УДК 004.738.52

# МЕТОДЫ ВИЗУАЛИЗАЦИИ РЕЗУЛЬТАТОВ ПОИСКА В ИНТЕРНЕТЕ

**Романовская Наталья Вадимовна[1], Сычёв Пётр Павлович[2]**

*[1]Студентка;*
*ГБОУ ВПО «Международный Университет природы, общества и человека «Дубна»,*
*Институт системного анализа и управления;*
*141980, Московская обл., г. Дубна, ул. Университетская, 19;*
*e-mail: natalia.romanovskaya@gmail.com.*

*[2]Доцент Института системного анализа и управления;*
*ГБОУ ВПО «Международный Университет природы, общества и человека «Дубна»,*
*Институт системного анализа и управления;*
*141980, Московская обл., г. Дубна, ул. Университетская, 19;*
*e-mail: sychov@dubna.ru.*

*Основной задачей поисковой системы является предоставление пользователю необходимой информации, но не всегда нужная информация располагается на первых страницах поисковой выдачи. Чем больше информации сможет рассмотреть пользователь, тем выше вероятность нахождения необходимых данных. Если эта информация будет расположена в виде списка, то процесс поиска будет длительным. Объединение результатов в группы и последующее представление их пользователю должно повысить качество и скорость поиска. Для этого в работе рассмотрен алгоритм кластеризации на основе суффиксного дерева, а также классификация результатов поиска по различным критериям.*

Ключевые слова: классификация, алгоритм кластеризации, поисковая система, суффиксное дерево, визуализация.

# METHODS OF VISUALIZATION OF SEARCH RESULTS ON THE INTERNET

**Romanovskaya Natalia[1], Sychov Peter[2]**

*[1]Student;*
*Dubna International University of Nature, Society, and Man,*
*Institute of system analysis and management;*
*141980, Dubna, Moscow reg., Universitetskaya str., 19;*
*e-mail: natalia.romanovskaya@gmail.com.*

*[2]Associate professor of Institute of system analysis and management;*
*Dubna International University of Nature, Society, and Man,*
*Institute of system analysis and management;*
*141980, Dubna, Moscow reg., Universitetskaya str., 19;*
*e-mail: sychov@dubna.ru.*

*Basic problem of a search engine is to provide a user necessary information, but the necessary information is not always located on the first pages of search delivery. The more information the user can consider, the higher is probability of finding necessary data. If this information is presented in the form of a list, the process of searching will take longer. Combining the results into groups and then presenting them to the user should increase quality and speed of search. For this in work the algorithm of clustering STC, and classification of search results were reviewed.*

Keywords: classification, algorithms of clusterization, search engine, suffix tree, visualization.

## *Введение*

Для поиска необходимой информации сейчас все чаще используют интернет и как следствие различные поисковые системы. Основной задачей поисковой системы является предоставление пользователю корректной информации по его запросу. Наиболее популярным способом представления информации является список найденных сайтов, располагающихся последовательно на страницах. На рис. 1 представлен результат поиска по запросу «процессор» в поисковой системе *Google*.

