Хазова Анастасия, МТР152

База данных lyrics.net: статистическая обработка и представление информации.

Readme.md
Все данные находятся по адресу: https://github.com/haniani/DH

База данных была взята с сайта https://toster.ru/q/45980, объем более 700 мб, поэтому скачивание по приведенной ссылке. В базу входит 56 198 исполнителей, 113 151 альбомов и 372 357 песен.

###

Поскольку база данных объемная, для обработки был установлен лимит в 1000 исполнителей, но значение может изменяться вручную.

В репозитории:
csvdict1.csv - словарь вида автор/альбом/песня, выкачанный из базы данных. Каждой песне соответствует запись об альбоме и исполнителе. В настоящее время закомментирован и выложен в полном объеме в репозиторий.
slovnik.csv - частотный словарь вида слово/частота. В порядке убывания представлены все слова, встретившиеся в названиях песен исполнителей (лимит - 1000 человек).
main.py - скрипт, позволяющий получить словари и различную статистическую информацию, а именно:

1. Корреляция Пирсона
2. Визуализация соотношения длины названий песен и альбомов
3. LDA
4. LSI
5. Частотный словарь
6. Словарь вида автор/альбом/песня

Используемый язык: python 3.4
Используемые модули:

```
import pymysql as db
import csv, os, sys
import pylab
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats
from nltk.tokenize.simple import SpaceTokenizer
from nltk.stem.porter import PorterStemmer
import gensim
from gensim import corpora, models, similarities
from stop_words import get_stop_words
```

Первый этап работы программы - подключение и создание базы данных. После чего необходимо загрузить дамп базы через консоль посредством следующей команды:

mysql --user=root --password=12345 music < /путь'/database.sql

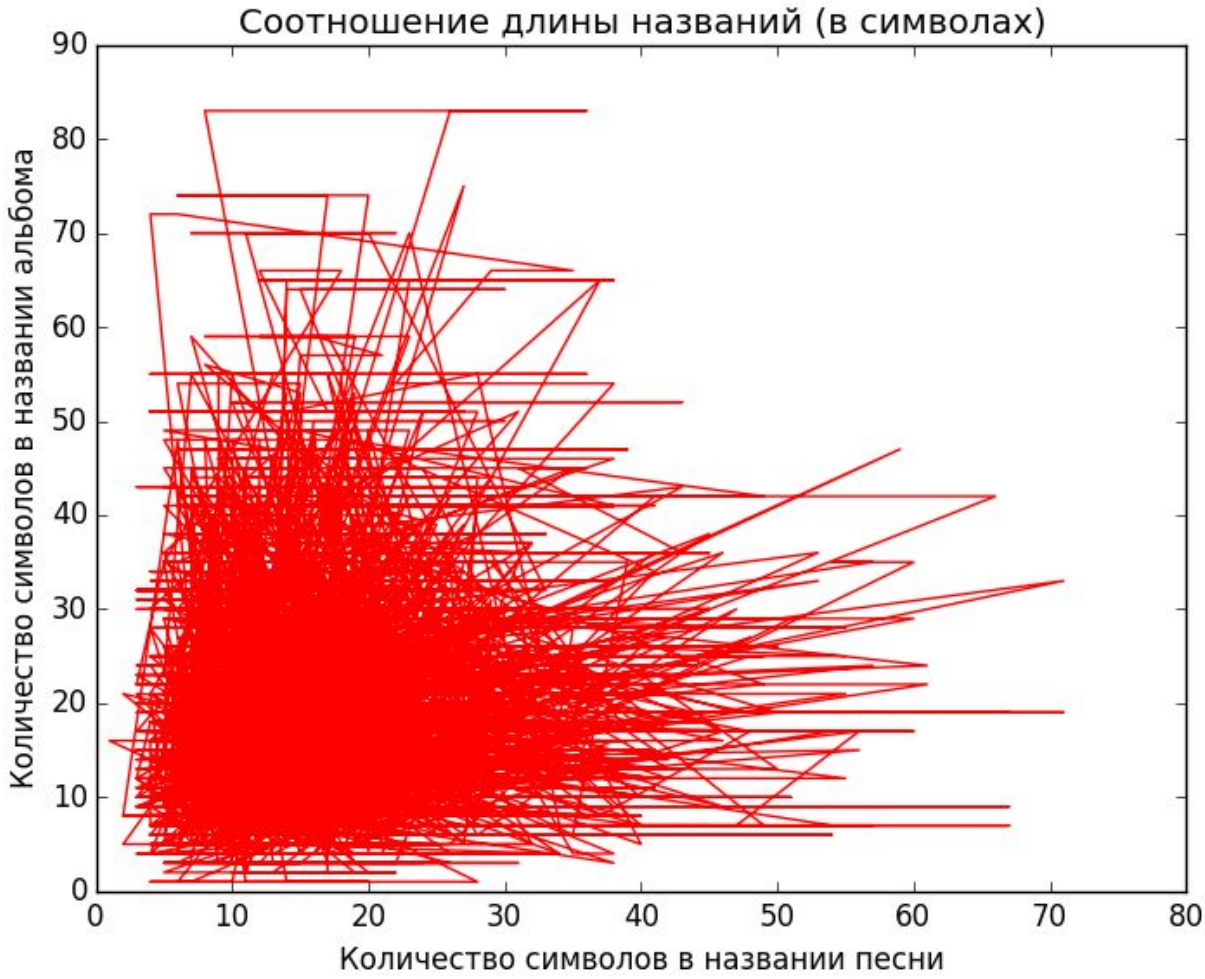
После чего идет выборка по следующим критериям:

