Санкт-Петербургский государственный университет
Факультет прикладной математики - процессов управления
Кафедра технологии программирования

Власов Игорь Алексеевич

Выпускная квалификационная работа бакалавра

Отслеживание взаимосвязей событий в новостном потоке

Направление 01.03.02
Прикладная математика и информатика

Научный руководитель:
старший преподаватель,
кафедра технологии программирования
Стученков Александр Борисович

Санкт-Петербург
2019 г.

Содержание

Введение

Несомненно, отличительной особенностью современного мира является постоянно увеличивающийся поток информации, непрерывно поступающей из разных источников. Одной из важных частей этого являются новости, которые приходит из разных источников, включая как традиционные СМИ, например, газеты, радио и телевидение, так и современные источники, такие как различные новостные сайты и социальные медиа. Отличительной особенностью является то, что в случае с социальными сетями авторами новостной повестки дня служат сами пользователи. Однако, парадоксальность ситуации заключается в том, что рост количества информации ведет к затруднению ее использования и снижению общего уровня информированности. Ведь увеличение темпов производства информации ведет, к так называемому, информационному шуму. В подобной ситуации возникает необходимость структурирования информации. Обобщение больших информационных потоков, которые непрерывно генерируются в средствах масс-медиа, требует новых подходов к их обработке. Кроме этого, есть еще несколько причин для развития данной области:

- получение новых знаний по определенному новостному событию;
- необходимость систематизации и упорядочивания знаний;
- акцентирование внимания на некоторых аспектах про какое-либо происшествие;
- представление информации в более наглядном и понятном виде.

Методы структурирования информации разнообразны. Причиной этого является множество способов ее представления и организации. В зависимости от целей, применяются различные технологии и методы структурирования.[13] Целью структурирования данных является выделение ключевых элементов из массива информации, а также логики взаимосвязанности этих элементов. Результатом такого упрощения является удобство получения и обработки информации конечным пользователем. Сложно отрицать, что современные технологии все больше замещают традиционные средства мас-

совой информации. На видеохостингах люди могут в любой момент посмотреть практически любые интересующие их сюжеты, репортажи и фильмы на абсолютно любые темы. Интернет издания агрегируют самые свежие новости текущего дня. К тому же, пользователи теперь не только потребляют информацию, но и сами становятся авторами. Людям почти каждый день необходимо обрабатывать множество новостных заметок в интернет СМИ и постов в социальных сетях.[14] Для облегчения обработки и улучшения усваивания информации существует необходимость в улучшении её структуры, чему и способствуют, в том числе, решения задач по анализу данных

Постановка задачи

Цель работы заключается в разработке системы по определению наличия взаимосвязей между событиями в информационном потоке для новостной ленты в социальной сети ‘Одноклассники’. Для реализации поставленной задачи требуется построить математическую модель, основываясь на семантической близости ’трендовых новостей’ и временном интервале между днями, в которых были выделены данные события. А также необходимо реализовать разработанный метод программными возможностями языка Python. Основываясь на поставленной задаче можно выделить следующие основополагающие моменты, решение которых необходимо найти для достижения поставленной цели. Во-первых, необходимо собрать, обработать и кластеризовать сырые данные из открытых групп и сообществ в социальной сети. А также выделить самые ‘трендовые’ события для каждой эпохи. Под эпохой подразумевается некий ограниченный промежуток времени, в рамках которого будут агрегироваться и обрабатываться события независимо от данных за остальное время, но учитывая результаты обработки за прошедшее время. В работе используется результат работы алгоритма по выделению трендов внутри социальной сети ‘ОК’.[9] А именно датасет, содержащий выборку по публикациям, а также самые ‘интересные’ и обсуждаемые новости с некоторыми метриками. Резюмируя, в рамках данной задачи по анализу данных можно выделить основные этапы для построения модели и реализации программной части, опираясь на имеющийся набор данных.:

- Выбор основной модели представления данных для дальнейшей обработки;
- Выбор функции для улучшения качества результатов системы с учетом временного фактора;
- Выбор метода удаления слабых зависимостей;
- Выбор программного обеспечения для наиболее удобной реализации построенной модели;

- Анализ полученных результатов;
- Планы для дальнейших исследований.

В итоге, решение этих вопросов позволит получить готовую систему для построения взаимосвязей между трендовыми событиями в новостном потоке.

Обзор литературы

В данном разделе представлен краткий обзор работ, которые послужили основой данному исследованию.

‘Finding and Linking Incidents in News’, Ao Feng and James Allan, 2007

В этом исследовании представлены реализация и сравнительный анализ трех методов для построения моделей системы обработки и отслеживания инцидентов.[2]

‘Event Threading within News Topics’, Ramesh Nallapati, Ao Feng, Fuchun Peng, James Allan, 2004[7]

В статье описываются несколько методов кластеризации новостей по событиям. Также представлены подходы для построения зависимостей между событиями для задачи тематического моделирования.

‘Исследование лексического метода вычисления схожести строк с учетом предварительнои обработки’, Н.В. Неелова, 2009[10]

В исследовании представлены результаты сравнения эффективности алгоритма, который основан на лексическом сравнении слов с использованием online-метода Джаккарда. Модель дополняется различными функциями для определения дубликатов и синонимов.

‘Method for measuring the semantic-similarity of textual documents’, Bermudez Soto Jose Gregorio, 2017[3]

Публикация содержит метод сравнения и сопоставления текстов. Работа основана на сравнении тестов на уровне представления текстовых пассажей.

‘Textual trends detection at ok’, E. A. Malyutin, D. Y. Bugaichenko, A. N. Mishenin, 2017[9]

В статье описана разработанная масштабируемая система для задачи детектирования и анализа трендов в социальной сети ‘Одноклассники’. В работе приведены архитектура и технические особенности компонентов, на основе которых была сконструирована дистрибутивная модель для анализа трендов.

‘Discovering Event Evolution Graphs From News Corpora’ Christopher C. Yang, Xiaodong Shi, and Chih-Ping Wei, 2009[4]

Статья освещает разработку метода для построения системы, целью которой будет отслеживание и выявление эволюции отношений между новостными событиями.