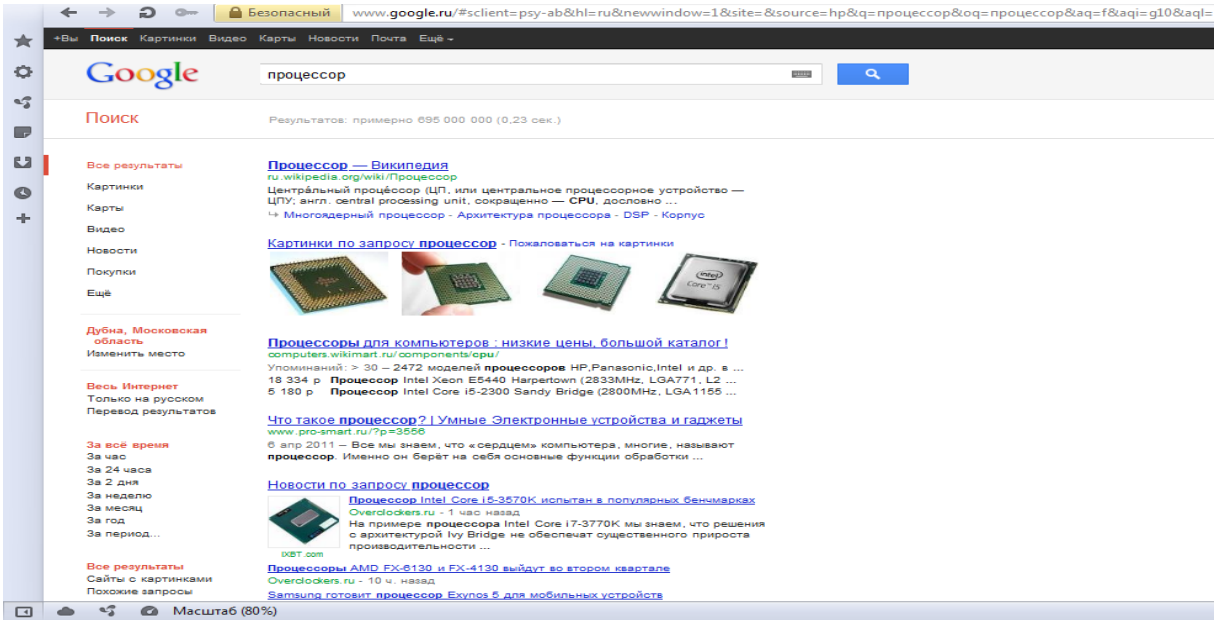


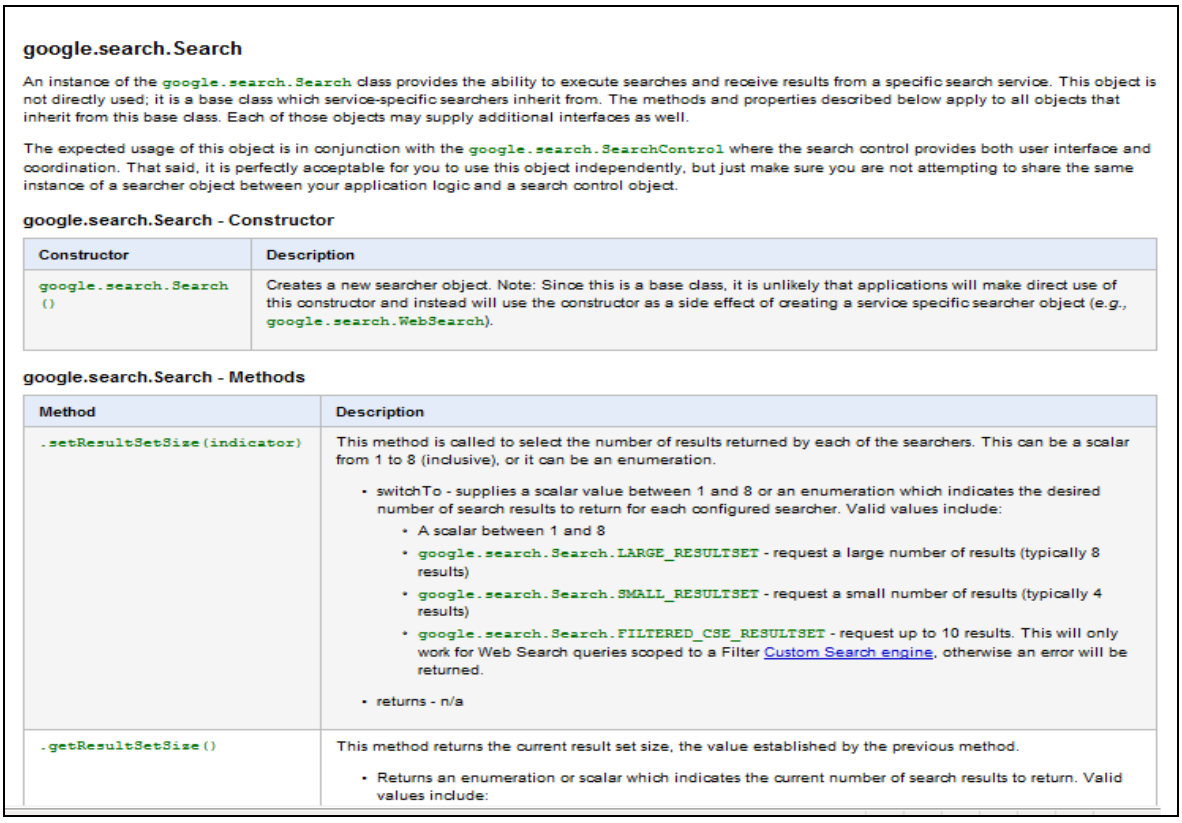

*Рис. 1. Результат поиска по запросу «процессор», полученный от Google*

В некоторых случаях необходимая информация может располагаться не на первых страницах, и пользователь, просто не увидев её, может завершить поиск. По этой причине для улучшения процесса поиска необходимо давать пользователю возможность просматривать большее количество результатов одновременно. Однако если эти сайты будут располагаться в традиционной форме списка, поиск необходимой информации будет затруднительным. Для улучшения процесса поиска полученные результаты можно объединять в группы и представлять в удобном пользователю виде.

В данной работе для группировки результатов используются методы классификации по различным критериям, а так же алгоритм кластеризации на основе суффиксного дерева (*STC*) [1].

## *Средства, предоставляемые поисковыми системами, для получения результатов поиска*

Такие поисковые системы как Яндекс, *Google*, *Bing* и другие предоставляют различные *API*, позволяющие получать результаты поисковых запросов. Поиск может производиться как по всему интернету, так и по конкретному сайту. Помимо этого можно осуществлять не только веб поиск, но и поиск по изображениям, видео, новостям и т.д. Для этого необходимо создать аккаунт в поисковой системе и получить адрес с уникальным номером, используя который можно отправлять запросы к поисковой системе. Данные, возвращаемые поисковой системой хранятся, в *XML* файле, из которого их можно извлекать и в дальнейшем использовать. В рамках этого исследования были использованы результаты, получаемые от следующих поисковых систем: Яндекс, *Google* и *Bing*. На рис. 2 приведён пример описания *API* поисковой системы *Google* [2].



**google.search.Search**

An instance of the `google.search.Search` class provides the ability to execute searches and receive results from a specific search service. This object is not directly used; it is a base class which service-specific searchers inherit from. The methods and properties described below apply to all objects that inherit from this base class. Each of those objects may supply additional interfaces as well.

