Contents

v

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, ...

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

To better understand the vocabulary that will be used in this report, we give four important definitions:

Debate The process of inquiry and advocacy; the seeking of a reasoned judgement on a proposition. ([5], p. 2)

Controversy Controversy is an essential prerequisite of debate. Where there is no clash of ideas, proposals, interests, or expressed positions on issues, there is no debate. ([5], p. 43)

Argumentation Reason giving in communicative situations by people whose purpose is the justification of acts, beliefs, attitudes, and values. ([5], p. 2)

Persuasion Communication intended to influence the acts, beliefs, attitudes, and values of others. ([5], p. 2)

2.1.2 Persuasion

Persuasion and argumentation are the essence of any debate about controversies. Whether on-line or face-to-face, people try to convince others about their opinions, values, and attitude towards that particular controversy using various kinds of argumentation.

Lets assume a made-up example from a discussion forum about single-sex education, a quite controversial topic. In one post, the author (i.e. *Jack*) writes:

> **#ex1 (forumpost, single-sex education)** I'm completely against single-sex education. This does not prepare students for real life where men and women live together!! Jack

Jacks intention here is not only to share his opinion but also to persuade other users in the debate (and potentially all readers on the Internet). We can thus treat his message as persuasive (also cf. definition above). The means he uses to persuade is argumentation, because he also gives some reasons to support his stance towards the discussed topic.

However, the way people argue is not always as clear as in the example above. Suppose we have the following text from an actual debate about home-schooling:

> **#203 (artcomment, homeschooling)** Teaching is not just subject knowledge (although Id be the last to downplay that). It is also meeting other people from all walks of life, dealing with new situations, finding friends etc. Ive always felt sorry for home-schooled kids. They are being denied their childhood and adolescence by parents who want to exercise total power of them, deny them the pains and pleasures that social experience brings. JuniusPublicus

The author does not say explicitly that he/she is against home-schooling. However, he/she provides some examples (necessity of social interaction, total power of parents) and expresses feelings (sorry for home-schooled kids), so we can infer out the *implicit* message. This example is thus also *persuasive* and *argumentative.*

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* [6] 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 document'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 ?

Analytics bridge the
Unstructured & Structured worlds

Unstructured
Information

Text, Chat,
Email, Audio,
Video

High-Value
Most Current Content
BUT ...
    Buried in Huge Volumes
    Lots of Noise, Implicit Semantics
    Inefficient Search

UIMA

- Identify Semantic Entities, Induce Structure
- Chats, Phone Calls, Transfers
- People, Places, Org, Events
- Times, Topics, Opinions, Relationships
- Threats, Plots, etc.

Structured
Information

Indices

DBs

KBs

Explicit Structure
Explicit Semantics
Efficient Search
Focused Content

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

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 [4] 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 written Java 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

