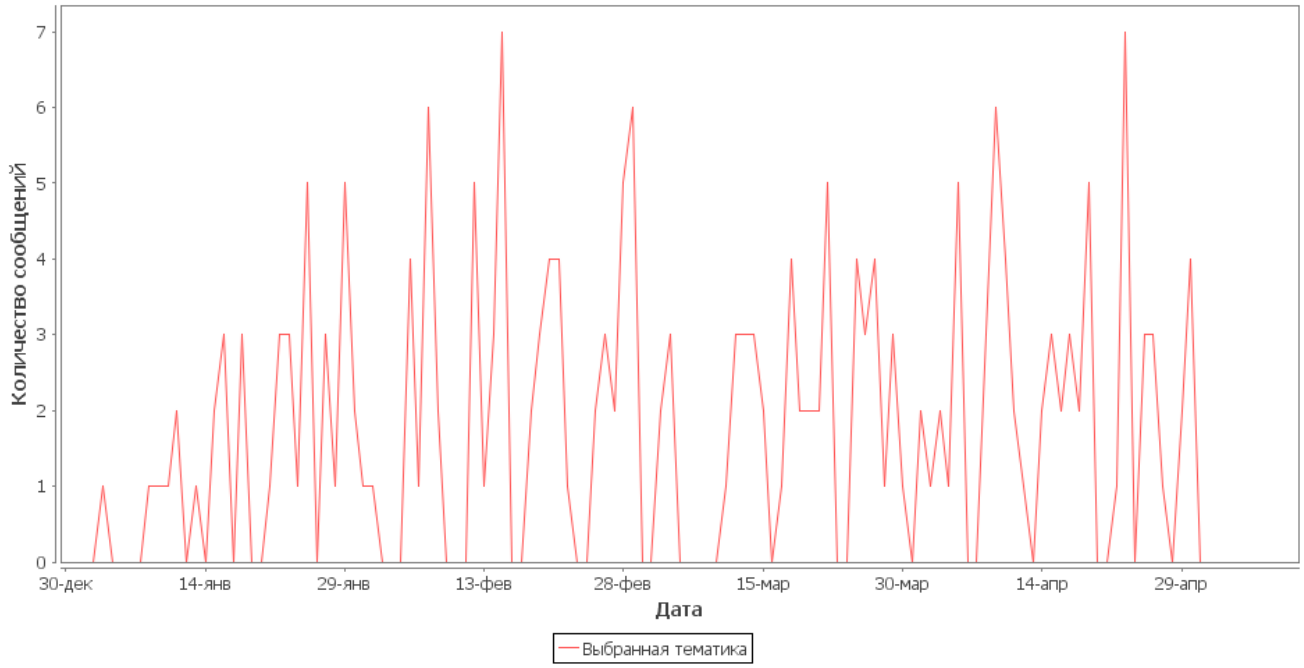
Промышленность



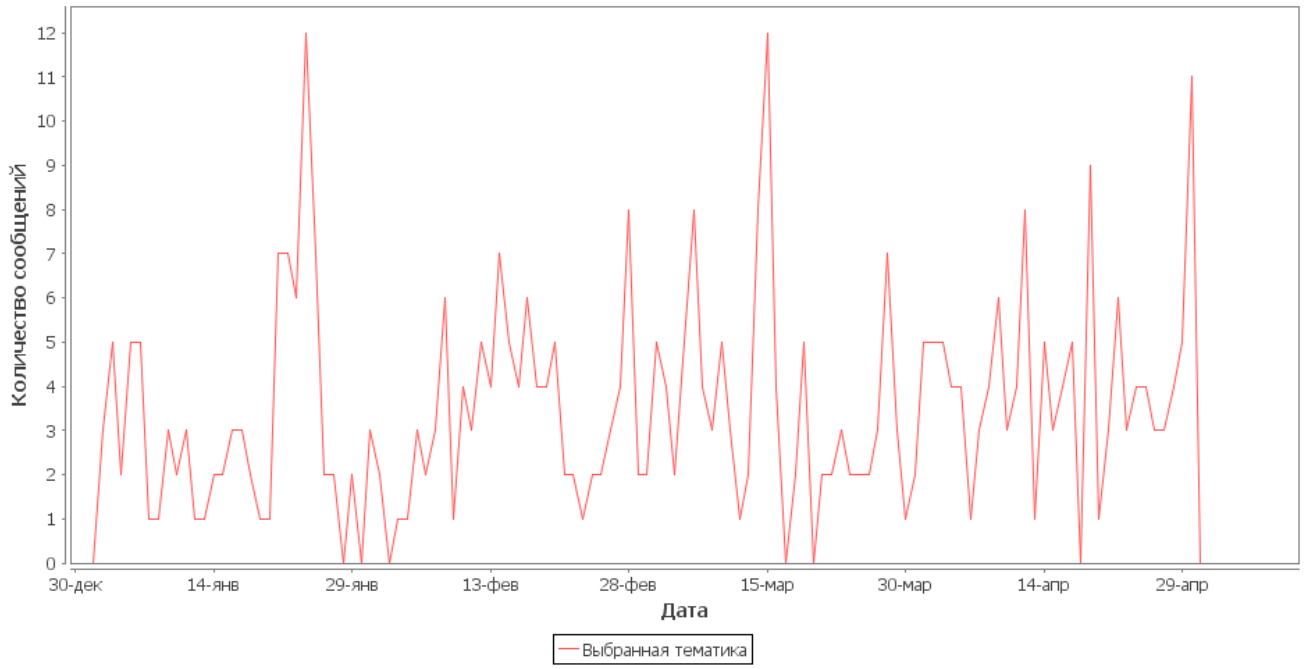
Регионы



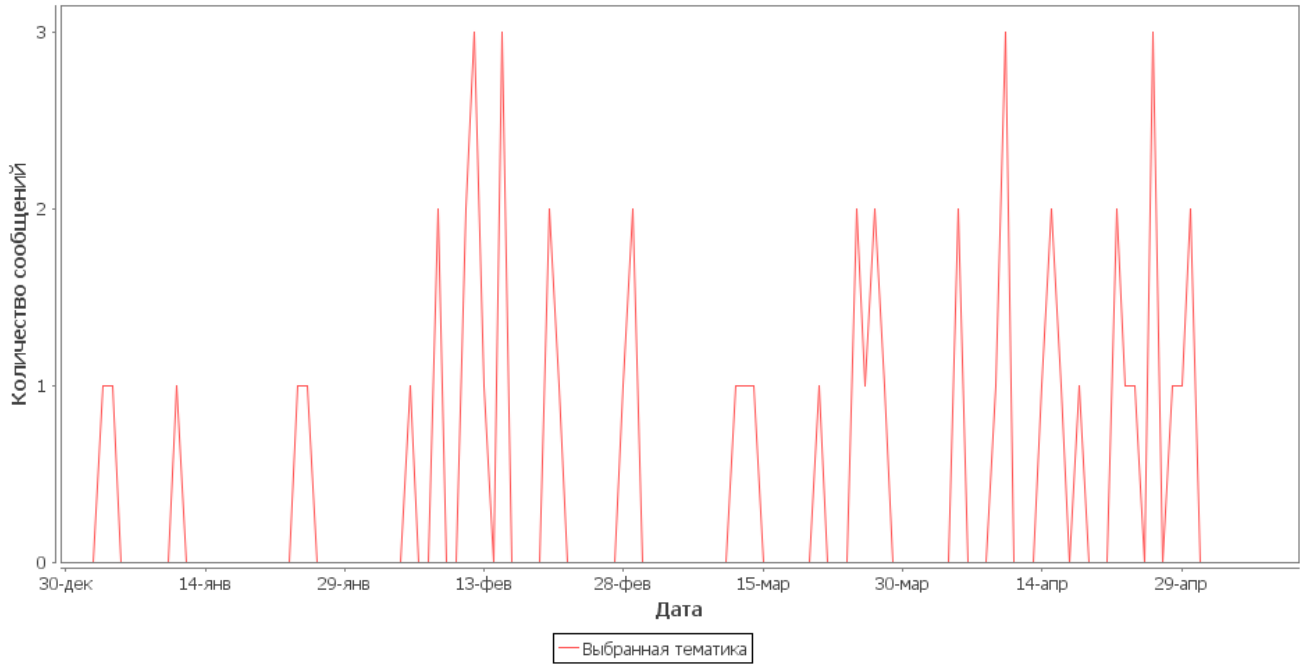
Религия



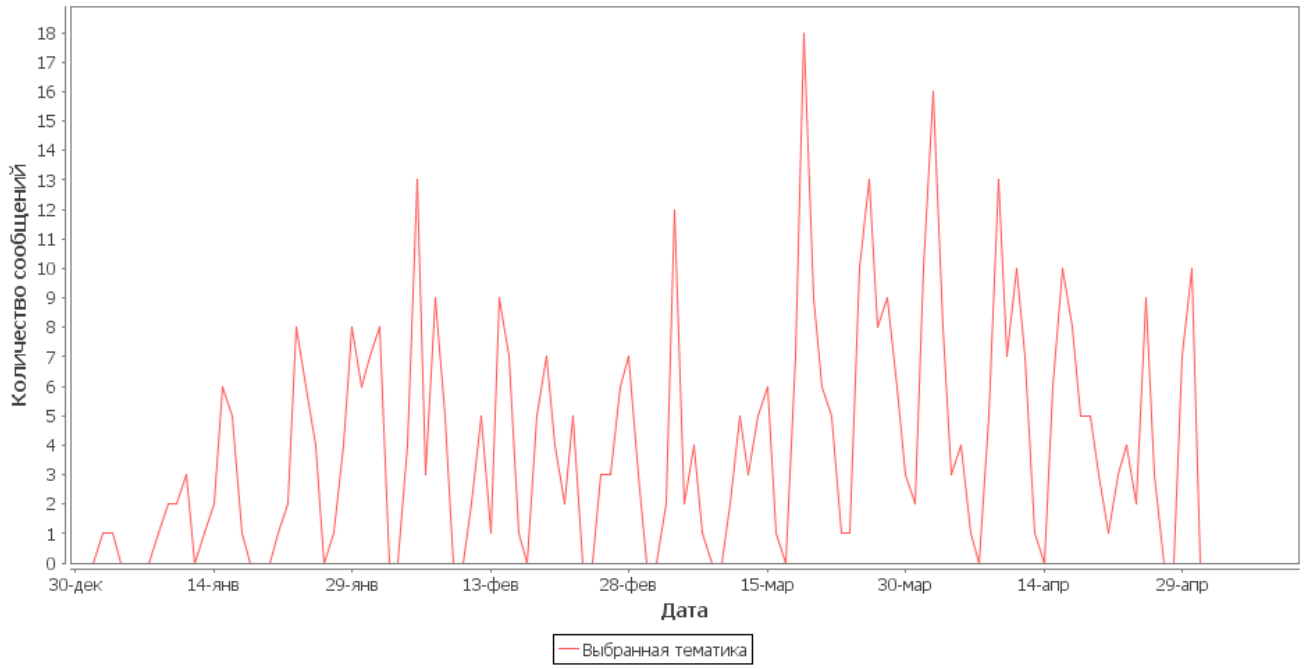
Реформы



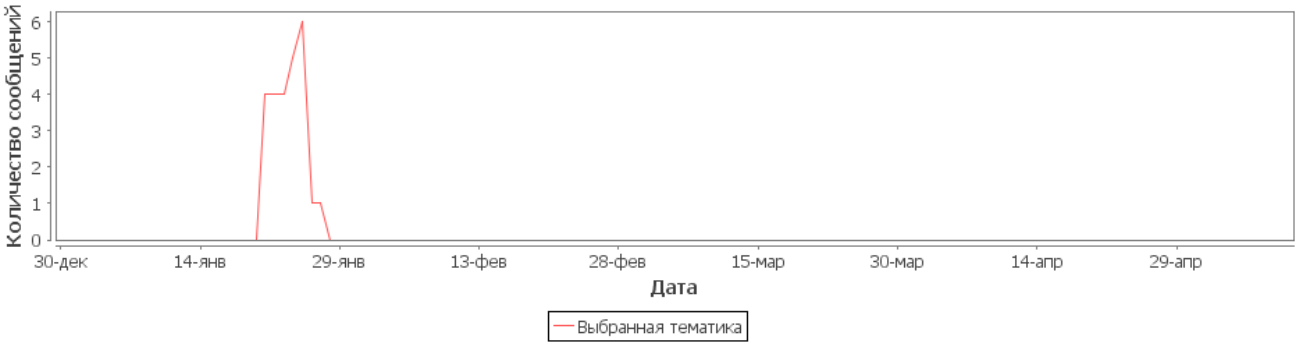
Спорт



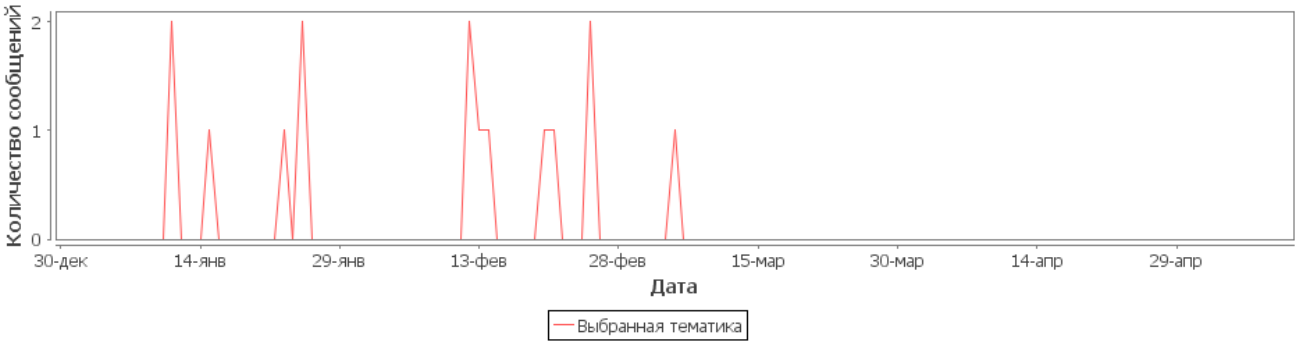
