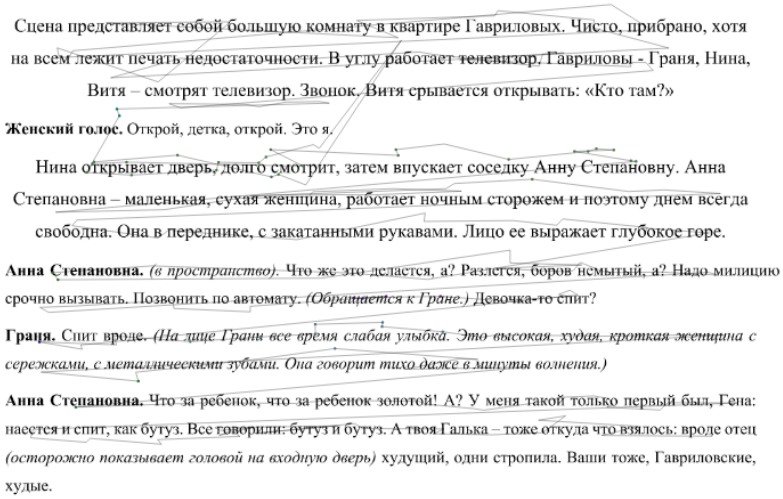
Программная реализация MultiCOR

С.Д. Макеев

Пред-обработка данных

Представим алгоритм предварительной обработки данных при загрузке их в корпус окулографических реакций MultiCOR. Глазодвигательные данные извлекаются из системы BeGaze, предназначенной для работы с окулографическими устройствами фирмы SensoMotoric Instruments (SMI).

Получение траектории взгляда


Сцена представляет собой большую комнату в квартире Гавриловых. Чисто, прибрано, хотя на всем лежит печать недостаточности. В углу работает телевизор. Гавриловы - Граня, Нина, Витя – смотрят телевизор. Звонок. Витя срывается открывать: «Кто там?»

Женский голос. Открой, детка, открой. Это я.

Нина открывает дверь, долго смотрит, затем впускает соседку Анну Степановну. Анна Степановна – маленькая, сухая женщина, работает ночным сторожем и поэтому днем всегда свободна. Она в переднике, с закатанными рукавами. Лицо ее выражает глубокое горе.

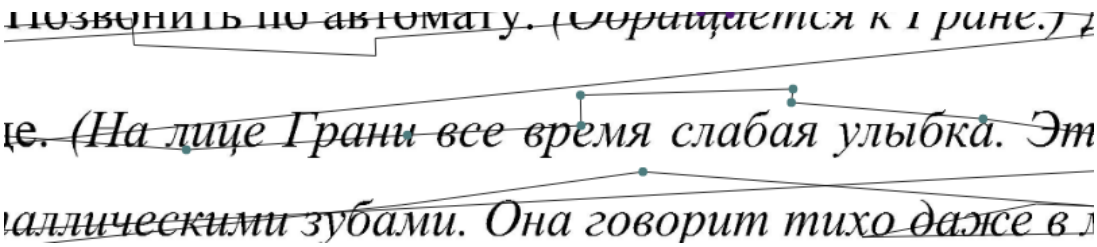
Анна Степановна. *(в пространство).* Что же это деется, а? Разлется, боров немытый, а? Надо милицию срочно вызывать. Позвонить по автомату. *(Обращается к Гране.)* Девочка-то спит?

Граня. Спит вроде. *(На лице Грани все время слабая улыбка. Это высокая, худая, кроткая женщина с сережками, с металлическими зубами. Она говорит тихо даже в минуты волнения.)*

Анна Степановна. Что за ребенок, что за ребенок золотой! А? У меня такой только первый был, Гена: наестся и спит, как бутуз. Все говорили: бутуз и бутуз. А твоя Галька – тоже откуда что взялось: вроде отец *(осторожно показывает головой на входную дверь)* худущий, одни стропила. Ваши тоже, Гавриловские, худые.

На первом этапе анализируется файл системы BeGaze, содержащий список всех событий, происходивших в течение некоторого эксперимента. Из этого списка отфильтровываются события, соответствующие фиксациям взгляда. Таким образом формируются траектории взгляда испытуемых при прочтении каждой стимульной страницы.

