# Система анализа тональности для русского языка (SDK) v2.1.1 для .NET и Java

## Руководство разработчика и системного администратора

SemanticAnalyzer, 2014-12-01

SemanticAnalyzer
Language intelligence

## Содержимое демо-пакета

1. SDK системы анализа тональности для .NET и Java
2. Тональные словари: негативной и позитивной полярностей
3. Лемматизатор для русского языка
4. Словари лемматизатора
5. Файлы с конфигурацией
6. Dll и jar библиотеки
7. Демо C# и Java приложения, иллюстрирующие использование SDK
8. Данное руководство
9. Лицензия

## Инсталляция

Распакуйте zip пакет с SDK в любую удобную папку на диске. Путь к папке не должен содержать пробелов. В данном руководстве мы используем для примера папку на диске c:/semanticanalyzer.

Откройте C# проект и отредактируйте путь в переменной testAnnotationsFile и confFileName. Добавьте все dll из папки dll в качестве references в проект. Скомпилируйте и запустите проект. В случае успешной инсталляции, вы должны увидеть на экране следующие строки:

POSITIVE

NEGATIVE

NEUTRAL

NEUTRAL

NEUTRAL

POSITIVE

OBJECTIVE

SUBJECTIVE

Для проверки Java SDK откройте java проект и проимпортируйте все jar файлы из подпапки lib. Отредактируйте путь к конфигурационному файлу системы в конструкторе класса SentimentEngineTest. Скомпилируйте и запустите приложение. Все тесты должны отработать верно.

SemanticAnalyzer
Language intelligence

## Введение в задачу определения тональностей

Определение тональностей (sentiment detection) в тексте на естественном языке является задачей области искусственного интеллекта. Постановка задачи может варьироваться в зависимости от преследуемых целей и предметной области. Выделяются два основных подхода к решению задачи: системы на основе машинного обучения с использованием размеченных данных (machine learning) и системы с набором лингвистических правил (rule-based systems).

Основными направлениями работы модуля определения тональностей являются:

- **Полярность**: определение класса полярности (позитивная, негативная, нейтральная или смешанная) к которому относится данное слово, конструкция, предложение или текст.
- **Сила тональности**: низкая (например, «ничего»), средняя (например, «слабовато») и высокая (например, «ужасно»).
- **Цель тональности:** тема (бренд, продукт и т.д.), на которую направлена тональность
- **Носитель тональности:** субъект, передающий тональность (часто автор высказывания или текста)
- **Субъективность / объективность:** отделение утверждений спекулятивного (сугубо субъективного) характера от объективных утверждений

Определение тональности ведётся на разном уровне детализации (granularity): уровне текста или уровне текста, в зависимости от конкретной задачи. Так, для выявления тональности по обзору кинофильма / книги может оказаться достаточным уровень текста (документа). В случае обзора продукта уровень предложения более уместен.

## Краткое описание принципов работы системы

Система относится к классу лингвистических пра́виловых систем, при этом она имеет элементы машинного обучения. Ядро системы содержит набор лингвистических правил, описывающих основные связи в предложении на русском языке. Компонентом машинного обучения в системе выступает алгоритм составления словаря полярностей для данной предметной области. Основной отличающейся от систем-аналогов характеристикой в текущей версии системы является возможность разметки тональности в отношении к входному предикату, который зачастую является названием бренда компании или её продукта, физическим лицом и т.д. Таким образом, можно отследить не только обобщённое распределение тональностей в корпусе текстов, но и построить карту тональностей в отношении к заданному поисковому предикату. Приведём пример (не передающий отношения авторов):

> Мне понравился новый iPhone, но вот GalaxyS неудобный.

| **Предикат** | **Тональность** |
|---|---|
| iPhone | positive |
| GalaxyS | negative |
| - (отсутствует) | neutral (в данном случае, смешанная) |

В текущей версии система умеет оценивать полярность (слова, конструкции предложения или текста) и цель тональности.

Информация о полярности отдельных слов находится в соответствующих словарях. Все слова содержатся в базовой форме. Примеры словарных статей:

**Позитивная полярность:**

благо

благой

вкусный

грандиозный

…

классный

красивый

красота

…

рекомендовать

…

симпатичный

скидка

**Негативная полярность:**

грустить

дождь

дорого

…

жесткий

жесть

жуткий

…

подделка

подстава

покраснеть

поломаться

…

скучный

## .NET API

Главный класс для доступа к системе: SentimentEngine. Объект этого класса можно получить, передав путь к конфигурационному файлу системы анализа тональности:

```
static string confFileName = “c:\\semanticanalyzer\\conf\\sentiment-module.properties”;
SentimentEngine sentimentEngine = new SentimentEngine(new java.io.File(confFileName));
```

Класс SentimentEngine имеет 5 открытых методов для определения тональности:

```
public Enumerations.Sentiment detectPolarityOfText(string Text)
```

```
        {
        }

        public Enumerations.Sentiment detectPolarityOfTextForKeywords(string Text,
List<String> SingleObject)
        {
        }

        public Enumerations.Sentiment detectPolarityOfTextForSynonyms(string Text,
List<List<String>> Synonyms)
        {
        }

        public Enumerations.Subjectivity isSubjective(string Text)
        {
        }

        public Enumerations.Subjectivity isSubjectiveForKeywords(string Text,
List<String> SingleObject)
        {
        }
```

`Text` содержит текст либо предложение, которые должны быть аннотированы тэгом тональности либо субъективности.

