

Использование Searchable DataStore для поиска закономерностей

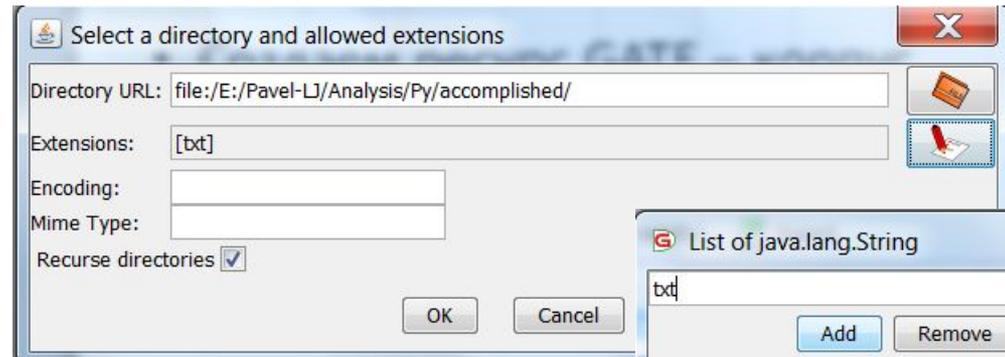
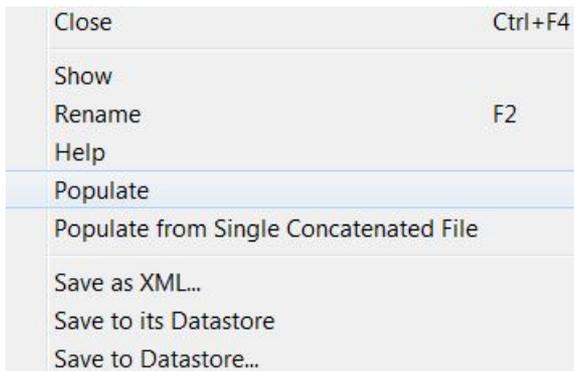
А.В.Поршнеv

Создаем корпус файлов

- Создаем ресурс GATE – корпус



- Наполняем корпус файлами

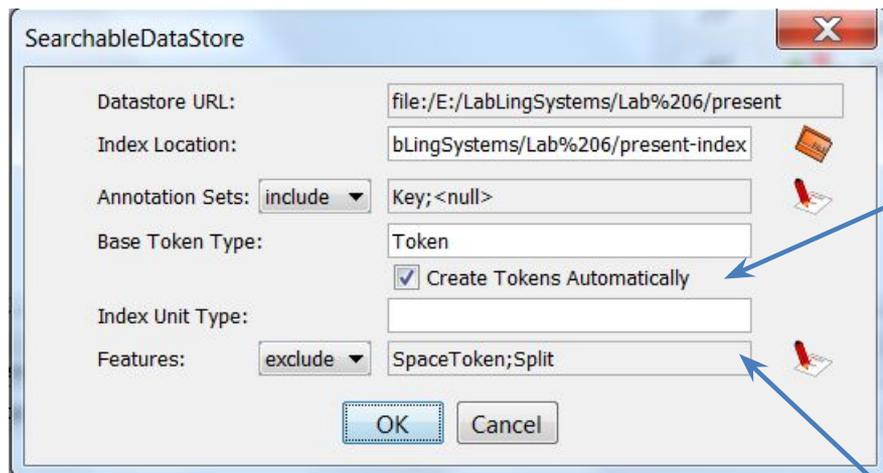
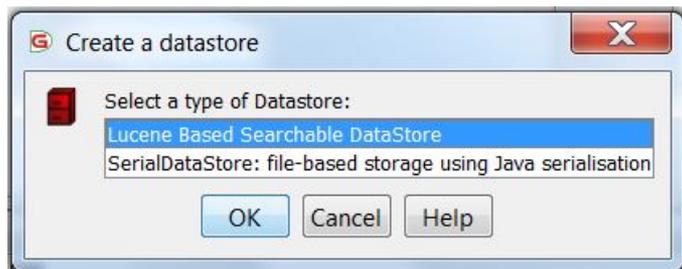


Обрабатываем корпус

- Создаем стандартную последовательность обработки
- Что позволяет нам не только искать слова, но части речи, именованные

Selected Processing resources		
!	Name	Type
	 Document Reset PR	Document Reset PR
	 ANNIE Sentence Splitter	ANNIE Sentence Splitter
	 ANNIE English Tokeniser	ANNIE English Tokeniser
	 ANNIE Gazetteer	ANNIE Gazetteer
	 ANNIE POS Tagger	ANNIE POS Tagger
	 ANNIE NE Transducer	ANNIE NE Transducer

