Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Organization 1

2 Background **2**

2.1 Argumentation 2

2.1.1 General Definitions 2

2.1.2 Persuasion 2

2.1.3 Persuasion 2.0 2

2.2 NLP and the DKPro Framework 2

2.2.1 Natural Language Processing 2

2.2.2 UIMA 2

2.2.3 DKPro Core 3

2.2.4 DKPro Text Classification 4

A DKPro Core Overview **6**

CHAPTER 1

Introduction

1.1 Motivation

I was motivated to write a Phd thesis because I did not want to work directly after finishing my study

1.2 Organization

This thesis is organized as follows, ...

CHAPTER 2

Background

The aim of this work is to create a model that predicts if an article comment or a forum post can be classified as ”persuasive” or ”non-persuasive”. We’ll first give general definitions about persuasion and argumentation, and how the corpus was annotated. Then, the tool that was used to perform the classfication, DKPro Text Classification Framework, will be introduced and its functionnalities will be explained. Finally we’ll discuss about the algorithm used and the metrics to evaluate our model.

2.1 Argumentation

2.1.1 General Definitions

2.1.2 Persuasion

2.1.3 Persuasion 2.0

2.2 NLP and the DKPro Framework

2.2.1 Natural Language Processing

2.2.2 UIMA

DKPro stands for *Darmstadt Knowledge Processing* [3] and it’s a software suite for NLP based on the Apache UIMA Framework. UIMA are software systems that analyse large volumes of unstructured information in order to discover knowledge that is relevant to an end user. An example UIMA application might ingest plain text and identify entities, such as persons, places, organizations; or relations, such as works-for or located-at:

The real power of UIMA is its Analysis Engines (AE) which basically analyse a document and record descriptive attributes. Those descriptive attributes will form the docu-

