**Ход работы**

*Статистические характеристики исходной коллекции:*

1) число кластеров: 1 кластер, состоящий из 2-ух вычислительных узлов и одного главного узла;

2) число элементов данных: 16597;

3) объем данных: 996 Кбайт, 16597 записей;

4) время заполнения коллекции данными: 6,5 с.

*Инвертированный индекс:*

1) Среднее время построения индекса: 405 с.

Таблица 3 – Время построения инвертированного индекса

| Попытка | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Время, с | 393 | 415 | 399 | 415 | 436 | 391 | 404 | 399 | 403 | 398 |

2) Размер индекса:

- в байтах: 800 Кбайт;
- количество записей: 20274.

*K-граммный индекс:*

1) Среднее время построения индекса: 162 с.

Таблица 4 – Время построения k-граммного индекса

| Попытка | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Время, с | 160 | 168 | 159 | 162 | 164 | 162 | 161 | 169 | 160 | 155 |

2) Размер индекса:

- в байтах: 1,60 Мбайт;
- количество записей: 11020.

*Статистические характеристики выполнения операции поиска*

Таблица 5 – Временные характеристики выполнения запросов

| № | Запрос | Время выполнения, с | | | | | | | | Среднее время выполнения, с | Наиболее медленные операции и время их выполнения |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | *bor | 4,3 | 3,35 | 2,9 | 2,89 | 2,97 | 2,98 | 3,54 | 3,35 | 3,28 | Поиск в инвертированном индексе (~2,57 с) |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | *ntilyclad | 2,76 | 3,33 | 3,17 | 2,67 | 2,52 | 2,66 | 2,74 | 2,58 | 2,8 | Поиск в k-граммном индексе (~2,37 с) |
| 3 | *ari | 31,74 | 31,56 | 32,08 | 30,82 | 29,82 | 31,64 | 29,29 | 29,98 | 30,86 | Поиск в инвертированном индексе (~30,3 с) |
| 4 | *ting | 1,7 | 1,52 | 1,61 | 1,62 | 1,65 | 1,55 | 1,59 | 1,61 | 1,6 | Поиск в инвертированном индексе (~0,97 с) |
| 5 | *love | 3,5 | 5,1 | 5,9 | 4,1 | 4,9 | 3,4 | 4,3 | 4,9 | 4,5 | Поиск в инвертированном индексе (~3,5 с) |
| 6 | *order | 2,7 | 2,4 | 2,7 | 2,5 | 3,4 | 3,6 | 3,5 | 2,8 | 2,9 | Поиск в инвертированном индексе (~1,4 с) |
| 7 | *mint | 2,2 | 1,9 | 2,1 | 1,8 | 2,6 | 2,3 | 2,3 | 2,7 | 2,2 | Поиск в инвертированном индексе (~1,6 с) |
| 8 | *ove | 3,4 | 3,6 | 4,6 | 4,1 | 3,6 | 4,5 | 3,6 | 4 | 3,4 | Поиск в инвертированном индексе (~2,9 с) |
| 9 | *ful | 1,4 | 1,2 | 2 | 1,5 | 1,4 | 1,3 | 1,9 | 1,3 | 1,5 | Поиск в инвертированном индексе (~0,7 с) |
| 10 | *use | 2 | 2,3 | 3,2 | 2,5 | 2,5 | 3,3 | 2,3 | 2,9 | 2,6 | Поиск в инвертированном индексе (~1,9 с) |

*Алгоритм построения индекса типа (1)*

При изменении коллекции (добавлении, удалении или изменении документа) происходит перестройка индексов, сначала инвертированного, затем k-граммного:

1) При добавлении документа: считывается содержимое указанного файла и происходит добавление документа в конец коллекции, после чего происходит вызов функции построения сначала инвертированного индекса, а на его основе k-граммного.

2) При удалении документа: пользователем указывается номер документа и происходит его удаление из коллекции, после чего происходит вызов функции построения сначала инвертированного индекса, а на его основе k-граммного.

3) При изменении документа: происходит изменение содержимого того документа коллекции, номер которого указан пользователем, после чего происходит вызов функции построения сначала инвертированного индекса, а на его основе k-граммного.

Таблица 1 – Замеры времени для (1) типа динамического индекса

| № | Время, с | | |
|---|---|---|---|
| | Удаление документа | Вставка документа | Изменение документа |
| 1 | 68 | 145 | 74 |
| 2 | 81 | 109 | 74 |
| 3 | 72 | 201 | 88 |
| 4 | 76 | 73 | 79 |
| 5 | 75 | 86 | 71 |
| 6 | 79 | 74 | 81 |
| 7 | 68 | 95 | 89 |
| 8 | 91 | 77 | 75 |
| 9 | 77 | 91 | 77 |
| 10 | 74 | 114 | 75 |
| 11 | 75 | 79 | 82 |
| 12 | 80 | 84 | 92 |
| 13 | 69 | 81 | 85 |
| 14 | 71 | 88 | 93 |
| 15 | 82 | 76 | 76 |
| 16 | 68 | 93 | 91 |
| 17 | 66 | 79 | 89 |
| 18 | 75 | 76 | 79 |
| 19 | 74 | 90 | 82 |
| 20 | 81 | 81 | 89 |
| Среднее время, с | 75,1 | 93,95 | 82,05 |

Таблица 2 – Замеры времени при удалении документа

| № \ Размер коллекции | Время, с | | | | | |
|---|---|---|---|---|---|---|
| | 500 | 1000 | 2000 | 4000 | 8000 | 16000 |
| 1 | 2,8 | 4,31 | 6,13 | 27,72 | 37,88 | 79,18 |
| 2 | 1,54 | 5,62 | 5,71 | 21,53 | 31,08 | 88,62 |
| 3 | 1,32 | 4,52 | 8,92 | 20,59 | 34,78 | 71,62 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 4 | 1,29 | 5,96 | 7,36 | 22,63 | 32,33 | 89,99 |
| 5 | 1,57 | 4,47 | 7,33 | 20,19 | 36,45 | 92,08 |
| 6 | 1,21 | 4,77 | 7,12 | 21,43 | 36,11 | 91,6 |
| 7 | 1,29 | 5,02 | 6,88 | 22,4 | 35,68 | 73,7 |
| 8 | 1,98 | 5,41 | 6,9 | 21,56 | 31,8 | 75,22 |
| 9 | 1,09 | 5,1 | 7,35 | 20,99 | 33,54 | 81,42 |
| 10 | 1,76 | 4,91 | 7,7 | 20,6 | 35,2 | 74,6 |
| Среднее время, с | 1,585 | 5,009 | 7,14 | 21,964 | 34,485 | 81,8 |

