МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Навчально-науковий інститут інноваційних освітніх технологій
Кафедра комп'ютерної інженерії

ГРИЧАННИК Ігор Романович

Пошукова оптимізація сайтів інтелектуальними засобами / Intellectual Tools for Search Engine Optimizations

спеціальність: 123 - Комп’ютерна інженерія
магістерська програма - Комп’ютерна інженерія

Магістерська робота

Виконав студент групи КІм-21
І. Р. Гричанник
Науковий керівник: к.т.н., доцент,
О. П. Адамів

Магістерську роботу допущено до захисту:

ТЕРНОПІЛЬ -2018

РЕЗЮМЕ

Магістерська робота на тему “Пошукова оптимізація сайту інтелектуальними засобами” на здобуття освітньо-кваліфікаційного рівня “Магістр” зі спеціальності “Комп’ютерні системи та мережі” написана обсягом 118 сторінок і містить 25 ілюстрацій, 2 таблиці, 4 додатки та 56 джерел за переліком посилань.

Метою роботи є розроблення програмного продукту, що дозволяє виконати процес пошукової оптимізації сайту з допомогою інтелектуальних засобів.

Методи досліджень. Для розв’язання поставлених задач у магістерській роботі використано: методи: методи системного аналізу, статистичні методи, методи інтелектуального аналізу даних.

Результати дослідження: алгоритм збору та кластеризації семантичного ядра сайту, на основі методів інтелектуального аналізу даних за допомогою алгоритму пошуку популярних наборів в базі даних пошукових запитів.

Результати роботи можуть бути впроваджені в системи адміністрування сайтів або в засоби підтримки роботи seo-фахівців для підвищення повноти, точності і зниження часу розробки семантичного ядра сайтів.

КЛЮЧОВІ СЛОВА: ПОШУКОВА ОПТИМІЗАЦІЯ, СЕМАНТИЧНЕ ЯДРО, АЛГОРИТМ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, АНАЛІЗ, ПОШУКОВИЙ ЗАПИТ, ПРОГРАМНА СИСТЕМА.

RESUME

Diploma work: "Intellectual Tools for Search Engine Optimizations" for obtaining the educational qualification of "Master" specialty "Computer systems and networks" written volume contains 118 pages and 25 illustrations, 2 tables, 4 applications and 56 sources for references .

The aim is to develop software that allows you to perform search engine optimization process using intelligent agents.

Research methods. To solve the tasks of master's thesis used: methods of system analysis, statistical methods, methods of data mining.

Results: clustering algorithm to collect and semantic site, based on data mining techniques using popular search algorithm sets of database searches.

The results can be implemented in the system of administration sites or means of support seo-experts to improve the completeness, accuracy and reduced development time semantic websites.

KEYWORDS: SEARCH ENGINE OPTIMIZATION, SEMANTIC CORE, ALGORITHMS, DATA MINING, ANALYSIS, SEARCH QUERIES, SOFTWARE SYSTEMS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

SEO – пошукова оптимізація;

API – інтерфейс програмування додатків;

URL – уніфікований локатор ресурсів;

K-means – метод кластеризації;

TF-IDF – статистичний показник;

ТОВ, ТзОВ – товариство з обмеженою відповідальністю;

HTML – мова розмітки гіпертексту;

IP – міжмережевий протокол;

PHP – скриптова мова програмування;

DNS – система доменних імен;

СУБД – система управління базами даних;

БД – бази даних;

MySQL – багатопотокова, розрахована на багато користувачів СУБД;

SQL – мова структурованих запитів;

IT – інформаційні технології;

XML – розширювана мова розмітки;

ТІЦ – тематичний індекс цитування;

PR – Page Rank;

SQLite – вбудований двигун баз даних.

ВСТУП

В умовах розвитку глобального інформаційного простору, коли виникають унікальні можливості в комунікації і інформаційному обміні, все більшого значення набуває мережа Інтернет, як засіб донесення інформації до широких мас. При цьому головним джерелом інформації в мережі виступають веб-сайти, на які є основним інформаційним навантаженням. Утримуючи в собі дані різних тематик, веб-ресурси являють собою хаотично розгалужену мережу з безлічі мільйонів електронних сторінок.

При цьому виникають великі труднощі в пошуку і орієнтуванні серед інформаційних потоків такого обсягу. Саме для навігації і успішного виявлення потрібної інформації, були створені пошукові системи. Неможливо уявити собі сучасний Інтернет без пошукових систем. Зберігаючи інформацію про мільярди веб-сторінок, ці системи є найбільш потужним інструментом для пошуку та розподілу інформації. Саме залучення уваги користувачів за допомогою демонстрації цільової веб-сторінки в пошуковій системі є метою пошукової оптимізації.

Актуальність предметної області. Особливої актуальності в даних умовах набуває вміння підвищити рейтинг сайту в очах пошукової системи, що автоматично забезпечить ресурсу для користувача увагу. Відбираючи велику кількість часу і витрат праці, заходи з пошукової оптимізації невпинно вимагають поліпшень в процесі їх виконання. У зв'язку з цим гостро постає питання про автоматизацію роботи фахівців, зайнятих оптимізацією сайтів з метою скорочення витрат часу і праці.

Мета: створення програмного продукту, що дозволяє виконати процес пошукової оптимізації сайту з допомогою інтелектуальних засобів.

Завдання:

- дослідити інформаційні джерела, що містять дані про пошукову оптимізацію, структуру та семантичне ядро сайтів;
- проаналізувати методи реалізації пошукової оптимізації з допомогою інтелектуального аналізу даних;
- вибрати методи, що відповідають особливостям даних про пошукові запити;
- дослідити існуючі методи кластеризації текстових даних;
- знизити кількість часу і праці на оптимізацію сайтів за допомогою розробки спеціалізованого додатку;
- реалізувати в даному додатку функціонал по роботі з ключовими словами, їх збору та кластеризації збору пошукової статистики запитів користувачів.

Об’єктом дослідження виступає пошукова оптимізація сайтів.

Предмет дослідження: сайт, що підлягає оптимізації.

Методи дослідження: методи системного аналізу, статистичні методи, методи інтелектуального аналізу даних.

Матеріали дослідження: дані відкритих джерел.

Наукова новизна одержаних результатів. Модифікований алгоритм збору та кластеризації семантичного ядра сайту, на основі методів інтелектуального аналізу даних за допомогою алгоритму пошуку популярних наборів в базі даних пошукових запитів.

Практичне значення отриманих результатів. Пропонований алгоритм може бути впроваджений в системи адміністрування сайтів або в засоби підтримки роботи seo-фахівців для підвищення повноти, точності і зниження часу розробки семантичного ядра сайтів.

Публікації та апробація результатів магістерської роботи. Отримані результати апробовані в межах в міжнародної наукової інтернет-конференції "Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення" та опубліковано одні тези доповіді по темі роботи [2].

Дипломна робота складається із трьох розділів, висновків, списку використаної літератури та додатків. У першому розділі проаналізовано методи пошукової оптимізації та розглянуто алгоритми пошукової системи Google.

В другому розділі обрані методи інтелектуального аналізу даних, що дозволяють вирішити задачу кластеризації семантичного ядра, та проаналізовані системи автоматичного просування сайтів.

У третьому розділі здійснено програмну реалізацію збору та кластеризації семантичного ядра сайту та проведено експериментальне дослідження розроблених алгоритмів.

1 АНАЛІЗ ПОШУКОВОЇ ОПТИМІЗАЦІЇ САЙТІВ ТА МЕТОДІВ ЇЇ РЕАЛІЗАЦІЇ

1.1 Визначення та предмет пошукової оптимізації

Щоб усвідомити безпосередньо розглянутий предмет і його призначення, необхідно вивчити його предметну область. Перш за все, розшифруємо саму абревіатуру «SEO»: Search Engine Optimization (англ. оптимізація під пошукові машини, пошукова оптимізація).

Під пошуковою оптимізацією розуміють комплекс заходів, спрямованих на підвищення позиції сайту в пошуковій видачі по певних пошукових запитах [3]. При цьому відповідність змісту сайту запитом називають релевантність. Пошукова оптимізація включає в себе як поліпшення внутрішнього змісту сайту (якості і кількості інформації, вдосконалення внутрішніх технічних аспектів реалізації сайту, поліпшення його візуальної привабливості для потенційних користувачів), так і роботу над зовнішніми факторами (прямі посилання на сайт, цитованість сайту на інших ресурсах та інше).

Процес оптимізації сайту для пошукових систем полягає в певній роботі над контентом сайту і його структурою, а також над внутрішніми і зовнішніми факторами, що дозволяють пошуковим машинам, грунтуючись на поточній версії їх алгоритму, привласнювати сайту деякий місце в пошуковій видачі серед інших відомих йому сторінок. При цьому фахівець, який займається оптимізацією сайту, повинен орієнтуватися не тільки в поточній версії пошукового алгоритму тієї чи іншої пошукової системи, але і знати деякі психологічні аспекти поведінки користувачів в мережі Інтернет.

Як правило, перед початком оптимізації ставиться завдання, на яке місце в пошуковій видачі очікується просунути сайт. З огляду на поведінкові фактори, такі, як втрата інтересу у користувачів в міру віддалення від вершини

пошукової видачі, метою оптимізації є безпосередньо перша сторінка пошуковика і три рядки другої сторінки.

В цілому, пошукова оптимізація ділиться на два види: зовнішню і внутрішню оптимізацію[3].

Внутрішня пошукова оптимізація - це набір дій і методів, спрямованих на поліпшення внутрішнього змісту і структури сайту. До внутрішньої оптимізації безпосередньо відноситься[5]:

- створення семантичного ядра сайту, тобто підбір тих ключових слів, за якими сайт буде відображатися в результатах пошуку;
- редагування текстів, виправлення орфографічних або логічних помилок;
- технічна робота над структурою сайту, як то: створення карти сайту, присвоювання сторінок легко запам'ятовуються і зрозумілих адрес, завдання інструкцій для пошукових роботів і інші методи, покликані спростити роботу з сайтом як користувачам, так і пошуковим машинам;
- робота над дизайном сайту і підвищення візуальної привабливості ресурсу;
- усунення технічних помилок, які уповільнюють або зовсім припиняють роботу сайту: починаючи від неіснуючих посилань і дублів сторінок, закінчуючи грубими семантичними і логічними помилками в коді сторінок, що діють на них скриптів, мережевих додатків.

Рисунок 1.1 – Схема робіт SEO-фахівця

До зовнішньої оптимізації відноситься такі методи, як нарощування маси на сайт і аналіз роботи, виробленої іншими оптимізаторами. Нарощування маси являє собою придбання (як комерційне, так і добровільне) посилань на просувний сайт. На зорі свого існування, пошукові системи орієнтувалися в основному на те, скільки посилань має сайт і згідно з цим привласнювали їм місце в пошуковій видачі. Зараз же пошуковики навчилися більш ретельно аналізувати зміст сайту, що, безсумнівно, сприятливо відбилося на якості Інтернет-ресурсів.

Не дивлячись на те, що внутрішня оптимізація сайту стала більш важлива, ніж зовнішня, останній вид оптимізації також важливий для підвищення позиції сайту. Зовнішня оптимізація впливає на такі показники, як PR.

PR (англ. Page Rank) - «ранг сторінки»[6], метод компанії «Google», що визначає кількість і популярність станиць, що посилаються на той чи інший ресурс. Кожній сторінці якого об'єкта в мережі присвоюється деяке початкове числове значення, яке змінюється в залежності від кількості посилань на нього. Суть методу в тому, що всі об'єкти, пов'язані гіперпосиланнями видаються у вигляді графа і чим більше посилаються на певну сторінку в мережі, тим «важливіше» вона виглядає для пошукової системи. Примітна особливість, що відрізняє цей метод від методу ТІЦ - це те, що ранг може присвоюватися як

сторінкам, так і для будь-якого об'єкта, тобто будь-якого графу[6].

Крім того, існує певний розподіл на типи пошукової оптимізації в залежності від застосовуваних методів, так звані: «біла», «сіра» і «чорна» оптимізація.

Під «білою» оптимізацією на увазі якийсь комплекс заходів, спрямованих на підвищення позиції сайту в пошуковій видачі, в результаті яких не порушуються будь-які правила, встановлені пошуковими сервісами для індексованих сайтів. Іншими словами, оптимізатор не повинен намагатися обдурити пошукову систему, змушуючи її, за допомогою деяких хитрощів, думати, ніби оптимізується сайт є краще (як в плані технічного виконання, так і по міститься інформації), ніж він є насправді.

«Чорна» оптимізація - навпаки, ставить собі за мету саме обман пошукової системи, що часто спричиняє за собою відповідні наслідки з боку пошукової машини. Сайт, викритий в шахрайстві, може бути знижений в загальній пошуковій видачі, а то і зовсім виключений з нього.

Що стосується «сірої» оптимізації, то вона не тягне серйозних санкцій з боку пошукових систем, тому що не порушує безпосередньо тих правил, які встановлюють пошукові системи для сайтів при індексації. Даний вид просування вимагає від фахівця високого рівня навичок і досвіду, а також знань внутрішнього устрою і принципів роботи пошукової машини.

Пошукова система в цілому складається з п'яти основних програмних компонентів[5]:

1) «павук» (англ. Spider) - модуль, що викачує сторінки з мережі Інтернет. Він сприймає інформацію сторінки в режимі вихідного коду і саме в такому вигляді зберігає її в базі даних;
2) «краулер» (англ. Crawler), що в перекладі з англійської означає «повзає». Цей модуль відповідальний за проглядання всіх посилань, що є на сторінці і занесення їх в базу даних. На основі цієї інформації він

формує шлях, по якому буде рухатися «Павук»;

3) індексатор (англ. Indexer) - даний модуль розділяє сторінку на складові елементи, такі як: заголовки, підзаголовки, основний текст, жирний і курсивний шрифт, а також інші інформативні елементи. Розділивши, таким чином, сторінку, він проводить її аналіз в залежності від поточного алгоритму пошукової системи;
4) база даних (англ. Database) - фактичне місце зберігання всієї накопиченої інформації про веб-сайтах, зібраної як «краулер», так і «павуком», а також результатів роботи індексатора та іншої інформації, необхідної для роботи системи;
5) система видачі результатів (англ. Search engine results engine) - це програмний модуль, який переглядає базу даних і вибирає найбільш релевантні запиту користувача сторінки.

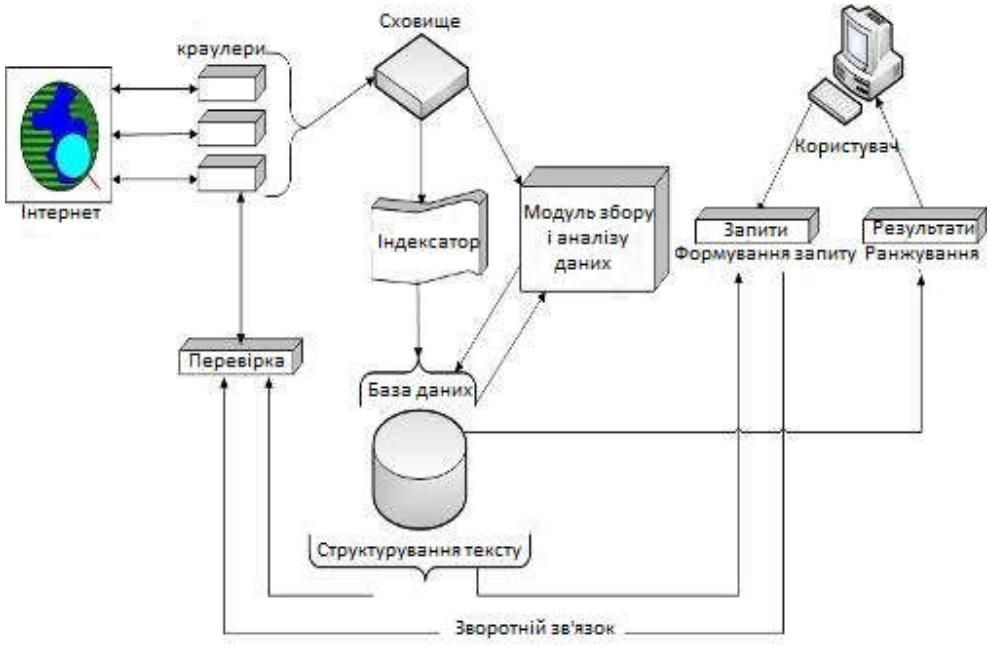

Рисунок 1.2 - Схема роботи пошукової машини

На зорі свого існування роботу пошукової машини виконували живі люди.

Вони переглядали всі знайдені ними сайти і сортували їх за каталогами. Пізніше, спираючись на цю базу каталогів, пошукові машини стали здійснювати самостійний пошук нових сторінок. Це відбувається наступним чином.

Спочатку «краулер» переглядає відомі йому ресурси в пошуку нових посилань. Він здійснює регулярні проходи всіх відомих йому посилань розшукуючи при цьому нові. Знаходячи їх, він виконує по ним перехід. Варто відзначити, що на кожне доменне ім'я «краулер» виділяється певний час для пошуку посилань. Після закінчення цього часу, «краулер» вирушає далі по мережі, повертаючись в наступному проході.

Далі починає свою роботу модуль, званий «павуком». Він користується знайденими «краулер» посиланнями, як картою, і викачує вміст сторінок в режимі вихідного коду і передає її для обробки індексатора. Цей модуль розділяє текст сторінки на складові елементи: заголовки, жирний і підкреслений текст, виділення абзаців та інше. Це робиться для зручності пошуку по проіндексованих документах. Оброблені сторінки надходять до бази даних пошукової системи. Слід зазначити, що на даний момент пошукові роботи проводять індексацію мультимедійних даних (таких, як аудіо- та відеофайли, флеш-анімація та інші вкрай неефективно.

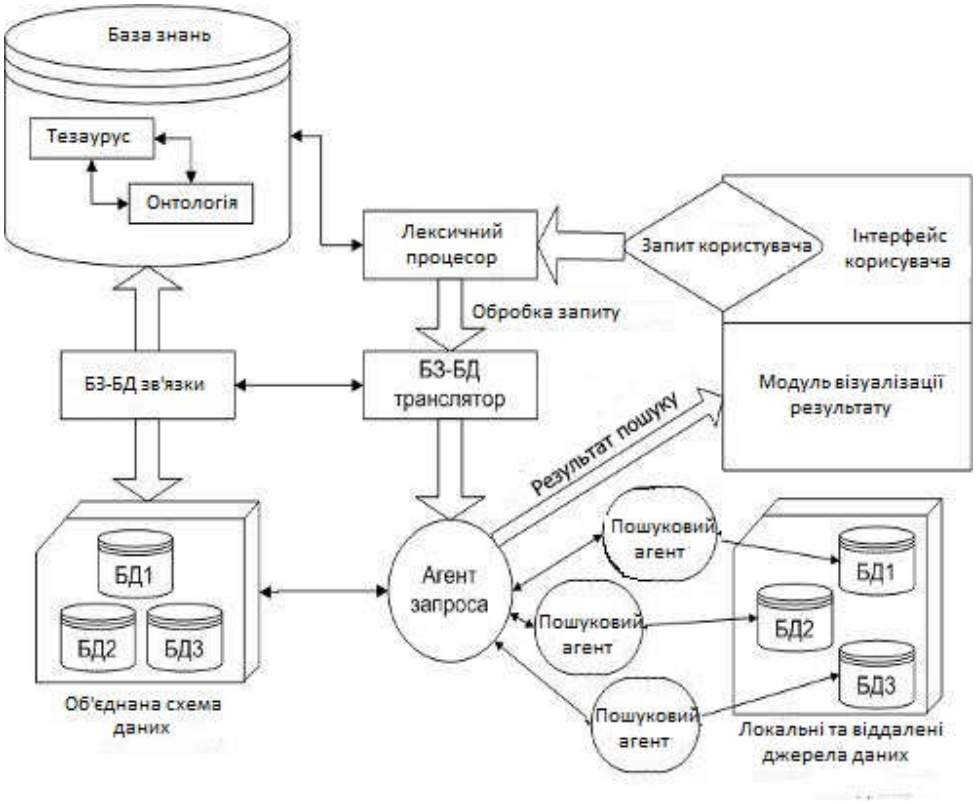

Рисунок 1.3 - Схема отримання даних з пошукової машини

Невірно вважати, ніби пошукова машина здійснює пошук заданої користувачем фрази по всій мережі Інтернет. Завантажені в режимі вихідного коду сторінки зберігаються в базі даних пошукової системи, а потім витягуються, коли користувач вводить свій запит в пошуковий рядок. Працюючи з таким величезним обсягом інформації, необхідно застосовувати особливі алгоритми, сортувальні результати пошуку [10]. Дані алгоритми, що постійно оновлюються і допрацьовувати, складають основу роботи пошукової системи. Саме вони визначають, які сторінки потраплять на перші сторінки пошукової видачі, які будуть відображатися після десятої сторінки, а які зовсім не відобразяться для користувача.

