УДК 519.688 Вестник СПбГУ. Прикладная математика... 2017. Т. 13. Вып. 3

*Е. А. Малютин, Д. Ю. Бугайченко, А. Н. Мишенин*

# ВЫДЕЛЕНИЕ ТЕКСТОВЫХ ТРЕНДОВ В СОЦИАЛЬНОЙ СЕТИ ОК

Санкт-Петербургский государственный университет, Российская Федерация,
199034, Санкт-Петербург, Университетская наб., 7–9

Социальные сети все чаще выступают не только как средство досуга или развлечения, но и как канал распространения информации, заменяя собой традиционные СМИ. В данной статье представлена модель масштабируемой системы выделения текстовых трендов, реализованная в социальной сети ОК. Акторы (пользователи и коммьюнити) совместно конструируют широкую новостную повестку, которая обладает определенной спецификой:

- текст написан пользователями, а не профессиональными журналистами, что усложняет его обработку;
- пользователи социальной сети генерируют текст на разных языках, что в классическом подходе к анализу медиапространства требует привлечения большого количества высокооплачиваемых специалистов;
- учитывая характер современного информационного пространства и время отклика социальной сети, необходима система, способная работать в режиме реального времени;
- социальные сети зачастую используются спамерами как площадка для продвижения и навязчивой рекламы, что требует привлечения дополнительных средств для фильтрации подобного контента.

Использование традиционных средств медиаанализа представляется крайне затруднительным, что естественным образом формирует запрос на разработку и внедрение программных средств детектирования и анализа текстовых трендов. В научной литературе при решении подобных задач предлагается использование одного из двух подходов: тематического моделирования с последующим анализом эволюции выделенных тем или построения дистрибутивных моделей, основанных на отслеживании частотных характеристик термов в корпусе. В статье приведен анализ существующих научных работ, основанных на обоих подходах с учетом специфики, предполагающей применение данной модели в рамках социальной сети. В результате было принято решение использовать дистрибутивную модель в качестве основы дальнейшей системы. ОК — одна из крупнейших социальных сетей на территории России и стран СНГ, акторы которой генерируют более 100M символов текста в день. Даже базовая обработка подобного потока информации является тяжелой технической задачей, так что при разработке необходимо прибегать к методам анализа «больших данных». Система детектирования трендов состоит из трех компонент:

- пакетный компонент, реализованный на основе фреймворка Apache Spark;
- потоковый компонент, реализованный на основе Apache Samza;
- mini-batch-компонент, реализованный на основе Spark Streaming.

В статье подробно описаны архитектура и технические особенности каждого из компонентов, приведены результаты работы системы, а также направления для дальнейшего исследования и развития. Библиогр. 13 назв. Ил. 7. Табл. 1.

*Ключевые слова*: анализ естественного языка, выделение трендов, большие данные.

*Малютин Евгений Алексеевич* — магистр; eugeny.malyutin@gmail.com
*Бугайченко Дмитрий Юрьевич* — кандидат физико-математических наук; dmitrybugaychenko@gmail.com
*Мишенин Алексей Николаевич* — старший преподаватель; alexey.mishenin@gmail.com

*Malyutin Evgeniy Alekseevich* — magister; eugeny.malyutin@gmail.com
*Bugaichenko Dmitriy Yurievich* — PhD of physical and mathematical science; dmitrybugaychenko@gmail.com
*Mishenin Alexey Nikolayevich* — senior teacher; alexey.mishenin@gmail.com



DOI: 10.21638/11701/spbu10.2017.308 

*E. A. Malyutin, D. Y. Bugaichenko, A. N. Mishenin*

## TEXTUAL TRENDS DETECTION AT OK

St. Petersburg State University, 7–9, Universitetskaya nab., St. Petersburg,
199034, Russian Federation

Social networks now serve not as a mere medium for entertainment, but as an information distribution channel that is replacing classical mass media. In this article we describe a scalable trend detection system implemented with the social network OK. Actors (users and communities) of social networks form a broad agenda. The content of social networks is specific:

- UGC (user generated content) is difficult to process;
- actors generate a multilingual text. This requires attracting a large number of highly paid professionals in the case of classical media analysis;
- modern social networks comprise a highly-connected society with high "response time". Therefore, the system must work in real time;
- social networks are used by spammers as a platform for promotion and obtrusive advertising, therefore the system should contain the ability to filter spam content.

Applying standard methods of media analysis to this seems impossible. It creates a natural demand for developing and implementing textual trend detection and analysis software. There are two main approaches of trend detection in academic papers: topic modeling (and further topics evolutionary analysis) and distributive models based on frequency-like properties of distinct terms. We conducted an analysis of scientific papers using both approaches taking into account the specific features of social networks. As a result of research, it was decided to use distributive models as a base for the system development. OK is one of the largest social networks in Russia and the CIS countries. Actors generate over 100M symbols of text every day. Even basic processing is a serious technical problem. So we are forced to use Big Data approaches through the development. We introduce lambda-architecture based on three main components:

- daily-batch processing component, based on Apache Spark;
- streaming processing component, based on Apache Samza;
- mini-batch processing component, based on Spark Streaming.