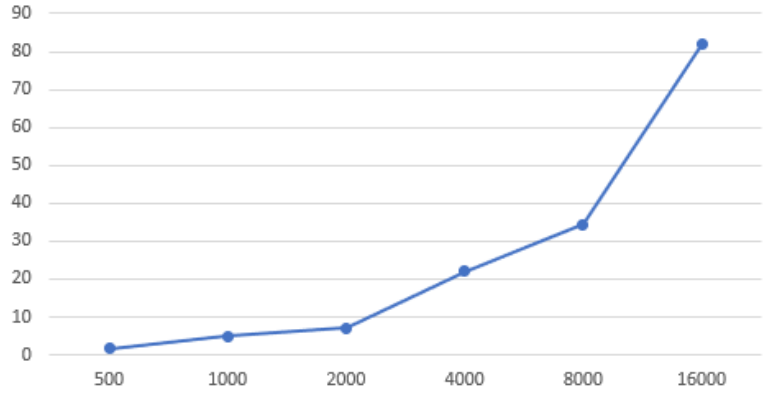


Рисунок 1 – Зависимость среднего времени удаления от размера коллекции

Таблица 3 – Замеры времени при вставке документа

| № \ Размер коллекции | Время, с | | | | | |
|---|---|---|---|---|---|---|
| | 500 | 1000 | 2000 | 4000 | 8000 | 16000 |
| 1 | 2,1 | 4,38 | 7,8 | 21,07 | 32,02 | 80,74 |
| 2 | 1,51 | 4,41 | 6,3 | 18,2 | 37,15 | 86,17 |
| 3 | 1,11 | 4,65 | 8,8 | 24,34 | 31,69 | 92,49 |
| 4 | 1,65 | 5,27 | 8,4 | 18,99 | 32,14 | 89,39 |
| 5 | 1,25 | 4,38 | 9,8 | 29,34 | 39,25 | 88,4 |
| 6 | 1,21 | 5,15 | 8,3 | 22,52 | 39,7 | 91,71 |
| 7 | 1,94 | 4,7 | 7,9 | 21,73 | 35,67 | 81,2 |
| 8 | 1,15 | 4,55 | 8,12 | 20,87 | 35,02 | 85,9 |
| 9 | 1,76 | 4,98 | 8,76 | 21,44 | 32,6 | 74,69 |
| 10 | 1,5 | 4,45 | 9,01 | 22,08 | 34,78 | 79,26 |
| Среднее время, с | 1,518 | 4,692 | 8,319 | 22,058 | 35,002 | 84,995 |

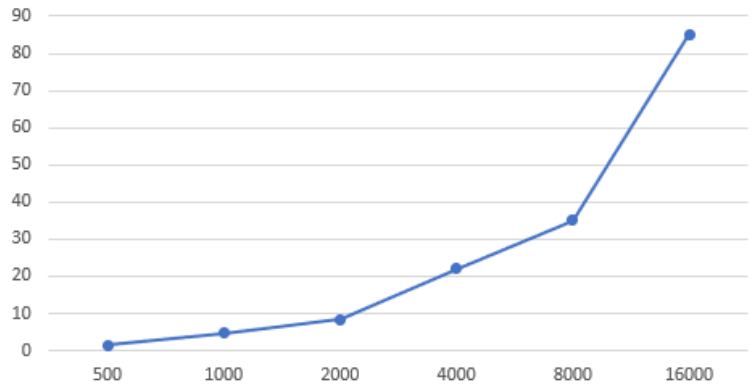


Рисунок 2 – Зависимость среднего времени вставки от размера коллекции

Таблица 4 – Замеры времени при изменении документа

| № \ Размер коллекции | Время, с | | | | | |
|---|---|---|---|---|---|---|
| | 500 | 1000 | 2000 | 4000 | 8000 | 16000 |
| 1 | 0,2158 | 1,11 | 6,14 | 17,01 | 39,84 | 131,08 |
| 2 | 0,4224 | 1,1 | 5,3 | 19,84 | 36,34 | 89,69 |
| 3 | 0,2227 | 1,68 | 7,97 | 18,98 | 36,31 | 120,07 |
| 4 | 0,3468 | 1,97 | 6,49 | 18,5 | 35,34 | 82,95 |
| 5 | 0,3276 | 1,56 | 10,86 | 19,1 | 33,86 | 91,2 |
| 6 | 0,2856 | 1,66 | 8,8 | 19,55 | 35,7 | 89,31 |
| 7 | 0,3177 | 1,9 | 6,92 | 19,2 | 35,91 | 96,7 |
| 8 | 0,3045 | 1,45 | 8,71 | 18,42 | 36,5 | 91,52 |
| 9 | 0,298 | 1,12 | 8,09 | 18,94 | 36,06 | 84,58 |
| 10 | 0,2556 | 1,82 | 7,28 | 18,05 | 37,2 | 99,54 |
| Среднее время, с | 0,2996 | 1,537 | 7,656 | 18,756 | 36,306 | 97,664 |

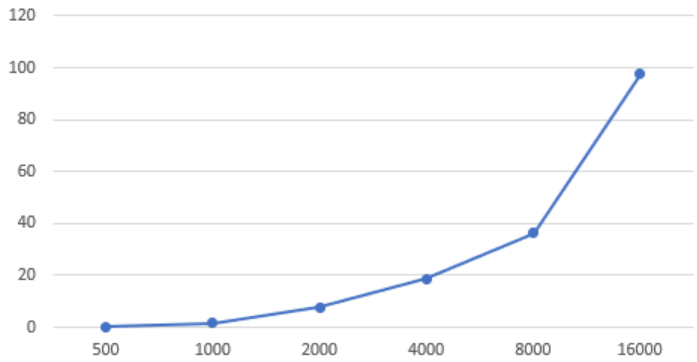


Рисунок 3 – Зависимость среднего времени изменения от размера коллекции

*Алгоритм построения индекса типа (2):*

1) При добавлении документа: на вход подается файл с содержанием нового документа. Добавляем новый документ в файл с новыми данными и присваиваем ему порядковый номер, перестраиваем вспомогательные индексы: инвертированный и k-граммный, на основе новых данных. Добавляем в вектор документов еще одну позицию со значением 1.

