Санкт-Петербургский государственный университет

Факультет прикладной математики – процессов управления

Кафедра Технологии Программирования

Горбатюк Анна Витальевна

Выпускная квалификационная работа бакалавра

Использование алгоритма контекстной кластеризации документов для кластеризации страниц и посещающих их пользователей без использования контента страниц

Направление 010300

Фундаментальная информатика и информационные технологии

Научный руководитель,

кандидат физ.-мат. наук,

доцент

Добрынин В.Ю.

Санкт-Петербург

2016

Содержание

Введение.

В настоящее время среди задач информационного поиска задача кластеризации информации занимает одну из лидирующих позиций. Существует множество способов решения данной задачи, но все так же остается вопрос о поиске наиболее выгодного, более быстрого, более точного метода из всех существующих методов, вопрос о том, какой метод и в какой задаче нужно применить, чтобы получить наиболее точные результаты за наименьшее количество времени и минимальные ресурсы.

Когда человек просматривает страницы в интернете, статьи и тексты он может легко понять к какой теме они относятся, какие ключевые слова можно выделить, понять суть, но в реальном мире, обработку информации в лоб невозможно доверить человеку, в связи с тем, что из-за больших объемов входных данных, он физически не сможет с этим справиться, либо это займет очень большое количество времени. Поэтому в задачах, связанных с поиском и обработкой информации необходимо автоматизировать процессы классификации и кластеризации, чтобы мы смогли автоматически получать краткую, но точную информацию о некотором наборе документов в виде статей, текстов или страниц, с которыми нам необходимо работать.

В данной работе будет рассматриваться задача кластеризации страниц и посещающих их пользователей. Актуальность данной работы заключается в том, что в случае кластеризации страниц выбранный метод контекстной документной кластеризации не использует контента данных страниц, поскольку документами в этом случае являются страницы, а словами в данных документах – пользователи, посетившие эти страницы. Это довольно выгодно при обработке очень больших коллекций данных, когда мы не в состоянии физически просмотреть содержимое каждой страницы, поскольку на это может уйти огромное количество времени. Подобное решение может быть очень полезно при анализе данных, группировке и распознавании

объектов, поиске информации, так же активное применение можно найти в задачах, связанных с web рекламой.

Метод контекстной документной кластеризации состоит из 2 этапов. На первом этапе находятся все контексты - вероятностные распределения набора слов, которые появляются вместе с данным словом в документе. Среди них находятся узкие контексты. Вопрос относительно определения понятия узких контекстов и их нахождения является довольно сложным, подробнее он будет описан в самой работе. На втором этапе узкие контексты используются как аттракторы кластеров. Аттрактор - узкий контекст, принадлежащий некоторому кластеру. Число аттракторов равно числу кластеров. Вычисляя расстояние Йенсена-Шеннона между документами и аттракторами кластеров, можно определить к какому из кластеров относится данный документ. Принадлежность документа к кластеру определяется наименьшим расстоянием с его аттрактором, относительно расстояний документа с другими аттракторами. Более подробно познакомиться с алгоритмом, его реализацией, практическими экспериментами, полученными экспериментальными данными и их анализом можно в данной работе.

Постановка задачи.

Начнем с того, что передо мной не стояло задачи анализа и исследования нескольких алгоритмов, с целью выбора наиболее эффективного. В начале моей научной деятельности мой научный руководитель предложил мне изучить алгоритм контекстной документной кластеризации, реализовать его и проследить его работу на некотором наборе данных, которые так же были предоставлены мне научным руководителем. О самих данных мы поговорим немного позже в главе 1. В связи с этим, передо мной были поставлены следующие задачи.

Имея заданный набор данных необходимо:

1) изучить алгоритм контекстной документной кластеризации;
2) реализовать предложенный алгоритм;
3) найти все контексты и разбить из на группы двумя различными способами для последующего выбора узких контекстов из этих групп. Определить на практике наиболее эффективный способ разбиения контекстов.
4) произвести кластеризацию заданного набора данных для каждого способа разбиения контекстов;
5) найти наиболее оптимальное количество кластеров для каждого способа разбиения контекстов;
6) оценить качество кластеризации.

Обзор литературы.

Одной из книг, в которых можно найти информацию об основных понятиях, задачах и проблемах информационного поиска и поиска в вебе, является книга [1], которая была написана преподавателями Станфордского и Штутгартского университетов и переведена на русский язык при поддержке компании «Яндекс».

Так же среди русских источников можно уделить внимание бакалаврской работе [2]. В своей работе автор использует контекстную документную кластеризацию для построения тематических моделей. Можно проследить все плюсы и минусы данного метода при работе с русскоязычными и англоязычными коллекциями документов.

Среди англоязычных источников непременно необходимо отметить статью [3], посвященную непосредственно контекстной документной кластеризации, содержит не только весь необходимый теоретический материал, но и практические данный полученные авторами статьи для таких коллекций документов как Reuters-21578 и 20 Usenet Newsgroups(20NG).

Дополнительную информацию о контекстной документной кластеризации так же можно найти в статье [4]. Авторы статьи проводят практические исследования, чтобы показать, что данный метод является стабильным.

В статье [5] мы так же можем наблюдать еще одно исследование посвященное методу контекстной документной кластеризации. Авторы пытаются оценить способность подхода тематической документной кластеризации определить общее между документами, принадлежащими одному кластеру.

Поскольку в ходе работы приходилось работать с базами данных, в книге [6] можно узнать подробнее о том, что такое база данных, какие виды баз данных бывают и их особенности.

Все начальные данные и полученные в ходе реализации алгоритма и проведения практических экспериментов хранились в MySQL базе данные. Об особенностях этой базы данных и об ее функционале, а также о взаимодействии с сервером можно прочитать в [7]. В [8] приведена документация, в ней так же можно найти всю необходимую информацию про работу с MySQL Workbench.

Для реализации алгоритма контекстной документной кластеризации я выбрала QT из-за простоты работы с различными базами данных, в том числе MySQL, простотой реализации GUI приложений. Прочитать о том, как создавать такие приложения и работать с базами данных, а также об основных библиотеках можно в книге [9]. Так же приведена официальная документация с сайта QT в следующей ссылке [10].

Для оценки качества кластеризации мною была заимствована идея, описанная автором в статье [11], в которой предлагался способ оценки качества тематических моделей людьми.

Глава 1. Начальные данные, их начальная обработка и хранение.

1.1 Начальные данные и их первоначальная обработка.