Уголовное право



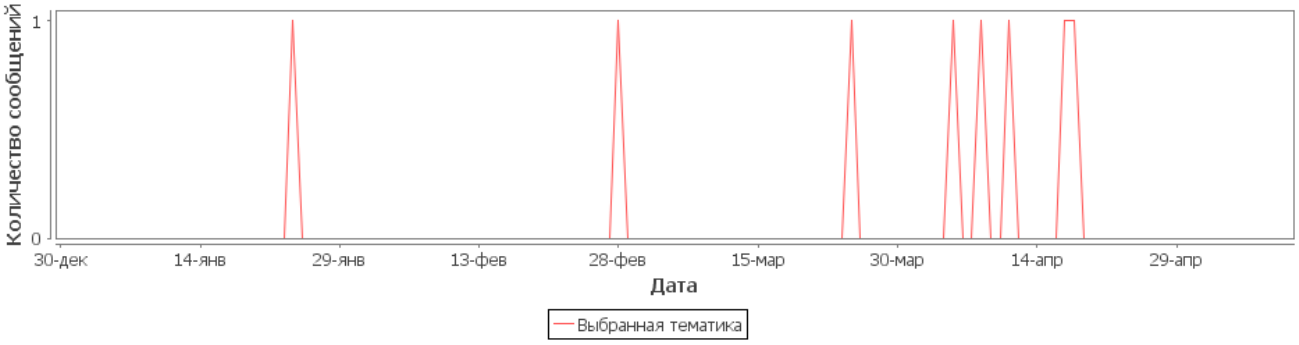
Экономика



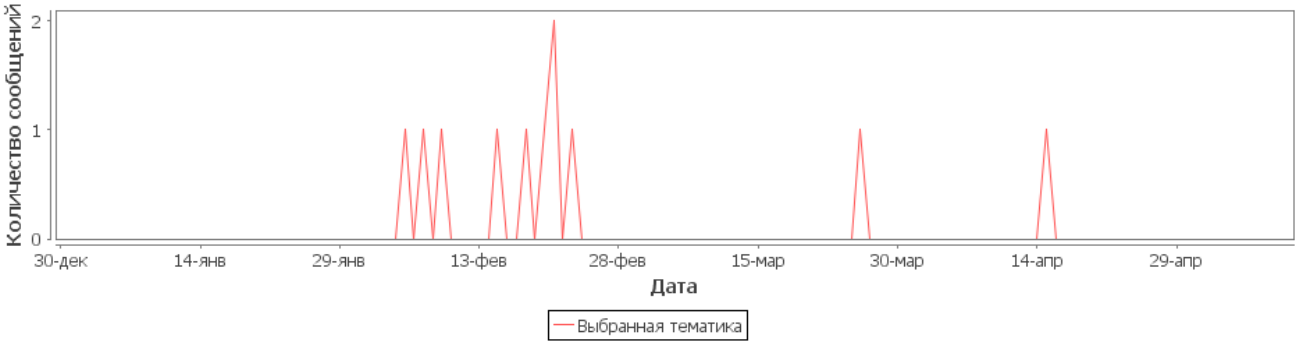
Australian Open



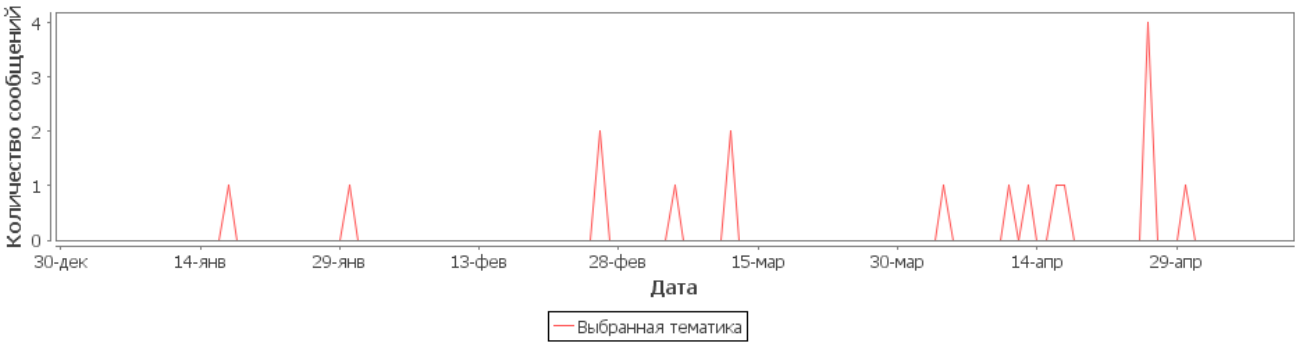
Борьба с курением



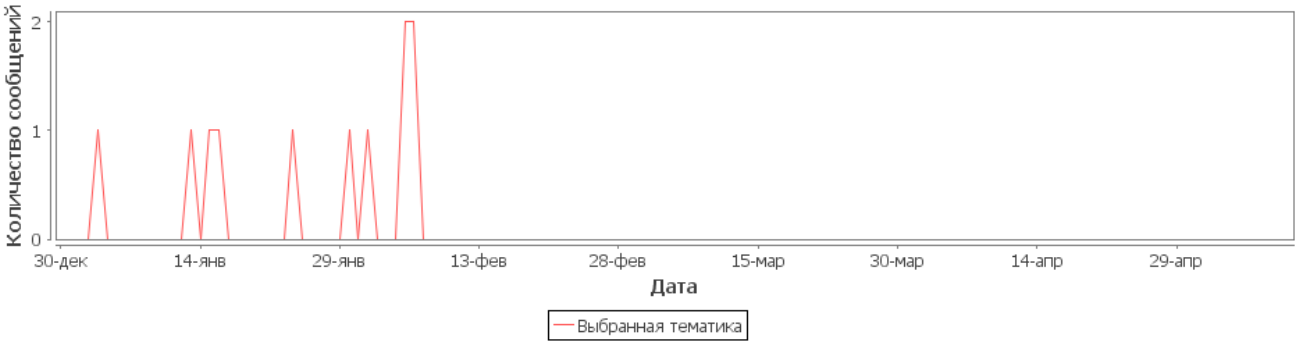
Выборы губернаторов



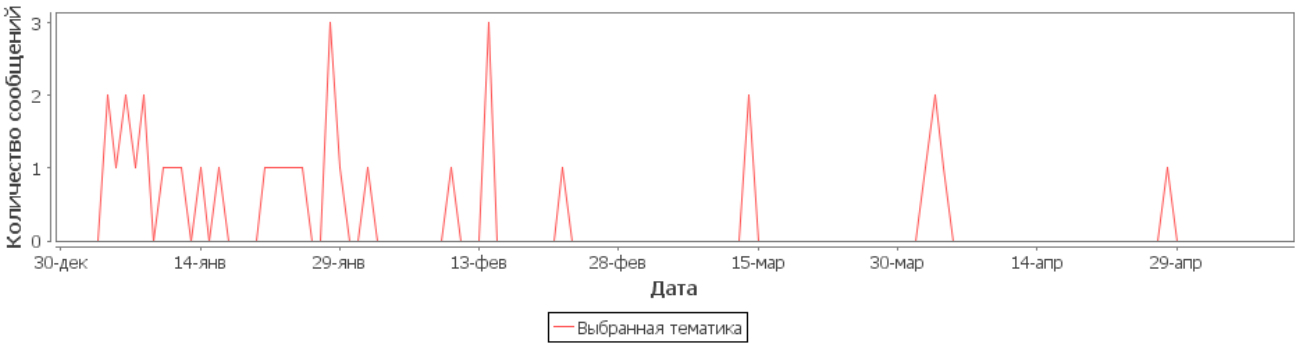