1. Артист (лимит в 1000 исполнителей).
2. Альбомы, принадлежащие артисту.
3. Песни, соответствующие выбранным ранее параметрам.

Модулями pylab и matplotlib.pyplot строим графики корреляции длин названий альбомов и песен.

```
correl = np.corrcoef(album, song)
```

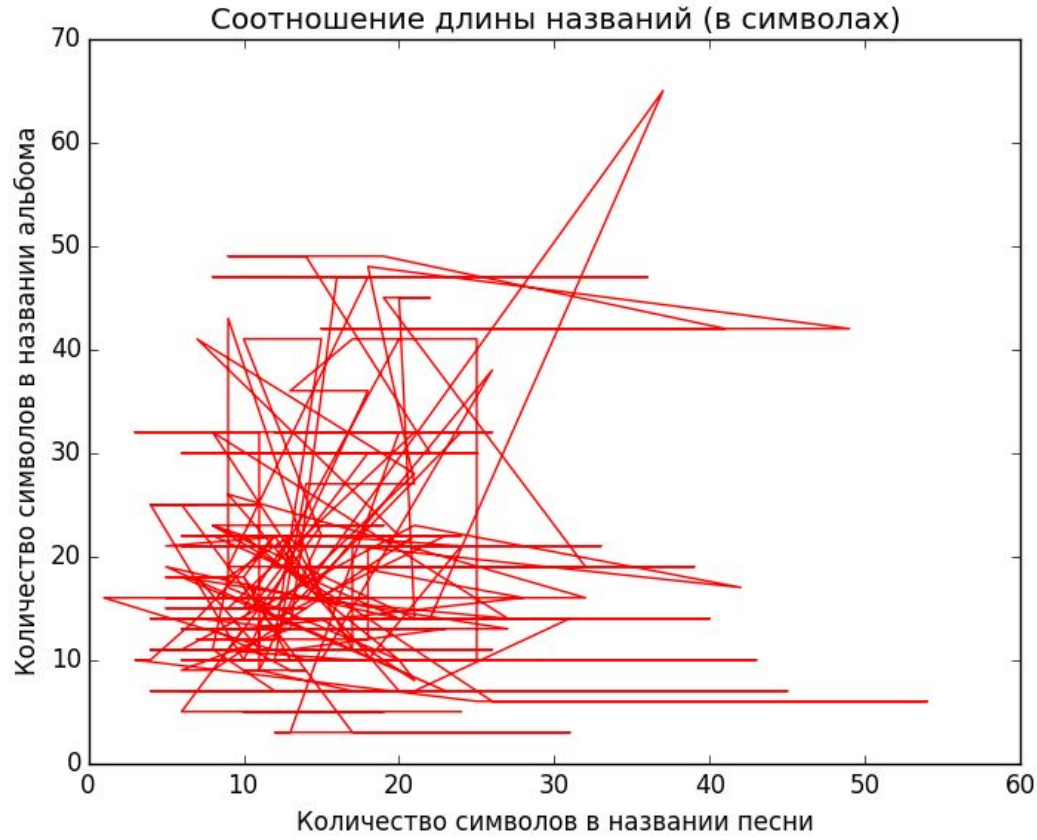
Корреляция Пирсона (0.098674182429171989, 4.0277823168197925e-14)

