Модификация схемы BM25 с помощью генетического алгоритма.

С.П. Воробьев, М.Б. Хорошко, ЮРГТУ (НПИ), Новочеркасск

Быстро растущие информационное пространство объединенных вычислительных сетей порождает новые потребности в обработке, представлении и особенно в поиске данных. На первое место выходит критерий релевантности, который позволяет при его корректном использовании повысить эффективность информационного поиска. Существует достаточно большое количество схем и моделей для решения задачи поиска, одной из которых является BM25.

Схема взвешивания Okapi BM25, была разработана как способ построения вероятностной модели, чувствительной к частоте термина и длине документа, но не использующей большого количества дополнительных параметров. В соответствии с ней каждый документd получает оценку по запросу q, определяемой следующей формулой:

$$Score_{d,q} = \sum_{t \epsilon q} w_{q,t} * w_{d,t}$$

где

$$w_{q,t} = \ln\left(\frac{N - f_t + 0.5}{tf_{q,t} + 0.5}\right) * tf_{q,t}$$

$$w_{d,t} = \frac{(k_1 + 1) f_{d,t}}{K_d + tf_{d,t}}$$

$$K_d = k_1((1 - b) + b\frac{W_d}{W_A})$$

Переменная k_1— это положительный параметр настройки, с помощью которого производится калибровка частоты термина. Переменная b— еще один параметр настройки ($0 \leq b \leq 1$), определяющий нормировку по длине документа. Рекомендуемые значения $k1$ и b - параметры, равны 1.2 и 0.75 соответственно; W_d и W_A – длина документа и средняя длина документа.

Для подбора параметров надстройки будем использовать следующий генетический алгоритм, который получает на вход количество коэффициентов(n)используемых в модели и возвращает подобранные коэффициенты. Общий алгоритм выглядит следующим образом:

1) Создается начальная популяция. Случайным образом из диапазона коэффициентов от C_{min}до C_{max} (диапазон устанавливается для каждого алгоритма), подбираем k_n наборов коэффициентов и переводим их в двоичный вид.
2) Вычисляем приспособленность хромосом. Оцениваем ошибку, для каждого набора коэффициентов.
3) Выбираем двух родителей с наименьшей ошибкой для операции скрещивания.
4) Выбор хромосом для операции мутации.
5) Оценка приспособленности нового набора коэффициентов.
6) Если ошибка n_1 - набора больше заданной ошибки ε_{enter}, то переходим к пункту 3, иначе пункт 7.
7) Полученный набор коэффициентов, который минимизирует ошибку, возвращается в модель поиска.

Рассмотрены более детально основные аспекты:

- Все коэффициенты генерируются изначально случайным образом по равномерному закону при ограничении сверху и снизу. Затем переводятся в двоичный вид, чтобы можно было применять операции скрещивания и мутации.
- Ошибка оценивается по следующей формуле:

$$\varepsilon = \sum_{i=0}^{n} (r(d_i, q_i) - score(d_i, q_i))^2$$

Где, $r(d_i, q_i)$– средняя оценка документа d_i экспертами, по запросу q_i. $score(d_i, q_i)$ – полученная релевантность документа d_i, по запросу q_i.

В ходе экспериментов получены оптимальные операции скрещивания и мутации.

*Операция отбора.*После проведения ряда экспериментов, было выявлено, что для более быстрого получения максимума целевой функции отбор хромосом должен осуществляться по следующему принципу. Для операции скрещивания берется два самых лучших хромосома, и случайным образом N_{kr} хромосом.

Для операции мутации берется два хромосома с самой низкой приспособленностью и N_{mut} хромосом.

*Операция скрещивания***.**Для выбора оптимальной операции скрещивания, был проведен ряд экспериментов с различными методами. В результате определилось два оптимальных метода показанные на рисунке 1. Для проверки эффективности случайным образом делалась выборка запросов от одного до ста. В качестве параметра определяющего оптимальность, бралась средняя оценка релевантности выдачи по данным запросам. Во время эксперимента отключались другие операции. Таким образом функция достигает максимума при сращивании методом «расчески» и очень близко при скрещивании «пополам» (рисунок 2). Решено оставить оба варианта в алгоритме и эксперименты доказали эффективность выбранного способа (рисунок 1). По различным запросам метод расчески достигает максимальной точки по одному набору запросов, метод пополам по двум, а использование двух методов по четырем.

