УДК 004

# Сравнение алгоритмов сжатия текста для активного словаря: код Хаффмана и код Шеннона-Фано

*Стрельцова Марина Николаевна*
*Приамурский государственный университет им. Шолом-Алейхема*
*Студент*

*Научный руководитель*
*Лучанинов Дмитрий Васильевич*
*Приамурский государственный университет имени Шолом-Алейхема*
*старший преподаватель*

**Аннотация**
В данной статье описывается сравнение эффективности двух алгоритмов сжатия, таких как оптимальный код Хаффмана, алгоритм Шеннона – Фано. Для проведения анализа использовался активный словарь на основе отрывка из литературного произведения. В ходе исследования необходимо было находить энтропию текста, среднюю длину кодовых слов, эффективность. В конце исследование сделано сравнение эффективности всех алгоритмов сжатия и выявлен наиболее действенный алгоритм сжатия для текста.
**Ключевые слова:** Код Хаффмана, алгоритм Шеннона-Фано, энтропия, средняя длина, кодовое слово, вероятность.

# Comparison of text compression algorithms for the active dictionary: Huffman code and Shannon-Fano code

*Streltsova Marina Nikolaevna*
*Sholom-Aleichem Priamursky State University*
*Student*

*Scientific adviser*
*Luchaninov Dmitry Vasilyevich*
*Sholom-Aleichem Priamursky State University*
*Senior lecturer*

**Abstract**
This article describes a comparison of the effectiveness of two compression algorithms, such as the optimal Huffman code, the Shannon-Fano algorithm. For the analysis, an active dictionary was used based on an excerpt from a literary work. In the course of the study, it was necessary to find the entropy of the text, the average length of code words, and efficiency. At the end of the study, a comparison was made of the effectiveness of all compression algorithms and the most effective compression algorithm for text was identified.

**Keywords:** Huffman code, Shannon-Fano algorithm, entropy, average length, code word, probability.

В современном мире происходит постоянное увеличение количества информации, которая передается по сети интернет. И проблема сжатия данных не перестает терять свою актуальность. Для решения данной проблемы существует специальные алгоритмы сжатия: код Хаффмана, код Шеннона-Фано.

Эти методы широко известны и применяются во многих исследованиях. Например, Д.О. Ерофеев в своей работе рассматривает реализацию нового подхода к алгоритму Хаффмана и в результате исследования получает программный код, который не только отлично реализует основную концепцию алгоритма Хаффмана, но и оставляет простор для дальнейшего совершенствования этого алгоритма [1]. О.П. Михайлова в своем исследовании рассматривает эффективность кодирования на фундаментальной теореме Шеннона о кодировании источников при отсутствии помех. Если обеспечивается минимальная средняя длина кодовых слов, то кодирование считается экономичным [2]. Е.М. Череданова и Е.А. Мамченко рассматривают метод эффективного кодирования текстовой информации и приходят к выводу, что большая эффективность кодирования сжатия текста достигается за счет учета вероятностей положения алфавита символов на различных позициях в словах [3]. Я.Г. Малиевский и Р.И. Баженов в своей работе разрабатывают приложение на C++ для сжатия данных с помощью алгоритма Д.Хаффмана [4]. Р.С. Добычин и А.Р. Петров в статье проводят сравнение на практике двух методов сжатия – алгоритм Шеннона-Фано и алгоритм Хаффмана [5]. Travis Gagie в своем исследование представляют первый алгоритм для однопроходного мгновенного кодирования [6].

Целью данного исследование является сравнение двух алгоритмов сжатия: код Хаффмана и код Шеннона-Фано и выявление наиболее эффективного алгоритма сжатия для текста.