2) При удалении документа: на вход подается номер документа, который нужно удалить. Для этого помечаем в векторе, длина которого равна размеру коллекции, данный документ, как недействительный: в позицию, равной номеру документа, ставим 0 (значит недействительный).

3) При изменении документа: на вход подается файл с содержанием измененного документа и номер документа, который будет изменен. Сначала происходит его удаление согласно шагу 2. Затем измененный документ записывается в файл с новыми данными, ему присваивается порядковый номер, происходит перестройка вспомогательных индексов: инвертированного и k-граммного на основе новых данных. Добавляем в вектор документов еще одну позицию со значением 1.

Таблица 5 – Замеры времени для (2) типа динамического индекса

| № | Время, с | | |
|---|---|---|---|
| | Удаление документа | Вставка документа | Изменение документа |
| 1 | 0,062 | 0,084 | 0,093 |
| 2 | 0,092 | 0,125 | 0,078 |
| 3 | 0,062 | 0,112 | 0,132 |
| 4 | 0,062 | 0,08 | 0,093 |
| 5 | 0,078 | 0,092 | 0,093 |
| 6 | 0,081 | 0,109 | 0,078 |
| 7 | 0,062 | 0,078 | 0,132 |
| 8 | 0,093 | 0,078 | 0,078 |
| 9 | 0,062 | 0,093 | 0,132 |
| 10 | 0,091 | 0,078 | 0,107 |
| 11 | 0,062 | 0,093 | 0,078 |
| 12 | 0,084 | 0,083 | 0,091 |
| 13 | 0,062 | 0,092 | 0,08 |
| 14 | 0,094 | 0,078 | 0,12 |
| 15 | 0,063 | 0,083 | 0,156 |
| 16 | 0,101 | 0,092 | 0,093 |

| | | | |
|---|---|---|---|
| 17 | 0,071 | 0,104 | 0,082 |
| 18 | 0,062 | 0,078 | 0,089 |
| 19 | 0,091 | 0,094 | 0,093 |
| 20 | 0,064 | 0,082 | 0,092 |
| Среднее время, с | 0,074 | 0,09 | 0,099 |

Таблица 6 – Замеры времени при удалении документа

| № \ Размер коллекции | Время, с | | | | | |
|---|---|---|---|---|---|---|
| | 500 | 1000 | 2000 | 4000 | 8000 | 16000 |
| 1 | 0,02 | 0,044 | 0,027 | 0,031 | 0,046 | 0,078 |
| 2 | 0,047 | 0,04 | 0,029 | 0,04 | 0,031 | 0,078 |
| 3 | 0,015 | 0,015 | 0,055 | 0,031 | 0,062 | 0,031 |
| 4 | 0,015 | 0,031 | 0,032 | 0,062 | 0,046 | 0,078 |
| 5 | 0,024 | 0,044 | 0,036 | 0,049 | 0,046 | 0,046 |
| 6 | 0,034 | 0,021 | 0,049 | 0,048 | 0,125 | 0,125 |
| 7 | 0,049 | 0,062 | 0,028 | 0,046 | 0,062 | 0,078 |
| 8 | 0,03 | 0,031 | 0,046 | 0,031 | 0,072 | 0,046 |
| 9 | 0,062 | 0,044 | 0,031 | 0,05 | 0,015 | 0,125 |
| 10 | 0,044 | 0,031 | 0,08 | 0,032 | 0,046 | 0,078 |
| Среднее время, с | 0,034 | 0,036 | 0,041 | 0,042 | 0,055 | 0,076 |

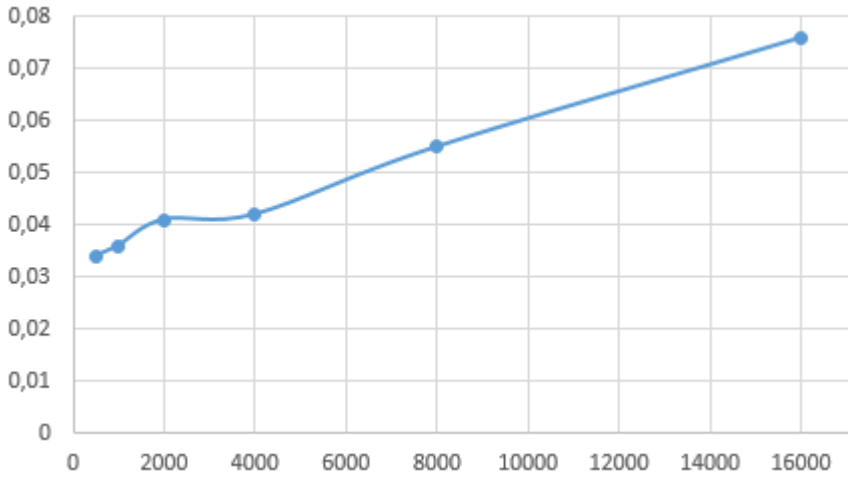


Рисунок 4 – Зависимость среднего времени удаления от размера коллекции

Таблица 7 – Замеры времени при вставке документа

| № \ Размер коллекции | Время, с | | | | | |
|---|---|---|---|---|---|---|
| | 500 | 1000 | 2000 | 4000 | 8000 | 16000 |
| 1 | 0,043 | 0,015 | 0,05 | 0,078 | 0,078 | 0,078 |
| 2 | 0,046 | 0,062 | 0,022 | 0,093 | 0,062 | 0,046 |
| 3 | 0,062 | 0,046 | 0,046 | 0,062 | 0,062 | 0,078 |
| 4 | 0,027 | 0,031 | 0,015 | 0,062 | 0,062 | 0,052 |
| 5 | 0,062 | 0,041 | 0,062 | 0,031 | 0,046 | 0,078 |
| 6 | 0,031 | 0,062 | 0,046 | 0,109 | 0,062 | 0,078 |
| 7 | 0,046 | 0,031 | 0,062 | 0,062 | 0,062 | 0,078 |
| 8 | 0,062 | 0,062 | 0,062 | 0,062 | 0,062 | 0,05 |
| 9 | 0,015 | 0,031 | 0,062 | 0,062 | 0,062 | 0,062 |
| 10 | 0,062 | 0,031 | 0,109 | 0,062 | 0,059 | 0,078 |
| Среднее время, с | 0,045 | 0,041 | 0,053 | 0,068 | 0,061 | 0,067 |

