OŚRODEK PRZETWARZANIA INFORMACJI
PAŃSTWOWY INSTYTUT BADAWCZY

www.opi.org.pl

Textual Information Retrieval

„NLP DAY – Marek Kozłowski, Maciej Kowalski”

WARSZAWA, 19.10.2018

Agenda

1. Information Retrieval
2. Classical approaches from Algorithms & Data Structures
3. NLP components
4. Lucene Search – as the efficient implementation of an inverted index
5. Keywords Indexing using Babelfy
6. Keywords Indexing using Word2Vec
7. Elasticsearch and Kibana in practice

What will we use during workshop?

Binaries:

- https://www.elastic.co/downloads/elasticsearch
- https://www.elastic.co/downloads/kibana

Data:

- https://drive.google.com/file/d/1fMt0OdwpQRzBbaUX6eqGg-_O5UiJriBy/view

Code:

- (part one – lucene, babelfy, w2v): https://github.com/dotjabber/nlpday/tree/master/part1
- (part two – elasticsearch, kibana): https://github.com/dotjabber/nlpday/tree/master/part2

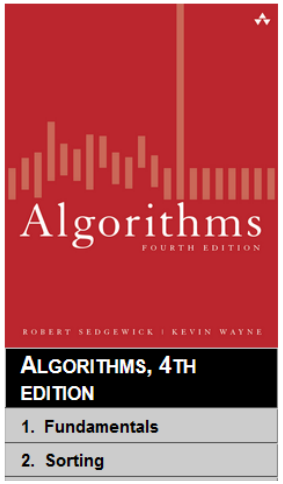
Information Retrieval

- Information retrieval is the science of searching for information in a document, searching for documents themselves, and also searching for metadata that describe data even non-textual formats.

- Textual Information Retrieval is the activity of obtaining documents/texts relevant to an information need from a collection of documents.

- Searches can be based on full-text or other content-based indexing as specified keywords.

Classical methods - examples



6.3 Suffix Arrays

This chapter under major construction.

Important note. *Beginning with Oracle and OpenJDK Java 7, Update 6, the* `substring()` *method takes* linear time and space *in the size of the extracted substring (instead of constant time and space). The* String API *provides no performance guarantees for any of its methods, including* `substring()` *and* `charAt()`.

The programs in the textbook and booksite have been updated to avoid any dependency on a constant-time substring operation. However, if you are using the third printing of the textbook (or earlier), consider yourself warned.

Suffix sorting and suffix arrays. Suffix sorting: given a string, sort the suffixes of that string in ascending order. Resulting sorted list is called a *suffix array*. Program SuffixArray.java builds a sufix array data structure.

```
/*****************************************************************************
 *  Compilation:  javac SuffixArray.java
 *  Execution:    java SuffixArray < input.txt
 *  Dependencies: StdIn.java StdOut.java
 *  Data files:   https://algs4.cs.princeton.edu/63suffix/abra.txt
 *
 *  A data type that computes the suffix array of a string.
 *
 *   % java SuffixArray < abra.txt
 *    i ind lcp rnk  select
 *   ---------------------------
 *    0  11   -   0  "!"
 *    1  10   0   1  "A!"
 *    2   7   1   2  "ABRA!"
 *    3   0   4   3  "ABRACADABRA!"
 *    4   3   1   4  "ACADABRA!"
 *    5   5   1   5  "ADABRA!"
 *    6   8   0   6  "BRA!"
 *    7   1   3   7  "BRACADABRA!"
 *    8   4   0   8  "CADABRA!"
 *    9   6   0   9  "DABRA!"
 *   10   9   0  10  "RA!"
 *   11   2   2  11  "RACADABRA!"
```

```
/**
 *  The {@code LongestCommonSubstring} class provides a {@link SuffixArray}
 *  client for computing the longest common substring that appears in two
 *  given strings.
 *  <p>
 *  This implementation computes the suffix array of each string and applies a
 *  merging operation to determine the longest common substring.
```

Classical methods - examples

1. Jaccard Index – a similarity measure between finite sets of items
2. Texts are represented as set of tokens (uni-gram, bi/tri -grams...)