Данные для решения поставленной задачи были предоставлены мне научным руководителем, предварительно эти данные были закодированы, в связи с коммерческой тайной. Пользователи были скрыты за уникальными идентификаторами. Стоит отметить, что все наши данные мы будем хранить в таблицах в базе данных, подробнее о хранении данных можно прочитать в разделе 1.2. В конечном итоге я имела доступ к таблице, хранящей следующие отношения:

<ссылка> : <идентификационный номер пользователя>

то есть нам известна ссылка и пользователь, который посетил данную ссылку. В общей сложности данная таблица содержит 1 354 325 записей. Всего 47 453 уникальных ссылок и 851 899 уникальных идентификационных номеров пользователей. В таблице 1.2.1 можно увидеть пример реальных данных:

url	id_user
http://www.eteknix.com/4k-gaming-showdown-amd-r9-290x-r9-280x-vs-nvidia-gtx-titan-gtx-780/11/	138142
http://www.eteknix.com/amd-silently-launched-fx-8310-oem-model/	257804
http://www.eteknix.com/amds-new-catalyst-15-4-beta-driver-is-optimized-for-gta-v	259455
http://www.eteknix.com/amds-new-catalyst-15-4-beta-driver-is-optimized-for-gta-v/	813802
http://www.eteknix.com/4k-gaming-showdown-amd-r9-290x-r9-280x-vs-nvidia-gtx-titan-gtx-780/12/	636013

Таб.1.2.1 Пример начальных данных.

Для того, чтобы получить кластеризации ссылок и посетивших их пользователей нам необходимо сгруппировать наши исходные данные следующим образом:

1 тип: Url : user_1, user_2,…user_n
То есть каждая уникальная ссылка будет документом, а каждый пользователь, который посетил данную ссылку, является словом. Данная коллекция документов необходима для кластеризации страниц. В таблице 1.2.2 приведен пример для одной ссылки.

url	id_user
http://www.eteknix.com/amd-will-enter-tablet-market-2015/	202560
http://www.eteknix.com/amd-will-enter-tablet-market-2015/	282665
http://www.eteknix.com/amd-will-enter-tablet-market-2015/	300027

Таб. 1.2.2 Пример одного документа для кластеризации ссылок

2 тип: User_Id: url_1, url_2, … url_n
В данном случае каждый уникальный идентификатор пользователя будет являться документом, а все ссылки, которые он посетил, будут словами.

Данная коллекция документов необходима для кластеризации пользователей. В таблице 1.2.3 приведен пример для одного пользователя.

url	id_user
http://www.majorgeeks.com/files/details/kaspersky_tdsskiller.html	23423
http://www.majorgeeks.com/mg/getmirror/kaspersky_tdsskiller,2.html	23423
http://www.majorgeeks.com/mg/sortdate/rootkit_removal.html	23423

Таб. 1.2.3 Пример одного документа для кластеризации пользователей.

Проделав данную работу, мы получим необходимую нам коллекцию документов для выбранной кластеризации.

1.2 Организация хранения данных в MySQL базе данных.

Все наши начальные, промежуточные и конечные данные мы будем хранить в MySQL базе данных в таблицах. Доступ к данным будет осуществляться через MySQL Workbench. Чтобы мы могли работать в этой среде, а также, чтобы программы, которые будут написаны, могли работать с

данными из таблиц, нам необходимо установить и запустить MySQL Server. Как сделать это можно прочитать в книге [7].

На рис. 1.1.1 приведена схема всех таблиц, которые использовались для хранения всех данных в ходе работы:

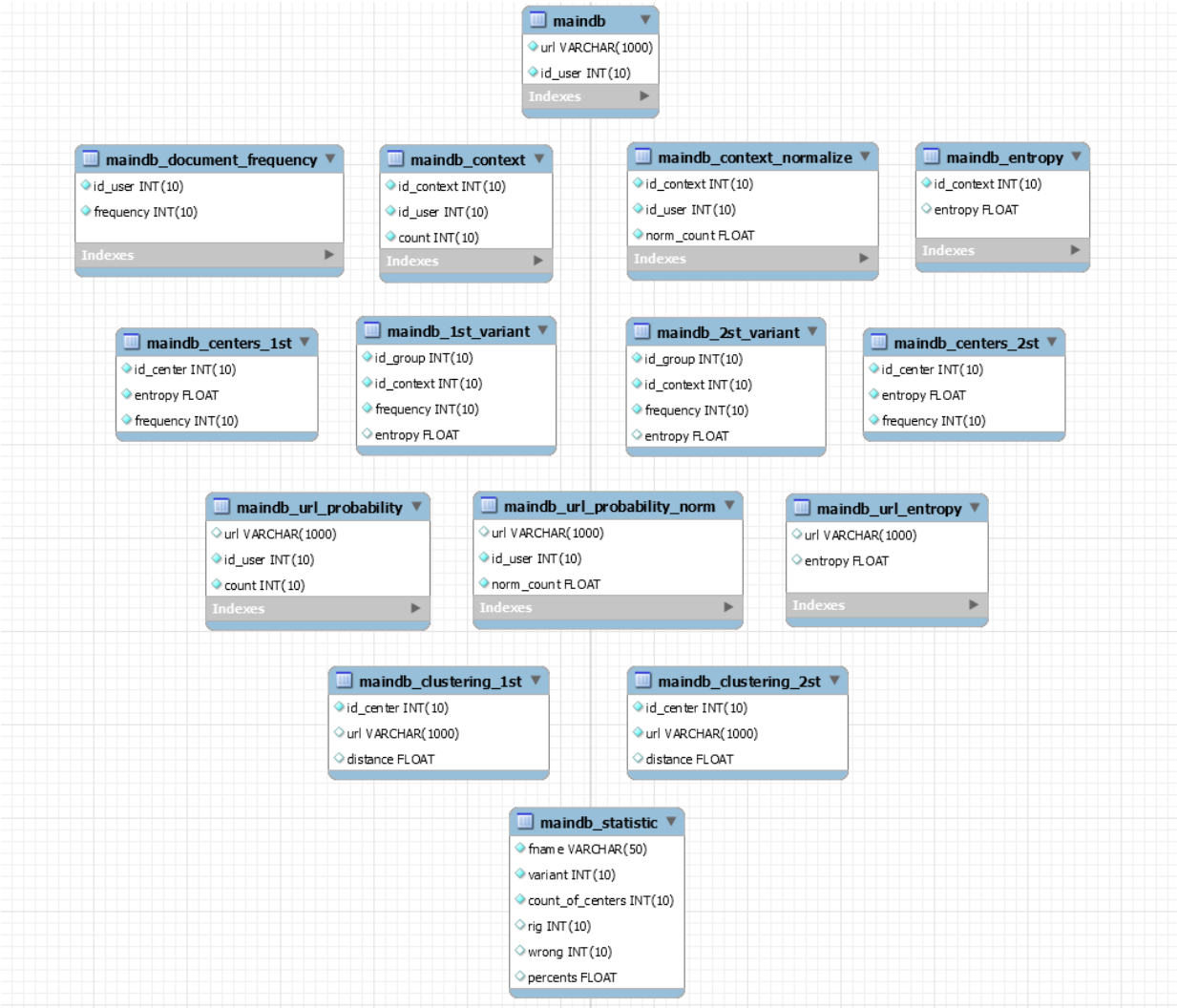

Рис. 1.1.1 Диаграмма таблиц базы данных

Теперь стоит упомянуть о том, что данная организация была построена для кластеризации ссылок, аналогичным образом можно построить диаграмму для кластеризации пользователей, но поскольку на практике мною была реализована только кластеризация ссылок, в данной работе будет приведено только одна диаграмма. На вопрос о том, почему на практике была реализована только кластеризация ссылок ответ можно будет найти чуть позже в главе 4.

maindb - первоначальные данные, содержащие информацию о том, какой пользователь посетил какую страницу.

maindb_document_frequency - содержит документную частоту.

maindb_context - содержит информацию о всех контекстах.

maindb_context_normalize – нормализованный вид данных из таблицы maindb_context.

maindb_entropy – содержит энтропии контекстов.

maindb_1st_variant – разбиение всех контекстов на группы первым способом.

maindb_2st_variant - разбиение всех контекстов на группы вторым способом.

maindb_centers_1st – узкие контексты, выбранные из групп контекстов, разбитых на группы первым способом.

maindb_centers_2st - узкие контексты, выбранные из групп контекстов, разбитых на группы вторым способом.