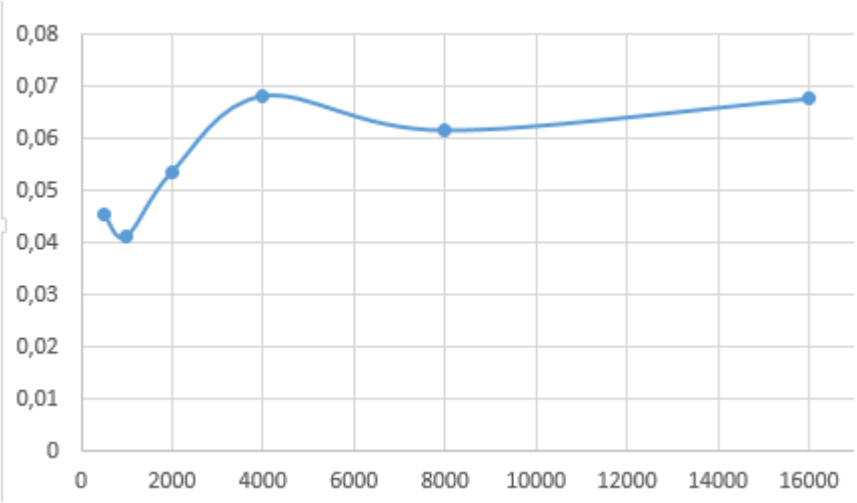


Рисунок 5 – Зависимость среднего времени вставки от размера коллекции

Таблица 8 – Замеры времени при изменении документа

| № \ Размер коллекции | Время, с | | | | | |
|---|---|---|---|---|---|---|
| | 500 | 1000 | 2000 | 4000 | 8000 | 16000 |
| 1 | 0,046 | 0,031 | 0,046 | 0,046 | 0,039 | 0,046 |
| 2 | 0,031 | 0,031 | 0,031 | 0,046 | 0,046 | 0,062 |
| 3 | 0,031 | 0,015 | 0,046 | 0,046 | 0,046 | 0,046 |
| 4 | 0,046 | 0,046 | 0,046 | 0,047 | 0,031 | 0,05 |
| 5 | 0,022 | 0,046 | 0,046 | 0,046 | 0,03 | 0,046 |
| 6 | 0,046 | 0,046 | 0,017 | 0,042 | 0,046 | 0,056 |
| 7 | 0,046 | 0,046 | 0,046 | 0,031 | 0,031 | 0,046 |
| 8 | 0,046 | 0,046 | 0,03 | 0,031 | 0,047 | 0,046 |
| 9 | 0,046 | 0,046 | 0,015 | 0,046 | 0,062 | 0,046 |
| 10 | 0,031 | 0,031 | 0,031 | 0,046 | 0,031 | 0,05 |
| Среднее время, с | 0,039 | 0,038 | 0,035 | 0,042 | 0,040 | 0,049 |

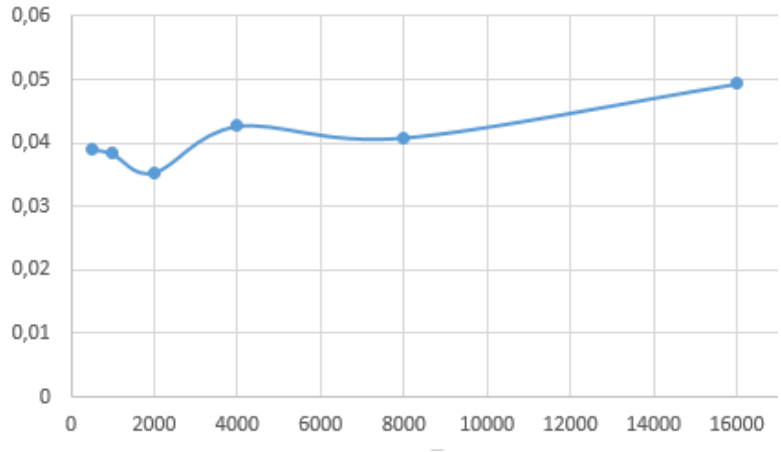


Рисунок 6 – Зависимость среднего времени изменения от размера коллекции

***Оценка качества полученной системы***

Были размечены 350 документов коллекции, указано для каких тем они подходят. Для оценки качества полученной поисковой системы было решено использовать следующие информационные потребности и соответствующие запросы к ним:

- найти все твиты, в которых есть упоминания о ночном времени суток и все, что с ним связано: *night; подходящих документов – 215;

```
Query: *night
3_gram of query: ['nig', 'igh', 'ght', 'ht$']
Count: 12
Result terms for query: [u'midknight', u'datenight', u'knight', u'midnight', u'overnight', u'tonight', u'
goodnight', u'bringonthenight', u'night', u'deathknight', u'gnight', u'2night']
```

Рисунок 7 – Результаты запроса *night

- найти все твиты, где есть упоминания о магазинах: *shop; подходящих документов – 36;

```
Query: *shop
3_gram of query: ['sho', 'hop', 'op$']
Count: 6
Result terms for query: [u'shop', u'photoshop', u'workshop', u'barbershop', u'matcheshop', u'bishop']
```

Рисунок 8 – Результаты запроса * shop

- найти все твиты, в которых упоминаются какие-либо планы: *plan; подходящих документов – 55;

```
Query: *plan
3_gram of query: ['pla', 'lan', 'an$']
Count: 6
Result terms for query: [u'explan', u'floorplan', u'financialplan', u'retirementplan', u'plan',
 u'airplan']
```

Рисунок 9 – Результаты запроса * plan

*a. Точность и полнота*

Чаще всего для оценки информационного поиска используются два показателя: точность (precision) и полнота (recall). Они определены для простого случая, когда информационно-поисковая система возвращает набор документов, соответствующих запросу.

Точность (P) - определяется как отношение числа релевантных документов, найденных системой, к общему числу найденных документов. Полнота (R) - отношение числа найденных релевантных документов, к общему числу релевантных документов в базе. Эти метрики можно рассчитать следующим образом:

$$P = \frac{tp}{tp + fp}; R = \frac{tp}{tp + fn}, \text{где}$$

| | Релевантные | Нерелевантные |
|---|---|---|
| Найденные | Истинно положительные (tp) | Ложно положительные (fp) |
| Не найденные | Ложно отрицательные (fn) | Истинно отрицательные (tn) |

Запрос *night:

- $P = \frac{208}{208+10} = 0{,}954;$
- $R = \frac{208}{208+7} = 0{,}967.$

Значения полноты и точности получились очень близкими к единице, полнота чуть лучше, но все равно обе метрики свидетельствуют о хороших характеристиках системы.

