DOI: 10.34220/MAMSP_179-184
УДК 004.9

РАЗРАБОТКА МАТЕМАТИЧЕСКОЙ МОДЕЛИ ДЛЯ НЕЙРОННОЙ СЕТИ РАСПОЗНАВАНИЯ ТЕКСТОВОЙ ИНФОРМАЦИИ

М.О. Чайка[1], И.А. Бунеев[1], В.А. Величко[1]

[1]ФГБОУ ВО «Воронежский государственный лесотехнический университет имени Г.Ф. Морозова»

В статье рассматривается математическая модель системы, которая обеспечивает распознавание изображений, представляющих собой текст или использующий аналогичную информацию в процессе генерации.

Ключевые слова: компоненты связности, нейронные сети, обучающие системы, распознавание, сегментация.

DEVELOPMENT OF A MATHEMATICAL MODEL FOR A NEURAL NETWORK FOR RECOGNIZING TEXT INFORMATION

M.O. Chaika[1], I.A. Buneev[1], V.A. Velichko[1]

[1]Voronezh State University of Forestry and Technologies named after G.F. Morozov

The article considers a mathematical model of a system that provides recognition of images that represent text or use similar information in the generation process.

Keywords. connectivity components, neural networks, learning systems, recognition, segmentation.

Оценка в текстовой категоризации часто проводится путем сравнения с заранее назначенными категориями в тестовых коллекциях, созданных для этой задачи. Оценка обычно исключает более глубокое рассмотрение контекстов, таких как реальные задачи конечного пользователя и информационные практики. Наиболее серьезной проблемой в оценках категоризации текста является отсутствие стандартных наборов данных, и показывает, как различные версии даже одной и той же коллекции оказывают сильное влияние на производительность. Это соответствует устоявшемуся знанию из исследований межиндексной

согласованности, что индексация человека очень непоследовательна и что непоследовательность является неотъемлемой чертой индексации, а не спорадической аномалией.

Входными данными является группа документов, которую требуется разделить по степени тематической близости этих текстов друг к другу.

Выходными данными будет матрица тематической близости, которая содержит в себе коэффициенты близости каждого текста относительно каждого другого текста.

Таким образом, перед нами стоит задача модифицировать алгоритм k ближайших соседей так, чтобы повысить качество выделения тематики текстов существенно не увеличивая сложность работы и не уменьшая ее скорость.

Правильное и достаточное автоматизированное представление темы текста должно включать не только ключевые слова, но и контекст этих слов, поскольку значение любого слова определяется в контексте тех слов, которые используются с ним, близких к тексту. Ключевые слова сами по себе изолированы от контекста и неадекватно отражают тематическую направленность людей. Существующие психолингвистические исследования [2, 3, 7] подтверждают этот тезис.

Добавление контекста к ключевому слову определяется следующими практическими соображениями: возможно ли, что характеристики распределения строк смогут повысить пороговое значение проведения тестирования и ухудшить качество анализа.

В качестве примера выделим слово в падеже. В этом случае возникает неопределенность в определении порогового значения, если частота одного слова превышает значительно остальные слова. Здесь можно предложить установить пороговое значение при выборе слова, либо его понизить. В первом случае будем иметь то, что текст представлен одним словом, а во втором – темой станут все слова. Чтобы избежать такие ситуации добавим ключевое слово в контекст. Это будет самым приемлемым в рассматриваемой проблеме.

Первоначально необходимо провести индексацию слов в каждом тексте отдельно. Для этого рассмотрим все слова текста как поток данных $d_i \in D$. Для удобства далее в этой части будем обозначать текст как d.

$$d = \{w_1, w_2, \dots, w_\mathrm{n}\},$$

где w_1 – информационный элемент, соответствующий конкретному слову из текста, а n = |d| – количество слов в тексте.

Множество всех уникальных для d слов обозначим как W:

$$W = \{w_1, w_2, \dots, w_{\mathrm{m}}\},$$

где m – количество уникальных слов в тексте d.

Вес каждого информационного элемента следует подсчитать так чтобы он, наилучшим образом характеризовал конкретный текст и был менее характерен для других текстов и тем более текстов, относящихся к другим тематическим кластерам. Для этого используется формула TF-IDF.

Вначале следует подсчитать количество повторений каждого $w_1 \in \mathrm{W}$ в потоке данных d и рассчитывается частота данного термина в тексте по формуле:

$$TF = \frac{\mathrm{n_w}}{|\mathrm{d}|},$$

где $\mathrm{n_w}$ – количество использования выбранного слова w в документе; $\mathrm{n_d}$ – количество слов.