The article describes in detail the architecture and technical features of each component. In conclusion we present the results of operating the system as well as discuss areas for further research and development. Refs 13. Figs 7. Table 1.

*Keywords*: natural language processing, trend detection, big data.

**Введение.** В условиях современного информационного общества социальные сети, выступая как источник и канал информации, заменяют собой традиционные СМИ. Пользователи и сообщества совместно конструируют широкую новостную повестку крайне разнообразной тематики. Наличие подобного открытого источника формирует широкий спектр задач для специалистов в области анализа данных.

Одна из этих задач — детектирование и анализ текстовых трендов: явлений или событий, набирающих популярность и обсуждаемых пользователями. Методы медиаанализа и выделения трендов, применимые к традиционным СМИ, почти невозможно адаптировать к контенту, генерируемому акторами социальной сети по следующим причинам:

- текст написан пользователями, а не профессиональными журналистами, что усложняет обработку и извлечение информации;
- пользователи социальной сети генерируют текст на разных языках, что в классическом подходе к анализу медиапространства требует привлечения большого количества высокооплачиваемых специалистов;
- учитывая характер современного информационного пространства и время отклика социальной сети, необходима система, способная работать в режиме реального времени;

- социальные сети зачастую используются спамерами как площадка для продвижения и навязчивой рекламы. Для стабильной и надежной работы системы следует использовать механизмы текстовой дедупликации.

Стоит отметить, что даже базовая агрегация подобного новостного потока — сложная техническая задача. В данной работе представлена система обнаружения и анализа текстовых трендов в рамках социальной сети ОК. Кроме того, описаны средства и технологии, позволившие реализовать эту систему.

**Обзор литературы.** Несмотря на то, что имеется большое количество исследований, посвященных задачам обработки естественного языка (в том числе и для русского), работ по задаче детектирования и анализа текстовых трендов на русском не имеется. Однако на английском языке они есть. Их можно разделить на группы, предполагающие: 1) системы, базирующиеся на тематическом моделировании [1, 2] и дальнейшем анализе изменения выделенных тематик (тематические); 2) следующие модели, основывающиеся на различного рода статистических характеристиках отдельных термов (дистрибутивные) [3, 4].

Тематические модели не подходят для дальнейшего исследования по целому ряду причин:

- большая часть моделей предполагает априорное знание количества тем в корпусе;
- в исходном виде тематическое моделирование осуществляется без учета временного фактора, внедрение времени производится путем итеративного перестроения моделей и отслеживания «изменений» выделенных тем, однако рассматриваемые способы изучения эволюции тем не позволяют как-либо оценить ее «популярность»;
- подобные модели так или иначе предполагают некоторую дискретизацию по времени, не позволяя проводить анализ в режиме реального времени;
- сомнительные возможности для масштабирования.

Дистрибутивные модели, в отличие от тематических, лишены этих проблем. Они базируются на выделении «ключевых» слов из пространства словарей, основываясь на их «популярности», отличаясь лишь способами формализации «популярности» («трендовости») и ее расчета. Но и при таком подходе появляются некоторые проблемы:

- в рассмотренных работах используются усеченные наборы данных, не соотносимые с реальными объемами;
- нет работ, прямо указывающих на возможность реализации подобной системы в реальном окружении;
- зачастую обнаруженные тренды сложно интерпретировать напрямую.

**Постановка задачи.** Обычно, когда говорят о «трендовых» явлениях, имеются в виду самые популярные. Однако подобный «наивный» подход не совсем верен. Если смотреть на явления с точки зрения пользователя, его больше интересует не популярность явления, а актуальность, новизна. Таким образом, «трендовость» — это рост популярности некоторой темы, взвешенная в ее историческом контексте. Хорошим примером тренда являются профессиональные праздники («День сотрудника Полиции») или новостные события (например, взрывы в аэропорту в Брюсселе).

ОК — одна из крупнейших социальных сетей России и стран СНГ. Ежедневно в нее заходят около 40 млн пользователей, которые совместно с миллионом активных сообществ генерируют около 100М символов текста в день более чем на 15 языках (использовались тексты и описания медиаконтента (фото и видео) с открытых страниц

пользователей и сообществ). Необходимость обработки такого объема информации на постоянной основе в пакетном или потоковом режиме заставляет применять инструменты и технологии «больших данных» (Big Data).

Следовательно, необходима система детектирования и анализа текстовых трендов, функционирующая в режиме реального времени и способная обрабатывать объемы новостного потока ОК. Имеет смысл выделить такие требования:

- возможность предобработки большого корпуса текста;
- горизонтальная масштабируемость системы;
- работа в режиме реального времени;
- детектирование трендовых термов;
- агрегирование и сопоставление выделенных термов с существующими текстами.