*b. F-мера*

Хорошей мерой для совместной оценки точности и полноты является F-мера, которая определяется как взвешенное гармоническое среднее точности P и полноты R:

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}}, \alpha\epsilon[0,1].$$

При α=1/2 F-мера придает одинаковый вес точности и полноте и называется сбалансированной (F1-мера).

Запрос *night: $F1 = \frac{2*0{,}954*0{,}967}{0{,}954+0{,}967} = \frac{1{,}845}{1{,}921} = 0{,}96.$

Значение F-меры получилось близким к единице, а значит, что и точность, и полнота в данной системе очень хороши.

*c. Кривая точность – полнота (с построением графика кривой)*

Она определяется аналогично ROC-кривой, описанной ниже, только по осям откладываются не FPR (False Positive Rate – доля неверно принятых объектов) и TPR (True Positive Rate – доля верно принятых объектов), а полнота (по оси абсцисс) и точность (по оси ординат).

Построим кривую точность-полнота для запроса вида *shop.

Таблица 9 – Данные для построения кривой

| k | Точность | Полнота |
|---|---|---|
| 1 | 1/1=1 | 1/36=0,027778 |
| 2 | 2/2=1 | 2/36=0,055556 |
| 3 | 3/3=1 | 3/36=0,083333 |
| 4 | 4/4=1 | 4/36=0,111111 |
| 5 | 5/5=1 | 5/38=0,138889 |
| 6 | 6/6=1 | 6/36=0,166667 |
| 7 | 7/7=1 | 7/36=0,194444 |
| 8 | 8/8=1 | 8/36=0,222222 |

| | | |
|---|---|---|
| 9 | 9/9=1 | 9/36=0,25 |
| 10 | 10/10=1 | 10/36=0, 277778 |
| 11 | 11/11=1 | 11/36=0, 305556 |
| 12 | 12/12=1 | 12/36=0, 333333 |
| 13 | 13/13=1 | 13/36=0, 361111 |
| 14 | 14/14=1 | 14/36=0, 388889 |
| 15 | 15/15=1 | 15/36=0, 416667 |
| 16 | 16/16=1 | 16/36=0, 444444 |
| 17 | 17/17=1 | 17/36=0, 472222 |
| 18 | 18/18=1 | 18/36=0,5 |
| 19 | 19/19=1 | 19/36=0, 527778 |
| 20 | 20/20=1 | 20/36=0, 555556 |
| 21 | 21/21=1 | 21/36=0, 583333 |
| 22 | 22/22=1 | 22/36=0, 611111 |
| 23 | 23/23=1 | 23/36=0, 638889 |
| 24 | 24/24=1 | 24/36=0, 666667 |
| 25 | 25/25=1 | 25/36=0, 694444 |
| 26 | 26/26=1 | 26/36=0,722222 |
| 27 | 26/27=0,962963 | 26/36=0,722222 |
| 28 | 26/28=0,928571 | 26/36=0,722222 |
| 29 | 26/29=0,896552 | 26/36=0,722222 |
| 30 | 26/30=0,866667 | 26/36=0,722222 |
| 31 | 27/31=0,870968 | 27/36=0, 75 |
| 32 | 28/32=0,875 | 28/36=0, 777778 |
| 33 | 29/33=0,878788 | 29/36=0, 805556 |
| 34 | 30/34=0,882353 | 30/36=0,833333 |
| 35 | 30/35=0,857143 | 30/36=0,833333 |
| 36 | 30/36=0,833333 | 30/36=0,833333 |
| 37 | 30/37=0,810811 | 30/36=0,833333 |
| 38 | 30/38=0,789474 | 30/36=0,833333 |

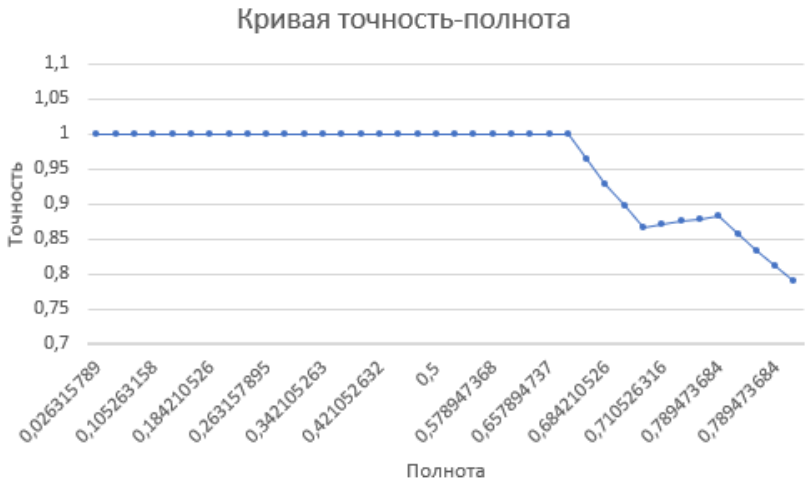


Рисунок 10 – Кривая точность-полнота

Из табл. 1 видно, что в случае, если (k+1)-й найденный документ оказывается нерелевантным, то полнота остается такой же, а точность начинает снижаться (при k=26 и k=34). Если же найден релевантный документ, то и точность, и полнота увеличиваются (например, при k=31) и кривая делает скачок вверх и вправо, что можно увидеть на рис. 7 (отрезок в конце кривой).

*d. Средняя (интерполированная) точность (с построением графика)*

Изучение кривой "точность-полнота" является весьма информативным, однако довольно часто желательно представить всю эту информацию с помощью нескольких и даже одного значения. Традиционно для этого используется средняя точность, интерполированная по одиннадцати точкам (eleven-point interpolated average precision).

Обычно 11 точек берут следующими: $r_j = \frac{j}{10}$, где $j$ от 0 до 10.

То есть вычисляется значение интерполированной точности на 11 уровнях полноты. Так как в нашей системе максимальное значение полноты 0.8, то берется до этого уровня.