Выставки вооружений



Глонасс

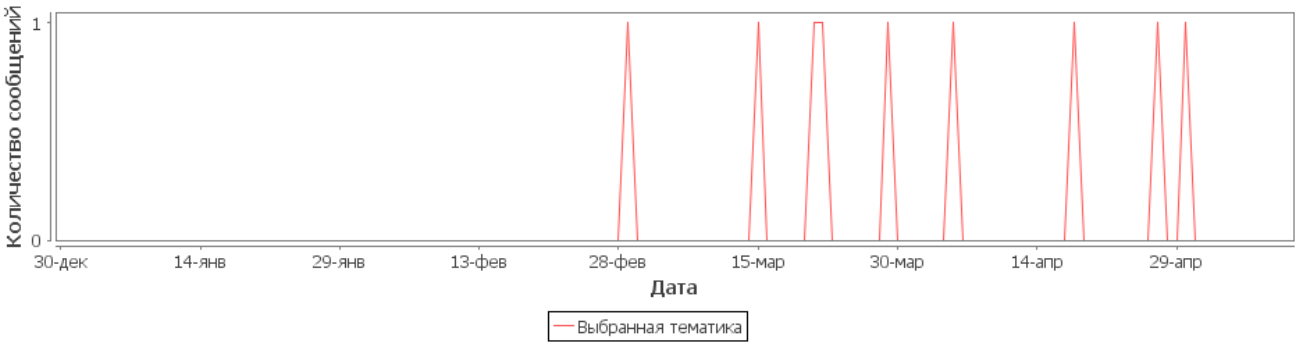


Госдолг США

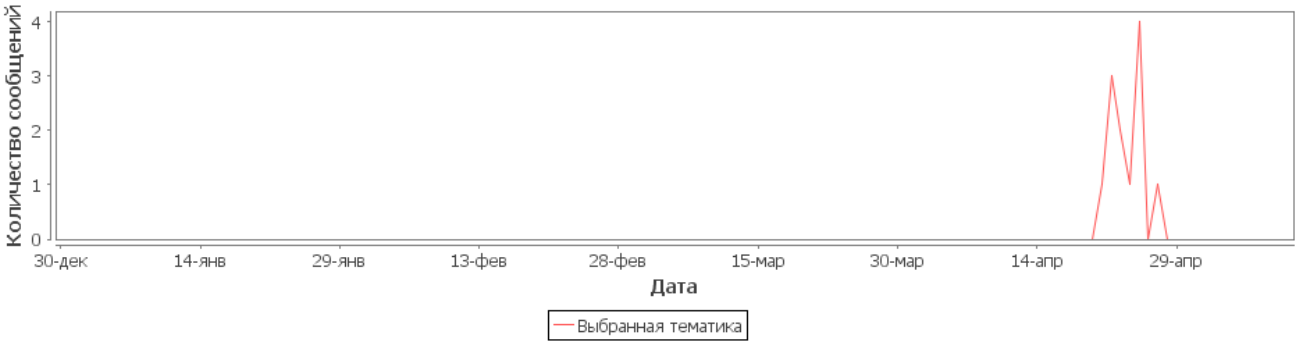


Дело Сергея Полонского

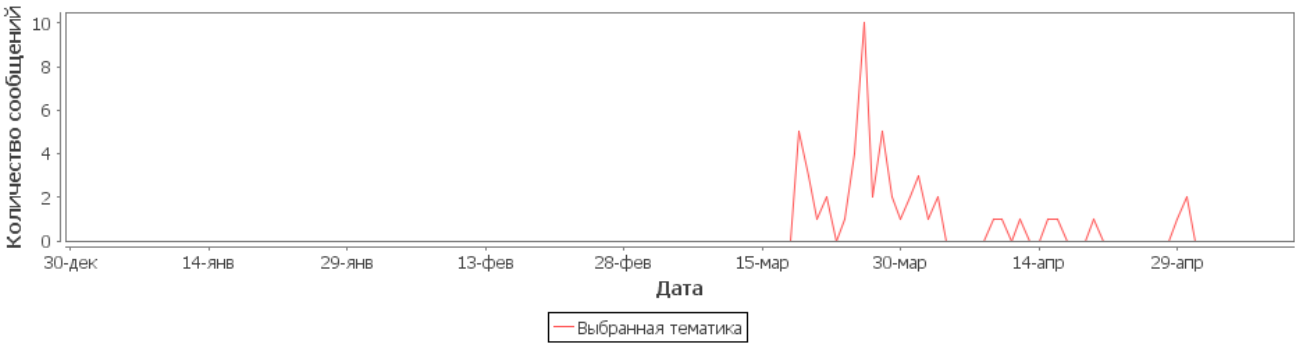
Досрочные выборы в Венесуэле



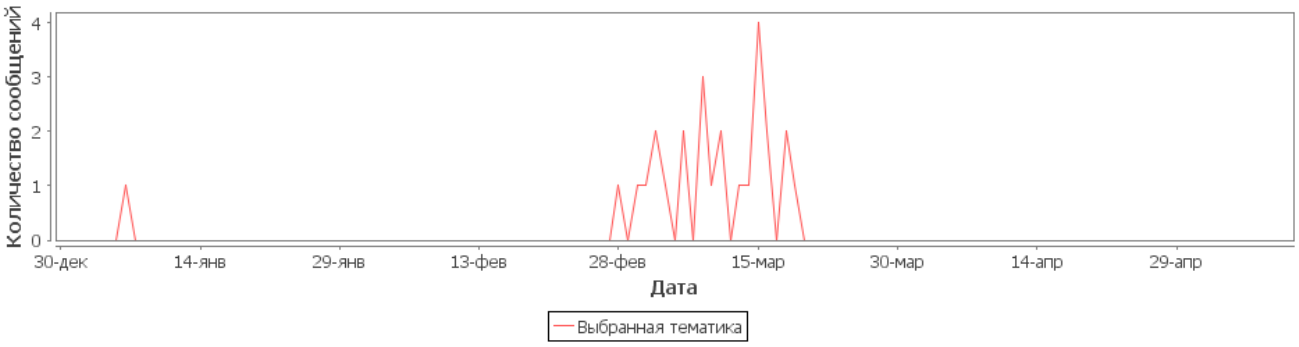
Закон о рекламе



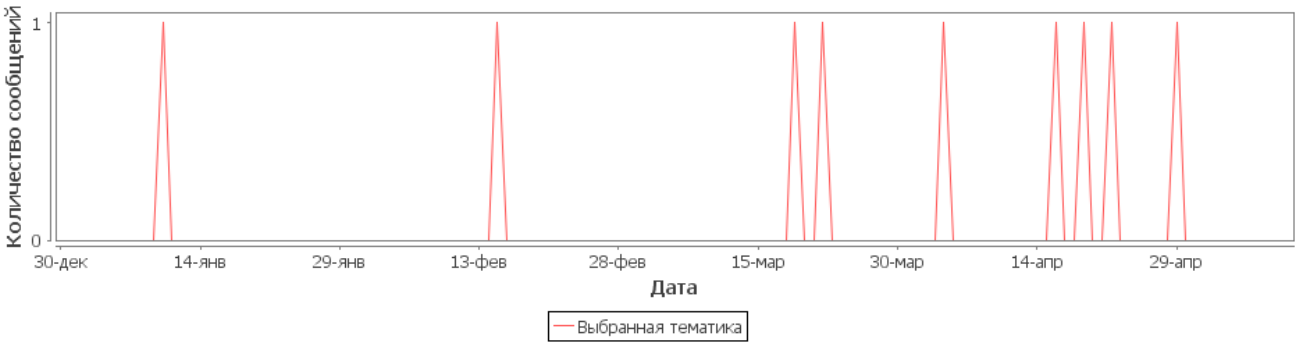