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

NLP components

1. Flexion -> in grammar, inflection is the modification of a word to express different grammatical categories such as tense, case, voice, aspect, person, number, gender, and mood etc.
2. Part of Speech Tags – sometimes some parts are more useful

```
/***
 * Interfejs filtra NLP obejmuja kontraktem: pobieranie tokenow, lematow, i tylko pasujacych do PoS patternu tokenow
 */
public interface NLPFilter {
    List<String> getTokens(String text);
    List<String> getLemmas(String text);
    List<String> getPoSMatchingTokens(String text, PoSPart pos);
}
```

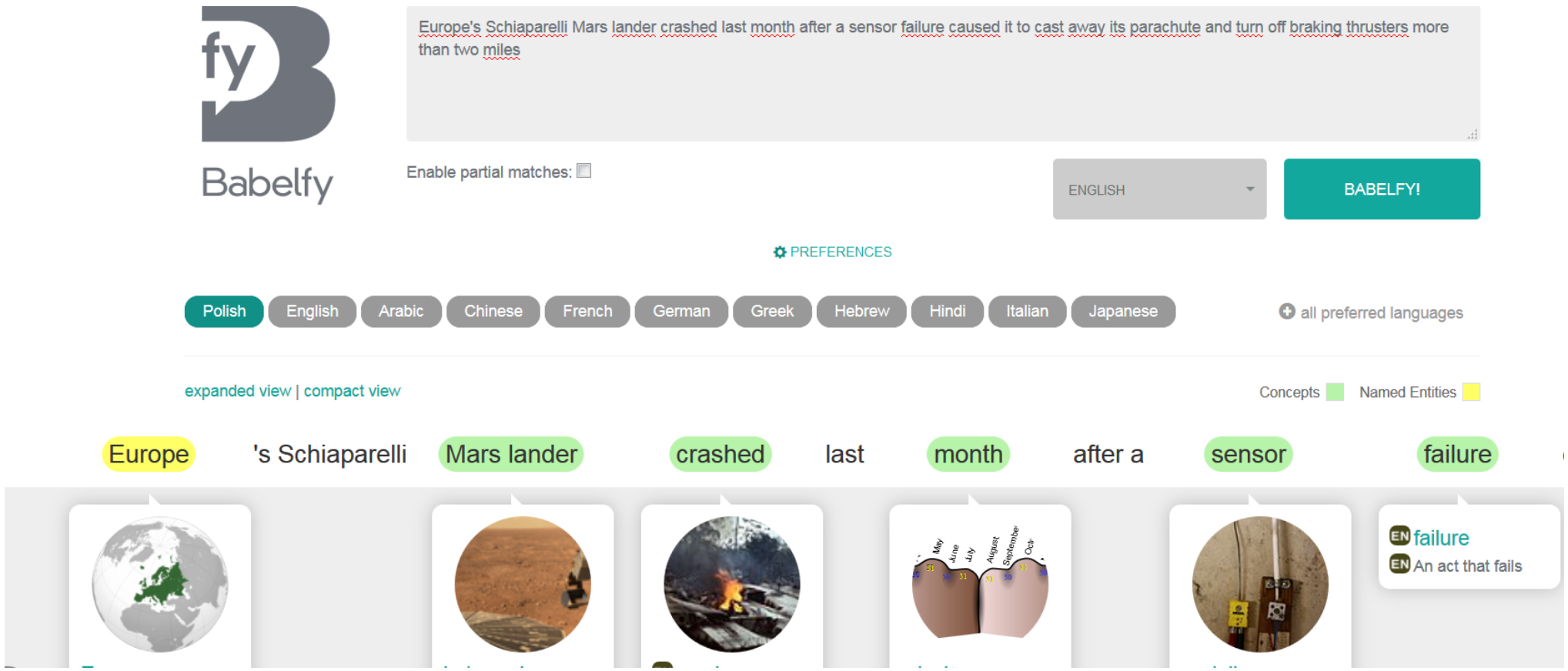
Inverted Index

1. IT is an index data structure storing a mapping from words to its locations in a corpora (set of documents), named in contrast to a forward index, which maps from documents to content.
2. The purpose of an inverted index is to allow fast full text searches, at a cost of increased processing