‘SigniTrend: Scalable Detection of Emerging Topics in Textual Streams by Hashed Significance Thresholds’, Erich Schubert, Michael Weiler, Hans-Peter Kriegel[5]

В работе описано использование экспоненциально-взвешенное скользящее среднее и экспоненциально-взвешенную скользящую дисперсию для выделения трендов. А также представлен метод масштабирования системы в условиях ограниченной памяти. [5]

Глава 1. Существующие решения

1.1 Общие концепции

Основные методы решения задач по обработке естественного языка[6] для анализа текстовых данных можно можно отнести к одной из двух категорий. Первая, это модели и системы, которые основаны на тематическом моделировании и дальнейшем анализе самих тем и их распределении. Второй же концепцией можно считать дистрибутивные методы, в основе которых лежит анализ с помощью различных статистических характеристик термов и биграм, например, с использованием разнообразных частотных метрик.

1.2 Тематическое моделирование

Задачей тематического моделирования является построение модели, дающей наилучший результат. Под тематической моделью подразумевается некая модель для текстовых данных, целью которой служит определение степени принадлежности каждого документа к определенной тематике. Результатом построения тематической модели может быть как непосредственно выявление множества тем, так и решение различных дополнительных задач, таких как:

- ранжирование документов по степени релевантности заданной теме (тематический поиск)
- ранжирование документов по степени тематического сходства с заданным документом или его фрагментом
- построение иерархического тематического каталога коллекций документов и выработка правила каталогизации новых документов
- определениеизменений темы со временем (предполагается, что для каждого документа известно время его создания)
- определение тематики для авторов (предполагается, что для каждого документа известен список второв)

- определение тематики различных сущностей (entities), связанных с документами (например, журналов, конференций, организаций, стран)
- разбиение документа на тематически однородные фрагменты.

Методы на основе построения тематических моделей являются решениями задач "мягкой"кластеризации. Это означает, что каждый из документов может принадлежать нескольким темам с той ли иной степенью точности.

Для построение вероятностной тематической модели принимается во внимаение несколько следующих предположений:

- Не имеет значения расположение документов в коллекции, а точнее их порядок;
- Не имеет значение порядок слов в документе, а сам документ представлется в виде вешка слов("bug-of-words");
- Частоупотребляемы слова и предлоги(союзы), а именно термы встречающиеся во многхи документах не имеют значения для определения тематики;
- Форма слова не влияет на его значение, а значит разные формы являются одним и тем же словом;
- Весь корпус можно представить в виде выборки пар 'документ-слово', (d,w), где $d \in D$, $w \in W$;
- Каждая тема $t \in T$ описывается неизвестным распределением $p(w|t)$ на множестве слов $w \in W$;
- Каждый документ $d \in D$ описывается неизвестным распределением p(t|d) на множестве тем $t \in T$;
- гипотеза условной независимости: $p(w|t, d) = p(w|t)$.

1.2.1 Вероятностный латентный семантический анализ

Развитие вероятностого тематического моделирования начинается с работы Томасом Хофманна 'Вероятностный латентный семантический анализ (probabilistic latent semantic analysis, PLSA)'. В ней каждая тема описывается вероятностным распределнием на множестве слов. А сама вероятностная модель для пар 'документ-слово' ($p(d, w)$) может быть представлена одним из способов:

$$\sum_{t \in T} p(t)p(w|t)p(d|t)$$

$$\sum_{t \in T} p(d)p(w|t)p(t|d)$$

$$\sum_{t \in T} p(w)p(t|w)p(d|t)$$

где T это все множество тем,
p(t)-неизвестное априорное распределение тем во всей коллекции
p(d)-априорное распределение на множестве документов, $p(d) = n_d/n$ при этом n это общая длина всех документов
p(t)- априорное распределение на множестве слов, $p(w) = n_w/n$, где n_w количество появлений терма во всех документы

1.2.2 Латентное размещение Дирихле

Одной из самых известных моделей тематического моделирования является 'Метод латентного размещения Дирихле (LDA)', который был предложен Дэвидом Блеем. В нем вероятностная модель осталась такой же как и в работе Хофманна, но с рядом дополнительных условий, которые помогают устранить некоторые недостатки модели PLSA. Первым предположением является то, что все векторы документов порождаются одним и тем же вероятностным распределением на нормированных |T|-мерных векторах. Это распределение из параметрического распределения семейства Дирихле. Еще одно предположене гласит, что все векторы тем также порождаются одним и тем же вероятностным распределением на

нормированных |W|-мерных веторах. В этом условии распределение также берется из параметрического семейства распределений Дирихле.
По результатам исследований обе модели PLSA и LDA показывают сопоставимое качество результатов на больших корпусах текста.

1.3 Дистрибутивная семантика

Областью исследования дистрибутивной семантики является вычисление семантической близости различных лексических единиц, основываясь на их распределении в большом количестве текстовых корпусов. Работы в данной области опираются на главную гипотезу дистрибутивной семантики, которая утверждает, что если лингвистические единицы встречатся в схожих контекстах, то они имеют близкое значение.[8] Основополагающей единицей в этих моделях служит контекстный вектор. Впервые идея использования такой конструкции был предложен Ч.Осгудом в рамках работ по представлению значений слов. А сам термин был введен С.Галлантом для описания смысла слов и разрешения лексической неоднозначности.
Спектр возможностей применения моделей дистрибутивной семантики достаточно широк, ниже приведены примеры основных задач:

- Определение семантической близости между различными лексическими конструкциями;
- Кластеризация по результам оценки семанической близости;
- Разрешение вопросов смысловой неоднозначности;
- Расширение поисковых запросов путем определния нахождения ассоциативных связей;
- Определение тематики документов;
- Определение тональности высказываний, заметок, статей;
- Возможность моделирования перифраз.

Активное развитие дистрибутивной семантики привело к большому количеству моделей, которые имеют различия по некоторым составляющим, например:

- количественная оценка частоты лексических конструкций относительно контекста
- мера, используемая для оценки расстояния между векторами
- подходы к уменьшению размерности матрицы.

