Software Engineering Conference Russia 2019
November 13-15, SPB

Применение фреймворка автоматического анализа текста на русском языке для решения прикладных задач

Екатерина Полицына Сергей Полицын

Александр Поречный

Московский авиационный институт

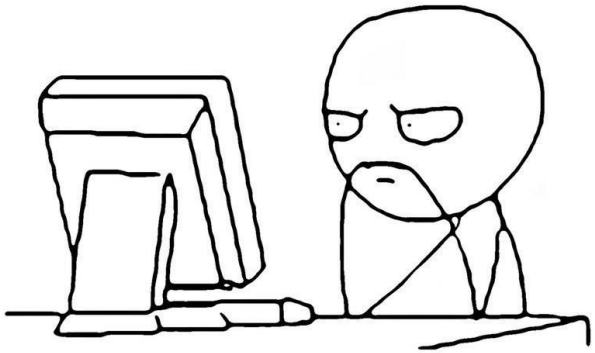
Что происходит?

✓Справочные ресурсы

✓Литература

✓Управление знаниями и задачами в проектах

✓Научные статьи



✓Мессенджеры

✓Социальные сети

✓Почта

✓Отчетность

✓Документация

Как с этими данными “бороться”?

МЕДИАЛОГИЯ

Яндекс

- Результаты автоматизированного анализа текста могут **существенно упростить**, **сократить и ускорить** работу человека
- **Автоматизация** в системах, которых работают в **большими объемами текстовых данных** (поисковые, новостные, рекомендательные системы)
- **Системы**, построенные на применении **автоматического анализа текста** (системы антиплагиата, спам-фильтры)
- Потребность в **автоматизации** обработки **текстов** в самых разных **прикладных системах** (документооборот, e-commerce)

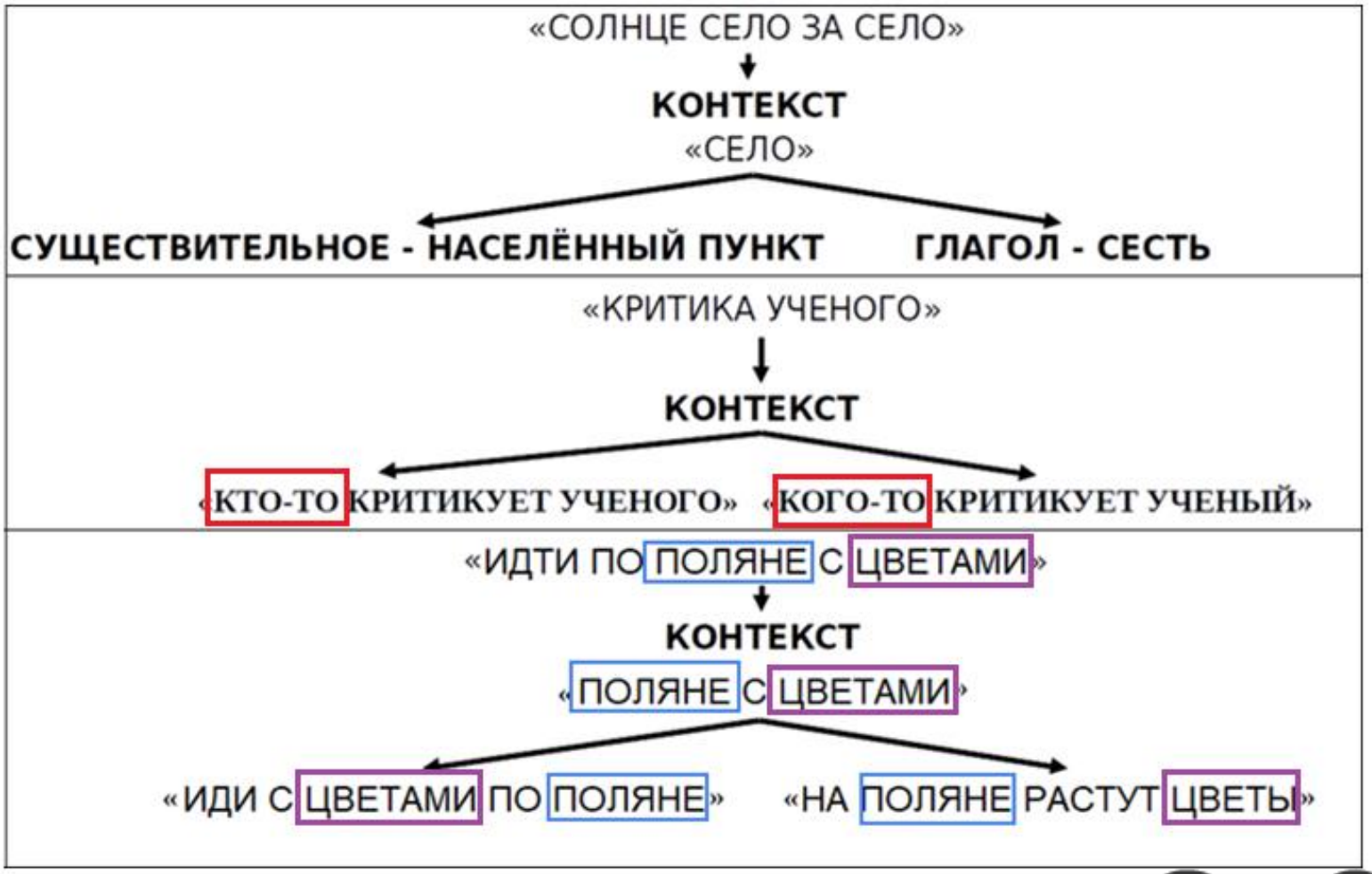
РУКОНТ
НАЦИОНАЛЬНЫЙ ЦИФРОВОЙ РЕСУРС

SCAN
интерфакс
INTERFAX

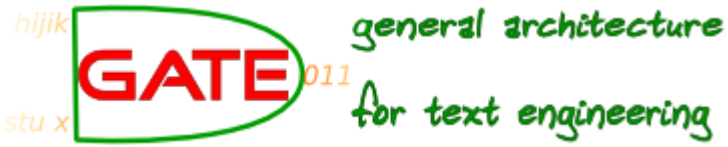
АНТИПЛАГИАТ

SECR

Проблемы компьютерной лингвистики



Компьютерная лингвистика



wavii

LingPipe

ABBYY

Морфологический анализатор
pymorphy2

TEXTERRA

CrossMorphy

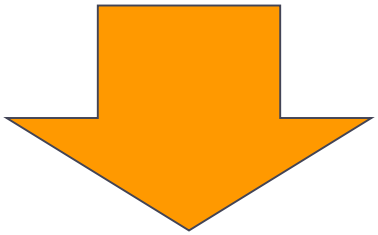
Яндекс

MyStem

Russian Morphology for Lucene

Почему нужен фреймворк?

