БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра информационных систем управления

ЧЕРНЯК ЕЛЕНА АЛЕКСАНДРОВНА

АВТОМАТИЧЕСКАЯ ИДЕНТИФИКАЦИЯ ЯЗЫКА ТЕКСТОГО ДОКУМЕНТА

Курсовая работа

Студентки 4 курса 2 группы

|  |  |
| --- | --- |
| “Допустить к защите“  **Зав. кафедрой** ИСУ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  “\_\_\_” \_\_\_\_\_\_\_\_\_\_\_ 2016 г. | **Руководитель**  Железко Борис Александрович,  к.т.н. |

**Минск 2016**

**Аннотация**

В курсовой работе рассмотрена проблема автоматической идентификации языка текстового документа. На основе анализа литературы изучаются методы определения языка текстовых документов. Разработаны модели и алгоритм решения данной проблемы.

**Анатацыя**

У курсавой рабоце разгледжана праблема аўтаматычнай ідэнтыфікацыі мовы тэкставага дакумента. На аснове аналізу літаратуры вывучаюцца метады вызначэння мовы тэкставых дакументаў. Распрацаваны мадэлі і алгарытмы вырашэння дадзенай праблемы.

**Annotation**

The coursework deals with the problem of automatic identification of a text document language. Based on the literature review, examines methods of determining the language of text documents. Models and algorithms developed in this work will help us to solve this problem.

**Реферат**

Курсовая работа, 22 страницы, 7 рисунков, 8 источников.

АВТОМАТИЧЕСКАЯ ИДЕНТИФИКЦИЯ ЯЗЫКА ТЕКСТОГО ДОКУМЕНТА

Объектом исследования являются системы определения языка текстовых документов.

Цель работы: разработка алгоритма и программного обеспечения для автоматической идентификации языка текстового документа.

Методология проведения работы: системный подход, открытые системы, инженерия знаний.

Результаты работы: модели, алгоритмы, программный инструментарий и методика его применения.

Область применения результатов: обработка текстовых документов.

При оформлении курсового проекта использовались стандарты ВАК АН Республики Беларусь.

**СОДЕРЖАНИЕ**

|  |  |  |
| --- | --- | --- |
|  | **ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ И ТЕРМИНОВ** | 4 |
|  | **ВВЕДЕНИЕ** | 5 |
|  | **ГЛАВА 1. ПРОБЛЕМЫ ИДЕНТИФИКАЦИИ ЯЗЫКА ТЕКСТОВОГО ДОКУМЕНТА** | 6 |
|  | 1.1 Возникновение проблемы идентификации языка текстового документа | 6 |
|  | 1.2 Методы определения языка текстового документа | 6 |
|  | 1.2.1 Метод коротких слов | 7 |
|  | 1.2.2 Метод частотных слов | 7 |
|  | 1.2.3 Метод N-грамм | 7 |
|  | 1.2.4 Статистический метод | 8 |
|  | 1.2.5 Алфавитный метод | 8 |
|  | 1.2.6 Метод грамматических слов | 8 |
|  | 1.3 Постановка задачи | 8 |
|  | 1.4 Выводы | 9 |
|  | **ГЛАВА 2. МОДЕЛИ И АЛГОРИТМЫ** | 10 |
|  | 2.1 Алгоритм N-грамм | 10 |
|  | 2.1.1 Предварительный разбор текстового документа | 10 |
|  | 2.1.2 Подстройка весовых коэффициентов | 11 |
|  | 2.2 Возможные недостатки выбранного алгоритма | 13 |
|  | 2.3 Выводы | 13 |
|  | **ГЛАВА 3. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ** | 14 |
|  | 3.1 Архитектура ПО | 14 |
|  | 3.2 Обоснование выбора программной платформы и среды разработки | 14 |
|  | 3.3 Разработка программного обеспечения | 15 |
|  | 3.4 Выводы | 15 |
|  | **ГЛАВА 4. ЭКСПЕРИМЕНТАЛЬНАЯ ПРОВЕРКА СИСТЕМЫ** | 17 |
|  | 4.1 Пример работы программы | 17 |
|  | 4.2 Анализ полученных результатов | 20 |
|  | 4.3 Выводы | 20 |
|  | **ЗАКЛЮЧЕНИЕ** | 21 |
|  | Список использованных источников | 22 |

# ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ И ТЕРМИНОВ

**ПО** – программное обеспечение

**ЕЯ** – естественный язык

**ПОД** – поисковый образ документа

**ПОЯ** – поисковый образ языка

**ВВЕДЕНИЕ**

В информационных системах различного типа, предназначенных для обработки в автоматическом режиме больших объемов текстов на естественных языках(ЕЯ), актуальны различные задачи распознавания текстовой информации. Требование автоматизации процессов обработки текстовой информации придают особую важность проблеме определения ЕЯ, на котором написан текст или часть текста.

