Дана загальна характеристика методів і алгоритмів автоматичного реферування тексту, таких як базовий алгоритм, алгоритм Freq, та LRU-K

Ключові слова: анотація, Freq, LRU-K, семантична матриця

Дана общая характеристика методов и алгоритмов автоматического реферирования текста, таких как базовый алгоритм, алгоритм Freq и LRU-K

Ключевые слова: аннотация, Freq, LRU-K, метод семантическая матрица

This article represents general features of methods and algorithms of automatic annotation of the text, such as basic algorithm, Freq algorithm and LRU-K

Key words: abstract, Freq, LRU-K, semantic matrix

УДК 004.4'414

АНАЛИЗ АЛГОРИТМОВ АВТОМАТИЧЕСКОГО РЕФЕРИРОВАНИЯ ТЕКСТА

Е.А. Гридина
Кафедра программной инженирии
Харьковский национальный университет радиоэлектроники
пр. Ленина 14, г. Харьков, Украина, 61166
Контактный тел.: 093-982-93-92
E-mail: Gridina_Kate@mail.ru

1. Введение

Искусство реферирования, или составления аннотаций, или кратких изложений материала, иными словами, извлечения наиболее важных или характерных фрагментов из одного или многих источников информации, стало неотъемлемой частью повседневной жизни. Новости, которые предлагает нам телевидение, – это суть реферат мировых событий дня.

В настоящее время известно много алгоритмов автоматического аннотирования или формирования краткого содержания документов, например МЛ Аннотатор, Золотой ключик, TextAnalyst, системы IBM Intelligent Text Miner, Oracle Context и Inxight Summarizer (компонент АИПС AltaVista). Их возможности ограничены выделением и выбором оригинальных фрагментов из исходного документа и соединением их в короткий текст.

Подготовка же краткого изложения предполагает передачу основной мысли текста, и не обязательно теми же словами. Текст, полученный путем соединения отрывочных фрагментов, лишен гладкости, его трудно читать.

Для достижения наиболее оптимального результата, необходимо совместить несколько алгоритмов реферирования. В данной работе нами будут рассмотрены три алгоритма формирования контекстно-зависимых аннотаций.

2. Исследование существующих систем автоматического реферирования

На международном рынке представлено множество программных продуктов, которые позволяют создавать авторефераты для текстовых файлов. Ориентированы они преимущественно для файлов, содержащих текст на английском языке. Продукты отечественного IT рынка способны составлять авторефераты на русском языке, вследствие этого они наиболее интересны для изучения и дальнейшего развития исследований в области автореферирования. Существуют три наиболее популярные программы автореферирования: МЛ Аннотатор, Золотой ключик, TextAnalyst.

Программа МЛ Аннотатор [1] составляет связный реферат документа. «Коэффициент сжатия» реферата задаётся пользователем. Программа может работать в двух режимах: реферирование и выделение ключевых слов. В режиме с его содержание. В режиме выделения ключевых слов производится выборка из текста наиболее информативных слов.

Процесс работы программы «Золотой ключик» [2] выглядит следующим образом: на стандартный вход программе подается произвольный текст на русском языке, на стандартном выходе программа формирует аннотацию данного текста и список рубрик, к которым относится данный текст.

TextAnalyst [3] используется в качестве инструмента для анализа содержания текстов, смыслового поиска информации, формирования электронных архивов.

Из рассмотренных программных продуктов, на данный момент можно выделить «Золотой ключик» как наименее гибкий и функциональный инструмент для задач автореферирования. Реализацию данной программы основанной на алгоритме, работающем по принципу фильтрации на базе тезауруса, трудно модифицировать под динамическую постоянно добавляющуюся базу знаний.

TextAnalyst как программный продукт, основанный на алгоритмах создания семантических сетей, проявляет гибкость при работе с базами знаний и алгоритмами формирования смыслового портрета.