Об контекстах, разбиении на группы всех контекстов и нахождении узких контекстов можно подробнее узнать в граве 2.

maindb_url_probability – распределение пользователей.

maindb_url_probability_norm – нормализованные данные из таблицы maindb_url_probability.

maindb_url_entropy – энтропии распределения пользователей.

Подробнее о распределении пользователей можно будет прочитать в разделе 3.2.

maindb_clustering_1st – содержит данные о том, какая ссылка в какой кластер попала. Аттракторами являются контексты из maindb_centers_1st.

maindb_clustering_2st - содержит данные о том, какая ссылка в какой кластер попала. Аттракторами являются контексты из maindb_centers_2st.

О кластеризации на основе узких контекстов можно узнать подробнее в разделе 3.3.

Глава 2. Нахождение узких контекстов.

2.1 Основные теоретические сведения.

В разделе 1.1 главы 1 мы определили, что такое коллекция документов и что является словами в каждом документе. Теперь дадим определение основным понятиям необходимым нам для понимания и решения поставленной задачи. Предложенная терминология была взята из источников [2] и [3].

Контекст слова - это вероятностное распределение набора слов, которые появляются вместе с данным словом в документе. Другими словами, контекст слова -это распределение условных вероятностей

$$p(Y|z),$$

где Y случайная величина со значением из словаря и z данное слово, описывающее контекст.

Отличительной чертой использования контекстов в кластеризации является тот факт, что мы используем общую информацию между документами, основанную на их векторах признаков, чтобы определить сходство. В случае кластеризации страниц мы используем информацию о посещении общих страниц пользователями, чтобы определить сходство.

Термин «узкий контекст слова» является очень трудным для описания. «Узкость» контекста определяется энтропией вероятностного распределения и документной частотой для контекстного слова z. Малые значения энтропии свидетельствуют о том, что контекстные слова описываются относительно небольшим количеством слов, а значит само слово может оказаться термином в какой-то теме. Поэтому нас будут интересовать не все контексты, а лишь те, что имеют наименьшую энтропию.

Энтропия – мера неопределенности, вычислить которую можно по следующей формуле:

$$H(x)=-\sum_{i=1}^{n} p(i)\log\ p(i)\ ,$$

то есть энтропия независимой случайной величины x с n возможными исходами(от 1 до n).

Если применить эту формулу для распределения условных вероятностей, получим:

$$H(Y|z)=H[p(Y\vee z)]=-\sum_y p(y\vee z)\log\ p(y\vee z)\ ,$$

где Y случайная величина со значением из словаря и z данное слово, описывающее контекст, $p(y\vee z)$ – эквивалентно вероятности, что слово y выбирается при условии, что документ выбирается случайным образом из D_z (с вероятностью пропорциональной количеству появлений слова в документе), где D_z - набор всех документов со словом z. Иначе, $p(y\vee z)$ можно представить в виде формулы :

$$p(y|z)=\frac{p(y,z)}{p(z)},$$
$$p(y,z)=\sum_x p(x)\,p(y\vee x)\,p(z\vee x),$$
$$p(z)=\sum_x p(x,z),$$
$$p(x)=\sum_y p(x,y),$$
$$p(x,y)=\frac{tf(x,y)}{\sum_{x',y'} tf(x',y')}$$

Таким образом получаем:

$$p(y\vee z)=\sum_{x\in Dz}\frac{tf(x,y)}{\sum_{y'} tf(x,y')}\cdot\frac{tf(x,z)}{\sum_{x'\in Dz} tf(x',z)}\ ,$$

Где x –документ из коллекции документов X, y – слово из У набора всех слов встречающихся в X, z данное слово, описывающее контекст, tf(x , y) – количество появлений слова y в документе x. Данная формула является неудобной для вычисления, в связи с тем, что функция p(x , y) должна пересчитываться каждый раз, когда добавляется новый документ в коллекцию. Поэтому вместо нее используют следующую формулу:

$$p(y\vee z)=\frac{\sum_{x\in Dz} tf(x,y)}{\sum_{x\in Dz,y'} tf(x,y')}\ .$$

Документной частотой называется число документов коллекции, в которых данное слово встречается хотя бы один раз.

$$df(z) = |\{x :\ tf(x, z)\ >\ 0\}|,$$

где z – слово, x – документ, tf(x , z) – количество появлений слова z в документе x.

2.2 Нахождение всех контекстов.

Для начала необходимо найти все контексты. Рассмотрим первый случай, когда мы хотим получить кластеризацию ссылок. Контексты будем хранить в отдельной таблице, которая будет выглядеть следующим образом:

<id_context> : <id_user> : <count>,

где id_context и id_user целочисленные идентификаторы пользователей, count целочисленное значение. Для двух пользователей мы будем подсчитывать количество ссылок, которые они посетили одновременно.

Следующим шагом будет нормализация получившихся значений. Эти значения будут представлены следующим образом:

<id_context> : <id_user> : <normal_value>,

где id_context и id_user целочисленные идентификаторы пользователей, normal_value число типа float. По каждому id_context сумма всех normal_value даст 1.

На рис.2.2.1 можно увидеть пример того, как выглядит один контекст для кластеризации ссылок. На рис. 2.2.2 представлен нормализованный вид контекста, который был представлен на рис. 2.2.1

id_context	id_user	count
15	15	8
15	208794	4
15	212156	4
15	455608	4
15	456064	4
15	500104	4
15	727175	4
15	759285	5
15	833561	4

Рис.2.2.1 Пример одного контекста для кластеризации пользователей.

id_context	id_user	norm_count
15	15	0.195122
15	208794	0.097561
15	212156	0.097561
15	455608	0.097561
15	456064	0.097561
15	500104	0.097561
15	727175	0.097561
15	759285	0.121951
15	833561	0.097561

Рис.2.2.2 Нормализованный вид контекста

Рассмотрим теперь второй случай, когда нас интересует кластеризация пользователей. Контексты так же будем хранить в отдельной таблице, которая будет выглядеть следующим образом:

<id_context> : <url> : <count>,

где id_context и id_user это строки по типу varchar, count целочисленное значение. Для двух ссылок мы будем подсчитывать количество пользователей, посетивших эти две ссылки одновременно. Аналогичным образом нужно нормализовать значения, чтобы получить:

<id_context> : <url> : <normal_value>.