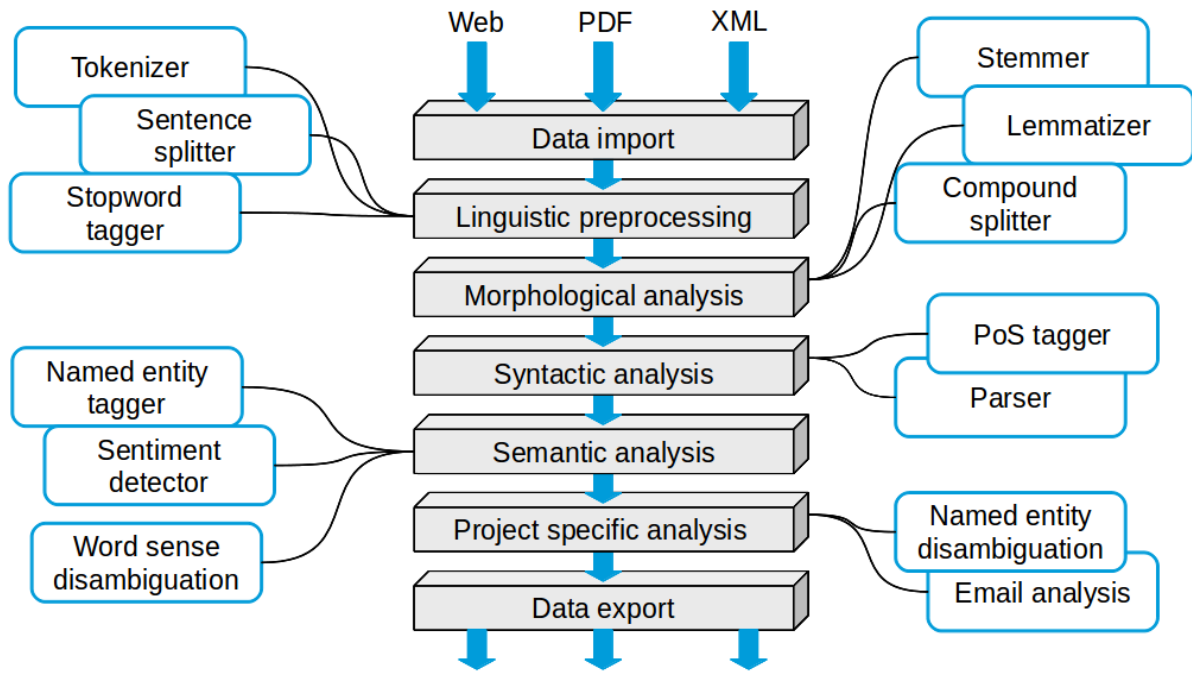Core has several annotators either developed in-house or wrapped[1] from the state-of-the-art NLP libraries. Here is a non exhaustive list:

- **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.

		Single-label	Multi-label	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...)

The TC developers give regularly some helpful tutorials on their website[4].

License and usage

While most DKPro TC modules are available under the Apache Software License (ASL) version 2, there are a few modules that depend on external libraries and are thus licensed under the General Public Licence (GPL).

---

[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.

[4]https://code.google.com/p/dkpro-tc/wiki/DemoExperiments

The SVN[5] commits of the different modules are available through Google code and the integration of new functionalities in a project is made possible by Maven (appendix B). It's also possible to use DKPro TC with Groovy which is an object-oriented programming language for the Java platform and destined to be run on a server.

---

[5]Apache Subversion (often abbreviated SVN, after the command name svn) is a software versioning and revision control system distributed as free software under the Apache License.

The Research Work

In this part, we'll see how the theoretical knowledge in Argumentation Theory and NLP presented were applied to a concrete case study. We'll first see the types of textual data we have, how they were annotated for the supervised learning task and then how features were engineered to perform an automatic classification.

3.1 Text Corpus

The data set used for this study was originally composed of 990 text files. They contain forum posts or articles about 6 different domains related to education, that provoke debate in the American society:

- **homeschooling**: It's the education of children outside the formal settings of public or private schools and is usually undertaken directly by parents or tutors.

- **redshirting**: The practice of postponing entrance into kindergarten of age-eligible children in order to allow extra time for socioemotional, intellectual, or physical growth.

- **prayer in schools**: Debate about whether or not a public school should allow and allocate time and buildings for religious practices.

- **public vs private schools**: Which kind of school offers the best education.

- **mainstreaming**: In the context of education, it's the practice of educating students with special needs in regular classes during specific time periods based on their skills.

- **single sex education**: The practice of conducting education where male and female students attend separate classes or in separate buildings or schools.

The meta-information for each text (id, type of post, domain) is given my the name of the file in itself such as follow:

$\frac{\text{Text}}{1}\frac{\text{Text}}{2}\frac{\text{Text}}{3}$??FIND A WAY TO DO IT THIS WAY

Later in the internship, I started to use *XMI* files[1] that contain more information about the post or comment in itself such as the author, the date, ect... ??Annexe with how xmi looks like ?