**Теоретическая часть.** Когда речь заходит о трендах, «наивный подход» предполагает рассматривать самые популярные термы. Однако это не совсем правильно. Не учитывая семантической значимости выделенных термов, можно отметить, что конечный пользователь предпочитает увидеть не только популярные, значимые события, но и обладающие «новизной». Таким образом, для детекта трендов недостаточно рассматривать только популярные слова, но следует учитывать их популярность в некоторой исторической перспективе. Стоит также учитывать, что абсолютная популярность — характеристика, обладающая достаточно сильной «сезонной» составляющей: средняя популярность слова, не участвующего в тренде, может значительно меняться в зависимости от дня недели, времени суток и т. д. Для первичной оценки встречаемости терма более целесообразно использовать относительную частоту, абсолютное количество упоминаний терма, нормированное на общее количество слов в корпусе за обследуемый период, более формально:

1. Из постановки задачи предполагается, что с каждым документом, содержащимся в коллекции, ассоциировано время его создания. Разобьем коллекцию на эпохи, и будем в дальнейшем оценивать дистрибутивные характеристики терма в рамках конкретной эпохи. Пусть $N_i^j$ — количество упоминания терма $i$ в корпусе текста за эпоху $j$. Рассмотрим величину

$$f_i^j = \frac{N_i^j}{\sum\limits_i (N_i^j)},$$

где $\sum\limits_i (N_i^j)$ — общая длина корпуса. Величину $f_i^j$ будем называть *относительной частотой терма* $i$ в корпусе $j$.

2. Для каждого терма $i$ введем экспоненциально-взвешенное скользящее среднее ($EWMA$) и экспоненциально-взвешенную скользящую дисперсию ($EWMVar$). Для расчета воспользуемся рекуррентными формулами, представленными в [5]:

$$\begin{gathered} \Delta_i^j \triangleq f_i^j - EWMA_i^{(j-1)}, \\ EWMA_i^j = EWMA_i^{(j-1)} + \alpha \cdot \Delta_i^j, \\ EWMVar_i^j = (1-\alpha) \cdot (EWMVar_i^{j-1} + \alpha \cdot (\Delta_i^j)^2). \end{gathered} \tag{1}$$

Параметр $\alpha$ можно задать, используя время полураспада:

$$\alpha = 1 - e^{\frac{\ln(\frac{1}{2})}{t_{half}}},$$

где $t_{half}$ — временной интервал в единицах измерения, соответствующих единицам агрегации потока текста в эпохи. В рамках работы реальной системы для инициализации «0-й» эпохи предполагалось $EWMA_i^0 = EWMVar_i^0 = 0 \quad (\forall i)$.

3. Для оценки «важности» (далее $sig$ — от significance) конкретного терма воспользуемся методом $z$-$score$ (который даже при отсутствии нормальности распределения может быть полезной эвристикой) в введенных ранее определениях:

$$sig_i^j = \frac{f_i^j - EWMA_i^{(j-1)}}{\sqrt{EWMVar_i^{(j-1)}}}.$$

4. Однако при применении таких вычислений с реальными данными часто появляются сложности. С одной стороны, редко встречающийся терм, увиденный впервые и употребленный всего лишь несколько раз за день, приобретает достаточно высокое значение $sig$; с другой — достаточно часто возникает ситуация, когда $EWMVar$ оказывается достаточно близким к нулю, что может приводить к серьезным погрешностям при вычислении. Для решения этих проблем модифицируем приведенную выше формулу следующим образом:

$$sig_i^j = \frac{f_i^j - \max(EWMA_i^{(j-1)}, \beta)}{\sqrt{EWMVar_i^{(j-1)} + \beta}}. \tag{2}$$

Параметр $\beta$ используется как смещение, для того чтобы избежать нулевого знаменателя или погрешности вычисления, а также в качестве шумового фильтра. В действительности при случайных флуктуациях редко встречающегося терма, для которого $EWMA < \beta$, нельзя статистически достоверно гарантировать его «трендовость».

Формулы (1) напрямую не подходят для потоковой оценки $EWMA$ и подразумевают некоторую агрегацию для $X$. Рекомендуется брать достаточно крупный временной интервал для агрегации (выделения эпохи), для уменьшения «разброса» $X$. Стоит заметить, что такие источники данных как социальные сети имеют весьма явный суточный цикл, и в этом случае более осмысленным выглядит использование суточной агрегации новостного потока. Стоит также отметить, что в выражениях (1) применяется «предыдущее» значение $EWMA$, т. е. текущее значение относительной встречаемости сравнивается с историческими знаниями об общей популярности терма.

Несмотря на то, что выражение (2) позволяет выделить значимые, «трендовые» термы (как продемонстрировано на рис. 1), опираясь на известную статистику предыдущего дня, их прямая интерпретация, как видно из таблицы, остается достаточно трудоемкой.