The expected usage of this object is in conjunction with the `google.search.SearchControl` where the search control provides both user interface and coordination. That said, it is perfectly acceptable for you to use this object independently, but just make sure you are not attempting to share the same instance of a searcher object between your application logic and a search control object.

**google.search.Search - Constructor**

| Constructor | Description |
|---|---|
| `google.search.Search()` | Creates a new searcher object. Note: Since this is a base class, it is unlikely that applications will make direct use of this constructor and instead will use the constructor as a side effect of creating a service specific searcher object (*e.g.*, `google.search.WebSearch`). |

**google.search.Search - Methods**

| Method | Description |
|---|---|
| `.setResultSetSize(indicator)` | This method is called to select the number of results returned by each of the searchers. This can be a scalar from 1 to 8 (inclusive), or it can be an enumeration.<br>• switchTo - supplies a scalar value between 1 and 8 or an enumeration which indicates the desired number of search results to return for each configured searcher. Valid values include:<br>  ◦ A scalar between 1 and 8<br>  ◦ `google.search.Search.LARGE_RESULTSET` - request a large number of results (typically 8 results)<br>  ◦ `google.search.Search.SMALL_RESULTSET` - request a small number of results (typically 4 results)<br>  ◦ `google.search.Search.FILTERED_CSE_RESULTSET` - request up to 10 results. This will only work for Web Search queries scoped to a Filter Custom Search engine, otherwise an error will be returned.<br>• returns - n/a |
| `.getResultSetSize()` | This method returns the current result set size, the value established by the previous method.<br>• Returns an enumeration or scalar which indicates the current number of search results to return. Valid values include: |

*Рис. 2. API поисковой системы Google для получения результатов поисковых запросов*

Однако при использовании *API* существует ряд ограничений. К примеру, компания *Google* ограничивает максимальное количество результатов, которое можно получить – 64 результата, а компания Яндекс позволяет делать не более 1000 запросов к поисковой системе в день. Снятие этих ограничений возможно в случае заключения договора с конкретной поисковой системой.