Таблица 10 – Значения интерполированной точности

| Полнота | 0 | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 |
|---|---|---|---|---|---|---|---|---|---|
| Интерполированная точность | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0,96 | 0,88 |

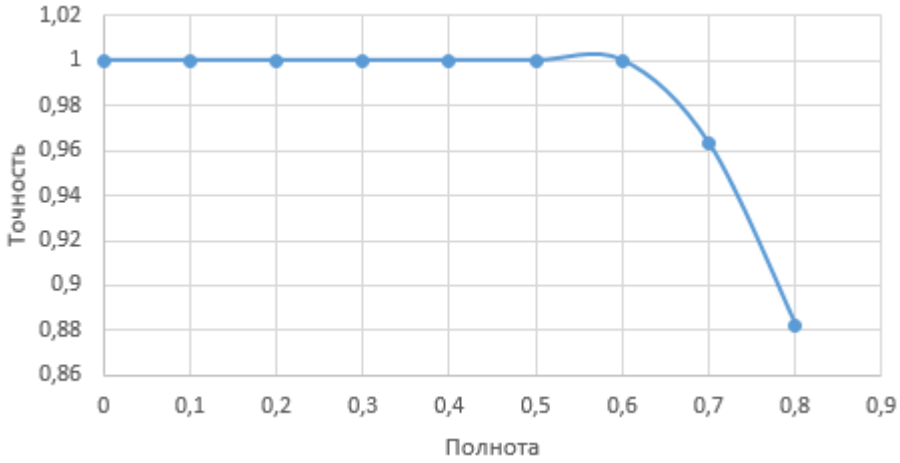


Рисунок 11 - График зависимости интерполированной точности от полноты

*e. Макроусреднение*

Сначала вычисляется итоговая метрика по каждому запросу, а затем результаты усредняются по всем запросам.

Характерен для оценки задач поиска, в которых важен результат в среднем по запросу, независимо от мощности ответа на этот запрос.

Итоговые метрики (полнота и точность) по каждому запросу:

$$P_1 = \frac{208}{208+10} = 0{,}954,\ R_1 = \frac{208}{208+7} = 0{,}967;$$

$$P_2 = \frac{30}{30+8} = 0{,}789,\ R_2 = \frac{30}{30+6} = 0{,}83;$$

$$P_3 = \frac{53}{53+7} = 0{,}883,\ R_3 = \frac{53}{53+2} = 0{,}963.$$

Макроусреднее по точности:

$$Macr_P = \frac{P_1 + P_2 + P_3}{3} = \frac{2{,}626}{3} = 0{,}875;$$

Макроусреднее по полноте:

$$Macr_R = \frac{0{,}967 + 0{,}83 + 0{,}963}{3} = 0{,}92.$$

*f. Микроусреднение*

Найти общий размер всех ответов, правильных ответов, искомых ответов и на их основе вычислить искомую метрику. Полезно в задачах классификации и фильтрации. В данной работе решено вычислять микроусреднее по точности и полноте.

Микроусреднее по точности:

$$Micr_P = \frac{(208 + 30 + 53)}{(208 + 30 + 53) + (10 + 8 + 7)} = \frac{291}{316} = 0{,}92;$$

Микроусреднее по полноте:

$$Micr_R = \frac{(208 + 30 + 53)}{(208 + 30 + 53) + (7 + 6 + 2)} = \frac{291}{306} = 0{,}95.$$

Микроусреднее по полноте и по точности показывает результат близкий к единице, это значит, что говорит значения точности и полноты близки к идеальным для нескольких запросов.

*g. MAP (Mean average precision)*

Одна из наиболее часто используемых метрик качества ранжирования. При точности на уровне k и при средней точности качество ранжирования оценивается для отдельно взятого объекта (поискового запроса). На практике объектов множество: мы имеем дело с миллионами поисковых запросов.

Рассмотрим множество документов, выданных системой вплоть до позиции очередного релевантного документа, и вычислим для этого множества значение точности. Усреднив значения точности всех таких множеств, мы получим среднюю точность (average precision – AP) одного запроса. Далее, для вычисления MAP, среднюю точность усредняют по всем запросам:

$$MAP = \frac{1}{|Q|}\sum_{j=1}^{|Q|} AP_{Kj}$$

Посчитаем среднюю точность для каждого запроса (k=10):

1) *night

$$AP_{k1} = \frac{\left(\frac{0}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10}\right)}{10} =$$

$$= \frac{0{,}5 + 0{,}33 + 0{,}25 + 0{,}2 + 0{,}16 + 0{,}14 + 0{,}125 + 0{,}11 + 0{,}1}{10} =$$

$$= \frac{1{,}915}{10} = 0{,}1915$$

2) *shop

$$AP_{k1} = \frac{\left(\frac{1}{1} + \frac{2}{2} + \frac{3}{3} + \frac{4}{4} + \frac{5}{5} + \frac{6}{6} + \frac{7}{7} + \frac{8}{8} + \frac{9}{9} + \frac{10}{10}\right)}{10} = \frac{10}{10} = 1$$

3) *plan

$$AP_{k3} = \frac{\left(\frac{0}{1} + \frac{0}{2} + \frac{0}{3} + \frac{0}{4} + \frac{0}{5} + \frac{0}{6} + \frac{1}{7} + \frac{2}{8} + \frac{3}{9} + \frac{4}{10}\right)}{10} =$$

$$= \frac{0{,}142 + 0{,}25 + 0{,}33 + 0{,}4}{10} = \frac{1{,}122}{10} = 0{,}1122$$

Посчитаем макроусредненную среднюю точность:

$$MAP = \frac{AP_{k1} + AP_{k2} + AP_{k3}}{3} = \frac{0{,}1915 + 1 + 0{,}1122}{3} = \frac{1{,}3037}{3} = 0{,}4345$$

Значения MAP, вычисленные для разных информационных потребностей (разных запросов) в рамках одной системы, обычно варьируются в широких пределах (от 0,1 до 0,7). В нашем случае показатель MAP оказался равным 0,43.

*h. Точность на уровне k = 10*

Точность на уровне n документов определяется как количество релевантных документов среди первых n выданных документов, делённое на n. Если система выдала более n документов, то эта величина равна точности системы на первых n документах результатов запроса. Если система выдала менее n документов, то точность на уровне n документов будет заведомо не выше точности системы.

Точность на уровне n документов характеризует способность системы выдавать релевантные документы в начале списка результатов. Например, если система выдает не более 10 документов на первой странице, то precision(10) отражает качество результатов системы, получаемых на первой странице.

Точность на уровне k = 10 для запроса *plan:

$$P_{(k=10)} = \frac{4}{10} = 0{,}4.$$

Точность на уровне k = 10 получилась не очень хорошей, что говорит о том, что система выдает мало релевантных документов в начале списка.