Не дивлячись на те, що пошукові системи намагаються якомога частіше оновлювати свої бази даних, існує безліч сайтів, які ніколи не потраплять в пошукову видачу, а саме:

- ресурси, доступ до яких захищений паролем;
- які пов'язані з іншими сайтами ресурси;
- сайти, які являють собою так званий «інформаційний шум»: кинуті, незавершені сайти.

Один з важливих аспектів, на які слід звернути увагу перед тим, як приступити до оптимізації, - це структура сайту. Пошукові машини обробляють не один мільйон веб-сторінок в день, так що переглядати велику кількість вкладених сторінок у них просто немає часу. Це означає, що сайти з дуже складною структурою будуть або індексуватися досить довго, або ігноруватися пошукачем взагалі. Для того, щоб цього не сталося, слід вдаватися до вкладеності не вище трьох. Наприклад: «http://mysite.com/directory1/directory2/directory3».

Інший аспект - це написання «грамотного» коду HTML-сторінок. Це також досить важливо, так як економить час читання сторінки для пошукача. При цьому слід враховувати кілька факторів[10]:

- всі парні теги повинні бути закриті (наприклад: <head> </ head>);
- не слід використовувати html-теги, які не підтримуються популярними браузерами;
- не слід використовувати застарілих тегів (наприклад: <frame> або <bgsound>);
- також не варто використовувати елементи коду, які пошукова машина ще не вміє обробляти (наприклад: деякі теги XML).

Пошукові машини важче індексують сторінки вище третього рівня вкладеності, тому важливо ще на стадії проектування веб-сайту залежить таку структуру, щоб будь-яка сторінка перебувала не глибше другого вкладеного каталогу.

Але при дотриманні простої структури каталогів, буває досить важко

вибудувати логічний структуру сайту. Особливо це може бути застосовано для великих, об'ємних сайтів зі складною логічною структурою. У випадках, коли структура сайту не може бути спрощена, можна вдатися до створення карти сайту.

При проектуванні сайту також слід приділити увагу точкам входу на сайт. Це ті посилальні шляхи, по яких користувач, рівно, як і пошукова машина, можуть дістатися до тієї чи іншої сторінки на сайті. При занадто громіздкій структурі сайт буде нижче оцінений як пошуковими системами, так і користувачами.

При структурі, запропонованої на рисунку 1.4, частина вмісту, як наприклад «Продукція» будуть важкодоступні через вкладеності, що дорівнює чотирьом[5]. ПМ не зможуть проіндексувати даний сайт, а якщо і проиндексирует, то даний процес розтягнеться на довгий час.

Рисунок 1.4 - Структура сайту з безліччю точок входу.

На рисунку 1.5 показана більш коректна структура сайту, що забезпечує рівень вкладеності, що дорівнює двом. Це забезпечує більш швидкий доступ до важливої інформації. Така вкладеність є правильнішою як для роботи пошукача, так і для зручності користувачів.

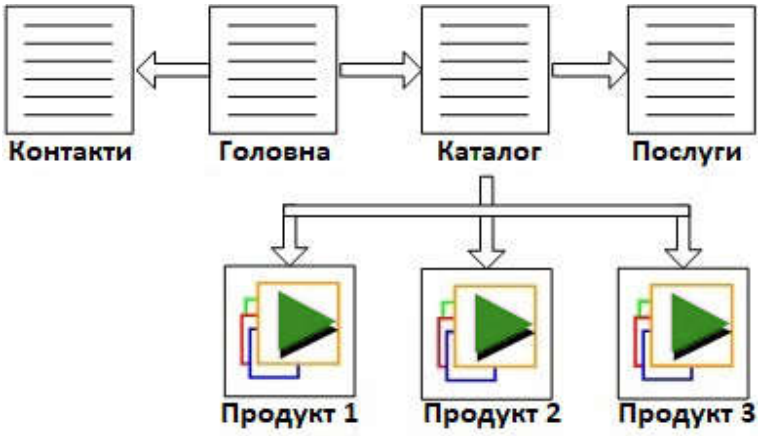

Рисунок 1.5 - Приклад структури сайту з множинними точками входу

При створенні структури сайту з множинними точками входу існує ряд рекомендацій:

1) кожна сторінка повинна бути самостійною по інформаційному змісту і структурному положенню на сайті;
2) сторінки сайту повинні бути оптимізовані під різні ключові слова для розширення потенційної аудиторії;
3) зовнішній вигляд кожної зі сторінок повинен відповідати єдиному стилю сайту;
4) сторінка повинна бути оптимізована для поліпшення індексації;
5) доступ до важливої інформації повинен здійснюватися за мінімальну кількість дій з боку користувача.

Для того щоб бути впевненим в якості свого коду, необхідно слідувати стандартам Консорціуму Всесвітньої павутини (Wide Web Consortium, W3C) [13]. Приведення сторінки до стандартів Консорціуму не вплине на позицію в пошуковій видачі, але полегшить роботу безпосередньо пошукової машини з сайтом.

Ключовим словом є слово або фраза, яку вводить в рядок запиту пошукової машини користувач. Від цієї фрази, крім інших факторів, залежить, який набір сторінок видасть пошуковик після ранжирування результатів. На даний момент провідні пошукові системи, такі, як Google, враховують у своїх алгоритмах ранжування ключові фрази не тільки у відповідних тегах веб-сторінки, але і безпосередньо в тексті [6].

При цьому, підбираючи ключові фрази, необхідно пам'ятати прості правила:

1) ключова фраза повинна бути точною. Користувач, проводячи пошук в мережі Інтернет, шукає цілком спеціалізовану інформацію;
2) ключова фраза повинна бути релевантною. Якщо на сторінці представлена інформація про кип'ятильники, а ключові слова стверджують, що сторінка про праски, то користувач буде введений в оману і, швидше за все, покине цю сторінку майже відразу. Також це тягне за собою санкції з боку пошукової системи;
3) ключова фраза повинна володіти певною щільністю. Щільність є емпіричною величиною і становить близько 3-5% від загальної кількості слів у тексті.

Щільність ключових слів являють собою величину, що визначає ставлення ключових слів до загальної кількості тексту.

Існує багато емпіричних формул для визначення щільності ключових слів на сторінці. Ось одна з них:

$$P = K \cdot \frac{100}{W}, \tag{1.1}$$

де P - шукана величина щільності, W - загальна кількість слів у тексті, K - кількість ключових слів[12].

Таким чином, маючи текст в 700 слів, ми потрапимо в рекомендований діапазон ключових слів при їх кількості від 21 до 35.

$$P = (K \cdot \frac{F}{W}) \cdot 100 \quad (1.2)$$

де P - шукана величина щільності, K - кількість ключових слів, F - кількість ключових фраз, W - загальна кількість слів у тексті[12].

При тексті в 700 слів рекомендований діапазон буде досягнутий при наявності в тексті 15 ключових фраз, що складаються з двох слів.

Дані розрахунки не вказують про стан ключових слів в тексті. Маючи все ті ж 17-35 ключових слів, але розмістивши їх, наприклад, в кінці тексту сторінки, а не на початку, можна отримати різні результати. Пошуковий робот, зайшовши на сторінку і не виявивши в її початку ключової інформації, не стане витрачати свої цінні продуктивні ресурси і залишить сторінку, позначивши як нерелевантні. Крім цього повинна зберігатися смислова і стилістична структура інформації веб-сторінки, інакше це може відштовхнути користувачів, які прийшли на сайт.

На ці параметри також впливає такий важливий в оптимізації поняття, як близькість ключових слів. Основний принцип його такий: ключові слова у фразі повинні бути розташовані якомога ближче. Варто пам'ятати, однак, що смислова цінність тексту не повинна губитися і в даному випадку[5].

При введенні пошукового запиту довжиною від двох і більше символів, пошукова машина звертає увагу не тільки на положення ключових слів на сторінці, а й їх близькість між собою, віддаючи перевагу найбільш близьким варіацій.

Не дивлячись на це, аналіз високорейтингових сторінок показує, що багато хто з них мають досить низьку щільність ключових слів [3]. Це дозволяє

говорити про те, що параметр щільності ключових фраз не є універсальним. Деякі з сторінок в першій десятці пошукових результатів можуть мати настільки низьку щільність ключових слів, що їх можна вважати близькою до нуля. Проте, їх висока позиція забезпечується, крім інших факторів, релевантними вхідними засланнями на сторінку.

Щільність і відстань між ключовими словами є одними із структурних характеристик веб-сторінки. Розташування інформації на сайті в загальному можна охарактеризувати схемою, наведеної на рисунку 1.6. У розділі 1 зазначаються заголовки статей, логотипи, основне меню, контактні телефони, форма авторизації та інші елементи. Колонки 3 та 5, як правило, використовуються як допоміжне меню і інформаційна (частіше рекламна) панель. Знаючи цю структуру, слід враховувати, що пошукові роботи читають сайт зліва-направо і зверху-вниз. Тим самим, розташування ключових слів, заголовків, контактних даних та іншої важливої інформації необхідно вибирати ближче до верхньої частини сторінки.

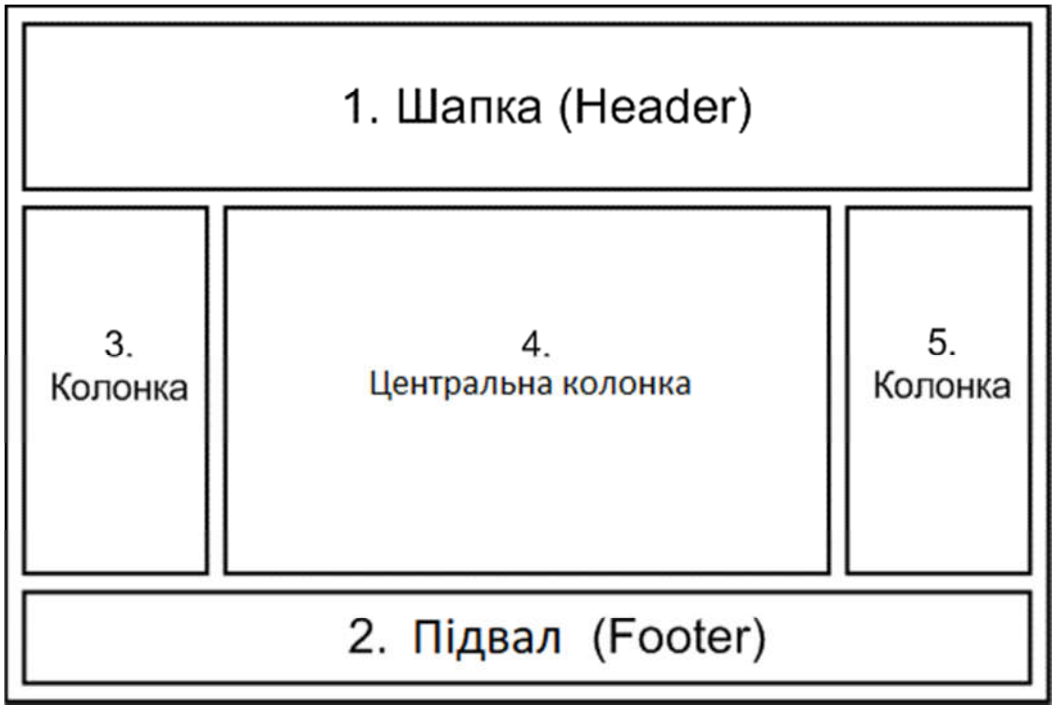

Рисунок 1.6 - Приклад шаблону сайту

Робота над ключовими словами є найбільш довгої і трудомісткої з усіх видів робіт SEO-фахівця. Складаючи список ключових слів, необхідно проаналізувати тематику, призначення сайту, врахувати його потенційну аудиторію і їх інтереси. Також фахівцями з просування враховується регіональні особливості цільової аудиторії.

Існують різні варіанти написання ключових фраз, що з'являються в результаті помилок або граматичних помилок. Виявлення таких і внесення їх до списку ключових слів сторінки також сприяє додатковому припливу відвідувачів, хоч цей приплив буде меншим, ніж від правильно підібраного фрази.

Ще одним важливим фактором при роботі з ключовими словами є коефіцієнт ефективності ключового слова KEI (англ. Keyword Effectiveness Index)[12]. Це числовий показник, вперше запропонований Самантою Рой, одним з провідних фахівців в галузі пошукової оптимізації. Цей коефіцієнт розраховується для кожної пошукової машини окремо і спирається на два параметри:

- Частоту пошуку (популярність) ключової фрази в обраному пошуковику;
- Кількість сайтів-конкурентів (конкурентність), що використовують ту ж ключову фразу.

З огляду на обидва ці параметра, обчислюється коефіцієнт ефективності ключової фрази.

$$KEI = \frac{P^2}{C}, \tag{1.3}$$

де KEI - коефіцієнт ефективності ключової фрази, P - параметр популярності фрази за останні два місяці, C - конкурентність фрази.

Шкала ефективності коефіцієнта KEI розподіляється наступним чином:

- коефіцієнт менше 10 - неефективна фраза;
- коефіцієнт в межах від 10 до 100 - фраза середньої ефективності;
- коефіцієнт в межах від 100 до 400 - фраза високої ефективності;
- коефіцієнт понад 400 - надефективна фраза.

З огляду на те, що пошукові машини для визначення тематики сайту спираються на звичайний текст, основна інформація на сторінках повинна бути представлена в текстовому вигляді. Графічна інформація, анімація, елементи програмного коду PHP і скриптів Java пошуковою машиною ігноруються і не вносяться в індекс, але вони є корисними з точки зору збільшення привабливості і функціоналу сайту для користувача.

Рекомендована довжина тексту на кожній сторінці - не менше 250 слів. Дана величина обумовлена тим, що пошуковий робот має обмежені апаратно-програмні ресурси, а користувачі можуть втратити інтерес до сайту при великій кількості однотипної інформації на ньому. Важливо, щоб контент сторінки періодично оновлювався. В цьому випадку пошукові роботи будуть більш охоче індексувати сторінку, так як в їх алгоритмах поведінки вітається сайти, які мають нову інформацію і періодично її оновлюють. Період оновлення може коливатися від хвилин і секунд (для сайтів біржових зведень) до днів і місяців (для періодичних Інтернет-видань і новинних стрічок).

Одна з основних труднощів при роботі над контентом сторінки, крім написання безпосередньо текстів, - це вбудовування ключових слів в необхідний текст. Головне правило тут - щоб текст після додавання ключових фраз зберігав стилістичну і змістову цілісність, при цьому не був перевантажений нав'язливими пропозиціями. В іншому випадку блок аналізу тексту пошукової машини вважатиме контент сторінки неякісним і знизить її

рейтинг, а користувачі, що не задовольнившись наданої інформацією, швидко покинуть сайт. При роботі над контентом сторінки рекомендується користуватися такими прийомами, як, наприклад «Заклик до дії» (англ. Call to Action). Даний прийом полягає в розміщенні на сторінці активних посилань на форми оформлення замовлень (для Інтернет-магазинів), різні форми зворотного зв'язку, в тому числі і сторінки замовлення зворотного дзвінка та інші дії [7].

Фахівці в області оптимізації рекомендують вибудовувати текстову інформацію таким чином, щоб змістити смисловий фокус веб-сторінки на дві або три основних ключових фрази. Після індексації сторінок пошуковими системами і подальшим їх ранжування, було емпірично встановлено, що присутній так званий ефект «накладення» ключових слів. Дане явище полягає в тому, що якщо сторінка сайту має високий рейтинг в будь-якому пошуковику за певною ключовою фразою, то перетинається з нею фраза, про яку йдеться на іншій сторінці сайту, також буде мати більш високий рейтинг.

Тег <title> являє собою назву всієї сторінки. Саме цей тег пошукова машина демонструє у вигляді посилання на сторінку в результатах пошукової видачі. Крім того, від привабливості інформації в даному тезі багато в чому залежить, чи перейде користувач на сайт. У зв'язку з цим можна виділити наступні аспекти, що впливають на сумісність з пошуковою машиною:

- тег заголовка для пошукової машини є повідомленням про зміст проглядається нею ресурсу. Невідповідність заголовки змісту сторінки тягне за собою санкції з боку пошукової системи;
- назва має містити саме ті ключові слова, які заявлені як пріоритетні іншим. Це підвищить якість сайту для користувачів і рейтинг сторінки для пошукача;
- положення ключових слів в заголовку є важливим: рекомендується розташовувати ключові слова якомога ближче до початку заголовка. Це обумовлено технічним особливостями роботи пошукових систем;

– пошукові машини знижують рейтинг сторінок-дублікатів. Внаслідок цього, не слід присвоювати різних сторінках однакові заголовки, так як це сприймається ПМ як дублювання.

Середня довжина заголовка (виключаючи службові символи, стоп-слова, пробіли) для ПМ не перевищує 60 символів, а у деяких - 70 символів. Варто враховувати, що через обмеженість ресурсної бази і при жорстких часових обмеженнях на обробку кожної сторінки, ПМ може зчитувати кількість символів заголовка менше заявленого. Всі інші символи понад ліміт пошукачем ігноруються і не розглядаються [9].

У тезі заголовка не рекомендується використовувати так звані стоп-слова. Дані слова настільки загально описують предметну область, що пошукові системи при індексуванні їх ігнорують. Також до стоп-словами відносять займенники, прийменники, службові символи начебто знаків оклику або питання, окремі символи, літери алфавіту або цифри (крім номерів телефонів). Незважаючи на те, що вони ігноруються пошуковою машиною, деякі з цих стоп-слів вносять ясність у зміст заголовка і покращують його сприйняття користувачем [7].

Слід виділити також теги заголовків тексту: <h1>, <h2>, ..., <h6>. Ці теги уможливлюють смислове розмітку тексту сторінки і покращують читаність тексту для користувача.

Нумерація тегів може бути співвіднесена з їх важливістю для пошукача. Важливість зменшується від 1 до 6. Вони також є хорошим місцем для розміщення ключових слів, але ці теги слід застосовувати з обережністю. При надмірному використанні теги можуть бути розцінені пошуковими роботами як недобросовісна оптимізація і накликати санкції [10].

Крім цього, текст в цих тегах, як і в тега посилань <a> можна виділити напівжирним шрифтом, що також приверне увагу пошукача. Для пошукової машини напівжирний текст в цілому означає важливу для розуміння

інформацію і є пріоритетною. Варто відзначити, що корисність даного виділення тим менше, чим частіше воно зустрічається в тексті.

Тег опису <description>. Даний тег є невидимим для користувача і містить в собі короткий опис сторінки. Зміст цього тега не впливає на позицію сайту в підсумковій видачі. Проте, цей тег представляє комерційну презентацію сайту в пошуковій видачі. Наявність грамотно підібраного опису, що містить релевантні ключові слова, може переконати користувача перейти безпосередньо на сайт [8].

Рекомендована довжина мета-опису обмежується в середньому 200-250 знаками, включаючи пробіли та службові символи. Всі символи понад вказану величини будуть проігноровані пошукачем. Не слід використовувати ключові фрази в описі сторінки. Існують рекомендації щодо обмеження кількості ключових слів в описі до 3-5 разів на сторінку [3].

Додавати мета-опису необхідно для кожної сторінки сайту, при цьому вони не повинні копіювати один одного. Тоді, маючи безліч пов'язаних описів, заголовків і основної інформації на сторінках, можна створити безліч точок входу на сайт, за якими користувачі зможуть знайти цікаву для них інформацію. Це збільшує приплив відвідувачів в порівнянні з ситуацією, коли вся оптимізація націлена на головну сторінку.

Тег ключових слів <keywords> являє собою службову інформацію сторінки, в якій вказується короткий набір ключових фраз, за якими слід характеризувати діяльність сайту. Довжина читається довжини тега для пошукових систем в середньому дорівнює 200 символів, включаючи пробіли та службові символи [5].

На початку 90-х, коли пошукові машини тільки починали своє становлення, вони виробляли ранжування сторінок, спираючись безпосередньо на ключові слова в тезі <keywords>. Це призводило до частих невідповідностей заявленої і фактичної інформації на сайті. Внаслідок цього, більшість

пошукових систем, в тому числі і найбільші, такі, як Google і Яндекс, стали менше враховувати даний тег при ранжируванні результатів пошуку.

