Распознавание, анализ и визуализация интернет-мемов

Германчук М.С., Козлова М.Г.

Крымский федеральный университет им. В. И. Вернадского, Симферополь, Россия

Всероссийская конференция ММРО-2019.
Россия, г. Москва, 26–29 ноября 2019 г.

Мем в соответствии с словарем Merriam-Webster – это «идеи, поведение или стиль, которые распространяются от человека к человеку в культуре». В 2015 интернет-мем стал представлять собой вирусные изображения, которые показывают определённые образы и обычно сопровождаются текстом.

Мем – вирусное изображение, которое состоит из изображения с образом и текста.

Среда распространения мемов – социальная сеть «Вконтакте».

Входные данные – ссылка на мем в СС «Вконтакте» или непосредственно изображение с текстом.

Цель – разработка системы, которая может правильно маркировать интернет-мем и предсказать его распространение.

Для разработки инструментария необходимо решить задачи:

1. Распозавание текста на изображении.
2. Классификация текста.
3. Использование различных метрик сети для корректировки. классификации «интернет-мема».
4. Сбор различных метрик для создания прогнозирующей системы.

ДЛЯ ЗАДАЧИ РАСПОЗНАВАНИЯ ТЕКСТА:

1. Пакет Tesseract
2. Использование OCR собственной разработки

ЭТАПЫ РАСПОЗНАВАНИЯ ТЕКСТА:

1. Анализ контуров. Проверка вложенности контуров и поиск дочерних контуров позволяет обнаруживать и распознавать как чёрный текст на белом фоне, так и наоборот. На этом этапе контура объединяются в строки, а строки в текст. Текстовые строки разбиваются на слова в зависимости от межстрочного интервала.
2. Непосредственно распознавание текста.
    1. Попытка распознать каждое слово по очереди. Каждое слово, которое было распознано классификатором с большим уровнем уверенности передаётся адаптивному классификатору в качестве обучающих данных. После чего адаптивный классификатор получает возможность распознать остальной текст более точно. Адаптивный классификатор может обучиться не сразу, поэтому слова, которые были плохо распознаны в первый раз распознаются повторно.
    2. Решаются проблема нечетких пробелов и делается проверка альтернативных гипотез для некоторых слов, чтобы распознать оставшийся текст.

Tesseract не всегда способен выделить текст, который наложен поверх изображения и показывает неудовлетворительные результаты при классификации небольших фраз или словосочетаний, характерных для «интернет-мема».

ДЛЯ ЗАДАЧИ РАСПОЗНАВАНИЯ ТЕКСТА:

1. Пакет Tesseract
2. Использование OCR собственной разработки

Для решения извлечения текста из «интернет-мема» OCR должна решать две задачи:

1. Обнаружение текста
2. Распознавание текста

Для обнаружения текста на изображении выбран алгоритм EAST (An Efficient and Accurate Scene Text Detector). Алгоритм использует полносвязную свёрточную нейронную сеть, которая принимает решения основываясь на уровне слов и строк, исключая промежуточные шаги. По сравнению с другими алгоритмами данный алгоритм выделяется высокой точностью и небольшим временем работы.

Ключевым компонентом предлагаемого алгоритма является модель нейронной сети, которая обучена непосредственному прогнозированию существования текстовых экземпляров и их геометрии из полных изображений. Модель представляет собой полносвязную сверточную нейронную сеть, адаптированную для обнаружения текста, которая выводит для каждого пикселя предсказания слов или текстовых строк. Это исключает промежуточные этапы, такие как предложение кандидата, формирование текстовой области и разбиение слов. Этапы последующей обработки включают только пороговое значение и порог для прогнозируемых геометрических фигур. Детектор называется EAST, поскольку он является методом обнаружения текста с эффективной и точной текстурой.

Распознавание текста:

1. Обработка изображения. Имеется изображение, большую часть которого составляет текст. Задача: выделить только текст и избавиться от шумов. Проблема: цвет текста неизвестен и может содержать несколько оттенков одного цвета. Поэтому на данном этапе изображение кластеризируется. После чего центроидом самого большого кластера и будет цвет текста в полученном изображении. Строится маска, которая полностью отделяет текст от остального изображения.
2. Выделение контуров букв, распознавание строк текста и объединение букв в слова. Для выделенных контуров находятся центры масс и строятся основные строки, на которых лежит текст. Это позволяет определять правильную последовательность обработки символов и избавляет от шумов, которые могли остаться после обработки изображения. Далее, основываясь на интервалах между буквами, контура объединяются в слова.
3. Непосредственное распознавание символов. Для этого использовалась свёрточная нейронная сеть, обученная на 47653 изображениях русских букв различных шрифтов. Точность распознавания нейросети достигает 98.22% на валидационной выборке и 98.8% на тестовой.