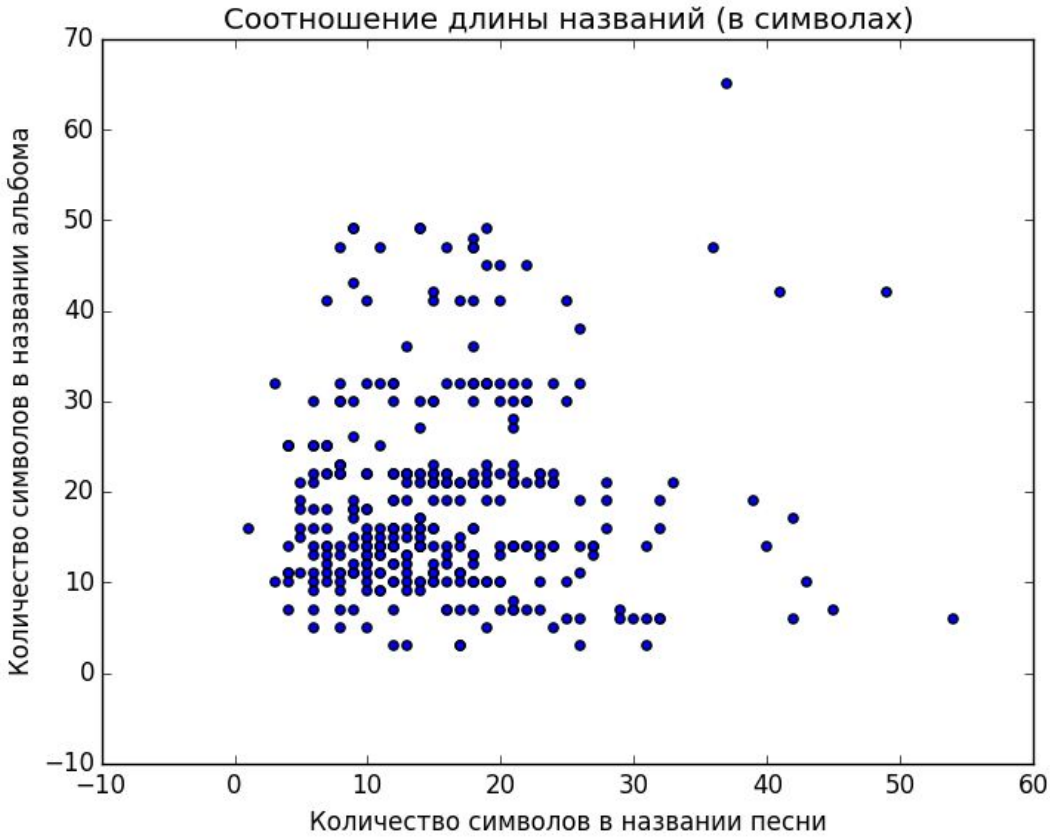

Соотношение длины названий (в символах)
Количество символов в названии альбома
Количество символов в названии песни
0
10
20
30
40
50
60
70
80
90

По оси OX обоих графиков откладываются точки, равные количеству символов в названиях альбомов из ранее сделанной выборки, по оси OY - количество символов в названиях песен. Графики позволяют увидеть не только распределение длины текстов (концентрируется в пределах 5-25 для песен и 5-30 для альбомов), но и наличие линейной зависимости между рассматриваемыми величинами. Высокая степень корреляции обусловлена характером выборки - была выбрана первая тысяча исполнителей, а поскольку каждый из них имеет по несколько альбомов, то это искусственно повышает коэффициент корреляции.

Рассмотрим две других выборки:

выборка 1 (параметры - лимит 1000 идентификаторов(исполнителей), год - 2000):