Таким чином, використовувати мета-теги ключових слів, слід з великою обережністю, так як великого приросту в рейтингу серед пошукових машин вони принесуть, але можуть викликати санкції з боку пошукової системи. При роботі з даними тегом не слід допускати повторення ключових слів, уникати ключових слів, які не відповідають змісту сайту, а також необхідно дотримуватися авторські права.

Особливістю даного тега є те, що в ньому допускається введення ключових слів з орфографічними помилками. Вони можуть з'являтися, коли користувачі вводять текст в рядок запиту пошукової машини, тому пошукові системи лояльно відносяться до даного прийому, так як він спрямований на підвищення зручності для користувачів.

Як показує практика, в даний тег слід включати не тільки безпосередні ключові слова, за якими ведеться оптимізація, а й пов'язані зі змістом сторінки слова. До них можна віднести:

1) різні варіанти ключових слів (відмінювання іменників, відмінювання дієслів, множинні форми);
2) скорочені форми ключових слів;
3) жаргонізми та інші вузькоспеціальні терміни, властиві освітлюваної області;
4) орфографічні помилки, друкарські помилки і інші форми неправильного опису.

З огляду на економії індексованого простору тега, а також для того, щоб ключові слова індексувалися в різних комбінаціях, можливо опущення ком з даного тега. Крім цього, у пошуковиків існує певна особливість - вони можуть об'єднувати поруч стоять фрази в смислові блоки.

1.2 Огляд алгоритмів роботи популярних пошукових систем

Для того щоб ефективно просувати той чи інший сайт одним з найважливіших чинників є актуальна інформація про пошукові системи якою володіє SEO-оптимізатор, адже від їх вибору залежить рентабельність сайту. Тому, для початку, потрібно проаналізувати рейтинги пошукових систем. Так як, більшість пошукових систем на даний час використовують геолокацію (що важливо для комерційних проектів), варто розлянути лише статистику для деяких країн.

Світ. За даними сайту www.smartinsights.com, згідно чистій ринковій частці (станом на квітень 2017 року) частка глобального маркетингу з точки зору використання Google становить понад 77% (рис. 1.7) [32]. Це ще раз підтверджує той факт, що Google є лідером на ринку, але також підкреслюється, що інші, такі як Yahoo, Bing, Baidu і т.д. не варто ігнорувати. Цікаво відзначити, що значна частка ринку Google як і раніше росте.

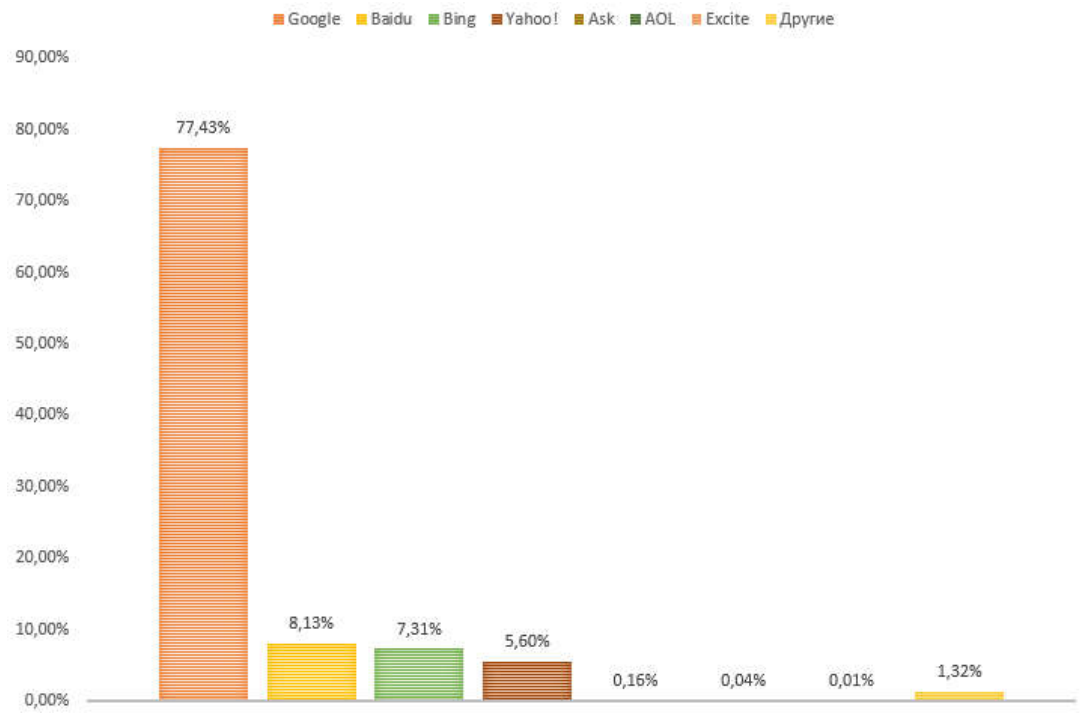

Рисунок 1.7– діаграма популярності пошукових систем для світу.

Популярні пошукові системи в Україні. В Україні незмінним лідером залишається Google. Додатковим чинником у збільшенні рейтингу цієї пошукової системи став указ «Про застосування персональних спеціальних економічних та інших обмежувальних заходів (санкцій)» відносно пошукових систем «Яндекс», «Mail.ru» та інших. Лідери на травень 2017 рік в Україні згідно з даними gs.statcounter.com (рис. 1.8) [32]:

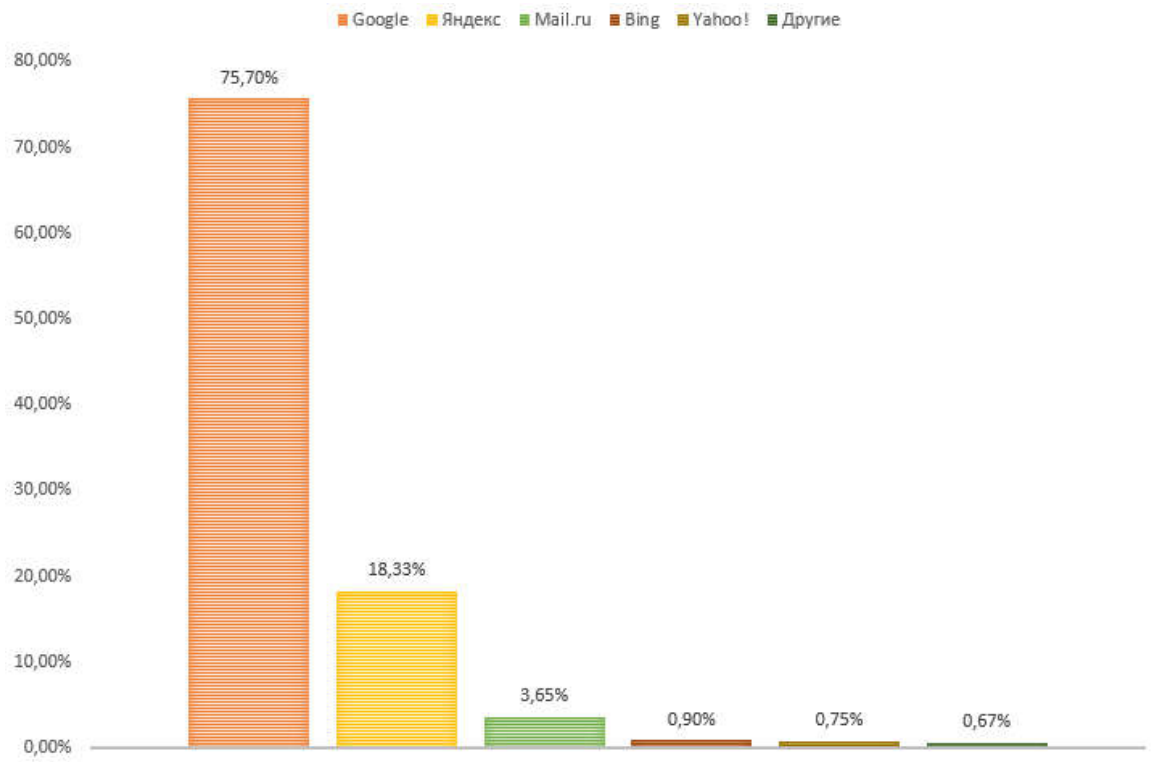

Рисунок 1.8 – статистика використання пошукових систем в Україні.

Підводячи підсумок відносно приведених вище даних, можна стверджувати що головними пошуковими системами в Україні, котрі мають зацікавити SEO-оптимізатора, є «Google». Тому варто розглянути алгоритми пошуку цієї системи.

Практично щомісяця Google вводить нові оновлення алгоритмів, при чому це ті що підтверджені офіційно. Далеко не всі з них мають істотний вплив на

результати пошукової видачі. Ми ж розглянемо алгоритми які внесли суттєві зміни в результати ранжування за останні роки.

Алгоритм Panda.

Запущений: 24 лютого 2011

Оновлення: приблизно щомісяця.

Мета: знижує позиції сайтів з низькоякісним контентом.

Алогоритм Google Panda використовується для виявлення сторінок з неунікальним контентом, з контентом, переповненим ключовими словами, спамом або автоматично згенерованим контентом. Алгоритм також може зачіпати сайти з дублюються на багатьох сторінках цього ж сайту інформацією і сайти з недостатньою кількістю контенту. Такі сторінки або сайти в цілому знижуються в рейтингу ранжування Google[35].

Google Penguin.

Запущено: 24 квітня 2012.

Оновлення: 25 травня 2012; 5 жовтня 2012; 22 травня 2013, 4 жовтня 2013; 17 жовтня 2014; 27 вересня 2016 року; 6 жовтня 2016 року; рараз в режимі реального часу.

Мета: знижувати в ранжуванні сайти з посилальними профілями що містять спам і сайти, котрі маніпулюють посилальної масою.

Посилання довго були вирішальним фактором в ранжуванні в Google і алгоритм Penguin був створений для виявлення і застосування санкцій до сайтів з неприродною посилальною масою. Цей алгоритм змінив розуміння просування сайтів під Google і став страшним сном для багатьох SEO-оптимізаторів. Penguin став початком кінця ери орендованих посилань як головної складової успішного просування. З осені 2016 року став частиною основного алгоритму та збільшив в рівній мірі як шанси потрапити під санкції, так і можливість їх зняття для раніше постраждалих сайтів.

Google's Pirate Update.

Запущено: серпень 2012.

Оновлення: жовтень 2014.

Мета: знижувати в ранжуванні сайти, які регулярно отримують скарги за зміст піратського (захищеного авторським правом) контенту.

Цей алгоритм був розроблений, щоб сайти з численними скаргами на піратський зміст не мали можливості високо підніматися в результатах пошуку Google. Більшість з порушених алгоритмом сайтів містили фільми, музику або книги, доступні для скачування або перегляду. До таких сайтів Google також відніс торрент-треккери і сайти-агрегатори посилань на файлообмінники, хоча формально вони не зберігають заборонені файли, але надають інформацію про те, як їх скачати з мережі. Точної кількості необхідних скарг для потрапляння під фільтр пошукач не озвучує, але однозначно ясно, що робота даного алгоритму далека від ідеалу - багато сайтів з сотнями DMCA скарг успішно ранжуються в трьох найвищих позиціях.

Hummingbird.

Запущено: 30 серпня 2013

Апдейти: немає.

Мета: надавати більш релевантні результати, ґрунтуючись на зміст пошукового запиту.

Алгоритм Колібрі вніс масштабні зміни в область інтерпретації пошуковиком запитів користувача. З виходом цього алгоритму основний акцент робився на надання результатів, заснованих на розумінні намірів користувача, а не на простому входження ключових слів, як це було раніше. Саме завдяки "Колібрі" Google покращив розуміння синонімів і поділ контенту за тематиками. Тільки після виходу цього алгоритму стало можливим побачити в результатах пошуку сторінку, на якій відсутній пошуковий запит користувача, але присутні його синоніми.

Pigeon.

Запущено: 24 июля 2014.

Оновлення: 22 грудень 2014.

Мета: надавати більш релевантні результати локального пошуку.

Даний алгоритм вплинув на результати видачі, в яких має значення місцезнаходження користувача. Незважаючи на очікування багатьох фахівців, в основному торкнувся тільки англомовного сегмента. Після виходу цього алгоритму місцезнаходження користувача і його відстань до об'єкта, пропонованого як результат видачі, стали враховуватися як фактори ранжування. Алгоритм "Pigeon" дозволив локальним нішевим бізнесам (ресторанам, кафе, освітнім установам) обійти в результатах видачі великі розкручені сайти і отримувати більше трафіку, тому варто подбати про реєстрацію в Google My Business і згадку свого сайту в каталогах і сайтах саме цього регіону.

Mobile Friendly Update.

Запущено: 21 квітня 2015.

Оновлення: немає.

Мета: підвищувати оптимізовані для мобільних пристроїв сторінки в результатах пошуку на мобільному пристрої.

Ця зміна дозволила дружнім до мобільних пристроїв сторінкам ранжуватися вище в результатах мобільного пошуку. На десктопну видачу даний апдейт не вплинув. Покликаний надавати користувачеві більш зручні для мобільного пристрою сторінки - без необхідності масштабування для читання тексту, без горизонтальної прокрутки, зі зручними для кліку елементами. Важливо відзначити, що дія алгоритму направлена саме на конкретну сторінку, а не на сайт в цілому. Так, одна сторінка може бути визнана оптимізованою для мобільних пристроїв і отримати зростання позицій, а інша сторінка навпаки бути заниженою у видачі [5].

RankBrain.

Запущено: 26 жовтня 2015

Оновлень немає.

Мета: надати користувачеві кращі результати, засновані на релевантності та машинному навчанні.

RankBrain - це система машинного навчання, що дозволяє Google краще розшифровувати і розуміти сенс запитів користувачів і надавати більш релевантні результати в залежності від контексту запиту. У широкому сенсі, алгоритм визначає тематику сторінки і те, наскільки релевантним є контент відповідно до запиту користувача. Система машинного навчання дозволяє з поведінки відвідувачів сайту визначити, наскільки наданий контент був корисний, «навчитись», і з часом надавати максимально корисні результати.

Алгоритм Possum.

Запущений 1 вересня 2016.

Оновлень немає.

Мета: надавати кращі, більш релевантні результати локальної видачі, грунтуючись на місцезнаходженні користувача.

Завдяки цьому алгоритму місцезнаходження користувача стало ще більш важливим фактором для відображення того чи іншого результату з Local Business - чим ближче користувач до адреси компанії, тим з більшою ймовірністю даний результат буде присутній у видачі. Алгоритм відфільтрував афілійовані організації, наприклад, з однаковими номерами телефонів або адресами. Це дозволило надавати більш різноманітні результати, щоб уникнути спроб узурпації локальної видачі. Ще однією особливістю алгоритму стала можливість кращого ранжування для компаній з фізичною адресою за межами певного міста.

Алгоритм Fred.

Запущений: 8 березня 2017

Оновлень немає.

Мета: фільтрувати низькоякісні сторінки з результатів пошуку, що мають за мету отримати прибуток від розміщення реклами та посилань на інші сайти.

За словами представників Google, Fred карає сайти, що порушують рекомендації для веб-майстрів. Таке твердження не дало корисної інформації спільноті seo-оптимізаторів, а ось практичні дослідження показали, що від Фреда страждають сайти з малоцінних контентом, що розміщують переоптимізовані ключовими словами тексти, велика кількість реклами або вихідних посилань[32].

1.3 Аналіз завдання дипломної роботи та постановка задач дослідження

Завдання дипломної роботи полягає в пошуковій оптимізації сайту з допомогою інтелектуальних засобів, тобто, ціль цього проекту - створення програмного засобу з допомогою засобів інтелектуального аналізу даних, метою якого є просування сайтів на вищі позиції у пошукових запитах популярних пошукових систем, таких як «Google».

Для розуміння поняття «SEO» варто виділити основні та актуальні методи оптимізації, а саме:

- адаптація сайтів під мобільні платформи: більшість клієнтів сьогодні користується мобільними платформами для web-серфінгу;
- використання і налаштування адаптивного дизайну на сайті: на даний час пошукові системи досить добре вміють розпінавати дизайн сайтів;
- створення унікального контенту – пошукові системи сильно наголошують на цьому;
- оптимізація семантичного ядра під голосові пошукові запити;

– адаптація ключових слів та правильне використання мета-тегів (не варто зациклюватись на цьому, так як через переоптимізацію можна отримати протилежний ефект);
– оптимізація швидкості завантаження сайту (згідно статистики 40% користувачів покидає сайт, якщо його завантаження займає більше 3 сек.).

2 АНАЛІЗ МЕТОДІВ ЗБОРУ ТА КЛАСТЕРИЗАЦІЇ СЕМАНТИЧНОГО ЯДРА САЙТУ

Семантичне ядро (СЯ) - це максимально повний список ключових слів, що описують тематику і спрямованість сайту і відповідних йому за змістом. Це визначальний фактор всього просування сайту. СЯ - це список запитів, за якими буде йти робота з підвищенням видимості ресурсу в пошукових системах[5]. Виходячи з СЯ будуть вибиратися посадкові сторінки, на які користувач зможе потрапити з пошуковика. Також від нього залежить, яким чином буде виконуватися оптимізація сайту і в якому напрямку буде доопрацьовуватися його функціональність і зовнішній вигляд.

При створенні СЯ враховуються типи запитів за спрямованістю: виділяють навігаційні, інформаційні та комерційні типи запитів.

Перший тип запитів – це, в загальному випадку, бажання користувача знайти конкретний сайт на який він хоче потрапити. Під такий тип запитів зазвичай просування не проводиться.

Інформаційний тип запитів спрямований на надання користувачеві корисної інформації, наприклад запит "модні сумки 2016" не вказує бажання користувача купити сумку, а говорить про потребу отримання актуальної інформації з даного питання. Такий тип запитів часто використовується для просування сайтів туристичної спрямованості.

Третій тип спрямований на покупку продукту або послуги, користувач користується таким типом запиту, коли хоче щось купити. Даний тип відрізняє наявність в запиті слів "купити", "ціна", "вартість", "замовити", "недорого" і

подібні. Серед комерційних запитів також виділяють конверсійні – запити котрі ведуть за собою покупку. Наприклад, запити типу «купити ноутбук недорого» зазвичай відображає бажання користувача дізнатись, які продукти представлені на ринку та їхні ціни і наврядчи буде використаний для здійснення покупки. В противагу візьмемо запит типу «купити ноутбук Samsung R60+». Такий запит можна назвати конверсійним, тому що користувач конкретизував своє бажання і лише шукає сайт де воно може бути реалізованим.

Також запити прийнято ділити по частотності в залежності від того, скільки користувачів на місяць набирають їх в пошуковику, на високочастотні (ВЧ) - запити загального характеру, які не дозволяють виявити потребу користувача ("автомобіль"); середньочастотні (СЧ) - уточнені ВЧ запити ("купити автомобіль"); і низькочастотні (НЧ) - максимально точні запити ("купити автомобіль jaguar xf зелений"). Певних рамок і кордонів, що відокремлюють ВЧ від СЧ, а СЧ від НЧ запитів, не існує. Вони сильно залежать від тематики, але ми будемо вважати низькочастотними ті запити, які набирають до 500-700 разів на місяць; середньочастотними - до 1-2 тисяч разів на місяць; високочастотними - понад 2 тисяч раз на місяць [3].

Оптимально комбінувати кілька запитів різних видів - це забезпечить, з одного боку, стабільний приплив потенційних покупців, максимально зацікавлених в придбанні товару, а з іншого - забезпечить гарну рекламу компанії, яка займає високі позиції у видачі. Попит по конкретним фразам можна подивитися в одному зі спеціалізованих онлайн-сервісів - наприклад, в "Яндекс. Вордстат".

2.1 Аналіз алгоритму складання семантичного ядра сайту

Роботи зі складання семантичного ядра діляться на кілька етапів: підготовка, виявлення основних сегментів сайту, розширення списку з використанням сервісів і програм, чистка, кластеризація і підбір посадочних сторінок. Підсумком проведених робіт є впровадження запитів в структуру сайту [10].

Існує два способи складання СЯ - автоматичний і ручний. Сформувати СЯ сайту допоможуть автоматичні онлайн-сервіси. Розглянемо найбільш популярні з них.

Google.Adwords (рис.2.1). Розробка найпопулярнішої пошукової системи. Сервіс допомагає в складанні списку ключових слів, дозволяючи отримати статистику запитів в Google. Все, що потрібно зробити, - ввести загальний запит. Крім кількості запитів в Google.Adwords можна отримати інформацію про конкурентності ключових слів.