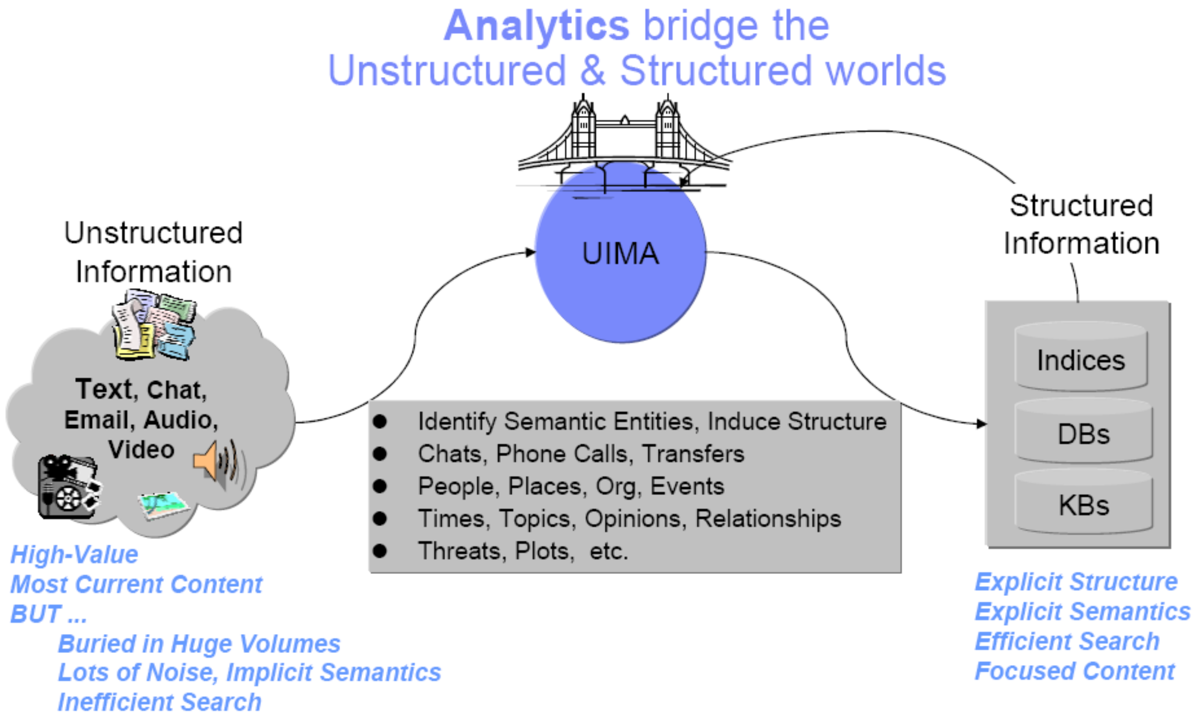


Figure 2.1: UIMA: Order unstructured data, from UIMA website [1]

ment's metadata that will be used for further analysis (such as Text Classification in our case).

GIVE EXAMPLES OF AE HERE AND DKPRO PROC PIPILINE AND CAS ?

Thus, the DKPRo software use UIMA's AE in order to collect structured information about textual data.

2.2.3 DKPro Core

Many NLP tools are already freely available in the NLP research community. DKPro Core [2] provides UIMA components wrapping these tools (and some original tools) so they can be used interchangeably in UIMA processing pipelines. The provided components wrap a constantly growing set of stand-of-the-art NLP tools and also include several original components covering a wide range of tasks including: tokenization/segmentation, compound splitting, stemming, part-of-speech tagging, lemmatization, constituency parsing, dependency parsing, named entity recognition, coreference resolution, language identification, spelling correction, grammar checking, and support for reading and writing various file and corpus formats.

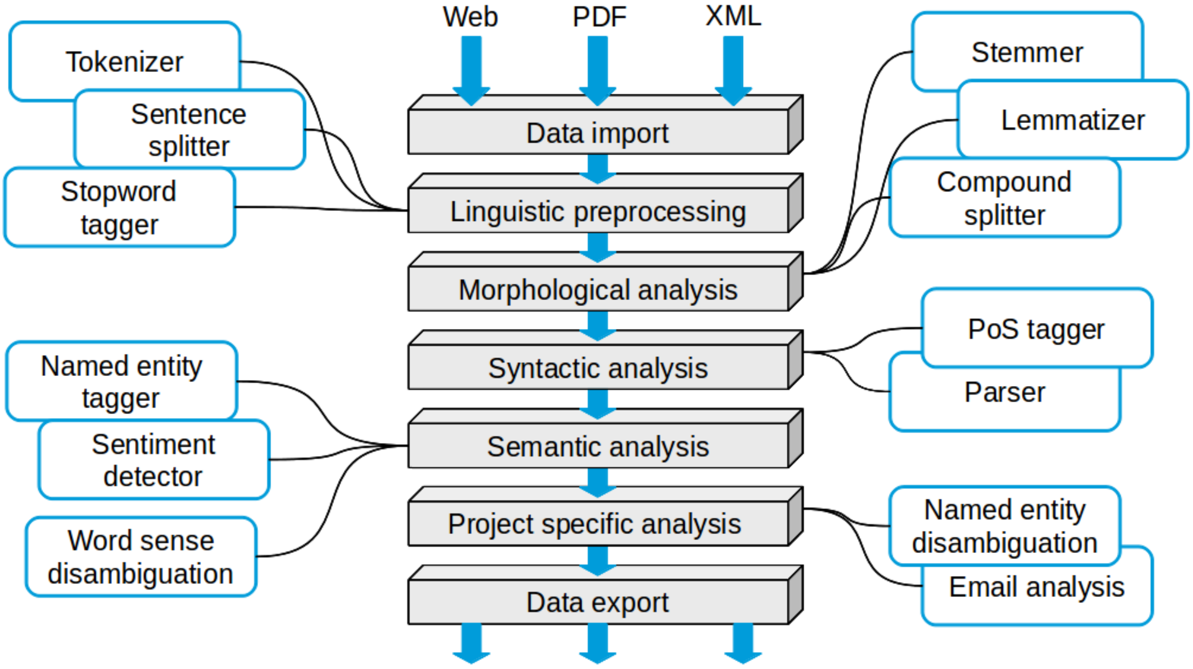


Figure 2.2: DKPro Core Pipeline

Core has several annotators either developed in-house or wrapped[1] from the state-of-the-art NLP libraries. Here is a non exhaustive list:

- **Stanford NLP** - segmentation, la lemmatisation, part of speech...
- **OpenNLP** - machine learning based toolkit for the processing of natural language text: tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and coreference resolution.
- **CleanNLP** - robust NLP components implemented in Java for part-of-speech tagging, dependency parsing, semantic role labelling...

2.2.4 DKPro Text Classification

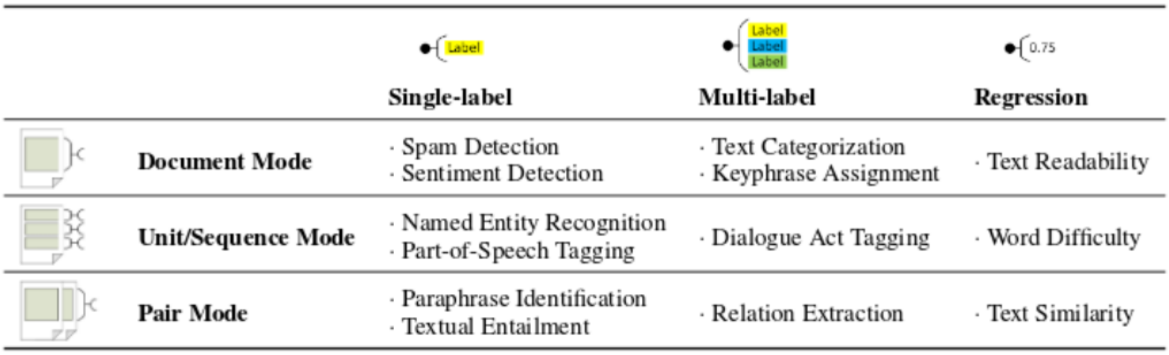
The aim of DKPro TC (commonly called TC) is to allow the user to apply machine learning algorithms easily on the extracted annotations. This framework was built in order to execute the following tasks:

- Supervised learning Classification. The user should provide annotated textual data.
- Can work atomically on text (word, sentence, paragraph) or on pairs of documents.
- Can perform single-label classification, multi-label classification and regression. DEFS

[1] A wrapper function is a subroutine in a software library or a computer program whose main purpose is to call a second subroutine or a system call with little or no additional computation. Source: Wikipedia

Concerning the algorithms used, TC relies on Weka[2] (*Waikato Environment for Knowledge Analysis*). Developed by the Waikato University in New Zealand, Weka is an open-source Data Mining software written in Java, which makes available to its users not only Machine Learning algorithms but also processing features (attribute selections and transformations) and a user interface with visualization tools. Regularly updated, Weka is one of the main state-of-the-art data mining software used in research.

Weka is integrated to TC thank to one major component: the **feature**. In the code, the feature is usually a class that computes a certain value (ex: length of a post, number of adjectives in a text, ect...) using the annotation provided by the DKPro pipeline. Features can be implemented by polymorphism from the mother-class *FeatureExtractorResource_ImplBase*. The results are then saved in an *ARFF* [3] file, which corresponds to Weka's format files.