1.3.1 Латентно-семантический анализ (ЛСА)

Одной из основных моделей в данном подходе является метод Латентно-семантический анализа (англ. Latent semantic analysis, LSA). Подход заключается в анализе взаимосвязей между корпусом документов и термами, встречающимися в них, а также выделении особенностей, присущих каждому элементу из опредленной тематики. В основе метода лежит принцип выделения латентных зависимостей между изучаемыми объектами. Подвидами решения задач с помощью ЛСА можно назвать:

- Сравнение лексических конструкций
- Сравнение документов
- Сравнение документов и термов

В качестве отправной точки метод использует матрицу 'термин-документ', которая содержит в ячейках информацию о частотности трема для какого-либо документа. В качестве весовой функции для слов в матрице служит TF-IDF.[2] Это такая статистическая мера, которая оценивает важность слова для конкретного документа относительно остального корпуса текстов. Как следует из названия, эта характеристика состоит из двух частей. TF (term frequency — частота слова) — отношение числа вхождений некоторого слова к общему числу слов документа. Таким образом, возможно оценить важность слова t_i для отдельного документа.

$$tf(t,d) = \frac{n_t}{\sum_k n_k},$$

где n_t число вхождений терма t в документе d. А сумма в знаменателе это общее число слов для документа d.

IDF (inverse document frequency — обратная частота документа) - это инвертированная частота появления слова во всех докумнтах корпуса. Смысловая суть этого множителя заключается в уменьшении значимости общеупотребительных слов.

$$idf(t, D) = \log \frac{|D|}{|d_i| \in D | t_i \in T},$$

где в числителе число документов в коллекции, а в знаменателе число документов из корпуса, в которых встречался термин t. Важное замечание, что основание логарифма в этой величине приводит лишь к пропорциональному изменению весов для каждого терма.

В итоге, по результатам вычисления этой характеристики можно выявить термины, наиболее часто употребляемые в данном документе, относительно остального корпуса. А конечная формула имеет вид:

$$tf - idf(t, d, D) = tf(t, d) \times idf(t, D)$$

Глава 2. Построение математической модели

2.1 Основные понятия

Прежде чем приступить к решению основной задачи необходимо дать определение нескольким терминам, которые будут использоваться в дальнейшем.

- Story - новостная статья или пост в социальной сети о каком-либо событии. В данной работе подразумевается, что в одной story повествуется об одном событии.
- Event - некоторое событие или новостной повод, о котором идет речь в Story. При этом учитывается тот факт, что одно событие может быть освещено в нескольких story. Иначе говоря, несколько Story рассказывают о случившемся в одном Event.
- Topic - тема или несколько объединенных подтем, в рамках которых могли произойти некоторые связанные между собой event.
- Event threading - поток событий, связанные события. В контексте работы подразумевается взаимосвязь между event. А если быть точнее, то цепочка из связанных между собой event внутри Topic.
- Эпоха - заранее определенный временной интервал.

2.2 Критерии наличия взаимосвязей

Следующим пунктом обозначим несколько критериев, которым должны удовлетворять взаимосвязи в построенной системе событий.

- Связанные event должны быть последовательны.
- Оба event имеют семантическую близость.
- Отсутствие промежуточных звеньев между event.

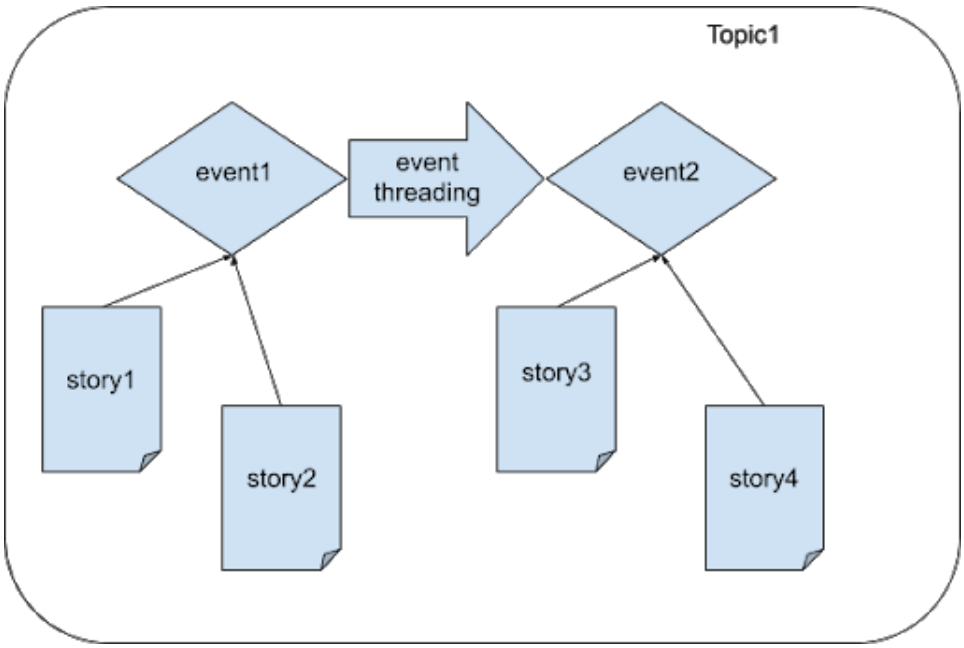

Рис. 1: Визуализация модели.

Теперь давайте более детально поясним, что означают условия и обоснуем необходимость для каждого из критериев.

Начнем с первого требования. Одно из событий должно предшествовать другому. Это означает, что events не могут располагаться в одной исторической эпохе. Другими словами статьи, входящие в одно событие не должны располагаться в одном дне. Это обусловлено архитектурой построения кластеров[9], в которой story кластеризуются в независимых эпохах. Эпоха в контексте работы подразумевается равной одному дню. Данное условие имеет место быть, поскольку схожие story внутри одной эпохи, с большей долей вероятности, будут уже относиться к одному Event.

Наличие семантичекой близости легко объяснить необходимым условием наличия схожего словарного запаса. Данный критерий отвечает за схожую смысловую составляющую. Ведь если в двух или более event речь идет про одно и тоже событие или происшествие, то с большей долей вероятности в них будут содержаться одинаковые термы, биграммы. Этот подход основывается на предположении дистрибутивной семантики, в которой статистические значения в документах являются основополагающей частью в определении их семантической близости.

Отсутствие промежуточных звеньев при построении ребра треда. Этот

критерии проще всего объяснить на примере. Предположим у нас есть три события, которые расположены в последовательных эпохах: event1, event2, event3. И предполагается, что имеются взаимосвязи event1-event2 и event2-event3, тогда построение треда из event1 в event3 будет нецелесообразным. Условие является дополнительным ограничением на систему. Цель у этого дополнения заключается в создании более удобной для понимания и дальнейшйего использования системы event threading.