Наибольшие перспективы в данной области видятся в развитии взаимодействия и совмещения алгоритмов формирования семантических сетей и алгоритмов поисковых машин в глобальной сети Интернет. И создание на базе совмещённых алгоритмов новых, общедоступных сервисов интеллектуального поис-

ка информации, а также систем автореферирования больших объёмов текстовой информации.

3. Описание методов автоматического реферирования

Для достижения хорошего результата необходимо совместить несколько алгоритмов аннотирования текста.

Оптимальным решением данной задачи мы выбрали несколько алгоритмов, анализ которых приведен ниже. Это алгоритмы Freq, LRU-K и семантический анализ [6].

Алгоритм Freq [4], учитывает частоту слов в окне длиной в 1000 слов вокруг анализируемого фрагмента (то есть фрагмент находился в середине окна). Отбирается 10 слов, которые встречались в данном фрагменте наиболее часто. Для расчетов используется следующая формула (1) для вычисления веса:

$$Wfrq = Wb + Sum \times (\log_2 Fk) \quad (1)$$

где Wb - вес, вычисленный по базовому алгоритму, Fk - сколько раз слово встретилась в окне в 1000 слов, включающий фрагмент.

Таким образом, наибольший вес получают те фрагменты, которые кроме того, что содержат наибольшее число слов запроса, но и большее количество слов часто встречающихся в документе.

При статистической обработке документа в качестве метрики значимости термина обычно используется частота его встречаемости. По нашему мнению, данной метрике свойственен ряд недостатков. Во-первых, частота не несет информации о распределении слова в документе – распределено ли оно равномерно по всему документу или только в некоторых фрагментах, во-вторых, вычисление частот для фрагментов документов может требовать относительно много ресурсов.

В последнее время определенную популярность приобрели более сложные статистические модели, обычно основанные на Марковских цепях. Однако эти модели достаточно сложны и имеют еще более высокую вычислительную сложность. Алгоритм LRU-K [5], является вариантом алгоритма «последний недавно использованный». Мы исходили из известного в психологии предположения, что человек в быстрой памяти сохраняет только относительно малое количество объектов, поэтому алгоритмы класса «последний используемый» должны показывать хорошие результаты.

Нам не известно упоминаний в литературе использования подобных алгоритмов для анализа текста. Алгоритм делает следующее:

1) при инициализации создается 3 структуры данных: массивы слов и 2 массива с указателями на слова (array1 и array2) и длинами k;

2) для каждого слова при обработке документа производится следующие действия:

а) поиск в массиве слов.

1. Если слово не найдено, то ссылка на него помещается в массив array1 в первую позицию, остальные позиции в массиве сдвигаются, самое последнее слово удаляется из array1 и из массива слов.

2. Если слово найдено и встречается в array1, то оно из него удаляется и переносится на первую позицию в array2. При этом, если array2 полностью заполнен, то последнее слово из него так же удаляется, как и в первом случае.

3. Если слово найдено и уже есть в array2, то оно просто перемещается на первую позицию.

Легко можно показать, что если бы слова в тексте имели равную вероятность появления, то после обработки фрагмента текста, содержащего слов намного больше k, содержимое массива array2 совпадало бы с k наиболее часто встретившихся слов. То есть данный алгоритм можно рассматривать как один из вариантов оценки локальной частоты терминов, при предположении равномерного распределения слов. Однако, предлагаемый алгоритм, кроме этого, должен выделять слова, которые имеют не только высокую частоту, но и равномерно распределенные вблизи выбираемого фрагмента.

Алгоритм упрощенно представлен на рис. 1.