а)

б)

в)

г)

д)

Рис.: (а) оригинальная картинка-мем; этапы распознавания текста: (б) извлечение изображения с текстом; (в) выделение цвета текста и построение маски; (г) нахождение центров масс контуров; (д) построение линий текста.

Псевдокод программы распознавания текста:

img; ▷ <*S* — это изображение, полученное на вход>
k = число кластеров;
threshold = порог межсимвольного расстояния;
Кластеризируем изображение *img* на *k* кластеров;
Находим наибольший кластер *C*;
Строим маску для *img* по цвету *C*;
Выделяем контура букв *contours*.
    lines = []
for <*cnt* in *contours* > do
    if *Moment* of *cnt* not in *lines* then
        добавляет *cnt* на линию;
for <*line* in *lines* > do
    meandist = среднее расстояние между контурами;
    for <*i* := 0 to *len*(*line*) − 1 > do
        if dist(*cnt*[*i*], *cnt*[*i* + 1]) > *meandist* + x_t*hreshold* then
            отделяет *cnt*[*i*] и *cnt*[*i* + 1] пробелом;
    result := ' ';
for <*line* in *lines* > do
    for <*i* := 0 to *len*(*line*) > do
        result += Классификация *cnt*[*i*];
show *result*;

В рамках данной задачи была собрана выборка, состоящая из 63 политических мемов и 44 неполитических. В качестве дополнительных объектов использовался текст комментариев из социальной сети «Вконтакте», содержащий как политический, так и не политический контекст. В общей сложности для обучения использовалось 168 предложений. Для валидации использовалась выборка, состоящая из 11 политических мемов и 7 не политических мемов.

Перед обучением классификатора данные были нормированы стеммером и преобразованы в векторный вид.

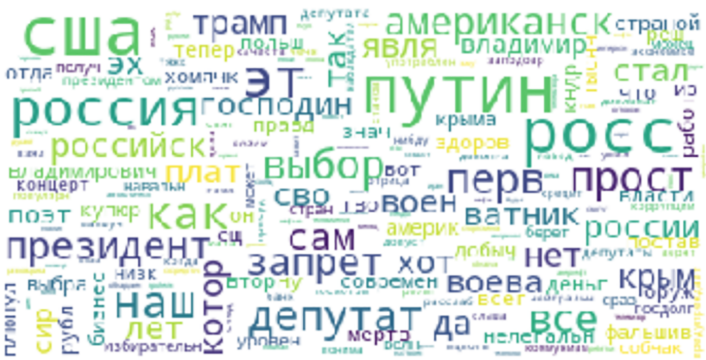
Учитывая небольшой размер тренировочных данных алгоритмом классификации был выбран метод опорных векторов. Метод опорных векторов – алгоритм обучения с учителем, использующийся для задач классификации и регрессионного анализа. Основная идея алгоритма заключается в том, чтобы найти гиперплоскость в N-мерном пространстве (N – количество признаков), которая четко классифицирует данные. Чтобы разделить два класса данных, существует множество возможных гиперплоскостей, которые можно было бы выбрать. Задача алгоритма – найти плоскость, которая имеет максимальный отступ т. е. максимальное расстояние между точками данных обоих классов. Максимизация отступа дает уверенность, так что будущие данные могут быть классифицированы с большей уверенностью.

Гиперпланы – это границы принятия решений, которые помогают классифицировать данные. Точки данных, находящиеся с обеих сторон гиперплоскости, можно отнести к разным классам. Кроме того, размер гиперплоскости зависит от количества функций. Если количество входных функций равно 2, то гиперплоскость – это линия. Если число входных функций равно 3, то гиперплоскость становится двумерной плоскостью. Опорные векторы – это точки данных, которые ближе к гиперплоскости и влияют на положение и ориентацию гиперплоскости. Используя эти векторы поддержки, максимизируется отступ классификатора. Алгоритм *SVM* стремится максимизировать разницу между точками данных и гиперплоскостью.

Для классов «политический мем» и «неполитический мем» построены облака слов:

- к классу «политические мемы» относились изображения, содержащие названия стран, регионов, имена политических лидеров;

- класс «неполитические мемы» не обладал чётко выделяющейся группой слов, так как относящиеся к этому классу предложения имели разную тематику.



а) б)

Рис.: Облака слов: (а) политические мемы; (б) неполитические мемы.

На тестовой выборке классификатор показал точность в 81.25%. По мере увеличения тренировочных данных будет расти и точность классификатора.

Заключение

1. Рассмотрены задачи по разработке инструментария обработки и анализа "интернет-мемов".
2. Предложены методы решения двух задач: извлечение текста из изображения и классификация текста.
3. Разработан и предложен собственный алгоритм распознавания текста на изображении.