Для информационных систем предприятий, систем документооборота особую важность приобретает задача определения в автоматическом режиме языка коротких текстовых сообщений размеров, характерных для сообщений электронной почты и аналогичных коммуникационных сервисов. В настоящее время известны достаточно точные методы распознавания языка длинных текстов, содержащих десятки слов и предложений. Широко популярны модели определения языка текста, использующие частоты буквосочетаний.

Возможно также использовать ранговые методы для идентификации языка текста, но они не применимы для коротких текстов. Проблема определения языка коротких сегментов текста остается актуальной, а достижение более высокой точности идентификации осуществлялось за счет более крупных моделей и медленной скорости классификации.

В данной курсовой работе рассматривается проблема автоматического определения языка текстового документа.

В первой главе была показана актуальность проблемы автоматической идентификации языка текстового документа, рассмотрены возможные методы её решения, поставлена задача разработки ПО для решения задачи языковой идентификации текста.

Во второй главе был рассмотрен алгоритм для решения задачи автоматического определения языка текста, выявлены его предположительные недостатки, которые могут быть подтверждены или опровергнуты только экспериментально.

В третьей главе была выбрана программная платформа и среда разработки, разработана архитектура и описаны этапы построения ПО.

В четвертой главе рассмотрена работа программы для автоматической идентификации языка текстового документа.

В результате была решена поставленная задача и показана работоспособность разработанного программного обеспечения.

**Глава 1. ПРОБЛЕМЫ ИДЕНТИФИКАЦИИ ЯЗЫКА ТЕКСТОВОГО ДОКУМЕНТА**

* 1. **Возникновение проблемы языковой идентификации и актуальность её решения**

Проблема автоматической идентификации языка текстовых документов изначально появилась в контексте создания специализированного поискового сервиса и классификатора текстов.

В современном мире, действует целый ряд систем классификации больших объемов текстовой информации, в основе которых лежат технологии компьютерной лингвистики и алгоритмов распознавания образов. В настоящее время существует огромное количество текстовых документов на различных языках.

Например, сегодня в Евросоюзе(ЕС) говорят на 23 официальных языках, среди которых английский, голландский, греческий, датский, испанский, итальянский, немецкий, португальский, финский, французский и шведский. Официальные тексты должны публиковаться на всех упомянутых языках. Все решения, принимаемые официальными органами ЕС, переводятся на все официальные языки, и граждане ЕС вправе обращаться в органы ЕС и получать быстрый ответ на свои запросы на любом из официальных языков.

Задача определения языка текстового документа является очень распространенной. Подобная функциональность существует в текстовых редакторах, в переключателе клавиатуры PuntoSwitcher, в системах машинного перевода и, разумеется, в системах поиска информации.

Возможность языковой идентификации была реализована в виде веб-сервиса в Google Language API, но она выполняется удаленно и имеет существенные ограничения: процедура опознания языка выполняется с существенной задержкой и асинхронна по своей природе.

Идентификация языка является важной стадией работы с текстовыми документами, так как от нее зависит эффективность приложений по обработке естественных языков (NLP): информационный поиск (IR), вопросно-ответные системы (QA), автоматическое реферирование (Summarization).

Таким образом, задача автоматической идентификации языка текстовых документов является весьма актуальной.

* 1. **Методы определения языка текстового документа**

В зависимости от применяемых правил построения поисковых образов и стратегий их сравнения различают следующие основные методы определения языка текстовых документов:

* коротких слов,
* частотных слов,
* N-грамм,
* статистический,
* алфавитный,
* грамматических слов,
* неграмматических слов.

