А.А. Дунаев

ИССЛЕДОВАТЕЛЬСКАЯ СИСТЕМА ДЛЯ АНАЛИЗА ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

ВВЕДЕНИЕ

Исследования в области автоматической обработки текста (АОТ) и формализации естественных языков, планомерно продвигаясь от самых простых методов анализа к более сложным, постепенно приближаются к такому уровню обработки текста, на котором уже возможно представление текста не просто в виде последовательности слов, а единым целым, обладающим неким смыслом, что уже соответствует человеческому восприятию.

Стремительное увеличение вычислительных мощностей сделало возможным применение трудоёмких лингвистических алгоритмов на больших объемах данных. Но, несмотря на то, что в области формализации естественных языков и систем АОТ, в частности, задействовано большое количество людей и мощностей, работающих в самых разных направлениях, результаты пока довольно скудны, так как ни одна из существующих моделей не может перекрыть структуру языка в целом, а объёмы данных, с которыми имеет дело лингвистика, очень большие.

Такое положение вещей само собой рождает задачу создания системы, удобной для отработки различных решений анализа с целью нахождения наиболее оптимальных и эффективных. Этому способствует то, что как сам анализ, так и программные комплексы, реализующие данные подходы, достаточно легко поддаются фрагментации, т.е. делению на функциональные блоки, выполняющие изолированную функциональность. Исходя из данной специфики проблемной области наиболее естественной задачей является создание модульного программного испытательного стенда, дающего возможность разрабатывать реализации отдельных функциональных блоков, применяя для каждого из них последние достижения в данной области, а затем исследовать их совместную работу путём точной настройки каждого из них в отдельности и гибкой компоновки между собой.

1. ОБЗОР СУЩЕСТВУЮЩИХ СИСТЕМ

В настоящее время лингвистами сформулированы различные теории, позволяющие в какой-то степени формализовать естественный язык. В основном, суть этих теорий сводится к тому, что предложению в тексте сопоставляются различные конечные объекты — графы, или, в общем случае, конечные модели, которые, как принято считать [1, 2], отражают смысл предложений.

1.1. Общие принципы систем обработки текстов

Компоненты, составляющие структуру систем анализа текстов — лингвистические процессоры, которые последовательно обрабатывают входной текст. Вход одного процессора является выходом другого [3, 4, 6].

Выделяются следующие компоненты:

- графематический анализ — выделение слов, цифровых комплексов, формул и т.д.;
- морфологический анализ — построение морфологической интерпретации слов входного текста;
- синтаксический анализ — построение дерева зависимостей всего предложения;
- семантический анализ — построение семантического графа текста.

Для каждого уровня разрабатывается свой язык представления. Язык представления, как полагается, состоит из констант и правила их комбинирования. На графематическом уровне константами являются графематические дескрипторы (ЛЕ — лексема, ЦК — цифровой комплекс и т.д.) На морфологическом уровне — граммемы (рд — родительный падеж, мн — множественное число). На синтаксическом — названия отношений (subj — отношение между подлежащим и сказуемым, circ — обстоятельство). О семантическом анализе будет сказано ниже.

Основой для построения уровней служат результаты работы предыдущих компонентов, но, что важно, последующие компоненты также могут улучшить представление предыдущих. Например, если для какого-то предложения синтаксический анализатор не смог построить полного дерева зависимостей, тогда, возможно, семантический анализатор сможет спроектировать построенный им семантический граф на синтаксис.

1.2. Графематический анализ

Графематический анализ — достаточно простой компонент, выполняющий первые предварительные действия над текстом. На вход компоненту подается текст, на выходе строится графематическая таблица, в которой на каждой строке стоит слово или разделитель из входного текста. Компонент выделяет некоторые аббревиатуры, имена с инициалами, даты и пр. Кроме деления текста на слова, компонент разбивает текст на абзацы и предложения (макросинтаксический анализ).

