Task 2.4:

- Language Identifier Module

- Query Translation Module

Alessio Bosca (alessio.bosca@celi.it) Luca Dini (dini@celi.it)

Language Identifier

- Key resource in order to enable multilinguality within Europeana.
- Given a chunk of text (metadata, user queries, ..) returns a ranked list of languages along with their probability.
- Needed :
  - at indexing time → detect metadata language and create language aware search indexes (i.e. title_en, title_de, ..)
  - at runtime → detect the language of user query in order to search in the proper fields and in order to enable query translation

Language Identifier Prototype

- Current prototype integrates different strategies with a voting mechanism:
  - **Corpus based frequency voter**: computes the frequency list associated to terms from language specific corpora specified at initialization time (now wikipedia abstracts).
  - **Character based voter**: computes the distance of character based language models from the text to be guessed. Models are automatically computed from the input corpora.
  - **StopWord based voter**: search for language specific stopwords
  - **Prior knowledge based voter**: collects a set of hypotesis and heuristics like region/browser language, main language of the collection, ..
- Each voter has a configurable weight in order to adapt the component to the different use cases (indexing time and runtime)

Configuration/Initialization

```
<?xml version="1.0" encoding="UTF-8"?>
<LanguageIdentifierConfig>
<TextCategorizer textCategorizerConfigFilePath="/work/Sviluppo/LanguageGuesser/fingerprints/my.conf"
    createFingerprintsFromCorpuses="false" />
<Corpus indexPath="/Indexes/itwiki" lang="it" field="summary"/> <!-- Lucene Indexes -->
<Corpus indexPath="/Indexes/frwiki" lang="fr" field="summary"/>
<Corpus indexPath="/Indexes/eswiki" lang="es" field="summary"/>
<Corpus indexPath="/Indexes/enwiki" lang="en" field="summary"/>
<Corpus indexPath="/Indexes/dewiki" lang="de" field="summary"/>
<Corpus indexPath="/Indexes/plwiki" lang="pl" field="summary"/>
<StopWordList stopWordsFilePath="/work/Sviluppo/LanguageGuesser/stopWords/englishST.txt" lang="en" />
<StopWordList stopWordsFilePath="/work/Sviluppo/LanguageGuesser/stopWords/germanST.txt" lang="de" />
<StopWordList stopWordsFilePath="/work/Sviluppo/LanguageGuesser/stopWords/italianST.txt" lang="it" />
<StopWordList stopWordsFilePath="/work/Sviluppo/LanguageGuesser/stopWords/frenchST.txt" lang="fr" />
<StopWordList stopWordsFilePath="/work/Sviluppo/LanguageGuesser/stopWords/polishST.txt" lang="pl" />
<StopWordList stopWordsFilePath="/work/Sviluppo/LanguageGuesser/stopWords/spanishST.txt" lang="es" />
</LanguageIdentifierConfig>
```

Language Identifier software API

- Functional requirements and software interface available on http://europeanalabs.eu/wiki/RequirementsLanguageIdentifier

- Current Proposal for software interface:
  - *TreeSet<RankedLanguage> guessLanguage(String text)*
  - *TreeSet<RankedLanguage> guessLanguage(String text, AlgorithmStrategy weights)*
  - *TreeSet<RankedLanguage> guessLanguage(String text, TreeSet<RankedLanguage> priorGuess)*
  - *TreeSet<RankedLanguage> guessLanguage(String text, TreeSet<RankedLanguage> weights, AlgorithmStrategy priorGuess)*

- AlgorithmStrategy allows for specifying the weight of the different voters.
  - Default weighting schemes used if none explicitly provided.

Example / Demo

- **Input**: “quadri Maigritte”
- **Analysis**:
    - stopword -> []
    - corpus -> [fr: 0.12612613, en: 0.0990991, pl: 0.027027028, it: 1.0, es: 0.027027028, de: 0.027027028,]
    - charachter -> [fr: 0.9760191, en: 0.8475337, pl: 0.0, it: 0.7570233, es: 0.27213466, de: 0.41461378,]
- **Results**:
    - it (0.727107)
    - fr (0.3558688)
    - en (0.30380967)
    - de (0.13789766)
    - es (0.09515391)
    - pl (0.013513514)

Open Issues

- Component integration in Europeana: two different approaches needed for language identification in the different use cases (indexing time and runtime).
    - **Indexing Time**. Proposal: customized Solr component (i.e. *UpdateRequestProcessor)* that creates language specific copyFields for the relevant language specific metadata.
    - **Runtime**: Proposal: customized Solr component (i.e. *RequestHandler)* that identifies the language and prepares the query targeted for the language specific fields.
- Tuning of the default voters weighting schemes for the use cases

Proposals? Suggestions?

Thank You!

Language Guesser Evaluation

- Proposal: participating to the CLEF track Log@CLEF:
  - The goal of LogCLEF is the analysis and classification of queries in order to understand search behavior in multilingual contexts and ultimately to improve search systems.
  - Test Data: both TEL search log and HTTP logs for the 2007/2008/2009 period

Query Translation System Overview

Objective: building translation modules that will process user queries and produce a suitable multilingual representation of the query in a set of target languages

Languages to be supported :

- ✓ **Core Languages**: *English, German, Spanish, French, Italian, Polish*
- ✓ **Secondary Languages**: *Portuguese, Hungarian, Swedish, Dutch*

Depends on the Europeana Language Resources Repository:

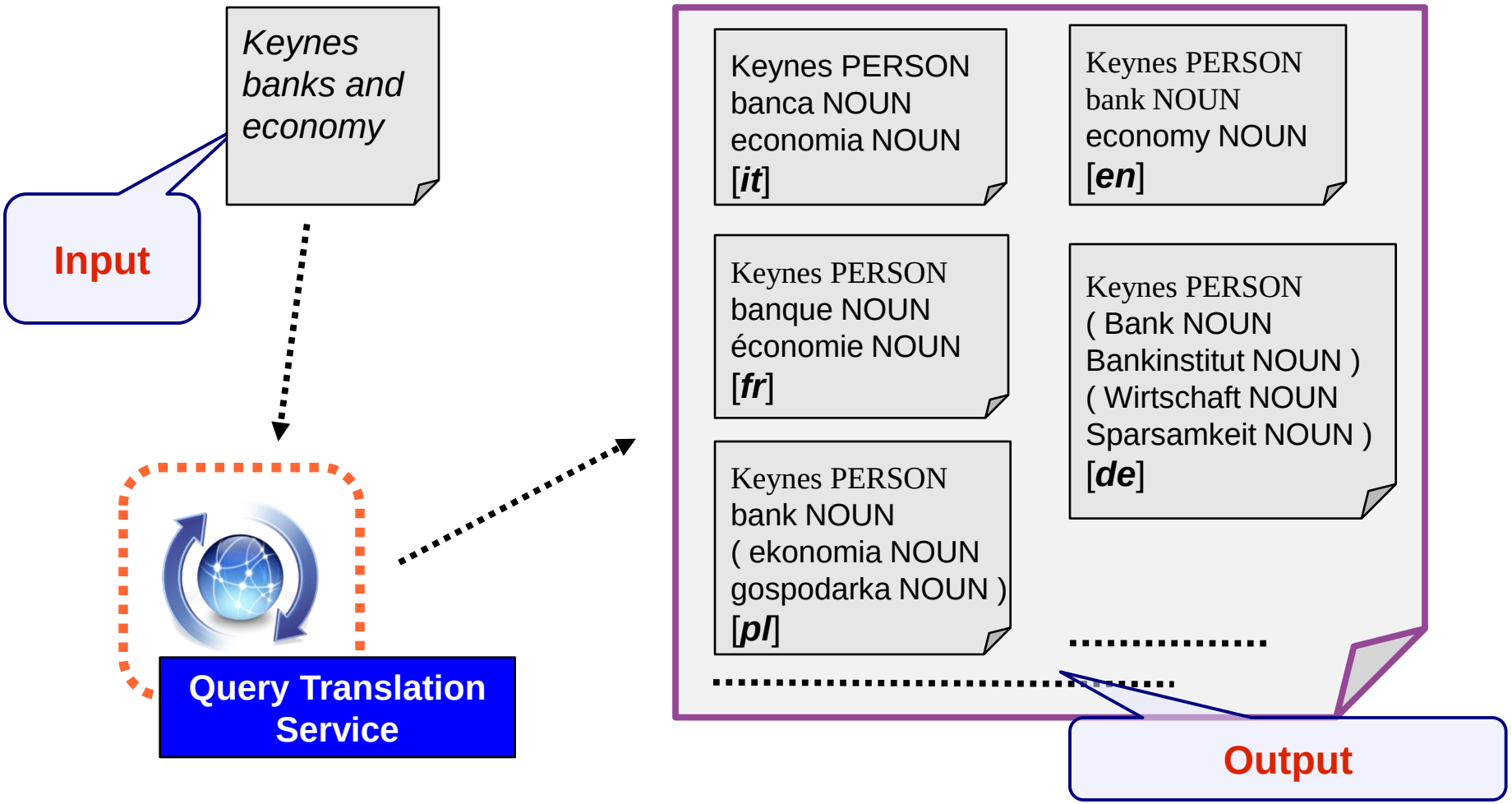
- *Language guesser*
- *Bilingual vocabularies,*
- *Stop-words lists*
- *Morphological analyzers*

Query Translation System functionalities

The overall functionalities offered by the Query Translation System should comprise:

- **Query Analysis:**
  - Named Entities/Multi-words/Compounds recognition; Lemmatization; Stop-words removal
- **Query Translation:**
  - By means of Bilingual Dictionaries (direct translation or multi-hop according to the available resources)
- **Translations Disambiguation:**
  - Selection of the most appropriate candidate translations with respect to the query context

Query Translation Example: *the big picture*



Query Translation Example: *behind the scenes*

Keynes banks and economy

1 Query Analysis

- *Keynes PERSON*
- *bank NOUN*
- ~~*and CONJ*~~
- *economy NOUN*

EN

2 Terms Translation

- *Keynes PERSON ↔ [Keynes PERSON]*
- *bank NOUN → [rive NOUN; banque NOUN; édifice bancaire NOUN; bord NOUN]*
- *economy NOUN → [économie NOUN, économique ADJ]*

3 Translation Disambiguation

-

- **bank NOUN → [*banque NOUN 0.85; édifice bancaire NOUN 0.7; rive NOUN 0.4; bord NOUN 0.25*]**

- **economy NOUN → [*économie NOUN 0.95; économique ADJ 0.6*]**

Guidelines

- **Modular approach**: the central service implements the core logic of the system (analysis/translation/disambiguation strategies) and orchestrates the different linguistic components actually wrapping the language resources (locally or via web services)

- **Variable language support**: it should be able to support languages with different types of resources available and consequently adapt its strategy (i.e. decompounding present or not, multiword detection present or not, stemmers / lemmatizers, ..)

- **Clear Interfaces:** all linguistic components from Language Resources must be compliant to with public IO specifications.
  - Different implementations for the same resource types can be integrated (i.e. with license or free) in the system
  - Any vendor at any time will be able to offer new language pairs or new language analysis services, allowing for an incremental language coverage within Europeana framework.

.

Proposals? Suggestions?

Thank You!