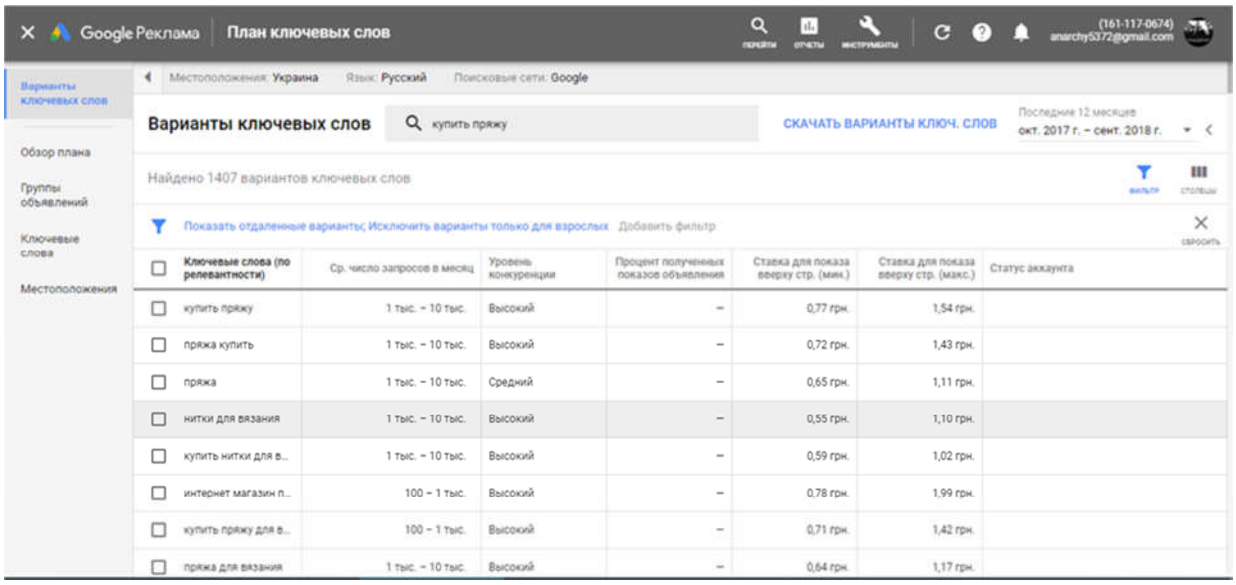

Рисунок 2.1 - Google.Adwords

Яндекс.Вордстат (рис. 2.2). Сервіс допомагає в складанні списку ключових слів, дозволяючи отримати статистику запитів в Яндексі. Аналогічним чином вводимо первинний запит і дивимося результат. На виході отримуємо списки основних і допоміжних запитів з прогнозом кількості показів за місяць.

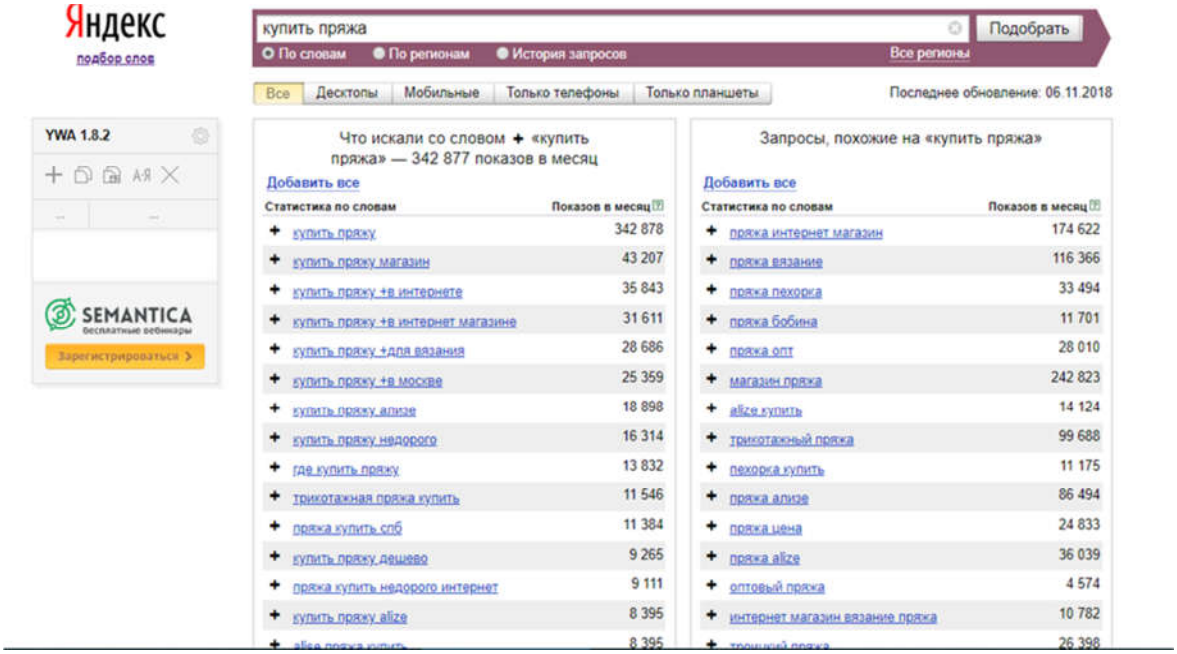

Рисунок 2.2 - Яндекс.Вордстат

Дані сервіси допоможуть швидко і без особливих зусиль скласти семантичне ядро сайту, однак, для опрацювання варіантів, які система не змогла передбачити рекомендується використовувати ручний спосіб складання СЯ.

Розглянемо докладніше ручний спосіб складання СЯ, який дозволить зробити більш якісне просування сайту. Розділимо роботу зі створення СЯ на кілька кроків.

Вивчення поточної (або створення нової) структури сайту. Для початку необхідно розділити сайт на сегменти і семантично описати їх.

Створення початкових запитів по тематиці сайту. На цьому етапі можна також скористатися одним з перерахованих вище автоматичних сервісів - генераторів. Крім того, можна скористатися таблицями Excel для пошуку всіх можливих комбінацій ключових слів. На виході повинен вийти великий список з комбінованих запитів, приблизно виглядає так: "зимові жіночі чоботи", "шкіряне взуття", " жіноче взуття", "спортивне взуття Nike" і т.д.

Аналіз конкурентів. Даний крок передбачає роботу з програмами, які допомагають аналізувати схожі запити у конкурентів. Ці програми легко знайти в Інтернеті. Запити, знайдені на даному етапі додаємо до загального списку запитів.

Чистка семантичного ядра. У списку, який ми склали на перших трьох етапах виявиться багато зайвих запитів, які не підходять нашому сайту. Наприклад, якщо ми не продаємо жіноче взуття фірми Nike, то запит "жіноче взуття Nike" буде зайвим. Для чищення списку можна використовувати готові програмні продукти, наприклад, "Аналіз груп" в програмі Key Collector. Після роботи з програмою можна перевірити список вручну. Також існує поняття "стоп-слова". Це слова, які апріорі не відносяться до нашого сайту. Наприклад, якщо ми продаємо взуття тільки в Тернопільській області, то запити із зазначенням інших регіонів потрібно додати в "блок" або зробити їх "стоп-словами". У програмі Key Collector для цього використовується кнопка "Стоп-слова", а в Excel дану роботу можна провести за допомогою "фільтра".

Додаткові дані. Щоб підготувати СЯ, необхідно зібрати наступні дані: частота запиту і геозалежні. Частоту запиту можна зібрати за допомогою сервісів Google.Adwords, Yandex.Wordstat або Key Collector. Нам знадобиться звичайна частота - скільки разів були введені запити в пошукову систему з включенням потрібного запиту в будь-якій формі і точна - скільки разів був

введений саме конкретний запит в потрібній формі. Геозалежні - запит, результати пошукової видачі по якому в різних регіонах відрізняються.

Визначення ефективності. Порахуємо коефіцієнти ефективності запитів в залученні відвідувачів на сайт. Для визначення ефективності запитів будемо використовувати формулу:

$$\text{Коеффіцієнт ефективності } (\%) = WS\text{ "!" } \cdot \frac{100}{W} \qquad (2.1.1)$$

де WS - частота запиту по wordstat.yandex.ru, WS "!" - частота запиту по wordstat.yandex.ru в лапках і знаком оклику перед кожним словом. Тим самим визначаємо найефективніші запити і концентруємося саме на них. Як правило, найбільший коефіцієнт виявляється у низькочастотних запитів.

Таким чином, найефективнішим і найбільш якісним способом складання семантичного ядра залишається ручне створення списку запитів з використанням допоміжних програм і оцінкою коефіцієнта ефективності запитів. Для створення семантичного ядра необхідно використовувати всі можливі джерела і програми-автоматизатори. Дана робота забирає у сео-оптимізатора багато часу і сил, тому було вирішено створити інструмент, який дозволить SEO-оптимізаторам складати та кластеризувати СЯ, при цьому зменшивши грошові та часові витрати на просування сайту.

2.2 Аналіз методів кластеризації текстових даних

Однією з найпоширеніших форм представлення знань є тексти на природній мові. Текстова інформація природна для людини, вона легко створюється, сприймається, поширюється і змінюється. Кількість інформаційних ресурсів в сучасному світі неухильно зростає через нових можливостей позиціонування текстів, пов'язаних з масштабним розвитком електронно-обчислювальної техніки, а також з підвищенням доступності методів запису і зберігання інформації. Необхідність вивчення і осмислення постійно зростаючого обсягу неструктурованою текстової інформації робить задачу аналізу цих даних актуальною на сьогоднішній день [16].

Завдання кластеризації даних - завдання по об'єднанню в групи об'єктів, схожих за певними ознаками - одне з фундаментальних питань Data Mining. Найчастіше, кластеризація даних має прикладний характер і список областей, в яких застосовується кластеризація досить широкий: аналіз текстів, сегментація зображень, прогнозування різних подій, маркетинг. В сучасних задачах кластеризації є першим етапом обробки даних для формування груп ознак, для яких, в подальшому будуть застосовані інші методи і моделі.

Варто відзначити, що завдання кластеризації документів має багато спільного із завданням класифікації текстів в заздалегідь створену і попередньо заповнену систему категорій. Незважаючи на попередню схожість, кластеризація має ряд своїх особливостей, які необхідно враховувати при вирішенні завдань. Всупереч добре вивченим і ефективним на практиці методам класифікації, підходи до вирішення завдання кластеризації в деякій мірі бідні і мають вельми обмежену практичну застосовність. Основна причина такої відмінності це те, що завдання кластеризації дуже погано піддається

формалізації. У той час як існують об'єктивні і досить точні методи оцінки якості класифікації, оцінка якості кластеризації, як правило, ґрунтується на думці експерта і її важко виразити в численних показниках. Іншими словами, однією з фундаментальних проблем кластеризації текстових даних є оцінка якості отриманих результатів, так як не існує єдиного, загальновизнаного у всіх випадках методу оцінки [19].

2.2.1 Існуючі методи кластеризації текстів

У загальному випадку задача кластеризації ґрунтується на метриках близькості. Вихідні документи подаються у вигляді вектора в просторі певних ознак. Підходи, для формування вектора ознак можуть істотно відрізняться один від одного і метрики можуть вираховуватися різними способами і є окремим завданням. У найпростішому випадку кожен ознака відповідає наявності слова або словосполучення в початковому тексті. Величина компоненти може визначаться також різними способами, наприклад, компонента може бути істиною (або одиницею) якщо розглядається слово/словосполучення присутній в даному тексті або нулем в протилежному випадку; величина може розраховуватися за кількістю входжень даного слова в документ (частота народження) або розраховуватися будь-якими іншими більш складними формулами, наприклад, враховувати середню зустрічальність конкретного слова за поточним набору тексту щодо всього корпусу документів[19]. Міра близькості між текстами в даному випадку буде розраховуватися як скалярний твір між векторами. Більшість алгоритмів кластеризації в якості вихідних даних використовується прямокутну матрицю S,

яка складена з векторів документів і квадратної матриці близькості (формула 2.1).

$$V = S \cdot S(t) \tag{2.1}$$

Основним недоліком методів, які використовують таку матрицю, є занадто велика розмірність простору ознак, деякі з яких є надлишковими і можуть позначитися на точності результату (маскувати схожість між документами при його відсутності).

Один з основних методів кластеризації є ієрархічна кластеризація. Такі алгоритми кластеризації будують на виході Дендрограмма (бінарне дерево), яке пов'язує всі тексти. Існують різні варіації таких алгоритмів, які можуть будувати дерево як зверху вниз (розглядають вихідний набір текстів як один єдиний кластер), так і від низу до верху (розглядають кожен текст з набору як один кластер). Таким чином, ми маємо всі перетину бінарного дерева, що показує підсумкові кластери. Яскравим прикладом таких алгоритмів можуть бути Single Link, Complete Link, Group Average [18]. Дані назви описують, яким чином, будуть визначатися відстані між кластерами:

- single Link - мінімальна відстань між парою об'єктів в сусідніх кластерах;
- complete Link - максимальна відстань між парою;
- group Average - середня відстань.

Основним недоліком даного методу є те, що в загальному випадку ми отримуємо повну Дендрограмма, тобто повне бінарне дерево по всіх текстах. І

кількість кластерів в даному випадку, так чи інакше має задаватися явно, щоб наочно побачити зріз дендрограмми.

Другою групою методів є неієрархічна кластеризація. Найпоширенішими алгоритмами в цій групі є KMeans і EM (Expectation Maximization) [24]. У найпростішому варіанті алгоритму KMeans потрібно завдання початкових положень Центроїд і числа кластерів, після чого запускається ітеративний процес, який стабілізує центроїди. На кожному кроці документи приписуються до кластеру з найближчим центроїдом. Після того як всі тексти розподіляться, буде вирахувано нове положення центроїдів. Якщо центроїди перестали переміщатися або досягнута умова закінчення процесу, то вважається що кластеризація виконана. Даний алгоритм практично завжди використовується з допоміжними алгоритмами, які можуть знайти оптимальне становище початкових центроїдів і кількість кластерів. Для обчислення положення початкових центроїдів використовується алгоритм Single / Average Link. В даному випадку, розмір випадкової вибірки розраховується за формулою 2.2:

$$V = \sqrt{k \cdot n} \tag{2.2}$$

де k - число кластерів, n - кількість вихідних текстів.

Для обчислення оптимального числа кластерів можна використовувати алгоритм Minimun Description Length (MDL). KMeans один з найпростіших і широковживаних алгоритмів кластеризації, також є окремим випадком загального методу EM. Метод Expectation Maximization працює з ймовірнісної моделлю визначення вихідного документа до якогось певного кластеру.

Вектори документів, в даному випадку, розглядаються як випадкова величина. Якщо заздалегідь відомі параметри розподілу, тоді можна обчислити умовну ймовірність приналежності вектора до кластеру [22].

Основним недоліком даного сімейства алгоритмів є те, що для текстових даних не завжди можна визначити параметри розподілу і метод не гарантує досягнення глобального мінімуму.

Ще один метод, який використовується для кластеризації текстів є метод аналізу основних компонент (PCA) [23]. Він використовує діагоналізації повної ковариаційної матриці термів. Даний метод має дуже невисокої швидкістю роботи, тому застосування його на скільки великих наборах текстів навряд чи можливо.

Існують різні спроби кластеризації текстів з урахуванням їх семантичної інформації. Дане завдання виникає тоді, коли ми маємо текстову інформацію, сформульовану різними словами і їх порядком, але несучу в собі схожий сенс, або схожими словами, але несе в собі протилежний зміст. В такому випадку, алгоритми засновані на чистому лексичному схожості можуть не дати вірний результат. Такі завдання вирішуються за допомогу наступних способів [27]:

- складання тезаурусів;
- алгоритмічний спосіб, який встановлює і враховує асоціативно-семантичні зв'язки між словами (LSA, pLSA, LDA).

Розглянемо обидва способи докладніше. Перший варіант - формування тезаурусів досить трудомісткий і дуже сильно залежить від специфіки самого завдання. Створити універсальний тезаурус, який можна буде використовувати для кластеризації будь-яких текстів не представляється можливим на даний момент. Широковживаних тезаурусом є база понять Вікіпедія. Але в такому разі, по-перше, ми покладаємося повністю на зовнішнє джерело даних, по-

друге, довіряємо точність накопичених знань, людям, які створюють даний портал. Також критичним недоліком є те, що близькість понять в даному випадку буде визначатися за близькості термінів відносно один одного в контексті тезауруса, а не в контексті початкового тексту. Другий варіант - алгоритмічний. Найпоширенішим алгоритмом вирішення даного завдання є алгоритм латентно-семантичного аналізу (LSA). Даний метод застосовується для рекомендаційних систем, інформаційного пошуку, кластеризації і ще ряду завдань. Алгоритм знаходить приховані смислові взаємозв'язки між об'єктами (об'єкти можуть бути абсолютно будь-якими). Погляньмо на цей алгоритм на прикладі - у нас є набір документів і ми хочемо знаходити попарно близькі документи за змістом [28]. Висновок про близькість ми можемо робити на основі того, які слова і як часто зустрічаються в цих документах. Аналогічно класичним методам необхідно виконати предобработку даних, виділити необхідні слова і сформувати таблицю ознак. Для підготовки даних, в найпростішому випадку можна використовувати наступний підхід - будемо враховувати тільки частоту зустрічальності слів. Припустимо, що кожна тема характеризується певним набором слів і частотою. Якщо в тексті конкретний набір слів вживається з певними частотами, то текст належить до певної теми. Порядок слів і морфологічні форми для нас будуть не важливі. Таким чином, будується таблиця слово-документ. В осередках буде зберігається або 0 або 1 в залежності від відсутності / наявності слова в тексті. У рядках будуть слова, а стовпці будуть відповідати документам із загального набору. Можна використовувати різні метрики, не обов'язково бінарну. Також поширено враховувати частоту слова в документі щодо загального набору даних (tf-idf). Далі для порівняння текстів вводиться міра схожості двох стовпців таблиці (Манхеттенська відстань, евклідова, косинусна і тд). Далі отримана матриця розкладається методом SVD (2.3):

$$A = U \cdot V \cdot WT \tag{2.3}$$

Далі відбувається виділення рядків матриці U і стовпців W, які відповідають найбільшим сингулярним числам (їх може бути від 2-х до мінімуму з числа термінів і документів). Конкретна кількість врахованих власних чисел визначається передбачуваним кількістю семантичних тим в завданні. А взагалі чим більше сингулярне число, тим сильніше в колекції проявлена тема [36].

Алгоритм LSA має ряд обмежень - семантичне значення документа визначається набором слів, які як правило йдуть разом, повністю ігнорується порядок слів, кожне слово має єдине значення. Основними ж недоліками алгоритму LSA є те, що ми припускаємо про те, що карта слів не має вид нормального розподіл (або має будь-який розподіл різних варіаціях і поліпшення алгоритму) і те, що ці алгоритми мають досить велику кількість параметрів, визначення яких, може істотно вплинути на якість одержуваного результату і визначається емпірично.

2.2.2 Вибір методу кластеризації для тектових даних

Завдання кластеризації з використанням семантичної інформації можна розділити на наступні етапи:

1) отримання семантично значимої інформації, рішення задачі вилучення даних;

2) представлення отриманої інформації у формалізованому вигляді;
3) обчислення міри відстані між ознаками;
4) безпосереднє виконання кластеризації.

Далі кожен з етапів буде розглянуто окремо і більш детально.

Витяг даних. Вихідні тексти на природній мові мають неформальну структуру, тобто ми будуємо пропозиції грунтуючись на правилах природної мови, використовуючи все різноманіття словника мови. Певні фіксовані конструкції, як в мовах програмування відсутні, тому завдання вилучення даних має велику актуальність і часто є першою етапом обробки текстів. Так як в рамках даної роботи завдання вилучення даних є допоміжною розглянемо її в обсязі, достатньому для подальшого дослідження. Витяг інформації - це варіант інформаційного пошуку, пов'язаний з виявленням сутностей і взаємозв'язків між ними, при якому з неструктурованого тексту виділяється структурована інформація, готова для подальшої обробки [29].

Першим етапом є графематичний аналіз. Він необхідний для поділу неструктурованого тексту на пропозиції і слова. графематичний аналіз включає в себе поділ вихідного тексту на слова і роздільники, виділення стійких словосполучень, виділення власних назв, виділення структурних елементів, виділення пропозицій з початкового тексту.