- ✓ Естественно-языковой текст **неоднозначен** и **плохо формализуемый**
- ✓ Автоматический анализ текста – **сложно**, “смысловой” анализ – **очень сложно**
- ✓ Выбор из многообразия существующих инструментов анализа текста - отдельная **исследовательская задача**, интеграция разрозненных инструментов - **ночной кошмар** инженера



- ✓ Инструменты **должны не отличаться** по подключению, использованию, сборке от других библиотек и фреймворков
- ✓ Нужны библиотеки, использующие **популярные языки** программирования
- ✓ **Максимально оптимально** расходовать ресурсы
- ✓ Для алгоритмов машинного обучения **при векторизации** документов необходимы инструменты, позволяющие это делать максимально эффективно

Разработка приложений с JMorfSdk

1. **Сервис** автоматического реферирования текстов
   http://abstracts.textanalysis.ru
2. **Плагин** для распознавания мошеннических сообщений

FRAUDHUNTER

Автор: vkfraudhunter

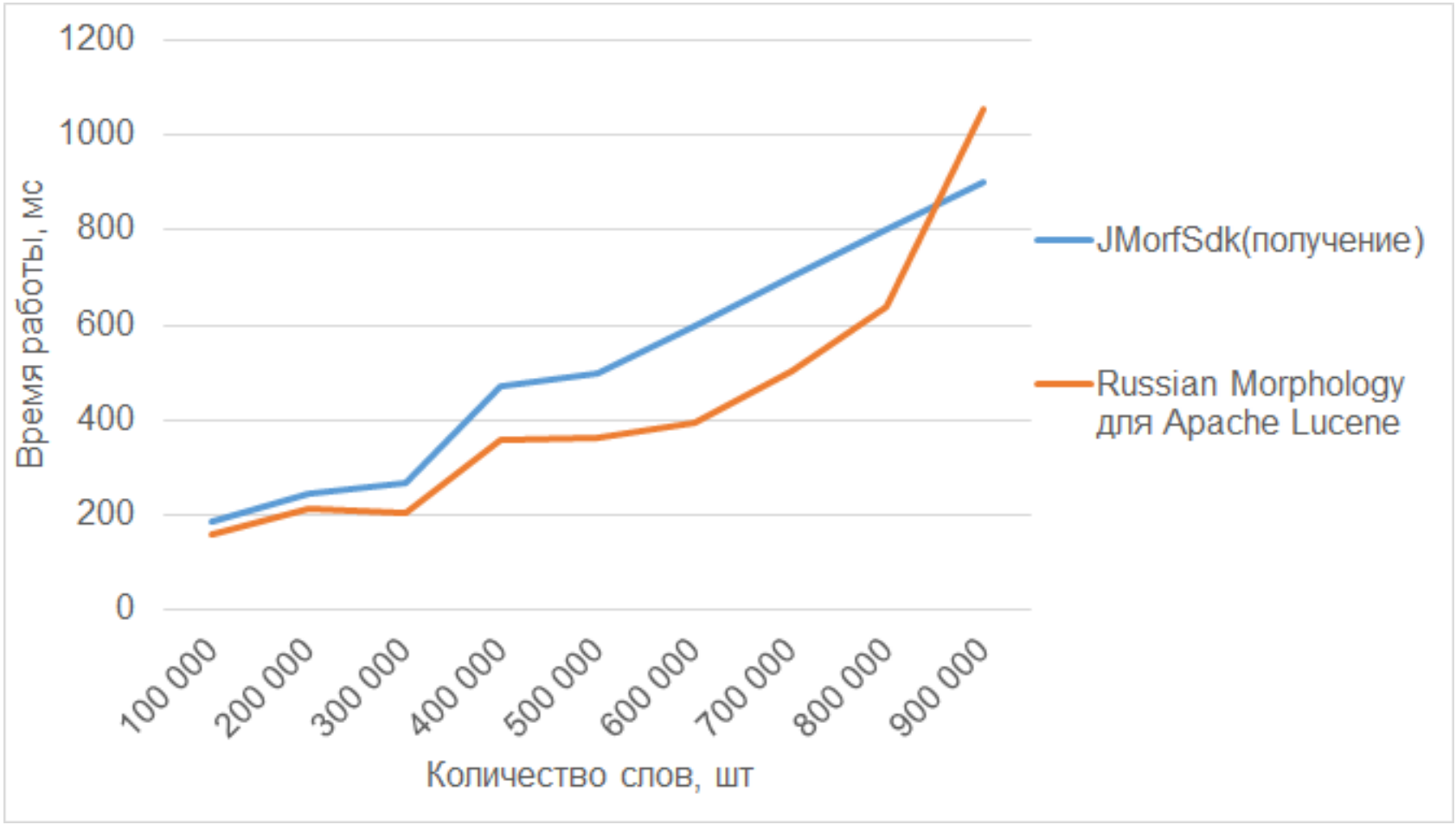
3. **Сервис** подбора синонимов с учетом тематики
   http://synonyms.textanalysis.ru

4. Приложение **FriendFinder** http://friendsfinder.textanalysis.ru
5. Приложение **TouristHelper 2.0**

Выводы и устраненные недостатки:

- **долгая** загрузка словаря
- повторное **использование** библиотеки
- нужна **модульность** в рамках даже одной библиотеки
- нужны аналогичные простые инструменты для семантико-синтаксического анализа
- нужно еще уменьшать требования к ресурсам