Для проведения исследования необходимо выбрать активный словарь, для этого используем фрагмент литературного произведения (рис. 1)

Он благополучно избегнул встречи со своею хозяйкой на лестнице. Каморка его приходилась под самою кровлей высокого пятиэтажного дома и походила более на шкаф, чем на квартиру. Квартирная же хозяйка его, у которой он нанимал эту каморку с обедом и прислугой, помещалась одною лестницей ниже, в отдельной квартире, и каждый раз, при выходе на улицу, ему непременно надо было проходить мимо хозяйкиной кухни, почти всегда настежь отворенной на лестницу. И каждый раз молодой человек, проходя мимо, чувствовал какое-то болезненное и трусливое ощущение, которого стыдился и от которого морщился. Он был должен кругом хозяйке и боялся с нею встретиться. Не то чтоб он был так труслив и забит, совсем даже напротив; но с некоторого времени он был в раздражительном и напряженном состоянии, похожем на ипохондрию. Он до того углубился в себя и уединился от всех, что боялся даже всякой встречи, не только встречи с хозяйкой. Он был задавлен бедностью; но даже стесненное положение перестало в последнее время тяготить его. Насущными делами своими он совсем перестал и не хотел заниматься. Никакой хозяйки, в сущности, он не боялся, что бы та ни замышляла против него. Но останавливаться на лестнице, слушать всякий вздор про всю эту обыденную дребедень, до которой ему нет никакого дела, все эти приставания о платеже, угрозы, жалобы, и при этом самому изворачиваться, извиняться, лгать, - нет уж, лучше проскользнуть как-нибудь кошкой по лестнице и улизнуть, чтобы никто не видал.

Рис. 1. Отрывок произведения для генерации активного словаря

В начале исследования необходимо выяснить частоту вхождения каждого символа в исходный текст. Общее количество символов составляет 1471. Далее необходимо высчитать вероятность использования каждого символа по формуле $P = \frac{m}{n}$ (m – количество повторений символа, n – общее количество символов).

Следующим шагом будет вычисление энтропии исходного текста.

$$H = \sum_{i=1}^{n} p_i \log_2 p_i$$

Результаты этих расчётов занесены в таблицу и представлены на рис. 2.

| символ | кол-во повторений | вероятность | кол-во всех сиволов |
|---|---|---|---|
| пробел | 227 | 0,154 | 1471 |
| О | 157 | 0,107 | Энтропия |
| Е | 97 | 0,066 | 4,474 |
| Н | 90 | 0,061 | |
| И | 84 | 0,057 | |
| Т | 74 | 0,050 | |
| А | 72 | 0,049 | |
| С | 65 | 0,044 | |
| Л | 58 | 0,039 | |
| Р | 49 | 0,033 | |
| В | 45 | 0,031 | |
| К | 42 | 0,029 | |
| Д | 36 | 0,024 | |
| Я | 34 | 0,023 | |
| У | 34 | 0,023 | |
| М | 33 | 0,022 | |
| , | 29 | 0,020 | |
| П | 27 | 0,018 | |
| Б | 24 | 0,016 | |
| Й | 23 | 0,016 | |
| Ь | 21 | 0,014 | |
| З | 21 | 0,014 | |
| Г | 20 | 0,014 | |
| Ж | 17 | 0,012 | |
| Ы | 17 | 0,012 | |
| Х | 16 | 0,011 | |
| Ч | 14 | 0,010 | |
| . | 11 | 0,007 | |
| Ю | 8 | 0,005 | |
| Ц | 6 | 0,004 | |
| Щ | 6 | 0,004 | |
| Э | 5 | 0,003 | |
| Ш | 5 | 0,003 | |
| - | 3 | 0,002 | |
| Ф | 1 | 0,001 | |

Рис. 2. Результаты расчётов

Оптимальный код Хаффмана – это метод оптимального побуквенного кодирования, был разработан в 1952 г. Д. Хаффманом. Оптимальный двоичный код Хаффмана имеет минимальную среднюю длину кодового слова среди всех побуквенных кодов для данного источника с алфавитом $A=\{a_1,a_2,\ldots a_n\}$ и вероятностями $p_i = P(a_i)$ и сумма всех вероятностей равна единице.

Рассмотрим построение кода Хаффмана на примере активного словаря. Для начала необходимо все символы активного словаря $A=\{a_1,a_2,\ldots a_n\}$ упорядочить по убыванию их вероятностей $p_1 \geq p_2 \geq \ldots \geq p_n$. Далее будем суммировать две наименьшие вероятности и полученную вероятность включать на соответствующее место в упорядоченном списке до тех пор, пока в списке не останется два символа (рис. 3). Эти два символа закодируем 0 и 1.