Контексты для кластеризации пользователей строятся аналогичным образом с контекстами для кластеризации страниц, поэтому соответствующие примеры не были приведены.

Необходимо уточнить, что при практической реализации, в связи с ограниченность ресурсов, при подсчете контекстов по первому типу мною было установлено следующее ограничение:

count > 3,

таким образом я учитывала только те контексты, значение которых удовлетворяло данному условию, то есть два пользователя посетили не менее 4 ссылок одновременно. В общей сложности такая таблица содержала 3098314 записей.

После приведения к нормальному виду необходимо для каждого контекста вычислить энтропию. В моем случае таблица, описывающая информацию о контексте и вычисленной для него энтропии, содержала 41920 записей. Так же для дальнейшего получения узких контекстов необходимо в отдельной таблице хранить для каждого слова документную частоту.

Пример вычисленной энтропии для контекста с рис 2.2.1 можно увидеть в таб. 2.2.1.

id_context	entropy
15	3.12317

Таб.2.2.1 Энтропия контекста

2.3 Определение узких контекстов.

В данной работе я рассматривала 2 варианта разбиения контекстов на группы для получения необходимого кол-ва узких контекстов. Мы рассмотрим каждый из них.

1 вариант: в этом варианте предполагалось разбить имеющиеся контексты таким образом, чтобы каждая группы содержала одинаковое кол-во контекстов. Предварительно они были упорядочены по документной частоте.

2 вариант: в этом варианте предполагалось разбить имеющиеся контексты таким образом, чтобы каждая группа содержала суммарно одинаковую частоту. То есть по сути в каждой группе находится разное количество контекстов, в отличии от первого варианта.

Для решения данной задачи была написана программа, в которой пользователь указывал количество групп, на которые он хочет разделить контексты, и выбирал вариант разбиения контекстов. На выходе получал таблицу имеющую вид:

<id_group> : <id_context> : <frequency>: <entropy>,

каждая запись содержала информацию о том в какую группу попал контекст, частоту и энтропию. Эта информацию потребуется в дальнейшем, когда из каждой группы для кластеризации мы будем выбирать определенное количество контекстов. В момент выбора внутри каждой группы мы упорядочим контексты по энтропии. Поскольку нас интересуют только контексты с минимальной энтропией.

Пример разбиения контекстов для кластеризации страниц на группы по 1 и 2 варианту можно увидеть на рис 2.3.1 и 2.3.2.

id_group	id_context	frequency	entropy
1	423549	1135	11.1933
1	423551	1000	11.1976
1	441735	967	11.6061
1	818311	632	10.8088
1	704786	549	11.1031
1	177791	511	9.88505
1	423550	506	10.7737
1	293611	500	10.7757
1	3639	458	10.4932
1	844948	415	10.9603
1	355280	347	11.9755

Рис. 2.3.1 Использование 1 варианта разбиения на группы

	id_group	id_context	frequency	entropy
▶	2	46719	69	7.82741
	2	749190	69	8.27388
	2	273651	69	8.69874
	2	243677	68	0.590724
	2	515220	68	10.0221
	2	767066	68	7.28626
	2	353703	68	11.3124
	2	675481	68	9.12983
	2	236670	68	10.4479
	2	448097	68	7.98782
	2	753070	67	7.98515

Рис.2.3.2 Использование 2 варианта разбиения на группы

Приведенные примеры содержат лишь частичные данные выбранные случайным образом, чтобы показать пример промежуточного варианта. В связи с тем, что практической реализации кластеризации пользователей не производилось, примеров разбиения на группы контекстов приведено не будет. На вопрос, почему же была проведена только кластеризация страниц, можно будет найти ответ в главе 4.

Глава 3. Кластеризация на основе узких контекстов.

3.1 Расстояние Йенсена-Шеннона.

Перед тем как приступить к алгоритму контекстной документной кластеризации, необходимо уделить особое внимание следующему определению:

Расстоянием Йенсена-Шеннона между двумя вероятностными распределениями $p_1(u)$ и $p_2(u)$ называется число

$$JS_{\{k1,k2\}}[p_1, p_2] = H[\dot{p}] - k_1 H[p_1] - k_2 H[p_2],$$

$k_1 \geq 0$, $k_2 \geq 0$, $k_1 + k_2 = 1$, $\dot{p} = k_1 p_1 + k_2 p_2$

Расстояние Йенсена-Шеннона обладает свойствами:

1) Является неотрицательной ограниченной функцией от p_1 и p_2.
2) $JS_{\{k1,k2\}}[p_1, p_2] = 0$ тогда и только тогда, когда $p_1 \equiv p_2$
3) Является вогнутой функцией от p_1 и p_2 с единственным максимальным значением в точке $\{0.5,0.5\}$.

Схожесть документа и узкого контекста будет вычисляться как расстояние Йенсена-Шеннона между двумя вероятностными распределениями.

При вычислении расстояние Йенсена-Шеннона на практике мною были использованы коэффициенты $k_1 = k_2 = 0.5$.

3.2 Нахождение распределения ссылок и пользователей.

Для нахождения кластеризации ссылок нам необходимо найти распределение пользователей. Мы будем хранить это в отдельной таблице, которая будет выглядеть следующим образом:

<url> : <id_user> : <count>,

где url – строка по типу varchar, id_user – целочисленный идентификатор пользователя, count- целочисленное значение. Для каждой уникальной ссылки посчитать сколько раз ее посетил каждый пользователь. Пример для одной ссылки можно увидеть на рис.3.2.1

url	id_user	count
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	653715	1
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	323344	1
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	112063	1
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	786557	1
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	554565	1
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	806037	1
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	654229	1
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	334662	1
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	621321	1
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	587940	1
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	243664	1
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	204294	1
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	352966	1
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	198410	1
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	164517	1

Рис.3.2.1 Пример вычисления распределения пользователей.

Для нахождения кластеризации пользователей нам необходимо найти распределение ссылок, оно будет выглядеть следующим образом:

<id_user> : <url> : <count>,

где url – строка по типу varchar, id_user – целочисленный идентификатор пользователя, count- целочисленное значение. То есть для каждого уникального пользователя нужно подсчитать сколько раз он посетил каждую ссылку.

Теперь нам необходимо привести к нормальной форме и вычислить соответственно энтропии.