Землетрясения в Китае



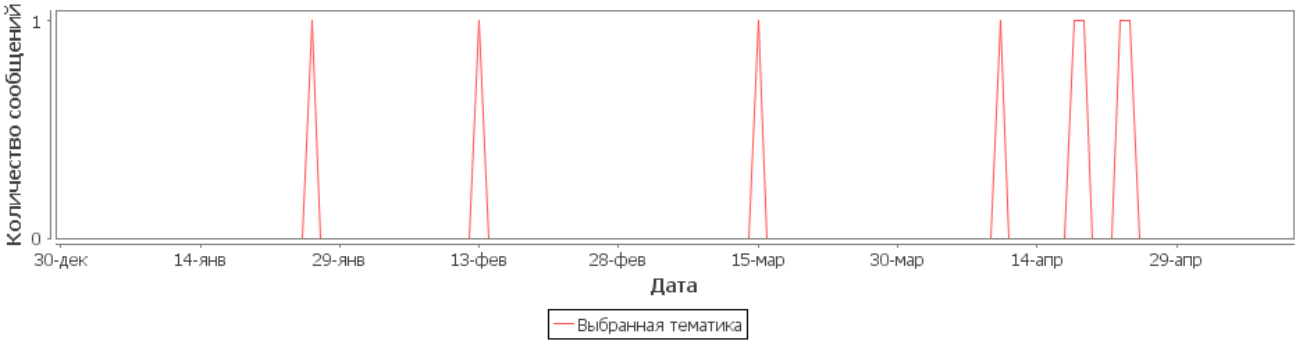
Кризис на Кипре



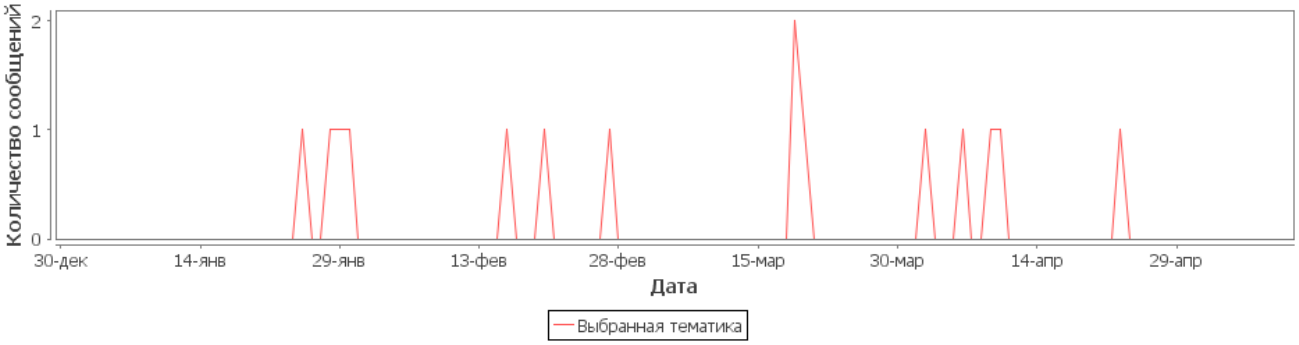
Кубок мира по Биатлону



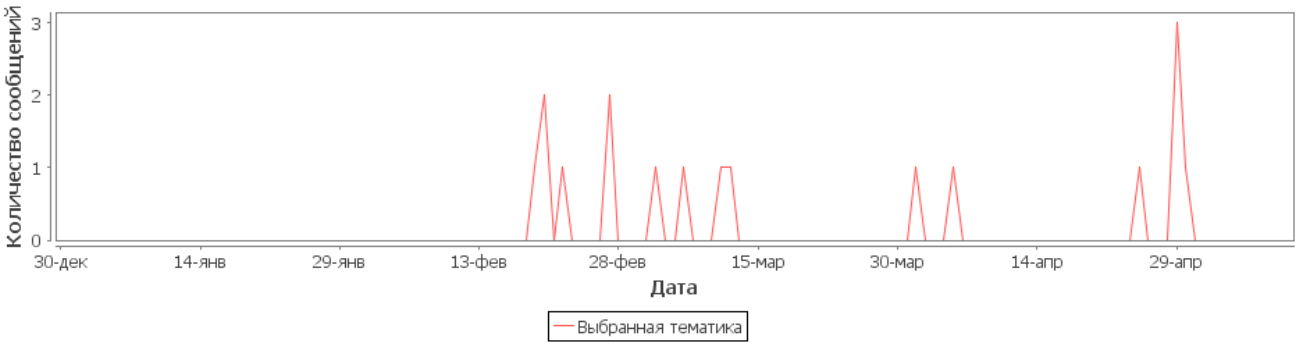
Лесные пожары



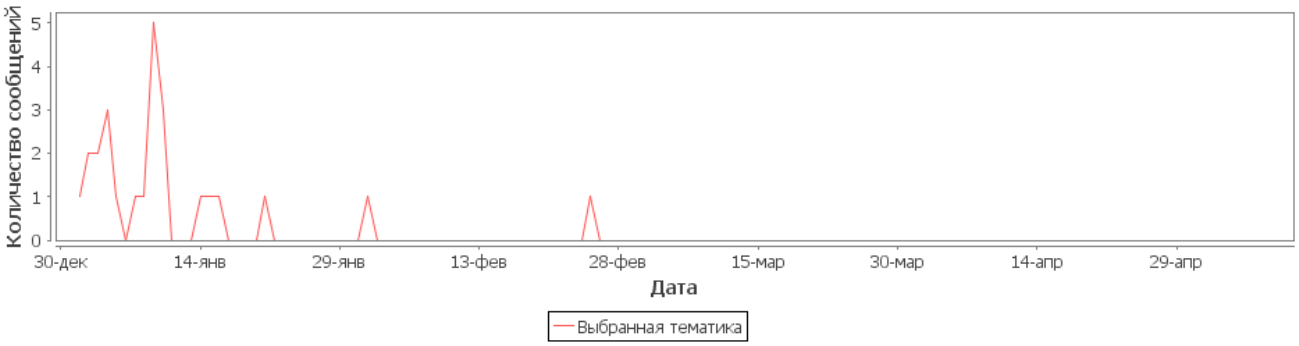
Музеи оружия



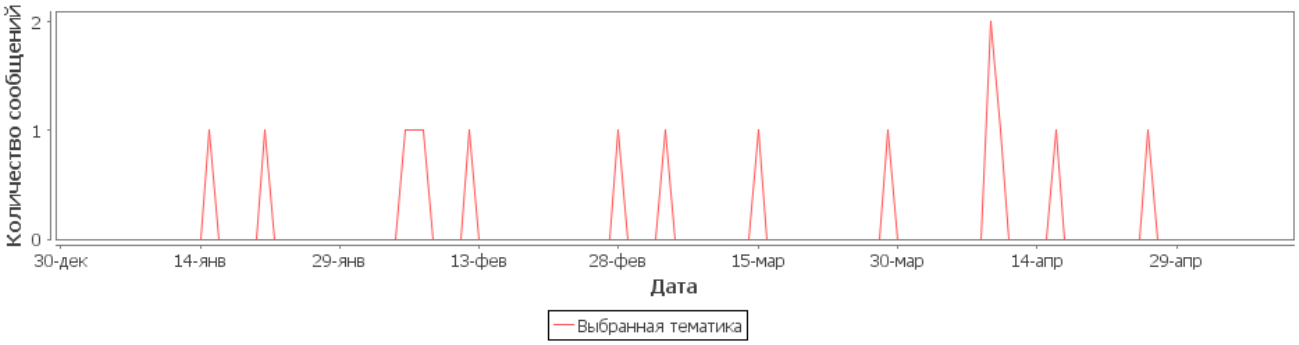