## *Классификация и кластеризация результатов поиска*

Использование классификации результатов позволяет пользователю получить средство навигации по множеству документов. Таким образом, любой пользователь может рассмотреть интересующую его группу документов, что позволит сократить время поиска. Для каждого типа поиска (видео, изображения и др.) выбраны свои критерии классификации. Для результатов веб поиска: классификация по формату страниц, возвращаемых поисковой системой (*.doc*, *.pdf*, *.ppt* или *.html*). Для результатов поиска по видео файлам: классификация файлов по длительности видео. Для поиска по изображениям: классификация изображений по размеру (маленькие, средние и большие), а так же по формату изображения (*.jpg*, *.bmp*, *.png* и *.gif*). Для поиска по новостным статьям: классификация новостей по дате публикации (новости за сегодня, за прошедшую неделю и т.д.). Выбор критериев для классификации основан на анализе информации, которую можно извлечь из результата поискового запроса. Результат классификации представлен в виде списка. На рис. 3 показаны веб документы в формате *.doc*, полученные в результате классификации результатов веб поиска.

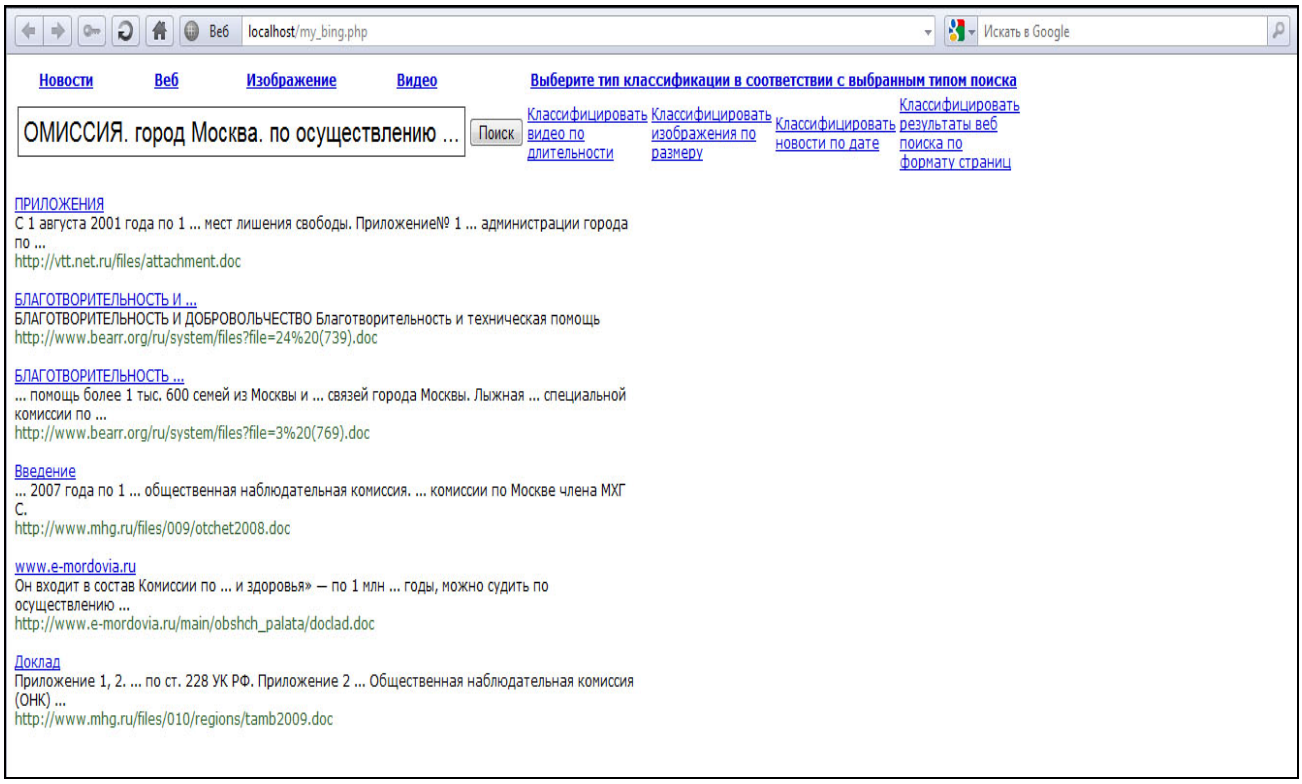



*Рис. 3. Документы в формате .doc, полученные в результате классификации результатов поиска*

Для кластеризации используются как полученные от поисковой системы результаты, так и поисковые результаты, которые были предварительно классифицированы. Результатом работы алгоритма кластеризации на основе суффиксного дерева будут кластеры, содержащие ссылки на документы, в которых встречается фраза, описывающая кластер. Таким образом, пользователь может рассматривать только интересующие его группы документов на основании их описания.