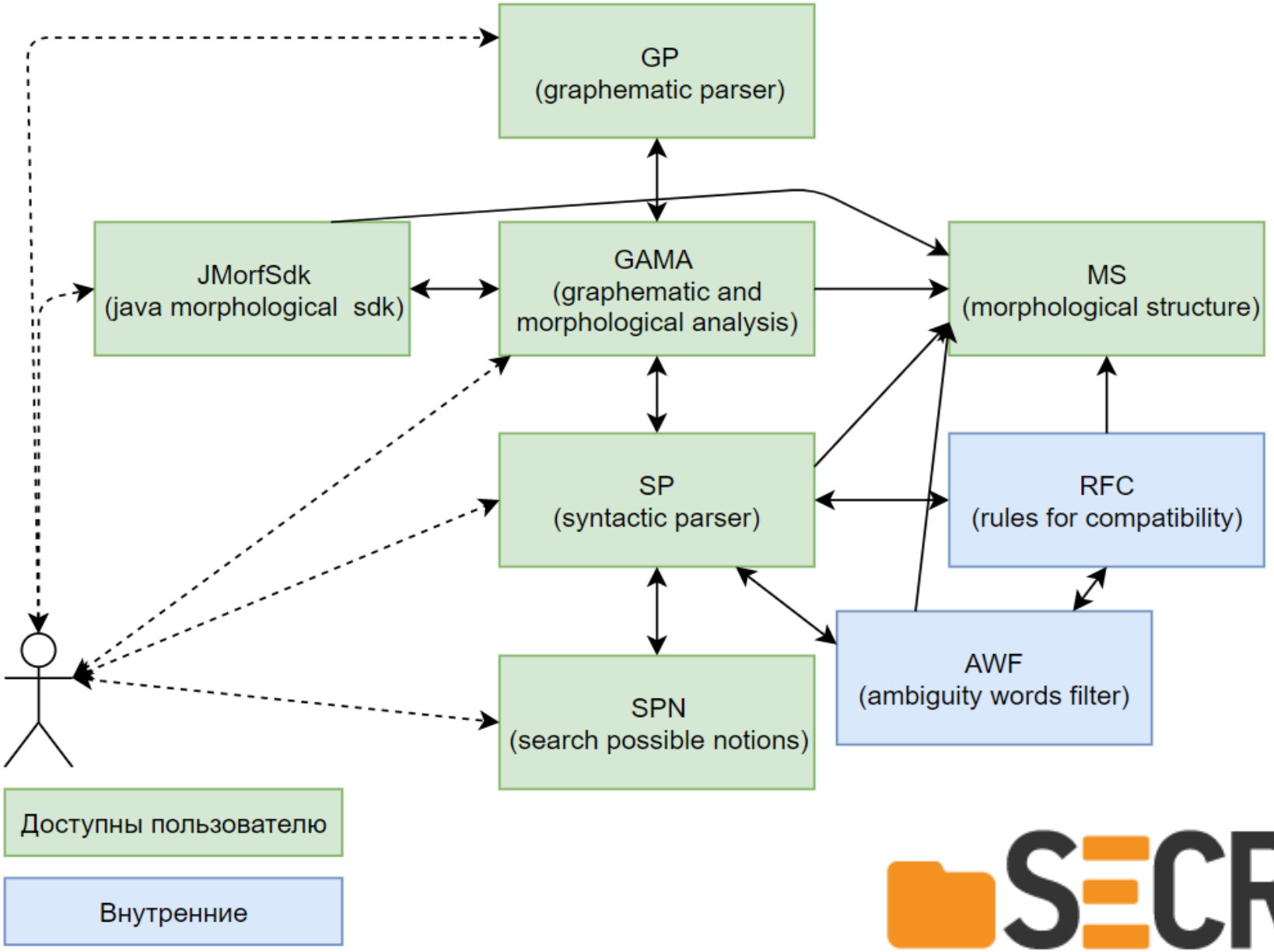
Сравнение с Java Russian Morphology для Apache Lucene



Какой нужен фреймворк?

1. Реализация **трех этапов анализа текста**: графематического, морфологического и семантико-синтаксического, - и инструмент, связывающий этапы друг с другом.
2. **API инструментов** анализа текста, который **не требует глубокого «погружения»** в основы и **особенности** обработки **естественного языка**.
3. **Готовые структуры данных** разобранного текста после **каждого этапа** анализа.
4. **Минимальная зависимость** от разработчиков инструмента:
   - ✓ Открытая разработка.
   - ✓ Не веб-API (возможна последующая дополнительная реализация на основании фреймворка).
5. **Возможность добавлять и заменять** отдельные **инструменты** фреймворка без изменения исходного кода остальных инструментов.
6. **Автономность**, т.е. функционирование не в составе другой крупной системы.
7. **Кроссплатформенность**.
8. **Поддержка русского языка.**

Структура фреймворка TAWT



GP (Graphematic Parser)

Инструмент GP (Graphematic Parser) реализует графематический этап анализа текста, основан на наборе правил, реализованных средствами регулярных выражений.

```
GraphematicParser parser = new GParserImpl();
List<List<String>> listBasicPhase = parser.parserSentence("Graphematic Parser - это программа"
        + " начального анализа естественно-языкового текста, рассматривающая его как цепочку"
        + " символов для получения информации, необходимой для следующих этапов анализа.");
System.out.println(listBasicPhase);
```

```
[
    [-, это, программа, начального, анализа, естественно-языкового, текста],
    [рассматривающая, его, как, цепочку, символов, для, получения, информации],
    [необходимой, для, следующих, этапов, анализа]
]
```

JMorfSdk

- ✓ Получение **морфологических характеристик** слова
- ✓ Получение **начальной формы** слова
- ✓ **Генерация слов** по заданным характеристикам

https://github.com/jalexpr/jmorfsdk

1. Использует **OpenCorpora** (390 тысяч уникальных слов, более 5 млн. словоформ).
2. Представление **морфологических характеристик** слов в виде **бинарной шкалы.**
3. **Целочисленное** внутреннее **представление слов**. Для исключения коллизий, используется два алгоритма хэширования.
4. **Строковое представление слов** хранится в **базе данных.**

Фильтрация слов по морфологическим характеристикам

```
List<String> words1 = Arrays.asList("осенний", "осенней", "площадь", "стол", "играть", "конференций", "на", "бежала");
for (String word : words1) {
    jMorfSdk.getAllCharacteristicsOfForm(word).forEach(form -> {
        if (form.getTheMorfCharacteristics(MorfologyParameters.Gender.class) == MorfologyParameters.Gender.FEMININ) {
            System.out.println(form + " - " + word);
        }
    });
}
```

```
initialFormString = осенний, typeOfSpeech = 18, morfCharacteristics = 488 - осенней
initialFormString = площадь, typeOfSpeech = 17, morfCharacteristics = 555 - площадь
initialFormString = конференция, typeOfSpeech = 17, morfCharacteristics = 187 - конференций
initialFormString = бегу, typeOfSpeech = 20, morfCharacteristics = 670760 - бежала
```

Фильтрация слов по морфологическим характеристикам

```
List<String> words2 = Arrays.asList("осенний", "осенней", "площадь", "стол", "играть", "конференций", "на", "бежала");
for (String word : words2) {
    jMorfSdk.getAllCharacteristicsOfForm(word).forEach(form -> {
        if (form.getTypeOfSpeech() == MorfologyParameters.TypeOfSpeech.NOUN) {
            System.out.println(form + " - " + word);
        }
    });
}
```

```
initialFormString = площадь, typeOfSpeech = 17, morfCharacteristics = 107 - площадь
initialFormString = стол, typeOfSpeech = 17, morfCharacteristics = 103 - стол
initialFormString = конференция, typeOfSpeech = 17, morfCharacteristics = 187 - конференций
```

Фильтрация слов по морфологическим характеристикам

```
List<String> words3 = Arrays.asList("осенний", "осенней", "площадь", "стол", "играть", "конференций", "на", "бежала");
for (String word : words3) {
    jMorfSdk.getAllCharacteristicsOfForm(word).forEach(form -> {
        if (form.getTypeOfSpeech() == MorfologyParameters.TypeOfSpeech.NOUN
        && form.getTheMorfCharacteristics(MorfologyParameters.Gender.class) == MorfologyParameters.Gender.FEMININ
        ) {
            System.out.println(form + " - " + word);
        }
    });
}
```

```
initialFormString = площадь, typeOfSpeech = 17, morfCharacteristics = 107 - площадь
initialFormString = конференция, typeOfSpeech = 17, morfCharacteristics = 187 - конференций
```