*i. R-точность (R-precision)*

По-другому точка баланса (breakeven point).

R-точность равна точности на уровне n документов для n равного количеству релевантных документов для данного запроса. Данная метрика призвана заменить точность на уровне в тех случаях, когда необходимо учесть большую разницу в количестве релевантных документов разных запросов.

Она вычисляется как точность при таком t, при котором полнота равна точности:

$$R\text{ - }precision = P([b(x) > t^*]),$$

где $t^* = \underset{t}{argmin}|P([b(x) > t]) - R([b(x) > t])|$.

Т.е. в нашей коллекции существует |Rel|=36 документов, релевантных запросу (*shop), тогда мы смотрим на первые 36 результатов системы и обнаруживаем, что 26 из них являются релевантными. Тогда R-точность равна:

$$R\text{ - }precision = \frac{26}{36} = 0{,}72,$$

что эквивалентно полноте в данном случае.

*j. Порог отсечения*

Идеальная модель обладает 100% чувствительностью и специфичностью. Однако на практике добиться этого невозможно, более того, невозможно одновременно повысить и чувствительность, и специфичность модели. Компромисс находится с помощью порога отсечения, т.к. пороговое значение влияет на соотношение $S_e$ и $S_p$.

Выбирая точку отсечения (cut-off value), можно управлять вероятностью правильного распознавания положительных и отрицательных примеров. При уменьшении порога отсечения увеличивается вероятность ошибочного распознавания положительных наблюдений (ложноположительных исходов), а при увеличении возрастает вероятность неправильного распознавания отрицательных наблюдений (ложноотрицательных исходов).

Так как в нашей системе пользователю важно получить, как можно больше правильных документов, то было решено в качестве порога отсечения использовать минимально допустимую чувствительность. Необходимо обеспечить чувствительность теста не менее 83%, тогда оптимальным порогом будет максимальная специфичность, которая достигается при 83% чувствительности.

*k. ROC – кривая (с построением графика), чувствительность и специфичность*

Иногда для оценки системы поиска используется кривая соотношений правильного и ложного обнаружения, или кривая ROC (receiver operating characteristics). Кривая ROC

представляет собой зависимость доли истинно положительных или чувствительности от доли ложно положительных результатов равной (1-специфичность). Чувствительность (sensitivity) — это просто синоним полноты. Специфичность вычисляется по формуле:

$$S_p = \frac{tn}{t_n + f_p}.$$

Строится график зависимости: по оси Y откладывается чувствительность $S_e$, по оси X откладывается (1–$S_p$), или, что тоже самое, FPR – доля ложно положительных случаев.

Кривая ROC всегда следует из левого нижнего угла в правый верхний угол. Для хорошей системы график в левом нижнем углу резко поднимается вверх. Поскольку множество истинно отрицательных всегда велико, уровень специфичности для всех информационных потребностей всегда будет близким к единице (и соответственно, доля ложно положительных всегда почти равна нулю).

Так как для последующего анализа необходима оценка ROC-кривой на промежутке от 0 до 1, то было решено нормировать значения чувствительности и специфичности от 0 до 1, приняв за единицу порог отсечения – 83%.

Построим ROC-кривую для запроса вида *shop.

Таблица 11 – Данные для ROC-кривой

| k | Специфичность | 1 - Специфичность | Полнота | Нормированная полнота | Нормированная 1-Специфичность |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 0,027778 | 0,033333333 | 0 |
| 2 | 1 | 0 | 0,055556 | 0,066666667 | 0 |
| 3 | 1 | 0 | 0,083333 | 0,1 | 0 |
| 4 | 1 | 0 | 0,111111 | 0,133333333 | 0 |
| 5 | 1 | 0 | 0,138889 | 0,166666667 | 0 |
| 6 | 1 | 0 | 0,166667 | 0,2 | 0 |
| 7 | 1 | 0 | 0,194444 | 0,233333333 | 0 |
| 8 | 1 | 0 | 0,222222 | 0,266666667 | 0 |
| 9 | 1 | 0 | 0,25 | 0,3 | 0 |
| 10 | 1 | 0 | 0,277778 | 0,333333333 | 0 |
| 11 | 1 | 0 | 0,305556 | 0,366666667 | 0 |
| 12 | 1 | 0 | 0,333333 | 0,4 | 0 |
| 13 | 1 | 0 | 0,361111 | 0,433333333 | 0 |
| 14 | 1 | 0 | 0,388889 | 0,466666667 | 0 |
| 15 | 1 | 0 | 0,416667 | 0,5 | 0 |
| 16 | 1 | 0 | 0,444444 | 0,533333333 | 0 |
| 17 | 1 | 0 | 0,472222 | 0,566666667 | 0 |
| 18 | 1 | 0 | 0,5 | 0,6 | 0 |
| 19 | 1 | 0 | 0,527778 | 0,633333333 | 0 |
| 20 | 1 | 0 | 0,555556 | 0,666666667 | 0 |
| 21 | 1 | 0 | 0,583333 | 0,7 | 0 |
| 22 | 1 | 0 | 0,611111 | 0,733333333 | 0 |
| 23 | 1 | 0 | 0,638889 | 0,766666667 | 0 |

| | | | | | |
|---|---|---|---|---|---|
| 24 | 1 | 0 | 0,666667 | 0,8 | 0 |
| 25 | 1 | 0 | 0,694444 | 0,833333333 | 0 |
| 26 | 1 | 0 | 0,722222 | 0,866666667 | 0 |
| 27 | 0,996815287 | 0,003184713 | 0,722222 | 0,866666667 | 0,125 |
| 28 | 0,993630573 | 0,006369427 | 0,722222 | 0,866666667 | 0,25 |
| 29 | 0,99044586 | 0,00955414 | 0,722222 | 0,866666667 | 0,375 |
| 30 | 0,987261146 | 0,012738854 | 0,722222 | 0,866666667 | 0,5 |
| 31 | 0,987261146 | 0,012738854 | 0,75 | 0,9 | 0,5 |
| 32 | 0,987261146 | 0,012738854 | 0,777778 | 0,933333333 | 0,5 |
| 33 | 0,987261146 | 0,012738854 | 0,805556 | 0,966666667 | 0,5 |
| 34 | 0,987261146 | 0,012738854 | 0,833333 | 1 | 0,5 |
| 35 | 0,984076433 | 0,015923567 | 0,833333 | 1 | 0,625 |
| 36 | 0,98089172 | 0,01910828 | 0,833333 | 1 | 0,75 |
| 37 | 0,977707006 | 0,022292994 | 0,833333 | 1 | 0,875 |
| 38 | 0,974522293 | 0,025477707 | 0,833333 | 1 | 1 |