Пример нормальной формы для данных с рис. 3.2.1 можно увидеть на рис. 3.2.2.

url	id_user	norm_count
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	653715	0.0666667
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	323344	0.0666667
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	112063	0.0666667
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	786557	0.0666667
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	554565	0.0666667
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	806037	0.0666667
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	654229	0.0666667
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	334662	0.0666667
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	621321	0.0666667
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	587940	0.0666667
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	243664	0.0666667
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	204294	0.0666667
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	352966	0.0666667
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	198410	0.0666667
http://www.eteknix.com/amd-release-new-kaveri-apu-soon	164517	0.0666667

3.2.2 Нормализованные данные распределения пользователей.

Аналогичным образом производится вычисление распределение ссылок и нормализация этих данных.

3.3 Контекстная документная кластеризации на основе узких контекстов.

Для решения задачи кластеризации была написана программа, в которой пользователь указывал необходимое количество кластеров. На

выходе пользователь получал результат, хранившийся в отдельной таблице в следующем виде:

<id_center> : <url> : <distance>,

где id_center – целочисленный идентификатор аттрактора кластера, url – строка по типу varchar, distance – значение по типу float.

То есть для каждой ссылки нам известно к какому кластеру программа отнесла данную ссылку и на каком расстоянии от аттрактора она находится.

Программа состоит из следующих частей:

1) Определение аттракторов кластеров, равное числу, заданному пользователем.
2) Вычисление расстояния Йенсена-Шеннона между каждой ссылкой и каждым аттрактором.

Рассмотрим, как реализована первая часть программы:

В данном случае id_group – переменная, обозначающая номер группы, count_of_group – количество групп на которые были разбиты все кластеры, id_center – номера кластеров, которые выбираются из заранее отсортированной таблицы. О том, как должны быть отсортированы данные можно узнать в разделе 2.3 подробнее.

Вход: count_of_centers –число кластеров.

Выход: centers – таблица, содержащая все аттракторы кластеров.

for id_group = 1 **to** count_of_group **do**

insert into centers **select** id_center **limit** count_of_centers/count_of_group

end do

Теперь рассмотрим вторую часть программы:

В данном случае url – переменная, обозначающая ссылку, center – переменная, обозначающая аттрактор кластера, который берется из таблицы centers, подающейся на вход. $JS_{\{0.5,0.5\}}$ – переменная, равная расстоянию Йенсена-Шеннона между двумя вероятностными распределениями, где p1 – распределение пользователей(ссылок) и p2 – аттрактор. Подробнее о том, как вычислять расстояние Йенсена-Шеннона можно узнать в разделе 3.1, о распределении пользователей(ссылок) в разделе 3.2. Контекстам посвящена глава 2. В result содержатся информация к какому кластеру была отнесена соответствующая ссылка с расстоянием minJS. Данная информация заносится в таблицу clustering, отображающую результат кластеризации.

Вход: centers – таблица, содержащая все аттракторы кластеров.

Выход: clustering –таблица, содержащая данные о кластеризации по типу, описанному ранее в данном разделе.

for every url **do**

minJS =100

for every center **do**

вычислить $JS_{\{0.5,0.5\}}$ между p1 и p2

if($JS_{\{0.5,0.5\}} <$ minJS **) then**

minJS = $JS_{\{0.5,0.5\}}$

result = (center,url,minJS)

end if

end do

insert into clustering **value(**result**)**

end do

Пример получившейся кластеризации с разбиением кластеров на группы по 1 и 2 варианту можно увидеть на 3.3.1 и 3.3.2 соответственно. Стоит отметить, что данные, приведенные на картинках, получены при кластеризации ссылок с 150 аттракторами, и были показаны лишь те данные, которые удовлетворяют условию distance < 0.85, поскольку при большем расстоянии сложно говорить о корректности присвоения ссылки конкретному кластеру.

id_center	url	distance
293641	http://www.eteknix.com/new-sandybridge-processors-planned-to-be-released-later-this-month/	0.534878
293641	http://www.eteknix.com/qualcomm-snapdragon-610-615-announced-q3-2014-801-expected-quarter/	0.534878
293641	http://www.overclockersclub.com/reviews/intel__core_i7_980x/3.htm	0.534878
293641	http://www.overclockersclub.com/reviews/intel_corei7_3820/12.htm	0.534878
293641	http://www.overclockersclub.com/reviews/phenom2_x4_975_840/	0.534878
293641	http://www.pocket-lint.com/news/106817-creative-ziio-7-tablet-photos	0.534878
293641	http://www.pocket-lint.com/hub/tesco-hudl-2	0.534878
293641	http://www.pocket-lint.com/news/120126-philips-original-radio-mini	0.534878
293641	http://www.pocket-lint.com/news/115543-soundfreaq-sound-stack-pictures-hands-on	0.534878
293641	http://www.pocket-lint.com/news/115543-soundfreaq-sound-stack-pictures-hands-on/gallery	0.534878
293641	http://www.pocket-lint.com/news/116757-archos-101-xs-tablet-preview	0.534878
293641	http://www.pocket-lint.com/news/121465-hands-on-acer-iconia-w3-review	0.534878
293641	http://www.pocket-lint.com/news/125652-aldi-s-medion-lifetab-e7316-budget-jelly-bean-tablet-to-launch-8-d...	0.534878
293641	http://www.pocket-lint.com/news/125275-pure-evoke-d2-digital-radio-now-with-bluetooth-launching-this-chri...	0.534878
293641	http://www.pocket-lint.com/news/125869-hands-on-bayan-audio-soundbook-x3-review	0.534878
293641	http://www.pocket-lint.com/news/125869-hands-on-bayan-audio-soundbook-x3-review/gallery	0.534878
293641	http://www.pocket-lint.com/review/124424-microsoft-surface-pro-2-review	0.534878

Рис.3.3.1 Кластеризация (1 вариант разбиения контекстов на группы)

	id_center	url	distance
▶	243671	http://www.eteknix.com/android-wear-gets-major-update/	0.599725
	136245	http://dev.majorgeeks.com/files/details/pixia.html	0.637925
	136245	http://dev.majorgeeks.com/files/details/twistedbrush_open_studio.html	0.637925
	243664	http://www.eteknix.com/adata-dashdrive-air-ae800-500gb-wireless-hdd-power-bank-review/8/	0.581553
	243678	http://www.eteknix.com/aerocool-strike-x-x-1000-fan-controller-review/	0.743523
	243671	http://www.eteknix.com/aerocool-gt-s-white-full-tower-chassis-review/5/	0.599725
	243665	http://www.eteknix.com/akasa-venom-voodoo-cpu-cooler-review/	0.565508
	243670	http://www.eteknix.com/asrock-z87-extreme6-lga-1150-atx-motherboard-review/15/	0.538065
	243679	http://www.eteknix.com/asus-rog-vulcan-anc-pro-gaming-headset-review/	0.653526
	243665	http://www.eteknix.com/asus-vg278h-3d-vision-2-monitor-kit-review/all/1/	0.565508
	97629	http://www.eteknix.com/asus-sabertooth-990fx-am3-motherboard-review/5/	0.594404
	243665	http://www.eteknix.com/bitfenix-phenom-mini-itx-chassis-review/all/1/	0.565508
	243670	http://www.eteknix.com/coolermaster-devastator-mouse-keyboard-review/all/1/	0.538065
	828924	http://www.eteknix.com/corsair-vengeance-pro-ddr3-2400mhz-16gb-ram-kit-review/3/	0.59294
	828924	http://www.eteknix.com/corsair-vengeance-pro-ddr3-2400mhz-16gb-ram-kit-review/4/	0.59294
	243672	http://www.eteknix.com/crucial-ballistix-sport-xt-ddr3-1866mhz-16gb-memory-kit-review/5/	0.655931
	273929	http://www.eteknix.com/daniel-radcliff-set-to-star-in-upcoming-grand-theft-auto-movie/	0.575396

Рис.3.3.2 Кластеризация (2 вариант разбиения контекстов на группы).