Графематическая таблица[1] состоит из двух столбцов. В первом столбце стоит некоторый кусок входного текста (выделенный по правилам, о которых будет сказано ниже), во втором столбце стоят графематические дескрипторы, характеризующие этот кусок текста. Например, для текста «Иван спал» будет построена таблица из трех строк:

Кусок входного текста	Графематические дескрипторы
Иван	ЛЕ Бб ПРД1
_	РЗД ПРБ
Спал	ЛЕ бб ПРД2

Так или иначе дескрипторы создают формальное описание текста на уровне графематики, которое уже поддаётся автоматизированной обработке в терминах лингвистических теорий.

1.3. Морфологический анализ

Морфологический компонент осуществляет морфоанализ и лемматизацию русских словоформ. Морфоанализ — приписывание словоформам морфологической информации, лемматизация — приведение текстовых форм слова к словарным. При лемматизации для каждого слова входного текста морфологический процессор выдает множество морфологических интерпретаций следующего вида:

- лемма,
- морфологическая часть речи,
- множество наборов граммем.

[1] Здесь и далее по тексту приводятся примеры реализаций, как это сделано в системе Диалинг [5].

Лемма — это нормальная форма слова. Например, для существительных — это единственное число (если оно есть у существительного), именительный падеж.

Граммема — это элементарный морфологический описатель, относящий словоформу к какому-то морфологическому классу, например, словоформе *стол* с леммой СТОЛ будут приписаны следующий набор граммем: «**мр, ед, им, но**», «**мр, ед, вн, но**». Таким образом, морфологический анализ выдает два варианта анализа словоформы *стол* с леммой СТОЛ внутри одной морфологической интерпретации: с винительным (**вн**) и именительным падежами (**им**).

Также большую роль здесь играет омонимичность словоформ. Например, у словоформы *стали* могут быть следующие интерпретации:

- сталь — существительное;
- стать — глагол.

Таким образом, видно, что морфологического анализа явно не достаточно для выбора одной конкретной морфологической интерпретации слова, к тому же, выбор одной интерпретации может повлиять на выбор интерпретации для соседних слов. Поэтому программы работают с целым набором возможных морфологических интерпретаций, постепенно выделяя наиболее вероятные на следующих этапах анализа.

1.4. Фрагментационный анализ

Фрагментационный анализ — деление предложения на неразрывные синтаксические единства (фрагменты), большие или равные словосочетанию (синтаксической группе), и установление частичной иерархии на множестве этих единств. Фрагменты — это главные и придаточные предложения в составе сложного, причастные, деепричастные и другие обособленные обороты. Иерархия отражает тот факт, что в предложении некоторые фрагменты синтаксически зависимы от других. Так, фрагмент «причастный оборот» будет подчиняться фрагменту, содержащему определяемое слово, придаточное предложение — главному.

Необходимость фрагментационного анализа в системе АОТ вызвана, в первую очередь, техническими причинами.

1.5. Синтаксический анализ

Следующим этапом, после морфологического и фрагментационного анализов, является этап синтаксической обработки текста. Цель синтакси-

ческого анализа — построение групп на предложении. Синтаксическая группа — это отрезок (первое слово группы — последнее слово группы) в предложении, для которого указан подотрезок — его главная группа. В частном случае группа — одно слово. Как видно из определения, синтаксические группы неразрывны, а из того, что две группы пересекаются, следует, что одна лежит в другой (т.е. является ее подотрезком).

Синтаксическую структуру предложения можно представить в виде дерева: корень (нулевой уровень) — само предложение; узлы — синтаксические группы (далее просто группы); листья — элементарные группы (слова); ребра — отношение «лежать непосредственно в» (A->B значит, что B лежит в A и при этом нет такой группы C, что B лежит в C и C лежит в A). До начала работы анализатора каждое слово — группа первого уровня (группы первого уровня не входят ни в какие группы кроме предложения) и кроме корня других групп нет. Результатом работы является «дерево» предложения, описывающее лингвистические отношения подчинения. По сути, это и есть математическая модель предложения на естественном языке.

2. ПОСТАНОВКА ЗАДАЧИ

В настоящее время ведутся активные исследования в области разработки алгоритмов анализа текстов. Результатом этих исследований являются десятки моделей и готовых алгоритмов, которым необходима проверка. При этом до сих пор не существует инструмента, предоставляющего удобные средства для разработки в данной области. Это вынуждает разработчика-лингвиста сосредотачивать внимание не только на написании алгоритма, но и на создании системы, способной запустить этот алгоритм, обеспечить его взаимодействие с остальными и предоставить необходимую информацию о его работе. Таким образом, главной задачей данной работы ставится создание исследовательского стенда для анализа текстов на естественном языке.