Для конструирования набора «трендов» (в данном случае набора слов, характеризующего некоторое событие или явление) из набора термов применяется

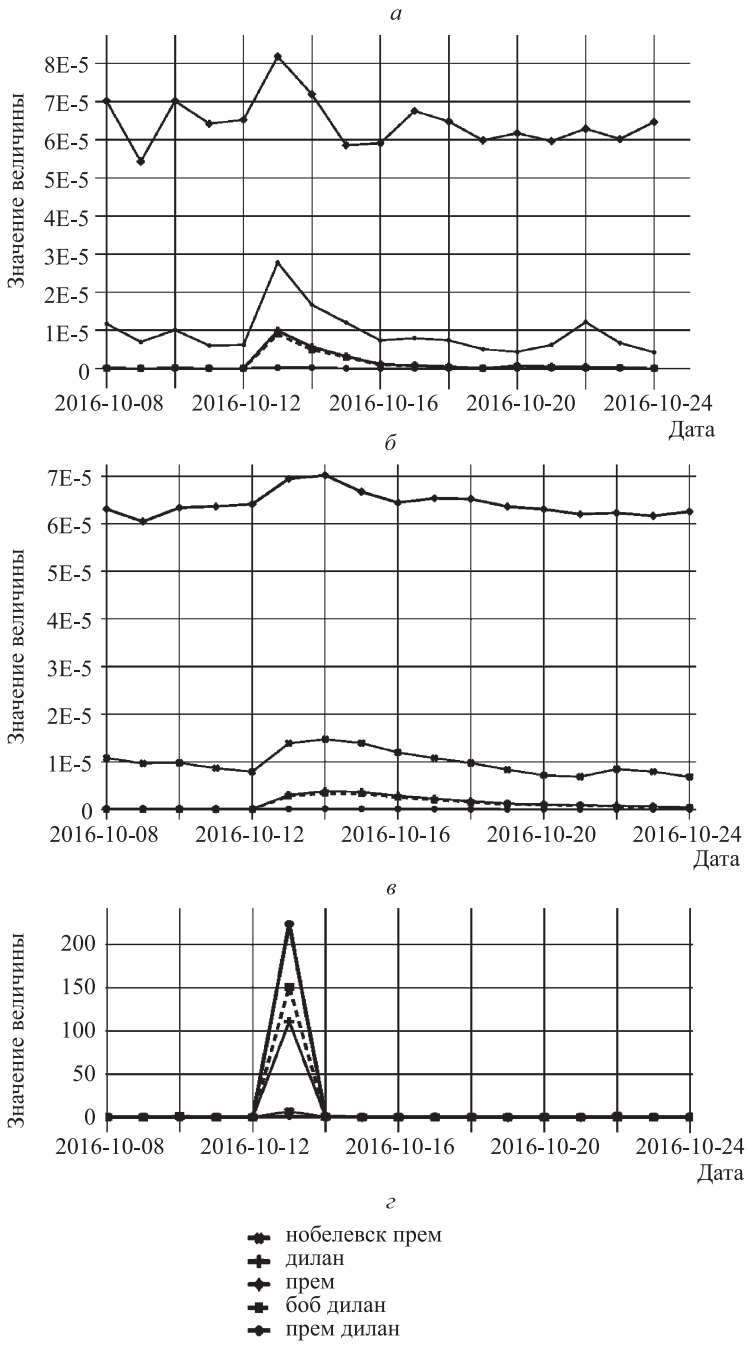


*Рис. 1.* Сравнительный график характеристик термов, имеющих отношение к присуждению Бобу Дилану Нобелевской премии

*а* — относительные частоты термов; *б* — $EWMA$; *в* — $EWMVar$; *г* — обозначения на графиках.

кластеризация на основе совстречаемости термов. В качестве меры похожести использовалась величина $npmi$ (нормированная взаимная информация):

$$npmi = \frac{\log(p(x,y)/p(x)\cdot p(y))}{-\log(p(x,y)},$$

где $p(x)$ — вероятность встретить терм $x$ в корпусе; $p(x,y)$ — вероятность встретить в одном документе термы $x$ и $y$.

**TOP-20 значимых термов за 12 октября 2016 г.**
**(слова и биграммы представлены в «сыром» виде, после стемминга)**

| Значимые термы | |
|---|---|
| наин ельцин | жертв ошибочн |
| вдов перв | михалков лжи |
| медведев установ | обвин михалков |
| тайн моисе | фигуристк медведев |
| моисе смотрет | мосул стал |
| пулеметчик всу | росс наин |
| ельцин обвин | фигуристк евген |
| успоко авак | повоня немн |
| авак останет | штаб иракск |
| серг микаэля | шкиряк украинц |

Для кластеризации использовался алгоритм DBSCAN [6] (рис. 2).