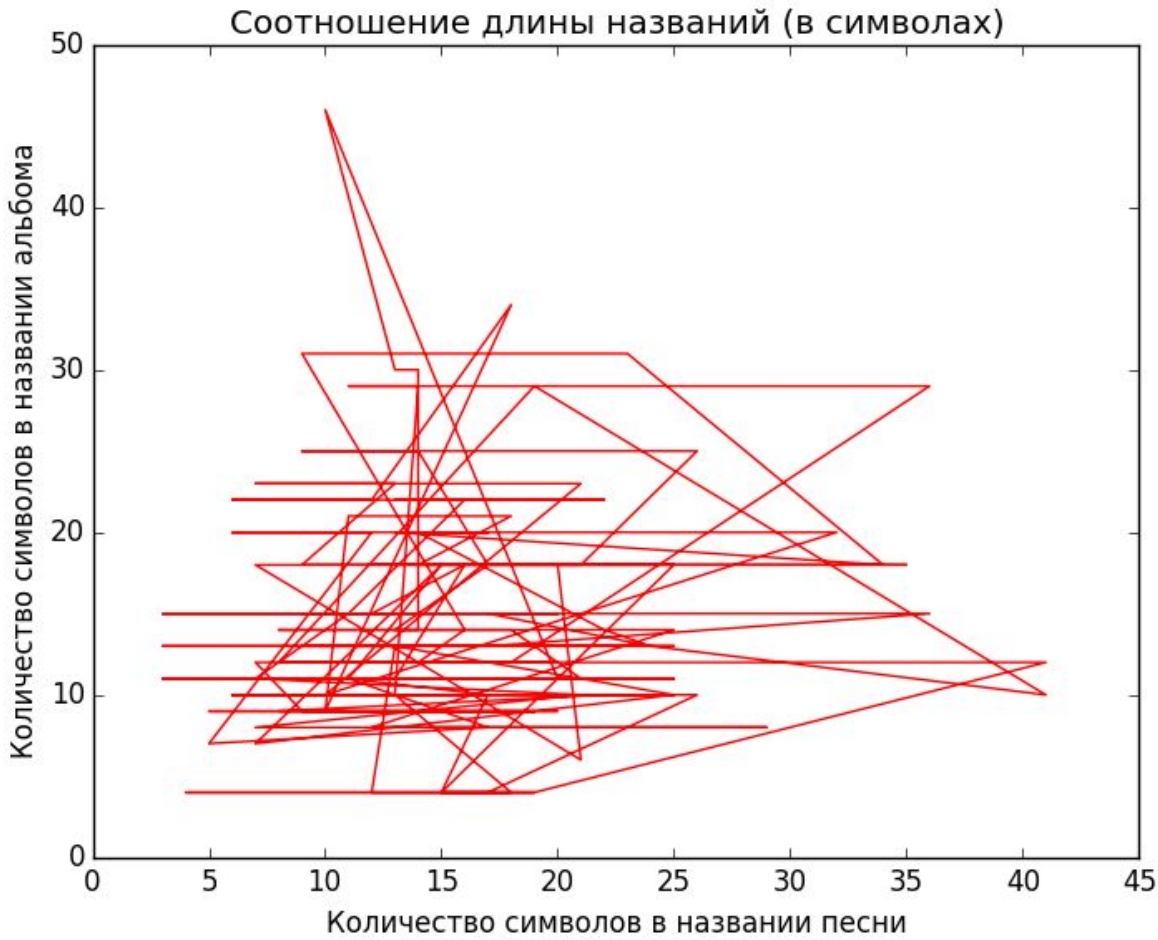

Соотношение длины названий (в символах)
Количество символов в названии альбома
Количество символов в названии песни
0
10
20
30
40
50
60
70