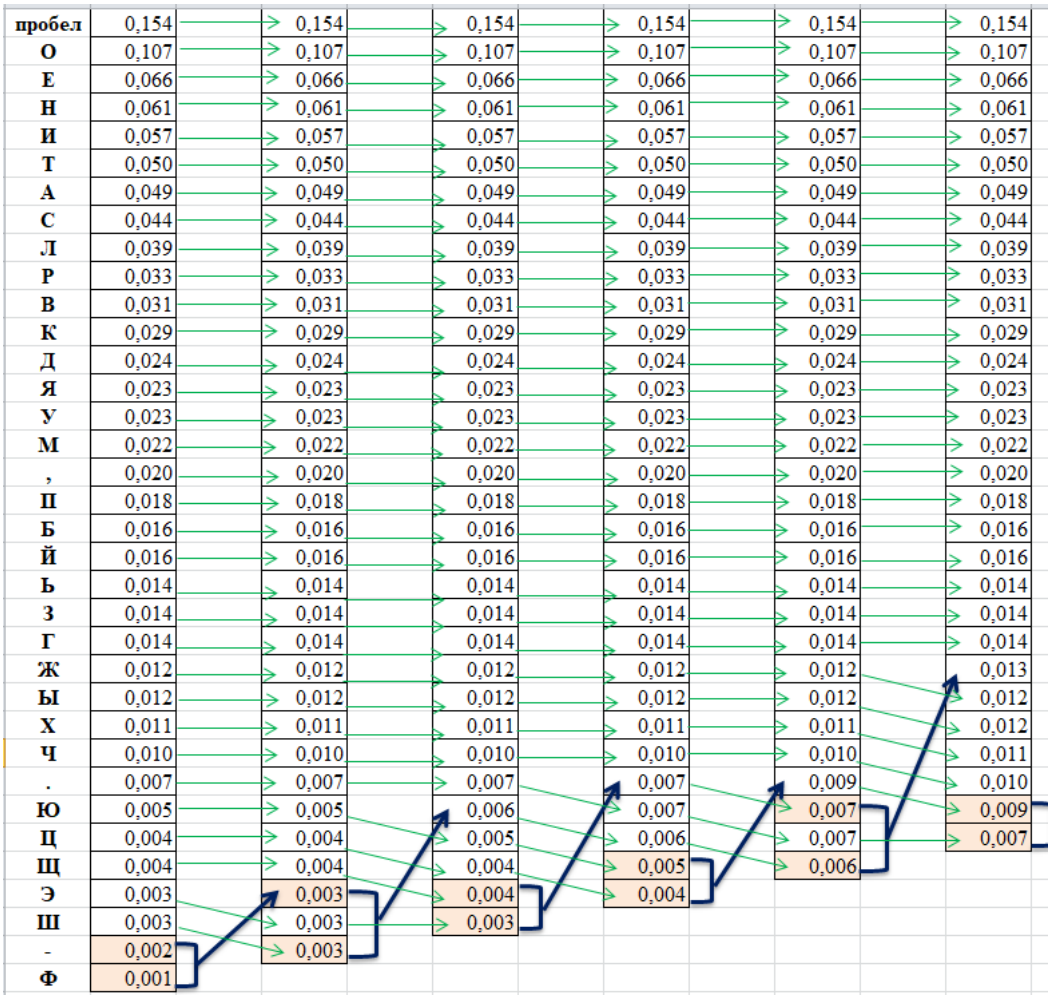


Рис. 3. Процесс построения кода Хаффмана

После завершения процесса построения и определения кодового слова, необходимо высчитать среднюю длину, построенного кода Хаффмана по формуле: $L_{cp}(P) = \sum P_i \times L_i$

$L_{ср}(P)$ =0,154*3+0,107*3+0,066*4+0,061*4+…+0,001*9=4,503

Далее необходимо найти эффективность сжатия текста $R = L_{cp} - H$

R= 4,503-4,474=0,029

Код Хаффмана чаще всего строится и хранится в виде двоичного дерева, где символы алфавита находятся в листьях, а на ветвях присваиваются значения 0(левая часть) и 1(правая часть) (рис. 4).