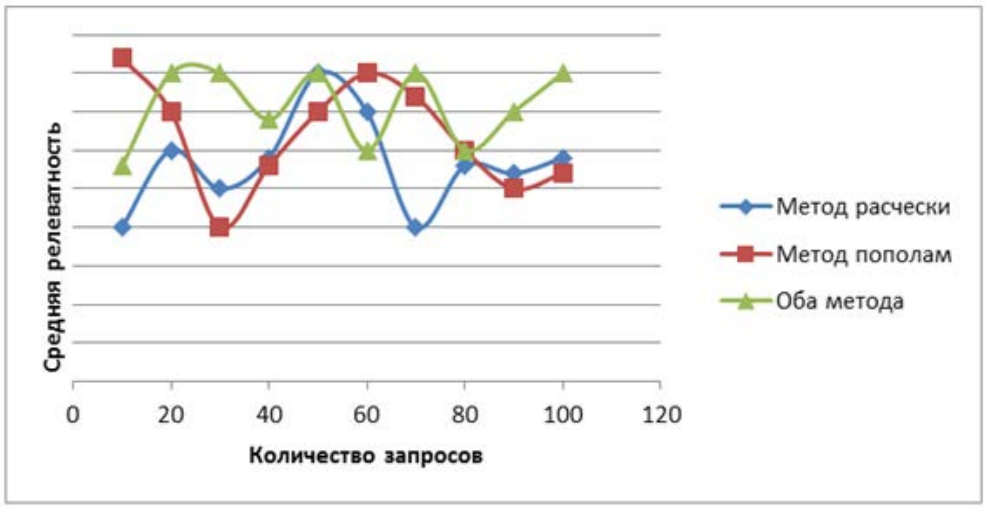

Рис. 1. Операции скрещивания

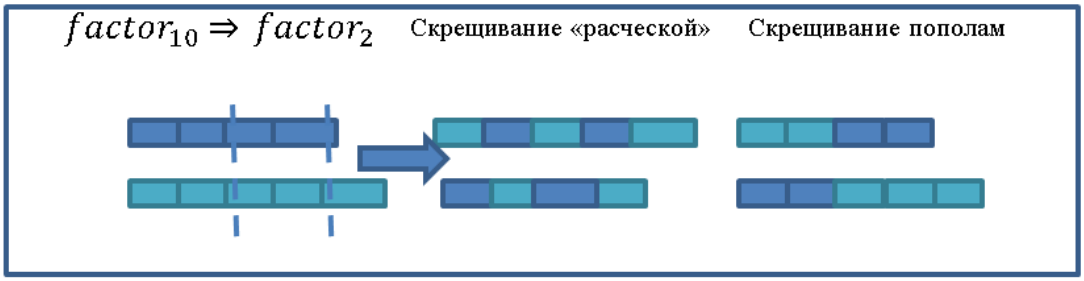

Рис. 2. Методы скрещивания. *При скрещивании «расческой» биты с двух коэффициентов меняются через один. При скрещивании методом пополам, берется половину бит с первого коэффициента и вторую половину со второго коэффициента.*

Операция мутации. Дляопределения оптимальной мутации, был проведен эксперимент, где оценивалась средняя релевантность документов выданных системой при отключенных других механизмах. В результате эксперимента выяснилось, что мутация достигает максимума при вероятности мутирования бита равной 40%. График зависимости результатов поиска от вероятности мутирования показан на рисунке 3.