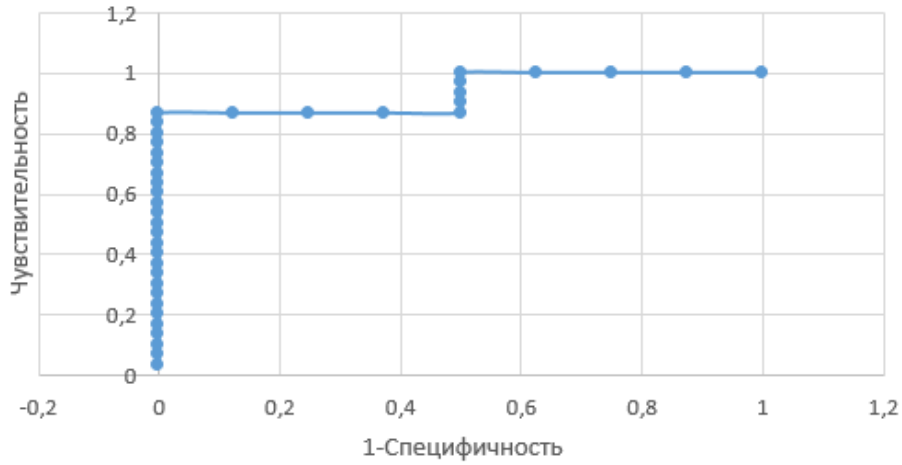


Рисунок 12 - ROC-кривая

Как видно из Рисунка 11 график в нижем левом углу достаточно резко поднимается вверх, что говорит о хорошей системе. Доля истинно положительных результатов гораздо больше доли ложно положительных.

*l. Показатель AUC*

Количественную интерпретацию ROC даёт показатель AUC (area under ROC curve, площадь под ROC-кривой) – площадь, ограниченная ROC-кривой и осью доли ложных положительных классификаций. Площадь принимает значение от 0 до 1. Чем выше показатель AUC, тем качественнее классификатор, при этом значение 0,5 демонстрирует непригодность выбранного метода классификации. Значение менее 0,5 говорит, что классификатор действует с точностью до наоборот.

Посчитаем площадь под кривой на основе построенной выше ROC-кривой.

$$AUC = 0{,}833333334 \cdot 0{,}5 + 0{,}966666667 \cdot 0{,}5 = 0{,}9000000005$$

Чем больше значение AUC, тем «лучше» система. Для хорошей системы в данном случае площадь должна быть как можно ближе к 1. Сравнивая полученный показатель AUC

с максимальным значением площади можно сказать, что система является довольно хорошей.

*m. Совокупная выгода в обычном и нормированном виде*

Cumulative Gain at k:

$$CG@k = \sum_{j=1}^{k} rel(j).$$

CG@k может использоваться и в случае небинарных значений функции релевантности rel(j).

Discounted cumulative gain at k – модификация cumulative gain at k, учитывающая порядок элементов в списке путем домножения релевантности элемента на вес равный обратному логарифму номера позиции:

$$DCG@k = \sum_{j=1}^{k} \frac{2^{rel(j)} - 1}{\log(j+1)}.$$

Если $rel(j)$ принимает только значения 0 и 1, то формула принимает вид:

$$DCG@k = \sum_{j=1}^{k} \frac{rel(j)}{\log(j+1)}.$$

Normalized discounted cumulative gain at k – нормализованная версия DCG@k:

$$NDCG@k = \frac{1}{|Q|} \sum_{q \in Q} n_q DCG@k,$$

где $n_q$ – показатель идеально ранжированной выдачи.

Рассчитаем совокупную выгоду в обычном и нормированном виде для k=10 для запроса *plan.

Определим функцию релевантности:

$$rel = \begin{cases} 0, & \text{документ нерелевантен запросу;} \\ 1, & \text{документ релевантен запросу.} \end{cases}$$

Посчитаем совокупную выгоду и дисконтированную совокупную выгоду:

$$CG@k = 0 + 0 + 0 + 0 + 0 + 0 + 1 + 1 + 1 + 1 = 4$$

$$DCG@k = 0 + \frac{0}{log3} + \frac{0}{log4} + \frac{0}{log5} + \frac{0}{log6} + \frac{0}{log7} + \frac{1}{log8} + \frac{1}{log9} + \frac{1}{log10} + \frac{1}{log11} =$$

$$= 0{,}33 + 0{,}315 + 0{,}301 + 0{,}289 = 1{,}235.$$

Определим показатель идеально нормированной выдачи, для этого рассмотрим идеальный исход для нашего запроса. Лучшим исходом бы была следующая выдача на запрос: {1,1,1,1,0,0,0,0,0,0}.

$$IDCG = DCG@k = 1 + \frac{1}{log3} + \frac{1}{log4} + \frac{1}{log5} + \frac{0}{log6} + \frac{0}{log7} + \frac{0}{log8} + \frac{0}{log9} + \frac{0}{log10} +$$
$$+ \frac{0}{log11} = 1 + 0{,}63 + 0{,}5 + 0{,}43 = 2{,}56.$$

Посчитаем совокупную выгоду в нормированном виде:

$$NDCG@k = \frac{DCG}{IDCG} = \frac{1{,}235}{2{,}56} = 0{,}48.$$

**Вывод**

В ходе проделанной работы были реализованы два типа динамических индексов для поисковой системы: перестройка заново (1) и вспомогательный индекс - весь индекс в одном файле (2). Операции вставки, удаления или изменения документа при 2 типе индекса происходит гораздо быстрее, но иногда необходим момент слияния, который значительно увеличивает время. Тип 1 индекса достаточно прост и его удобно использовать, когда изменения в коллекции документов происходят очень редко, так как при выполнении поиска происходит проход лишь по одному файлу с индексом, а не по двум, как в случае со 2-ым индексом. Также при выполнении поиска с использованием 2-го индекса необходимо хранить в памяти вектор недействительных документов и проходиться по нему, что также увеличивает время. Но если изменения происходят часто, то его использование предпочтительнее.

Также были успешно вычислены метрики качества информационного поиска и построены графики по некоторым из них. Согласно данным метрикам можно оценить качество информационного поиска построенной системы как хорошее. Система успешно удовлетворяет информационные потребности, которые требуются от нее.