

Automatic translation: German to Simple German a Prototype

University of applied Science Darmstadt
Special Field Media

Presentation for the workshop Aspects of Linguistic Complexity
Author: Andreas Stiegelmayr

Teacher: Prof. Dr. Melanie Siegel

Presented at 8.05.2018

Table of Contents

1 Introduction

1.1 Motivation

1.2 Research Question

1.3 Goals

2 Tools

2.1 Apache UIMA

2.2 DKPro Core

2.3 Language Tool

Topics

3 Experiment

3.1 Data

3.2 Demonstration

3.2 Results

4 Future Work

5 Conclusion

6 Sources

1 Introduction

1.1 Motivation 1.2 Research Question 1.3 Goals

Why do we need Simple Language?

- Inclusion of people with disability's
- Reduction of risks at work
- European law

Problems in the area of machine Natural Language Processing

- Not enough data
-

1 Introduction

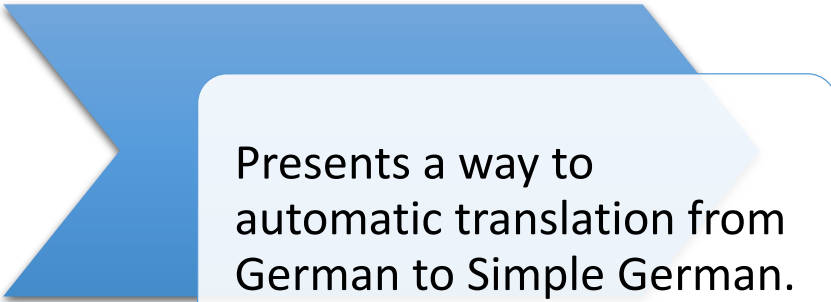
1.1 Motivation 1.2 Research Question 1.3 Goals

- *“Is it possible to built a program under use of the language tool to translates German into Simple German?”*

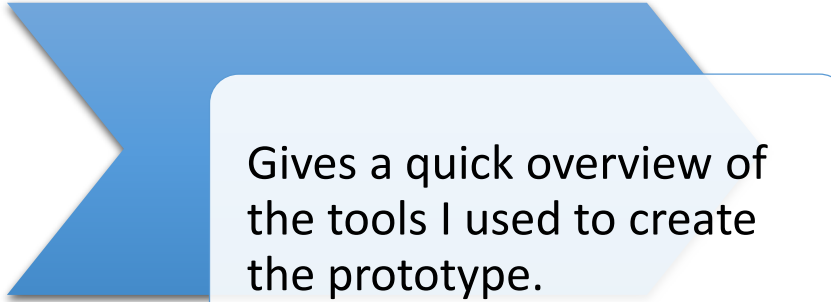


1 Introduction

1.1 Motivation 1.2 Research Question 1.3 Goals



Presents a way to automatic translation from German to Simple German.



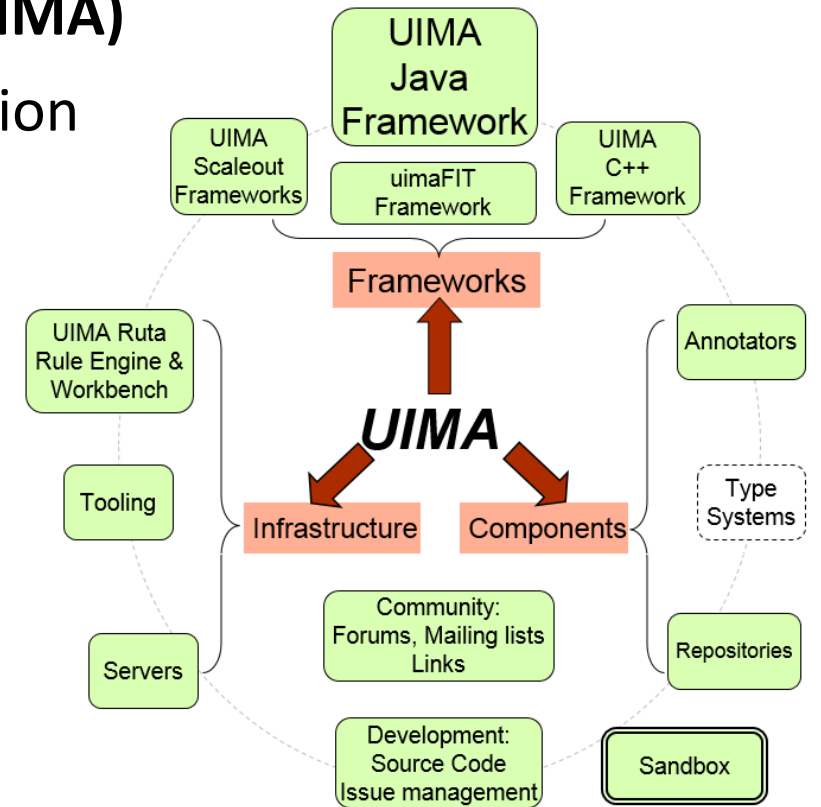
Gives a quick overview of the tools I used to create the prototype.