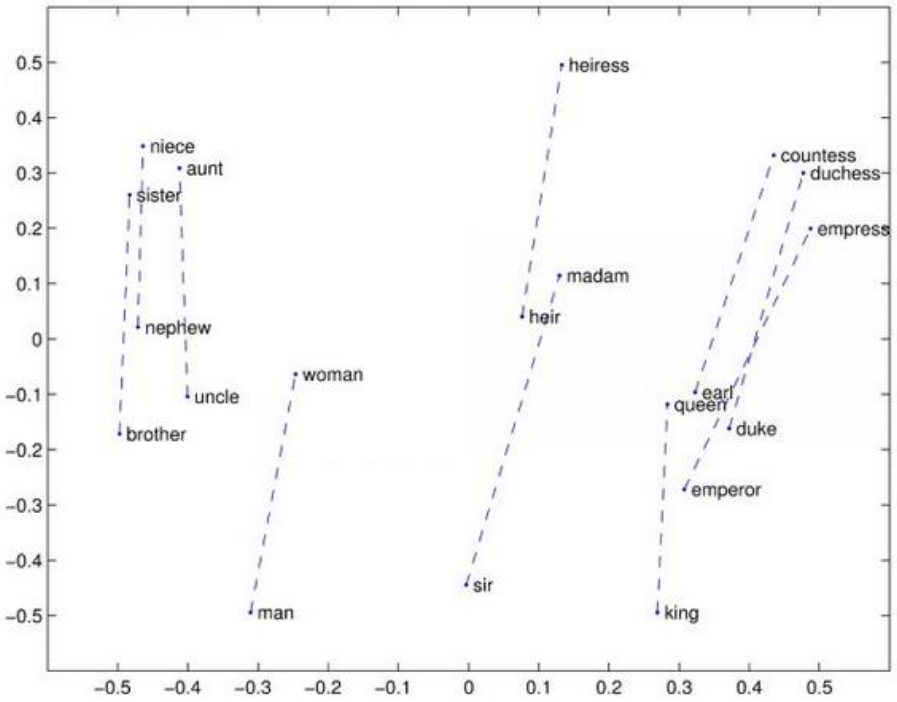
Apache Lucene™ is a high-performance, full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform.

Apache Lucene is an open source project available for free download. Please use the links on the right to access Lucene.

