УДК 004.6

В. В. Хрущак
В. І. Месюра

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ СУМАРИЗАЦІЇ ДОКУМЕНТІВ

Вінницький національний технічний університет

Анотація
В цій роботі розглянуто деякі алгоритми побудови мульти-документної сумаризації. Також проведено їх порівняльний аналіз.
Ключові слова: сумаризація, зведення, алгоритми.

Abstract
In permoed an study of algorithms for multi-document summarization. Also conducted a comparative analysis.
Keywords: summarization, summary, algoritms.

Вступ

Автоматичне створення об'єднаного документу на основі багатьох є однією з найстарших проблем інформаційного пошуку. Зростаюча кількість використання мобільних пристроїв і поширення інформації в усі галузі людської активності породжує необхідність в сумаризації кластерів пов'язаних документів (мульти-документна сумаризація).

Будуючи зведення (результат сумаризації) зазвичай оптимізують три параметри:

- Актуальність – відповідність результатів очікуванням;
- Надмірність – зведення не повинно містити повторення;
- Розмірність – зведення, як правило обмежене по довжині;

Оптимізація всіх трьох параметрів є складною задачею, відомою, як глобальна проблема логічного виведення. Ця задача вирішується одним з двох шляхів. Перший передбачає оптимізацію актуальності та надмірності окремо. Інший підхід це оптимізація обох параметрів одночасно, один з таких методів це MMR – максимально гранична актуальність.

Постановка задачі

На вході в нас є колекція документів $D = \{D_1, \dots, D_k\}$. Кожен документ D містить набір текстових блоків $D_i = \{t_1, \dots, t_m\}$ (слів, речень, абзаців, тощо). Додатково ми визначимо три функції:

1. *Rel(i)* – актуальність блоку t_i з зведення S.
2. *Red(i,j)* – надмірність між блоками t_i і t_j, які присутні в зведені S.
3. *l(i)* – розмірність блоку t_i.

Тоді задачу сумаризації можна представити, як:

$$S = \arg\max_{S \subseteq D} s(S) = \arg\max_{S \subseteq D} \sum_{t_i \in S} Rel(i) - \sum_{t_i t_j \in S, i<j} Red(i,j) \quad (1)$$

при чому

$$\sum_{t_i \in S} l(i) \le K$$

де s(S) - показник ефективності сумаризації,
K – частина вводу, обмеження розмірності зведення.

Алгоритми розв'язку задачі

Жадібний алгоритм. Простий алгоритм оптимізації функції (1), який включає блоки з найбільш релевантною інформацією, а потім обирає з них, ті що максимізують функцію (1). Перевагою цього алгоритму є простота та мала потреба в обчислювальних ресурсах. В найгіршому випадку алгоритм виконуватиметься за *O(n log n + Kn)*. Звісно, він має значний недолік, так як він обирає найбільш релевантні блоки він може обрати такий, який буде найбільш релевантним, але і міститиме багато "шуму", а отже, так як в нас є обмеження до розмірності зведення K в нас залишиться мало місця для решти блоків. Такий випадок часто трапляється при сумаризації новин, які, зазвичай, містять речення великої довжини. Псевдо код для реалізації жадібного алгоритму наведено на рисунку 1.

Ввід: $\boldsymbol{D} = \{t_1, \ldots, t_n\}, K$

(a) Жадібний алгоритм

1. sort $\boldsymbol{D}$ so that $Rel(i) > Rel(i+1)\ \forall i$
2. $S = \{t_1\}$
3. while $\sum_{t_i \in S} l(i) < K$
4. $\quad t_j = \arg\max_{t_j \in \boldsymbol{D} - S} s(S \cup \{t_j\})$
5. $\quad S = S \cup \{t_j\}$
6. return S

(b) Алгоритм "рюкзак"

1. $S[i][0] = \{\}\ \forall 1 \leq i \leq n$
2. for i: $1 \ldots n$
3. $\quad$ for k: $1 \ldots K$
4. $\quad\quad S' = S[i-1][k]$
5. $\quad\quad S'' = S[i-1][k - l(i)] \cup \{t_i\}$
6. $\quad\quad$ if $s(S') > s(S'')$ then
7. $\quad\quad\quad S[i][k] = S'$
8. $\quad\quad$ else
9. $\quad\quad\quad S[i][k] = S''$
10. return $\arg\max_{S[n][k],\ k \leq K} s(S[n][k])$