2 Tools

2.1 Apache UIMA 1.2 DKPro 2.3 Language Tool

Unstructured Information Management Applications (UIMA)

- Java Framework to analyse large unstructured information
- Examples for use:
 - language identification
 - language specific segmentation
 - sentence boundary detection
 - entity detection



UIMA: <https://uima.apache.org/>

2 Tools

2.1 Apache UIMA 1.2 DKPro 2.3 Language Tool



DKPro Darmstadt Knowledge Processing Repository

- DKPro is a Natural Language Toolkit based on UIMA
- Easy accessible with Maven
- DKPro Components:
 - DKPro Core
 - DKPro TC
 - DKPro Statistics
 - DKPro Similarity
 - DKPro UBY
 - DKPro WSD

```
private static CollectionReaderDescription reader;
private static AnalysisEngineDescription transformText;
private static ClassLoader classloader = Thread.currentThread().getContextClassLoader();

public static void main(String[] args) {

    try {
        reader = createReaderDescription(TextReader.class,
            TextReader.PARAM_SOURCE_LOCATION, classloader.getResource("data/corpus.txt"));
        transformText = createEngineDescription(TransformText.class);

        SimplePipeline.runPipeline(reader, transformText);
    } catch (UIMAException | IOException e) {
        System.out.println(e.getMessage());
    }
}
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>de.project.texttechnologies</groupId>
  <artifactId>texttoSimpleGerman</artifactId>
  <version>1.0-SNAPSHOT</version>

  <repositories>
    <repository>
      <id>uip-oss-model-releases</id>
      <url>http://soliberg.uip.informatik.tu-darmstadt.de/artifactory/public-model-releases-local</url>
    </repository>
  </repositories>

  <properties>
    <dkpro.core.version>1.9.1</dkpro.core.version>
  </properties>
  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>de.tudarmstadt.uip.dkpro.core</groupId>
        <artifactId>de.tudarmstadt.uip.dkpro.core-aslc</artifactId>
        <version>1</version>
        <type>pom</type>
        <scope>import</scope>
      </dependency>
    </dependencies>
  </dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>de.tudarmstadt.uip.dkpro.core</groupId>
      <artifactId>de.tudarmstadt.uip.dkpro.core-openssl-aslc</artifactId>
    </dependency>
  </dependencies>
</project>
```

DKPro: <http://dkpro.github.io/>

2 Tools

2.1 Apache UIMA 1.2 DKPro 2.3 Language Tool

DKPro Core Components

- Reader (Text, Brat, Conll, XML, Tiger XML etc.)
- Text analysis (Chunker, POS Tagger, NER, Parser, etc.)
- Writer

DKPro TC Components:

- Text analysis (avg. Token, avg. POS Tags, Number of Character etc.)