Морфологічний аналіз виконує нормалізацію слів, тобто приведення слів до їх початкової незмінною формі, і виділяє набір параметрів приписаних до даної словоформи.

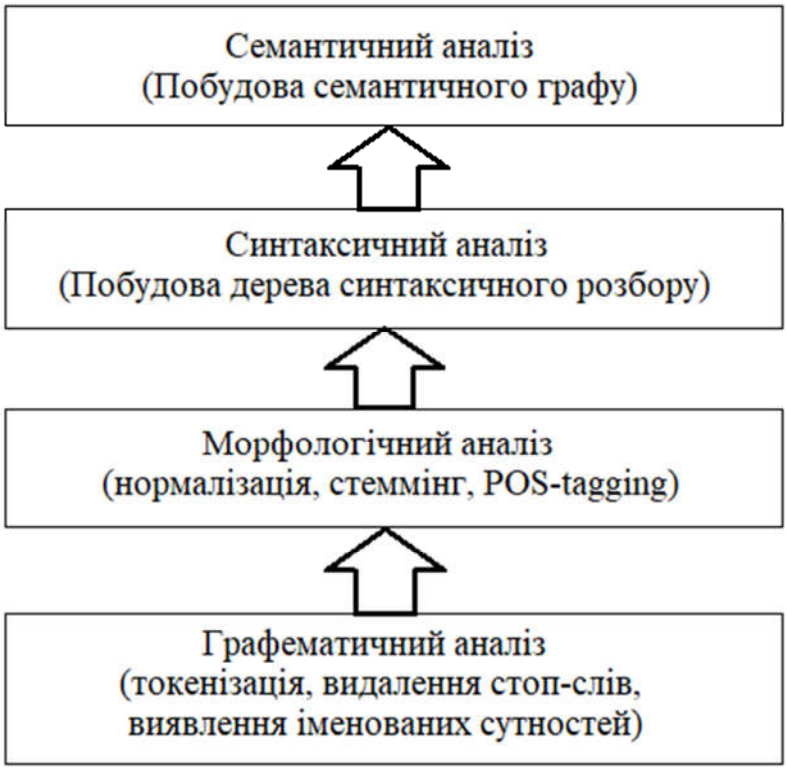

Рисунок 2.3 - Етапи аналізу текстової інформації природною мовою

Синтаксичний аналіз визначає ролі слів і їх взаємозв'язку, в результаті чого ми отримуємо дерево синтаксичного розбору.

Семантичний аналіз шукає смислові зв'язки між виділеними поняттями, і грунтується на результатах роботи попередніх аналізаторів. На даному етапі з'являється формальне подання сенсу тексту.

Внаслідок роботи аналізаторів необхідно отримати з повністю неструктурованою інформації на природній мові певну, можливу для подальшої обробки електронно-обчислювальними засобами, структуровану інформацію. Оптимальним варіантом виглядає структура, схожа на граф, в якому, буду відображатися суті з їх властивостями і взаємозв'язку між цими сутностями. Формалізована структура представлена на рисунку 2.4.

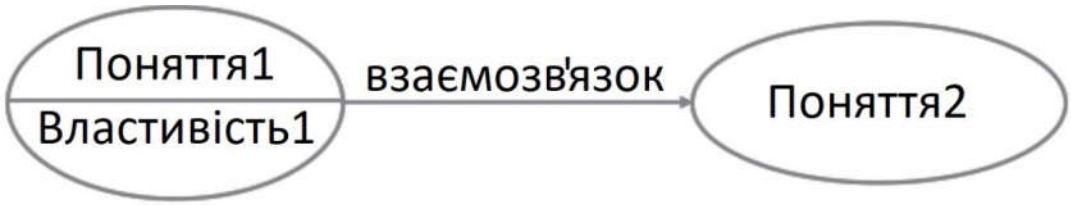

Рисунок 2.4 - Структура семантичного графа

Для вирішення завдання вилучення фактів виділяють три основні підходи [29]:

- з використанням онтологій;
- машинне навчання (ML);
- з використанням формалізованих правил.

Онтологія включає в себе поняття і відносини між цими поняттями, описує дані, які треба зробити з тексту. Онтологія це докладний опис галузі знань. У контексті вирішення завдання отримання даних в онтологіях описуються різні поняття, відносини між цими поняттями і їх характеристики. Такий метод дозволяє нам будувати гіпотези стосовно об'єктів в тексті і підтверджувати їх або відхиляти.

Машинне навчання використовує великі обсяги вхідних даних і грунтується на статистиці. На підставі ймовірності появи тієї чи іншої лексичної одиниці в певному контексті, система визначає її в певний результуючий набір. Основним недоліком даного підхід є складність навчання системи. Система дуже сильно залежна від того, на яких наборах даних вона навчена. Якщо вхідні тексти будуть сильно відрізняться від навченою вибірки результати роботи

алгоритмів будуть некоректними. Для навчання алгоритмів використовуються структуровані тексти розмічені вручну і цей процес складно автоматизувати.

Підхід, заснований на правилах, є написання шаблонів (регулярних виразів певного типу) вручну деякою неформальною мовою. Експерт складає дані вирази, що описують певні факти в тексті, які необхідно витягти. І далі система працює за цими шаблонами. Типовими прикладами можна виділити виділення власних назв або дат.

Підводячи проміжні підсумки, ми можемо зробити висновок про те, що розробити алгоритм обробки природної мови можна використовуючи як машинне навчання, так і набір правил. Основною відмінністю є те, що для написання правил (шаблонів) необхідно залучати експерта, в той час як навчання системи з використанням машинного навчання практично не буде використовувати людські ресурси. Але підготовка розмічених корпусів може зайняти значно багато часу, яке перевищує час, витрачений на написання шаблонів [31].

Формалізація даних. Для роботи алгоритму кластеризації необхідно отриману структуру формалізувати в таблицю. Щоб вирішити це завдання, необхідно скласти вектор характеристик для кожного об'єкта. Це можуть бути як числові значення, наприклад, вік людини, частота народження слів в тексті, зростання або вага і так далі. Також можуть бути категорійні характеристики (якісні). Зупинимося на кількісну характеристику як більш оптимальної для вирішення нашої задачі, так як розрахувати за формулою вектор слова в тексті простіше і логічніше, ніж давати якісну характеристику. Таблиця, як було сказано раніше, має такий вигляд - в рядках вказані номери текстів, по стовпцях - унікальні слова, на перетині частота народження кожного слова в конкретному тексті. Частота народження слова в тексті, на початковому етапі, буде вважатися як елементарне кількість входжень слова в текст. Необхідно спочатку

порівняти результат роботи такої попередньої обробки даних зі звичайною кластеризацией класичними методами. Як поліпшення алгоритму розрахунок буде проводитися на основі структури виділеного семантичного графа. Семантичної граф являє собою спрямований граф, вершинами якого є слова української та російської мов, що відображають основні поняття і їх властивості, а ребра відображають взаємозв'язки між основними поняттями. Частота народження в даному випадку буде розраховуватися наступним чином:

- повний збіг розглянутого графа з графом поточного документа прирівнюється до одиниці;
- якщо розглянутий граф повний (тобто є 2 поняття і взаємозв'язок між ними), то в разі зворотного зв'язку параметр буде дорівнює 0;
- якщо розглянутий граф є в поточному документі, але чи основні поняття, або їх властивості (при наявності), або взаємозв'язок виражена словом синонімом, який зберігає сенс, то кожен синонім забирає від повного збігу 0,2;
- в разі якщо розглянутий граф відсутня в поточному документі, то параметр буде дорівнює 0.

Для роботи алгоритму кластеризації необхідний метод, за яким можна розрахувати близькість об'єктів, тобто необхідно певним чином ввести міру близькості між ознаками таблиці, отриманої в попередньому пункті. В якості запобіжного близькості можна використовувати різні формули. Вибір даної формули є окремим завданням, на якій ми не будемо зупинятися дуже детально. Розглянемо часто використовувані формули, за якими розраховується близькість між ознаками [32].

Евклідова відстань - найбільш поширена функція відстані між ознаками. Дана відстань є геометричною відстанню в багатовимірному просторі і розраховується за формулою 2.4.

$$p(x, x') = \sqrt{\sum_i^n (x_i - x_i')} \quad (2.4)$$

Квадрат евклідової відстані - схожа на першу формулу, але надає більше ваги більш віддаленим один від одного об'єктів (2.5).

$$p(x, x') = \sqrt{\sum_i^n (x_i - x_i')^2} \quad (2.5)$$

Манхеттенська відстань - відстань, що є середнім різниці за координатами [33]. Працює схожим чином з евклідовою відстанню і дає схожі результати, але вплив окремих великих викидів зменшується (2.6).

$$p(x, x_i) = \sum_i^n |x_i - x_i'| \quad (2.6)$$

Відстань Чебишева - використовується в тих випадках, коли потрібно визначити відмінність об'єктів [33]. Факт відмінності встановлюється при розходженні з якої-небудь одній координаті (2.7).

$$p(x, x') = \max(|x_i - x_i'|) \quad (2.7)$$

Степенева відстань - це відстань збільшує або зменшує вагу для розмірності, об'єкти яких сильно відрізняються один від одного [33]. Розраховується за формулою (2.8):

$$p(x, x') = \sqrt[r]{\sum_i^n (x_i - x_i')^p} \tag{2.8}$$

де r і p - параметри, що задаються користувачем.

Якщо ці параметри збігаються то формула перетворюється в відстань Евкліда. Цей параметр відповідає за поступове зважування різниць за окремими координатами, а також відповідальний за прогресивне зважування великих відстаней між об'єктами.

Косинусна міра - для двох векторів це косинус кута між ними [34]. Допомагає виявити пропорційне схожість (2.9).

$$\cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\|_2 \cdot \|\vec{y}\|_2} \tag{2.9}$$

Вибір метрики, як було раніше сказано, по суті окреме завдання. Рішення про те, яку метрику вибрати, лежить на дослідника. Можна зробити припущення, що оптимальним вирішенням для нашої задачі буде використання манхеттенської і косинусної відстані. Манхеттенська відстань дає хороші результати в разі несхожості текстів за смисловим змістом, а косинусна відстань, може дати більш точні результати, якщо тексти близькі за семантикою.

В ході виконання даного підрозділу, було проаналізовані існуючі алгоритми кластеризації та обрано такий, що є найбільш корисним з точки зору вирішення поставленого завдання. На початковому етапі будемо використовувати алгоритми ієрархічної кластеризації та алгоритм kmeans. Дані алгоритми були обрані з огляду на те, що вони легко і швидко реалізуються.

2.3 Огляд систем автоматичного просування сайтів

На сьогоднішній день існує безліч програмних продуктів для автоматичного просування сайтів. Всі вони мають певний набір можливостей, наприклад:

- аналіз програмного коду веб-сторінок;
- відстеження місця сторінок в пошукових системах по запитах;
- занесення сторінок в списки каталогів;
- допомога в складанні семантичного ядра сайту;
- відстеження рейтингу сторінок в пошукових системах (ТІЦ, PR);
- аналіз посилань, спрямованих на сторінки;
- виявлення непрацюючих і тупикових посилань на сторінках;
- допомога в аналізі оптимизационной діяльності конкурентів;
- автоматична закупівля вхідних посилань і ефективне управління ними.

Згідно з дослідженням за 2017 рік, можна зробити висновок про розподіл популярності інструментів автоматичного просування серед компаній, що займаються оптимізацією веб-сторінок. Дослідження зачіпало український та російський сегменти підприємництва і включало понад 40 компаній [12]. Графічне представлення дослідження зазначено в рисунку 2.5.

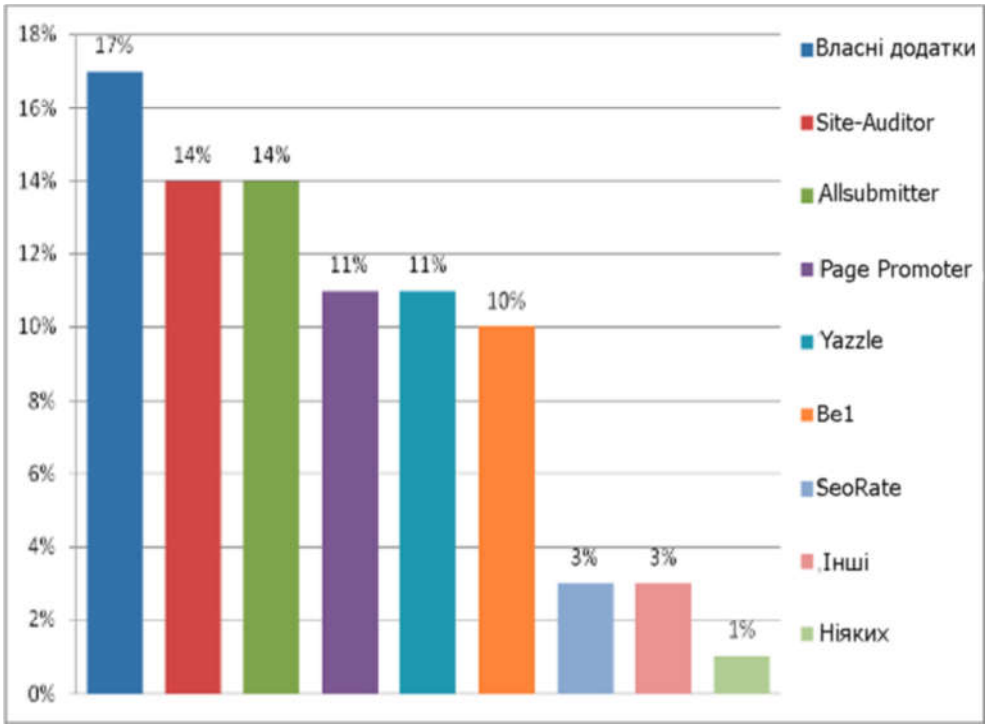

Рисунок 2.5 - Розподіл популярності інструментів автоматичного просування веб-сторінок серед оптимізаційних компаній.

Грунтуючись на цих даних можна зробити висновок про те, що приватні оптимізатори більш воліють користуватись готовими продуктами, а оптимізаційні компанії - навпаки, більш налаштовані на розробку власних програм. Це пов'язано з тим, що готові програмні продукти є більш витратними для підприємств за рахунок корпоративного ліцензування. Приватним оптимізаторам зазвичай недоцільно вдаватися застосування до власних програм через часові, матеріальні та витрати праці на їх створення.

Багато виробників надають можливість пробного використання своїх програм, але як показує практика, функціонал пробних версій часто сильно обмежений і час їх використання не дозволяє виробляти тривалого супроводу SEO-проектів. Всі вони мають схожі можливості, описані в таблиці 2.1.

Таблиця 2.1 - Програмні засоби для автоматичного просування сайтів

Назва	Адреса	Пояснення
Allsubmitter	www.allsubmitter.ru	Даний інструмент дозволяє реєструвати сайти в різних каталогах
Site-Auditor	www.site-auditor.ru	Додаток дозволяє отримувати статистику про авторитетність сторінок в пошукових системах, визначати позиції в пошуковій видачі по запитах, складати семантичне ядро сайту
Be1	www.be1.ru	Онлайн-додаток, що дозволяє проаналізувати сайт для пошукового просування
Yazzle	www.yazzle.ru	Цей додаток є комплексним інструментом пошукового просування сайтів, що дозволяє також визначати вартість вхідних посилань і аналізувати посилання на індексацію в пошуковій системі Яндекс
Page Promoter	www.netpromoter.ru	Додаток дозволяє працювати як зі статистичними пошуковими даними, так і проводити аналіз поточного стану сайту в пошукових системах
SeoRate	www.seorate.ru	Аналітичний інструмент для моніторингу видимості сайту і його рейтингу в пошукових системах
SeoPult	www.seopult.ru	Повністю автоматизована система просування сайтів в пошукових системах

Додаток Allsubmitter. Даний програмний комплекс дозволяє додавати сайти в різні каталоги, пошукові системи, форуми і блоги в автоматичному режимі. Додаток додає посилання на просувний сайт на ресурсах відповідних тематик, піднімаючи тим самим його кількість посилань популярність.

Рисунок 2.6 - Знімок головної сторінки сайту додатки Allsubmitter

Крім цього, програма здатна оцінювати вартість і конкуренцію просування по певних запитах і посиланнях, аналізувати ключові слова і позицію сайту в пошуковій видачі.

Цей додаток є пропрієтарним і вимагає плати за використання.

Додаток Site-Auditor. Програмний комплекс, призначений для аналізу різних параметрів сайту: ТІЦ, число проіндексованих пошуковою машиною сторінок сайту, статистика вхідних посилань на сайт, інформація про знаходження в великих каталогах.

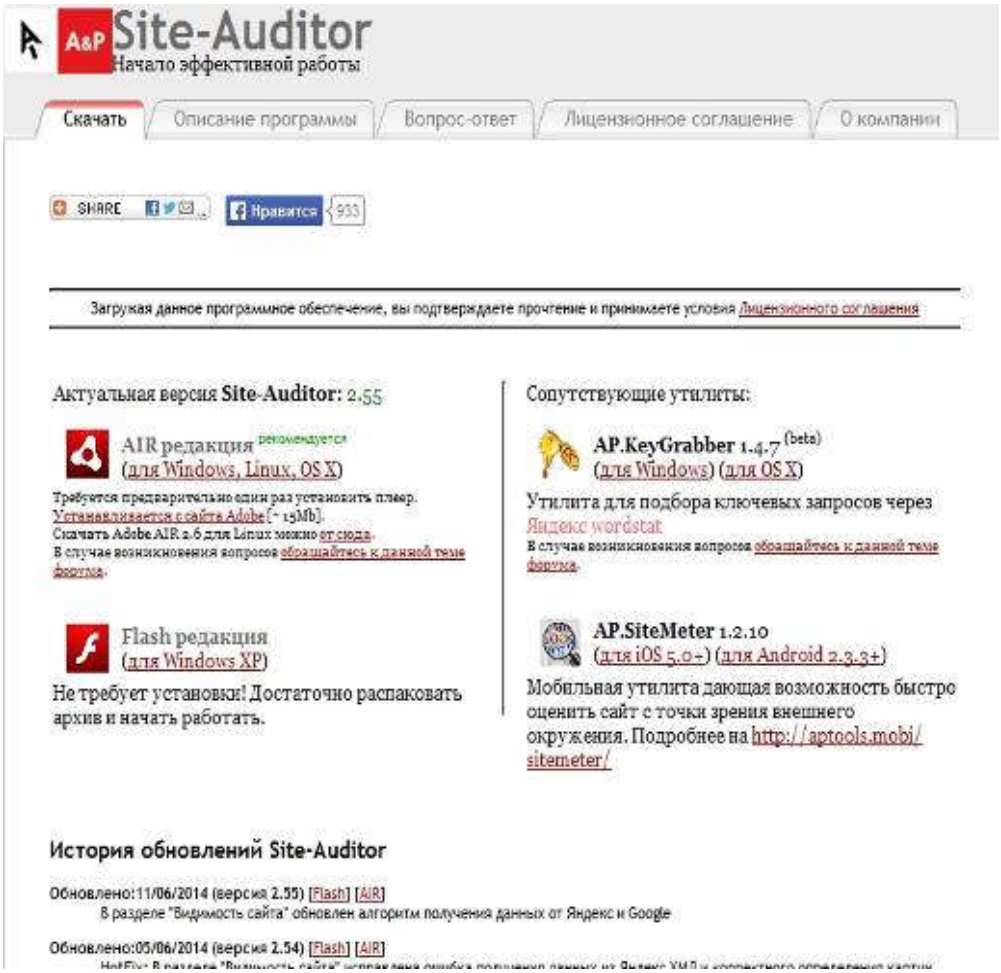

Рисунок 2.7- Знімок головної сторінки сайту додатки Site-Auditor

Крім заявлених функцій, дане програмне рішення дозволяє вести роботу над семантичним ядром сайту, відслідковувати позицію сайту в пошукових системах, проводити аналіз конкурентних посилань, а також виявляти, які зі сторінок сайту найбільше відповідають заявленим вимогам. Цей додаток є вільно поширюваним.

Онлайн-інструмент BE1. Авторські права на який належать Григорію Селезньову. Він дозволяє відстежувати ряд параметрів сайту:

- авторитетність сайту в пошукових системах;
- посилання, спрямовані на сторінки;
- ступінь індексації сайту пошуковими ситемами;
- переглядати системний файл robots.txt;
- отримувати статистику сайту від сервера.

Рисунок 2.8 - Знімок головної сторінки сайту додатки Be1

Цей додаток є вільно поширюваним і не вимагає установки на жорсткий диск комп'ютера.