Глава 4. Эксперименты и экспериментальные данные.

4.1 Программа, получающая статистику.

В ходе исследований была написана программа для проверки корректности кластеризации ссылок и получения экспериментальных данных. Как выглядит программа можно увидеть на рис.4.1.1

Стоить отметить, что кластеризация пользователей не была проведена в связи с тем, что не хватает информации о пользователях, чтобы оценить корректность выполненной кластеризации. Мы не знаем ни возраста, ни интересов ни любой другой информацию, которая была бы нам полезна при оценке кластеризации. Поэтому не имело смысла производить кластеризацию, когда нам известны только уникальные идентификаторы. Однако данная работа отвечает на вопрос о том, как произвести кластеризацию пользователей.

Данная программа для каждого кластера выбирает случайным образом 2-3 ссылки, в зависимости от количества ссылок, принадлежащих данному кластеру, а также случайным образом выбирает лишнюю ссылку из любого другого кластера. Пользователю, который проходит данный тест необходимо определить, какая из перечисленных ссылок является лишней и указать ее номер. Программа учтет корректность или некорректность данного ответа, выдаст пользователю информацию о его выборе и после полного прохождения занесет данные в специальную таблицу базы данных. Перед началом работы каждый пользователь вводит свою фамилию и имя, это является обязательным условием проведения теста.

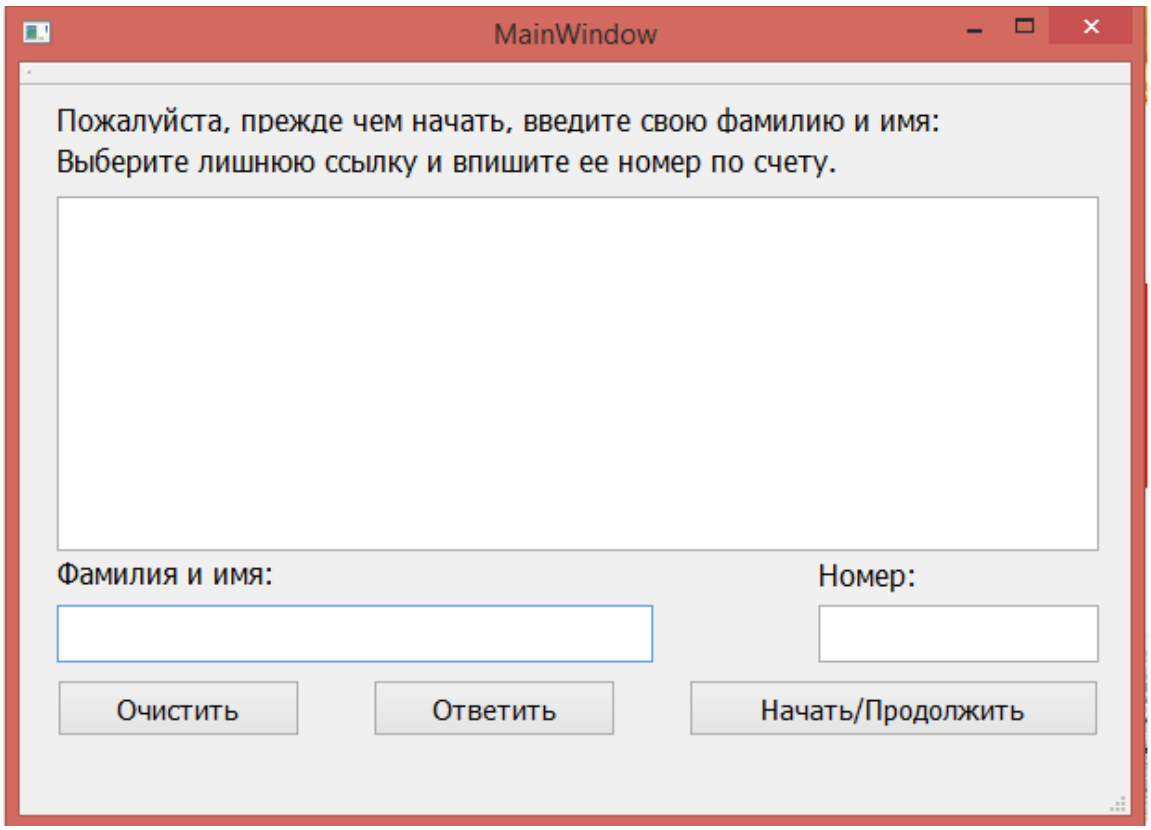

Рис. 4.1.1 Общий вид программы

В общем случае данные, которые мы получаем на выходе и хранящиеся в таблице выглядят следующим образом:

<fsname> : <variant> : <count_of_centers> : <right> : <wrong> : <percent>,

где fsname – фамилия и имя тестируемого, variant – вариант разбиения контекстов, подробнее можно прочитать в разделе 2.3, count_of_centers – количество аттракторов кластеризации, right – количество правильных ответов, wrong – количество неправильных ответов, percent – процент правильных ответов.

Пример работы программы вы можете видеть на рисунках 4.1.2 и 4.1.3. На 4.1.2 виден наглядный пример предоставляемого выбора, на рисунке 4.1.3 мы видим, как реагирует программа на выбор пользователя.