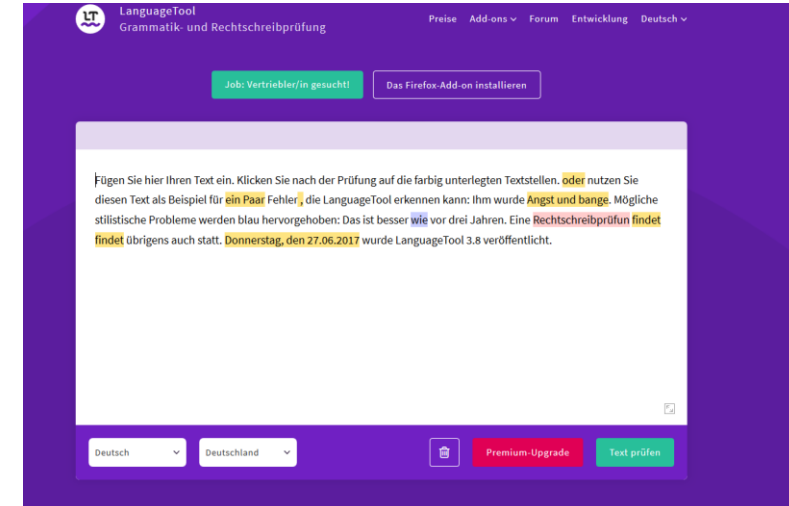
2 Tools

2.1 Apache UIMA 1.2 DKPro 2.3 Language Tool



Language Tool

- Java Framework for Natural Language Processing
- Language Tool Components:
 - Simple Speech (German)
 - Grammar Checker
 - Lemmatizer
 - Etc...



Language Tool: <https://languagetool.org/>

3 Experiment

3.1 Data 3.2 Presentation 3.3 Results

What kind of Data can we use?

- Any digital plane text
- Text of the demo is from Wikipedia:

Eine **Korrelation** ([mittellat. *correlatio*](#) für „Wechselbeziehung“) beschreibt eine Beziehung zwischen zwei oder mehreren Merkmalen, Ereignissen, Zuständen oder Funktionen. Zwischen Merkmalen, Ereignissen oder Zuständen braucht keine [kausale Beziehung](#) zu bestehen: manche Elemente eines Systems beeinflussen sich gegenseitig nicht; oder es besteht eine [stochastische](#) (= vom [Zufall](#) beeinflusste) Beziehung zwischen ihnen.

3 Experiment

3.1 Data 3.2 Presentation 3.3 Results

- Have fun with the presentation of the Prototype

3 Experiment

3.1 Data 3.2 Presentation 3.3 Results

Eine Korrelation **mittellat. correlatio** für Wechselbeziehung beschreibt eine Beziehung zwischen **2** oder mehreren Merkmalen, Ereignissen, Zuständen oder Funktionen. Zwischen Merkmalen, Ereignissen oder Zuständen braucht keine kausale Beziehung zu bestehen: manche Elemente **von dem System** beeinflussen sich gegenseitig nicht; oder es besteht eine stochastische **= vom Zufall beeinflusste** Beziehung zwischen ihnen.

4 Future Work

- Convert integrate bracket text
- Acronym detection
- Possibility of detecting phrases and transform them into Simple Speech

5 Conclusion

It is possible to create a program that can help to transcribe a corpus from German into Simple German.

Thank you for your attention!

Abbildungsverzeichnis:

- **Abbildung 1:** Abbildung 1: Wikipedia: Leo Breiman, https://de.wikipedia.org/wiki/Leo_Breiman , (Zugriff am 09.07.2017 um 15:00)
- **Abbildung 2:** scimagojr: „IEEE Transactions on Neural Networks and Learning Systems“, <http://www.scimagojr.com/journalsearch.php?q=21100235616&tip=sid&clean=0> , (Zugriff am 17.7.2017 um 15:00)
- **Abbildung 3:** scimagojr: Journal „Neural Networks“, <http://www.scimagojr.com/journalsearch.php?q=24804&tip=sid&clean=0>, (Zugriff am 17.7.2017 um 15:00)
- **Abbildung 4:** scimagojr: „Journal Rankings“, <http://www.scimagojr.com/journalrank.php?area=2600&category=2613> , (Zugriff am 17.7.2017 um 15:02)
- **Abbildung 5:** tvtropes: „Rashomon (Film)“<http://tvtropes.org/pmwiki/pmwiki.php/Film/Rashomon> (Zugriff am 17.7.2017 um 15:03)
- **Abbildung 6:** wikipedia: Occam; <https://de.wikipedia.org/wiki/Occam> , (Zugriff am 17.7.2017 um 15:03)
- **Abbildung 7:** Breiman, L. (2001). Statistical modeling: The two cultures. Statistical Science.
- **Abbildung 8:** Breiman, L. (2001). Statistical modeling: The two cultures. Statistical Science.
- **Abbildung 9:** Breiman, L. (2001). Statistical modeling: The two cultures. Statistical Science.