Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

«Извлечение отношений между словами на основе их векторного представления»

Автор: Лоскутов Игнат Анатольевич _____
Направление подготовки (специальность): 01.03.02 Прикладная математика и информатика
Квалификация: Бакалавр
Руководитель: Фильченков А.А., доц. каф. КТ, канд. техн. наук_____

К защите допустить
Зав. кафедрой Васильев В.Н., докт. техн. наук, проф. _____

«____» _____ 20_____ г.

Санкт-Петербург, 2017 г.

Студент Лоскутов И.А. **Группа** М3439 **Кафедра** компьютерных технологий **Факультет** информационных технологий и программирования

Направленность (профиль), специализация Математические модели и алгоритмы разработки программного обеспечения

Квалификационная работа выполнена с оценкой _____

Дата защиты «15» июня 2017 г.

Секретарь ГЭК _____

Листов хранения _____

Демонстрационных материалов/Чертежей хранения _____

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Текстовую информацию можно разделить на две различных категории: факты и мнения. Факты являются объективными высказываниями о некоторых событиях или сущностях. Мнения являются субъективными высказываниями, которые обыч- но выражают отношение людей к различным событиям, явлениям или сущностям. Но если мнений на какую то тему довольно большое количество, то, проанализиро- вав их соответствующим образом, можно выяснить объективную оценку изучаемого события или сущности. Таким образом, на основе множества мнений можно синте- зировать новые факты, что, несомненно, полезно и достойно исследования.

С развитием и распространением Интернета появляется все больше и больше мнений, выражаемых самыми разными людьми. Это, к примеру, отзывы о товарах в интернет-магазинах, реакция на мировые события в блогах, социальных сетях и на интернет-форумах. В результате получается очень объемный контент, создаваемый всеми пользователями. Сейчас, когда человек хочет приобрести какой то продукт, ему уже не так необходимо узнавать мнение своих друзей на этот счет. Достаточно лишь почитать отзывы в интернете и сделать вывод о качестве предстоящей покупки.

С другой стороны производителям и продавцам тоже необходимо понимать отношение пользователей к их продуктам или услугам. Для таких целей всегда традиционно нанимались консультанты или проводились соответствующие опросы. Однако, мнений сейчас становится очень много, так как число их источников постоянно растет, поэтому все больше времени и средств тратится на их анализ.

Отсюда и появляется задача автоматического анализа тональности текста - это задача определения эмоционального отношения автора текста к некоторому объекту (объекту реального мира, событию, процессу или их свойствам/атрибутам), выраженному в тексте. Тональность всего текста в целом можно определить как функцию (например, сумму) лексических тональностей составляющих его единиц (предложений, слов) и правил их сочетания. Соответственно, в простейшем случае тональная оценка может быть позитивная или негативная.

Методов решения этой задачи достаточно много, потому как мнения бывают разных видов: например, к анализу статей в журналах и к сообщениям в социальных сетях с неформальной лексикой нужны совсем разные подходы. Последние в свою очередь очень сильно набирает популярность, ежедневно размещаются миллионы сообщений обычных пользователей с суждениями о том с чем они сталкиваются в повседневной жизни.

Примером такой социальной сети является Twitter — социальная сеть для публичного обмена короткими (до 140 символов) сообщениями, которой пользуются сотни миллионов пользователей. Что очень важно для данного исследования: случайные twitter-сообщения редко имеют логическую связь между собой и пишутся на самые разнообразные темы. Это делает его очень удобным для тестирования новых реализаций методов анализа тональности.

В настоящей дипломной работе рассматривается задача построения словаря эмоционально окрашенных слов и последующее его применения для анализа тональности текстового корпуса.

В главе 1 будет рассмотрена постановка задачи и предоставлен обзор существующих решений.

В главе 2 будет проведено исследование и построение решения задачи.

В главе 3 будет описана практическая часть: инструменты разработки, общая схема работы и архитектура системы.

В главе 4 будут предоставлены сравнения с существующими решениями на конкретных примерах и сделаны выводы по эффективности предложенного алгоритма.

ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ И ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ

1.1. Постановка задачи

Целью данной дипломной работы является исследование и разработка метода распространяющейся активации для составления словаря эмоционально окрашенных слов на русском языке, а также последующая оценка эффективности его применения. Тестирование метода будет производиться уже на размеченном текстовом корпусе с равным количеством позитивных и негативных примеров. Мерой точности будет выступать доля верно распознанной эмоциональной окраски. Для достижения цели необходимо решить следующие задачи:

а) Исследовать предметную область, изучить существующие методы автоматического анализа тональности текстов;
б) Провести теоретическое исследование применимости метода распространяющейся активации для задачи построения словаря эмоционально окрашенных слов;
в) Реализовать метод распространяющейся активации и его варианты с целью решения задачи построения словаря эмоционально окрашенных слов;
г) Произвести экспериментальное сравнения реализованных алгоритмов.

1.2. Обзор существующих решений

Сделав большое обобщение, можно разделить существующие подходы на следу- ющие категории:

а) Подходы, основанные на словарях
б) Подходы, основанные на машинном обучении
в) Подходы, основанные на правилах

1.2.1. Использование словарей

Такой подход использует так называемый тональный словарь (affective lexicon) для анализа текста. В простом виде в тональном словаре словам ставится в соответствие вероятность быть отнесенным к определенной эмоции. Эта вероятность обычно рассчитывается на основе какого-либо текстового корпуса.

Этот подход имеет свои недостатки:

— Одно слово может иметь несколько значений, имеющих сильно различающиеся тональности;
— Результаты будут плохими, если анализировать текст такого жанра, который сильно отличается по языковым свойствам от жанра текстового корпуса, на основе которого составлялся словарь;

Тем не менее этот подход очень интересен для исследования: можно использо- вать различные текстовые корпуса и применять различные алгоритмы к ним для построения словарей. Именно этот подход изучается в данной дипломной работе.

Чтобы проанализировать текст, основываясь на этом подходе, можно воспользоваться следующим алгоритмом: сначала каждому слову в тексте присвоить его значением тональности из словаря (если оно присутствует в словаре), а затем вычислить общую тональность всего текста. Вычислять общую тональность можно разными способами. Самый простой из них – среднее арифметическое всех значений. Более сложный – обучить классификатор.

Стоит рассмотреть несколько известных словарей, специально размеченных с учётом эмоциональной составляющей.

1.2.2. WordNet-Affect

WordNet – это электронный тезаурус для английского языка, разработанный в Принстонском университете. Базовой словарной единицей в WordNet является не отдельное слово, а так называемый синонимический ряд (синсеты), объединяющий слова со схожим значением и по сути своей являющимися узлами сети.

WordNet-Affect был создан на основе WordNet для английского языка путём выбора и отнесения синсетов к различным эмоциональным понятиям. Синсеты основных частей речи были вручную размечены специальными эмоциональными метками, которые характеризуют различные состояния, выражающие эмоциональные отклики, или ситуации, которые вызывают эмоции. Все такие метки обьединяются в четыре дополнительных эмоциональных метки: позитивная, негативная, неоднозначная и нейтральная.

Физическая структура WordNet-Affect состоит из шести файлов-категорий: радость, страх, гнев, печаль, отвращение, удивление. На данный момент в этом словаре около 2900 синсетов и 4800 слов.

1.2.3. SentiWordNet

SentiWordNet - это словарь, полученный посредством автоматического аннотирования синсетов из WordNet в соответствии с его степенью позитивности, негативности и объективности. Каждая из этих степеней оценивается значением из интервала (0; 1), причем все три в сумме должны давать 1.

Процесс создания SentiWordNet состоял из двух шагов:

а) Используются методы машинного обучения с частичным привлечением учителя. Вначале выбиралось небольшое число синсетов, которые размечались вручную. Затем на них было обучено несколько классификаторов, которые должны были определять численные оценки каждого из синсетов. Таким образом через полученные модели были размечены все оставшиеся синсеты.

б) Затем к данным применялась модель случайного блуждания, чтобы установить окончательные оценки объективной, позитивной или негативной составляющей каждого синсета

1.2.4. SenticNet

SenticNet это еще один семантический тезаурус. Его отличие от двух рассмотренных состоит в том, что WordNet-Affect и SentiWordNet обеспечивают связывание слов и эмоциональных понятий на синтаксическом уровне, а SenticNet связывает понятия на семантическом уровне.