`SingleObject` и `Synonyms` указывают на целевой объект, по отношению к которому должна быть проанализирована тональность либо субъективность. Обратите внимание, что в случае `Synonyms` первый найденный вариант написания объекта в тексте будет использован анализатором тональности.

Возвращаемые значения имеют следующие метки (в зависимости от вызванного метода):

```
public enum Sentiment {
    POSITIVE,
    NEGATIVE,
    NEUTRAL,
    UNKNOWN
}

public enum Subjectivity {
    OBJECTIVE,
    SUBJECTIVE,
    UNKNOWN
}
```

Метка UNKNOWN в обоих списках меток возникает в случае внутренней ошибки системы. Свяжитесь с поддержкой и передайте входные параметры для исследования и исправления проблемы, также сообщения ошибок, которая вывела система.

Все 5 методов самостоятельно сегментируют текст на предложения. Однако предусмотрена возможность сегментации текста таким образом, который наиболее подходит для контекста вашей задачи. В этом случае предложения передаются в систему одно за другим. В этом случае отдельные метки тональности либо субъективности будут возвращены клиентскому коду для каждого предложения.

Несколько слов об анализе на субъективность. Если текст либо предложение содержит любую эмоциональность, он помечается как субъективный. В этом случае, не имеет значения, какая именно тональность присутствует в тексте: позитивная, негативная, либо смешанная. Если и только если текст не содержит никакой тональности и эмоциональной окраски, он помечается, как объективный.

## Java API

Для правильного функционирования модуля в виде библиотеки на сервере необходим пакет JRE7 или более свежей версии(http://www.java.com/en/download/help/download_options.xml).

Главный класс для доступа к системе: SentimentEngine. Для его инстанциирования необходимо передать путь к конфигурационному файлу системы анализа тональности:

```
sentimentEngine = new SentimentEngine(new File("c:/semanticanalyzer/conf/sentiment-module.properties"));
```

Доступные методы для анализа тональности либо субъективности те же, что и в API для платформы .NET. Пример:

```
Enumerations.Sentiment sentiment = sentimentEngine.detectPolarityOfText("Сегодня хорошая погода"); // возвращает метку Enumerations.Sentiment.POSITIVE
```

## Тональные словари

Тональные словари поставляются в зашифрованном бинарном виде:

dict/sentiment.neg.norm.encrypted содержит негативные полярные единицы
dict/sentiment.pos.norm.encrypted содержит позитивные полярные единицы

Пользовательские тональные словари поддерживаются в незашифрованном виде, как текстовые файлы. Одна строка содержит одну тональную единицу (слово в базовой форме).

## Конфигурационные файлы

Конфигурационный файл системы анализа тональности находится в файле conf/sentiment-module.properties. Он содержит четыре записи:

```
# lemmatizer's properties
lemmatizer.props=c:/semanticanalyzer/conf/lemmatizer-ru.properties
# sentiment module's dictionaries directory
```

```
sentiment.dict.dir=c:/semanticanalyzer/dict/
# files with user polarity
sentiment.dict.user.dir=c:/semanticanalyzer/dict_user/
# license file
sentiment.dict.dir=c:/semanticanalyzer/license/licence.lic
```

Здесь указаные относительные пути. Они могут быть указаны и как абсолютные. Однако, обратите внимание, что только прямой слэш(”/”) может быть использован в качестве разделителя пути в файловой системе.

Конфигурационный файл лемматизатора находится в файле conf/lemmatizer-ru.properties. Он содержит следующие записи:

```
# enabled morphological dictionaries in order of analysis algorithm
morph.dics=user main lexgroup wkt

morph.dic.main.type=morph
morph.dic.main.stem=c:/semanticanalyzer/data/morph-dics-enc/a
morph.dic.main.fok=c:/semanticanalyzer/data/morph-dics-enc/b
morph.dic.main.encrypted=true

morph.dic.lexgroup.type=morph
morph.dic.lexgroup.stem=c:/semanticanalyzer/data/morph-dics-enc/lga
morph.dic.lexgroup.fok=c:/semanticanalyzer/data/morph-dics-enc/lgb
morph.dic.lexgroup.encrypted=true

morph.dic.wkt.type=morph
morph.dic.wkt.stem=c:/semanticanalyzer/data/morph-dics-enc/wa
morph.dic.wkt.fok=c:/semanticanalyzer/data/morph-dics-enc/wb
morph.dic.wkt.encrypted=true

morph.dic.user.type=example
morph.dic.user.file=c:/semanticanalyzer/data/morph-dics/UserDict.txt

morph.compoundprefixes=c:/semanticanalyzer/data/morph-dics/compound_prefixes.txt
guesser.enable=true
```

Пользовательский словарь выделен жирным шрифтом. Он содержит новые слова (слева от символа табуляции) со ссылкой на известные лемматизатору слова (справа от символа табуляции) для описания парадигмы склонения:

```
# new-word<TAB>example-word
бичевый     полный
нокия партия
че-то что-то
чо-то что-то
вай-фай     пай
твиттер     свитер
```