Додаток Yazzle. Багатофункціональний інструмент аналізу та управління просуванням сайту. Має безліч функцій, серед яких:

- аналіз конкуренції сайту за запитами і даними з сайтів-конкурентів;
- оцінка засобів, необхідних на просування, посилаючись на інформацію з сайтів посилальних бірж;
- аналіз посилань на сторінки сайту;

– збір інформації про відвідуваність сайту з урахуванням певних запитів;
– аналіз вихідного html-коду сторінок;
– пошук і визначення пов'язаних посиланнями мереж сайтів;
– побудова карти сайту;
– формування семантичного ядра сайту;
– відстеження рейтингу сторінок в пошукових системах.

Рисунок 2.9 - Знімок головної сторінки сайту додатки Yazzle

Цей додаток є пропрієтарним і вимагає плати за використання.

Додаток Page Promoter. Крім широкого функціоналу, дає можливість звертатися до розробників за професійною консультацією щодо просування. Додаток призначений для вирішення завдань за двома напрямками:

1) аналіз статистики і моніторинг позицій сайту для спрощення оптимізації та просування його в пошукових системах;
2) інструментарій для фінансового моніторингу та обґрунтування стратегій просування в пошукових системах.

Рисунок 2.10 - Знімок головної сторінки сайту додатки Page Promoter

Крім іншого, додаток здійснює роботу з файлами robots.txt, перевірку вхідних і внутрішніх посилань сайту, допомога при складанні семантичного ядра.

Цей додаток є пропрієтарним і вимагає плати за використання.

Додаток Seorate. Даний інструмент націлений на визначення видимості сайту для пошукових систем по тій чи іншій тематиці. Цей додаток дозволяє:

- оцінювати кількість користувачів, які побачили сайт в пошуковій видачі;
- аналізувати позиції обраних сайтів в пошукових системах не далі 30 позицій від початку списку пошукової видачі;
- аналізувати ключові слова за запитами в пошукових системах;
- визначати видимість сайту для пошукових павуків і кількість потрапили в індекс сторінок;
- огляд посилальної мережі, яку утворює цільової сайт з прямими і

зворотними посиланнями на інші ресурси.

Рисунок 2.11 - Знімок головної сторінки сайту додатки SeoRate

Цей додаток є пропрієтарним і вимагає плати за використання. Крім цього передбачається безкоштовний режим роботи з обмеженням безлічі функцій.

Система SeoPult. Даний інструментарій відрізняється від інших тим, що він є автоматизованою онлайн-системою з просування сайтів. Дана система успішно функціонує з 2008 року. Вона дозволяє розраховувати бюджети просування сайтів, проводити семантичний і авторський аналіз текстів сторінок, в автоматичному режимі працювати з посиланнями, що ведуть на сайт.

Дана система є пропрієтарною і платою за її використання є стягування комісії. Розмір комісії розраховується, виходячи з бюджету просування, і мінімальна її величина становить 10%.

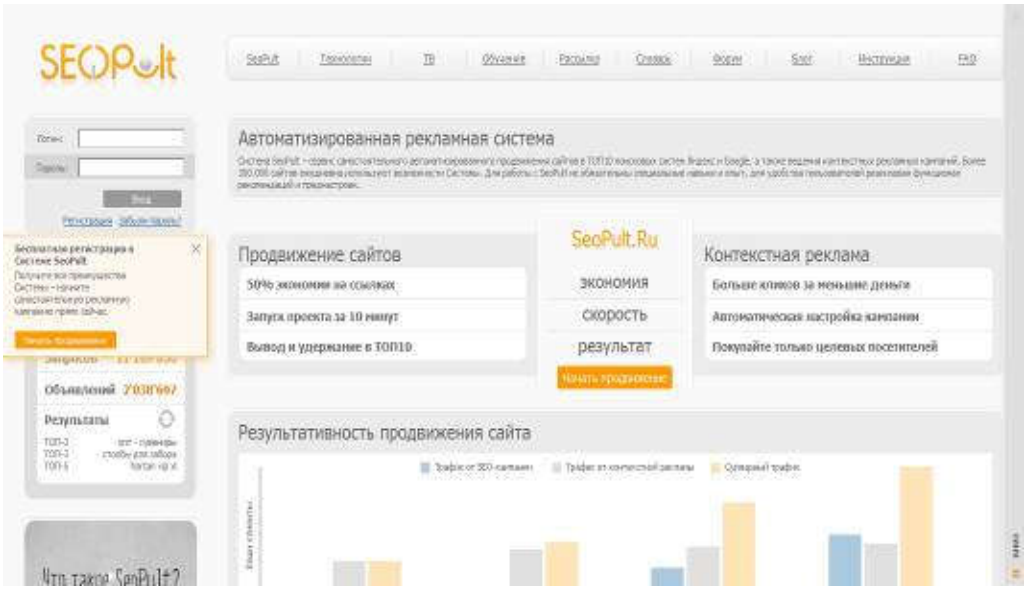

Рисунок 2.12 - Знімок головної сторінки сайту додатки SeoPult

Висновки до другого розділу

В даному розділі були проаналізовані методи кластеризації сематничного ядра сайту, а саме ручний спосіб кластеризації та використання засобів інтелектуального аналізу даних для її виконання. Для кластеризації були обрані методи ієрархічної кластеризації та алгоритм K-means. Дані алгоритми були обрані з огляду на те, що вони легко і швидко реалізуються, ієрархічна кластеризація буде використана для візуалізації дерева кластерів та вибору числа кількості кластерів, а алгоритм K-means – для безпосередього робиття пошукових запитів на кластери.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ЗБОРУ ТА КЛАСТЕРИЗАЦІЇ СЕМАНТИЧНОГО ЯДРА САЙТУ

3.1 Вибір серидовища реалізації програмного додатку

В ході опису предметної області, що досліджується в дипломній роботі, а також методик і засобів виконання завдань по оптимізації і просуванню сайтів в пошукових системах виділився ряд проблем, а саме:

1) характер існуючих робіт по оптимізації внутрішнього наповнення сайту, а також аналіз цільових слів, під які ведеться оптимізація, вимагає великої кількості часу. Дана вимога, сильно скорочує кількість завдань, які виконуються SEO-спеціалістом за робочий час, знижуючи тим самим його продуктивність;
2) існуючий набір програмних засобів в умовах жорстокого капіталізму і всеосяжних авторських прав умовно можна розділити на дві категорії. До першої можна віднести програмні інструменти безкоштовні або умовно безкоштовні. До другої - платні програми, що вимагають вливань коштів;
3) в сучасних умовах постійної зміни інтересів користувачів, а також все більш жорсткої політики пошукових систем щодо рейтингу сайтів, виникає потреба в постійному аналізі та поліпшення контентного наповнення сайту.

Для вирішення цих проблем пропонується створити програму, яка здійснює такі функції:

- збір статистики запитів пошукової системи Google про кількість переходів по певним ключовим словом;
- збір статистики за запитами, які були здійснені в пошуковій системі

Google в межах однієї сесії браузера;

- отримання списку синонімів ключових слів, на основі пошукової статистики системи Google.

Програмний додаток реалізований на мові Python 3.7 в середовищі розробки Spyder, що входить в пакет Anaconda3. Цей вибір був зроблений з ряду причин, а саме:

- простота самої мови;
- велика кількість існуючих бібліотек, які допоможуть в вирішенні поставлених задач;
- висока можливість інтеграції в різні серидовища;
- швидкість опрацювання даних.

3.2 Опис роботи та структури програмного додатку

У програмі реалізована робота з пошуковою системою Google. Вибір на користь цієї системи зроблений з огляду на те, що Google та Yandex є найбільш великими постачальниками пошукових послуг Інтернету на даний момент, але після введення указу Президента України "Заборона Інтернет-провайдерам надання послуг з доступу користувачам мережі Інтернет до ресурсів сервісів «Mail.ru» та соціально-орієнтованих ресурсів «Вконтакте» та «Одноклассники»" до яких належить і «Yandex», варто відмовитись від оптимізації під дану пошукову систему[57].

В основі програми лежить алгоритм K-means, котрий реалізує кластеризацію пошукових запитів, але в ньому є один суттєвий недолік –

алгоритм не може вирахувати кількість кластерів[32]. Для того щоб усунути цей недолік можна скористатися двома методами:

- використати алгоритм Minimum Description Length;
- ручний підбір кількості кластерів.

З урахуванням того, що текстові дані погано обробляються машинними засобами та будуть формалізовані, був використаний другий варіант з огляду на те, що користувач (людина) краще реалізує семантичний аналіз тексту ніж алгоритм, але для того щоб цей вибір був простіший було прийняте рішення використати ієрархічну кластеризацію з бібліотеками numpy та scipy для візуалізації дерева виділених кластерів пошукових запитів.

Для формалізації даних була використана лемматизація (переведення слів в безвідмінкову форму) та векторизація алгоритмом TF-IDF. Далі буде детально описана робота програми.

3.2.1 Отримання пошукових запитів з допомогою API

Даний блок реалізує в своїй роботі API-інтерфейси пошукових систем. Технологія API (англ. Application Programming Interface) являє собою інтерфейс програмування додатків або, як ще кажуть, інтерфейс прикладного програмування[41]. Це готовий набір бібліотек методів, класів, процедур, констант і інших даних, що надаються тим чи іншим сервісом для зовнішніх програмних продуктів. Дана технологія використовується в багатьох сферах інформаційних технологій, в тому числі і в сфері пошукових систем. Доступ до API надається пошуковими системами, щоб підвищити зручність роботи з їх даними для програмістів.

API в таких випадках необхідний для автоматизації подібних процесів і

зручний в роботі з великими обсягами даних. Наприклад, розробник сайту може написати програму, обновляющую оголошення при зміні цін на товари або послуги або організувати процес вивантаження статистики для аналізу. Наша ситуація заснована на необхідності зібрати дані ресурсу AdWords, що при великих обсягах вручну досить проблематично.

Для роботи з API AdWords компанії Google перш за все необхідно отримати доступ до багатьох сегментів ресурсу, що перш за все зобов'язує будь-якого охочого зареєструватися в системі Google Cloud Platform.

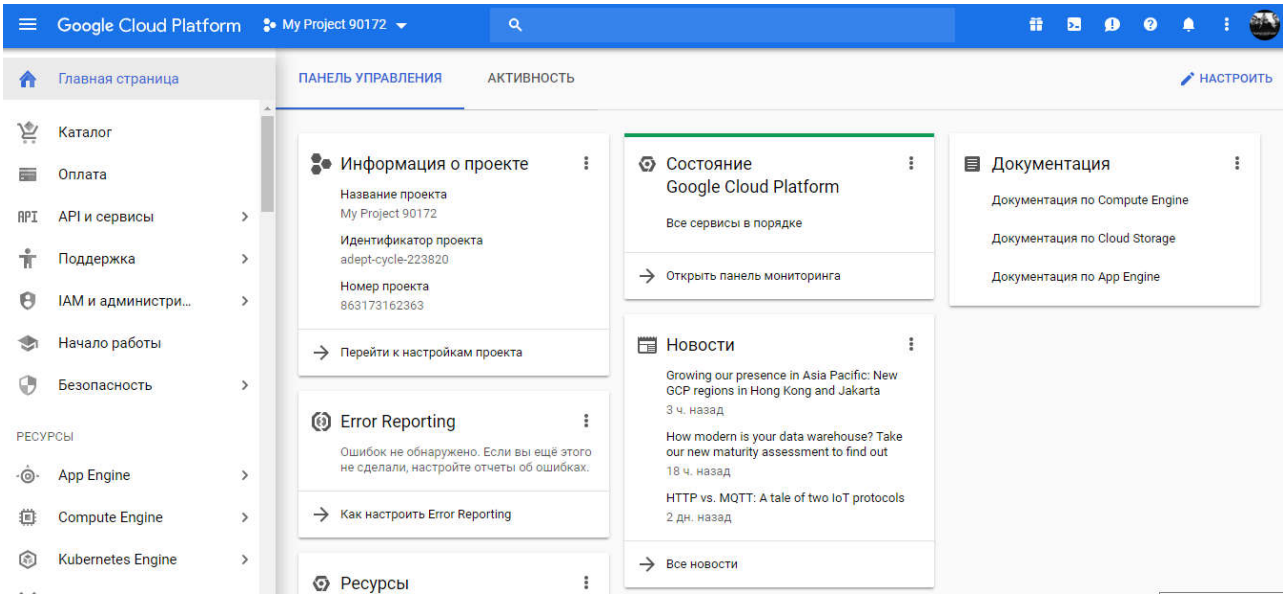

Рисунок 3.1 – Вікно аккаунту Google Cloud Platform

Так, щоб продовжити роботу, крім реєстрації в самому сервісі AdWords, необхідно авторизуватись за допомогою OAuth2. В результаті цього клієнтська програма, що використовує AdWords API, може отримати доступ до облікового запису AdWords без адреси електронної пошти та пароля користувача.

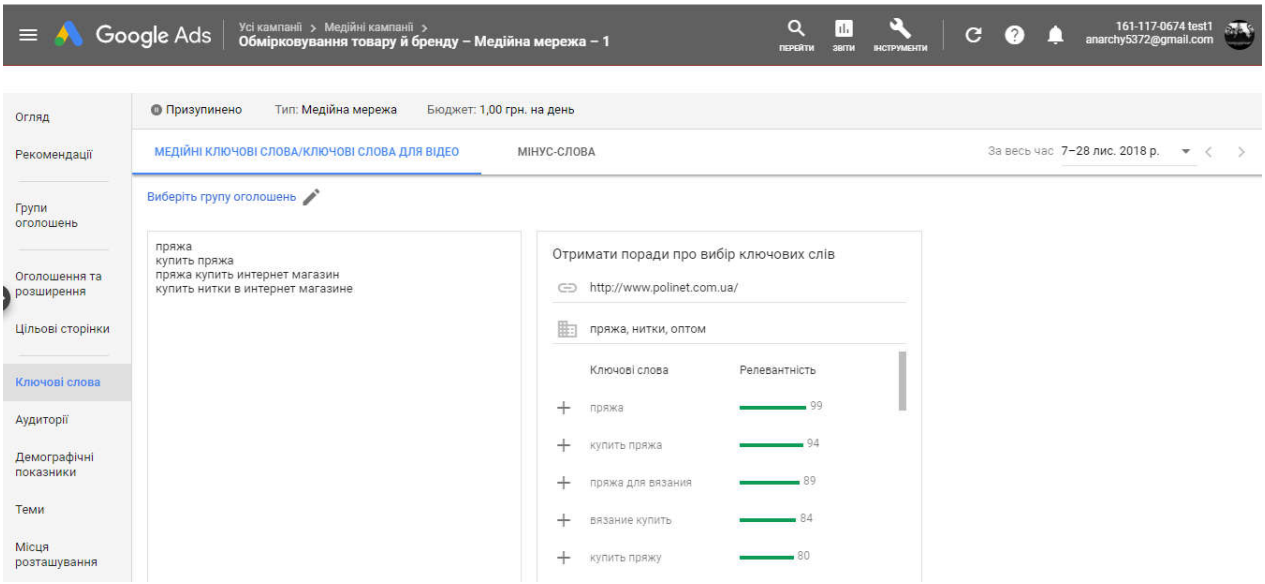

Рисунок 3.2 – Вікно проекту в Google AdWords

OAuth2-авторизація - це механізм, що дозволяє третій особі, в даному випадку з додатком, здійснювати дії від імені користувача без використання пароля та логіна[43]. Така авторизація використовує OAuth2-токени - рядки, що містять в собі ряд зашифрованих даних:

- аккаунт
- ідентифікатор додатки
- перелік прав і дозволених дій.

За даним списку відразу можна зробити, що кожен токен унікальний, тому що містить у собі два унікальних ключа: логін користувача і ідентифікатор самого додатка.

Існує тестовий і повний доступ до API AdWords. Тестовий зручний при навчанні, так як дає додатком права на роботу в спеціальному тестовому середовищі, де всі дії реальні, але робота здійснюється з нереальними даними. Повний активний як в тестовому середовищі, так і в реальному і дає права

керувати рекламою.

Запити здійснюються в форматі JSON (англ. JavaScript Object Notation). Даний формат є текстовим і заснований на мові JavaScript. Як правило, він використовується разом з цією мовою. Даний формат має мовну незалежність, лаконічність і сумісність з багатьма мовами програмування. Синтаксис даного формату має два види:

1) набір пар «ключ: значення». У різних мовах програмування це може бути об'єктом, словником, структурою і т.д. Ключ може бути виключно рядком, а значенням може бути будь-яка форма;
2) упорядкований набір значень. Даний набір в багатьох мовах програмування зустрічається як вектор, послідовність, масив або список значень;

Перед здійсненням запитів, додаток встановлює зв'язок з сервером Google, використовуючи спеціальний токен. Цей токен потрібен для встановлення інтерфейсу між сервером пошукача і програмою.

Функціонал обміну даними з серверами цієї пошукової системи здійснюється за рахунок використання спеціалізованих бібліотек, доступних для завантаження на офіційному ресурсі Google AdWords.

З допомогою можливостей API AdWords ми отримуємо список ключових слів і передаємо їх в модуль нормалізації. Лістинг модуля отримання ключових слів з сервісу Google описано в додатку А.

3.2.2 Нормалізація отриманих даних

Перед векторизацією даних потрібно їх нормалізувати. Цей процес називається лемматизацією - приведення словоформи в нормальну форму слова. Для лемматизації отриманих пошукових запитів використана бібліотека pymorphy2 написаний на мові Python, а саме його клас MorphAnalyzer[34].

Pymorphy2 дозволяє:

- приводити слово до нормальної форми (наприклад, "люди -> людина", або "гуляв -> гуляти");
- ставити слово в потрібну форму (ставити слово у множину, змінювати відмінок слова і т.д.);
- повертати граматичну інформацію про слово (число, рід, відмінок, частина мови і т.д.).

При роботі pymorphy2 використовується словник OpenCorpora; для незнайомих слів будуються гіпотези. Бібліотека досить швидка: зараз швидкість роботи - від декількох тисяч слів в секунду до ста тисяч слів в секунду (в залежності від виконуваної операції, інтерпретатора і встановлених пакетів); споживання пам'яті - 10 ... 20Мб.

Словники поширюються окремими пакетами:

- pymorphy2-dicts-ru для російської мови;
- pymorphy2-dicts-uk для української мови (експериментальний).

Морфологічний аналіз - це визначення характеристик слова на основі того, як це слово пишеться. При морфологічному аналізі НЕ використовується інформація по сусідніх словах.

У pymorphy2 для морфологічного аналізу слів є клас MorphAnalyzer.

```
>>> import  pymorphy2
>>> morph  =  pymorphy2.MorphAnalyzer ()
```

За замовчуванням вживається словник для російської мови; щоб замість російського включити український словник, потрібно встановити пакет pymorphy2-dicts-uk[35]:

```
>>> morph  =  pymorphy2.MorphAnalyzer (lang = 'uk')
```

Екземпляри класу MorphAnalyzer зазвичай займають приблизно 15Мб оперативної пам'яті, тому що завантажують в пам'ять словники, дані. Тому варто організувати код так, щоб створити екземпляр MorphAnalyzer заздалегідь і надалі працювати з цим єдиним екземпляром.

Після опрацювання пошукових запитів в класі MorphAnalyzer дані передаються на Tf-Idf векторизатор для подальшої обробки.

3.2.3 Векторизація даних в TfidfVectorizer

TF-IDF (від англ. TF — term frequency, IDF — inverse document frequency) — статистичний показник, що використовується для оцінки важливості слів у контексті документа, що є частиною колекції документів чи корпусу. Вага (значимість) слова пропорційна кількості вживань цього слова у документі, і обернено пропорційна частоті вживання слова у інших документах колекції.

TF-IDF часто використовується для подання текстових даних (в даному випадку – текстових фраз) у вигляді числових векторів, що відображають

важливість використання кожного слова з деякого набору слів (кількість слів набору визначає розмірність вектора)[34]. Подібна модель називається векторною моделлю і дає можливість порівнювати тексти, порівнюючи їх представляють вектора в певний метриці (евклідова відстань, косинусна міра, манхеттенська відстань, відстань Чебишова та інші), тобто виконувати кластерний аналіз.

Term frequency - це частотність слова, наскільки часто слово зустрічається в документі. Логічно припустити, що в довгих документах слово може зустрітися в великих кількостях, тому абсолютні числа тут не підходять. Тому застосовують відносні - ділять кількість разів, коли потрібне слово зустрілось в тексті, на загальну кількість слів у тексті.