### Метод коротких слов

Метод коротких слов при построении ПОЯ использует слова определенной длины, не превышающей заданный порог. Например, в работе [2] на основании корпуса текстов, содер­жащего документы общим объемом один миллион символов для каждого идентифицируемого языка, извлекались лексемы длиной до пяти символов, встретившиеся в тексте более трех раз. Вероятность появления в тексте i-й лексемы рассчитывалась как отношение ее частоты к общей сумме частот всех лексем из полученного набора. Предложение входного документа разбива­лось на лексемы, и лексемам, присутствующим в ПОЯ, назначались их частоты, а отсутствую­щим - некоторая минимальная частота. Вероятность принадлежности предложения языку рас­считывалась как произведение вероятностей его лексем.

### Метод частотных слов

Согласно методу частотных слов, ПОЯ представляет собой набор слов, обладающих наи­большей частотой встречаемости в сравнении с остальными словами документов из обучающе­го корпуса. Количество слов в ПОЯ, необходимых для определения языка, у разных авторов отличается на порядок; например, в [3] использовалась тысяча слов, а в работе [4] - всего сто. Вероятность принадлежности входного документа языку определяется на основании пословно­го сравнения с ПОЯ с учетом рассчитанных вероятностей его элементов.

### Метод N-грамм

Как и в методах, описанных выше, ПОЯ по методу N-грамм [2] строится на основа­нии тренировочного корпуса текстов. Составляющими ПОЯ являются N-граммы - последова­тельности символов строки текста переменной или строго фиксированной длины. На­пример, в [2] в ПОЯ включались триграммы, встречавшиеся в текстах тренировочного корпуса более ста раз, и их количество изменялось от 2550 до 3560 в зависимости от языка. Вероятность отнесения документа к какому-либо языку рассчитывалась подобно методу коротких слов.

**1.2.4 Статический метод**

Статический метод заключается в следующем: на основании тренировочного корпуса текстов для каждого языка определялось рас­пределение вероятностей униграмм и биграмм, образующих поисковые образы языков, или триграмм в случае экспериментов с отсканированными документами. Далее для входного до­кумента также строилось распределение вероятностей составляющих его N-грамм. Используя метрику Кульбака - Лейблера [5], вычислялась относительная энтропия между вероятностны­ми распределениями входного документа и всеми ПОЯ. Тексту назначался язык с минимальной относительной энтропией.

### Алфавитный метод

Алфавитный метод заключается в определении языка на основании обнаружен­ных в анализируемом тексте характерных диакритических знаков - специальных значков, до­бавляемых к буквам того или иного алфавита с целью обозначить изменение их стандартного чтения или же указать на какую-либо особую роль, которую звук, обозначенный буквой с диа­критикой, играет в слове.

### Метод грамматических слов

Метод грамматических слов состоит в поиске слов, обладающих малым лексическим значением, но необходимых для выражения грамматических и других отношений в предложе­нии, которые характерны для рассматриваемого языка. Этими словами являются предлоги, союзы, артикли и т. д. [1]

## 1.3 Постановка задачи

Для нашего приложения необходим алгоритм автоматической идентифи­кации языка текстового документа, работающий быстро и не требующей боль­ших словарей для своей работы, а также позволяющий легко добавлять новые идентифицируемые языки.

Как показал проведенный анализ, указанные ранее методы достаточно эффективны только при определенных ограничениях на размер текста входного текстового документа и сравнительные характеристики тех естественных языков, на множестве которых решается задача.

На основании этого, был выбран алгоритм N-грамм.

Задача состоит в следующем:

* более детально изучить алгоритм N-грамм;
* разработать приложение, реализующее алгоритм N-грамм;
* проанализировать полученные результаты.

## 1.4 Выводы

В первой главе были получены следующие основные результаты:

* проведен анализ предметной области;
* проанализированы основные существующие методы

определения языка текстового документа;

* дана постановка задачи;
* сформулированы требования к решению поставленной задачи.

**Глава 2. МОДЕЛИ И АЛГОРИТМЫ**

## 2.1 Алгоритм N-грамм

В работе будет предложен метод для опреде­ления языка и кодировки документа по его содержимо­му, основываясь на статистиках документов, для которых язык и кодировка известны заранее. Подсчитыва­ются частоты N-грамм (сочетаний символов или под­строк, длиной не более N) и предполагается, что при­мерно 300 самых часто используемых N-грамм сильно зависят от языка.

Алгоритм заключается в нахождении частот N-грамм для всех тестовых документов, для которых из­вестен язык, а также для каждого документа, язык кото­рого пытаемся определить. После этого среди всех тес­товых документов находим тот, для которого расстоя­ние от его N-граммной статистики до статистики тести­руемого документа минимально. После этого языком тестируемого документа считается язык найденного тестового документа.