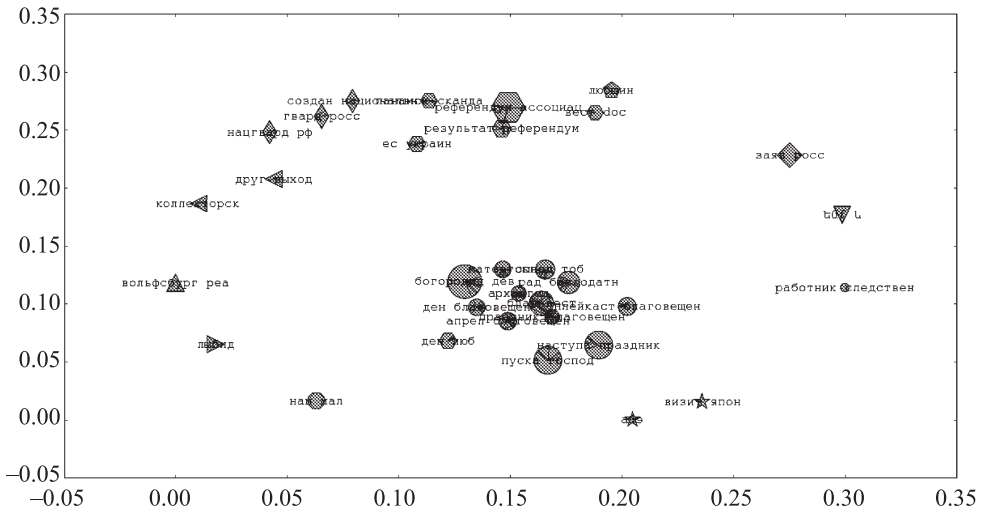


*Рис. 2.* Визуализация результатов кластеризации за 4 июня 2016 г.
Разными формами вершин обозначены различные кластеры. Размер вершины зависит от популярности терма. На осях приведены значения координат.

Изложенная выше модель хорошо подходит для пакетной обработки данных. Однако применение ее при потоковой обработке сопряжено с определенными трудностями. Одна из них — расчет относительной частоты в потоковом режиме. Действительно, представленная модель, основанная на частотных характеристиках, предполагает наличие какого-либо временного промежутка, в рамках которого производится агрегация. В связи с этим в потоковом режиме при расчете $sig(x)$ для каждого терма

использовалось экспоненциально-взвешенное среднее абсолютного числа встречаемости терма, нормированное на экспоненциально-взвешенное среднее объема корпуса.

Кроме того, результат работы потоковой части системы — отдельные слова и биграммы. И их все также сложно интерпретировать по отдельности. Подход, применяемый для пакетной обработки, в данном случае не подходит. С одной стороны, нет технической возможности держать весь объем текста в памяти и постоянно подсчитывать совстречаемость «трендовых» термов, с другой — методы потоковой кластеризации достаточно слабо развиты, а качество их работы находится на достаточно низком уровне. Для визуализации и агрегирования результатов работы потокового компонента следует обратиться к набирающей в последнее время популярность технике — mini-batch-анализу. В рамках mini-batch-системы поток данных дискретизируется по времени, путем объединения сообщений, пришедших за определенный (небольшой) интервал времени в массивы (пакеты, батчи), а расчеты и обработка происходят в рамках каждого доступного батча.

Таким образом, общая система детектирования трендов состоит из трех модулей: пакетного, отвечающего за точные ежедневные, но недостаточно актуальные расчеты, потокового, выполняющего роль актуальной, но упрощенной оценки, и mini-batch-модуля, необходимого для визуализации и агрегирования результатов работы потокового компонента и, фактически, отвечает за сервисный уровень.

Такое строение систем «big-data аналитики», состоящей из трех уровней: пакетного, потокового и сервисного, называется «lambda-архитектура».

**Практическая часть.**

***Пакетная обработка.*** С практической точки зрения построение описанной системы — сложная техническая задача. В качестве основной платформы для реализации пакетных компонентов был выбран Apache Spark — программный каркас с открытым исходным кодом для реализации распределенной обработки структурированных и неструктурированных данных, интегрированный в экосистему Hadoop. В отличие от классической схемы Hadoop Spark использует специализированные примитивы для рекурентной обработки в оперативной памяти без применения дисковых хранилищ, что позволяет получить значительный выигрыш в скорости работы для некоторого класса задач.

Архитектура компонента представлена на рис. 3.

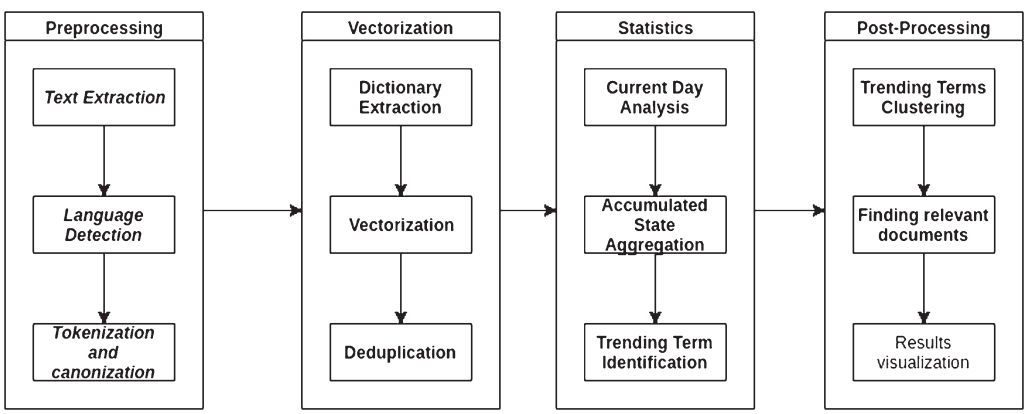


*Рис. 3.* Архитектура пакетного компонента

Кратко опишем каждый этап:

• Text Extraction: в качестве системы для транспортировки логов используется Apache Kafka. Данные поступают в Spark в виде «сырых» JSON-логов. Извлечение данных происходит с помощью средств Apache Spark. Применяется начальная фильтрация по длине текста, типу объекта (пост, видео, фото, комментарии) и количеству лайков на момент экстракции.

• Language Detection: производится на основе open-source-библиотеки [7]. Детектирование языка основано на «top trigram distribution». Добавлены дополнительные модули [8] для детектирования языков стран СНГ (азербайджанский, армянский, грузинский, казахский и т. д.). Стоит отметить, что знание об априорном распределении языков достаточно сильно влияет на качество работы алгоритма.

• Tokenization and canonization: разделение текста на токены-термы и их стемминг производятся на основе Apache Lucene. В составе Lucene имеются готовые профили для 23 языков (включая, например, русский, английский, армянский, латвийский), однако нет профилей для многих языков стран СНГ.

• Dictionary Extraction: извлечение словаря средствами Apache Spark. Средний словарь на эпоху (день) содержит 1M термов. Сохраняется индекс слов предыдущего дня.

• Vectorization: преобразование текстов в модель Bag-Of-Words.

• Deduplication: удаление дубликатов. Основано на методе случайных бинарных проекций [9]. Используется случайный базис — 18-битный хэш, 50% разреженности, в качестве меры похожести для документов — косинусное расстояние.

• Current Day Statistics: подсчет относительных частот токенов и биграмм. Фильтрация по частоте (шумовой порог). Стоит отметить, что следует применять разные пороги для термов и биграмм.

• Accumulated State Aggregation: подсчет $EWMA$ и $EWMVar$ для термов.

• Trending Term Identification: расчет значимости для термов. Вводится дополнительный шумовой фильтр.

• Trending Term Clustering: кластеризация термов на основе $npmi$, используются DBSCAN, имплементация ELKI [10].

• Finding Relevant Document: для каждого документа находятся релевантные документы на основе процента термов из кластера в составе документа. Для каждого кластера-тренда выбирается список наиболее рейтинговых (лайки) документов. Производится подсчет уникальных авторов/групп/IP для спам-фильтрации.

• Results Visualization: визуализация результатов на основе геолокации, текстов и трендов-кластеров с возможностью навигации по датам (пример см. на рис. 4).

***Потоковая обработка.*** Для потоковой обработки такого объемного массива информации в качестве транспортной системы использовалась Apache Kafka, в качестве фреймворка для вычислений — Apache Samza. Apache Kafka — распределенный программный брокер сообщений, который обладает высокой пропускной способностью, легко горизонтально масштабируется, имеется возможность временного хранения данных в HDFS для пакетной обработки [11].

Apache Samza — фреймворк для распределенной обработки потоков данных, интегрированный в экосистему Hadoop. Обладает встроенными механизмами сохранения состояния и восстановления в случае падения системы [12]. Системой для транспорта сообщений служит Apache Kafka.

Схема работы потокового компонента приведена на рис. 5.


*Рис. 4.* Визуализация результатов батчевой обработки

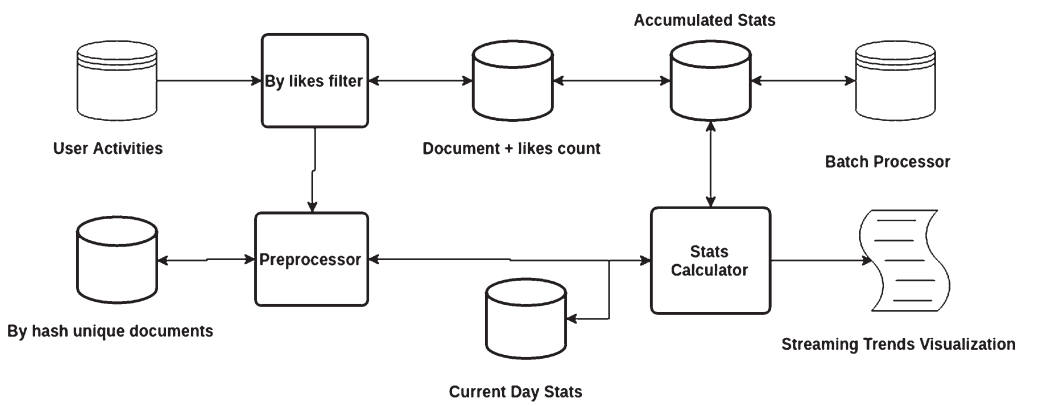


*Рис. 5.* Схема работы потокового компонента

**Mini-batch-компонент.** Для реализации mini-batch-обработки был использован Spark Streaming — компонент Apache Spark, в качестве средства для визуализации — Apache Zeppelin [13].