Inverse Document Frequency - це зворотна частотність документів. Вона вимірює безпосередню важливість слова. Тобто, коли ми вираховували TF, всі терміни вважаються рівними за важливістю один одному. Але всім відомо, що, наприклад, прийменники зустрічаються дуже часто, хоча практично не впливають на зміст тексту. Тому для того щоб усунути цю проблему є IDF. Він вважається як логарифм від загальної кількості слів, поділений на кількість слів, в яких зустрічається потрібне слово.

В Python векторизатор TF-IDF присутній в бібліотеці scikit-learn. Ініціалізація здійснюється наступним чином:

>>> from sklearn.feature_extraction.text import TfidfVectorizer.

В даному випадку потрібно створити векторизатор лише з одним параметром - мінімальною частотою слова. Для векторизатора cv мінімальна частота слова, яке він враховує дорівнює двійці, тобто всі слова з частотою вживання менше двох – не будуть враховуватись при побудові вектора. Якщо слово зустрічається тільки один раз у всіх ключових фразах - то немає сенсу

займати оперативну пам'ять подібною інформацією. Ми не зможемо знайти будь-які відповідні перетини по слову, у якого немає пари.

Для зручності було прийняте рішення зробити один цілісний клас лемматизації та векторизації, котрий має наступний вигляд:

```
>>> class LemmTfIdfVectorizer(TfidfVectorizer):
>>>    def build_analyzer(self):
            analyzer = super(TfidfVectorizer, self) .build_analyzer()
            return lambda x: (m.parse(word) [0].normal_form for word in analyzer(x))
```

Для нього існує кілька загальних налаштувань[34]:

- stop-words - список слів, які не враховуватимуться при векторизації;
- token_pattern - регулярний вираз, за яким рядок розбивається на токени;
- max_df - токени мають частотність вище цього значення не будуть враховуватися (коефіцієнт - 0.9 означатиме що 10% найбільш часто вживаних слів будуть відкинуті);
- min_df - токени мають частотність нижче цього значення не будуть враховуватися (коефіцієнт - 0.1 означатиме що 10% найбільш рідкісних слів будуть відкинуті).

Приклад роботи класу з ручним набором фраз показаний на рисунку 3.3.

	купити	пряжа	україна	опт
купити пряжу	0.522842	0.673255	0.00000	0.00000
купити пряжу україна	0.453295	0.767495	0.391484	0.00000
купити пряжу в україні оптом	0.391484	0.522842	0.504107	0.66284

Рисунок 3.3 – результат роботи лемматизації та векторизації класу LemmTfIdfVectorizer.

Після отримання нормалізованих та векторизованих даних переходимо до безпосередньо самої кластеризації.

3.2.4 Ієрахрічна кластеризація

Як вже було згадано раніше, для кластеризації буде використаний алгоритм K-means, який не має заздалегідь визначеного числа кластерів. Тому щоб вирішити цю задачу було вирішено скористатись ієрархічною кластеризацією, та візуалізацією її результатів роботи[31].

Ієрархічні алгоритми не розбивають вибірку напряму на кластери, а створюють своєрідну систему вкладень. На виході ми отримуємо дерево кластерів, коренем якого є вся вибірка, а гілками - найбільш дрібні кластери.

Ініціалізація даного алгоритму в Python відбувається наступним чином (лістинг програми описаний в додатку Б):

```
>>> from scipy.cluster.hierarchy import linkage, dendrogram, fcluster.
```

Для візуалізації результатів використана бібліотека mathplotlib:

```
>>> import matplotlib.pyplot as plt
>>> from matplotlib import rc
>>> …
>>> font = {'family': 'Verdana', 'weight': 'normal'}
>>> rc('font', **font)

>>> fig, ax = plt.subplots()
>>> dendrogram(link_M,
        ax = ax,
```

```
        labels = keywords,
        leaf_font_size=8)
>>> plt.show()
```

Результатом таких дій буде дерево кластерів зображене на рисунку 3.4.

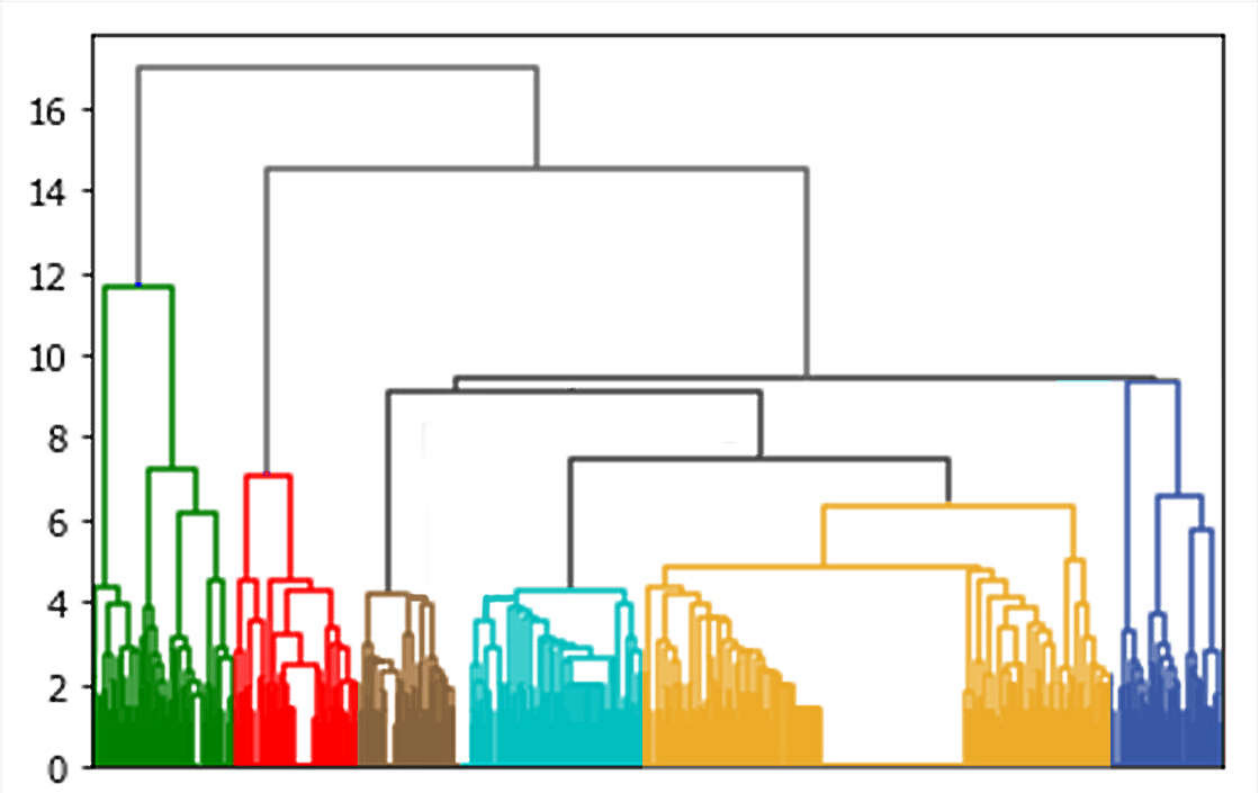

Рисунок 3.4 – Результат роботи ієрархічної кластеризації.

В даному випадку програма вирішила, що оптимальним числом кластерів для пошукових запитів є шість, але варто відзначити що відстань між гілками п’ятого виділеного програмою кластера є значною, а це в свою чергу означає що лексичні значення ключових слів у даному кластері сильно

різняться між собою. В з’язку з цим вирішено розділити цей кластер на два окремі. Тому в наступному кроці вирішено використовувати сім кластерів.

3.2.5 Кластеризація пошукових запитів з допомогою алгоритму K-means

Так як у попередньому розділі було описано теоретичну основу даного методу кластеризації, не будемо на цьому зупинятись. Варто лише згадати, що метою даного методу є розділення n спостережень на k кластерів, так щоб кожне спостереження належало до кластера з найближчим до нього середнім значенням. Так як основною проблемою цього методу є визначення кількості кластерів, для її вирішення була проведена попередня ієрархічна кластеризація, в ході виконання якої було вирішено використовувати сім кластерів.

Даний алгоритм кластеризації у мові програмування Python є частиною бібліотеки scikit-learn, та ініціалізується наступним чином[35]:

```
>>> from sklearn.cluster import KMeans
```

Нижче описаний лістинг підготовчої частини, відбувається завантаження ключових сів з попередньо завантаженого зовнішнього текстового файлу, заповнюємо список стоп-слів, якщо це потрібно. Повний лістинг даного модуля описаний в додатку В.

```
>>> k_means = KMeans(n_clusters=int(np.round(np.divide(len(keywords), 7))),
                     init='k-means++',
                     n_init=10,
                     max_iter=300,
                     tol=0.0001
                    )
>>> k_means.fit(x)
```

Для настройки алгоритму K-Means використовують такі основні параметри:

- n_clusters - кількість кластерів, на які будуть ділитися дані;
- max_iter - максимальна кількість ітерацій. Робота буде примусово зупинено при досягненні цього числа. Іноді алгоритм «залипає» на деяких видах даних, і вказується конкретне статичне число, яке дозволяє уникнути нескінченної роботи;
- n_init - кількість повторних ініціалізацій алгоритму з різними початковими центроїдами. Чим більше ініціалізацій - тим краще кластери описують реальну структуру даних і тим довше працює алгоритм;
- tol - довірча межа, при досягненні якого кластеризація буде зупинена;
- n_jobs - кількість паралельних потоків роботи алгоритму;
- random_state - параметр дозволяє домогтися відтворюваності одних і тих же результатів при різних запусках алгоритму.

Для збереження результатів роботи програми використовуються наступні команди:

```
>>> with open ('k-means.csv', 'w') as f:
>>>   for cln in dct:
>>>       for keyword in dct [cln]:
>>>           f.write('{};{}\n'.format(cln, keyword))
```

Після проведення кластеризації результати роботи збережені у файлі “k-means.csv” (додаток Г). Структура файлу наступна: у файлі знаходяться ключові слова та присвоєні їм мітки кластерів.

3.3 Оптимізація сементичного ядра сайту

Сайт http://www.polinet.com.ua/ містить друкований матеріал (рис. 3.5). Навігація здійснюється за допомогою меню зліва.

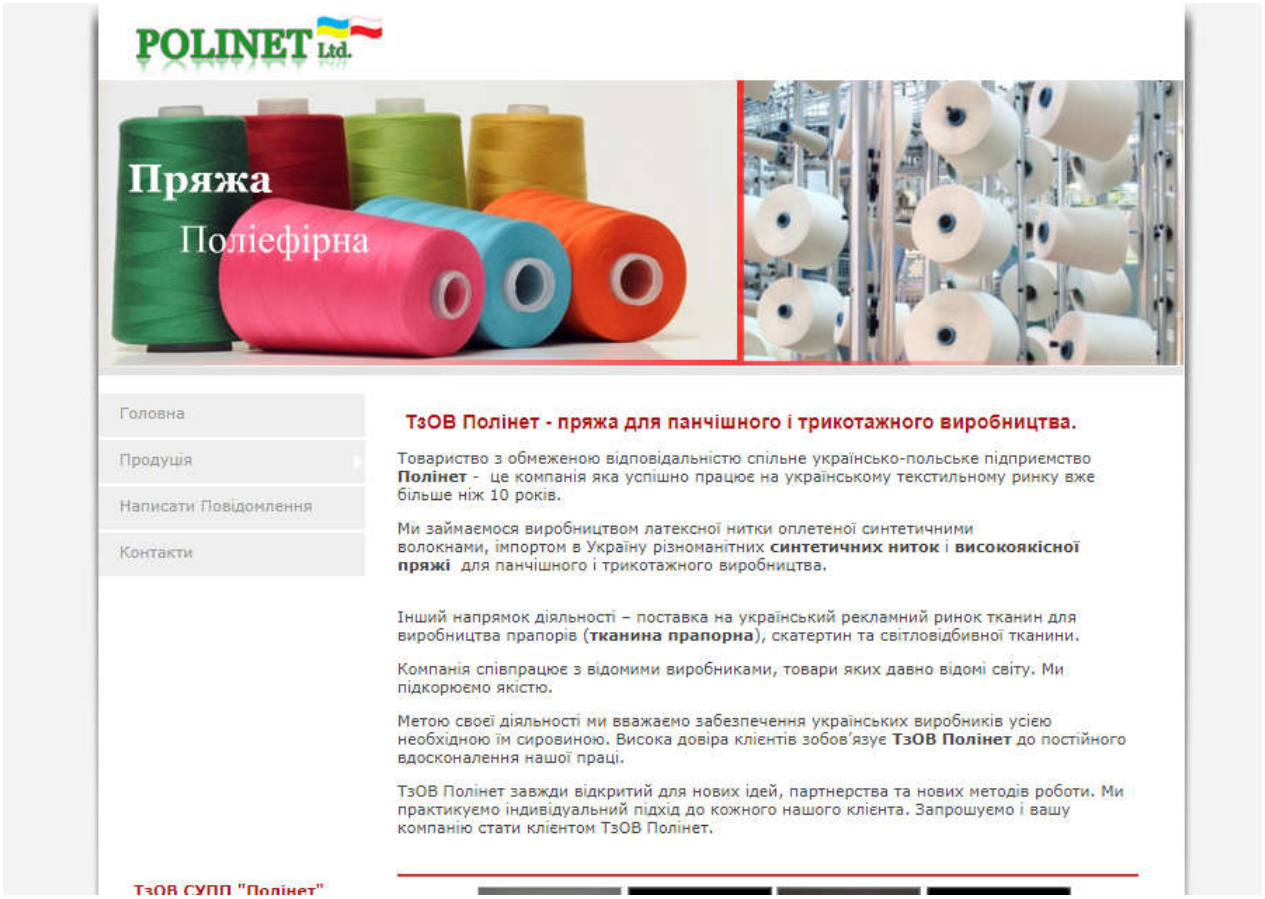

Рисунок 3.5 – Головна сторінка сайту «Полінет»

Матеріал, розміщений на сайті, націлений на інформування клієнтів про діяльність компанії і залучення нових замовлень. На сайті надана інформація:

- про компанію;
- продукція;

- контакти;
- зворотній зв'язок.

Розроблений сайт доповнений розділами: прайс-лист і оформити замовлення. У розділі прайс-лист розташований список послуг, їх вартість і можливість замовлення необхідної послуги. При виборі послуги, перейшовши на сторінку «написати повідомлення» оформити замовлення, заповнивши необхідні поля, її можна буде замовити, відправивши про це заявку.

Актуальність вибору в якості інформаційного ресурсу - web-сайт крім усього іншого зумовлена тим, що інформація, розміщена на сайті доступна як власнику, так і користувачеві, з будь-якого місця, обладнаного комп'ютером та інтернетом.

З розвитком Інтернет і збільшенням числа користувачів багато організацій створюють для своїх клієнтів не просто інформаційні сайти, а цілі Інтернет портали, список функцій яких досить широкий. Наявність подібного ресурсу підвищує лояльність клієнтів, і робить організацію більш конкурентоспроможною.

Аналіз структури сайту - важливе питання, тому що від нього буде залежати зручність користування вашим сайтом.

На етапі аналізу концепції web-сайту створюється документ, який служить вихідним матеріалом для створення web-сайту: розробки сценарію, графічної концепції і структури, програмних інструментів, що забезпечують необхідні функціональні ресурси, і т.д. У сценарії повинні бути визначені: основні структурні рішення організації web-сайту, які конкретно інформаційні матеріали опубліковуватимуться на сайті їх обсяг, які функціональні можливості надані відвідувачам сайту і адміністратору, яким чином будуть оновлюватися інформаційні матеріали і контролюватися відвідуваність web-

сайту. Далі опишемо структуру web-сайту (таблиця 3.1).

Головна сторінка. Головна сторінка це обличчя сайту. Тут зазвичай наводиться коротка приваблива інформація про напрямки діяльності компанії, послуг, що надаються. Основне призначення головної сторінки web-сайту - розповісти користувачам про компанію, її переваги. Розроблений сайт надає продукцію, такого виду як пряжа, нитки різних типів, імпорт в Україну різноманітних синтетичних ниток і високоякісної пряжі для панчішного і трикотажного виробництва. Це й відображено на головній сторінці шляхом розміщення на ній прямих посилань на головну інформацію про підприємство, її послуги.

Каталог продукції. Розділ «Продукція» - каталог товарів, пропонованих компанією. Це сторінка, де міститься повний перелік послуг. У розділі «Продукція» існують підрозділи з окремим прайс-листом і можливістю оформити замовлення.

Написати повідомлення. На цій сторінці представлена інформацію про зв’язок з компанією, оформлення замовлень.

Контакти. Розділ «Контакти» - містить контактні дані компанії. У ньому міститься максимально можлива контактна інформація, інформація про відповідальних осіб, інформація про режим роботи, схема проїзду, форма відправки електронного повідомлення - все, що допоможе відвідувачу оперативно зв'язатися з підприємством.

Таблиця 3.1 - Інформаційна структура сайту ТОВ «Полінет»

Розділ	Категорія
Головна	
Продукція	Пряжа
	Нитка поліестрова
	Нитка поліамідна
	Нитка поліпропіленова
	Еластомірна мононитка
	Тканина прапорна
	Гумка латексна
Написати повідомлення	
Контакти	

Залежно від подальшого розвитку компанії сценарій сайту може бути усічений або доповнений специфічними елементами, які можуть з'явитися або зникнути на підприємстві. Сценарій web-сайту повинен повністю відповідати меті створення сайту і бути орієнтований на відповідну цільову аудиторію.

Web-сайт, представлений в даному дипломному проекті, має просту і зрозумілу структуру.

Сайт розміщено на платному хостингу за адресою: https://ua.ukrline.com.ua (рис.3.6).

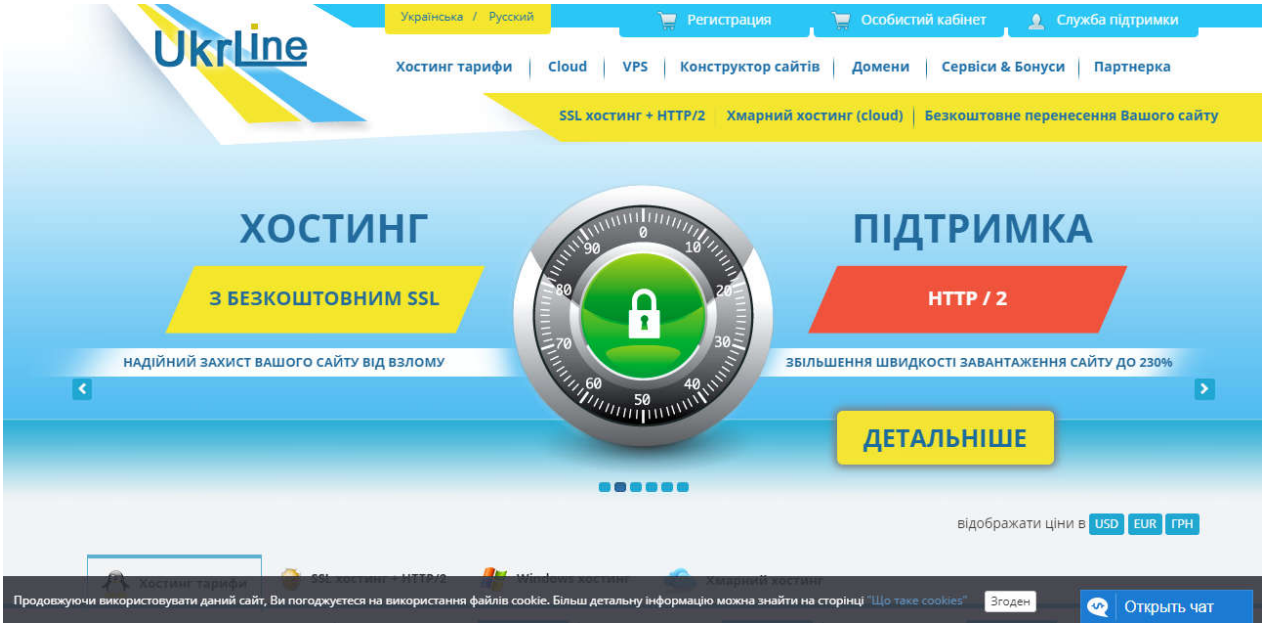

Рисунок 3.6 – Головна сторінка сайту «УкрЛайн»

«ТОВ Полінет» - назва сайту. Навігація по сайту, за винятком переходу на головну сторінку, здійснюється за рахунок меню зліва або кліка по логотипі сайту.