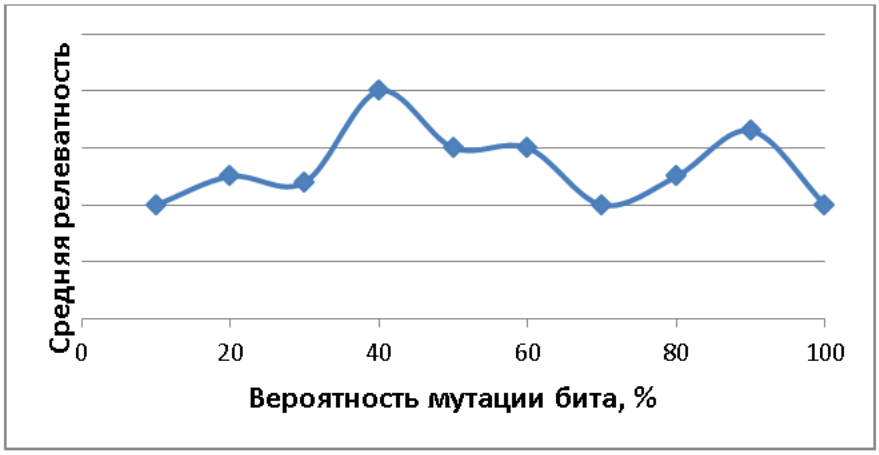

Рис. 3. Зависимость результатов поиска от вероятности мутирования бита

Для проведения эксперимента, было создано две базы запросов – документов. Первая база используется для обучения алгоритма, вторая для оценки. Тестовые коллекции были предоставлены организацией РОМИП, брались две коллекции:

- псевдослучайная выборка сайтов из домена narod.ru объемом 728 000 документов.
- набор, содержащий новостные сообщения из 25 источников и охватывающий 3 временных интервала (около 31 500 документов).

Были сформированы запросы трех типов:

- информационные запросы,
- навигационные запросы,
- транзакционные запросы.

Всего сформировано около 5 000 запросов в равных соотношениях.

Эксперимент. Реализуем модель***OkapiBM25 и*** ее модификацию, где в качестве параметров надстройки будут выступать подобранные значения с помощью генетического алгоритма. Сравниваются полученные метрики оценки для двух систем по 30 запросам.

Полнота $(recall)$ вычисляется как отношение найденных релевантных документов к общему количеству релевантных документов:

Полнота характеризует способность системы находить нужные пользователю документы, но не учитывает количество нерелевантных документов, выдаваемых пользователю. Полнота показана на рисунке 4.

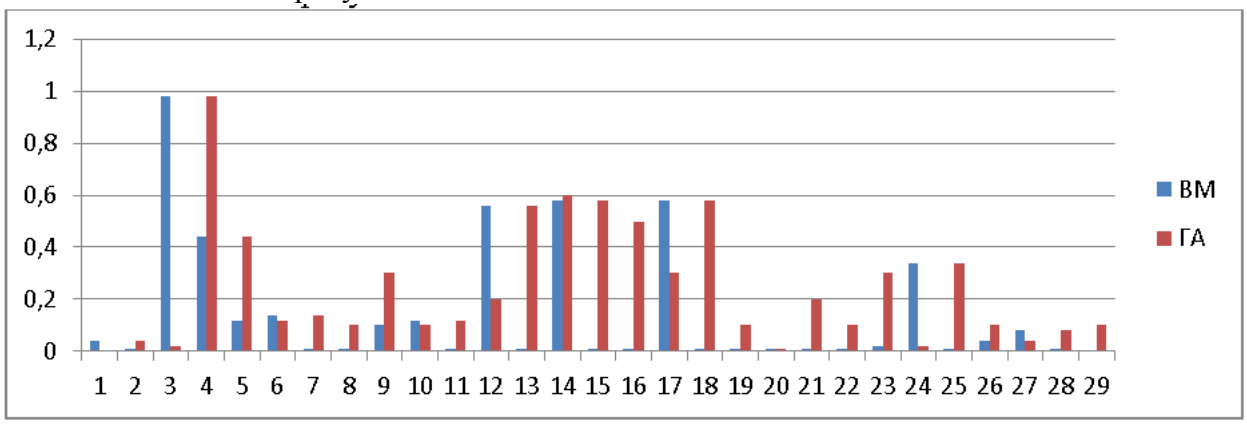

Рис.4. Полнота

Среднее значение полноты: ВМ=0,173, ГА =0,241. ГА показывает лучшую полноту, в среднем на 40%, т.е. пользователь получит на 40% больше релевантных документов.

Точность (*precision*) вычисляется как отношение найденных релевантных документов к общему количеству найденных документов.

Точность характеризует способность системы выдавать в списке результатов только релевантные документы. Точность алгоритмов показана на рисунке 5.

