Данные файлы предназначены для морфологического разбора слов.

Mtlemm.dll это базовая библиотека для морфологического разбора. Написана на С. Содержит одну функцию

int MtLemmFunc1 (byte \*buff, byte \*datapath, TMorfAnswer \*TMorfAnswerPtr);

buff – слово, которое необходимо разобрать

datapath – путь к Database

TMorfAnswerPtr – структура для хранения морфологического разбора

Возвращает :

< 0 если не найден путь к Database

0 если слово не найдено

Иначе количество найденных значений(например слово косой имеет больше одного значения)

Database файлы, содержащие служебную информацию. Без них ничего работать не будет. Пытаться разобрать их бесполезно, нужно просто смириться и кидать этот каталог в папку с проектом.

Tmorf.h содержит описание структур, необходимых для хранения данных морфологического анализа, например TMorfAnswer;

Все последующие файлы не обязательны, и служат для удобства работы

MorfologLibrary.dll обертка, облегчающая импорт функциональности в Managed код ( например C#)

Содержит две функции:

TMorfAnswerMarshal\* MorfologStruct(char\* word, char\* path,int& state) – возвращает промежуточную, легко конвертируемую в безопасный код структуру. При этом сохраняется только реальная информация, а вся системная удаляется.

char\* Morfolog(char\* word, char\* path) – возвращает результат анализа, собранный в строку по следующему шаблону:

слово|Часть речи|Число|Род|Падеж|Вид|Наклонение|Лицо|Время||;

слово|Часть речи|Число|Род|Падеж|Вид|Наклонение|Лицо|Время||;

и т.д.

Ещё сильнее упрощает конвертацию в безопасный код, но в разобранной строке может присутствовать мусор.

Варианты использования:

Unmanaged (С,C++)

Managed (C#)

Unmanaged (С,C++)

Чтобы использовать функционал на С,С++ необходимо:

1) Инициализировать буфер для результата

TMorfAnswer\* TMorfAnswerPtr = new TMorfAnswer[100];

2) Получить Handle библиотеки

LPCWSTR name = L"mtlemm.dll";

HINSTANCE hLib = LoadLibrary(name);

3) Получить указатель на функцию

DWORD(CALLBACK\* pfnMtLemmFunc1)(byte \*buff, byte \*datapath, struct TMorfAnswer \*TMorfAnswerPtr);

(FARPROC&)pfnMtLemmFunc1 = GetProcAddress(hLib, "MtLemmFunc1");

4)вызвать функцию

int res = pfnMtLemmFunc1((byte\*)word, (byte\*)path, &TMorfAnswerPtr[0]);

5)в конце освободить ресурсы

FreeLibrary(hLib);

Managed (C#)

Для использования кода на С# кроме всего прочего необходимо добавить к проекту файл Morph.cs

Все классы лежат в пространстве имен Lingua

Содержит в себе следующие классы:

class Morph

/// Место, где хранится папка с базой данных

public string Path

public virtual List<List<string>> MorphAnalys(string word); возвращает двумерный массив

словоформа1 [характеристика1][ характеристика2][ характеристика3]

словоформа2 [характеристика1][ характеристика2][ характеристика3]

Является оберткой, которая использует функцию Morfolog

class FormatedMorph : Morph

public override List<List<string>> MorphAnalys(string word); тоже что и у базового класса, только результат проходит через фильтр, удаляющий мусор и пустую информацию

class StructuredMorph

/// Место, где хранится папка с базой данных

public string Path

public virtual List<TMorfAnswer> MorphAnalys(string word);

возвращает список заполненных структур, каждая из которых характеризует словоформу.

Является оберткой над MorfologStruct. В отличие от ранее рассмотренных возвращает не список строк, а упорядоченную структуру, что упрощает процесс извлечения информации.

//структура для Маршалинга из unmanaged кода

[StructLayout(LayoutKind.Sequential)]

unsafe struct TMorfAnswerMarshal

{

public byte\* MainForm;

public Class Class;

public Animation Animation;

public Gender Gender;

public Number Number;

public Case Case;

public AlternativeForm Alternative;

public Aspect Aspect;

public Person Person;

public Voice Voice;

public Reflection Reflection;

public Time Time;

public Mood Mood;

public ComparativeDegree ComparativeDegree;

};

Эта структура предназначена для развертывания указателя возвращаемого MorfologStruct. Является низкоуровневой не безопасной структурой.

//безопасная версия TMorfAnswerMarshal

struct TMorfAnswer

{

public string MainForm;

public Class Class;

public Animation Animation;

public Gender Gender;

public Number Number;

public Case Case;

public AlternativeForm Alternative;

public Aspect Aspect;

public Person Person;

public Voice Voice;

public Reflection Reflection;

public Time Time;

public Mood Mood;

public ComparativeDegree ComparativeDegree;

};

Механизм использования каждого класса один:

Lingua.StructuredMorph m = new Lingua.StructuredMorph();

var list = m.MorphAnalys("вороной");

Также возможно изменить путь расположения Database, указав в свойстве Path новый путь