МОСКОВСКИЙ КОМИТЕТ ОБРАЗОВАНИЯ

ЛИЦЕЙ №1533 (ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ)

**Выпускная работа**по специальности «Прикладное программирование»  
учеников группы 11.4  
Гайдамашко Даниила Олеговича,  
Карпенко Максима Дмитриевича

Кластеризация  
результатов веб-поиска

**Научный руководитель:**   
Завриев Николай Константинович  
преподаватель лицея1533 (информационных технологий)

**Москва, 2015 г.**

Оглавление

[Введение 3](#_Toc429430519)

[Ввод в предметную область 4](#_Toc429430520)

[Постановка задачи 7](#_Toc429430521)

[Обзор аналогов 9](#_Toc429430522)

[Решение 13](#_Toc429430523)

[Ход работы над проектом 14](#_Toc429430524)

# Введение

Интернет – крупнейшая в мире информационная сеть, и, в связи с этим, поиск информации – одна из важнейших его функций. Поэтому не удивительно, что поисковые сайты (далее - поисковики) пользуются огромной популярностью. Как правило, использование поисковика удобно и позволяет достаточно быстро найти необходимую информацию, но довольно часто для достижения результата приходится изрядно «попотеть». Одной из основных причин этого является ситуация, когда по запросу мы получаем результаты, значительно расходящиеся по смыслу. Например, на запрос «кремень» поисковик выдаёт результаты двух типов: кремень-минерал и кремень-телесериал. Причём результаты эти смешаны и расположены хаотично, что, собственно, и является проблемой, усложняющей поиск. Конечно, можно поиск сузить, добавив в запрос дополнительные ключевые слова, однако это не решает проблему полностью, равно как и не всегда доступно. Как же нам выйти из сложившейся ситуации? Одно из решений – распределить результаты поиска по смысловым группам и давать пользователю возможность искать в той группе, которая ему нужна, и название этого решения – кластеризация результатов web-поиска.

# Ввод в предметную область

Первым делом, введём несколько терминов, которыми мы будем в дальнейшем пользоваться:

*Кластер* - класс родственных элементов статистической совокупности.

*Кластеризация результатов web-поиска* - группировка результатов поиска в поисковой системе по тому или иному признаку с целью сделать результат поиска более удобным.

Для выполнения кластеризации результатов поиска существует множество методов кластеризации. Ниже представлены основные методы с краткой характеристикой.

1. *Custom Search Folders* – этот метод позволяет сузить результаты поиска путём распределения их по «папкам» (folders). Выбором одной из предложенных папок пользователь сужает диапазон рассматриваемых объектов. Объектами в данном случае являются HTML ссылки. Папки имеют иерархическую структуру, что дает возможность всё более и более сужать результат поиска. По сути дела папки являются центроидами кластеров, к которым затем соотносятся документы (сайты). Процесс распределения по папкам занимает не много времени, потому что матрица близости документов уже есть, она как правило считается в режиме пре-процессинга. Благодаря этому названия папок имеют читаемый вид. Таким образом, система обладает высокой скоростью работы и хорошей наглядностью. Однако папки нуждаются в периодическом обновлении своей структуры, чтобы соответствовать количеству информации в сети.
2. *Suffix Tree Clustering*. Кластеры образуются в узлах специального вида дерева – суффиксного дерева, которое строится из слов и фраз входных документов. Достоинства метода: высокая скорость работы. По времени и занимаемой памяти дерево строится пропорционально количеству документов. Наихудшая теоретическая верхняя граница времени построения - пропорционально квадрату количества документов; хорошая наглядность представления результатов. Общие фрагменты текстов и фраз выступают в качестве названия кластеров, – это имеет большой смысл, т.к. не надо затрачивать дополнительных усилий для определения подходящего имени. Недостатки метода состоят в необходимости повторной обработки текстов документов.
3. *Латентно-семантический анализ (ЛСА)* — это метод обработки информации на естественном языке, анализирующий взаимосвязь между коллекцией документов и терминами в них встречающимися, сопоставляющий некоторые факторы (тематики) всем документам и терминам.

В основе метода латентно-семантического анализа лежат принципы факторного анализа, в частности, выявление латентных связей изучаемых явлений или объектов. При классификации / кластеризации документов этот метод используется для извлечения контекстно-зависимых значений лексических единиц при помощи статистической обработки больших корпусов текстов.

LSA/LSI - это реализация основных принципов факторного анализа применительно ко множеству документов. Кроме того, метод позволяет успешно преодолевать проблемы синонимии и омонимии, присущие текстовому корпусу. LSA позволяет преодолевать их, основываясь только на статистической информации о множестве документов/терминов.

Нами был выбран метод ЛСА для использования в дипломной работе, т.к. он не нуждается в предварительной настройке на специфический набор документов, его не надо обучать, а также это лучший метод для выявления латентных зависимостей. Недостаток метода – долгая обработка запросов, содержащих сотни тысяч объектов из-за огромного количества вычислений, пресекается ограниченным количеством обрабатываемых web-страниц.