Apache Zeppelin — проект с открытым исходным кодом для быстрого прототипирования big-data-компонент, реализованный в виде интегрированной среды с возможностью применения Apache Spark, Spark Sql, HTML/CSS/Javascript/Angular и многих других интерпретеров для обработки и визуализации данных. Схема и результат работы компонента представлены на рис. 6 и 7.

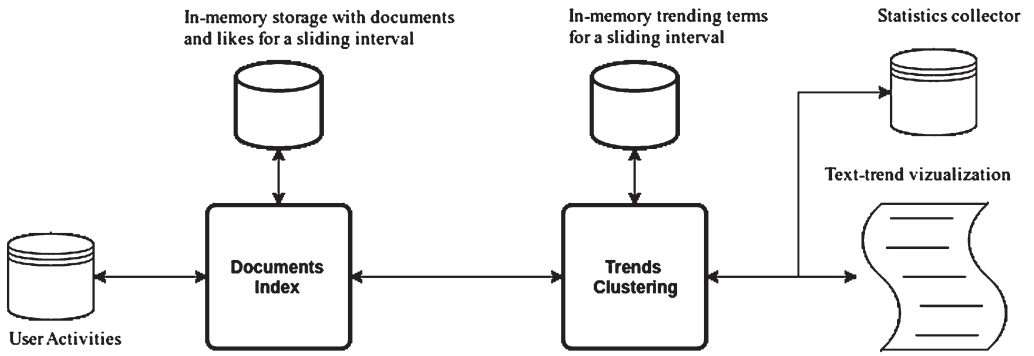


*Рис. 6.* Схема работы mini-batch-компонента

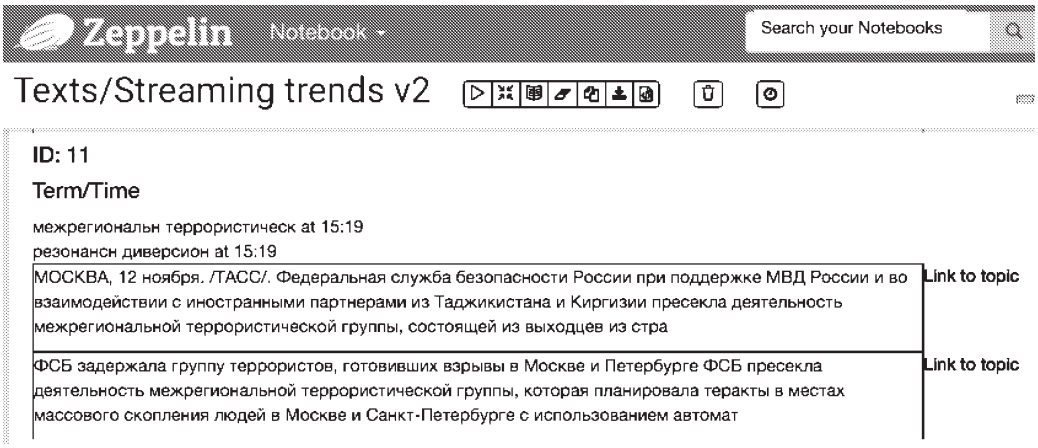


*Рис. 7.* Результат работы mini-batch-компонента

**Заключение.** В рамках проделанной работы была предложена дистрибутивная модель детектирования трендов. Кроме того, был реализован стек препроцессинга и анализа текста на основе распространенного фреймворка Spark. Модель показала свою работоспособность в рамках ОК. На ее основе была реализована big-data-система анализа данных с использованием современных фреймворков, включенных в общую lambda-архитектуру, которая была интегрирована в общую инфраструктуру ОК.

Результаты работы показывают успешность дистрибутивных моделей для задачи детектирования и анализа трендов. В действительности эти модели отличаются простотой математического аппарата и легкостью для горизонтального масштабирования, что позволило без особых трудностей как переложить изначально пакетный алгоритм на потоковый способ обработки данных, так и реализовать данную систему на стандартных и общепринятых фреймворках.

Описанная система имеет большой потенциал и возможности для приложений: таргетирование новостных событий с учетом интересов пользователя, улучшение ранжирования контента с учетом его актуальности, формирование выборок и дайджестов медиасообществ портала и т. д. В будущем планируется работа над развитием и расширением такой системы.

## Литература


1. *Lau J. H., Collier N., Baldwin T.* On-line trend analysis with topic models: twitter trends detection topic model online // Proceedings of COLING: technical papers. Mumbai, 2012. P. 1519–1534.