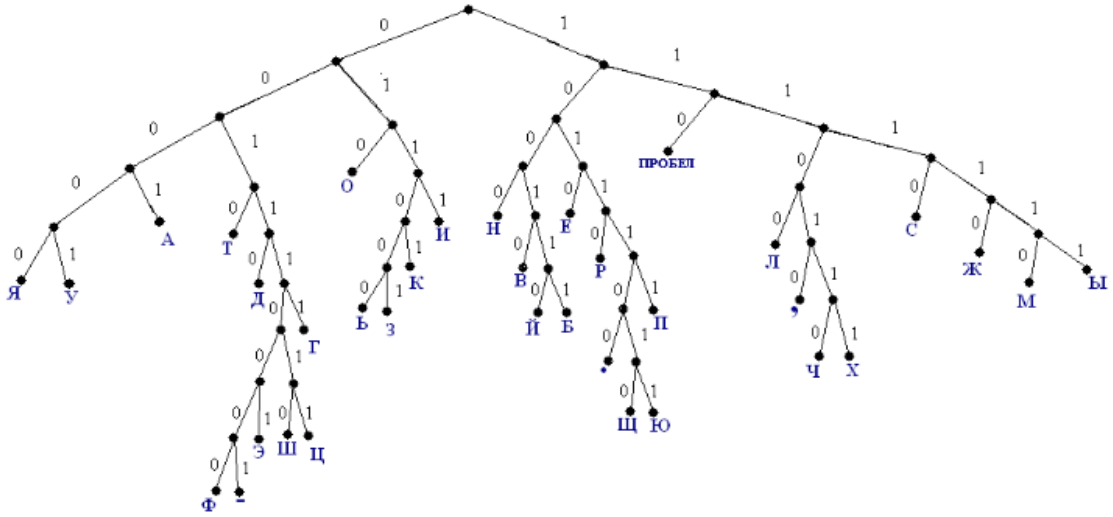


Рис. 4. Двоичное дерево для кода Хаффмана

Алгоритм Шеннона-Фано является одним из первых алгоритмов сжатия, который впервые определили американские учёные Клод Шеннон и Роберт Фано. Этот метод сжатия имеет большое сходство с алгоритмом Хаффмана, который появился на пару лет позже и является логическим продолжением алгоритма Шеннона. Алгоритм использует коды переменной длины: часто встречающийся символ кодируется кодом меньшей длины, редко встречающийся – кодом большей длины. Коды Шеннона-Фано – префиксные, то есть никакое кодовое слово не является префиксом любого другого. Это свойство позволяет однозначно декодировать любую последовательность кодовых слов.

Алгоритм построения кода Шеннона-Фано (рис. 5):

1. Упорядочим символы активного словаря по убыванию их вероятностей.

2. Вычислим величины $Q_i$, i=1…n которые называются кумулятивные вероятности

$Q_0 = 0$

$Q_1 = p_1$

$Q_2 = p_1 + p_2$

…

$Q_i = p_1 + p_2 + \ldots + p_i$

$Q_n = p_1 + p_2 + \ldots + p_i + \ldots + p_n = 1$

3. Представим $Q_{i-1}$ в двоичной системе счисления и возьмем в качестве кодового слова первые $|-\log p_i|$ двоичных знаков после запятой, i=1…n.

Для вероятностей, представленных в виде десятичных дробей, удобно определить длину кодового слова Li из соотношения