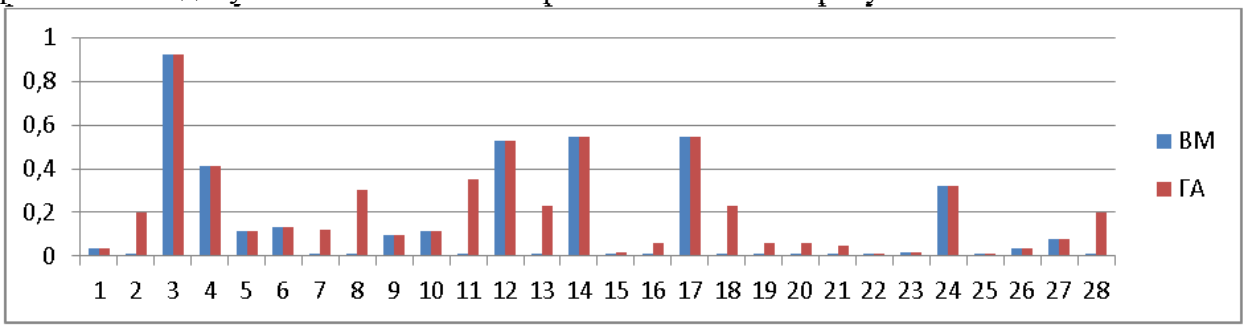

Рис.5. Точность

Среднее значение точности: ВМ=0,167, ГА=0,217. ГА показывает точность, выше на 30%, т.е. больше вероятность, что пользователь получит только релевантные документы на свой запрос.

Аккуратность (*accuracy*) вычисляется как отношение правильно принятых системой решений к общему числу решений. Аккуратность алгоритмов показана на рисунке 6.

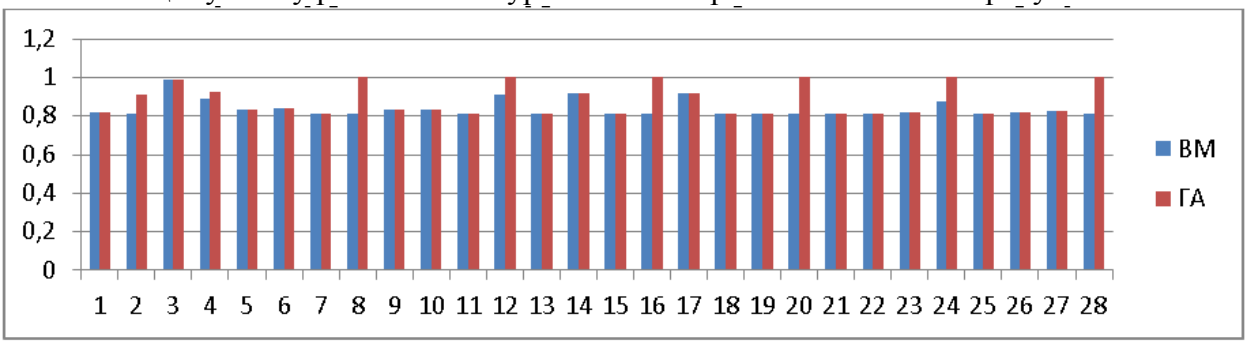

Рис.6. Аккуратность

Среднее значение аккуратности: ВМ=0,832, ГА=0,873. ГА обладает более лучшей аккуратностью на 5%, т.е. система принимает больше правильных решений.

Ошибка $(error)$ вычисляется как отношение неправильно принятых системой решений к общему числу решений. Ошибка алгоритмов полказана на рисунке 7.