Создаем индексированную БД



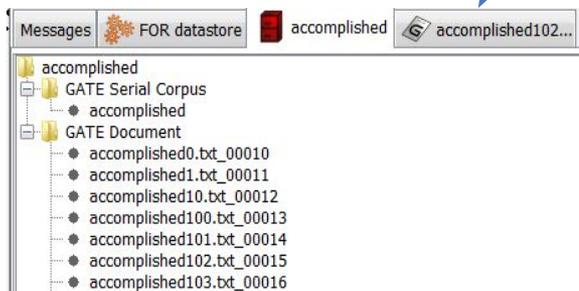
При создании индексов тесты м.б. автоматически разбиты на единицы, но чтобы иметь информацию о частях речи нужно применить к тексту браузеру создания индексов по умолчанию не включаются SpaceToken и Split

Значит нельзя будет поймать последовательность {Token}{SpaceToken}, но обычно в этом нет необходимости

У БД есть два вида

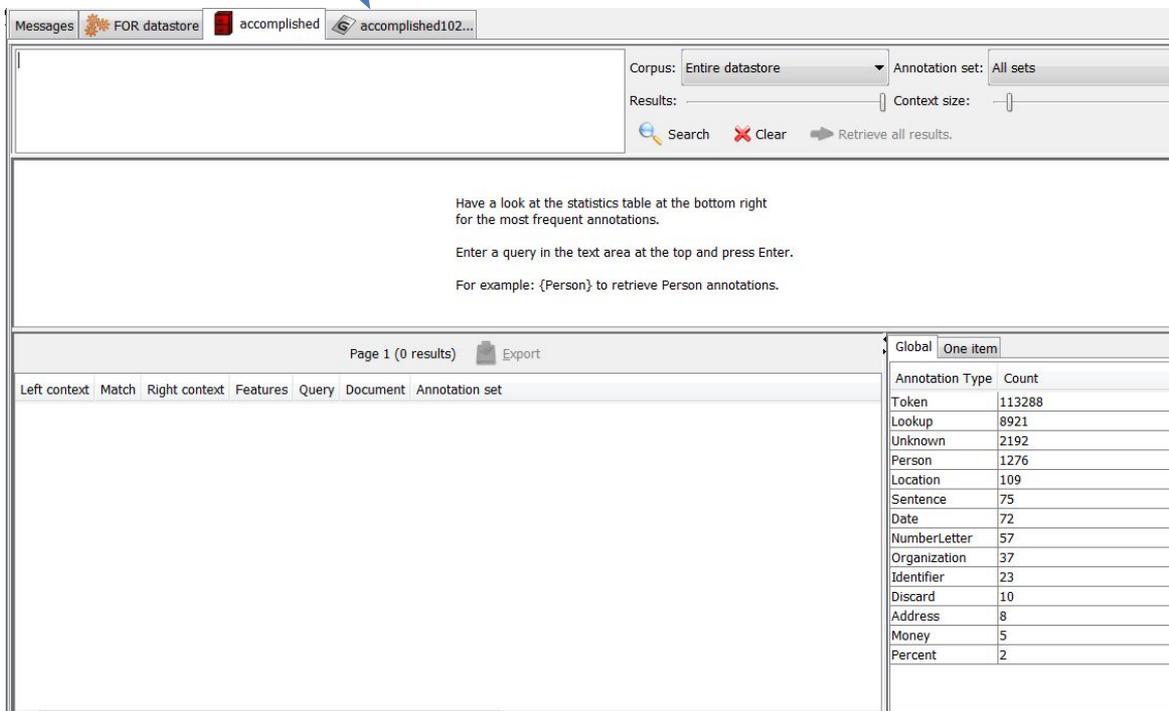
Serial Datastore Viewer

Lucene Datastore Searcher



Messages FOR datastore accomplished accomplished102...

- accomplished
 - GATE Serial Corpus
 - accomplished
 - GATE Document
 - accomplished0.bt_00010
 - accomplished1.bt_00011
 - accomplished10.bt_00012
 - accomplished100.bt_00013
 - accomplished101.bt_00014
 - accomplished102.bt_00015
 - accomplished103.bt_00016



Messages FOR datastore accomplished accomplished102...

Corpus: Entire datastore Annotation set: All sets

Results: Context size:

Search Clear Retrieve all results.

Have a look at the statistics table at the bottom right for the most frequent annotations.

Enter a query in the text area at the top and press Enter.

For example: {Person} to retrieve Person annotations.