SenticNet построен на основе так называемых "sentic-вычислений". Это парадигма, которая использует методы искусственного интеллекта и семантической паутины для обработки мнений на естественном языке. Такая концепция позволяет проводить анализ документов не только на уровне целых страниц и текстов, но и на уровне предложений, что позволяет оценивать тексты на более высоком уровне детализации.

SenticNet сопоставляет каждому понятию Sentic-вектор"с численными значениями таких величин, как приятность, внимание, чуткость и способность, а также величину тональности, для задачи анализа тональности текста. На данный момент в этом тезаурусе около 14000 понятий.

1.3. Машинное обучение

Этот метод является наиболее часто используемым в исследованиях, поскольку есть довольно широкий выбор классификаторов, показывающих хорошие результаты, например: метод опорных векторов, наивный байесовский классификатор и т. д.

Краткий алгоритм работы этого метода:

а) Собирается коллекция документов (текстов), на основе которой будет обучаться классификатор;
б) Каждый документ (текст) раскладывается в виде вектора признаков, по которым он будет исследоваться;
в) Для каждого документа (текста) вручную указывается тип тональности;
г) Выбранный классификатор обучается на этой коллекции;
д) Можно использовать обученную модель для определения тональности другой коллекции.

1.4. Набор правил

Еще один подход заключается в создании набора правил, применяя которые, система делает заключение о тональности текста. Для этого текст разбивается на слова или последовательности слов (N-grams). Затем полученные данные используются для выделения часто встречающихся шаблонов, которым присваивается положительная или отрицательная оценка. Выделенные шаблоны применяются при создании правил вида “ЕСЛИ условие, ТО заключение”. Такие правила представляют собой комбинации различных членов предложения между собой. В итоге, если выработана хорошая система правил, то любое предложение сводится к одному из правил, для которого определена тональность. Часто этот метод использует словари для начального присвоения тональности отдельным словам.

Основная проблема этого подхода состоит в трудоемкости создания качественной системы правил. Использование этого метода позволяет классифицировать документы с высокой точностью, но только при условии, что используются правила выработанные для конкретной предметной области документа. Поэтому для создания такой системы

правил требуются значительные усилия экспертов в рассматриваемой области.

ГЛАВА 2. ИССЛЕДОВАНИЕ И ПОСТРОЕНИЕ РЕШЕНИЯ ЗАДАЧИ

Основной задачей, поставленной в данной работе является разработка метода построения словаря эмоционально окрашенных слов на основе алгоритма распространяющейся активации. Соответственно решение данной задачи включает в себя следующие пункты:

а) Теоретическое исследование применимости метода распространяющейся активации для задачи построения словаря эмоционально окрашенных слов;
б) Реализация метода распространяющейся активации и его вариантов;
в) Экспериментальное сравнение реализованных алгоритмов.

2.1. Теоретическое исследование применимости метода распространяющейся активации

2.1.1. Алгоритм распространяющейся активации

Пусть задан граф с N вершинами, каждой из которых сопоставлено значение активации $A[i]$, которое является вещественным числом на отрезке $[0; 1]$. Каждое ребро $[i, j]$, соединяющее вершины i и j имеет вес, обозначаемый как $W[i, j]$, который тоже является вещественным числом с отрезка $[0; 1]$. Также задан коэффициент затухания D, имеющий вещественное значение с отрезка $[0; 1]$.

Процедура:

а) Все значения активации $[i]$ обнуляются. Выбирается несколько вершин, с которых начнется активация и устанавливается их $A[i]$;
б) Для каждой вершины, имеющей ненулевое $A[i]$, рассчитывается значение активации всех его соседних вершин(которые соединены с ним ребром $[i, j]$) по следующей формуле $A[j] = max(A[j]; A[i] * W[i, j] * D)$;
в) Пункт 2 повторяется до тех пор пока суммарное значение изменения активации по всем вершинам не станет меньше какого то установленного малого значения.

2.1.2. Применение метода распространяющейся активации для задачи построения словаря эмоционально окрашенных слов

Для построения необходимого графа слова выступают как вершины, а наличие ребра между словами означает существования биграммы

из них в текстовом корпусе. Каждая вершина при этом уникальна и в итоге получается граф всех словосочетаний, считанных из набора данных. Веса ребер для каждой биграммы также заданы как какая-либо её характеристика. Например, это может быть количество вхождения биграммы в текст.

