**ЛАБОРАТОРНАЯ РАБОТА №4.**

Цель работы – овладеть техникой использование N-grams для распознавания и классификации текстов.

N-Grams

*N-gram* - це N-character срез более длинной строки. *N-gram* дополняется символами “\_” для индикации начало (конца) слова.

Например, слово “TEXT” включает следующие N-grams:

bi-grams: \_T, TE, EX, XT, T\_

tri-grams: \_TE, TEX, EXT, XT\_, T\_ \_

quad-grams: \_TEX, TEXT, EXT\_, XT\_ \_, T\_ \_ \_

В общем случае, строка длиной *k*, дополнена символами подчеркивания, содержит *k* +1 bi-grams, *k*+1 tri-grams, *k*+1 quad-grams і т.д..

*Генерация профилей (Profiles) частот N-Gram.*

Процедура “Generate Profile” на рис.1 считывает входной текст и подсчитывает количество вхождений всех N-grams. Для этого необходимо:

Разбить текст на слова,которые содержат только буквы и апострофы. Дополнить их символами подчеркивания в начале и конце слова.

Для каждого слова сгенерировать N-grams для N от 2 до 5 (включая символы подчеркивания).

Разместить N-grams слова в хэш-таблице увеличивая количество вхождений.

Таблица должна быть отсортирована в порядке убывания числа вхождений.

*Процедура класификации.*

Класификация (рис. 1) включает следующие этапы.

1. Обучение (левая ветка алгоритма на рис. 1).

Выбрать тексты,которые представляют собой образцы категорий.

Сгенерировать профили частот N-grams для каждого из текстов.

2. Распознавание.

Сгенерировать профиль частот N-grams текста для распознавания.

Посчитать расстояния (distance measure) между профилем частот текста для распознавания и профилями частот образцов категорий.

Выбрать категорию с найменьшим значением расстояния.



Figure 1. Dataflow For N-Gram-Based Text Categorization

Значения расстояний вычисляются по схеме *Out-Of-Place* (рис. 2).



Figure 2. Calculating The Out-Of-Place Measure Between Two Profiles

Величина *max* зависит от длины профилей частот і подбирается экспериментально.

Задание на работу.

Спроектировать класс TextСlassifier. Интерфейс класса TextСlassifier может выглядить, например,таким образом

Рекомендованные параметры

//score for no match in frequency map

#define MAX\_SCORE 150

#define LOWER\_NGRAM\_LEN 2

#define UPPER\_NGRAM\_LEN 5

//Длина профилей категорий й текстов,которые распознаются,должна быть одинаковой і ограниченной. Например,

#define CATEGORY\_LEN 1000

Длина категорий й текстов,которые распознаются,от 5 до 10 Kb.

 /\* N-grams frequencies profile \*/

 typedef vector<string,unsigned long> frequency\_map;

 class TextСlassifier {

 public:

 /\* Name of category defaults to 'Unknown.' \*/

 TextСlassifier();

/\*Param classification - name of the category (e.g., "English"). \*/

 TextСlassifier(const string category);

/\* returns the map N-grams and their frequencies. \*/

 frequency\_map freqs() { return \_freqs; }

/\* returns the name of the classifier.\*/

 string category() { return category; }

/\*Param classification - name of the category.\*/

 void setCategory(string& cat) {category = cat;}

/\*learns the frequencies of N-grams in a corpus.\*/

 void TextСlassifier::learn(istream& in);

/\*dumps the frequencies of N-grams in a corpus.

File structure is as follows:

\* lines which start with c - comments

\* line which starts with s - classification

\* lines with N-grams start with n and contain

\* N-gram and its quantity separated with space or tab\*/

 void TextСlassifier::dumpProfile(ostream& out);

/\*learns the frequencies of N-grams from a file created by.

dumpProfile() function \*/

 void TextСlassifier::learnfromDump(istream& in);

/\*returns distance measure between 2 TextСlassifiers\*/

 long int TextСlassifier::score(const TextСlassifier &);

 . . .

private:

 /\*! internal frequency map \*/

 frequency\_map \_freqs;

 /\*! internal name of classifer \*/

 string \_category;

 };

Распознавание выполнять на примере текстов на языках English, Italian, French і Spain.

Результаты работы оформлять в виде проекта С++-програми,который включает следующие файлы:

1. Файл *объявления* класа TextСlassifier.h.

2. Файл *определения* класа TextСlassifier.cpp.

3. Файл тест-драйву, например, <Test.cpp>.

Тестирование проводить на текстах длиной от 5K до 10K bytes.