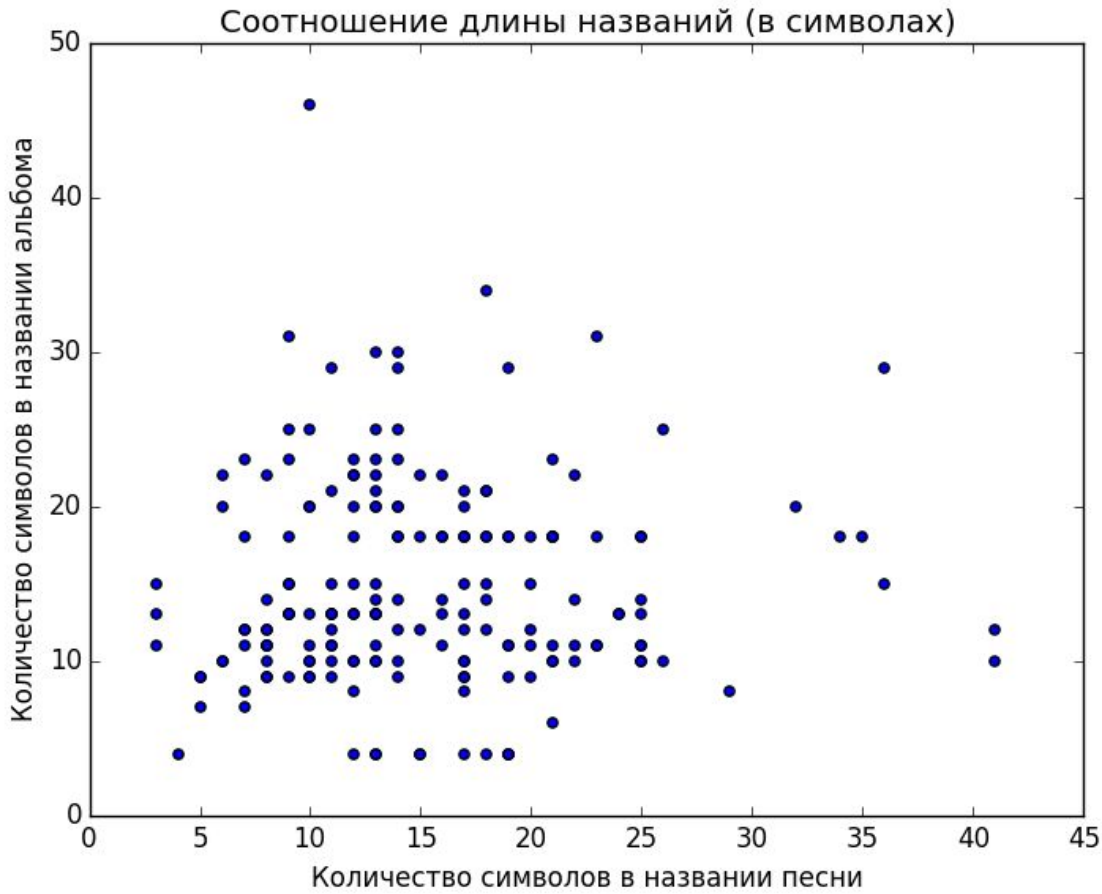

Соотношение длины названий (в символах)
Количество символов в названии альбома
Количество символов в названии песни
−10
0
10
20
30
40
50
60
70

Корреляция Пирсона (0.040526382504812025, 0.4597334371306383),

снизилась, поскольку в первом случае в корреляцию входили альбомы и песни в основном одних и тех же исполнителей, а в данной выборке эти показатели разные.

Для надежности проверим вторую выборку(параметры - лимит 1000 идентификаторов(исполнителей), год - 1995):





Корреляция Пирсона (0.042955579330537778, 0.56153225795336037)

Действительно, с введением параметра выборки по году коэффициент корреляции снижается по сравнению с первой выборкой, но остается близким ко второй.

LDA (Латентное размещение Дирихле), применяющееся после предварительной обработки названий песен (токенизация, применение стоп-слов, стемминг) позволяет выделить темы, преобладающие в выборке. Для первой 1000 исполнителей результаты оказались следующими:

Результат работы LDA1:

[(0, '0.016*girl + 0.016*babi + 0.015*night'), (1, '0.029*get + 0.028*come + 0.016*[*]'), (2, '0.022*heart + 0.015*away + 0.014*just'), (3, '0.039*one + 0.025*world + 0.021*take'), (4, '0.115*love + 0.025*know + 0.018*day'), (5, '0.031*go + 0.028*life + 0.028*like'), (6, '0.024*way + 0.022*never + 0.017*want')]

Показатели позволяют судить о лирической направленности анализируемых микро-текстов.

Рассмотрим вторую и третью выборки:

Результат работы LDA2:

[(0, '0.072*[live] + 0.014*never + 0.014*one'), (1, '0.020*way + 0.014*everi + 0.014*god'), (2, '0.044*love + 0.038*take + 0.013*need'), (3, '0.037*make + 0.022*know + 0.022*peopl'), (4, '0.044*[singl + 0.043*version] + 0.019*heart'), (5, '0.029*like + 0.020*[dvd] + 0.020*[*]'), (6, '0.030*life + 0.018*get + 0.018*can')]

Результат работы LDA3:

[(0, '0.035*eye + 0.024*cri + 0.013*young'), (1, '0.036*life + 0.025*dream + 0.025*live'), (2, "0.015*talk + 0.015*(it' + 0.015*2"), (3, '0.119*love + 0.026*sophist + 0.026*ladi'), (4, '0.034*angel + 0.023*fuck + 0.023*girl'), (5, '0.022*heart + 0.022*diga + 0.022*kind'), (6, '0.037*die + 0.025*go + 0.025*sun')]

Данные результаты демонстрируют, что названия песен в 2000 году по сравнению с 1995 стали более позитивно ориентированными (ряды “live, never, one; love, take, need” по сравнению с рядами “die, go, sun; eye, cry, young”).