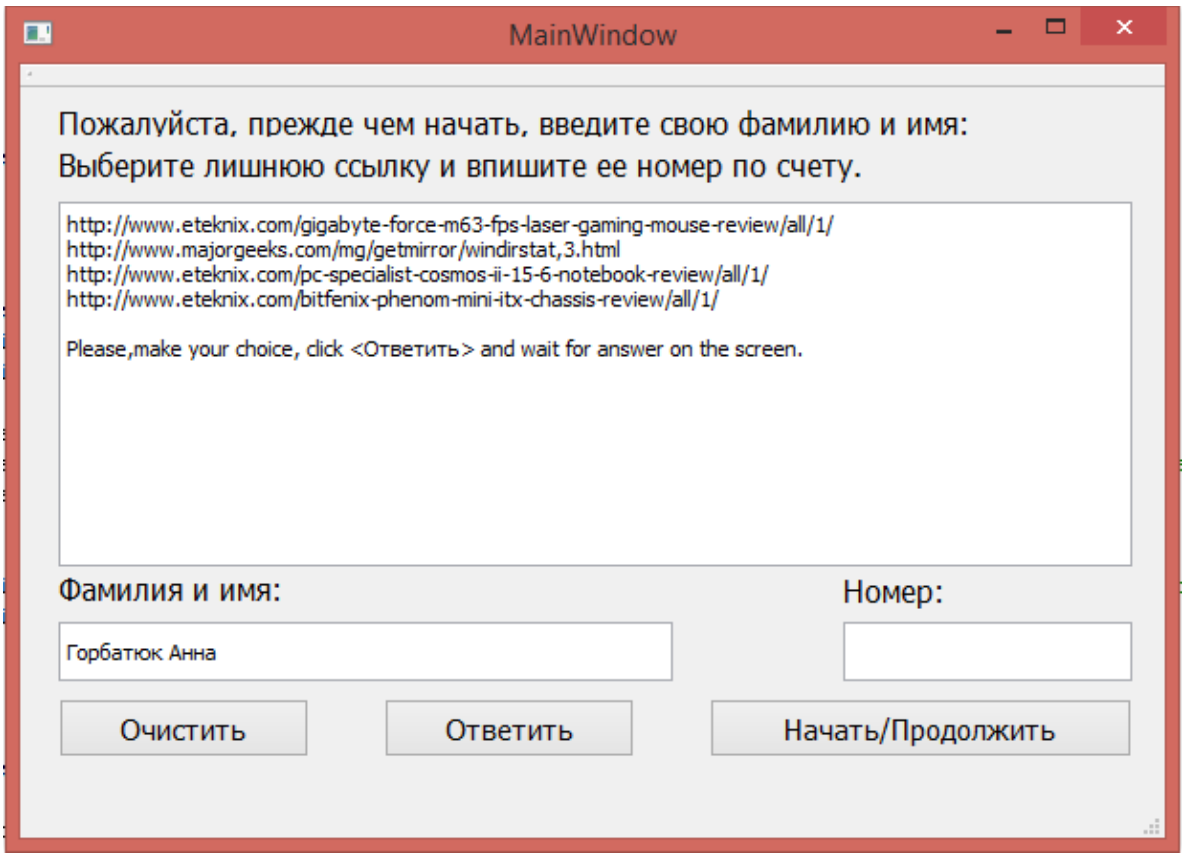

Рис. 4.1.2 До выбора пользователя.

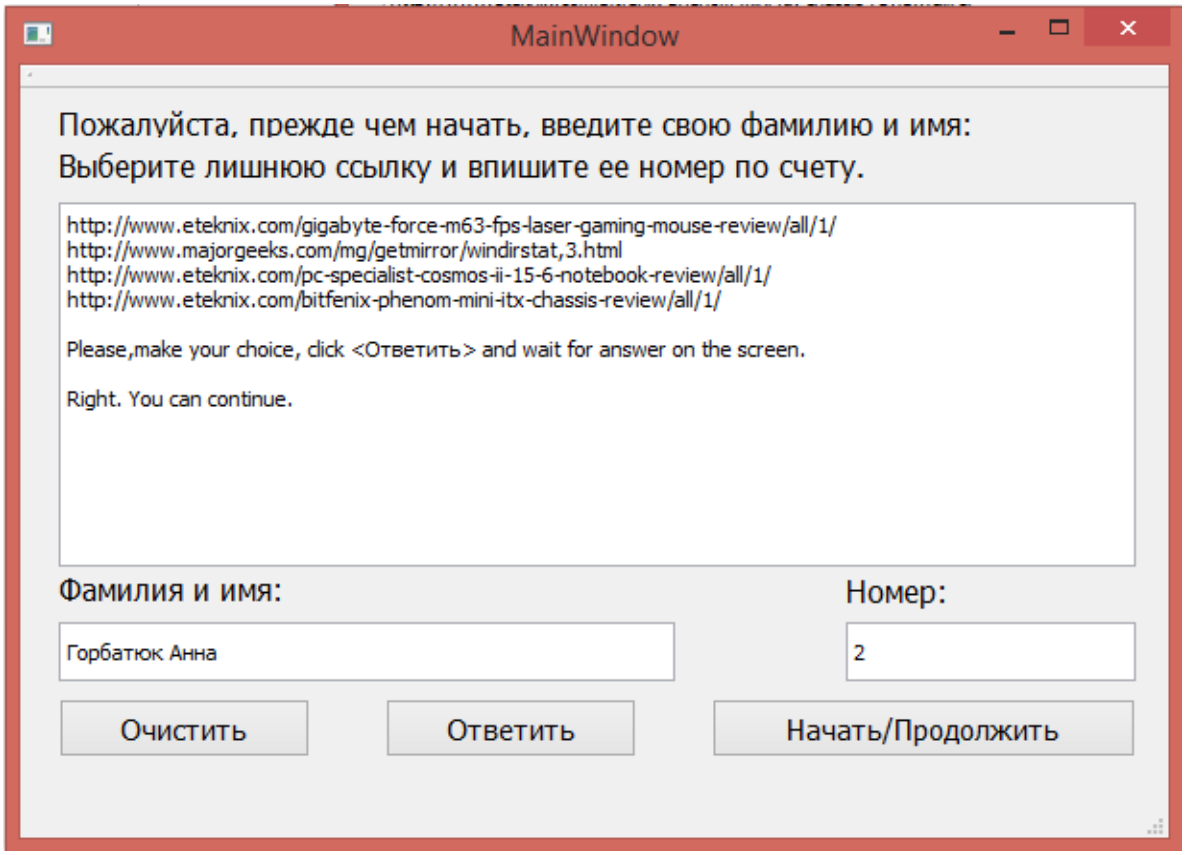

Рис. 4.1.3 После выбора пользователя.

Стоит учесть, что в таблице, получаемой после кластеризации, содержатся данные у которых минимальное расстояние до аттрактора кластера является довольно большим числом, поэтому данная программа обрабатывает только те данные, которые удовлетворяют условию:

$$distance < 0.85,$$

то есть в каждом кластере мы будем учитывать только те ссылки, расстояние до аттрактора у которых не превышает заданного числа. В обратном случае мы не можем утверждать о корректности присваивания ссылки данному кластеру из-за слишком большого расстояния.

Так же программа обрабатывает случаи, в которых кластеру принадлежит лишь одна ссылка, удовлетворяющая нашему условию, в этом случае мы так же не можем определить корректность присваивания данной ссылки данному кластеру, поэтому эти случаи так же не рассматриваются.

В конечном счете, после наложения всех условий программа имеет конечное число кластеров, в которых все ссылки удовлетворяют условию и количество ссылок для каждого кластера больше 1. Программа проверяет каждый такой кластер и просит пользователя сделать выбор, подсчитывает процент правильных ответов, по результатам работы данной программы мы можем утверждать о корректности проделанной кластеризации и проследить как изменение количества кластеров и выбор разбиения кластеров влияет на конечный результат.