Важно отметить, что результатом работы должен быть законченный продукт, подходящий для применения его в качестве полноценного анализатора текстов, а именно стенд, предоставляющий необходимые функции для ведения исследований в области разработки алгоритмов анализа.

Система должна обеспечивать:

- возможность загрузки и редактирования анализируемых текстов;
- анализ текста посредством программируемого конвейера, составленного из разрабатываемых независимо компонентов;

- просмотр результатов анализа текста каждым из компонентов;
- обеспечение замера производительности работы компонентов и визуализацию этих данных;
- возможность независимой разработки компонентов анализатора с последующей возможностью включения в конвейер;
- функции работы со словарями — нахождение словарных статей, возможность создания и подключения новых словарей;
- приемлемое время работы

Отметим, что морфологический и синтаксический анализы производятся посредством использования внешних модулей (системы Диалинг).

В результате проведённого исследования современных технологий и средств разработки, был решён вопрос выбора инструментов для решения поставленной задачи. Кратко основные положения можно представить следующим образом.

- Совместимость с самыми современными технологиями и их использование:
  - язык реализации исследовательского стенда — C#[2];
  - описание и реализация бизнес-логики программируемых модулей анализатора.
- Расширяемость — исследовательский стенд предоставляет возможность изменять существующие блоки анализатора и создавать новые.
- Простота использования — использование графического представления для создания моделей компонентов анализа.
- Безопасность и защищённость — архитектура предоставляет возможность разрабатывать и подключать модули любой сложности (никак не ограничивая их внутреннюю структуру архитектурными особенностями), без предоставления при этом исходного кода.
- Поддержка языков — система позволяет использовать для разработки компонентов анализатора модули, написанные на следующих языках: C, C++, Managed C++, Pascal, Visual Basic и др.

[2] На момент написания программы для исполнения приложений на языке C# в среде операционной системы Microsoft Windows использовались ряд дополнениний.

Принципиальная схема архитектуры

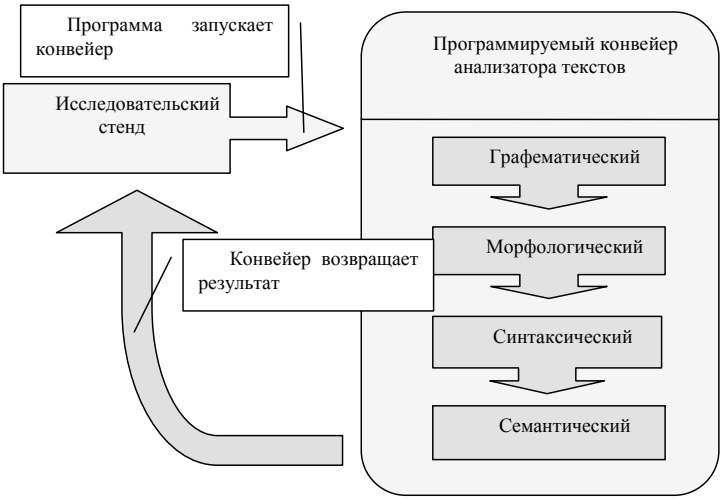


Рис. 1

Как уже упоминалось ранее, для реализации данной архитектуры (рис. 1) был выбран язык C#. Исследовательский стенд является Windows-приложением, предоставляющим функциональность работы со стендом со стороны пользователя, реализуя такие возможности, как загрузка, отображение и редактирование текста, запуск анализа текста и отображение результатов анализа.

Он взаимодействует с программируемым конвейером[3], который и является изменяемой компонентой программного комплекса. Программируемый конвейер предоставляет функциональность работы со стендом со стороны исследователя — разработчика алгоритмов анализа — реализуя такие возможности, как подключение модулей анализатора к программе, а также связывание их в единый конвейер и обеспечение всей функциональности, необходимой для их совместной работы.

[3] Программируемый конвейер — приложение, реализованное на основе технологии Microsoft Framework.

