АЛГОРИТМ УНИВЕРСАЛЬНОЙ ДЕСЯТИЧНОЙ КЛАССИФИКАЦИИ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Третьяков Ф. И.

Серебряная Л. В. – к-т. техн. наук, доцент

Универсальная десятичная классификация (УДК) — удобрый способ каталогизировать научную текстовую информацию. Данная работа показывает, как данная задача может быть решена автоматизированна и представлен алгоритм универсальной десятичной классификации.

На сегодняшний день существует большое количество неупорядоченной текстовой информации. Поэтому поиск и классификация необходимой информации по ключевому слову является одной из важнейших задач. Особенно остро проблема стоит в сфере науки, потому как исследователю часто приходится изучить множество научных работ, прежде чем найти что-то важное для себя. Иногда можно, только взглянув на работу, определить ее тематику, а бывает, что приходится прочитать большую часть текста, чтобы понять его смысл.

С помощью УДК выполняется классификация информации, необходимая во всем мире для систематизации произведений науки, литературы и искусства, периодической печати, различных видов документов и организации картотек [1]. В настоящее время УДК назначается вручную на основе специальных справочников библиотекарями или специально обученным персоналом. Данная работа посвящена методам и средствам, позволяющим автоматически присваивать работе УДК, не привлекая к этому человека. Поэтому цель работы можно определить как автоматизация универсальной десятичной классификации [2].

Поставленная задача сводится к тому, что для каждого текста, входящего в множество из n текстов, определить категорию m из УДК.

Рассмотрим алгоритм классификации.

1. Происходит обработка названий всех категорий с помощью модуля, выделив корни слов и поместив результаты в соответствующий словарь. Каждая строка в нем имеет ключ, которым является корень слова, а значение в строке – количество всех словоформ по ключу из названия категории.

2. Выполняется шаг 1 для всех текстов, применив его не к названиям текстов, а к ним самим.

Дерево УДК состоит из 126441 категорий. Это очень много для решения обычной задачи классификации. Любой, да и выбранный алгоритм будет работать чрезвычайно медленно, поэтому тут нужно использовать положительную сторону УДК — иерархию. Имеет смысл проходить не по всем категория, а использовать проход по дереву. Причем, тут есть особенность каждая вершина дерева может быть искомым классом. То есть мы должны учитывать не только листы, а и сами ветви. Поэтому имеет смысл строить алгоритм следующим образом:

3. Выбираем начальный список категорий. Пусть это будут сыновья невидимой родительской категории (самые верхние категории УДК).

4. Для каждого текста находится наиболее подходящая категория. Ее номер определяется значения переменной T, вычисленной по следующей формуле:

$$T = \sum_{\substack{i=0,\\ j=0}}^{\substack{i<n,\\ j<m}} a_i \times b_j ,$$

где n – размер словаря категории, m – размер словаря текста, a_i – слово из словаря категории, b_j – слово из словаря текста.

5. Происходит выбор категории для текста, где T максимально. Если таких T два, три или более, то будем проходить по всем деревьям параллельно и в итоге у нас получится УДК, соединенный через плюс списком категорий.

6. Далее, если данная категория имеет подкатегории, то выбираем уже из них, но к ним прибавляем родительскую категорию, потому как текст может относиться и к ней и переходим к шагу 7. Если же подкатегорий нет, то мы нашли нашу категорию а и завершаем алгоритм.

7. Ищем T для выбранных категорий. Если наибольший T для родительской, то выбираем ее и заканчиваем алгоритм. Если же побеждает дочерняя, то опять переходим к шагу 6.

Результат, получаемый с помощью данной классификации можно улучшить, если будет введено машинное обучение. В итоге отнесения текста по названию к категории, слова, входящие в состав названия делятся на два типа: которые присутствуют в тексте, и которые не присутствуют. И слова, которые не входят в текст по сути тоже являются маркерами данной категории. Значит, их следует как-то помечать как входящие в эту категорию.

Список использованных источников:

1. Толстых, В. К. 1. TextMining. Глубинный анализ текста / В. К. Толстых // Из цикла лекций «Современные Internet-технологии» для студентов 5-го курса кафедры Компьютерных технологий физического факультета Донецкого национального университета. – Донецк, 2011. – 213 с.
2. Браславский, П. Ю. Прикладные задачи информатики / П. Ю. Браславский.– Москва: Вильямс, 2005. – 168 с.