Babelfy as a Knowledge&Content-Based Indexer

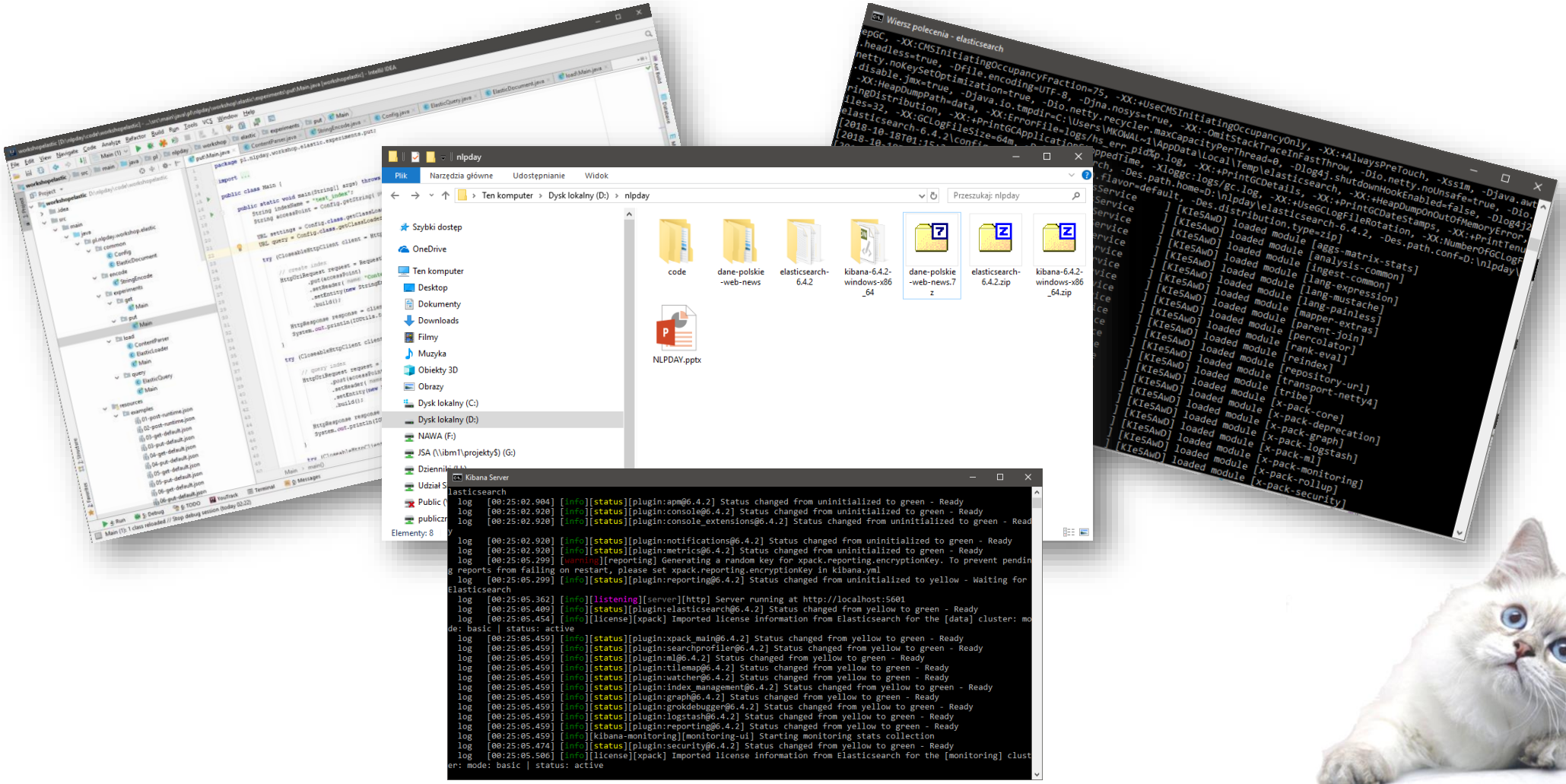


Word2Vec in DL4J as Sparse Content Based Indexer




elastic
kibana

Let’s check part two…



Analyzers, Tokenizers, Filters, Normalizers…

- https://www.elastic.co/guide/en/elasticsearch/reference/current/analyzer-anatomy.html

Standard
Pattern
Whitespace
Fingerprint

Mappings
External Resouces

StopToken
ASCIIFolding
Stemmer
Length

How can I check it?


```
PUT my_index
{
    "settings": {
        "analysis": {
            "analyzer": {
                "std_folded": {
                    "type": "custom",
                    "tokenizer": "standard",
                    "filter": [ "lowercase"]
                }
            }
        }
    },
    "mappings": {
        "_doc": {
            "properties": {
                "my_text": {
                    "type": "text",
                    "analyzer": "std_folded„
                }
            }
        }
    }
}
```