Расстояние между статистиками подсчитывается следующим образом: все N-граммы сортируются в по­рядке убывания частоты их появления, затем для каж­дой N-граммы вычисляется разница её позиций в отсор­тированном списке N-грамм тестового и тестируемого документов. Расстояние между статистиками определя­ется как сумма разностей позиций каждой N-граммы:

,

где позиции i-й N-граммы в тестовом и тестируемом документах соответственно. Значение N предлагается использовать равным 3 (включительно).

### 2.1.1 Предварительный разбор текстового документа

На этапе предварительного разбора производит­ся разбор корпуса документов и выявление в нем:

1. Уникальных для языка N-грамм с длиной до 3-х символов включительно (включая и одиночные символы).
2. Часто встречающихся (но не уникальных по всему корпусу) в языке N-грамм с длиной до 3-х симво­лов включительно.

Из тонкостей разбора можно отметить следую­щие моменты:

1. Тексты предварительно разбиваются на слова (мо­нолитные последовательности из символов, соот­ветствующих данному языку).
2. Все слова приводятся к одному регистру (напри­мер, к верхнему).
3. Сочетания кодируются следующим образом – если мощность N-граммы меньше, чем 3, то отсутст­вующие символы заменяются на символ с кодом 0.
4. К словам слева и справа добавляется по пробелу (или любой другой символ, гарантировано не яв­ляющийся буквой из какого-то алфавита). Это по­зволяет в дальнейшем отличить сочетание с мощ­ностью до 2-х включительно, стоящее в конце, от такового же, стоящего в середине и в конце слова. Такая мелочь позволила увеличить точность распо­знавания языка примерно в 1.5 раза и более.
5. Уникальные N-граммы для того, чтобы попасть в окончательный список N-грамм, должны превы­сить определенный порог встречаемости (на прак­тике использовался уровень в 10-15 раз). Это по­зволяет убрать случайные сочетания, нехарактер­ные для данного языка.
6. Часто встречающиеся сочетания сортировались по частоте, и выбиралось определенное их количество - Nfreq .Эта процедура выполнялась отдельно для сочетаний разной длины. Т.е., при Nfreq =16, в ко­нечную таблицу сочетаний попадало 16 триграмм, 16 би­грамм. Для одиночных символов сделано исключе­ние - для них количество выбирается как меньшее значение из Nfreq, либо 1/8 от размера алфавита (учитывается количество символов одного регист­ра). Эксперименты показали, что оптимальное зна­чение Nfreq (для разных текстов и языковых набо­ров), лежит, как правило, в промежутке 64.. .128.

### 2.1.2 Подстройка весовых коэффициентов

Для выявления языка документа, к нему приме­нялся разбор, аналогичный тому, который велся на предварительном этапе, только теперь никакие слова не отфильтровывались, а выявляемые N-граммы сравнива­лись с таблицей (в практической реализации использо­валась структура, известная как хэш-таблица) N-грамм, построенной на предварительном этапе. Если находи­лось соответствие, то весовой коэффициент языка, ко­торому принадлежала табличная N-грамма увеличивал­ся по следующему правилу.

Если найденная N-грамма являлась уникальной по отношению ко всем остальным языкам, то весовой коэффициент увеличивался на WU. В текущей реализа­ции WU=10. Если N-грамма просто являлась часто встречающейся в соответствующем языке, то весовой коэффициент увеличивался на aN. В текущей реализа­ции было принято, что aN=N. Данный выбор весовых коэффициентов является эмпирическим, и на практике он показал себя надежно.

Было произведено исследование 3-х вариантов алгоритма оценки текстов:

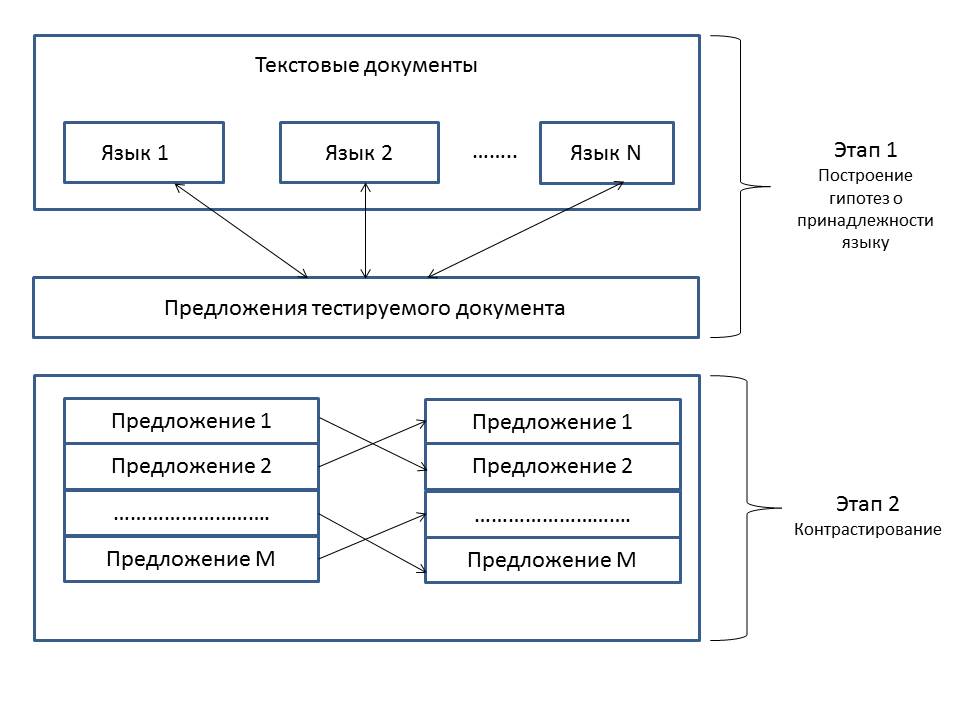
1. Учитываются все найденные сочетания - как уни­кальные, так и неуникальные.
2. Учитываются все уникальные сочетания. Неуни­кальные учитываются только в том случае, если в других языках такие сочетания не прошли в часто встречающиеся (т.е., остались за пределами, кото­рые очерчиваются параметром Nfreq).
3. Учитываются только уникальные сочетания.

Эксперименты показали, что результаты 1-го ал­горитма, учитывающего все, как уникальные, так и час­то встречающиеся сочетания, самые неубедительные.

Лучше всего проявляет себя алгоритм 3, который работает только с уникальными сочетаниями. Од­нако у него имеется один изъян - из-за очень большой требовательности к используемым данным, на коротких последовательностях он может просто не успеть вы­брать главный язык. Т.е., если встречается предложение из двух-трех слов, то очень часто все языки имеют ве­совой коэффициент 0. Кроме того, исследование пока­зало, что имеются языки, имеющие мало уникальных сочетаний (например, испанский), что также затрудняет его применение.

В связи с этим наиболее надежной видится сле­дующая схема работы алгоритма распознавания языка - на первом этапе текст обрабатывается по 3-му алгорит­му. Если отношение коэффициентов 1-го и 2-го по зна­чимости языков больше, чем 2, а абсолютные значения весовых коэффициентов выше некоторого порога (на­пример, 30-50), то на этом и останавливаемся. В про­тивном случае, обрабатываем текст повторно, исполь­зуя алгоритм 2. В этом случае, для принятия решения о провале на каком-то языке попадают отдельные слова на другом, их вполне можно считать инвариантами для основного языка.

Несмотря на достаточно надежную работу алго­ритма, его оригинальных характеристик часто бывает недостаточно в случае, когда обрабатывается текст, содержащий части на похожих языках, например, на русском и украинском. Для того, чтобы сделать его работу надежной, алгоритм становится двух проходным (рисунок 2.1) – первый проход выполняется как опи­сано выше, но каждое предложение помечается самым вероятным языком только в случае уверенного распо­знавания (либо детектируется только один язык с весо­вым коэффициентом не ниже определенного порога, либо самый вероятный язык опережает следующего претендента не менее, чем в 2 раза). Оставшиеся же предложения подвергаются обработке на втором прохо­де - так называемой процедуре контрастирования. Она заключается в том, что языком предложения становится неуверенно распознанный язык, если окружающие его принадлежности текста определенному языку, доста­точно будет только превышение весовым коэффициен­том определенного порога. [7]

 **Рисунок 2.1 – Два прохода алгоритма**

**2.2 Возможные недостатки выбранного алгоритма**

## Исходя из доступных источников, основным недостатком описанного выше ал­горитма может оказаться неустойчивость определения языка малых текстовых фрагментов — отдельные предложе­ния на похожих языках могут распознаваться с ошибками.