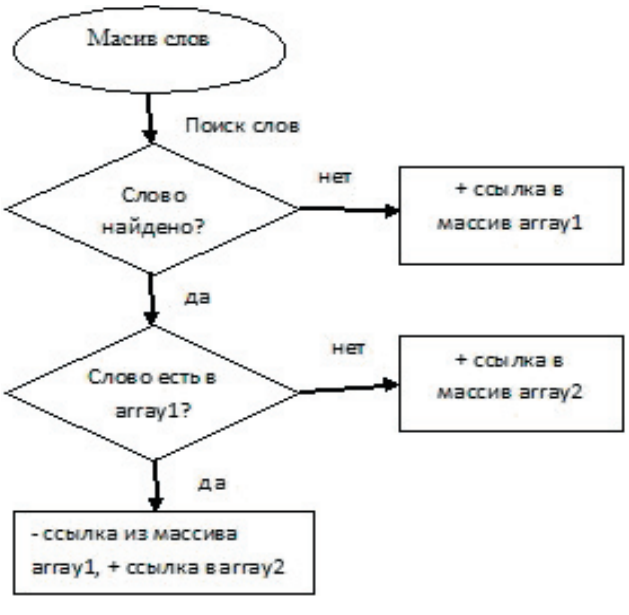

Рис. 1. Алгоритм работы LRU-k

Вычисление веса фрагмента производится также как и для алгоритма Freq, но вместо суммы по наиболее часто встречающимся словам, к весу, вычисленному по базовому алгоритму, прибавляется количество слов из массива array2, встретившихся в анализируемом фрагменте.

Для автореферирования необходимо определять семантическую близость между предложениями текста [7].

Алгоритм семантического анализа может решить эту задачу. Назовем этот метод семантической матрицей.

Каждому предложению текста соответствует определенная семантическая мера, это частота вхождения слова в иерархию классификатора умноженная на глубину иерархии. Потом необходимо определить семантическую связность пар предложений текста по формуле (2):

$$Scoup = Mcom - Msent \quad (2),$$

где Mcom - совокупная мера двух предложений, Msent - мера каждого предложения. Получаем наборы семантической связности предложений, из которых потом формируем матрицу. В связи с особенностью алгоритма, матрица симметрична, на главной диагонали получаем нули, так как связность предложения с самим собой в данный момент не нужна при анализе. Элементы матрицы выстраиваются по принципу того, что пара предложений с наибольшей связностью – это первая часть элементов аннотации. Следующая пара предложений проверяется на семантическую связность с группами, которые имеют большую связность, и добавляются к той или ино группе, связность с которой наибольшая, либо образуют независимую группу, если связность отсутствует. Как результат получается несколько фрагментов с семантической связностью. В целом алгоритм реферирования таков:

1) морфологический анализ;

3) устранение омонимии слов в предложении;

4) построение семантической матрицы;

5) выделение семантически связных групп предложений.

Этот алгоритм, при реализации сможет пользователю выполнять такие функции как задание степень сжатия, порог семантической связности, количество групп рефератов одного текста.

4. Вывод

Учет при формировании аннотации кроме слов запроса других слов документа позволяет значительно увеличить качество формирования аннотаций с точки зрения пользователя, а алгоритм LRU-K позволяет оценивать важность слов не хуже, чем при использовании классического подхода, основанного на частоте слов, при этом имеет значительно более высокое быстродействие.

Добавление семантического анализа решит проблему связности текста и границы перехода.

Таким образом, совмещение нескольких алгоритмов позволит улучшить качество рефрирования, ведь каждый из них решает определенную задачу, характерную только этому алгоритму.

Литература

1. http://www.pc-freeware.com/soft/ML-Annotator-1.0.
2. http://www.textar.ru/index2.html.
3. www.analyst.ru.
4. http://www.cs.bilkent.edu.tr/~guvenir/courses/cs550/Workshop/Yasin_Uzun.pdf.
5. http://ru.wikipedia.org/wiki/Алгоритмы_кэширования.
6. http://www.intuit.ru/department/sa/compilersdev/9/.
7. Никитиа С. Е. Семантический анализ языка науки: На материале лингвистики , Книжный дом ”ЛИБРОКОМ” 2010 - №2 с.146.