Обнаружение попаданий в область интереса


позвонить по автомату. (Обращается к Гране.)

е. (На лице Грани все время слабая улыбка. Эт

аллическими зубами. Она говорит тихо даже в

Из имеющихся фиксаций выделяются те, которые попадают в размеченные зоны интереса. Данные о границах зон интереса берутся напрямую из файлов системы BeGaze.

Выделение участков траектории

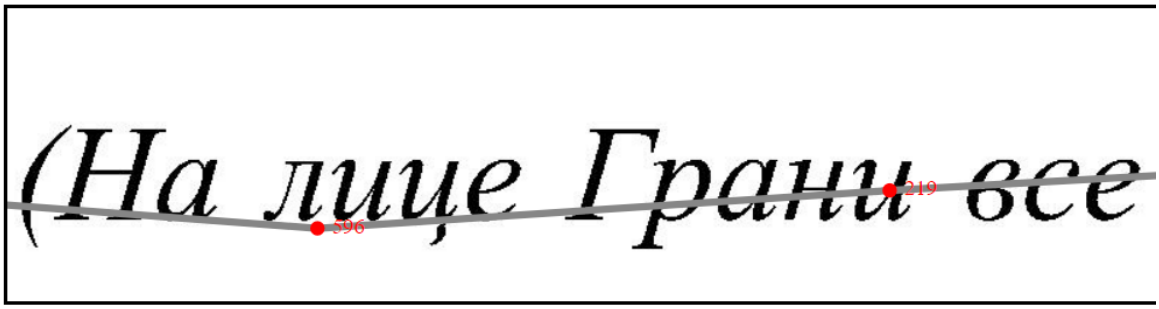
На основе данных о попаданиях в зоны интереса, выделяются участки траектории, которые будут считаться «прочтениями» соответствующих зон. Цель данного этапа — исключить из рассмотрения случайные попадания в эту зону.

На рисунке приведён пример такого выделения. Обведённая цветом линия — это участок траектории, признанный «прочтением» данной фразы. Одинокая цветная точка внизу — случайное попадание, которое мы не признаём частью прочтения.

Отделение «настоящих» попаданий от случайных — процесс не точный. Так или иначе, требуется принимать некоторые эвристические решения. Программа MultiCOR использует следующую эвристику:

1. Найти самое первое и самое последнее попадание. Далее будет рассматриваться участок траектории от первого до последнего попадания.
2. Вычислить «медианное» попадание (т.е. момент, до и после которого находится по половине всех попаданий).
3. Если в начале рассматриваемого участка есть фрагмент, в котором попаданий оказывается меньше, чем не-попаданий, «отрезать» этот фрагмент и убрать его из рассмотрения. Повторять этот процесс, пока такие фрагменты не закончатся, или до достижения «медианного» попадания.
4. Аналогично пункту 3, произвести «отрезание» фрагментов, но теперь с конца.
5. Участок, оставшийся не отрезанным, и будет искомым «прочтением».

Вычисление характеристик



Количество фиксаций: 7
Длительность первой фиксации: 596
Самая длительная фиксация: 596

Из полученного участка траектории вычисляются ключевые глазодвигательные характеристики: количество фиксаций, максимум и минимум длительности фиксаций (в миллисекундах) и другие. Эти характеристики будут ключами для поиска нужных пользователю прочтений в корпусе.

Внутреннее устройство базы данных

В действующей базе данных MultiCOR существуют четыре типа сущностей: страницы, зоны, реакции и признаки.

Страница — содержит название стимульной страницы, эксперимента, в котором она участвовала, и растровое изображение страницы в формате JPG.
Зона — описывает зону интереса на странице. Содержит название зоны (взятое из BeGaze), ссылку на страницу, на которой она находится, и координаты её расположения, в виде прямоугольника или ломаной.
Реакция — описывает конкретное прочтение некоторой зоны. Содержит ссылку на саму зону, нужный участок траектории (список координат и длительностей фиксаций) и заранее посчитанные глазодвигательные характеристики — числа, по которым можно искать.
Признак — описывает факт наличия у данной зоны некоторого лингвистического или когнитивного признака, взятого из разметки. Каждая запись в хранилище признаков содержит одну пару: «(номер страницы; номер признака)». Таким образом, пользователь может как получить список признаков данной зоны, так и произвести поиск зон, обладающих данным признаком.