Получение заданной формы слова

```
jMorfSdk.getAllCharacteristicsOfForm("дорогой").forEach(form -> {
    if (form.getTheMorfCharacteristics(MorfologyParameters.Case.IDENTIFIER)
            == MorfologyParameters.Case.GENITIVE) {
        System.out.println(form);
    }
});
```

```
initialFormString = дорогой  typeOfSpeech = 18, morfCharacteristics = 4264
```

```
jMorfSdk.getAllCharacteristicsOfForm("дорогой").forEach(form -> {
    if (form.getTypeOfSpeech() == MorfologyParameters.TypeOfSpeech.NOUN) {
        System.out.println(form);
    }
});
```

```
initialFormString = дорога  typeOfSpeech = 17, morfCharacteristics = 363
```

Генерация слова с заданными характеристиками

```
jMorfSdk.getDerivativeForm("дерево",
        MorfologyParameters.TypeOfSpeech.NOUN,
        MorfologyParameters.Numbers.SINGULAR)
        .forEach(System.out::println);
```

```
дерева  - родительный падеж
дереву  - дательный падеж
дерево  - винительный падеж
деревом - творительный падеж
дереве  - предложный падеж
```

MS (Morphological Structure)

Инструмент MS (Morphological Structure) – инструмент загрузки, хранения, обработки и выдачи **форм слова и его характеристик**, содержит описание всех **структур данных**, которые применяются во фреймворке, конвертирует словари, применяемые в JMorfSdk, из исходного формата в формат, который используется во фреймворке.

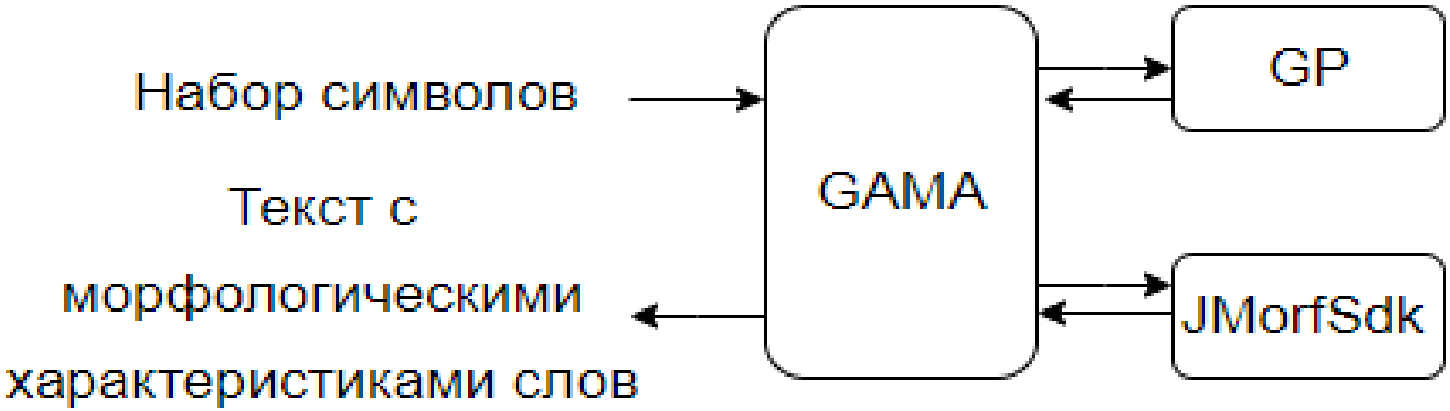
Отдельный модуль - для обеспечения:

- независимости от представления и способов хранения словаря;
- независимости от структуры хранения слов, словоформ, оборотов, предложений, на каждом уровне анализа;
- возможности реализации поддержки других языков.

GAMA

Инструмент GAMA (Graphematic and Morphological Analysis) – агрегирует методы **графематического** и **морфологического** этапов анализа текста, **поддерживает замену** инструментов графематического и морфологического анализа.

GAMA предоставляет простой интерфейс для работы с инструментами наиболее популярных этапов анализа.



Результат работы GP

```
GraphematicParser parser = new GParserImpl();
List<List<List<String>>> listBasicsPhase = parser.parserParagraph("Осенний марафон -"
        + " стало ясно, что будет с российской валютой. Справедливый курс,"
        + " по мнению аналитиков, – на уровне 65-66.");
System.out.println(listBasicsPhase);
```

```
[
    [
        [Осенний, марафон, –, стало, ясно], [что, будет, с, российской, валютой]
    ],
    [
        [Справедливый, курс], [по, мнению, аналитиков], [на, уровне, 65-66]
    ]
]
```

Результат работы GAMA

```
Gama gama = new Gama();
gama.init();
RefSentenceList sentenceList = gama.getMorphParagraph("Осенний марафон -"
        + " стало ясно, что будет с российской валютой. Справедливый курс,"
        + " по мнению аналитиков, – на уровне 65-66.");
System.out.println(sentenceList);
```

```
RefSentenceList:
    RefBearingPhraseList:
        RefWordList:
            RefOmoFormList - осенний : [{isInit=1,осенний,ToS=18,morf=100},
                                        {isInit=0,осенний,ToS=18,morf=551}],
            RefOmoFormList - марафон : [{isInit=1,марафон,ToS=17,morf=103},
                                        {isInit=0,марафон,ToS=17,morf=551}],
            RefOmoFormList - '-':      [],
            RefOmoFormList - стал :    [{isInit=0,стало,ToS=20,morf=669740}],
            RefOmoFormList - ясно :    [{isInit=1,ясно,ToS=9,morf=8388608},
                                        {isInit=0,ясно,ToS=19,morf=4140}]],
            RefWordList:
                RefOmoFormList - что : [{isInit=1,что,ToS=13,morf=0},
                                        {isInit=1,что,ToS=30,morf=108},
                ...
            ...
        ...
    RefBearingPhraseList:
        RefWordList:
            RefOmoFormList - справедливый :[{isInit=1,справедливый,ToS=18,morf=4196},
                                            {isInit=0,справедливый,ToS=18,morf=4647}],
            RefOmoFormList - курс :        [{isInit=1,курс,ToS=17,morf=103},
                                            {isInit=0,курс,ToS=17,morf=551}]],
            RefWordList:
                RefOmoFormList - по :      [{isInit=1,по,ToS=17,morf=-2130706321},
                                            {isInit=0,по,ToS=17,morf=-2130706193},
                ...
```

RFC (Rules for Compatibility)

«***Одинаковым последовательностям*** символов классов слов соответствуют ***одинаковые синтаксические структуры***»
(проф. Белоногов Г.Г.)