2.2. Приложение исследовательского стенда

Внешний вид пользовательского интерфейса представлен на рис. 2.

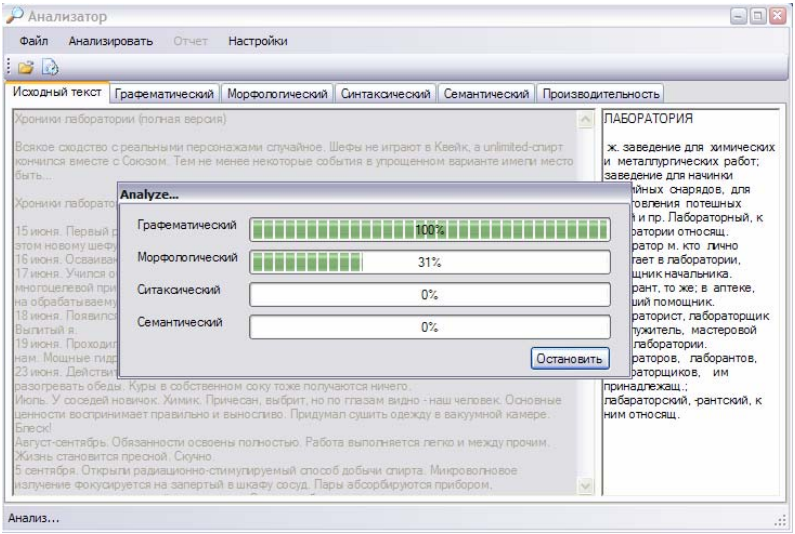


Рис. 2

2.2.1. Результаты оптимизации

Разработка данного приложения проводилась на основе предыдущих работ, проведённых в данном направлении. Для обеспечения приемлемой производительности были разработаны структуры хранения и управления данными.

Применение данных моделей привело к значительному ускорению работы программы. Для тестирования производились выборки одних и тех же слов из оптимизированных и неоптимизированных словарей (рис. 3).

Важно отметить, что в данном тесте не использовался механизм кэширования, т.е. и в оптимизированных словарях каждый раз происходило обращение к жесткому диску. При выборке же слов из памяти, время можно считать равным нулю.

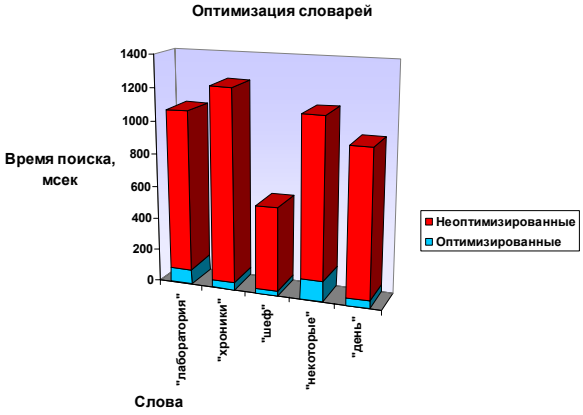
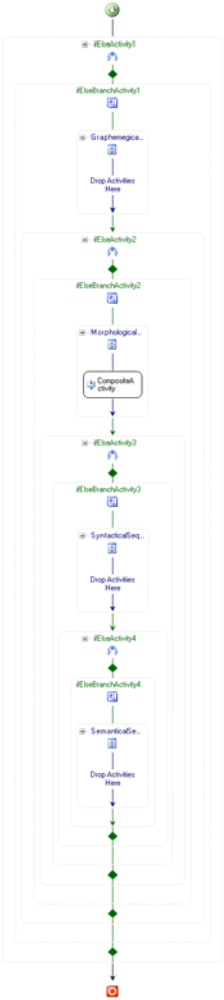


Рис. 3



2.3. Программируемый конвейер

Основной функцией данного программного модуля является предоставление конвейера анализатора в виде четырёх последовательно исполняемых модулей: графематического, морфологического, синтаксического и семантического. Эти модули реализуют логику, описанную в разд. 1. Эта часть программы реализована на основе технологии Microsoft Framework. Схема конвейера отображена на схеме.