4.2 Анализ полученных экспериментальных данных.

Используя программу, описанную в разделе 4.1, мне удалось получить некоторые экспериментальные данные. В моем исследовании принимали участие 4 человека. Результаты, которые удалось получить, отображены на следующей таблице 4.2.1. + и – помечены количество правильных и неправильных ответов соответственно, в колонке аттракторы указано с каким числом кластеров была выполнена кластеризация, в графе вариант указан тип разбиения контекстов.

Фамилия и имя	Вариант	Аттракторы	+	-	Процент

					ы
Горбатюк Анна	1	50	7	3	70
Сырбул Александра	1	50	9	1	90
Бертисова Людмила	1	50	7	3	70
Рыжкова Елизавета	1	50	7	3	70
Горбатюк Анна	2	50	13	3	81.25
Сырбул Александра	2	50	12	4	75
Бертисова Людмила	2	50	14	2	87.5
Рыжкова Елизавета	2	50	12	4	75
Горбатюк Анна	1	100	20	1	95.2381
Сырбул Александра	1	100	16	5	76.1905
Бертисова Людмила	1	100	16	5	76.1905
Рыжкова Елизавета	1	100	17	4	80.9524
Горбатюк Анна	2	100	23	3	88.4615
Сырбул Александра	2	100	19	7	73.0769
Бертисова Людмила	2	100	19	7	73.0769
Рыжкова Елизавета	2	100	18	8	69.2308
Горбатюк Анна	1	150	26	3	89.6552
Сырбул Александра	1	150	22	7	75.8621
Бертисова Людмила	1	150	25	4	86.2069
Рыжкова Елизавета	1	150	20	9	68.9655
Горбатюк Анна	2	150	34	4	89.4737
Сырбул Александра	2	150	32	6	84.2105
Бертисова Людмила	2	150	34	4	89.4737
Рыжкова Елизавета	2	150	30	8	78.9474

Таблица 4.2.1. Статистические данные

Стоит отметить, что благодаря тому, что в каждой ссылке содержится название страницы можно в целом оценить правильность и корректность кластеризации.

При лучших обстоятельства можно было бы отбросить самый большой и самый маленький процент и получить более точный средний процент правильных ответов в каждой категории, но в связи с тем, что в данном исследовании принимало участие всего 4 человека, учитываться будут все результаты.

Средний процент правильных ответов можно проследить в следующей таблице 4.2.2:

Вариант	Аттракторы	Средний процент
1	50	75
2	50	79.6875
1	100	82.142875
2	100	75.961525
1	150	80.172425
2	150	85.526325

4.2.2 Средний процент правильных ответов.

Можно увидеть, что при кластеризации с 50 и 150 кластерами лучше показал себя второй вариант разбиения, средний процент правильных ответов выше. Сказать почему второй вариант разбиения показал себя хуже первого при кластеризации с 100 кластерами очень сложно, возможно это связано с человеческим фактором, с некоторой невнимательностью тестируемых. Возможно это связано с тем, что все ссылки как бы то ни было связаны с технической тематикой и не все тестируемые хорошо разбираются в ней. В целом второй вариант разбиения показал себя более стабильным. Если мы посмотрим на количество кластеров, которые содержат больше одной ссылки и расстояние этих ссылок не превышает 0.85, то увидим, что по этому параметру второй вариант показал себя стабильно лучше во всех случаях.

В любом случае, оба варианта показали хороший результат.

Вывод.

В данной работе были решены задачи, которые ставились передо мной в самом начале работы. А именно бы изучен алгоритм контекстной документной кластеризации, рассмотрены кластеризация страниц и пользователей, посетивших эти страницы, с использование выбранного алгоритма. В ходе работы мы выяснили как находить контексты и узкие контексты, как происходит второй этап кластеризации на основе уже полученных узких контекстов. Так же был реализован алгоритм контекстной документной кластеризации для кластеризации ссылок. Проведены эксперименты с 4 испытуемыми и были получены экспериментальные данные. В разделе 4.2 приведены полученные данные и проведен их анализ.

Как итог данной работы мы имеем:

1) Контекстная документная кластеризация хорошо показала себя во всех испытаниях.
2) Наиболее высокий процент правильных ответов и стабильность показал второй вариант разбиения.
3) Первый вариант показал лучший результат при кластеризации со 100 аттракторами, второй вариант показал лучший результат при кластеризации со 150 аттракторами.
4) Средний процент правильных ответов не опускался ниже 75 процентов.

Заключение.

На данный момент все поставленные задачи были частично решены, получены промежуточные вычисления и проведен анализ. Однако стоит учесть тот факт, что область исследования темы контекстной документной кластеризации очень широка. Исследования в этой области можно продолжить дальше, для получения более точных данных и их анализа. Как пример, можно провести большее количество испытаний с различным количеством аттракторов, привлечь в исследования большее количество испытуемых, тем самым мы сможем получить более точный процент правильных ответов. Поэкспериментировать с разбиением на группы, посмотреть, как изменяется результат от выбора числа групп. Оценить более точно, имеется ли значительное превосходство второго варианта разбиения контекстов над первым. Данное работа имеет еще много направление для исследования. Так же в перспективе можно углубиться в изучение данного подхода и найти другие задачи, которые могу быть решены благодаря данному методу быть может более качественно, нежели классическими методами.

Список литературы.

1. Кристофер Д. Маннинг, Прабхакар Рагхаван, Хайнрих Шютце «Введение в информационный поиск», Москва, Санкт-Петербург, Киев, 2011.
2. Алексей Гринчук «Использование контекстной документной кластеризации для улучшения качества построения тематических моделей.» Бакалаврская работа, Московский государственный физико-технический университет, 2015.
3. Dobrynin V., Patterson D., Rooney N. «Contextual document clustering». In Proceeding of the 26th European Conference on Information Retrieval Research. Springer-Verlag Berlin Heidelberg, 2004.
4. Niall Rooney, David Patterson, Mykola Galushka, Vladimir Dobrynin, and Elena Smirnova «An investigation into the stability of contextual document clustering». JASIST, 2008.
5. Niall Rooney, Hui Wang, Fiona Browne, Fergal Monaghan, Jann Müller, Alan Sergeant, Zhiwei Lin, Philip Taylor, Vladimir Dobrynin «An Exploration into the Use of Contextual Document Clustering for Cluster Sentiment Analysis», Hissar, Bulgaria, 2011.
6. К. Дж. Дейт «Введение в системы баз данных». Москва, Санкт-Петербург, Киев, 2005.
7. В. Гольцман «MySQL 5.0», «Питер» Санкт-Петербург, 2010.
8. http://dev.mysql.com/doc/
9. Шлеев М. «Профессиональное программирование на c++ QT 4.8», Санкт-Петербург «БХВ-Петербург», 2012
10. http://doc.qt.io/
11. Jonathan Chang, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, David M. Blei «Reading Tea Leaves: How Humans Interpret Topic Models», Neural Information Processing Systems, 2009.