Следует также отметить, что процесс морфоло­гической обработки является сравнительно трудоемким, сами словари занимают в памяти от единиц до десятков мегабайт на один язык, поэтому перебор даже по десят­ку возможных языков может оказаться весьма медленным.

## 2.3 Выводы

Во второй главе получены следующие основные результаты:

* разработан и описан алгоритм создания приложения для идентификации языка текстового документа;
* выявлены предположительные недостатки выбранного алгоритма, которые могут быть подтверждены или опровергнуты экспериментально;
* осуществлена адаптация алгоритма под предстоящую реализацию.

**Глава 3. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**3.1 Архитектура ПО**

В программировании приложения был использован объектно-ориентированный подход. Приложение разбито на модули согласно функциональности. Имеются следующие агенты:

1. Агент пользовательского интерфейса преобразует команды пользователя и передаёт их соответствующим агентам;
2. Агент загрузки нового профиля языка, позволяет пользователю загрузить необходимый файл при помощи удобного интерфейса навигации по файловой системе устройства;
3. Агент загрузки всех доступных профилей в систему;
4. Агент определения языка текстового документа;
5. Агент работы с командной строкой;

Важной положительной характеристикой данной архитектуры является её адаптивность под современные языки программирования (Java, C#, Objective C, C++).

Все преобразования над текстом будут вынесены каждое в отдельный модуль.

**3.2 Обоснование выбора программной платформы и среды разработки**

В качестве платформы была выбрана Java от Oracle. Эта программная платформа не является специфической для какого-либо одного [процессора](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80) или [операционной системы](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), но механизм выполнения (называемый [виртуальной машиной](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D1%80%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%B0)) и [компилятор](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%82%D0%BE%D1%80) с набором библиотек, которые реализованы для различного [аппаратного обеспечения](https://ru.wikipedia.org/wiki/%D0%90%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) и различных операционных систем, чтобы Java-программы могли работать везде одинаково. Данная платформа широко распространена, обладает отличной документацией и удобством использования предлагаемых средств.

Существует возможность перенести сложные вычисления в динамически подключаемые библиотеки в пакете приложения, однако при этом приложение уже придётся пересобрать для устройств с различной архитектурой. За пользовательский интерфейс всегда отвечает код, написанный на языке Java.

Классической средой разработки на языке Java является среды разработки Eclipse, который мы будем использовать. Критерием выбора данной среды служит отлаженность, подробная документация и большое сообщество разработчиков. В этом состоит преимущество перед другими языками программирования.

Так же нам потребуется библиотека JSON. За счёт своей лаконичности по сравнению с XML, формат JSON может быть более подходящим для сериализации сложных структур. Также будет использована библиотека JUnit для тестирования некоторых классов.

**3.3 Разработка программного обеспечения**

Каждый из агентов архитектуры реализован в форме Java-класса:

* Агент пользовательского интерфейса – LangDetectSimple.java  
  Отвечает за графическое представление программы. А так же содержит в себе исполняющую функцию main. Перед использованием приложения требуется проинициализировать фукцию init. Она загрузит все доступные профили языков в систему. Делается автоматически.
* Агент добавления нового профиля – GenProfile.java  
  Необходим для добавления профиля, т.е. нового языка. Загружает либо текстовый документ, либо с сайта Wikipedia профиль по языку. Wikipedia абстрактные файлы базы данных могут быть извлечены из " Wikipedia Downloads" (http://download.wikimedia.org/). Они образуют '(language code)wiki-(version)-abstract.xml'
* Агент загрузки всех доступных профилей– DetectorFactory.java

Шаблон фабрики для класса Detector.

* Агент определения языка – Detector.java

Класс для определения языка из указанного текста.   
Его экземпляр может быть построен с помощью класса фабрики DetectorFactory. После добавления целевого текста в Detector с помощью методов append(Reader) или append(String), детектор обеспечивает результаты обнаружения язык для целевого текста с помощью detect() или getProbabilities(). detect () метод возвращает одно имя языка, который имеет самую высокую вероятность.   
 getProbabilities () возвращает список нескольких языков и их вероятностей.

* Агент работы с командной строкой - Command.java

Позволяет приложению(будучи собранным в библиотеку) использовать командную строку.  
Реализованы методы добавления нового профиля языка, определения языка текстового документа.