Данная система позволяет разрабатывать элементы анализатора на уровне WYSIWYG. Для потенциальных пользователей, специалистов в области лингвистики, но не программистов, эта возможность, безусловно, имеет большое значение. Разработка анализаторов, с учётом возможностей данной технологии, сводится к написанию функциональных элементарных блоков и последующей их компоновке с использованием графического представления.

2.4. Блоки программируемого конвейера

В рамках работы были реализованы и протестированы два первых компонента конвейера анализатора.

2.4.1. Графематический анализатор

Это первый компонент программируемого конвейера анализатора текстов. Его задача описана в разд. 1.2. В рамках задачи данный компонент был реализован на основе Microsoft Framework. Результаты работы компонента можно увидеть на рис. 4.

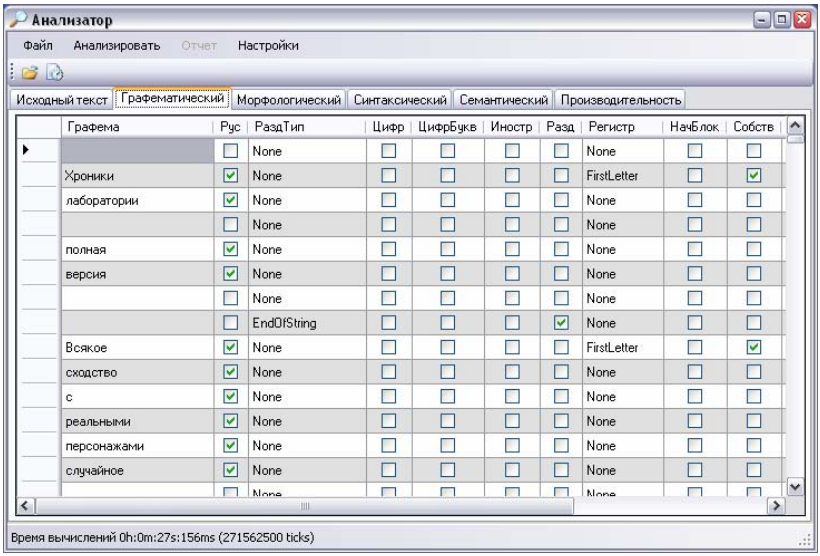
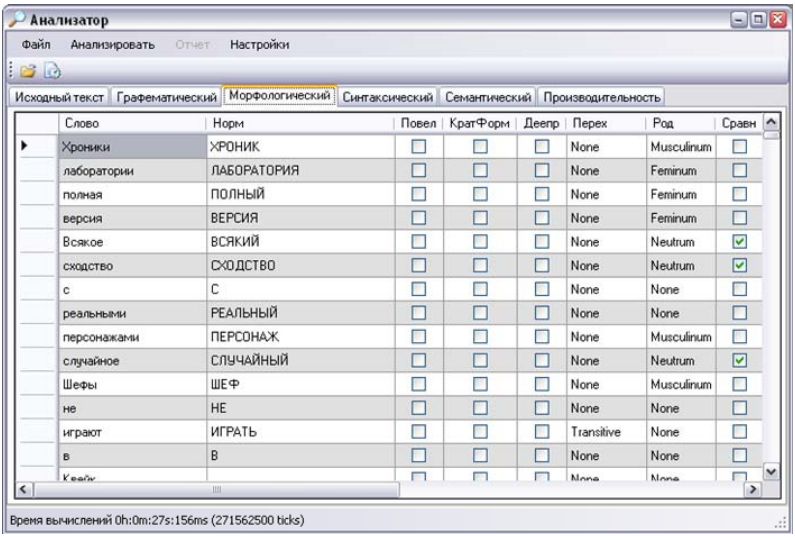


Рис. 4

2.4.2. Морфологический анализатор

Это второй блок программируемого конвейера анализатора текстов. Его задача описана в разд. 1.3. В рамках задачи данный компонент был также реализован на основе Microsoft Framework. Результаты работы компонента можно увидеть на рис. 5.