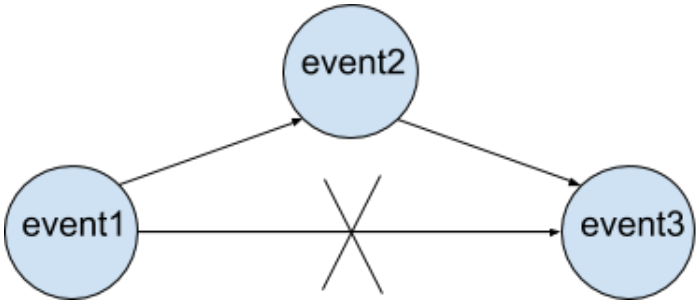

Рис. 2: Threading.

2.3 Векторные модели

Гипотеза о дистрибутивной семантике гласит, что лингвистические единицы, встречающиеся в схожих контекстах, имеют близкие значения[8]. При этом подходе основной идеей для моделирования является векторная модель семантики(VSM). Это означает, что каждый документ из коллекции, а в нашем случае story, которые входят в event, представляются в виде точки(вектора) в многомерном пространстве. Согласно предположению, близко лежащие друг к другу точки соответствуют семантически схожим документам.

Существует три основных подхода к оценкам схожести и построению векторных моделей семантики.

- Сходство документов
- Сходство слов
- Сходство отношений

Самой распространенной моделью для поиска схожих документов является матрица 'термин-документ'. В ней каждая строка определеяет отдельный термин, а каждый столбец соотнесен документу. В данном подходе документ представляется в виде мешка слов(bag of words), что говорит нам о неважности порядка вхождения каждого терма, но крайней информативности его количества появлений. Допустим у нас есть матрица X 'термин-документ'. Если в выборке имеется n документов и m уникальных слов, то X будет иметь m строк и n столбцов. Если рассмотреть i-ое слово из всего словаря корпуса, и j-ый документ, то элемент x_{ij} матрицы X будет являться количеством употреблений терма w_i в документе d_j. В некоторых моделях используются различные весовые функции, например tf-idf.

Еще один вариант получения матрицы поиска схожих событий это матрица пара-модель. В ней каждая строка описывает заданную пару слов, например дровосек:дерево, строитель: дом. А для каждого столбца определено отношение, происходящие между схожими парами слов.

Наиболее подходящей для решения нашей задачи из VSM является модель слово-контекст. Она удобна для нахождения сходства не между целыми документами, а между его частями, такими как параграфы или предложения. Ведь согласно гипотезе, термы, встречающиеся в схожих контекстах стремятся иметь схожий смысл. Выбор обусловлен решением задачи для небольшой выборки термов/биграм из шортлиста кластера для каждого event внутри дня.

2.4 Функции сходства

При моделировании взаимосвязей между событиями основной составляющей будет являться функция нахождения расстояния между векторами, а согласно дистрибутивной гипотезе, это и будет являться оценкой семантической близости. Есть несколько способов вычисления этой характеристики[12], одним из самых известных является функция косинусного сходства, которая используется во многих работах по анализу данных[7].

$$\frac{\sum_{i=1}^{n} \alpha_i \times \beta_i}{\sqrt{\sum_{i=1}^{n} (\alpha_i)^2} \times \sqrt{\sum_{i=1}^{n} (\beta_i)^2}},$$

где α, β это вектора для которых расчитывается расстояние.

Еще одним способом определения схожести документов является коэффициент Жаккара, предложенный Полем Жаккаром в 1901. Данный подход основывается на нахождении количества общих слов или других конструкций, например словосочетаний. По факту коэффициент отражает меру пересечения для двух множеств с учетом их размерности. Если выражаться более формальным языком, то данная формула может выглядеть, например:

$$K_{ij} = \frac{c}{a + b - c}$$

, в которой a и b это количество уникальных терминов для каждого из документов, а c это количество терминов, которые встречаются в обоих множествах.

2.5 Временная зависимость

Анализируя данные о трендах в социальной сети нельзя не учитывать временной фактор. Ведь все события и происшествия имеют конкретную временную метку, а именно привязку ко дню публикации. Это учитывается в построении кластеров, а точнее event при выделении трендов.[9] Для упрощения вычислений и обобщения модели выделим скользящее 'окно' в несколько дней до и после даты публикации, в рамках которого будем рассматривать события. Вообще говоря, при наличии необходимых вычислительных мощностей окно можно определить практически любым значением дней. Тем самым достигается возможность масштабирования системы при необходимости. Теперь необходимо оценить вероятность связности событий по прошествию нескольких дней. В качестве основы воспользуемся некоторой затухающей функцией, решающей похожие задачи[4]. Немного адаптировав функцию в условиях необходимых условий и требований имеем:

$$\tau = exp^{\frac{2*(1-t)}{l}}$$

В данной модификации функция зависит от t, что является модулем количества дней между рассматриваемыми статьями, а l - это размер рассматриваемого окна, значение которого можно варировать при необходимости.

Ниже приведен пример временного коэффициента для окна размера 5:

Таблица 1: Значения временного коэффициента при рассматриваемом окне в 5 дней

t	τ
1	1
2	0.819
3	0.67
4	0.549
5	0.449

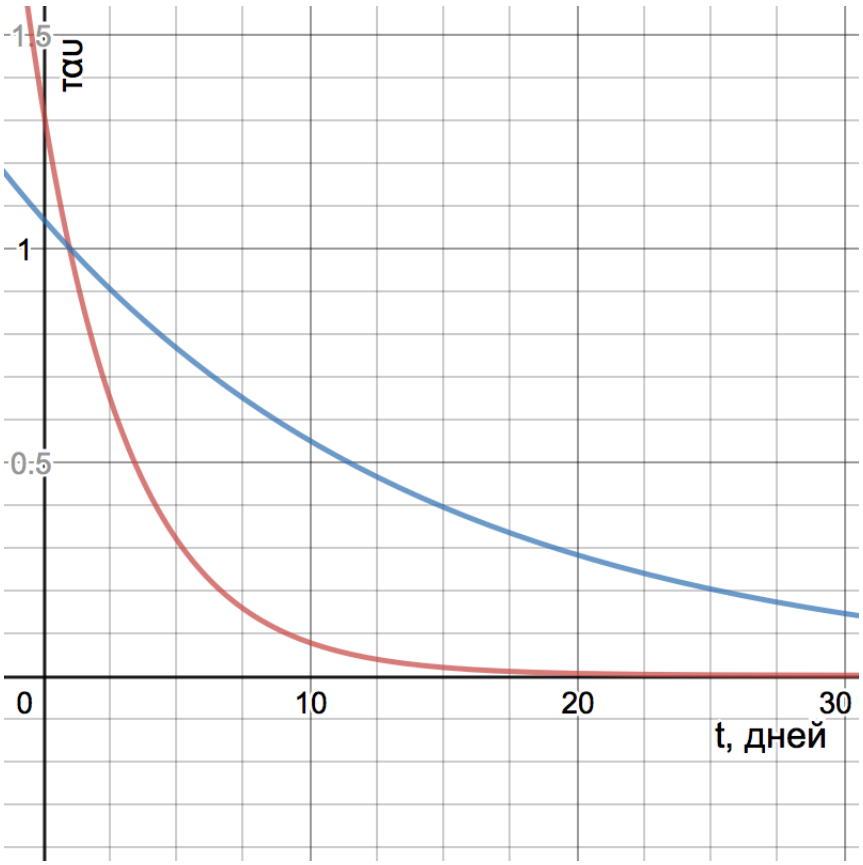

Рис. 3: Сравнение измения значения для разнх значений окна

На рисунке 3 наглядно изображено сравнение результатов функции τ для окон размером в неделю(7 дней) и в месяц(30 дней). Красным на графике выделен результат для функции при выборе 'недельного' окна

$\tau = exp^{\frac{2*(1-t)}{7}}$, в то время как синим отмечен результат для окна длиной в месяц $\tau = exp^{\frac{2*(1-t)}{30}}$ Можно заметить, что при большем количестве дней, входящих в рассматривемое окно, результаты для сосдених дней будут отличаться все меньше.

2.6 Удаление слабых зависимостей

Немаловажным этапом при построении взаимосвязей между событиями является стадия так называемой очистки. На этом этапе необходимо убрать из получившегося результата слабые связей, если это необходимо. В противном случае, так или иначе практически любые event могут иметь связывающее ребро, хотя вероятность, что они принадлежат одной тематике, возможно, будет крайне мала. В работе[4] предложены несколько функций для удаления слабых зависимостей. Статический порог. Этот метод использует некоторое константное значение оценки для удаления связей. В частности, если результат расчетов, полученный для двух event ниже, чем , мы считаем, что данные события не являются частью одной тематики. А следовательно эти события не являются тредом новостей в одном topic. Таким образом, данный метод исключает взаимосвязи ниже заданного порога. Формально можно представить метод статического порога как $G = (E, L)$, где $L = (ei, ej)|score((ei, ej))$, а e_i,e_j это множество event(событий), L' это множество ребер взаимосвязей между событиями, а score(e_i,e_j) результат построенной модели для двух событий e_i и e_j.
Static Pruning: В этом методе помимо использования статического порога , также устанавливаются верхние границы по количеству предшествующих или последующих событий для каждого event. Это равносильно установке максимального количества исходящих или входящих ребер, разрешенных для каждого события. Если степень вершины превышает верхнюю границу N_i, то необходимо отсортировать входящие(исходящие) ребра в порядке убывания значений. Определяющим фактором будет служить результат вычислений модели для каждой пары. Далее в соответствии с их оценками необходимо оставить только N_i ребер с наибольшим значением целевой функции.

Dynamic Pruning:Этот метод похож на метод статического сокращения(Static Pruning).Различие между ними заключается в ограничение на количество ребер. При статистическом сокращении накладывается ограничение отдельно на каждую из вершин, в то время как в этом методе верхний порог количества связей регулирует общее количество ребер. Разберем на примере, допустим у нас есть N_0 вершин, следовательно максимальное количество ребер для полного графа будет $N_0\frac{N-1}{2}$. В этом случае мы заменяем общее количество связей на некоторую константу L, которая будет отвечать за суммарное количество всех ребер. При этом подходе удаляются самые слабые связи во всей системе, а не для каждой вершины в отдельности. Этот принцип позволяет оставлять только самые качественные взаимосвязи и удалять наименее значительные.

2.7 Удаление сквозных зависимостей

Последним этапом при обработке трендовых новостей послужит еще одна ‘очистка’ от лишних зависимостей. Согласно предположению, все события, между которыми осталась связь после очистки на предыдущем шаге являются относящимися к одной тематике. Если говорить более условно, то все event для которых существует threading принадлежат к одному topic. Таким образом может возникнуть ситуация, при которой посты, которые объединены в треды, могут содержать лишние связи. Подобная ситуация изображена на рисунке 2. Это может привести к менее удобному структурированию для дальнейшего использования или же пропускам и ошибкам на этапе обхода графа, а следовательно и выдаче новостей.

Глава 3. Практическая реализация

3.1 Описание датасета

Перед тем как описывать этапы построения самой модели для анализа трендов, необходимо сказать пару слов о наборе данных, на основе которых будут построены взаимосвязи между event. Датасет предоставлен одной из самых популярных социальных сетей в России и странах СНГ 'Одноклассники'(OK). Ежедневно её используют до 40 миллионов пользователей, которые публикуют десятки миллионов постов и сообщений на открытых страницах. К тому же, авторы статей не ограничивается русским алфавитом и разнообразие сети насчитывает порядка 15 различных языков. Несмотря на общие корни, каждый язык требует отдельного внимания при решении задачи обработки естественного языка. Набор данных представлен датасетом, в котором агрегированы результаты работы «детектора трендов» в социальной сети OK (одноклассники) за период 10.10.2018-31.12.2018. Записи представлены только из открытых групп и сообществ. Сами данные разделены на две подкатегории. В первой собраны непосредственно трендовые посты с метками групп/сообществ и отметками об их кластерах. Помимо этого там имеются сырые необработанные записи, а также массивы слов, которые токенезированы и стеммированы с помощью apache lucene. Вторая часть является набором с уже трендовыми термами, их кластерами и указаниями на принадлежность к группам/сообществам.

3.2 Программные компоненты

В этой главе информация посвящена составляющим программной реализации данной модели. А в частности речь пойдет про инструменты и компоненты, благодаря которым получилось реализовать вышеизложенную систему.

Основой для написания послужил язык программирования Python версии 3.6, поскольку он обладает необходимым набором инструментов для анализа данных и огромным комьюнити для решения возникающих вопросов и трудностей. Также немаловажным плюсом является наличие одной

из наибольших базы с набором всевозможных библиотек и утилит, которые упрощают вычисления.