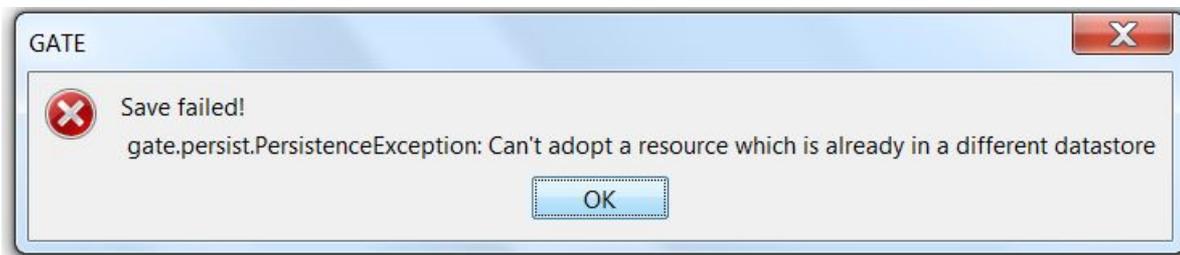
Page 1 (0 results) Export

Left context	Match	Right context	Features	Query	Document	Annotation set
--------------	-------	---------------	----------	-------	----------	----------------

Annotation Type	Count
Token	113288
Lookup	8921
Unknown	2192
Person	1276
Location	109
Sentence	75
Date	72
NumberLetter	57
Organization	37
Identifier	23
Discard	10
Address	8
Money	5
Percent	2

Внимание

- Если корпус сохранен в одном представлении, нельзя его сохранить в другом



Простой поиск

Слово или фраза для поиска

Сколько результатов отображать на одной странице

Annotations in the image:

- Blue arrow pointing to the search box: Слово или фраза для поиска
- Blue arrow pointing to the 'Results' field: Сколько результатов отображать на одной странице
- Green circle around the search term 'happy': Слово или фраза для поиска
- Orange arrow pointing to the 'Context size' slider: Размер контекста
- Purple arrow pointing to the context window: Слово или фраза для поиска

Left context	Match	Right context	Features	Query	Document
. What starts as a	happy	twist of fate ends up	Token.category=JJ	happy	accomplished208.txt_0008A__14317904
. What starts as a	happy	twist of fate ends up	Token.category=JJ	happy	accomplished208.txt_0008A__14317904
Which was totally not a	happy	memory. ВЪНEnough to know	Token.category=JJ	happy	accomplished266.txt_000CA__14317905
Which was totally not a	happy	memory. ВЪНEnough to know	Token.category=JJ	happy	accomplished266.txt_000CA__14317905

Annotation Type	Count
Token	478174
Lookup	37687
Unknown	9962
Person	5064
Location	1283

Что можно найти

Messages FOR datastore accomplished accomplished102... accomplished

happy

Corpus: accomplished Annotation set: All sets

Results: Context size:

Search Clear Retrieve all results.

Context . What starts as a happy twist of fate ends up

Token.string What starts as a happy twist of fate ends up

Configure

Page 1 (108 results) Export

Global	One item
Token.category=	tation Type Count
happy	i 478174
JJ	p 37687
Token.category=	wn 9962
happy	n 5064
JJ	on 1283

Left context

- . What starts as
- . What starts as
- Which was totally not
- Which was totally not

What starts as a

Which was totally not a

as that makes me a

I am

happy Token.category= JJ

happy Token.category= JJ

happy Token.category= JJ

happy Token.category= JJ

Теперь чуть сложнее

- Можно задавать паттерны, как в правилах JARE

Например

Вместо

not a happy

```
{Token.string=="not"}{Token=="a"}{Token=="happy"}
```

Или чуть шире

```
{Token.string=="not"}{Token=="a"}{Token=="JJ"}
```

Или еще шире

```
{Token.category=="RB"}({Token.category=="DT"})?{Token.category=="JJ"}
```

Можно экспортировать результаты

Annic Results and Statistics

Parameters

- Corpus: **Entire datastore**
- Annotation set: **All sets**
- Query Issued: **{Token.category=="JJ"}**
- Context Window: **5**

Results

Left context	Match	Right context	Features	Query	Document
on and give it a	read-through	. I'm extraordinarily proud	Token.category=JJ	{Token.category=="JJ"}	accomplished118.txt_00026__143
of it. In a	few	years, we can do	Token.category=JJ	{Token.category=="JJ"}	accomplished118.txt_00026__143
on and give it a	read-through	. I'm extraordinarily proud	Token.category=JJ	{Token.category=="JJ"}	accomplished118.txt_00026__143
of it. In a	few	years, we can do	Token.category=JJ	{Token.category=="JJ"}	accomplished118.txt_00026__143
comm. I'm a	new	icon maker, so I	Token.category=JJ	{Token.category=="JJ"}	accomplished131.txt_00035__143