Рисунок 1 – (а) Жадібний алгоритм. (б) Алгоритм динамічного програмування "рюкзак"

Динамічний алгоритм. Фактично є модифікацією жадібного алгоритму за допомогою методів динамічного програмування. Блоки можуть викидатись з набору та замінюватись на більш відповідні. Для цього використовують два показника. Перший *S[i − 1][k]* – показник довжини k, використовуючи блоки $\{t_1, \ldots, t_{i-1}\}$. Другий це показник довжини $k - l(i)$ враховуючи поточний блок t_i . Тоді в S[i][k] записується блок з найбільшими показниками.

Алгоритм лінійного програмування. Проблема сумаризації може бути сформована, як задача лінійного програмування (рисунок 2). В цій постановці введені змінні α_i та α_{ij}, які дорівнюють 1, коли блок включено до зведення. Ціль це обрати такі змінні α_i та α_{ij}, щоб максимізувати оцінку зведеного документу.

Максимізувати $\sum_i \alpha_i Rel(i) - \sum_{i<j} \alpha_{ij} Red(i,j)$

так, що $\forall i, j$:

(1) $\alpha_i, \alpha_{ij} \in \{0, 1\}$
(2) $\sum_i \alpha_i l(i) \leq K$
(3) $\alpha_{ij} - \alpha_i \leq 0$
(4) $\alpha_{ij} - \alpha_j \leq 0$
(5) $\alpha_i + \alpha_j - \alpha_{ij} \leq 1$

Рисунок 2 – Постановка задачі сумаризації, як задачі лінійного програмування

Результати дослідження

Для проведення тестування, як текстовий блок було обрано речення. Кожен блок представлено, як набір слів. Для тестів було обрано пакет прикладних програм ROUGE, а наведені показники є оцінками ROUGE-1 та ROUGE-2 з цього пакету. В якості вхідного набору документів обрано набір DUC 2002, для якого вже існують власноруч створені зведення. Результати дослідження наведені на рисунку 3.

	Довжина зведення		
	50	100	200
Жадібний	26.8 / 5.1	33.5 / 6.9	40.1 / 9.5
Рюкзак	27.9 / 5.9	34.8 / 7.3	41.2 / 10.0
ЛП	28.1 / 5.8	34.6 / 7.2	41.5 / 10.3

Рисунок 3 – Результати дослідження

Висновки

В цій роботі було досліджено три варіанти алгоритму мульти-документної сумаризації. Проведені дослідження показали, що алгоритм динамічного програмування показав найбільш оптимальні показники точності та можливостей для розширення, в порівнянні з жадібним алгоритмом та алгоритмом лінійного програмування.

Для подальших досліджень потрібно детальніше розглянути алгоритми побудови зведених текстів та провести модифікацію вже розглянутих, з дослідами, що покажуть їх ефективність.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. R. Barzilay Inferring Strategies for Sentence Ordering in Multidocument News Summarization /R. Barzilay, N. Elhadad, and K. McKeown. - Journal of Artificial Intelligence Research, 2002. - 35–55 pp.
2. T.H. Cormen. Introduction to Algorithms / T.H. Cormen, C.E. Leiserson, and R.L. Rivest. - MIT Press/McGraw-Hill, 1990.
3. J. Goldstein Multi-document summarization by sentence extraction / J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz - In Proceedings of the ANLP/NAACL Workshop on Automatic Summarization, 2000.

Хрущак Володимир Вікторович – аспірант кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця. vladimir.khruschak@gmail.com

Володимир Іванович Месюра — канд. техн. наук, доцент, професор кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця.

***Khruschak* V. Volodymyr** — Department of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia. vladimir.khruschak@gmail.com

***Volodymyr I. Mesyura* —** Cand. Sc., Assistant Professor, Professor of the Computer Science Chair, Vinnytsia National Technical University, Vinnytsia.