Программной оболочкой послужил Jupyter notebook. Это утилита для интерактивных вычислений, которая часто используется в задачах анализа данных. Одним из главных преимуществ данного программного обеспечения служит возможность запуска программы на удаленном сервере и совместная разработка. Помимо этого, jupyter notebook обладает набором встроенных функций и методов, позволяющим с помощью него легко визуализировать полученные данные, что является плюсом при решении задач подобного рода.

Библиотекой в программировании называется набор модулей и компонент, которые используются для выполнения той или иной задачи. Необходимость таких составляющих легко объяснима. Зачастую для выполнения каких-либо операций или вычислений, например, перемножения матриц, необходимо использование повторяющихся частей кода, что затрачивает большое количество времени. Намного практичнее использовать один и тот же код для одинаковых операций, а не переписывать его каждый раз. Именно для этого многие функции и классы объединяют в 'пакеты' для более практичного и удобного использования.

Вспомогательной платформой для вычисления сходства между лексическими конструкциями будет служить сервис 'RusVectōrēs' [11]. Этот инструмент позволяет исследовать отношения между словами в дистрибутивных моделях. А также выполнить различные операции, например:

- оценить семантическое сходство между словами
- найти слова, которые будут наиболее близки к заданному
- находить решение для аналогий, вида A относится к B, как C относится к D
- выполнение разных математических операций для векторов слов
- изображение семантических карт для отношений между словами.

Помимо всего этого, сервис предоставляет API, с помощью которого можно удаленно выполнить некоторые операции. В настоящее время сервис выдает результаты для двух видов запросов. Первый это нахождение ближайших соседей для слова, а второй это вычисление значения семантической близости между парой слов. Внутри ресурса есть несколько доступных моделей, которые были обучены на различных больших корпусах. Для реализации данной задачи был выбран Национальный Корпус Русского Языка(НКРЯ) в полном объёме. В составе обучающей выборки предложены порядка 270 миллионов слов. НКРЯ является информационно-справочной системой, основанная на собрании в электронной форме текстов на русском языке.

Поскольку возможности сервиса ограничены русским языком, необходимым средством является библиотека для определения принадлежности документа тому или иному языку, таким средством является программный пакет ‘langdetect’. Эта библиотека является прямой аналогией библиотеки ‘language-detection’ для языка программирования Java от компании Google. Недостатком этого решения является его недетерминированность, что означает, если запустить его на слишком коротком или неоднозначном тексте, то можно получить разные результаты для каждого его запуска. Это существенный недостаток, но в нашем случае этого удается избежать, поскольку объема сообщений из story хватает для его однозначного детектирования.

В качестве инструмента для оперирования веб-запросами была использована библиотека ’Requests’. Она состоит из методов для упрощения выполнения запросов(request) к серверу и обработки ответов(response). Для начала необходимо создать экземпляр класса requests и вызвать метод GET.

$$r = requests.get('url')$$

Данной строчкой формируется GET запрос к серверу, который указан в аргументах функции. Результат обработки можно проверить вызвав функцию ’status_code’, которая вернет статус запроса в общепринятой кодировке.

- 1xx: Informational (информационные)
- 2xx: Success (успешно)
- 3xx: Redirection (перенаправление)
- 4xx: Client Error (ошибка клиента)
- 5xx: Server Error (ошибка сервера)

В результате выполнения запроса 3.2 приходит строка с результатом выполнения оценки и некоторй дополнительной информацией, например, принадлежность слов к части речи.

3.3 Ход программной реализации

Для начала необходимо собрать результаты выдления трендов из датасета. Для работы нам будет интересен раздел с трендами. Во-первых, определимся со структурой данных. Поскольку данные представлены в виде json, то выбор был сделан в пользу такой встроенной структуры данных языка python как словарь(dict). Словари в Python являются неупорядоченными коллекциями произвольных объектов с доступом по ключу. В нашем случае ключом будет служить пара объектов, а именно месяц и день публикации статьи. Теперь каждый из дней представлен элементом словаря с возможностью доступа. Элементы ассоциативного массива, которым является словарь, представлены в виде пары 'key-value', где ключ мы уже обозначили, а вот значением будет служить набор кластеров в каждом из дней. Для этих целей удобнее всего использовать такую структуру, как list(список). Под списком подразумевается упорядоченная коллекция объектов с доступом по индексу. Каждый из элементов листа представляет собой уже известную структуру данны типа словарь. Поскольку каждый кластер в эпохе содержит несколько полей(ключей) и соответствующие им значения. В итоге, необходимо обработать датасет по вышеизложенному принципу и получить на выходе объект словарь, обращаясь к которму можно получить доступ к 'трендовым термам' для каждого кластера в каждом дне.

Следующим этапом необходимо определить функцию временной зависимости 2.5. Поскольку было решено выбрать скользящее окно размера 5, то реализуем функцию с единственным входным параметром. Этот параметр является количеством дней между новостями. В результате, на выходе функции получаем числовое значение-коэффициент для модели.

Дальнейшим действием является этап нахождение числовых значений функции определния семантической близости посредством отправки GET запросов к API сервиса 'RusVectōrēs'. Для упрощения дальнейшей обработки результатов респонсов и более этичного и структурированного кода было решено вынести данные шаги в отдельную функцию. В качестве входных параметров метод прнимает два значения типа string. В языке программирования python string является одним из базовых типов данных и служит для создания строковых переменных, которыми и являются 'трендовые' данные. Далее с помощью библиотеки requests формируется GET запрос, при условии кода 'успешно' можно обработать респонс и возвратить числовое значение типа float, который также является одним из базовых типов данных в языке программирования и обозначает числовые данные с плавающей точкой.