Для каждой вершины в графе будет два значения активации: первое (будем его называть $P[i]$) является вероятностью принадлежности слова к классу позитивно окрашенных, второе значение (назовем его $N[i]$) - это вероятность принадлежности к классу негативно окрашенных слов. Для инициализации вручную составляется два списка слов. В первом списке слова придающие тексту заведомо позитивную окраску, поэтому их $P[i]$ устанавливается равным единице при инициализации графа. Во втором списке слова придающие негативную окраску, их $N[i]$ становятся равны также единице. Оба списка представлены в Приложении 1. После этого на графе запускается распространяющаяся активация.

2.2. Реализация метода распространяющейся активации и его вариантов

2.2.1. Использование национального корпуса русского языка

Первым был выбран текстовый корпус ruscorpora, который составлен, по большей части, из литературы XX века. Для создания графа были использованы частоты вхождения каждой биграммы в весь корпус, предоставляемые создателями этой коллекции документов.

Теперь $W[i,j]$ - это число вхождения биграммы в корпус и оно, соответственно, больше 1. Но $A[j]$ это вероятность быть отнесенным к определенному классу тональности, а так как $W[i,j]$ теперь может быть довольно большим числом, то $A[j]$ может стать больше единицы, что недопустимо. Поэтому было принято решение нормировать веса ребер, исходящих из вершины, на наибольший из них.

Значение активации рассчитывается по следующей модифицированной формуле: $A[j] = max(A[j]; A[i] * \frac{W[i,j]}{max_i(W[i,j])} * D)$, где максимум берется по всем i при фиксированном j. Соответственно на этом наборе данных запускался алгоритм распространяющейся активации и на выходе получался набор данных из слов и сопоставленных каждому из них

двух значений: вероятность внести позитвный вклад в текст и вероятность внести негативный вклад в текст.

2.2.2. Технология Word2Vec

Word2Vec – технология статистической обработки больших массиво текстовой информации. Она делает отображение текстового корпуса на множество векторных представлений слов из этого корпуса. Такие векторные представления основаны на том как часто слова встречаются вместе или при каких схожих обстоятельствах они употребляются.

Основу Word2Vec составляют два алгоритма: continuous bag-of-words и skipgram. Continuous bag-of-words – это модель в которой слова из предложения или текста рассматриваются как одно большое множество, игнорируя порядок в котором они следовали друг за другом. Skip-gram – это обобщение понятия N-граммы, в котором слова из рассматриваемого текста не обязаны браться последовательно, а могут пропускаться.

По сути, два этих алгоритма можно рассматривать как выходной слой в нейронной сети, которая используется для обучения Word2Vec. Притом continuous bagof-words используется для предсказания слова по контексту, который подается на входной слой, а skip-gram используется для предсказания контекста по слову, поданному на вход.

Чтобы получить качественные зависимости между векторными представлениями слов, нужно обучать нейронную сеть на больших текстовых корпусах. В данной работе используется Word2Vec, обученный на русской Википедии.

Поскольку каждое слово представимо в виде численного вектора, есть возможность считать косинус угла между ними. Соответственно модуль этой величины всегда меньше единицы и в дальнейшем она будет называться сходством между словами. Чем она больше, тем более синонимичны рассматриваемые два слова между собой.

2.2.3. Создание словаря с использованием Word2Vec

На этом этапе было необходимо осуществить предварительную обработку векторных представлений всех слов, встречаемых в русской Википедии. Из них составлялись биграммы по следующему принципу: для

каждого слова находилось 50 наиболее схожих с ним, и соответственно числом, характеризующим биграмму, являлось сходство между её элементами.

Теперь в методе распространяющейся активации $W[i,j]$ это сходство между сло- вами, и, так как оно по модулю не превышает единицу, $A[i]$ будет лежать в пределах $[0;1]$. Таким образом, значение активации можно рассчитывать следующим образом: $A[j] = max(A[j]; A[i] * W[i,j] * D)$. Тем не менее тестировался и вариант с нормировкой со следующей формулой: $A[j] = max(A[j]; A[i] * \frac{W[i,j]}{max_i(W[i,j])} * D)$.

В результате обработки также получился словарь, состоящий из слов русского языка и соответствующих каждому из них два значения: его позитивная оценка и негативная.