№	Класс главного слова	Класс зависимого слова	Правило	Удаленность	
				слева	справа
1	сущ., отгл., сущ.	сущ., отгл., сущ.	Зависимое в род. падеже или главное в дат. или твор. падеже	нет	2
2	предлог	сущ.	Совпадение по падежу	2	∞
3	сущ.	прил.	Совпадение по падежу, числу, а также роду, если сущ. в ед.ч.	2	2
4	отгл., прил.	сущ.	Совпадение по падежу	нет	2
	…				

AWF (Ambiguity Words Filter)

«СОЛНЦЕ СЕЛО ЗА СЕЛО»

↓

«СЕЛО»

СУЩЕСТВИТЕЛЬНОЕ-НАСЕЛЁННЫЙ ПУНКТ

ГЛАГОЛ-СЕСТЬ

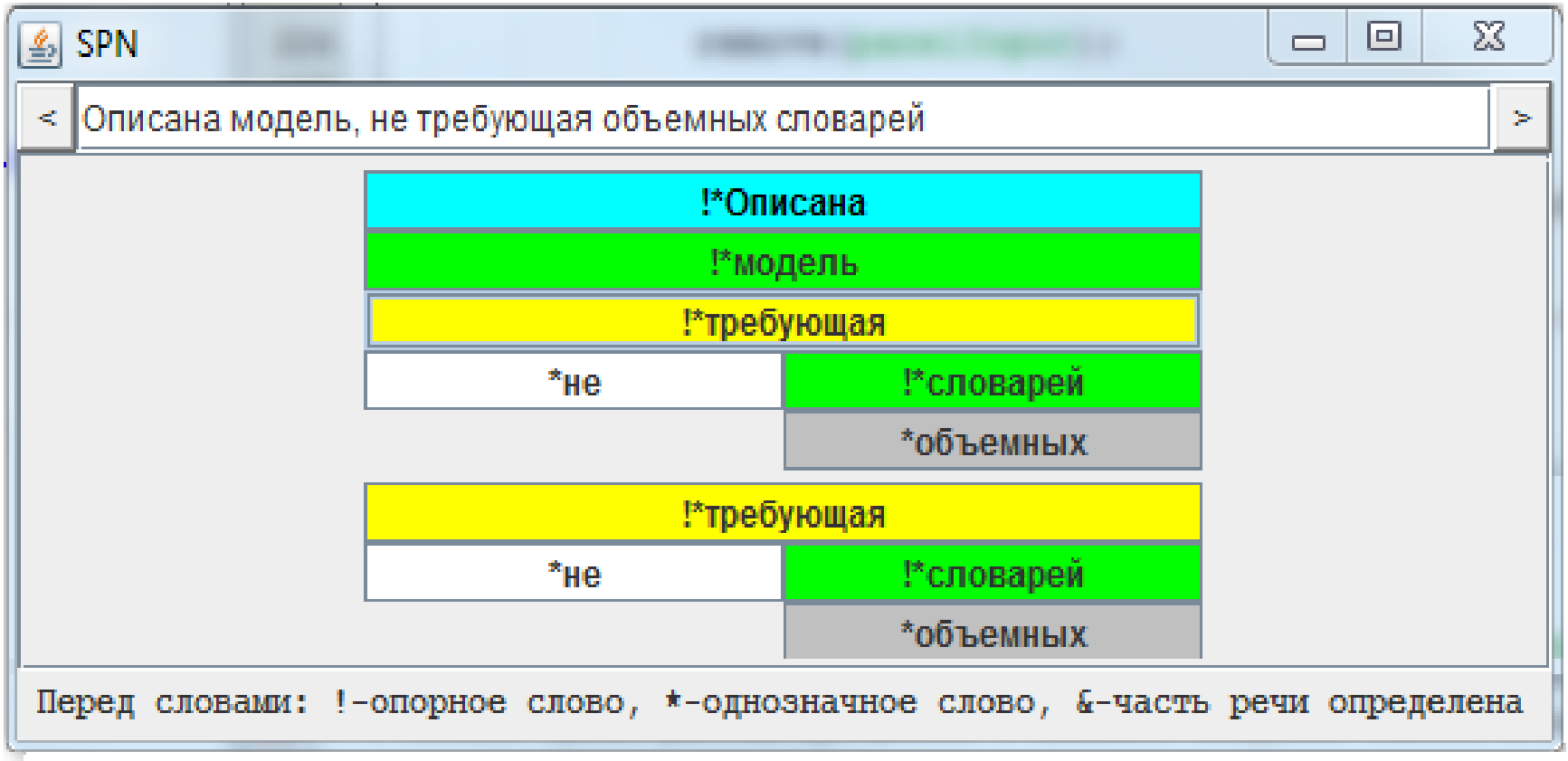
СУЩ. + ГЛ. + ПРЕДЛОГ + ГЛ.
СУЩ. + СУЩ. + ПРЕДЛОГ + ГЛ.
СУЩ. + СУЩ. + **ПРЕДЛОГ + СУЩ.**
СУЩ. + ГЛ. + **ПРЕДЛОГ + СУЩ.**

SP (Syntactic Parser)

```
SyntaxParser sp = new SyntaxParser();
sp.init();
List<BearingPhraseSP> phrase
        = sp.getTreeSentence( text: "Стало ясно, что будет с российской валютой.");
phrase.forEach(System.out::println);
```

```
BearingPhraseSP{
  words=[
    word={стало, ToS=20}, main=[нет], dependents={ясно, ToS=9},
    word={ясно, ToS=9}, main={стало, ToS=20}, dependents=[нет]
  ],
  mainOmoForm={стало, ToS=20}
}
BearingPhraseSP{
  words=[
    word={будет, ToS=20}, main=[нет], dependents={с, ToS=12},
    word={с, ToS=12}, main={будет, ToS=20}, dependents={валютой, ToS=17},
    word={российской, ToS=18}, main={валютой, ToS=17}, dependents=[нет],
    word={валютой, ToS=17}, main={с, ToS=12}, dependents={российской, ToS=18}
  ],
  mainOmoForm={будет, ToS=20}
}
```