В основній частині сайту представлена інформація про підприємство, є нижня панель, в якій коротко описані важливі розділи сайту з переходом на повний текст.

У нижній частині сайту (footer) розташовується контактні дані, сторінки сайту і можливість підписатися на розсилку сайту.

Редагування сторінок власником може здійснюватися безпосередньо, зі сторінки сайту, натисканням на напис «Редагувати сторінку». Вхід в панель управління сайтом здійснюється за адресою:http://ua.ukrline.com.ua/billing.php. Також був задіяний локальний сервер Vertrigo - це спеціальний додаток, яке встановлюється на комп'ютер і перетворює його в повноцінний сервер.

VertigoServ - це високопрофесійний і простий в установці набір, що складається з Apache (HTTP web-сервер), PHP (скриптова мова програмування), MySQL (багатопотокова, розрахована на багато користувачів СУБД), SQLite (вбудований двигун баз даних), SQLiteManager (багатомовна web-утиліта для управління БД SQLite), PhpMyAdmin (утиліта, написана на PHP для адміністрування БД MySQL) для платформи Windows. Файли проекту розміщуються в папці www, яка є однією з директорій сервера [37].

Директорія також містить піддиректорію «images», в якій розміщені всі зображення, які використовуються в темі. Підтримуються як файли зображень з розширенням «.jpeg», так і файли зображень з розширенням «.png».

Локальний сервер Vertrigo, як уже говорилося вище, дозволяє допрацьовувати і переглядати всі зміни без використання хостингу і віддаленого сервера. Для початку роботи з сервером необхідно запустити Vertrigo. Це робиться за допомогою, створюваного при установці дистрибутива, значка «VertrigoServ». При натискання на значок «VertrigoServ» з'являється вікно, що відображає стан запуску програми.

Після цього можна приступати до перегляду сторінок в браузері.

Після установки Vertrigo на комп'ютер і запуску, можна приступити до створення бази даних.

Необхідно запустити браузер і в адресному рядку ввести http://localhost/phpmyadmin, після чого ввести дані для входу (ім'я: root; пароль: vertigo) і потрапляємо в панель для створення нових БД[40].

Після чого, слід ввести нове ім'я БД і натиснути створити. Потім слід створити віртуальний домен для майбутнього сайту. Для цього необхідно перейти в директорію C:\Program Files\VertrigoServ\www і створити в ній папку polinet (ім'я може бути довільним: site.my або просто site). Усередині створеної

директорії розпакувати файли дистрибутив. Потім запустити браузер і в адресному рядку написати http://localhost/polinet/.

Необхідно натиснути на кнопку «Створити файл налаштувань». На наступній сторінці потрібно натиснути «Вперед!» І ввести дані бази, натиснути кнопку «Відправити». Після того, як установка завершена, з'являється вікно. Натискаємо «Увійти».

Постійна адреса панелі адміністратора http://localhost/polinet/wp-admin.php. У разі, якщо пароль втрачено, він висилається на вказаний при установці e-mail.

Доступ до сайту здійснюється за адресою http://polinet.com.ua/. До редагування зовнішнього вигляду за адресою http://polinet.com.ua/wp-login.php. При переході за даною адресою з'являється вікно для введення логіна і пароля. Після введення потрібних даних здійснюється перехід на адресу http://polinet.com.ua/wp-admin/, де відкривається стандартне вікно і панель інструментів.

Для заміни файлів, прав доступу і інших параметрів необхідно перейти за адресою https://ua.ukrline.com.ua/viewinvoice.php?id=25497, ввести запитувані логін і пароль, які вказуються при реєстрації і вибравши необхідний розділ внести необхідні зміни.

Перед початком просування сайту слід оцінити його конкурентів. Для нашого підприємства, наприклад, вибираємо пошуковий запит «пряжа» і вводимо його в сервісі Google AdWords. У пошуковій видачі насамперед дивимося на кількість оголошень в Google AdWords, на наш запит їх чотири. Це означає, що цей запит є таким, що має низьку конкурентоспроможність. Основний сайт підприємства по цьому запиту стоїть на 97 позиції. Для складання семантичного ядра запитів скористаємось програмою, робота якої

описана в попередніх підрозділах. Для сайту http://polinet.com.ua/ підібрано наступне семантичне ядро: високочастотні запити – пряжа, нитки; середньочастотні – нитка поліестерна; низькочастотні – пряжа бавовняно-поліестерова. Проведемо аналіз популярності слів, а також зразкового кількості запитів по цій тематиці (в місяць). Оптимізація сайту для пошукових систем:

Розроблений сайт варто оптимізувати наступним чином:

- ключові слова повинні бути в назві сайту, в назві статей, в самому тексті і бажано в доменному імені;
- заголовок статті виділено тегом <h1>, назва підприємства тегом <h2>, на одній сторінці слід розміщувати тільки один тег <h1> і кілька <h2> - <h6>;
- зображення в тезі alt подпісани ключовими запитами, тому що по цих запитах Google видає картинки;
- посилання на сторінки називаються по темі сторінки без зайвих параметрів id ідентифікаторів.

Ефект від оптимізації сайту почне виразно проявлятися в період часу від двох тижнів до місяця підтримки (час внесення сайту в каталоги і рейтинги). Результатом роботи є оптимізація сайту підприємства, який надає необхідну інформацію для замовників.

З'явилися розділи, що допоможуть підвищити ефективність обслуговування і звільнять час працівників для інших справ.

В даний момент, на сайті ТОВ «Полінет» реалізований розділ «оформити замовлення». У розділі пропонується можливість замовлення необхідної послуги. При виборі послуга вноситься в кошик і, перейшовши на сторінку оформити замовлення, заповнивши необхідні поля, її можна буде замовити, відправивши про це заявку.

Описано керівництво по адмініструванню сайтом та розміщення його в мережі Інтернет, а також конкретні методи щодо подальшого просування.

Оптимізація семантичного ядра сайту http://polinet.com.ua/, відбувалась наступним чином:

- отримання повного списку ключових запитів;
- очистка невідповідних тематиці сайту ключових слів;
- групування та кластеризація семантичного ярда;
- аналіз структури сайту та його реструктуризація;
- роміщення ключових запитів в тегах <h1> - <h6>;
- заміна та використання ключових слів в тегах <alt> зображень;
- оптимізація url сторінок транслітерацією для кращого індексування пошуковими роботами.

Завдяки таким діям сайт http://polinet.com.ua/ був піднятий в пошуковій видачі пошукової системи Google з 97-ої до 92 позиції, тобто на п’ять позицій вверх.

Висновки до третього розділу

В ході роботи над пошуковою оптимізацією сайту «Полінет» (http://polinet.com.ua/) були створені програмні додатки для збору і кластеризації СЯ сайту на основі аглоритмів K-means та ієрархічної кластеризації з наступними можливостями:

1) автоматичний збір ключових слів з допомогою сервісу Google AdWords API;

2) візуалізація дерева кластерів для отримання оптимальної кількості кластерів;
3) кластеризація пошукових запитів зі схожим лексичним значенням;
4) збереження результатів роботи в «.csv» форматі.

Отримане семантичне ядро було використане для проведення оптимізації над сайтом «Полінет» та аналізу структури сайту. Результатом проведеної оптимізації є підняття сайту на п’ять позицій в пошуковій системі Google.

Алгоритм, що запропонований в даній роботі, може бути впроваджений в системи адміністрування сайтів або в засоби підтримки роботи seo-фахівців для підвищення повноти та зниження часу розробки семантичного ядра сайтів.

ВИСНОВКИ

В ході виконання даної дипломної роботи був проведений детальний аналіз літературних джерел за темами «пошукова оптимізація сайту» та «інтелектуальний аналіз даних». За результатами проведеного дослідження можна сформулювати наступні висновки:

1) проаналізовано предмет, основні методи роботи і програмні засоби пошукової оптимізації сайтів;
2) виявлено найбільш суттєві проблеми, що виникають при роботі фахівця з пошукової оптимізації, в даному випадку витрати часу на складання правильного семантичного ядра;
3) запропоновано шляхи вирішення виявлених проблем і на основі даних пропозицій створено програмний засіб для автоматизації роботи в галузі пошукової оптимізації.

В ході програмної реалізації алгоритму збору та кластеризації семантичного ядра сайту було вирішено ряд проблем:

- збір максимально повної кількості ключових слів був проведений з допомогою сервісу API Google AdWords;
- кластеризація та видалення не тематичних запитів здійснювався алгоритмами ієрахічної кластеризації та K-means.
- Після отримання кластеризованого СЯ, були проведені наступні заходи для оптимізації сайту:
- редагування вмісту мета-тегів;
- розміщення ключових запитів в тегах <h1> - <h6>;
- заміна та використання ключових слів в тегах <alt> зображень;

– оптимізація url сторінок для кращого індексування пошуковими роботами;
– налаштування перенаправлень сторінок сайту (редирект 404).

Практичне значення отриманих результатів. Даний програмний додаток може бути впроваджений в системи адміністрування сайтів або в засоби підтримки роботи SEO-фахівців для підвищення повноти, точності і зниження часу розробки семантичного ядра сайтів.

Наукова новизна отриманих результатів полягає у створенні модифікованого алгоритму збору та кластеризації семантичного ядра сайту, за допомогою алгоритму пошуку популярних наборів в базі даних пошукових запитів та кластеризації їх методами інтелектуального аналізу даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Авдоніна Л І. Н., Гусєва Т. В. Письмові роботи наукового стилю. М.: «Інфра-М», 2012.-72 с.
2. Гричанник І. Р. Пошукова оптимізація сайту інтелектуальними засобами. Матеріали міжнародної наукової інтернет-конференції "Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення (випуск 34)" / Збірник тез доповідей: випуск 34 (м. Тернопіль, 11 грудня 2018 р.). – Частина 1. – Тернопіль. – 2018. – 16 - 17 с.
3. Яковлєв А.А. Раскрутка и продвижение сайтов. Основы, секреты, трюки. СПб: БХВ-Петербург, 2007. - 336 с.
4. Барсегян А.А. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP. Учебное пособие. / А.А. Барсегян, М.С. Куприянов, В.В. Степаненко, И.И. Холод. - М.:БХВ-Петербург, 2007. – 331 с.
5. Ашманов И., Иванов А. Оптимизация и продвижение сайтов в поисковых системах. СПб .: Питер, 2011. - 464 с.
6. Севостьянов І.О. Пошукова оптимізація. Практичний посібник з просування сайту в Інтернеті. СПб .: Пітер, 2010. - 240 с.
7. Алексунін В.А., Родігін В. В. Електронна комерція і маркетинг в Інтернеті. М .: Дашков і Ко, 2008. - 214 с.
8. Буреніна Т.А. Маркетинг на базі інтернет-технологій. М .: Благовіст-В, 2005. - 152 с.
9. Рассел Д., Кон Р. Латентно-семантичний аналіз. М .: VSD, 2013 - 100 с.
10. Євдокімов Н.В. Основи тематичної оптимізації. Ефективна інтернет-комерція і просування сайтів в Інтернет. Київ: Вільямс, 2007. - 160 с.

11. Кошик А. Веб-аналітика: аналіз інформації про відвідувачів веб-сайтів. Київ: Діалектика, 2009. - 464 с.
12. Кліфтон Б. Google Analytics: професійний аналіз відвідуваності веб-сайтів. М: Вільямс, 2009. - 400 с.
13. World Wide Web Consortium (W3C). - 2014. - Електронний ресурс Консорціуму Всесвітньої павутини. - (рос.). [Електронний ресурс]. Режим доступу: http: //w3c.org.ru/?Page_id= 44 (дата звернення 18.05.2017).
14. Нестеров С.А. Анализ статистики выполнения тестовых заданий в среде дистанционного обучения MOODLE. / С.А. Нестеров. - Современные информационные технологии и ИТ-образование. Т.12 (№ 4), 2016. ISSN 2411-1473. С.62-67.
15. Нестеров С.А. Базы данных. Интеллектуальный анализ данных. Учебное пособие. / С.А. Нестеров. - СПБ.: Изд-во Политехн. ун-та, 2011. – 272 с.
16. Станкевич Л.А. Интеллектуальные информационные и управляющие системы. Учебное пособие. / Л.А. Станкевич. - СПБ.: Изд-во Политехн. ун-та, 2011. – 201 с.
17. Educational Data Mining: введение [Електронний ресурс]. Режим доступу: https://habrahabr.ru/post/181053/ (дата звернення 05.04.2018).
18. Cristóbal Romero. Handbook of Educational Data Mining / Cristóbal Romero, Sebastian Ventura, Mykola Pechenizkiy - Taylor and Francis Group, 2011. – 502 с.
19. Основні поняття інтелектуального аналізу даних. [Електронний ресурс]. Режим доступу: https://msdn.microsoft.com/ru-ru/library/ms174949 (дата звернення 05.04.2018).
20. Обучение с учителем [Електронний ресурс]. Режим доступу: http://www.machinelearning.ru/wiki/index.php?title=%D0%9E%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81_%D1%83%D1%8

7%D0%B8%D1%82%D0%B5%D0%BB%D0%B5%D0%BC (дата звернення 17.06.2018).

21. Чубукова И.А. Data Mining. Задачи Data Mining. Классификация и кластеризация [Електронний ресурс]. Режим доступу: http://www.intuit.ru/studies/courses/6/6/lecture/166?page=4 (дата звернення 12.07.2018).
22. Чубукова И.А. Data Mining. Методы кластерного анализа. Иерархические методы [Електронний ресурс]. Режим доступу: http://www.intuit.ru/studies/courses/6/6/lecture/182?page=4 (дата звернення 12.07.2018).
23. Чубукова И.А. Data Mining. Методы кластерного анализа. Итеративные методы [Електронний ресурс]. Режим доступу: http://www.intuit.ru/studies/courses/6/6/lecture/184?page=1 (дата звернення 12.07.2018)
24. Деревья решений и алгоритмы их построения [Електронний ресурс]. Режим доступу: http://datareview.info/article/derevya-resheniy-i-algoritmyi-ih-postroeniya/ (дата звернення 14.08.2018).
25. Нейронные сети [Електронний ресурс]. Режим доступу: http://neuralnet.info/%D0%B3%D0%BB%D0%B0%D0%B2%D0%B02%D0%BE%D1%81%D0%BD%D0%BE%D0%B2%D1%8B%D0%B8%D0%BD%D1%81/ (дата звернення 14.08.2018).
26. Чубукова И.А. Data Mining. Методы поиска ассоциативных правил [Електронний ресурс]. Режим доступу: http://www.intuit.ru/studies/courses/6/6/lecture/186?page=4 (дата звернення 17.09.2018).
27. Структуры интеллектуального анализа данных (служби Analysis Services — інтелектуальний аналіз даних). [Електронний ресурс]. Режим доступу:

https://msdn.microsoft.com/ru-ru/library/ms174757.aspx (дата звернення 17.09.2018).

28. Типы содержимого (інтелектуальний аналіз даних). [Електронний ресурс]. Режим доступу: https://technet.microsoft.com/ru-ru/library/ms174572.aspx (дата звернення 17.09.2018).
29. CREATE MINING STRUCTURE (розширення інтелектуального аналізу даних). [Електронний ресурс]. Режим доступу: https://msdn.microsoft.com/ru-ru/library/ms131977.aspx (дата звернення 20.09.2018).
30. ALTER MINING MODEL (розширення інтелектуального аналізу даних). [Електронний ресурс]. Режим доступу: https://msdn.microsoft.com/ru-ru/library/ms132066.aspx (дата звернення 20.09.2018).
31. CREATE MINING MODEL (розширення інтелектуального аналізу даних). [Електронний ресурс]. Режим доступу: https://msdn.microsoft.com/ru-ru/library/ms131986.aspx (дата звернення 20.09.2018).
32. INSERT INTO (розширення інтелектуального аналізу даних). [Електронний ресурс]. Режим доступу: https://msdn.microsoft.com/ru-ru/library/ms132169(v=sql.110).aspx (дата звернення 21.09.2018).
33. Алгоритм кластеризации (Майкрософт). [Електронний ресурс]. Режим доступу: https://msdn.microsoft.com/ru-ru/library/ms174879.aspx (дата звернення 22.09.2018).
34. Образовательный портал Pythonic way [Електронний ресурс]. Режим доступу: http://pythonicway.com/ (дата обращения 22.09.2018).
35. Python. Основи і застосування [Електронний ресурс]. Режим доступу: //Stepic.org. URL: https://stepik.org/course/Python--основы-и-применение-512 (Дата звернення 22.09.2018)
36. Хайдуков Д. С.Застосування кластерного аналізу в державному управлінні // Філософія математики: актуальні проблеми. - М.: МАКС Пресс, 2009. - 143 с.

37. Шкляр М. Основи наукових досліджень. М.: «Дашков і Ко», 2012. - 206 с.
38. Bai Y., Jin Z. Prediction of SARS epidemic by BP neural networks with online prediction strategy / Chaos, Solitons and Fractals. - 2005. - Vol. 26, № 2. -P. 559-569.
39. Ward Hierarchical grouping to optimize an objective function / J. of the American Statistical Association, 1963. - 236 P.X.
40. Еникеев Е. С. Управление ИТ-проектами / Е. С. Еникеев / Computerworld. - 2002. - [Електронний ресурс]. Режим доступу: www.osp.ru/cw.
41. Кузнецов М.В. PHP и MySQL / М.В. Кузнецов, И.В. Симдянов, С.В. Голышев Санк-Петербург: БХВ-Петербург, 2007. - 450 с.
42. Матросов А.В. HTML 4 / А.В. Матросов, А.О. Сергеев, М.П. Чаунин – Санк-Петербург: БХВ-Петербург, 2007. - 672 с.
43. Курепин Р.Р. Самоучитель PHP 5: учебн. пособие / Р.Р. Курепин – Санк-Петербург: БХВ-Петербург, 2006. - 550 с.
44. Романовский В.А. Использование HTML 4: учебн. пособие / В.А. Романовский – Санк-Петербург: Питер, 2007. - 350 с.
45. Кузнецов М.В. Практика создания Web-сайтов: учебн. пособие / М.В. Кузнецов – Санк-Петербург: БХВ-Петербург, 2008. - 922 с.
46. Конверс Р.Р. PHP 5 и MySQL: учебн. пособие / Р.Р. Конверс–Санк-Петербург: БХВ-Петербург, 2006. - 223 с.
47. Гаевский А.Ю. Создание Web-страниц и Web-сайтов. HTML и JavaScript: самоучитель / А.Ю. Гаевский – М.: Триумф, 2008. - 464 с.
48. Bishop C. M. Pattern Recognition and Machine Learning. New York : Springer, 2006. xx, 738 p., ill.
49. Hua, X.-H. Clickage: towards bridging semantic and intent gaps via mining click logs of search engines / X.-S. Hua, L. Yang, J. Wang, J. Wang, M. Ye, K. Wang, Y. Rui, J. Li // ACM Multimedia. 2013. P. 243–252.

50. Huang, P.-S. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data / P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, L. Heck // 22nd ACM international conference on Conference on information & knowledge management : proceedings. 2013. P. 2333–2338.
51. Joachims, T. Optimizing Search Engines using Clickthrough Data / T. Joachims // ACM Conference on Knowledge Discovery and Data Mining : proceedings. 2002. P. 133-–142.
52. Lowe, D. G. Distinctive Image Features from Scale-Invariant / D. G. Lowe // International Journal of Computer Vision. 2004. Vol. 60, Iss. 2. P. 91–110.
53. G. E. Hinton // 27th International Conference on Machine Learning (ICML'10):proceedings. 2010. P. 807–814.
54. Браславский П., Соколов Е. Сравнение четырех методов автоматического извлечения двухсловных терминов из текста// Компьютерная лингвистика и интеллектуальные технологии: Труды Междунар. конф. Диалог'2006. - М.: Изд-во РГГУ, 2006. - С. 88-94
55. Методичні рекомендації до виконання магістерської роботи з освітнього ступеня "Магістр". Спеціальність: 123 - Комп'ютерна інженерія. Магістерська програма - Комп'ютерна інженерія. / О.М. Березький, Л.О. Дубчак, Г.М. Мельник /Під ред. О.М. Березького – Тернопіль: ТНЕУ, 2018.– 41 с.
56. Рішення РНБО «Про застосування персональних спеціальних економічних та інших обмежувальних заходів (санкцій)», які передбачають блокування mail.ru, «Яндекса», соцмереж «ВКонтакте» та «Однокласники» від 28 квітня 2017 року.