Добрый день, меня зовут Антон и тема моей курсовой работы “Об одной задаче поиска по сходству” Начнем с небольшого введения

Слайд 3

**Поиск по сходству**— это поиск информации, при котором выполняется сопоставление информации заданному образцу поиска или близкому к этому образцу значению. Алгоритмы нечеткого поиска используются в большинстве современных поисковых систем.

Эти алгоритмы характеризуются *метрикой* — функцией расстояния между двумя словами, позволяющей оценить степень их сходства в данном контексте.

Слайд 4

Этот метод был придуман довольно давно, и является наиболее широко используемым, так как его реализация крайне проста, и он обеспечивает достаточно хорошую производительность.

Алгоритм основывается на принципе: «Если слово **А** совпадает со словом **Б** с учетом нескольких ошибок, то с большой долей вероятности у них будет хотя бы одна общая подстрока длины *N*». Эти подстроки длины *N* и называются *N*-граммами.

Но так же у него есть свои минусы. Данная проблема происходит, например, когда пользователь ошибся в одной букве в корне слова при вводе, тогда соответствующие триграммы корня не будут найдены в словаре и процент нахождения правильно исправленного слова заметно снизится.

Между тем, метод N-грамм оставляет полный простор для использования собственных метрик с произвольными свойствами и сложностью, но за это приходится платить — при наличии словарей больших размеров данный метод будет долго выполняться и требовать больших ресурсов и мощностей.

Слайд 5

Наиболее часто используемыми на практике являются триграммы — подстроки длины 3. Выбор большего значения *N* ведет к ограничению на минимальную длину слова, при которой возможно обнаружение ошибок. При поиске похожей фразы, ищутся одинаковые триграммы. Чем больше равных триграмм, тем больше фраза схожа с исходной.

**Формула, для расчета метрики**: *P* = (*a* + *b*) / (2 \* *m*), где

* *m* – число совпадающих триграмм
* *a* – общее число триграмм в первой строке
* *b* – общее число триграмм во второй строке

Слайд 6

Далее идет парочка примеров. Как вы видите в консоли отображается исходное слово, которое мы взяли для примера и совершили в нем намеренно 2 ошибки. Программа считала его из окна приложения и разложила на триграммы. Далее в ней происходят вычисления, скрытые от наших глаз.

Открывается txt словарь русского языка, в котором слова по порядку разбиваются на триграммы и записываются в массив для сравнения с исходным словом. Если слово прошло проверку, то оно со своей триграммой и *метрикой* попадает в консоль, а затем и в окно приложения. Для меньшего затрата времени выписываются только первые 5 слов удовлетворяющие метрики. На втором примере показано, что программа умеет менять верхний регистр на нижний для успешного сравнения слов.

На этом мой доклад подошел к концу. Спасибо за внимание!!!