Алгоритм кластеризации *STC* является модификацией алгоритма Укконена (алгоритма построения суффиксного дерева) [3, 4]. Основными достоинствами алгоритма *STC* являются линейная скорость построения дерева, в качестве названия кластера используется фраза, встречающаяся в документах, принадлежащих этому кластеру, а так же то, что документы рассматриваются как последовательность связанных слов, а не как множество элементов.

Принцип работы алгоритма состоит в следующем. Для документов, получаемых в ответ на запрос к поисковой системе, строится дерево. Единицей, находящейся на рёбрах дерева, является слово или словосочетание. Каждой вершине дерева соответствует фраза. Её можно получить, объединив все слова/словосочетания, находящиеся на рёбрах на пути от корня дерева к данной вершине. В вершинах дерева, имеющих потомков, расположены ссылки на документы, в которых встречается фраза, соответствующая вершине. Множества документов, на которые указывают эти ссылки, образуют базовые кластеры. Затем происходит укрупнение базовых кластеров и получение окончательного набора кластеров.

Работа алгоритма кластеризации состоит из выполнения трёх этапов [1]:

1. Предварительная обработка документа. Этот этап включает в себя: удаление знаков препинания, чисел, иностранных символов, приведение всех слов к одному регистру, а также удаление стоп-слов. Стоп-слова – самые часто встречающиеся слова в тексте, такие как: предлоги, союзы, частицы, которые не оказывают влияния на тематику документа. Помимо удаления стоп-слов нужно произвести процедуру стемминга (от английского *stem* – корень). Данная процедура заключается в выделении значимой части слова. С помощью стемминга слова «алгоритм» и «алгоритму» будут приведены к одному слову «алгоритм», что позволит уменьшить число базовых кластеров и повысит качество работы алгоритма. Для реализации процедуры стемминга был выбран стемминг Портера.

2. Построение суффиксного дерева на основании обработанных документов. Результатом выполнения этого этапа являются базовые кластеры. Каждая вершина дерева, у которой больше чем один потомок, представляет собой базовый кластер.

3. Объединение базовых кластеров. Базовые кластеры могут перекрываться или быть идентичными. Для того, чтобы избежать это, производится объединение базовых кластеров. Для сравнения кластеров определена бинарная мера сходства между базовыми кластерами на основе перекрытия наборов документов базовых кластеров. Пусть есть два базовых кластера $B_m$ и $B_n$, где $|B_m|$ и $|B_n|$ – количество документов в них соответственно, а $|B_m \cap B_n|$ – количество документов, общих для обоих базовых кластеров. Мера сходства между кластерами равна 1, когда выполняются следующие условия:

$$|B_m \cap B_n|/|B_n| > 0.5 \text{ и } |B_m \cap B_n|/|B_m| > 0.5. \quad (1)$$

В противном случае мера сходства равна 0.

Объединённый базовый кластер будет содержать документы, содержащиеся в обоих кластерах (мера сходства этих кластеров равна 1).

Пример построения суффиксного дерева для трех строк показан на рис. 4.