	Label **Single-label**	Label Label Label **Multi-label**	0.75 **Regression**
Document Mode	· Spam Detection · Sentiment Detection	· Text Categorization · Keyphrase Assignment	· Text Readability
Unit/Sequence Mode	· Named Entity Recognition · Part-of-Speech Tagging	· Dialogue Act Tagging	· Word Difficulty
Pair Mode	· Paraphrase Identification · Textual Entailment	· Relation Extraction	· Text Similarity

Figure 2.3: The different usages of DKPro TC

In a nutshell, TC adds 4 more steps to DKPro Core:

- A step where the data are labelled, since we're doing supervised learning. In practice, the label information is extracted my a function of the reader ??WHERE do we define reader ?
- Extraction of the features from the annotations.
- Data Processing and Cross Validation ??DEF
- Report of the results (Accuracy, Macro F-Measure, ect...)

[2] http://www.cs.waikato.ac.nz/ml/weka/

[3] An ARFF (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes.

APPENDIX A

DKPro Core Overview

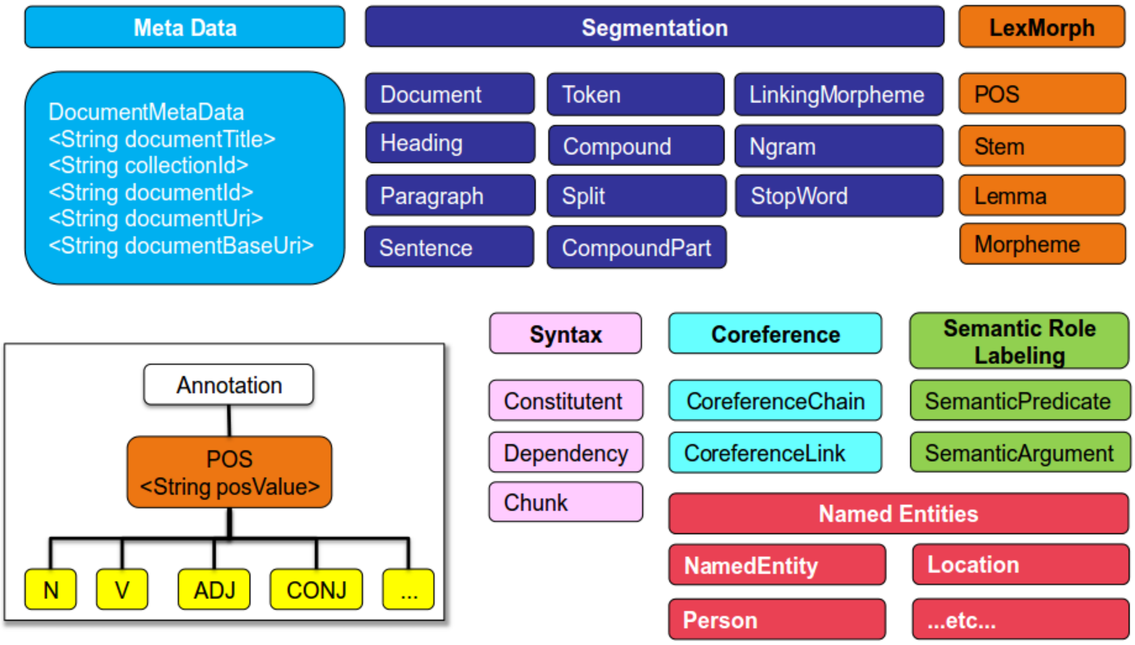


Figure A.1: DKPro Type System

Glossary

mathematics Mathematics is what mathematicians do. 3

Bibliography

[1] Overview and setup, uima.apache.org, 2006.

[2] Richard Eckart de Castilho and Iryna Gurevych. A broad-coverage collection of portable nlp components for building shareable analysis pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT (OIAF4HLT) at COLING 2014*, pages 1–11, August 2014.

[3] Iryna Gurevych, Max Mühlhäuser, Christof Müller, Jürgen Steimle, Markus Weimer, and Torsten Zesch. Darmstadt knowledge processing repository based on uima. In *Proceedings of the First Workshop on Unstructured Information Management Architecture at Biannual Conference of the Society for Computational Linguistics and Language Technology*, Tübingen, Germany, April 2007.