Пример работы инструмента семантико-синтаксического анализа



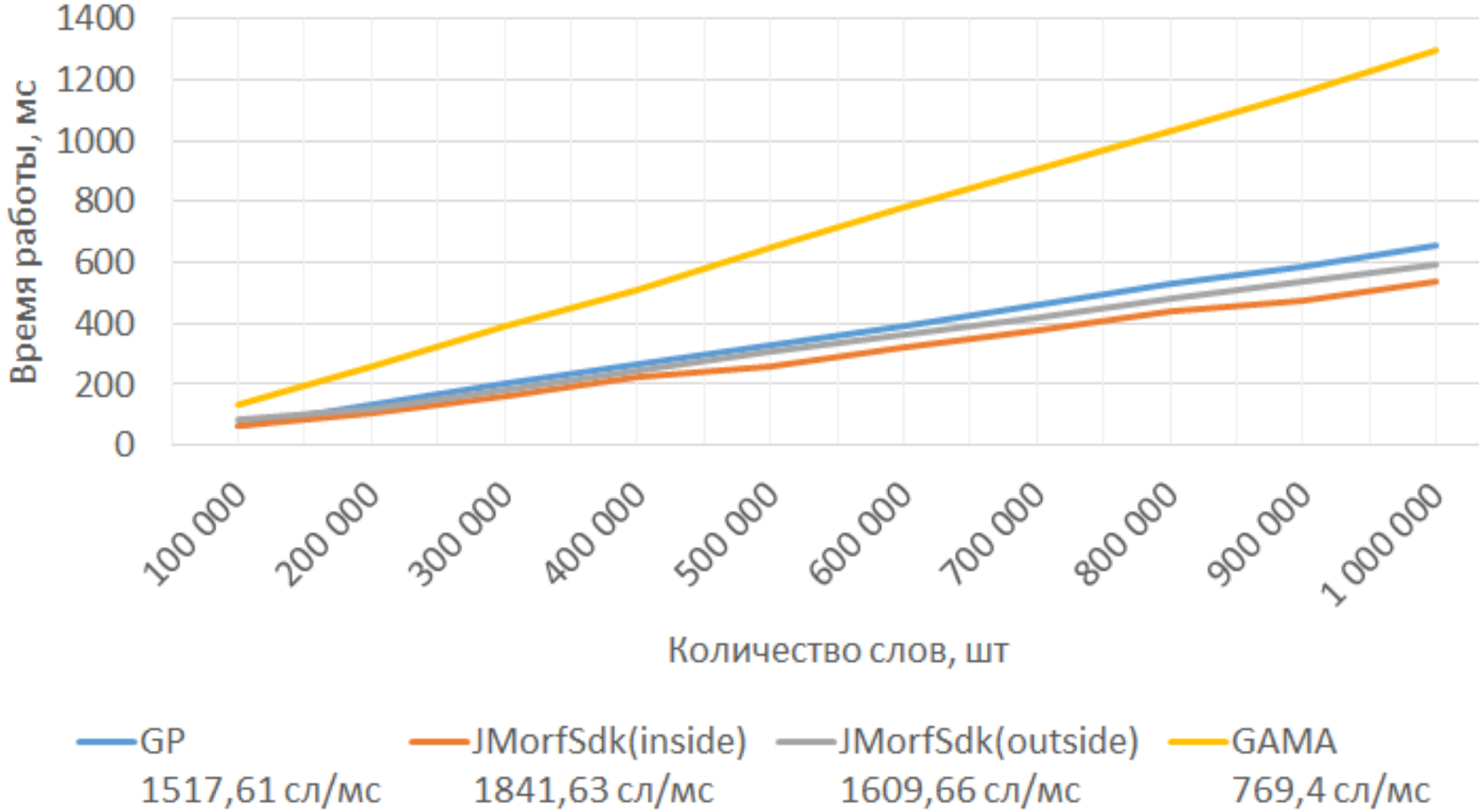
SPN (Search Possible Notions)

«Понятие - это **устойчивое сочетание слов**, **выражающие целостное значение** и по функции соотносящееся с конкретным объектом или явлением реального мира» (проф. Белоногов Г.Г.).

ПДД	
Кол. повторов	Понятие
659	транспортное средство
379	Правительство РФ
297	Постановление РФ
…	…
73	Российская Федерация
61	грузовой автомобиль
56	механическое средство
…	…

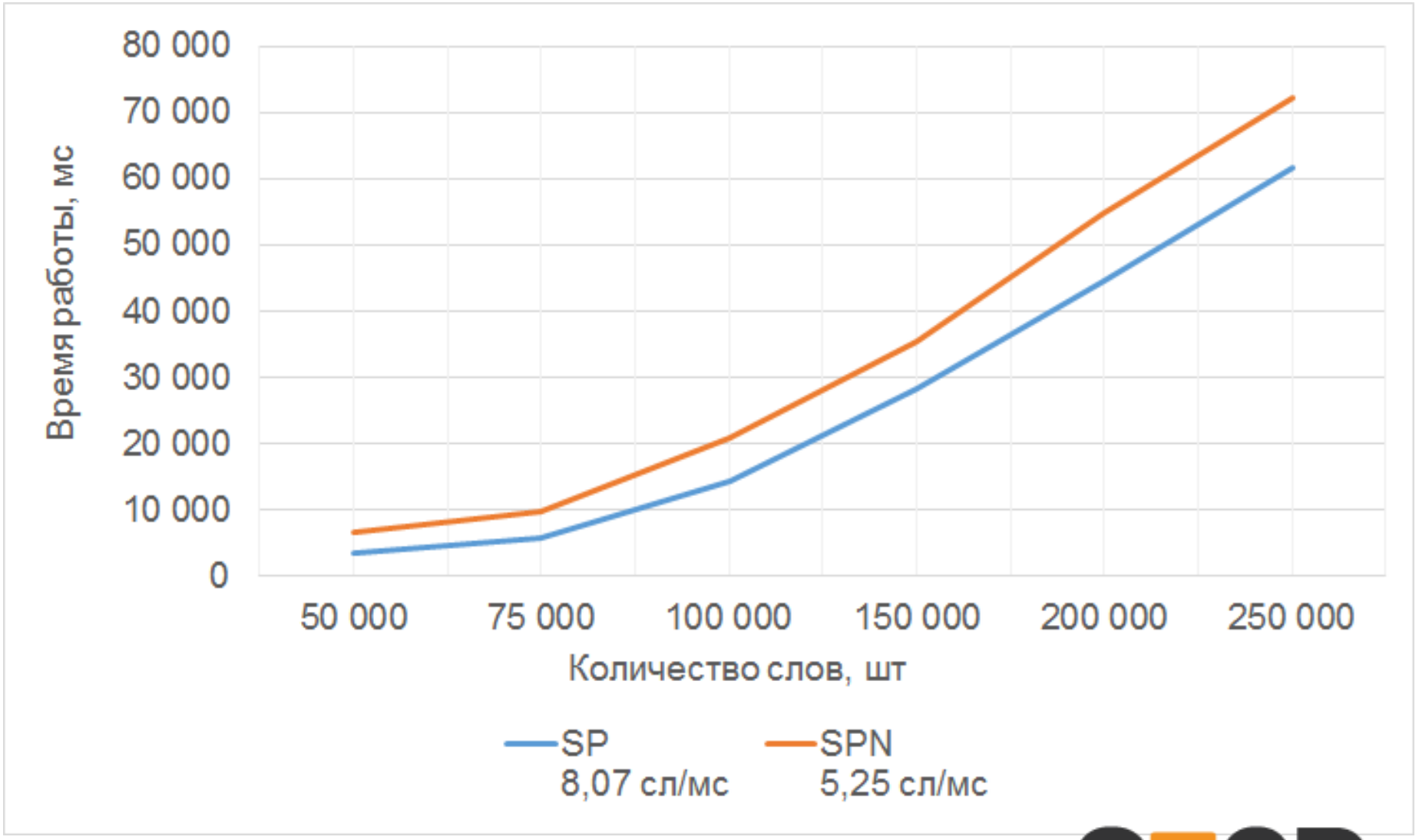
Большая кулинарная книга. Сборник.	
Кол. повторов	Понятие
1414	сливочное масло
952	репчатый лук
765	лавровый лист
…	…
206	стакан воды
185	слабый огонь
174	чайная ложка
…	…

