УСОВЕРШЕНСТВОВАННЫЙ МЕТОД ПОРТЕРА ДЛЯ УНИВЕРСАЛЬНОЙ ДЕСЯТИЧНОЙ КЛАСИФИКАЦИИ

Л. В. Серебряная, Ф. И. Третьяков
Кафедра программного обеспечения информационных технологий, Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: Fiodor.Tretyakov@gmail.com, l_silver@mail.ru

Универсальная десятичная классификация - организованная и упорядоченная система хранения научной документации. К ней может быть применима автоматизация. В данной работе автоматизация выполнена с помощью программного средства базирующегося на алгоритме стемминга Портера. В рамках данной работы был предложен усовершенствованный алгоритм Портера для русского языка.

Введение

На сегодняшний день существует большое количество неупорядоченной текстовой информации. Поэтому поиск и классификация необходимой информации по является одной из важнейших задач. В сфере научных исследований проблема стоит таким образом, что 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 иногда приходится прочитать большую часть текста, чтобы понять его значимость для исследователя.

Чтобы сразу было известно, к какой области знаний относится научная работа, была придумана универсальная десятичная классификация (УДК). Она является обязательным атрибутом любой печатной или электронной копии научной работы. С помощью УДК выполняется классификация информации, необходимая во всем мире для систематизации произведений науки и организации картотек [1].

I. Постановка задачи

В настоящее время УДК назначается вручную на основе специальных справочников библиотекарями - специально обученным персоналом. Данная работа посвящена методам и средствам, позволяющим автоматически присваивать работе УДК, не привлекая к этому человеческий фактор. Поэтому цель работы можно определить как автоматизацию универсальной десятичной классификации.

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

Предмет исследования работы – универсальная десятичная классификация текстов.

II. Особенности стемминга

Существует несколько способов решить данную задачу. Прежде всего, выбор метода зависит от количества исходных данных. Если имеется набор текстов-образцов и категорий, то наилучшим выбором будет контролируемое обучение и последующая классификация. Затем необходимо определить решающее правило и разделяющую функцию, с помощью которых будет выполняться классификация текстов. Сами тексты обрабатываются и из них выделяются метрики, которые подставляются в качестве параметра в разделяющую функцию, в результате чего определяется принадлежность текстов к одному из классов [3].

В русском языке лексемы имеют сложные и разнообразные структуры, что существенно затрудняет процедуру классификации текстов. Однокоренные слова могут иметь различные окончания, суффиксы и приставки, которые не должны влиять на результат классификации. Однако при проверке формального совпадения однокоренных лексем и получении отрицательного результата сравнения классификация текстов, построенная на основе неточных результатов сравнения, оказывается неверной. Поэтому для анализа русского языка в качестве разделяющей необходимо выбрать функцию, оперирующую только частью слова и выдающую ответ на его основе. Такой частью может быть корень слова, но как правило, его довольно сложно выделить [3]. Поэтому необходимо найти часть, которую с одной стороны выделить возможно, а с другой она будет способствовать повышению точности поиска [2].

Стемминг – один из способов выделения определенной части слова. Это процесс нахождения основы слова заданной лексемы. Основа не всегда совпадает с морфологическим корнем слова [1].

Для решения задачи классификации используется специальный алгоритм стемминга под названием стеммер [3]. Он может выделять значимую часть слова (стем).

Для выделения корня слова был разработан программный модуль, включающий в себя стеммер, флексер и лемматизатор. Стеммер использует эвристическую модель.

III. Алгоритм стемминга

Для создания эвристического стеммера необходимы словари окончаний, формы причастий и деепричастий, суффиксов и приставок. По данным словарей и будет эвристически определяться часть речи. Суть алгоритма сводится к определению части речи для слова по его окончанию, используя словари окончаний. Порядок определения задается уникальностью окончания данной части речи. К примеру, окончания причастий невозможно спутать ни с чем другим, поэтому, стемминг начинается именно с них.

Стемминг будет проходить по следующему алгоритму.

1. Происходит поиск окончаний причастий и деепричастий в слове. Если оно найдено, то удаляется и выполняется переход к шагу 3.
2. Осуществляется поиск окончаний прилагательных, глаголов или существительных. Если они найдены, то удаляются.
3. Если слово оканчивается на «и», оно удаляется.
4. С начала слова в нем ищется последовательность: гласная-согласная. Все буквы после этого сочетания будут блоком n. Если ее нет или блок n пустой, переход к шагу 7.
5. Ищется в блоке n блок m. Это блок, который следует после конструкции гласная-согласная. Если его нет, или он пустой, переход к шагу 7.
6. Ищутся в блоке m части слова «ост» и «ость». Если они найдены, то удаляются.
7. Если слово имеет окончание «ейш» или «ейше», то оно удаляется.
8. Если на конце слова найдено удвоенное «н», второе «н» удаляется.
9. Если слово имеет окончание «ейш» или «ейше», то оно удаляется.
10. Если на конце слова «ь», он удаляется.

Стеммер позволяет сделать аналитическую обработку текстов на русском языке более релевантным. Минусами является сложность модуля и пониженная точность.

IV. Алгоритм классификации

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.

Рассмотрим алгоритм классификации на основе созданного стеммера.

1. Происходит обработка описаний всех категорий УДК с помощью модуля, путем выделения стемов и размещением результатов в словарь категории. Каждая строка в нем имеет ключ, которым является стем, а значение в строке – количество всех словоформ по ключу из описания категории.
2. Выполняется шаг 1 для всех текстов, применив его не к названиям текстов, а к ним самим.
3. Выбираем категории первого уровня вложенности из дерева УДК.
4. Для каждого текста находится наиболее подходящая из выбранных категорий. Ее номер определяется значением переменной T, вычисленной по следующей формуле:

$$T = \sum_{i=0,j=0}^{n,m} a_i \times b_j (1)$$

5. Происходит выбор категории для текста, где T максимально. Если таких $T > 1$, то необходим проход по деревьям категорий с получением УДК, соединенным через плюс.
6. Если выбранная 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.

В итоге мы получается систему, которая позволяет присвоить УДК тексту с высокой скоростью, точностью и автоматизированно.

1. V. Tolstoy, *Deep analysis of the text. From the series of lectures "Modern Internet-technologies"for students of the 5th grade of the Department of Computer Technology Faculty of Physics*, Donetsk, Ukraine: Department of CS, 2005.
2. A. Prutskov, *Algorithmic Provision of a Universal Method for Word-Form Generation and Recognition* // Automatic Documentation and Mathematical Linguistics, 2011, Vol. 45, No. 5, pp. 232-238.
3. F. Tretyakov and L. Serebryanaya, *Russian texts' classification method*, Minsk, Belarus: International scientific and technical conference dedicated to the 50th anniversary of the MRTI-BSUIR sourcebook, 2014.