2. *Ahmed A., Xing E. P.* Timeline: A dynamic hierarchical dirichlet process model for recovering birth/death and evolution of topics in text stream // Proceedings of the Twenty-Sixth Conference. Conference on Uncertainty in Artificial Intelligence. 2010. Vol. 20. P. 29.

3. *Schubert E., Weiler M., Kriegel H.-P.* Signitrend: scalable detection of emerging topics in textual streams by hashed significance thresholds // Proceedings of the 20th ACM SIGKDD International conference on Knowledge discovery and data mining. 2014. P. 871–880.

4. *Cvijikj I. P., Michahelles F.* Monitoring trends on facebook // Dependable, Autonomic and Secure Computing (DASC), 2011. IEEE Ninth Intern. Conference on. 2011. P. 895–902.

5. *Finch T.* Incremental calculation of weighted mean and variance: technical report. Cambridge, 2009. Vol. 4.

6. *Ester M., Kriegel H.-P., Sander J., Xu X.* A density-based algorithm for discovering clusters in large spatial databases with noise // Kdd. 1996. Vol. 96, N 34. P. 226–231.

7. Open-source library for language detection. URL: https://github.com/optimaize/language-detector (accessed: 26.02.2017).

8. Additional language profile for CIS-languages. URL: https://github.com/denniean/language_profiles (accessed: 26.02.2017).

9. *Jeffrey D., Ullman Anand Rajaraman, Jure Leskovec.* Mining of massive datasets, 2013. URL: http://infolab.stanford.edu/~ullman/mmds.html (accessed: 26.02.2017).

10. Open-source library for data analysis. URL: https://elki-project.github.io/ (accessed: 26.02.2017).

11. Scalable stream processing platform. URL: https://kafka.apache.org/ (accessed: 26.02.2017).

12. Apache Samza: distributed stream processing framework. URL: http://samza.apache.org/ (accessed: 26.02.2017).

13. Apache Zeppelin: web-dashboard for interactive data analysis. URL: https://zeppelin.apache.org/ (accessed: 26.02.2017).



**Для цитирования:** *Малютин Е. А., Бугайченко Д. Ю., Мишенин* А. Н. Выделение текстовых трендов в социальной сети ОК // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. 2017. Т. 13. Вып. 3. С. 313–325. DOI: 10.21638/11701/spbu10.2017.308


## References


1. Lau J. H., Collier N., Baldwin T. On-line trend analysis with topic models: twitter trends detection topic model online. *Proceedings of COLING*: Technical Papers. Mumbai, 2012, pp. 1519–1534.

2. Ahmed A., Xing E. P. Timeline: A dynamic hierarchical dirichlet process model for recovering birth/death and evolution of topics in text stream. *Proceedings of the Twenty-Sixth Conference. Conference on Uncertainty in Artificial Intelligence*, 2010, iss. 20, p. 29.

3. Schubert E., Weiler M., Kriegel H.-P. Signitrend: scalable detection of emerging topics in textual streams by hashed significance thresholds. *Proceedings of the 20th ACM SIGKDD International conference on Knowledge discovery and data mining*, 2014, pp. 871–880.

4. Cvijikj I. P., Michahelles F. Monitoring trends on facebook. *Dependable, Autonomic and Secure Computing* (*DASC*), *IEEE Ninth International Conference on.*, 2011, pp. 895–902.

5. Finch T. *Incremental calculation of weighted mean and variance.* Technical report. Cambridge, 2009, vol. 4.

6. Ester M., Kriegel H.-P., Sander J., Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd*, 1996, vol. 96, no. 34, pp. 226–231.

7. *Open-source library for language detection.* Available at: https://github.com/optimaize/language-detector (accessed: 26.02.2017).

8. *Additional language profile for CIS-languages.* Available at: https://github.com/denniean/language_profiles (accessed: 26.02.2017).

9. Jeffrey D., Ullman Anand Rajaraman, Jure Leskovec. *Mining of massive datasets, 2013.* Available at: http://infolab.stanford.edu/~ullman/mmds.html (accessed: 26.02.2017).

10. *Open-source library for data analysis.* Available at: https://elki-project.github.io/ (accessed: 26.02.2017).

11. *Scalable stream processing platform.* Available at: https://kafka.apache.org/ (accessed: 26.02.2017).

12. *Apache Samza: distributed stream processing framework.* Available at: http://samza.apache.org/ (accessed: 26.02.2017).

13. *Apache Zeppelin: web-dashboard for interactive data analysis.* Available at: https://zeppelin.apache.org/ (accessed: 26.02.2017).

**For citation:** Malyutin E. A., Bugaichenko D. Y., Mishenin A. N. Textual trends detection at OK. *Vestnik of Saint Petersburg University. Applied Mathematics. Computer Science. Control Processes*, 2017, vol. 13, iss. 3, pp. 313–325. DOI: 10.21638/11701/spbu10.2017.308

Статья рекомендована к печати проф. В. Д. Добрыниным.

Статья поступила в редакцию 5 марта 2017 г.

Статья принята к печати 8 июня 2017 г.