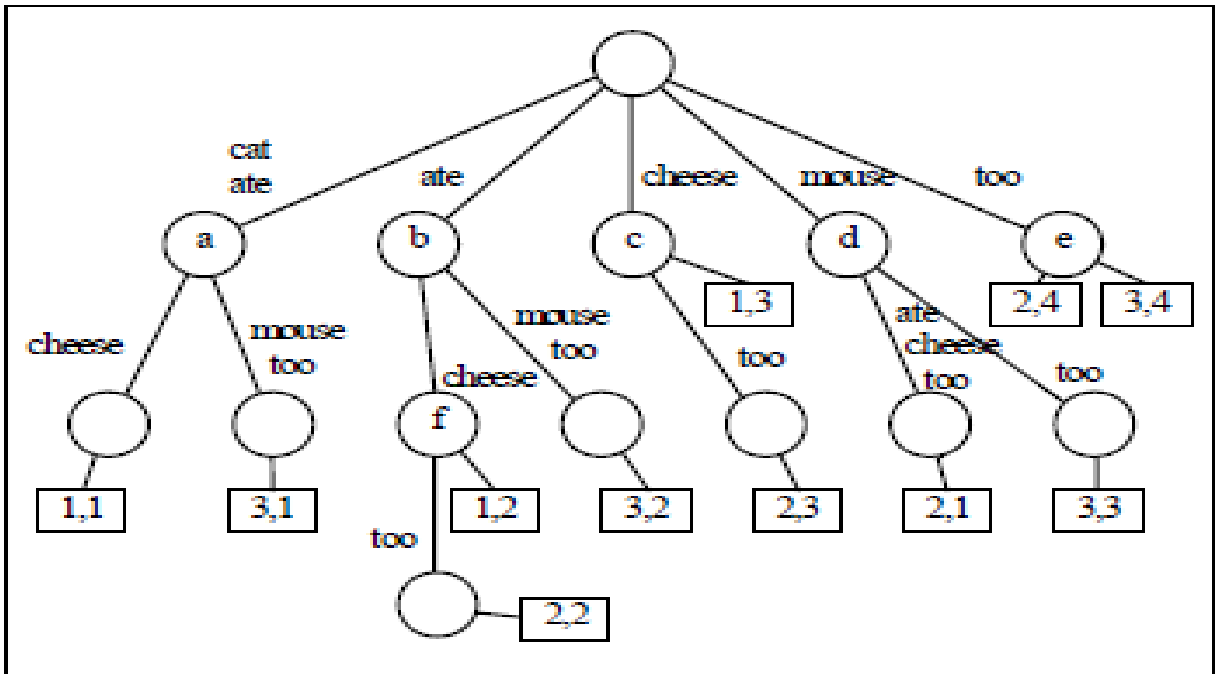



*Рис. 4. Суффиксное дерево для трех строк: «cat ate cheese», «mouse ate cheese too», «cat ate mouse too»*

## *Визуализация результатов поиска*

На основании *API* поисковых систем был создан сервис, реализующий выбранные методы классификации и кластеризации, и представляющий полученные результаты в удобной для пользователя форме.

На рис. 5 показан результат кластеризации результатов веб поиска по запросу «инкапсуляция». По центру располагается кластер, содержащий наибольшее количество документов. По мере удаления от центра количество документов в кластерах уменьшается. Фраза, описывающая кластер с большим количеством документов, будет иметь больший размер шрифта (величина шрифта пропорциональна количеству документов в кластере). При наведении курсора на кластер, он будет увеличиваться и будет выведена информация о количестве документов в данном кластере. Справа за линией располагаются результаты поиска, не вошедшие ни в один кластер. При наведении курсора на результат поиска будет выведено его описание.