Далее нам потребуется следующая терминология:

* *Стоп слова* - слова которые встречаются в каждом тексте и не несут в себе смысловой нагрузки, это, прежде всего, все союзы, частицы, предлоги и множество других слов.
* *Стемминг* - это процесс нахождения основы слова для заданного исходного слова. Основа слова необязательно совпадает с морфологическим корнем слова. Для стемминга используется *алгоритм Портера*. Главный плюс стеммера Портера заключается в том, что он не использует никаких словарей и выделение основы осуществляется путем преобразования слова согласно определенным правилам. Недостаток алгоритма в том, что в языках есть исключения, не подходящих под правила (неправильные глаголы в английском (buy-bought), одинаково оканчивающиеся слова разных частей речи с разными лексическими значениями в русском (плоть - колоть)).

Алгоритм метода ЛСА.

1. На входе мы получаем n-ное число текстов, анализ которых далее и будем проводить.
2. Выполняется обработка текстов, которая включает в себя:

* Удаление знаков препинания и других символов, не несущих семантического значения.
* Стемминг слов.
* Удаление стоп-слов.
* Удаление чисел.
* Удаление из текста слов, не встречающихся ни в одном из остальных текстов.

1. На основе оставшихся слов формируется частотная матрица. В этой матрице строки соответствуют индексированным словам, а столбцы — документам. В каждой ячейке матрицы указано, какое количество раз слово встречается в соответствующем тексте.
2. Выполняется сингулярное разложение частотной матрицы.

* *Сингулярным разложением* матрицы A размером MxN называется её представление в виде   
  A = U\*W\*V^T,  
  где U - ортогональная матрица размером MxM, V - ортогональная матрица размером NxN, W - матрица размером MxN, на главной диагонали которой находятся неотрицательные числа, расположенные в порядке убывания, а все внедиагональные элементы равны нолю. Диагональные элементы матрицы W называются сингулярными числами.

С учетом свойств матрицы W, большей частью состоящей из нулей, для получения матрицы A требуется не M столбцов матрицы U, а лишь первые min(M,N) столбцов (в примере выше - три столбца), аналогично, лишь первые min(M,N) строк матрицы V T влияют на результат произведения. Эти столбцы и строки называются левыми и правыми сингулярными векторами.

Достоинство сингулярного разложения состоит в том, что оно выделяет ключевые составляющие матрицы, позволяя игнорировать шумы. Согласно простым правилам произведения матриц, видно, что столбцы и строки соответствующие меньшим сингулярным значениям дают наименьший вклад в итоговое произведение.

1. На основе полученных в ходе сингулярного разложения матриц U и V^t формируется взвешенный граф:

По сути, в строках матрицы(U) содержатся координаты тега в неком многомерном пространстве, а в столбцах матрицы(V^t) - координаты текста в этом же пространстве. Таким образом, множество вершин нашего графа будет представлено множествами тегов и текстов с «привязанными» к ним координатами.

После формирования множества вершин, все вершины соединяются рёбрами, так, чтобы каждые две из них были соединены ребром.

1. В полученном графе происходит выделение кластеров при помощи метода минимального остовного дерева. В графе строится минимальное остовное дерево и удаляются все рёбра, в это дерево не входящие. После этого, путём удаления самых длинных рёбер, граф разделяется на некоторое количество кластеров, где кластером считается группа вершин, соединённых рёбрами.

# Постановка задачи

В нашем дипломном проекте мы поставили перед собой две задачи: первичную и вторичную. Первичная задача является обязательной, в то время как вторичную необходимо выполнить по мере возможностей, на основе полученного в ходе решения первичной опыта. Ниже представлены описания обеих задач.

Задача №1

Создать приложение, реализующее латентно-семантический анализ нескольких текстов и графически представляющее результаты анализа. В связи с этим, приложение должно:

1. Получать на входе несколько текстов.
2. Выполнять обработку текстов, предписанную методом ЛСА, с последующим формированием графа, множество вершин которого представлено тегами слов и текстами.
3. Выполнять выделение кластеров вершин при помощи метода минимального остовного дерева с возможностью регулировки коэффициента выделения пользователем.
4. Представлять граф, полученный в пункте (2) в графическом виде и иллюстрировать разделение вершин по кластерам.

Задача №2

Создать веб-сайт, устроенный по принципу стандартных интернет-поисковиков, который реализует кластеризацию результатов веб-поиска на запрос, введенный пользователем. Для этого ресурс должен:

1. Отправлять запрос в уже существующий поисковик (Google, Yandex, Yahoo, Rambler и пр.) и получать оттуда ответы на запрос.
2. Составление кластеров происходит путем обработки и тегирования веб-страниц — присваивания каждой набор тэгов. Чем больше тэгов совпадает, тем больше вероятность текстов веб-страниц быть на одну и ту же тему и находиться в одном кластере.
3. Выдавать результаты не линейно, как в изначальных поисковиках, а по группам в виде открывающихся папок. В каждой папке находится множество ссылок на веб-страницы, сходные по тематике и относящиеся к одному “главному объекту”.

