Отчет научно исследовательской работы за 4 семестр Пахомова Александра Сергеевича МИВТ18, научный руководитель Леверьев Владимир Семенович старший преподаватель.

Тема: «Разработка мобильного приложения для распознавания якутского текста с бумажных носителей»

**Создание собственного словаря с помощью Tesseract OCR**

Для решения задачи распознавания образов был выбран OCR модуль для Python: Tesseract.Python-tesseract это библиотека с открыты исходным кодом для OCR поддерживаемая Google, которая дает высокие результаты точности и поддерживает более 100 языков.

Для определения слова на конкретном языке Tesseract использует языковые модели и словари. Языковая модель содержит в себе значения параметров модели нейронной сети и другие данные обучения. Например, языковая модель для русского языка хранится в файле rus.traineddata. Пользователь может создать свой список слов для Tesseract так, чтобы Tesseract мог научиться их распознавать.

Tesseract позволяет расширять стандартный словарь для любого поддерживаемого языка добавлением личных текстов или научить языковую модель всецело сменив текста обычного словаря собственными текстами.

Tesseract пользуется со специальными файлами типа .dawg для всевозможных категорий слов в словаре. К примеру, файл .word-dawg применяется для ведущих текстов словаря, а файл freq-dawg — для более нередко встречающихся текстов.

**Изменение языковой модели Tesseract**

Этим методом, возможно пользоваться для версий Tesseract 3.0.5 и 4.0.0. Единственное различие Tesseract 4.0.0 от более ранней 3.0.х в том, собственно что в версии 4 в базе Tesseract лежит модель LSTM и файлы словаря dawg имеют расширение lstmdawg (в версии v3.0.5 они имеют расширение -dawg). К примере на файле с наиболее часто встречающихся слов теперь имеет расширение lstm-freq-dawg вместо freq-dawg, а файл unicharset получил расширение lstm-unicharset (ранее .unicharset).

Для начала устанавливается библиотека Tesseract OCR. Я пользуюсь ОС Ubuntu и Tesseract v4. Установка Tesseract с поддержкой пакета apt:

sudo apt update && sudo apt install tesseract-ocr

Не считая самой библиотеки Tesseract данная команда еще установит все нужные инструменты для обучения языковой модели (training tools).

Для начала расширяется языковая модель собственными текстами. К примеру расширим модель для русского языка. Организуем обычный текстовый файл с заглавием font\_sakha и добавим в него текста, которые мы желаем прибавить в слова модели, по одному с новой строки.

Вслед за тем переходим в папку tessdata. Необходимы права root для выполнения операций в данной системной папке.

sudo su

После того как мы дали права суперпользователя скопируем файл font\_sakha в эту папку. Распакуем файл языковой модели rus.traineddata в папку traineddat\_backup

combine\_tessdata -u rus.traineddata traineddat\_backup/rus.

Эта команда извлекает все файлы, необходимые для компиляции языковой модели в папку traineddat\_backup.

Теперь создадим файл rus.lstm-word-dawg из нашего файла font\_sakha с помощью утилиты wordlist2dawg.

wordlist2dawg font\_sakha eng.lstm-word-dawg traineddat\_backup/eng.lstm-unicharset

и скомпилируем новый файл языковой модели rus.traineddata.

combine\_tessdata -o rus.traineddata rus.lstm-word-dawg

Получается файл языковой модели rus.traineddata из наших собственных слов словаря.

Теперь обучим языковую модель rus полностью заменив слова стандартного словаря своими словами. Сначала нам нужно сделать бэкап всех файлов dawg (.lstm-word-dawg, .lstm-freq-dawg итд), находящихся в папке traineddat\_backup, поскольку мы заменим их новыми. Просто создадим папку tmp и перенесем в нее все файлы dawg.

**Заключение**

По итогам четвертого семестра научно-исследовательской работы магистранта была проведена работа по создание собственного словаря (Якутского) с помощью Tesseract OCR.