**3.4 Выводы**

В третьей главе были получены следующие основные результаты:

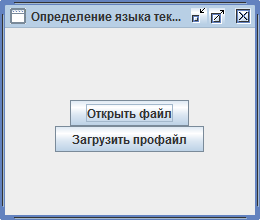
* разработана архитектура ПО;
* выбрана программная платформа и среда разработки, а также обоснован их выбор;
* разработано ПО.

**Глава 4. ЭКСПЕРИМЕНТАЛЬНАЯ ПРОВЕРКА СИСТЕМЫ**

## 4.1 Пример работы программы

Список изначально доступных языков: английский, белорусский, итальянский, латвийский, литовский, немецкий, польский, румынский, русский, турецкий, украинский, французский, эстонский.

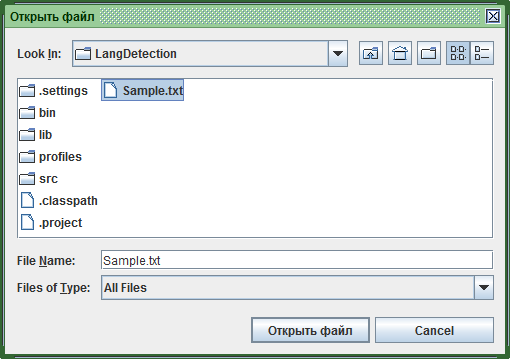
Открывается главное окно приложения (рисунок 4.1)



**Рисунок 4.1 – Главное окно приложения**

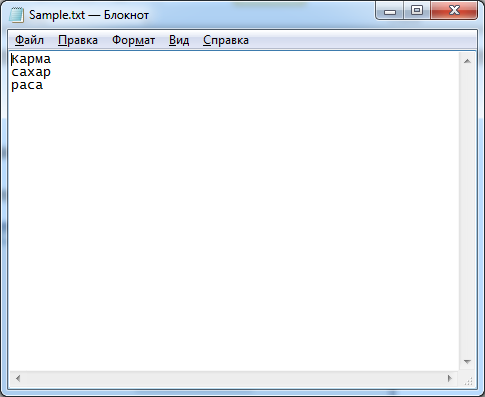
Открываем текстовый документ для языковой идентификации

(рисунок 4.2)

****

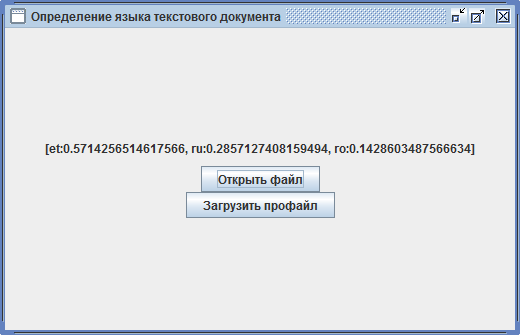
**Рисунок 4.2 – Открытие текстового документа**

Текстовый документ имеет следующий вид (рисунок 4.3):



**Рисунок 4.3 – Содержание текстового документа**

Смотрим результат (рисунок 4.4)

****

**Рисунок 4.4 – Вывод результата**

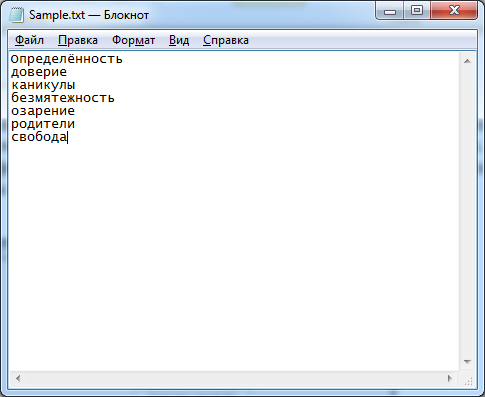
Программа определила, что язык является следующим из возможных с соответствующей вероятность:

эстонский – 57%;

русский – 29%;

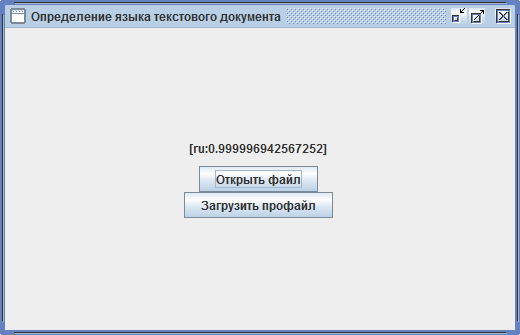
румынский – 14%.