Форма поиска

На странице поиска корпуса MultiCOR можно задавать как глазодвигательные, так и лингвистические/когнитивные параметры, а можно и те и другие сразу.

Глазодвигательные параметры — это просто числа: количество фиксаций в штуках и их длительности в миллисекундах. Пользователь может вводить максимум и минимум по каждому параметру. Все поля необязательные: можно искать «всё, что больше X», или «всё, что меньше Y», или «от X до Y», или вообще не задействовать данный параметр.

Лингвистические/когнитивные признаки представлены в виде категорий, в каждой из которых можно отметить галочкой один или несколько пунктов. Если в категории есть хотя бы одна отметка, то это интерпретируется как поисковых запрос: «показать ТОЛЬКО те зоны, у которых есть ХОТЯ БЫ ОДИН из отмеченных признаков». Если есть отметки сразу в нескольких категориях, то выдаются только те зоны, которые отвечают сразу ВСЕМ запросам по категориям. Категории, в которых нет ни одной отметки, просто игнорируются.

Проще говоря: между пунктами внутри категории ставится знак ИЛИ, а между категориями — знак И.

Пример: если в форме поиска отмечены пункты «жирный шрифт», «курсив», «точка», «восклицательный или вопросительный знак» и «имя прилагательное», это следует интерпретировать как:

Показать только те зоны, в которых
(текст выделен жирным шрифтом ИЛИ курсивом)
И
(есть точка ИЛИ восклицательный/вопросительный знак)
И
(есть хотя бы одно прилагательное)

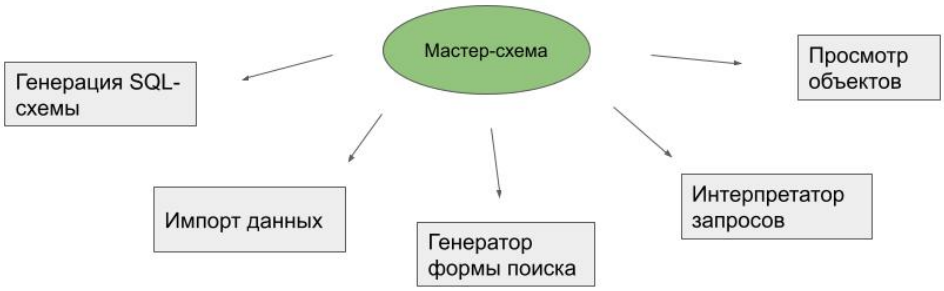
Структура программного продукта

Программа, реализующая создание базы данных MultiCOR и поиск по ней, написана на языке программирования Lua и использует СУБД SQLite. Пользователь взаимодействует с веб-сервером, который поддерживает следующие действия:

- отобразить форму поиска
- произвести поиск по запросу и выдать список результатов
- показать подробную информацию о данном прочтении, включая визуализацию траектории взгляда
- показать подробную информацию о данной зоне интереса, включая её изображение и ссылки на все её прочтения

Отдельно существует модуль, осуществляющий ранее описанную пред-обработку данных и запись их в базу.

Программа обращается с данными по принципам декларативного программирования. Информация о том, какие глазодвигательные и лингвистические данные есть в корпусе, хранится в специальном списке, называемом «мастер-схемой». В этой схеме по каждому признаку хранится сразу всё: его название и описание, его местоположение на форме поиска, его условное обозначение в закодированных поисковых запросах, и способ его хранения в базе данных.



Каждый модуль программы пользуется мастер-схемой, получая из неё ту часть информации, которая нужна этому модулю. Поскольку мастер-схема является единым «источником истины» для всех модулей программы, предотвращаются разногласия между модулями. Например, процедура, интерпретирующая поисковый запрос и запускающая сам процесс поиска, точно знает, как будет выглядеть запрос, поскольку пользуется теми же данными, что и форма поиска, из которой этот запрос пришёл. Она также знает, как соотносятся части поискового запроса с полями базы данных, причём процедура, которая формировала эти поля, тоже пользовалась той же мастер-схемой. Таким образом, все части программы могут полагаться друг на друга. Если требуется что-то изменить в организации хранения данных и поиска, достаточно внести изменение в мастер-схему, и это изменения отразится сразу во всех модулях.