Скорость работы GP, JMorfSdk, GAMA



*Intel(R) Core(™) i7-3630QM 2.4GHz, 8 GB RAM, Windows 7

Скорость работы SP, SPN



*Intel(R) Core(™) i7-3630QM 2.4GHz, 8 GB RAM, Windows 7

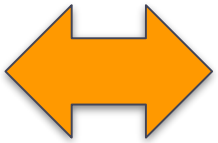
Автоматизация подготовки технической документации

- **Валидация** структуры документа на соответствие ГОСТу.
- **Валидация** наличия в разделе «**Термины и определения**» всех употребляемых в тексте документа аббревиатур.
- **Поиск** схожих технических решений на основе ТЗ или описания эксплуатационных условий:
  - применение средств **морфологического** и **семантико-синтаксического** анализа;
  - выделение **ключевых слов**;
  - построение **краткого содержания** документа.

Применение инструментов TAWT

Валидация структуры документа

ГОСТ 34.602-89



ТЗ на автоматизированный ОКГУ

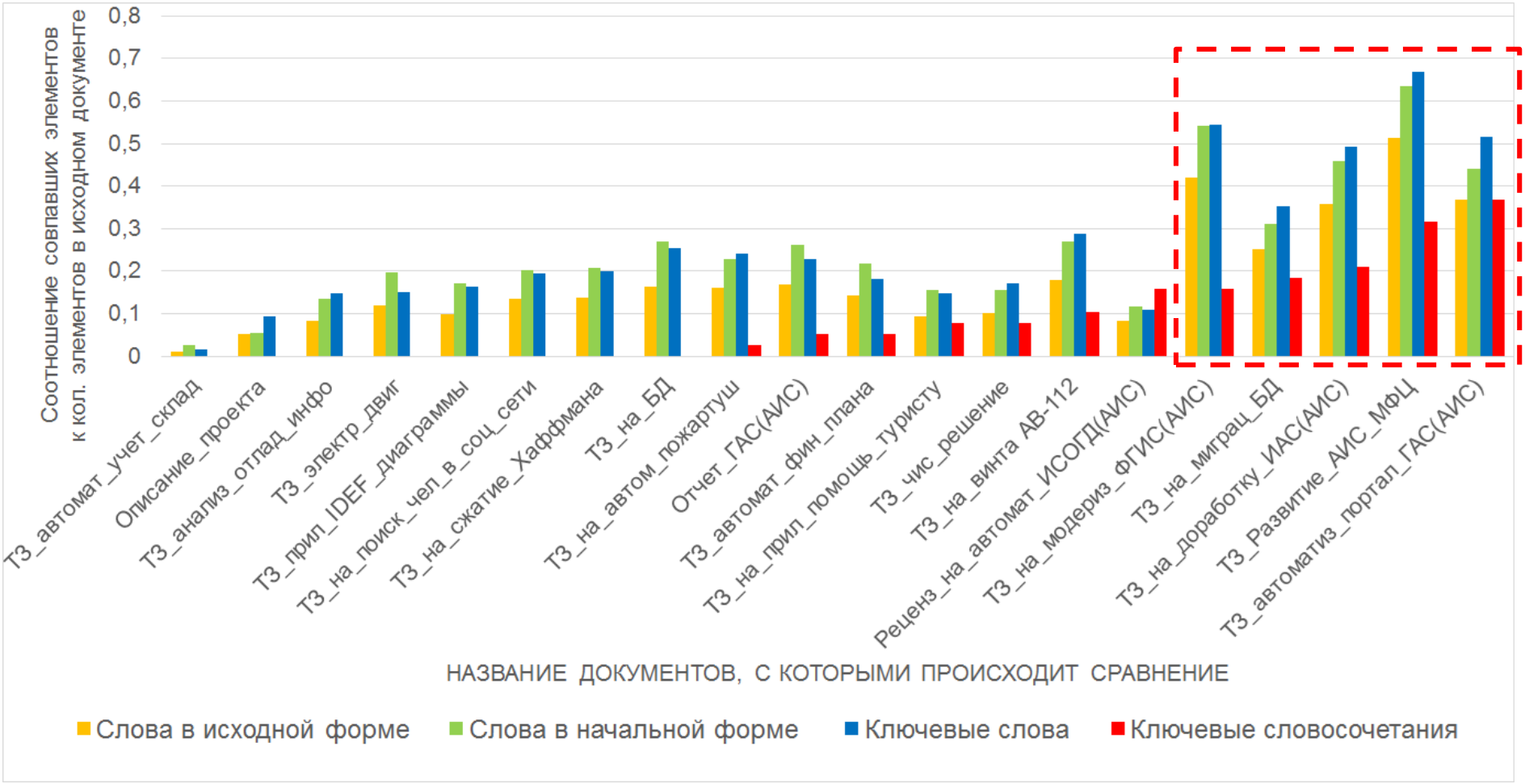
1) Общие сведения
2) Назначение и цели создания (развития) системы
3) Характеристика объектов автоматизации
4) Требования к системе
5) Состав и содержание работ по созданию системы
6) Порядок контроля и приемки системы
7) **Требования к составу и содержанию работ по вводу системы в действие**
8) Требования к документированию
9) **Источники разработки**

1) **Сокращения и наименования**
2) Общие сведения
3) Назначение и цели работ по поддержке
4) Характеристика объекта автоматизации
5) Требования к АМ ОКГУ
6) Состав и содержание работ
7) Общие требования к приемке
8) Требования к документированию