Налоги на малый бизнес



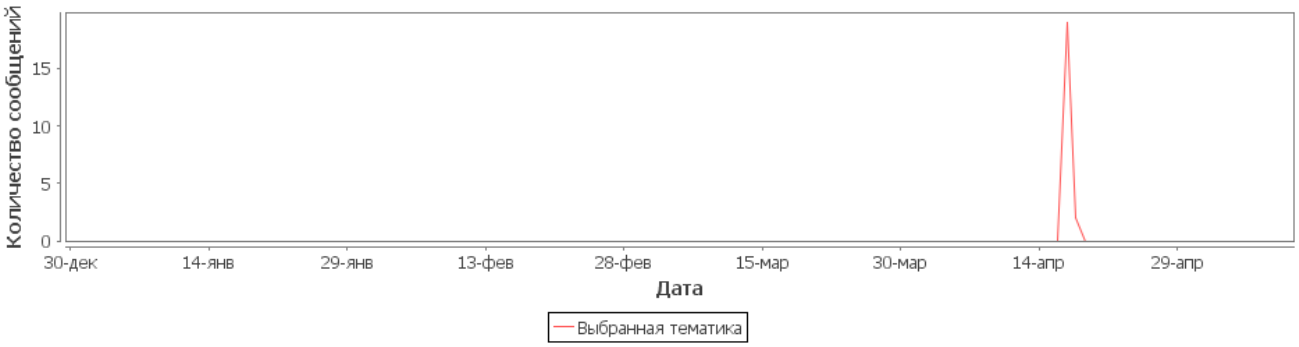
Нападения пиратов



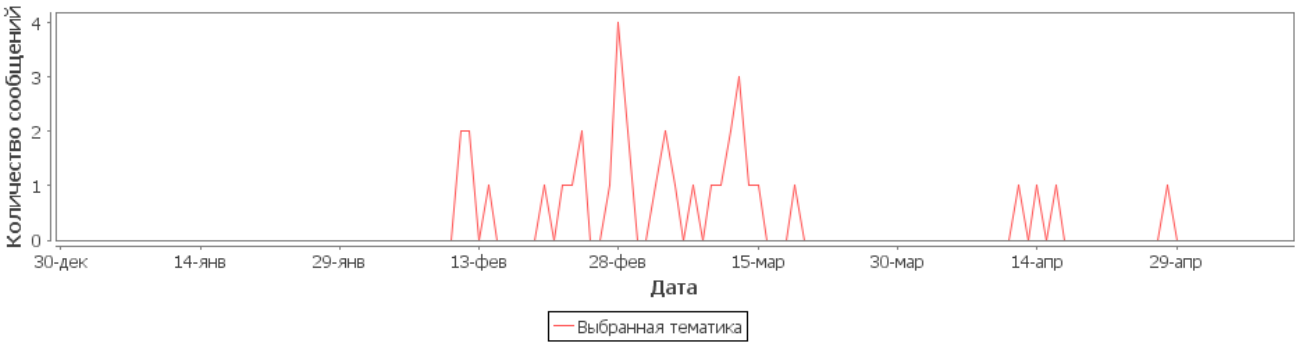
Новый год



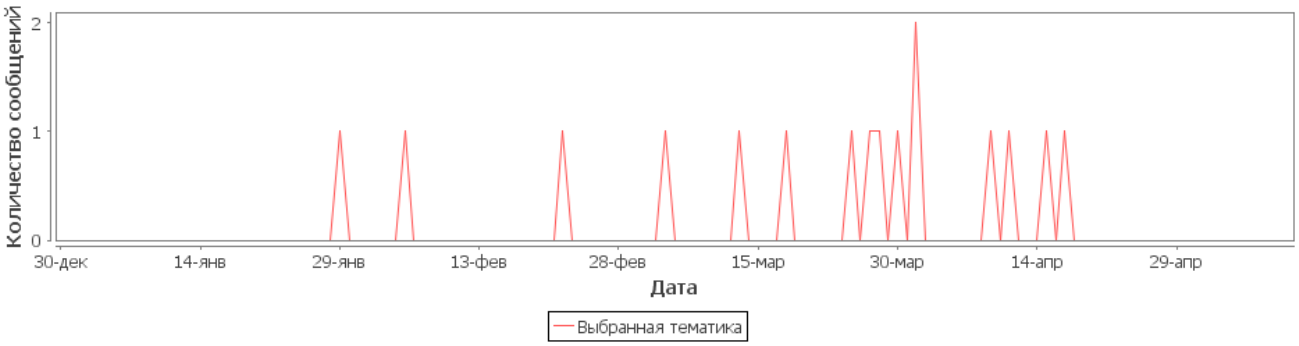
Олимпиада-2014



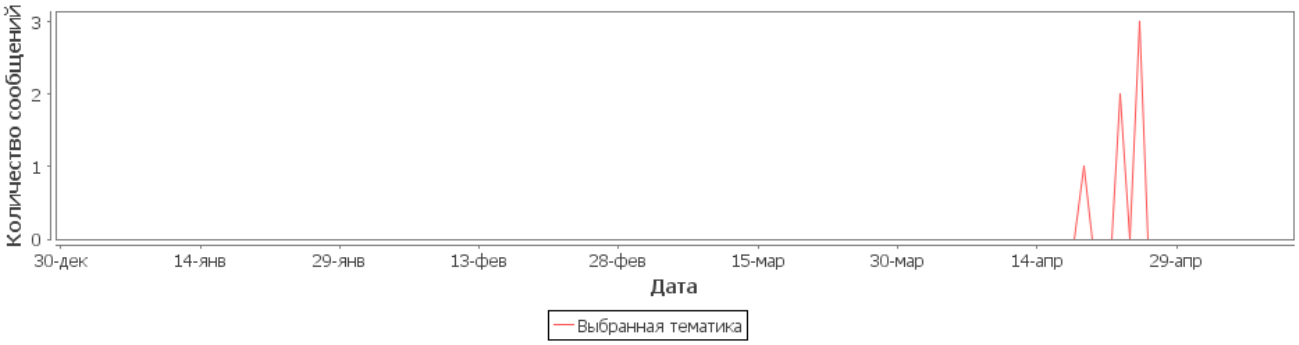
Отчёт Медведева перед Госдумой