# Обзор аналогов

**Полных аналогов ни графического инструмента кластеризации текстов, ни веб-инструмента кластеризации результатов интернет-поиска найдено не было, он в настоящий момент уже существуют веб-ресурсы, реализующие кластеризацию интернет-поиска, и, как следствие, упрощение поискового процесса, а также инструменты для выявления семантики текстов.**

|  |  |  |
| --- | --- | --- |
| Название инструмента | Достоинства | Недостатки |
| **Yyppi (ранее Clusty)** | * Кластеризация предварительно отобранных результатов поиска проводится по трем критериям группировки:   - кластеризация по наиболее часто встречаемым в результатах поиска ключевым словам;  - группировка результатов поиска по поисковым системам, в которых были найдены результаты поиска;  - группировка результатов поиска по доменным зонам; * Результаты кластеризации по наиболее часто встречаемым в результатах поиска ключевым словам представляются в виде списка пунктов меню, по которым возможна пересортировка результатов выдачи; * При отображении кластеры упорядочиваются по статистике найденных слов; * Возможность выделения слов, часто встречающихся совместно со словами поискового запроса; | * Поиск с учетом словоизменения (учет морфологии) реализован только для английского языка; * Поиск по энциклопедиям реализован только на основе использования метапоиска по англоязычной части энциклопедии wkipedia, даже если запрос пользователя введен на русском языке; * Кластерный анализ не позволяет получать точные результаты поиска при вводе запроса пользователя на естественном языке; * Метапоисковая система не использует ни синтаксического, ни семантического анализа текстов; |
| **Compreno** | * Лексический, морфологический, синтаксический и семантический анализы текстов; * Использование Универсальной Семантической Иерархии (УСИ), способной описывать явления от общего к частному; * Подбор слов для перевода осуществляется из понятийного набора, который находится на ветке универсального семантического дерева и содержит в себе аналоги слова, в т.ч. и из второго языка; * Каждое слово из переводимого предложения описывается максимальным набором понятийных эквивалентов на всех уровнях смысловой иерархии, поскольку нижестоящие элементы системы по иерархии наследуют признаки вышестоящих элементов. | * Невозможность применения семантико-синтаксического анализа в массовых поисковых системах из-за очень высоких требования к компьютерным мощностям, необходимым для обработки и индексации информационных массивов на понятийном уровне. |
| **Nigma** | * Кластерный метапоиск по документам на русском и английском языке. * Тщательный учет специфики русскоязычных запросов. * Мощная система исправления орфографических ошибок и опечаток в запросах. * Возможность выбирать поисковики, на которые отправляются запросы. * Альтернативный AJAX-интерфейс, в котором кластеры отображаются в виде облака тегов. * Метапоиск по базам изображений выбранных поисковиков; |  |

# Решение

## Математическая модель

## Ход работы над проектом

Работа над проектом, а именно над графическим инструментом, была начата на языке C# в среде разработки Visual Studio с помощью Microsoft .NET Framework и интерфейса программирования приложений Windows Forms.

Изначальная идея заключалась в том, чтобы в одну форму приложения вводились тексты, которые бы затем обрабатывались (удалялись знаки препинания, стоп-слова, происходил стемминг оставшихся слов), и затем формировалась частотная матрица повторяющихся стемов в разных текстах. Далее происходило бы сингулярное разложение данной матрицы, и <……..>. По данным

Для реализации данного плана была найдена кросс-платформенная библиотека численного анализа ALGLIB, содержащая необходимый нам метод сингулярного разложения матриц. Был найден отдельный список стоп-слов на английском и русском языках, и последний был дополнен вручную. Затем были найдены библиотеки, реализующие алгоритм стемминга Портера для русского и английского языков<…..>

Сравнив характеристики различных алгоритмов стемминга, особенно скорости их работы, мы выбрали

#TODO

Для реализации веб-инструмента было принято решение использовать популярный веб-фреймворк Django для языка Python, поскольку данный язык предназначен для быстрой разработки приложений и подходит для создания веб-сайта больше, чем C#. Кроме того, существует огромное количество готовых модулей на Python, среди которых есть и те, что могут помочь нам в решении поставленной задачи.

Поскольку библиотеки кластеризации введенных текстов мы к моменту начала работы над сайтом реализовали на C#, то изначально было принято решение об интеграции этих двух разных языков – Python и C#. Решение мы нашли в языке Ironpython, который и является интеграцией языка Python и .NET Framework. Однако, решив проблему интеграции Ironpython и C# библиотек, мы столкнулись с