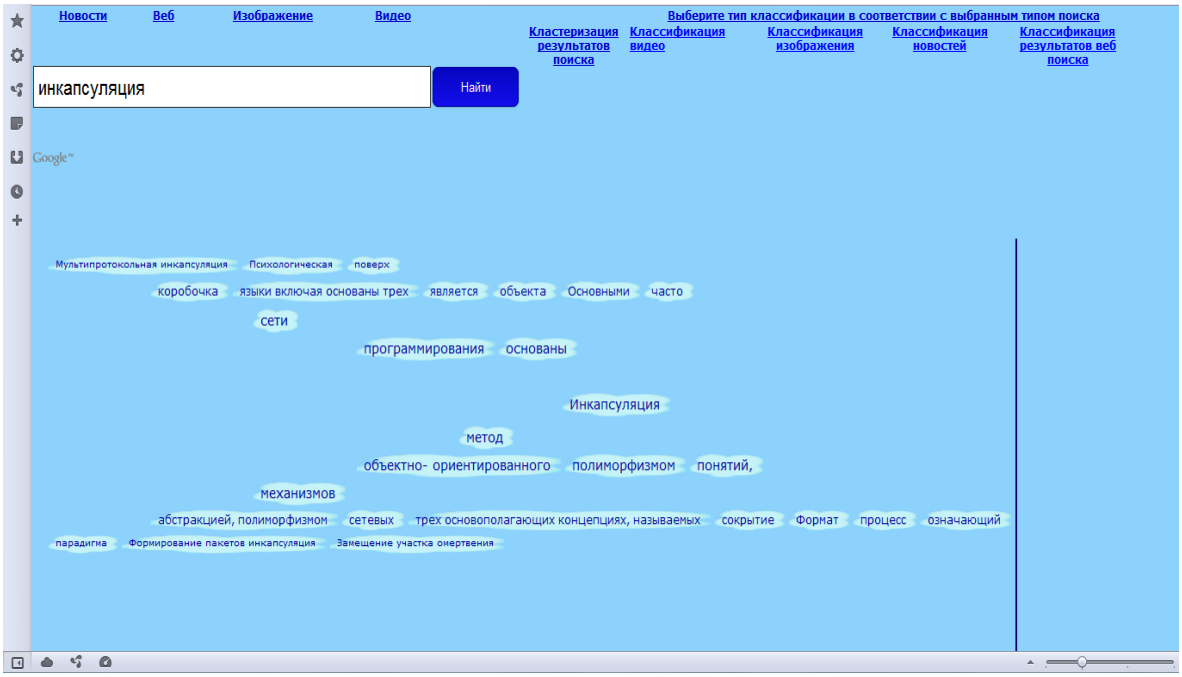



*Рис. 5. Кластеризация результатов веб поиска по запросу «инкапсуляция»*

На рис. 6 показана кластеризация результатов поиска изображений по запросу «процессор». В отличие от поиска по веб документам, при наведении на изображение оно будет увеличиваться и так же будет показано его описание.

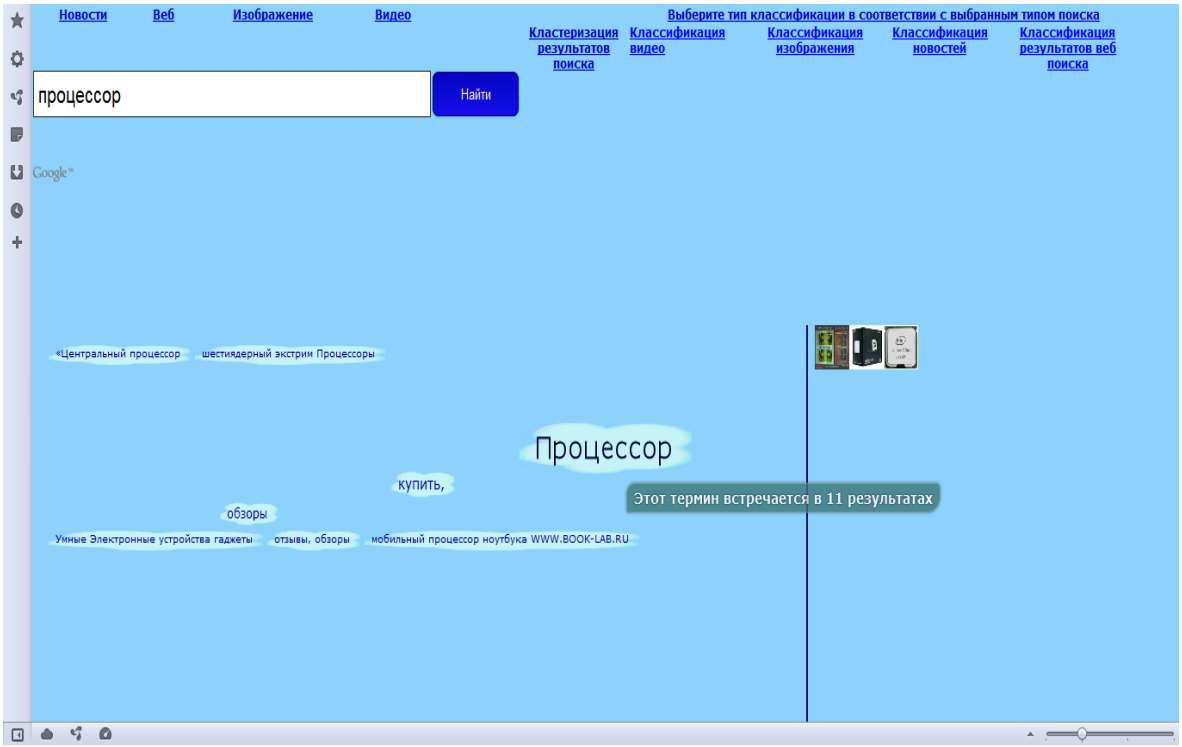



*Рис. 6. Кластеризация результатов поиска изображений по запросу «процессор»*

Помимо этого, если нажать правой кнопкой мыши на изображение, их можно будет последовательно просмотреть (рис. 7), начиная с выбранного изображения.