http://127.0.0.1:9200/my_index/_analyze?pretty

+

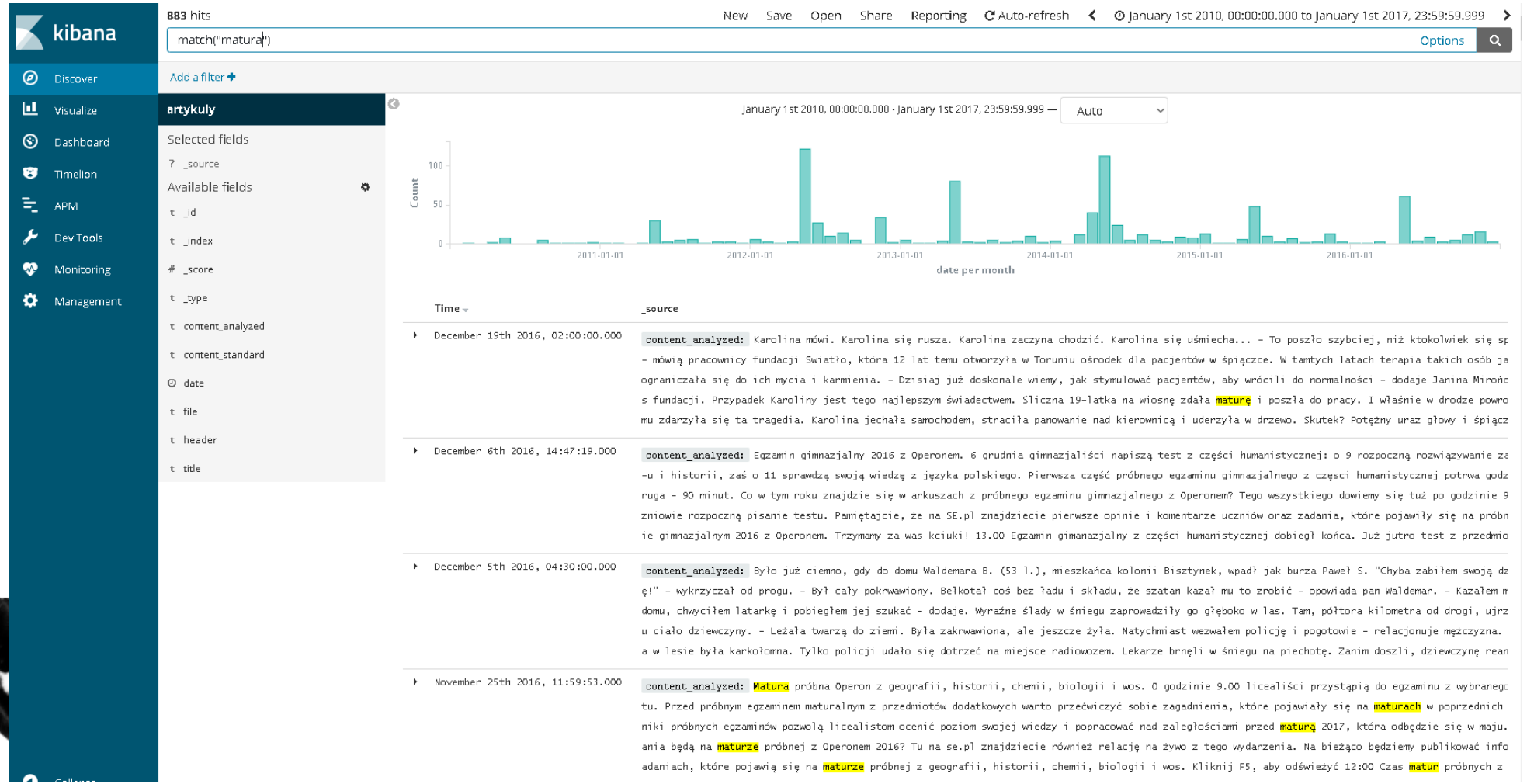
```
POST _analyze
{
    "tokenizer": "standard",
    "filter": [ "lowercase", "asciifolding"],
    "text": "Is this déja vu?"
}
```

```
POST _analyze
{
    "tokenizer": "standard",
    "filter": [ "lowercase", "asciifolding"],
    "text": "Is this déja vu?"
}
```

http://127.0.0.1:9200/_analyze?pretty

- https://www.elastic.co/guide/en/elasticsearch/reference/current/_testing_analyzers.html

Is it really necessary?



Teh Questions?