Валидация аббревиатур

ВИД АНАЛИЗА	РЕЗУЛЬТАТ
Графематический анализ с использованием регулярных выражений	(СНиП), **ТЕХНОЛОГИЙ**, ЕСМЭВ, **ОПИСАНИЕ**, **ЗАДАНИЕ**, ИТ, **ПРИЛОЖЕНИЕ**, СНИЛС;, КПП., **ЦЕЛОСТНОСТИ**, ОЗУ, СВТ, **ВЗАИМОДЕЙСТВИЯ**, БД, ФСБ, **СОСТАВИЛИ**, ГОСТ, **ДОРАБОТКУ**, **ПРИНЦИПЫ**, ЛВС., **НАРУШЕНИЙ**, ТЗ, **ЛАБОРАТОРИЯ**, ТП, ТС, УП., (НСД), ЭЛТ, **УСТОЙЧИВОСТИ**, АС:, **НОВЫХ**, (ФИО, МВД, **КОМПЛЕКСУ**, ПЭВМ, **ФУНКЦИОНИРОВАНИЯ**, **АРХИТЕКТУРА**, ЭВМ), **СРЕДСТВ**, ГБ, (СЗИ)., (**ПОСТРОЕНИЯ**, **ИНФОРМАЦИОННЫХ**, ОКГУ, МБ, ЗИП., РД, ЧТЗ, **ВХОДЯЩИХ**, **НА**, **ПОЛИТИКА**, **БЕЗОПАСНОСТИ**, УХЛ, ОАО, ФССП, «ИНН»., **АКЦИОНЕРНОЕ**, **ОБЩЕСТВО**, ЭЦП, ИНН, **ВОЗМОЖНЫХ**, **ЗАКРЫТОЕ**, **ТЕХНИЧЕСКОЕ**, ФМС, **СОГЛАСОВАННО**, ИУПП., ОС, **ТЕХНИЧЕСКИХ**, ФНС, ФСТЭК), (**ПОДСИСТЕМЫ**, **СИСТЕМУ**, **СИСТЕМЫ**, **СОГЛАСОВАНО**, **ПОДСИСТЕМ**
Графематический и морфологический анализ с использованием инструментов GP и JMorfSdk	**ОКГУ, ОС, ЕСМЭВ, ЭЛТ, ОЗУ, ИУПП, ИТ, ЧТЗ, ГК-93-ОФ, АС, ЗИП, МВД, СЗИ, ПЭВМ, ФНС, ФСТЭК, УП, ГБ, СВТ, ОАО, БД, ФСБ, ЛВС, ГОСТ, ЭВМ, УХЛ, ФССП, СНиП, РД, КПП, МБ, ТЗ, ЭЦП, ИНН, ФМС, ФИО, ТП, ТС, СНИЛС, НСД**

Поиск похожих документов



Реферирование

15-летняя российская фигуристка Александра Трусова выиграла произвольную программу на командном турнире Japan Open в Сайтаме.

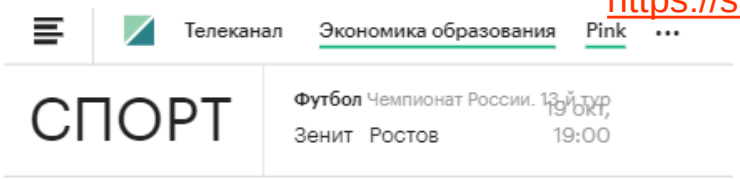
Второе место в произвольной программе заняла олимпийская чемпионка Алина Загитова (154,41 балла), также выступавшая за сборную Европы.

На Мемориале Ондрея Непелы она получила 163,78 балла за произвольную программу и 238,69 балла в сумме двух прокатов.

В сентябре 2018 года олимпийская чемпионка получила на турнире серии «Челленджер» в немецком Оберстдорфе 158,50 балла в произвольной программе и 238,43 балла в сумме.

РБК

https://sportrbc.ru/news/5d9849119a7947e260733c48



ГЛАВНОЕ Футбол Хоккей Единоборства Теннис Формула-1 Другие
Рейтинг букмекеров

05 ОКТ, 10:46 33 632

Фигуристка Трусова опередила Загитову на турнире в Японии

Российские спортсменки выступали за сборную Европы на командном турнире Japan Open в Сайтаме

SECR

Развитие фреймворка TAWT

- **Расширение набора правил**, реализация правил на основе **моделей управления** словами в модуле семантико-синтаксического анализа
- Разработка **новых модулей** комплекса и **расширение** набора **структур данных**
- **Проведение исследований** в области компьютерной лингвистики для выявления надежных **критериев поиска похожих по смыслу** текстовых **документов** с учетом их **особенностей** (стиля текста, соотношения объема текстов, решаемой задачи и др.)
- Реализация **модулей** морфологического и семантико-синтаксического анализа для **Lucene**
- Реализация и предоставление **веб-API** на базе фреймворка и веб-сервиса

Где использовался фреймворк?

Инструменты анализа технической документации:

- валидация структуры и аббревиатур;
- поиск и сокращение объема данных для ознакомления.

Прикладные системы:

- Android-приложение TouristHelper 2.0;
- плагин для Chrome FraudHunter;
- сервис поиска людей по интересам FriendFinder;
- сервис реферирования текстов;
- сервис подбора тематических синонимов.

Создание программных средств для решения задач:

- выделения именованных сущностей;
- выделения ключевых слов и словосочетаний;
- классификации текстов;
- развертывания сокращений;
- реферирования текстов;
- разметки корпусов текстов;
- построения семантической сети по толковым словарям.

Фреймворк TAWT

Подключение зависимостей в Maven:

```
<dependencies>
    <dependency>
        <groupId>com.github.jalexpr</groupId>
        <artifactId>tawt</artifactId>
        <version>master-SNAPSHOT</version>
    </dependency>
</dependencies>
```

Подключение репозитория:

```
<repositories>
    <repository>
        <id>jitpack.io</id>
        <url>https://jitpack.io</url>
    </repository>
</repositories>
```

Модули фреймворка:

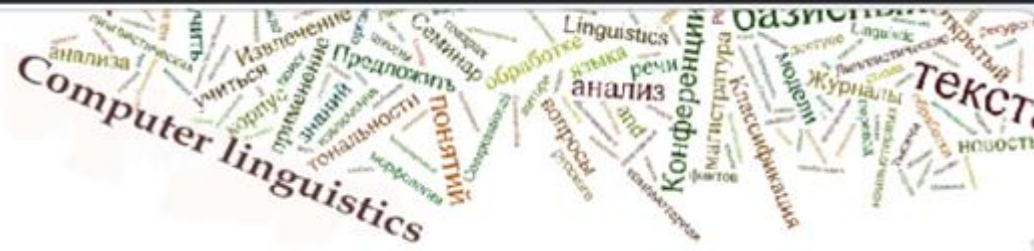
https://github.com/jalexpr/graphematic-parser

https://github.com/jalexpr/jmorfsdk

https://github.com/jalexpr/gama

https://github.com/jalexpr/SPN

Лицензия: Apache License 2.0



СПАСИБО ЗА ВНИМАНИЕ!

Ekaterina Politsyna
Sergey Politsyn
Alexander Porechny

http://textanalysis.ru/
tasystem@yandex.ru