Таким образом, мы рассчитаем повторения каждого уникального слова в тексте. Затем вычисляется обратная частота документа IDF. Она используется для уменьшения веса общеупотребительных слов, которые не имеют непосредственного отношения к тематике конкретного текста.

$$IDF = log_{10}\left(\frac{|D|}{D_w}\right),$$

где |D| – это количество текстов, а D_w – это количество тех текстов, где встречается слово w.

Итоговый весовой коэффициент каждого w будет рассчитываться по формуле:

$$m_i = TF * IDF$$

Далее необходимо провести сортировку массива $\mathrm{W} = \{w_1, w_2, \dots, w_{\mathrm{n}}\}$ в порядке убывания весового коэффициента каждого слова w_i.

Таким образом, тематику конкретного текста в рамках задачи будет определять набор ключевых слов и их весовых коэффициентов

$$T = \{m_1 w_1, m_2 w_2, \dots, m_i w_i\}, i = n = |W|.$$

Каждый элемент в множестве T является информационным признаком конкретного текста. Для того что бы сократить пространство признаков в расчет принимаются только те элементы, весовой коэффициент которых выше определенного порогового значения. Возникает проблема при определении порогового значения, отделяющего все слова от ключевого. На выбор порогового значения оказывает существенное влияние рассматриваемый текст. Например, такие его свойства как $\mathrm{m(G(W,E))_{Max}}$, $\mathrm{m(G(W,E))_{Min}}$ и n(W).

Обучение системы является альтернативой ручному вводу описаний объектов и параметров функции, решающей поставленную задачу.

Обучение системы – процедура самонастройки распознавания на основе поступающей информации и возможна в процессе подготовки системы, а также в процессе выполнения по мере «накопления опыта». Этот вид обучения относится к типу обучения с учителем. Обучающей выборкой является набор объектов называемых образцами. Параметры решающей функции корректируются в процессе сравнения результатов. Таким образом, мы получаем априорную информацию о множестве объектов, распознаваемых в процессе решения задачи [6]:

$$T = I_0(Ω_1, \dots, Ω_i) .$$

Описания, предложенные учителем с указанием их принадлежности к классу, можно представить как совокупность описаний (1):

$$\Big(I(\omega_1), \dots, I(\omega_{r_2}), \quad I(\omega_{r_1+1}), \dots, I(\omega_{r_2}), \dots, I(\omega_{r_j})\Big),$$

где объекты $\omega_1, \dots, \omega_{r_2} \in Ω_1$, объекты $\omega_{r_1+1}, \dots, \omega_{r_2} \in Ω_2$ и т.д.

Обучающая выборка есть матрица, строки которой содержат названия объектов $\omega_i, i = 1, \dots, r_j$, а столбцы — названия признаков $x_j, j = 1, \dots, N$ (см. рисунок 1). Элементы таблицы –признаки объектов $x_j(\omega_i)$. Строки таблицы группируются по классам $Ω_j, j = 1, \dots, m$

	x_1	x_2	...	x_N	Классы
ω_1	$x_1(\omega_1)$	$x_2(\omega_1)$	...	$x_N(\omega_1)$	Ω_1
ω_2	$x_1(\omega_2)$	$x_2(\omega_2)$	...	$x_N(\omega_2)$	
...	...	...	...	...	
ω_{r_1}	$x_1(\omega_{r_1})$	$x_2(\omega_{r_1})$	...	$x_N\omega_{r_1}$	
...	...	...	...	...	...
$\omega_{r_{m-1}+1}$	$x_1(\omega_{r_{m-1}+1})$	$x_2(\omega_{r_{m-1}+1})$	...	$x_N(\omega_{r_{m-1}+1})$	Ω_m
...	...	...	...	...	
ω_{r_m}	$x_1(\omega_{r_m})$	$x_2(\omega_{r_m})$	...	$x_N(\omega_{r_m})$	

Рисунок 1 – Обучающая выборка

Пары векторов $\{x^s, d^s\}, s = 1 \dots S$, где $\{x^s\} = \{x^1, \dots, x^s\}$ – входные вектора x, $\{d^s\} = \{d^1, \dots, d^s\}$ – эталоны выходных векторов [5]. Пары $\{x^s, d^s\}$ – обучающее множество.

Для обучения сети необходимо чтобы число элементов S было достаточным для формирования набора параметров, которые выдадут требуемое отображение. Число пар в обучающем множестве не определено, но их не должно быть слишком мало или много.