Следующая используемая нами метрика - LSI (Латентно-семантический анализ), используемая для кластеризации. Посредством анализа данных статистически выбирает наиболее значимые слова, а затем группирует в сравнительно однородные группы (в данном случае тематика песен прослеживается так же отчетливо. Количество кластеров было установлено на значение 3 и меняется вручную.

Результаты ее применения таковы:

Результат работы LSI1:

[(0, '-1.000*"love" + 0.000*"wonder" + -0.000*"move" + 0.000*"hit" + 0.000*"street" + 0.000*"us" + 0.000*"ladi" + -0.000*"rock" + -0.000*"road" + 0.000*"eye"'), (1, '1.000*"one" + 0.006*"move" + 0.005*"wanna" + -0.004*"enough" + -0.004*"now" + -0.004*"us" + -0.004*"crazi" + -0.004*"track]" + 0.004*"believ" + -0.004*"name"'), (2, '-0.999*"go" +

0.007*"de" + -0.006*"name" + 0.006*"end" + -0.005*"track]" + 0.005*"wanna" + -0.005*"beauti" + -0.005*"will" + -0.005*"mix]" + 0.005*"noth"')]

Результат работы LSI2:
[(0, '-1.000*"[live]" + 0.001*"wish" + -0.001*"wobbl" + -0.001*"sun" + 0.001*"loser" + -0.001*"rose" + 0.001*"gone" + -0.001*"lost" + -0.001*"alon" + -0.000*"torn"'), (1, '1.000*"[dvd]" + -0.004*"wish" + -0.003*"dream" + -0.003*"smile" + 0.003*"stranger" + -0.003*"best" + -0.002*"eye" + -0.002*"use" + 0.002*"noth" + 0.002*"gonna"'), (2, '1.000*"version]" + -0.006*"lane" + 0.003*"clogger" + -0.003*"rose" + 0.003*"black" + 0.003*"wish" + 0.003*"+" + -0.002*"beauti" + 0.002*"mouth" + 0.002*"dream"')]

Результат работы LSI3:
[(0, '1.000*"love" + 0.000*"pagan" + 0.000*"paradis" + -0.000*"onion" + 0.000*"ship" + -0.000*"(thi" + -0.000*"wood" + 0.000*"dont" + -0.000*"song" + -0.000*"bird"'), (1, '1.000*"go" + 0.000*"ladi" + 0.000*"die" + 0.000*"life" + 0.000*"night" + 0.000*"eye" + 0.000*"sophist" + -0.000*"sun" + -0.000*"talk" + 0.000*"pagan"'), (2, '-0.541*"life" + -0.442*"eye" + -0.394*"dream" + 0.365*"sophist" + -0.309*"cri" + 0.275*"angel" + 0.226*"die" + 0.034*"ladi" + 0.000*"let" + 0.000*"night"')]

Со сменой входных данных изменились сами тематические показатели на основе текстов, но сама тематика не поменялась и наблюдается та же положительная динамика в отношении экспрессивности и позитивной ориентированности.

Недостатком использования последних двух метрик является значительное снижение скорости обработки при увеличении объема входных данных.

Формирование частотного словаря происходит посредством обхода сформированного на основе базы словаря, содержащего информацию обо всех песнях в базе данных.

В настоящий момент словарь является полным, но при необходимости возможно исключение местоимений, служебных частиц и стоп-слов.

Номер места	Слово	Частота
1	the	52586
2	you	30310
3	i	23060
4	of	21169
5	a	21053
6	love	19111
7	me	18599
8	to	18229
9	in	16940
10	my	15067

При рассмотрении топ-10 слов из словаря можно наблюдать, что служебные слова и местоимения являются наиболее употребительными в микро-текстах проанализированных нами названиях. Но помимо них в топе присутствует и слово love.

Если мы применим список стоп-слов, то выдача значительно изменится. Обратим внимание на топ-10:

Номер места	Слово	Частота
1	love	19111
2	time	4430
3	song	4275
4	one	4243
5	little	4031
6	heart	3894

7	night	3688
8	go	3602
9	day	3587
10	like	3564

Таким образом, для первой таблицы мы видим, что в топ-10 превалируют стоп-слова, но если рассматривать топ-100, то ближе к концу сотни появляется достаточное количество обычных слов.
При введении списка таблица демонстрирует, что слово love стало вершиной списка. Остальные слова, как и первое, подтверждают нашу гипотезу о тематической направленности полученных данных.