	Слово	Норм	Повел	КратФорм	Деепр	Перех	Род	Сравн
▶	Хроники	ХРОНИК	☐	☐	☐	None	Musculinum	☐
	лаборатории	ЛАБОРАТОРИЯ	☐	☐	☐	None	Feminum	☐
	полная	ПОЛНЫЙ	☐	☐	☐	None	Feminum	☐
	версия	ВЕРСИЯ	☐	☐	☐	None	Feminum	☐
	Всякое	ВСЯКИЙ	☐	☐	☐	None	Neutrum	☑
	сходство	СХОДСТВО	☐	☐	☐	None	Neutrum	☑
	с	С	☐	☐	☐	None	None	☐
	реальными	РЕАЛЬНЫЙ	☐	☐	☐	None	None	☐
	персонажами	ПЕРСОНАЖ	☐	☐	☐	None	Musculinum	☐
	случайное	СЛУЧАЙНЫЙ	☐	☐	☐	None	Neutrum	☑
	Шефы	ШЕФ	☐	☐	☐	None	Musculinum	☐
	не	НЕ	☐	☐	☐	None	None	☐
	играют	ИГРАТЬ	☐	☐	☐	Transitive	None	☐
	в	В	☐	☐	☐	None	None	☐
	Квейк		☐	☐	☐	None	None	☐

Рис. 5

ЗАКЛЮЧЕНИЕ

В результате работы был создан исследовательский стенд для анализа текстов на естественном языке. Более или менее успешно были решены все задачи, поставленные в рамках данной работы. Созданный инструмент уже на настоящий момент времени позволяет вести на нём работу по испытанию разрабатываемых лингвистических моделей. Это крайне важно, потому что именно практическое применение было одним из основных факторов, повлиявших на решение о разработке данного исследовательского стенда.

Выполненные работы по оптимизации и разработке удобного интерфейса позволили предоставить не только функциональный, но и комфортный интерфейс. А разработка гибкой архитектуры дала возможность разработчикам алгоритмов анализа самим выбирать как средства реализации (в том числе и основанные на современной технологии Microsoft Framework, так и способы отображения информации при диагностике алгоритмов.

Также в результате работы были теоретически проанализированы и технически оценены приоритетные направления развития данного программного комплекса. Среди которых можно выделить следующие.

- Стандартизация хранения словарных данных путём использования распространённых баз данных (также является шагом к оптимизации скорости работы).
- Обеспечение возможности использования Интернет в качестве информационной базы.
- Реализация механизма отождествления выражений на базе языка REFAL.
- Реализация компонента синтаксического анализа.
- Реализация системы распределённого анализа на основе данного программного комплекса.
- Создание интерфейса для возможности использования функций данного приложения другими программами в своих целях.
- Добавление функции исполнения скриптов — программ на псевдоязыке — позволяющее автоматизировано обрабатывать массивы текстов (анализ множества файлов, складирование результатов).
- Создание универсального формата отчётов о проведённых анализах с целью сравнения результатов работы различных алгоритмов.

Некоторые их этих задач могут лечь в основу дальнейшей исследовательской деятельности в данном направлении. Таким образом, помимо решения конкретной задачи, была подготовлена база для последующих работ и определены перспективы развития.

СПИСОК ЛИТЕРАТУРЫ

1. Мельчук И.А. Опыт теории лингвистических моделей типа «Смысл⇔Текст». — М.: Наука, 1974. — 315 с.
2. Леонтьева Н.Н. Система французско-русского автоматического перевода (ФРАП): лингвистические решения, состав, реализация // МП и ПЛ. Проблемы создания системы автом. перевода / Сб. научн. трудов МГПИИЯ им. М. Тореза. — Вып. 271. — М., 1987. — С. 6–25.
3. Леонтьева Н.Н. ПОЛИТекст: информационный анализ политических текстов // Сб. НТИ. — 1995. — Сер. 2, N 4.
4. Кудряшова И.М. О семантическом словаре в системе ФРАП // Сб. научн. трудов. — М.: МГПИИЯ им. М. Тореза, 1986. — Вып. 271. — 8 с.
5. Сокирко А.В. Семантические словари в автоматической обработке текста. // Канд. дисс., МГПИИЯ. — М., 2000. — 108 с.
6. Кулагина О.С. Исследования по машинному переводу. — М.: Наука, 1979. — 127 с.