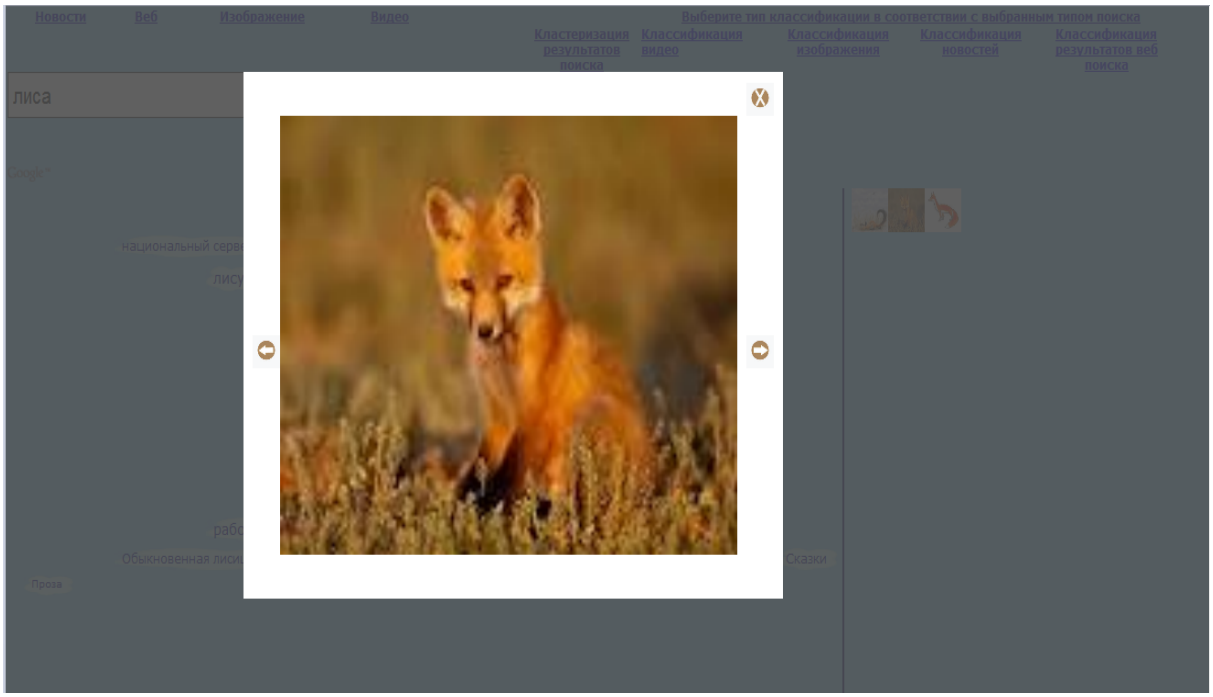



*Рис. 7. Последовательный просмотр изображений*

## *Заключение*

В ходе исследования были рассмотрены способы объединения информации, полученной от поисковой системы, позволяющие улучшить процесс поиска. Рассмотренные средства визуализации результатов поиска предоставляют пользователю возможность удобного и эффективного поиска среди большого количества результатов, получаемых от поисковой системы.

## *Список литературы*

1. Zamir O., Etzioni O. Web document clustering: a feasibility demonstration, in // Proceedings of the 19th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98). – 1998. – Pp. 46-54.
2. Google Ajax Search API. – [Электронный ресурс]. URL: https://developers.google.com/web-search/docs/reference?hl=ru-RU.
3. Ukkonen E. On-line construction of suffix-trees // Algorithmica 14. – 1995. – Pp. 249-260.
4. Гасфилд Д. Строки, деревья и последовательности в алгоритмах: Информатика и вычислительная биология / Пер. с англ. И.В. Романовского. – Спб.: Невский диалект; БХВ-Петербург, 2003. – С. 654.