2.2.4. Модификация метода на машинном обучении

Как было описано в обзоре существующих решений, задачи анализа тональности текста можно решать методами машинного обучения. А полученный словарь можно использовать для модификации признаков, используемых для обучения классификатора.

Изначально векторами признаков являются числа вхождений каждой униграммы из корпуса в элементы выборки. То есть, признаков столько же, сколько и униграмм во всем текстовом корпусе. В качестве классификатора использовался наивный байесовский классификатор.

Модификация этого метода словарем заключается в следующем: для текстового корпуса, на котором будет обучаться классификатор, рассчитывается тональность каждого элемента корпуса с помощью полученного словаря. И полученные значения тональности можно использовать как еще один дополнительный вектор признаков для обучения классификатора.

2.2.5. Экспериментальное сравнение реализованных алгоритмов

Полученные в результате работы алгоритма словари были протестированы на корпусе сообщений Twitter ИСП РАН. Он состоит из 6328 вручную размеченных на три категории тональности твитов: положительные, нейтральные и негативные. В таблице 1 приведено описание корпуса:

Таблица 1 – Описание корпуса для тестирования

Тональность	Число сообщений
Позитивная	1618
Негативная	1607
Нейтральная	3103

Сообщения с нейтральной тональностью нам не нужны, поэтому они были удалены из рассматриваемого набора данных. Так как число позитивных и негативных сообщений в наборе данных практически равно, то в качестве меры качества классификации была выбрана точность: отношение числа верно распознанных эмоциональных оттенков сообщений к общему числу сообщений. Решение о том, какую эмоциональную окраску имеет сообщение, принималось самым простым способом: большей из сумм негативных и позитивных окрасок слов в сообщении. В таблице 2 приведены результаты тестирования полученных словарей:

Таблица 2 – Результаты тестирования

Словарь	Точность
На основе Национального корпуса русского языка	0.56
На основе Word2Vec без нормировки	0.685
На основе Word2Vec с нормировкой	0.64

Топ-100 позитивных и негативных слов, найденных с помощью распространяющейся активации алгоритмом на основе Word2Vec без нормировки, есть в приложении 2.

Еще одним протестированным методом была модификация на машинном обучении. Дополнительный вектор признаков рассчитывался на основе словаря созданного с использованием Word2Vec без нормировки, так как он показал лучшие результаты. Для обучения классификатора и расчета дополнительного вектора признаков использовался тот же текстовый корпус Twitter. В таблице 3 приведены результаты тестирования реализации данного метода:

Можно наблюдать, что при большом обьеме выборки прироста качества практически нет. Тем не менее данная модификация показывает

Таблица 3 – Результаты тестирования модификации

Части корпуса		Качество классификации		
Train	Test	NB	NB с модификацией	Улучшение
2903	322	0.8561 ± 0.0012	0.8579 ± 0.0012	0.0017 ± 0.0006
322	2903	0.7517 ± 0.0007	0.7659 ± 0.0006	0.0142 ± 0.0005

лучший прирост при обучении классификатора на выборке маленького обьема.

ГЛАВА 3. ОПИСАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ

Данный метод был реализован на языке Scala при помощи фреймворка Apache Spark, предназначенного для параллельной обработки больших объемов данных, поскольку для создания словаря требовалась обработка большого текстового корпуса.

3.1. Использованный инструментарий

В качестве основного языка разработки был выбран Scala. Язык Scala является современным кроссплатформенным языком, сочетающий возможности функционального и объектно-ориентированного программирования.

Также широко использовался фреймворк Apache Spark является мощным средством для реализации параллельных алгоритмов по обработке больших объемов данных на кластере. Это проект с открытым исходным кодом(open source), предоставляющий удобный и лаконичный интерфейс для написания программ на языках Scala, Java, Python и R. Выбор в пользу фреймворка Spark был сделан по нескольким причинам:

- В нем есть подсистема для построения графов и работы с ними;
- В нем реализована удобная работа с большими данными посредством реализации собственной структуры данных (RDD);
- Один из самых быстрых фреймворков для распределенных вычислений и в связи с этим очень динамично развивающийся.