Применение в лабораторной работе 8

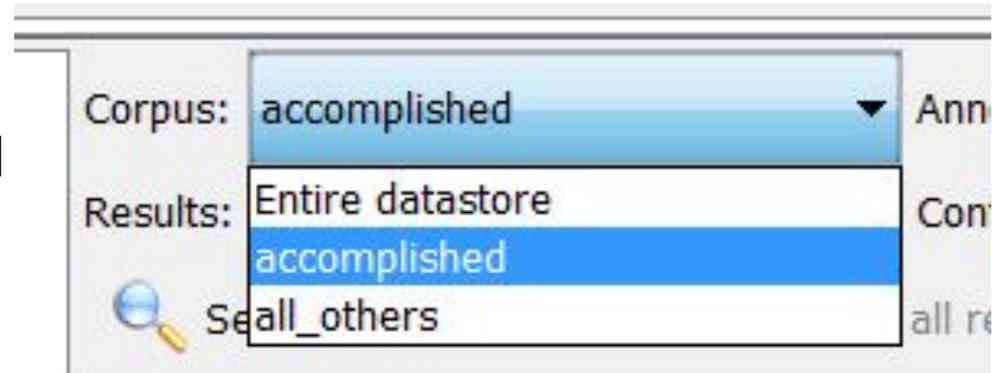
1. Проанализировать частоты встречаемости прилагательных, глаголов и т.д.
2. Проанализировать частоты встречаемости грамматических паттернов в выбранной категории настроений
3. Попытаться выявить паттерны характеризующие настроение (i feel happy, oh so happy)
4. Определить частоты встречающихся слов
5. Определить наиболее информативные слова по Mutual Information Criteria
6. Определить наиболее информативные паттерны
7. Выявить наиболее информативные паттерны учитывающие содержание слов

Mutual Information Criteria

	C_{happy}	$C_{\text{all other}}$	N
$W_{\text{happy}=1}$	N_{11}	N_{10}	N_1
$W_{\text{happy}=0}$	N_{01}	N_{00}	N_0

$$MI = \frac{N_{11}}{N} \log_2 \frac{NN_{11}}{N_1.N_1} + \frac{N_{01}}{N} \log_2 \frac{NN_{01}}{N_0.N_1} + \frac{N_{10}}{N} \log_2 \frac{NN_{10}}{N_1.N_0} + \frac{N_{00}}{N} \log_2 \frac{NN_{00}}{N_0.N_0}$$

- Делаем два корпуса БД – выбранное настроение и все другие
- Тогда можно выбрать и получить частоты



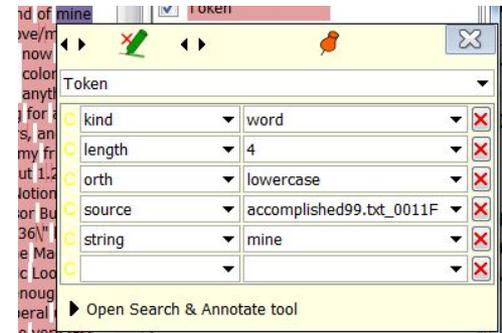
Расчет MIC

- Встречаемость считается по всей коллекции, а для расчета MIC требуется знать в скольких документах встретилось искомое

1. способ – использовать данные из файла экспорта

Features	Query	Document	Annotation
Token.category=JJ	{Token.category=="JJ"}	accomplished118.txt_00026__1431790404480__5714	Key

2. способ – поместить метку документа, потом посчитать кол-во неповторяющихся меток



3. способ написать программу на JAVA с использованием средств GATE

Код для добавления метки документа

Phase: firstpass

Input: Token

Options: control = appelt

Rule: AddDocName

({Token}): t-Token

--> {

AnnotationSet AS-Token = bindings.get("t-Token");

Annotation A-Token = AS-Token.iterator().next();

FeatureMap newAnnFeatures = Factory.newFeatureMap();

newAnnFeatures.putAll(A-Token.getFeatures()); // перезаписываем все свойства

newAnnFeatures.put("source", doc.getName()); // добавляем новое - source

inputAS.remove(A-Token); // Убираем Token которые мы изменяли, чтобы потом добавить

outputAS.add(AS-Token.firstChild(), AS-Token.lastNode(), "Token", newAnnFeatures);

// Добавляем разметку Token

}