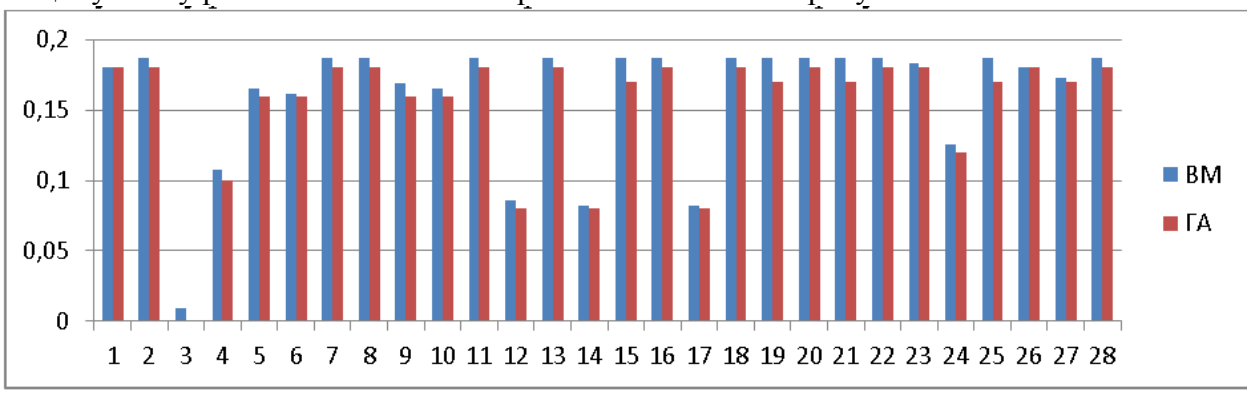

Рис.7. Ошибка

Среднее значение ошибки: ВМ=0,167, ГА=0,150. ГА обладает меньшей ошибкой на 10%, т.е. системой на 10% меньше принято неправильных решений.

F-мера (F) часто используется как единая метрика, объединяющая метрики полноты и точности в одну метрику. F-мера для данного запроса вычисляется по формуле:

$$F = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}}$$

Отметим основные свойства:

- $0 \leq F \leq 1$
- если $\text{recall} = 0$ или $\text{precision} = 0$, то $F = 0$
- если $\text{recall} = \text{precision}$, то $F = \text{recall} = \text{precision}$
- $\min(reacall, precision) \leq F \leq \frac{r+p}{2}$

F-мера алгоритмов полказана на рисунке 8.

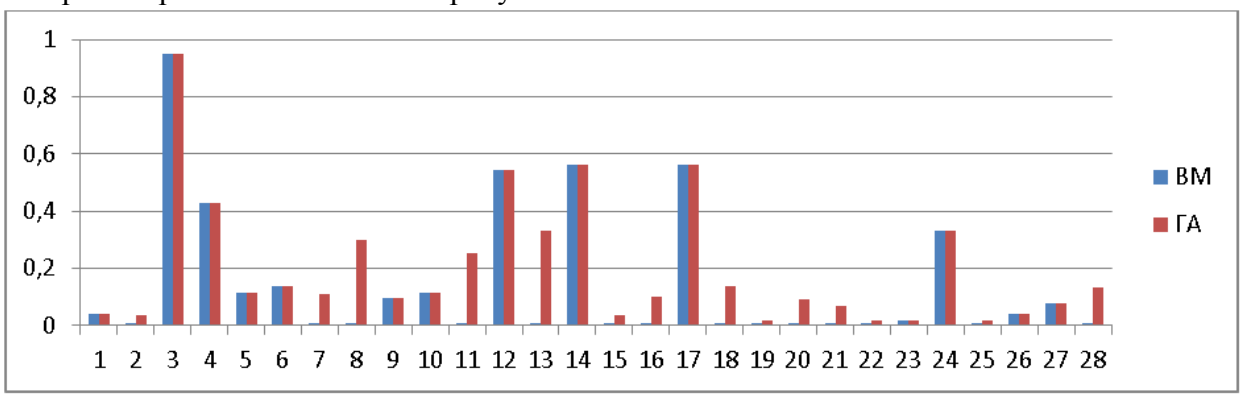

Рис.8. F-мера

Среднее значение f-мера: ВМ=0,17, ГА=0,24. ГА на 40% позволяет улучшить данную метрика, т.е. в среднем ГА выдает лучше результаты на 40%.

Таким образом, модификация с генетическим алгоритмом позволяет улучшить базовую модель в среднем на 40%, т.е. пользователь получит на свой ответ больше релевантных документов на 40%, вероятность того что на запрос будут только релевантные ответы на 30%, на 5% системой принято больше правильных решений, на 10% меньше не правильных.

Литература

1. Sparck Jones, Karen, S. Walker.A probalistic model of information retrieval.б.м. : IP&M, 2000.
2. Маннинг, Кристофер Д. Введение в информационный поиск. М. : Вильямс, 2011.