На основе вышенаписанных функций можно приступить к части построения весовых ребер между трендами. Процесс заключается в итеративном проходе по элементам созданного словаря и вычисления для каждой из пар произведения результатов функции семантической близости и функции временной зависимости. По результатам вычислений на основе статического порога, пропускаем результаты для тех пар, чьи веса принимают значения менее 0.3, а подходящие ребра добавляем в общий список 'хороших' ребер. При этом присутствует необходимость проверки листа уже существующих ребер на предмет наличия данной пары, поскольку выбранное скользящее окно распространяется в обе стороны от текущей эпохи, в нашем случае текущего дня. После этого этапа необходимо еще раз отфильтровать полученные ребра уже методом 'Static Pruning'. В этом методе необходимо обозначить максимально допустимую степень для каждой вершины. Так как целью работы является построение взаимосвязей между трендами и речь идет о новостных потоках, то имеет место быть

предположению о наличии у веришны не более 3 взаимосвязей с событиями из прошлых эпох, и не более 3 связей с событиями из последующих эпох. Общая архитектура системы представлена на 4

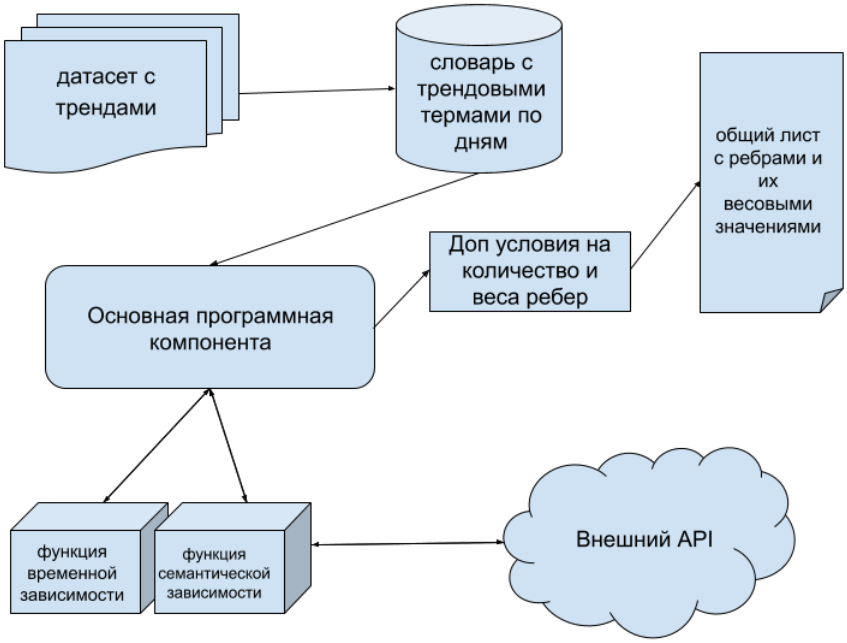

Рис. 4: Архитектура системы.

3.4 Полученные результаты

Рассмотрим полученные результаты на примере выборки за один месяц. В изначальном наборе данных суммарное количество кластеров, то есть количество трендовых событий было равно 1638. Из-за возможности обработки только русскоязычных текстов количество кластеров было значительно уменьшеноно до порядка 900. Такой большой скачок обусловлен широкой аудиторией социальной сети и наличию большого количества текстов, содержащих не только русские слова. По результатам работы алгоритма количество событий было уменьшено до 40. Такое малое количество оставшихся вершин получено в результате слишком выского проходного порога. Тем самым оставались только самые значительные зависимости. Помимо этого, небольшое значение скользящего окна также сильно повлияло на резкое уменьшение общих значений весовой функции для каждого

ребра. Так как величина рассматриваемого окна для каждого дня обратно пропорциональна скорости затухания функции временной зависимости.

Таблица 2: Значения временного коэффициента при рассматриваемом окне в 5 дней

t	τ
1	1
2	0.513
3	0.263

В таблице наглядно видно, что при выборе размера окна в три дня, с увеличением количества дней между новостями, значения функции уменьшаются почти в два раза, что играет сущесвтенную роль при вычислении результатов весов ребер.

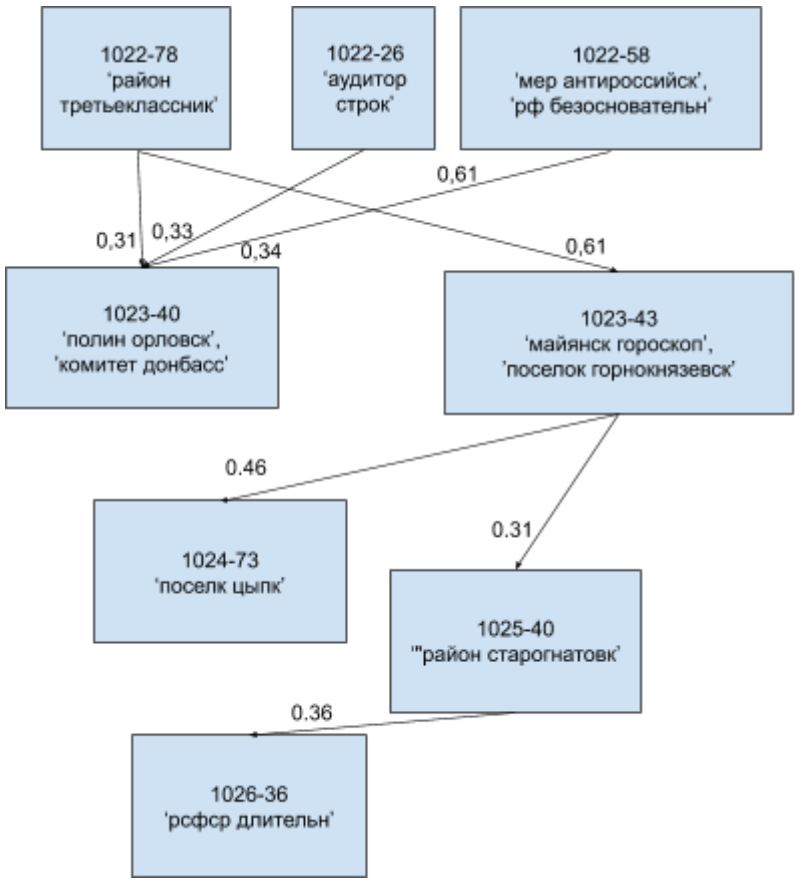

Рис. 5: Пример построения треда

На 5 можно увидеть один из примеров получившихся зависимостей. Из-за специфики решавшейся задачи и начального набора данных, для модели затруднительно применение обычных метрик оценки качества, таких как полнота, точность или, например, F-мера. Тем не менее, в получившихся ребрах присутствуют продолжительные треды соизмеримые с длиной ограничивающего окна. Полученное распределение количества значений весовых функций можно увидеть на 6.