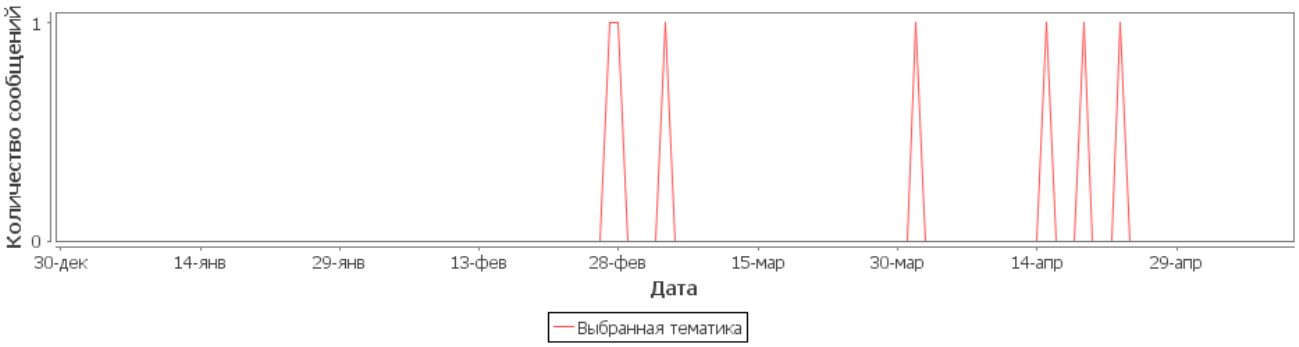
Папа Римский Франциск



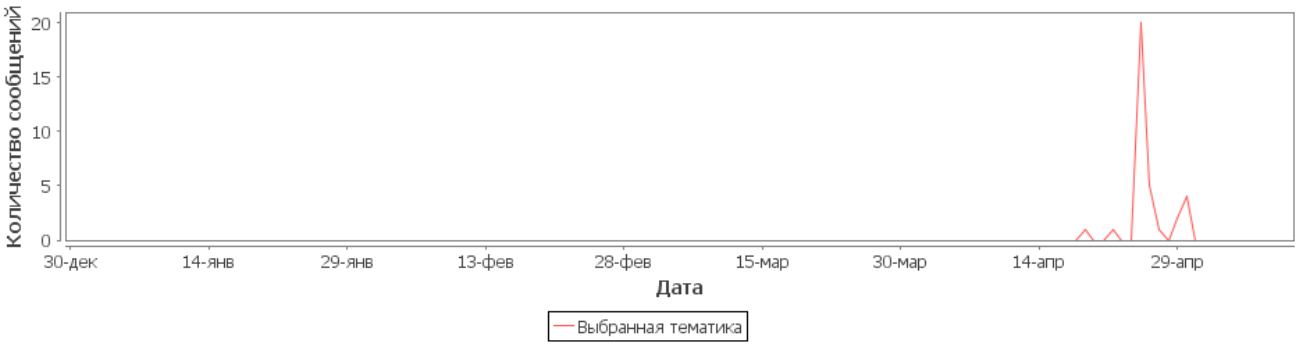
Пенсионная реформа



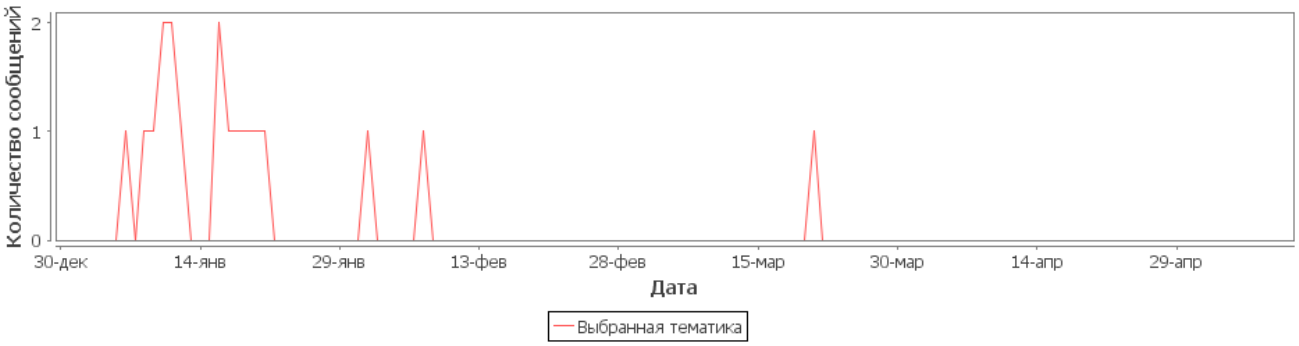
Празднование Дня Победы



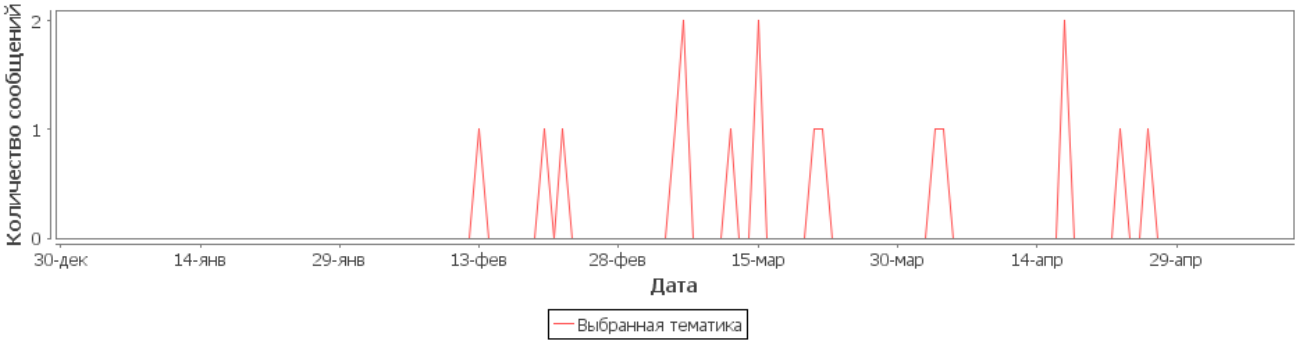
Призыв в армию



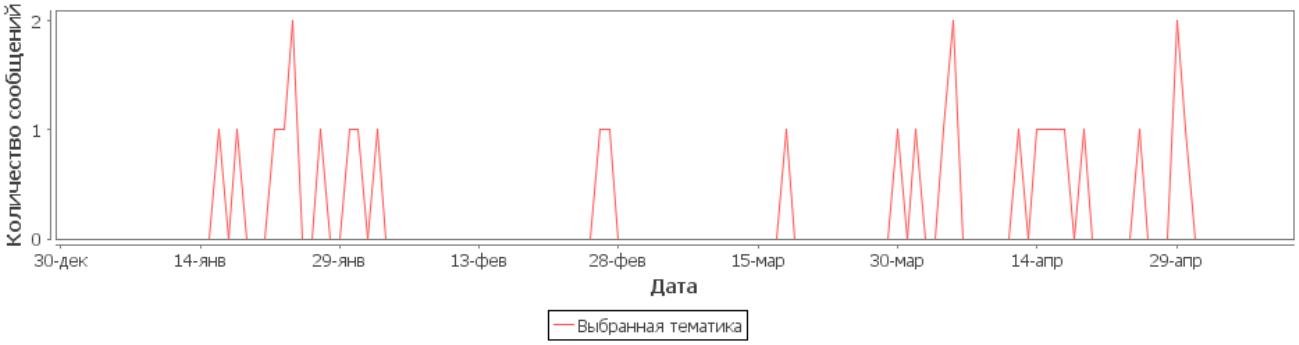
Прямая линия с Владимиром Путиным