Основная структура данных, используемая в Spark - RDD (Resilient Disturbed Dataset). Это неизменяемая, устойчивая к сбоям структура данных. То есть по сути любая операция над RDD порождает новую структуру данных, при этом Spark хранит всю историю происхождения конкретной RDD, чтобы в случае сбоя можно было легко восстановить её. Еще одним важным аспектом работы в Spark является возможность многократного доступа к загруженным в память данным. Для работы с графами использовался GraphX - один из компонентов Apache Spark. GraphX основан на концепции bulk-synchronous parallel - модели параллельных вычислений. Вычисления в этой модели происходят посредством последовательности супершагов, каждый из них состоит из трех компонент:

а) Параллельное вычисление: каждый вычислительный узел выполняет локальные вычисления и использует данные находящиеся непосредственно на узле. Вычисления протекают асинхронно с остальными узлами, однако они могут прерываться сообщениями с других узлов;
б) Пересылка сообщений: процесс обмена данными между узлами
в) Барьерная синхронизация: когда узел достигает барьерной точки (общей для всех узлов), он ожидает, пока все остальные узлы не достигнут этого же барьера.

В GraphX эта модель воплощена следующим образом. Выполняется последовательность супершагов, причем, каждый из них содержит следующие шаги:

а) Вершины графа получают входящие сообщения от своих соседей, посланные с предыдущего супершага;
б) Основываясь на информации из сообщений, рассчитывается новое значение вершины;
в) Отправляются сообщения соседним вершинам.

Эта последовательность супершагов выполняется, пока на каком то шаге не окажется 0 полученных и отправленных сообщений всеми вершинами графа. В соответствии с описанной концепцией пересылка сообщений возможна только между соседними вершинами.

Для того чтобы задать граф, необходимо указать RDD вершин графа и RDD ребер. Причем, кратные ребра и петли в графе допускаются. Еще одной важной концепцией в GraphX является понятие триплета - обьекта, в котором, помимо информации о ребре, также хранится информация о вершинах, к нему примыкающих.

3.2. Общая схема работы

Сначала система считывает в RDD набор данных с биграммами и их численными характеристиками. Он преобразовывается к триплетному виду, то есть это становится RDD типа ((String, String), Double), так как в наборе данных содержатся пары слов с поставленными к ним в соответствие числами. Далее из всех словосочетаний составляется список уникальных слов, который будет использован в дальнейшем для создания множества вершин.

Дальше начинается работа непосредственно с GraphX. Набор данных теперь представляет из себя структуру RDD[((String, String), Double)] и при этом каждый элемент этой структуры представляет из себя ребро. То есть получается, что множество ребер уже задано. Далее создается RDD вершин на основе списка уникальных слов: каждое из них становится вершиной и каждой вершине ставится в соответствие два числа - вероятности внести позитивный и негативный вклады в текст. По умолчанию эти числа равны 0. Часть вершин из вручную составленного списка инициализируются посредством замены их вероятностей на 1.

На сформированном графе запускается алгоритм распространяющейся активации. Его конкретная реализация основана на следующих супершагах:

а) Для каждой вершины с ненулевым значением активации посылаются сообщения соседним вершинам с посчитанным значеним активации для них по заданной формуле;
б) Все вершины агрегируют сообщения от своих соседей;
в) Рассчитывается новое значение активации для каждой вершины на основе полученных сообщений. Как правило выбирается наибольшее значение активации, среди всех присланных и текущего.

Алгоритм завершает свою работу после того как суммарное значение активации всего графа перестаёт меняться.

После этого все вершины графа имеют рассчитанные для них значения активации. Слово и соответствующие ему значения активации записываются в файл, который и является искомым словарём.

3.3. Архитектура системы

Архитектура модулей разработанного метода представлена на Рис. 1. Класс Builder является основным и он возвращает искомый словарь. Первичная обработка данных происходит в классе Parser. Инициализация графа происходит в классе Initializer. В классе Converter строятся необходимые структуры данных для графа. Далее, в классе Activator на инициализированном графе выполняется распространя- ющаяся активация.

Диаграмма классов	ToD

ЗАКЛЮЧЕНИЕ

В рамках данной дипломной работы получены следующие результаты:

а) Изучены существующие методы автоматического анализа тональности текста;

б) Разработан и реализован метод построения словаря на основе алгоритма распространяющейся активации;

в) Проведена экспериментальная оценка эффективности метода и его модификаций на текстовом корпусе Twitter-сообщений.