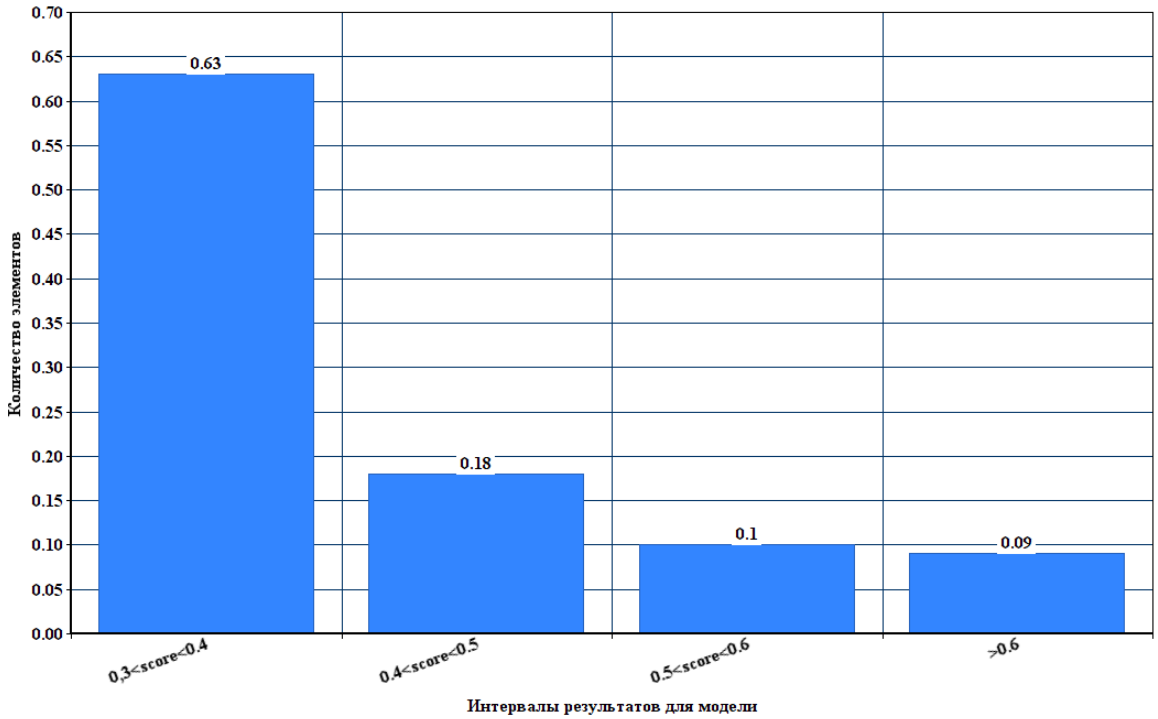

Рис. 6: Распределение результатов

Выводы

Задача структурирования информации и отслеживания изменений новостных событий в настоящее время является весьма актуальной. В течение данного исследования мной были изучены основные подходы к решению задач в данной области, рассмотрены различные алгоритмы для нахождения расстояния между разными семантическими конструкциями, такими как термы, документы. Также мной были проведены исследования на тему влияния временной зависимости на качество взаимовязей между 'трендовыми' новостями. Также рассмотрены были основные методы для удаления лишних ребер во взвешенных графах. Еще одним этам данной работы было программная реализация построенной модели. Для этих целей был проведен анализ документаций для нескольких библиотек, необходимых в решении поставленной задачи. Полученная в результате работы система имеет ряд недостатков, несмотря на это, она работает стабильно и показывает неплохой результат.

В дальнейшем планируется модифицировать полученную систему путем увеличения количества рассматриваемых факторов, оказывающих влияние на наличие связности между новостями. А также одним из последующих усовершенствований может служить адаптация работы модели для интерактивного режима отслеживания.

Заключение

В рамках проделанной работы была реализована модель для анализа наличия взаимосвязей между выделенными новостными трендами внутри социальной сети 'Одноклассники'. Предложенная система имеет большой потенциал для усовершенствований, в том числе для возможности работы в интерактивном режиме. За счет программной реализации существует возможность интегрирования предложенной модели в существующие сервисы. Немаловажным аспектом является наличие возможности для адаптации существующего решения к задачам по анализу данных в смежных областях.

Список литературы

[1] Ao Feng, James Allan. «Incident Threading for News Passages. ». CIKM'09, November 2–6, 2009, Hong Kong, China.

[2] Ao Feng and James Allan. «Finding and Linking Incidents in News. ». CIKM'07, November 6–8, 2007, Lisboa, Portugal.

[3] Bermudez Soto José Gregorio. «METHOD FOR MEASURING THE SEMANTIC-SIMILARITY OF TEXTUAL DOCUMENTS. «Izvestiya SFedU. Engineering Sciences, 2017.

[4] Congcong YangXiaodong ShiChih-Ping Wei. «Discovering Event Evolution Graphs From News Corpora «August 2009IEEE Transactions on Systems Man and Cybernetics - Part A Systems and Humans 39(4):850 - 863

[5] Erich Schubert, Michael Weiler, Hans-Peter Kriegel. «SigniTrend: Scalable Detection of Emerging Topics in Textual Streams by Hashed Significance Thresholds. »KDD'14, August 24–27, 2014, New York, NY, USA.

[6] James AllanRon PapkaVictor Lavrenko. «On-Line New Event Detection and Tracking.»August 2017ACM SIGIR Forum 51(2):185-193.

[7] Ramesh Nallapati, Ao Feng, Fuchun Peng, James Allan. «Event Threading within News Topics. ». CIKM'04, November 8–13, 2004, Washington,DC,USA.

[8] Magnus Sahlgren. «The distributional hypothesis. »Rivista di Linguistica 20.1 (2008), pp. 33-53.

[9] 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

[10] N. Neelova. «Investigating the lexical method of strings similarity computation based on preliminary processing.». 2009

[11] Kutuzov A., Kuzmenko E. «WebVectors: A Toolkit for Building Web Interfaces for Vector Semantic Models». Ignatov D. et al. (eds) Analysis of Images, Social Networks and Texts. AIST 2016. Communications in Computer and Information Science, vol 661. Springer, Cham.

[12] Kriukova A. V. «COMPUTING SEMANTIC SIMILARITY OF RUSSIAN TEXTS BY MEANS OF DKPRO SIMILARITY TOOL. ». Компьютерная лингвистика и вычислительные онтологии, 2017.

[13] С. Х. Г. Бермудес, С. У. Керимова. «О методе определения текстовой близости основанном на семантических классах. ». Электронный научный журнал «Инженерный вестник Дона», 2016.

[14] Воронкин Алексей Сергеевич. «Социальные сети: эволюция, структура, анализ.»Образовательные технологии и общество, 2014.