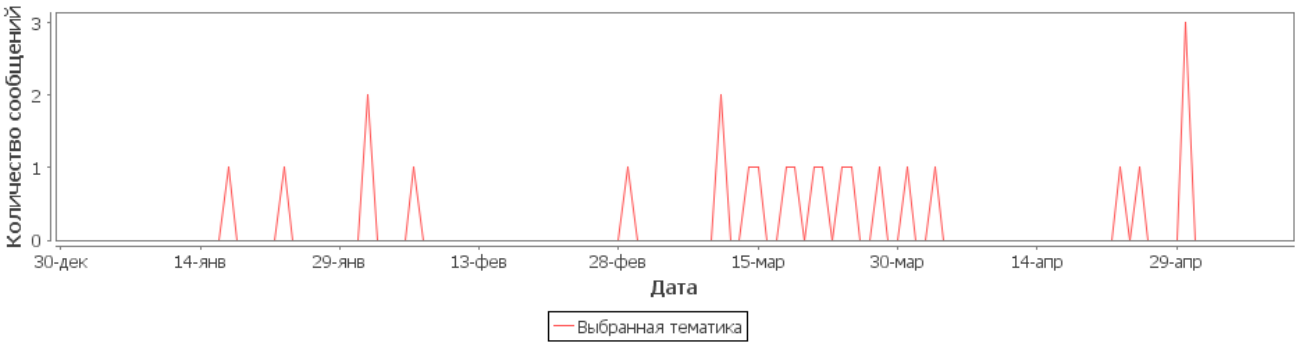
Ралли Дакар



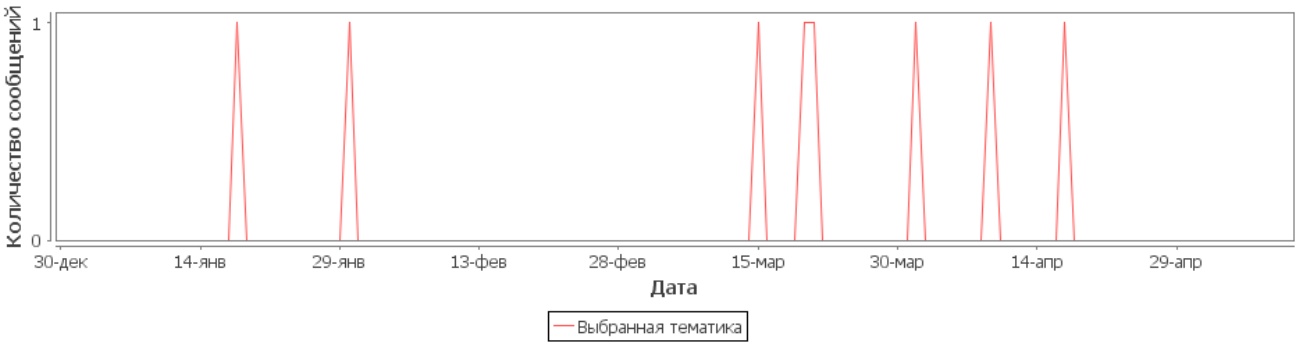
Реформа образования



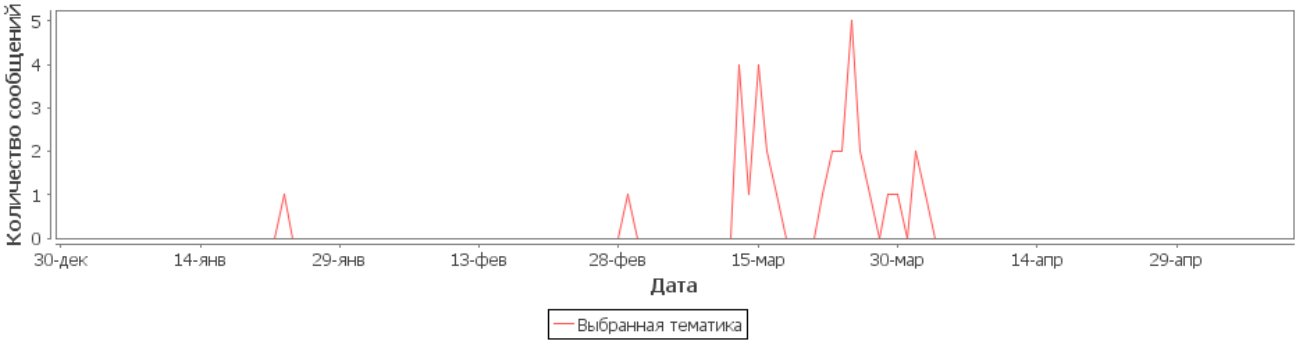
Россия и США



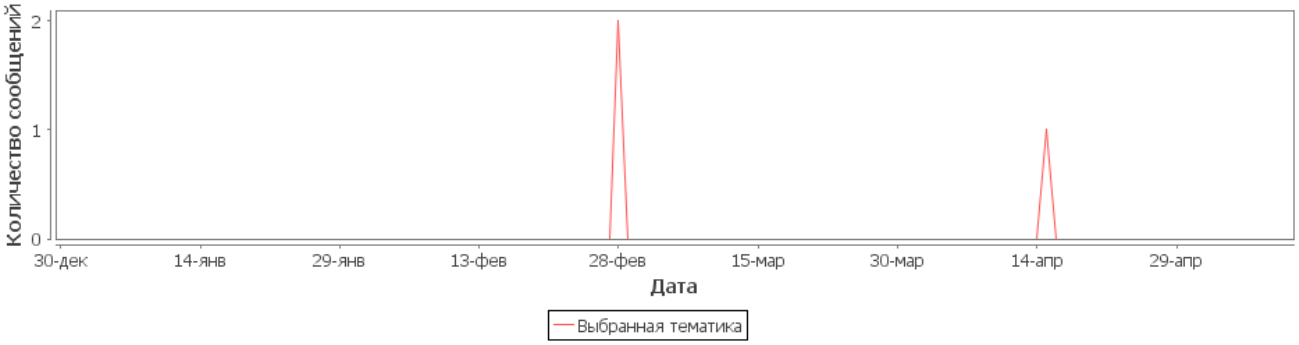
Ситуация в Сирии



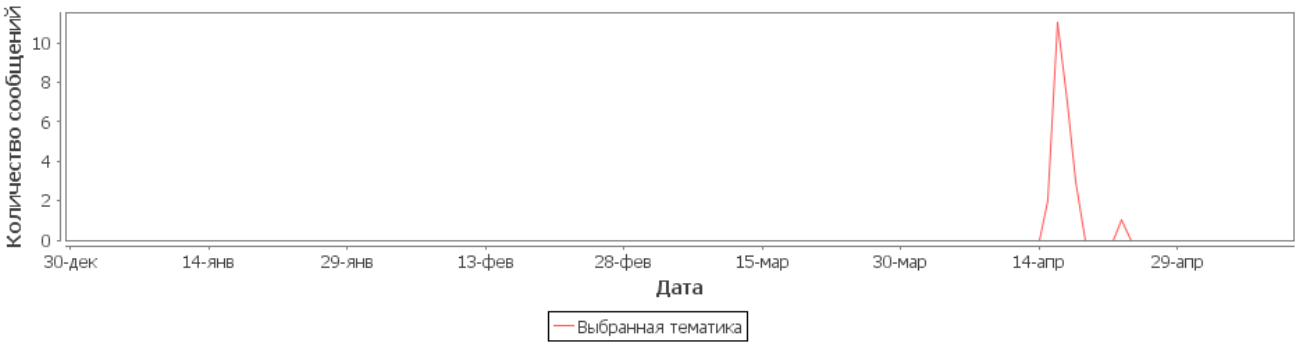
Сколково