В файле Sample.txt хранилось всего 3 слова русского алфавита. При таком небольшом количестве слов, результат получился неточным. Отредактируем файл Sample.txt до размера семи слов русского алфавита различной длины (рисунок 4.5)



**Рисунок 4.5 – Содержание отредактированного текстового документа**

Вывод результата работы программы (рисунок 4.6):

****

**Рисунок 4.4 – Вывод результата отредактированного документа**

Программа определила, что язык текстового документа является русским с вероятность 99,9%. Точность определения стала почти максимальной.

Таким образом, решена поставленная задача и показана работоспособность разработанного программного обеспечения.

**4.2 Анализ полученных результатов**

Программа распознавания языка и клас­сификации текстов написана на Java. Систе­ма распознает следующие языки: английский, белорусский, итальянский, латвийский, литовский, немецкий, польский, румынский, русский, турецкий, украинский, французский, эстонский. Было протестировано три группы различной длины из 20 доку­ментов. В качестве результатов средняя пол­нота: 86.3%. средняя точность: 89.2%. Ос­новными недостатками описанного алгорит­ма являются: неустойчивость определения языка малых текстовых документов - от­дельные предложения на похожих языках распознаются неуверенно и с ошибками. Кроме того, системы автоматической клас­сификации текстов показывают лучшие вре­менные результаты на 64-битной аппаратной платформе, чем на 32-битной аппаратной платформе, которая работает медленней.

**4.3 Выводы**

В четвертой главе были получены следующие основные результаты:

* продемонстрировано применение разработанного ПО для

автоматической идентификации языка текстового документа;

* проведен анализ полученных результатов.

**ЗАКЛЮЧЕНИЕ**

В результате выполнения курсовой работы были получены следующие основные результаты:

* проведен анализ предметной области;
* показана актуальность проблемы автоматической идентификации языка текстового документа;
* проанализированы основные существующие методы определения языка текстового документа;
* поставлена задача разработки ПО и описан подход к её решению;
* разработан и описан алгоритм;
* разработана архитектура ПО;
* выбрана программная платформа и среда разработки;
* разработано ПО;
* продемонстрировано применение разработанного ПО для решения поставленной задачи.

Стоит отметить универсальность разработанной системы. Достаточно заменить профиль языка, например, на литературный, и мы получим более точную оценку языка текстового документа.

В рассмотренном выше примере были получены высокие значения точности и полноты определения языка. Это объясняется небольшим количеством доступных профилей языков, с определенными в них уникальными и ключевыми частями слов. Приходим к выводу, что эти коэффициенты будут более точные, если рассматривать тексты значительно большего объема, а так же более широкую группу языков.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Крапивин, Ю.Б. Автоматическое определение языка текстового документа для основных европейских языков / Ю.Б. Крапивин / Информа­тика. - 2011.
2. Grefenstette, G. Comparing two language identification schemes / G. Grefenstette // The Third Intern. Conf. on Statistical Analysis of Textual Data. - Rome, 1995.
3. Cowie, J. Language recognition for mono- and multilingual documents / J. Cowie, Y. Ludovic, R. Zacharski // Proc. of the Vextal Conference. - Venice, 1999.
4. Natural Language Identification using Corpus-based Models / C. Souter [et al.] // Hermes Journal of Linguistics. - 1994. - № 13. - P. 183-203.\
5. Kullback - Leibler\_divergence [Electronic resource] // Wikipedia. - Mode of access : <http://en.wikipedia.org/wiki/> Kullback-Leibler\_divergence. - Date of access : 15.12.2010.
6. Cavnar, W. B. and J. M. Trenkle, “N-Gram-Based Text Categorization” In Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, NV, UNLV Publications/Reprographics, pp. 161-175, 11-13 April 1994
7. Сотник С.Л. Идентификация языка UNICODE-текста по N-граммам длиной до 4-х включительно (квадрограммам). Журнал “Математичне моделювання” №1,2(15) 2006, стр. 111-112, Днепродзержинск, издательство ДГТУ.
8. Зайцев В.Г., Лан Чун Линь Проектирование и внедрение системы автоматической классификации документов для конечного множества языков.