Для анализа эффективности работы данных алгоритмов и методов было принято решение разработать реализацию на языке Python и на нескольких практических упрощенных примерах протестировать и сравнить работу алгоритмов. Язык программирования Python был выбран в связи со скоростью его работы, наличием большого количества готовых решений для организации работы с текстами, объектной ориентацией данного языка и удобством использования.

Основными функциями программы являются:

Texting – Функция, которая отвечает за первичную обработку каждого документа, считывание его из памяти, выделение отдельных слов, подсчет их весовых коэффициентов и ранжирование выборки по мере убывания частоты использования каждого слова.

FindADocs – Вспомогательная функция, основной задачей которой является поиск и подсчет количества документов, в которых встречается определенное слово.

TFIDF – Функция для вычисления окончательного весового коэффициента каждого слова по формуле TF-IDF. Она определяет частоту использования каждого термина и умножает ее на обратную частоту распространенности этого термина во всей выборке документов.

TwoTextAnalize – Функция определения тематической близости для двух текстов. Она производит вычисление тематической близости каждого слова первого текста этому же слову во втором документе (если это слово имеется) и суммирует результаты получая коэффициент для целого текста.

GetSurraundlings – Функция, выделяющая контекст для каждого слова и сохраняющая его отдельно.

Общая структура программы представлена на рисунке 2.

В дальнейшем есть возможность использовать полученный алгоритм, с учетом незначительной адаптации, в библиотечных системах, поисковых системах и более сложных алгоритмах, основанных на использовании нейронных сетей

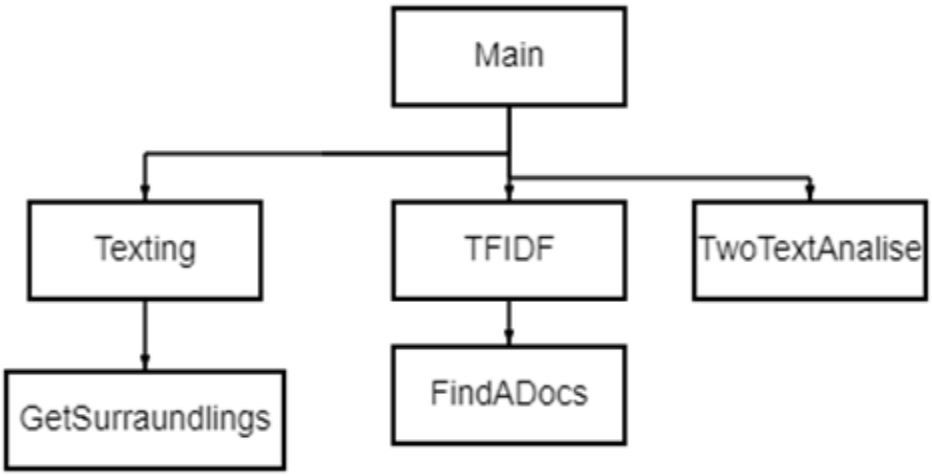

Рисунок 2 – Общая структура программы

Список литературы

1. Журавлёв, Ю.И. Распознавание. Классификация. Прогноз / Ю.И. Журавлёв // Математические методы и их применение. – 1989. – Вып. 2. – С. 208–327.

2. Заенцев, И.В. Нейронные сети: Основные модели / И.В. Заенцев. – Воронеж, 1999. – С. 76

3. Оксюта, О.В. Система распознавания дорожных знаков с использованием искусственных нейронных сетей / О.В. Оксюта, А.М. Милютин // Моделирование систем и процессов. – 2017. – Т. 10, № 1. – С. 64-67.

4. Змеев, А.А. Сравнительный анализ архитектур нейронных сетей для использования их на практике / А.А. Змеев, В.В. Лавлинский, С.Н. Яньшин // Моделирование систем и процессов. – 2017. – Т. 10, № 4. – С. 18-26

5. Лавлинский, В.В. Применение математического описания действий для целенаправленных систем на основе методов нейронных сетей / В.В. Лавлинский, С.Н. Яньшин // Моделирование систем и процессов. – 2017. – Т. 10, № 2. – С. 17-23.

6. Лавлинский, В.В. Информационные системы для извлечения данных из неструктурированного текста с использованием онтологий / В.В. Лавлинский, Ю.О. Зольникова // Моделирование систем и процессов. – 2018. – Т. 11, № 3. – С. 30-34.

7. Чубур, Т.А. Дискурсивная объективация вербально-ментальных единиц в русской и английской лингвоконцептосферах в ракурсе закономерностей исторических изменений языка / Т.А. Чубур // Моделирование систем и процессов. – 2017. – Т. 10, № 1. – С. 76-80.