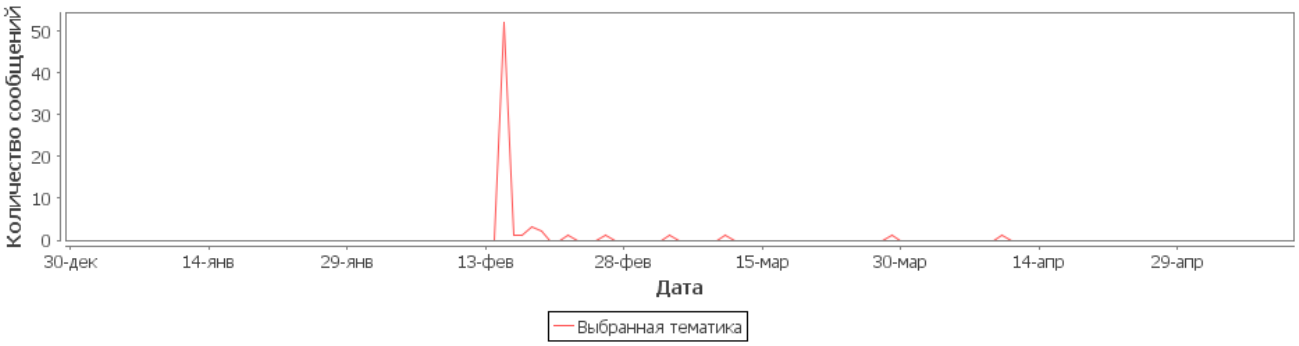
Снегопад в Москве



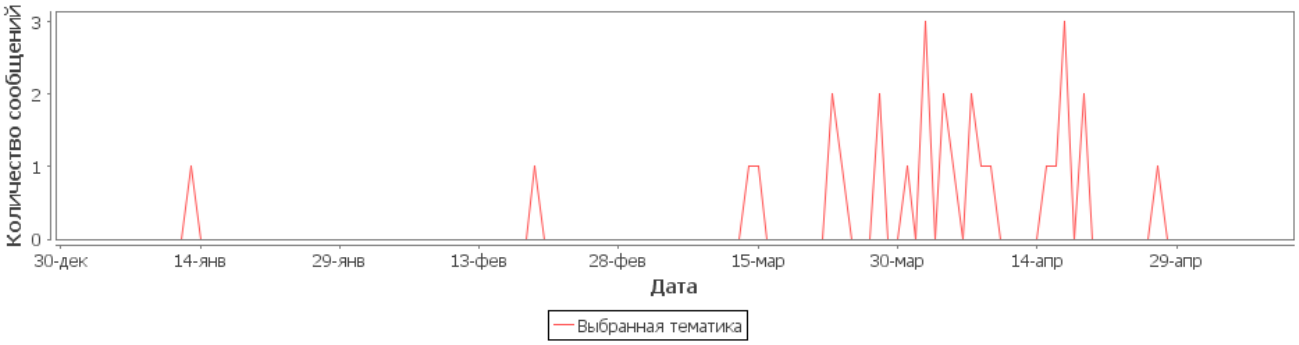
Тайны НЛО



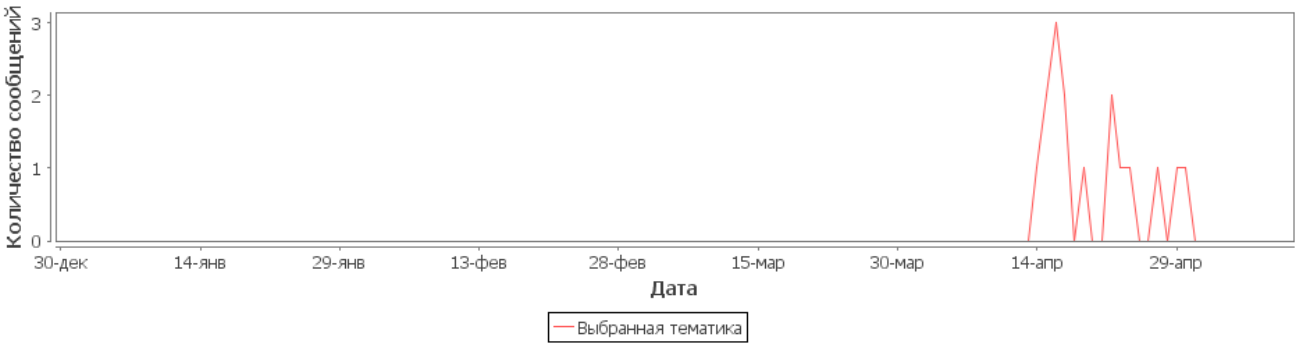
Форум ИТ-2020



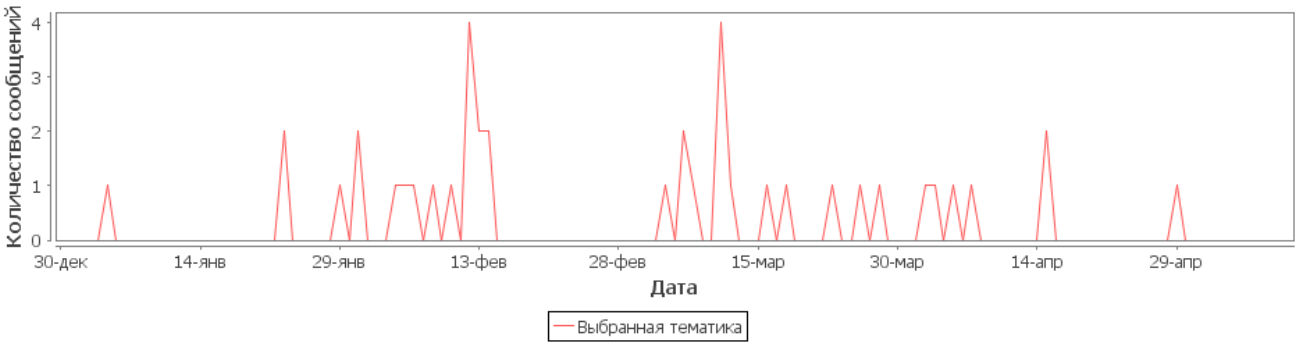
Челябинский метеорит



Чемпионат КХЛ



Эпидемия птичьего гриппа



Ядерная программа КНДР