$$\frac{1}{2^{Li}} \le p_i < \frac{1}{2^{Li-1}}, i = 1,\ldots,n.$$

| Символ | Вероятность (Pi) | Кумулятивные вероятности ($Q_{i-1}$) | Длина кодового слова (Li) | Кодовое слово |
|---|---|---|---|---|
| пробел | $0,5^3 \le 0,154 < 0,5^2$ | 0 | 3 | 000 |
| О | $0,5^4 \le 0,107 < 0,5^3$ | 0,154 | 4 | 1000 |
| Е | $0,5^4 \le 0,066 < 0,5^3$ | 0,261 | 4 | 0100 |
| Н | $0,5^5 \le 0,061 < 0,5^4$ | 0,327 | 5 | 10000 |
| И | $0,5^5 \le 0,057 < 0,5^4$ | 0,388 | 5 | 01000 |
| Т | $0,5^5 \le 0,050 < 0,5^4$ | 0,445 | 5 | 00100 |
| А | $0,5^5 \le 0,049 < 0,5^4$ | 0,495 | 5 | 00010 |
| С | $0,5^5 \le 0,044 < 0,5^4$ | 0,544 | 5 | 00001 |
| Л | $0,5^5 \le 0,039 < 0,5^4$ | 0,588 | 5 | 11000 |
| Р | $0,5^5 \le 0,033 < 0,5^4$ | 0,627 | 5 | 01100 |
| В | $0,5^6 \le 0,031 < 0,5^5$ | 0,66 | 6 | 110000 |
| К | $0,5^6 \le 0,029 < 0,5^5$ | 0,691 | 6 | 001100 |
| Д | $0,5^6 \le 0,024 < 0,5^5$ | 0,72 | 6 | 000110 |
| Я | $0,5^6 \le 0,023 < 0,5^5$ | 0,744 | 6 | 000111 |
| У | $0,5^6 \le 0,023 < 0,5^5$ | 0,767 | 6 | 000011 |
| М | $0,5^6 \le 0,022 < 0,5^5$ | 0,79 | 6 | 111000 |
| , | $0,5^6 \le 0,020 < 0,5^5$ | 0,812 | 6 | 011100 |
| П | $0,5^6 \le 0,018 < 0,5^5$ | 0,832 | 6 | 001110 |
| Б | $0,5^6 \le 0,016 < 0,5^5$ | 0,85 | 6 | 000111 |

Рис. 5. Часть построенного алгоритма кода Шеннона-Фано

После нахождения кодового слова высчитываем среднюю длину кодового слова по формуле: $L_{cp}(P) = \sum P_i \times L_i$

$L_{cp}(P)$ =0,154*3 + 0,107*4 + 0,066*4 + 0,061*5 +…+ 0,001*10 = 4,562

Далее необходимо найти эффективность сжатия теста $R = L_{cp} - H$

R= 4,562-4,474=0,088

Для подведения итога данного исследования и выявления наиболее эффективного алгоритма сжатия, необходимо сравнить полученные результаты R (эффективности сжатия текста) (Таблица 1).

Таблица 1 – Результаты R

| Код Хаффмана | Код Шеннона-Фано |
|---|---|
| R = 0,029 | R = 0,088 |

Эффективность сжатия у кода Хаффмана является меньшей, чем у кода Шеннона-Фано, отсюда следует, что код Хаффмана является наиболее действенным алгоритмом сжатия для текста с использованием активного словаря.

В результате данного исследования были рассмотрены два алгоритма сжатия: код Хаффмана и код Шеннона-Фано. На основе актуального словаря были вычислены кодовые слова символов, длины кодовых слов и средние

длины кодовых слов, найдена эффективность сжатия. Сравнивая эффективность сжатия, был определен наилучший алгоритм сжатия для текста.

**Библиографический список**


1. Ерофеев Д.О. Алгоритм Хаффмана и новый метод его реализации // Наука и современность. 2010. №2-2. С.338-343.
2. Михайлова О.П. Метод Шеннона-Фано как способ эффективного кодирования информации // Интеграция наук. 2016. №3 (3). С. 39-40.
3. Череданова Е.М., Мамченко Е.А. Алгоритмы сжатия текстовых файлов // Молодой ученый. 2017. № 44(178). С. 24-26.
4. Малиевский Я.Г., Баженов Р.И. Разработка компьютерной программы сжатия файлов на основе кода Хаффмана // Актуальные направления научных исследований XXI века: теория и практика. 2015. №7-4 (18-4). С. 434-437.
5. Добычин Р.С., Петров А.Р. Сравнительный анализ работы алгоритмов Шеннона-Фано и Хаффмана // Междисциплинарные исследования в области математического моделирования и информатики. 2016. С. 12-14.
6. Travis Gagie Dynamic Shannon coding // Information Processing Letters. 2007. № 102. С. 113-117.