3.1.1 Manual Annotation

As mentioned before, the classification we want to perform is a supervised learning problem since it wants to imitate the human decision on judging if a post is persuasive or not. An annotation guideline was written by Ivan Habernal [7] in order for the annotators to understand the task. In this section, we'll discuss about the general ideas of this guideline.

3.1.1.1 Sources of the data

The textual data that will be used for the studies come from to kind of online sources:
```
artcomment:  Article Comments, reactions to online articles
forumpost:   Forum Posts, posts in online debates
```

3.1.1.2 Categories in Persuasion

The task: Distinguish, whether the comment is persuasive regarding the discussed topic. The key question to answer is: *Does the author intend to convince us clearly about his/her attitude or opinion towards the topic?* If the answer is yes, we classify the comment as persuasive. There are two main categories in this task, namely `P1:Persuasive` and `P2:Non-persuasive`. The second category is further divided into more categories, that basically cover the various phenomena that may be encountered in the data.

However, It is not necessary to categorize the data exactly into one of the categories under `P2:Non-persuasive`. For example, a particular text may be both off-topic and out-of-context; in that case, choose either of these categories.

Remember: we are mainly interested in finding the `P1:Persuasive` documents that represent on-topic texts with intentions to persuade and convince the readers.

```
Quick overview of possible distinct categories:
P1:  Persuasive
P2:  Non-persuasive
    P2.1:  Out-of-context or reaction to other comment
    P2.3:  Off-topic
    P2.4:  Personal worries
    P2.5:  Story-sharing without intentions to persuade
    P2.7:  Impossible to decide about persuasiveness without deep background
knowledge
```
While the annotation phase, the annotators were charged of determining if a text was in

---

[1]The XML Metadata Interchange (XMI) is an Object Management Group (OMG) standard for exchanging metadata information via Extensible Markup Language (XML).

the P1 class and if not they had to specify to which of the seven P2 subclasses it belongs to.

3.1.1.3 Examples

Clearly belongs to P1

> **#2013 (forumpost, single-sex-education)** School should be co-ed. Some children are awkward with others of their own gender. For example, certain girls who are tom-boys might not be comfortable in a room full of girls. The mixed genders are preparing kids for the real world, where things are not segregated.

In #2013, the author states that schools should not be single-sex, also provides reasons why he/she thinks so.

Problematic P1 case

This section discusses some examples that were marked as P1 by two annotators but as P2 by a third annotator. We will explain, where the third annotator made an error.

> **#300 (artcomment, homeschooling)** This is not representative of most homeschoolers. This is a very, very small minority. Lets compare that to entire schools in the public school system that cater their teaching to make sure their kids pass the standardized tests so they can keep funding, meanwhile the kids cant understand concepts that arent covered on the tests.
> I was homeschooled in Texas, where there is no government oversight of homeschooling. I graduated high school at age 16 with 24 college credits under my belt, was accepted into every university I applied to (all the major schools in Texas), and graduated college in three years at age 19 after being on the Deans List every semester except one. Neither of my parents has a college degree and would not be deemed "qualified" to teach me. Somehow I didnt just make it, I thrived. Parental involvement works.

The second paragraph in #300 contains a statement "parental involvement works", which is clearly in favor of homeschooling.

P2.1: Non-persuasive, out-of-context or reaction to other comment

> **#3245 (artcomment, public-private-schools)** Why are they bad, they still pay taxes but dont use the service so there is more money for the system to use on fixing its issues. Even when everyone is doing everything they can to fix something does not mean it will be fixed.

In #3245, without any context, we can only roughly guess what the author is writing about.

P2.3: Non-persuasive, off-topic

> **#2049 (artcomment, single-sex-education)** Single-sex education. A poem./ Dearest people, the people/ always arguing and full of hate,/ why oh why should we ever/ turn out this way? Single-sex,/ co-educational, why does it matter?/ Girls, boys, everyone;/ WE CANNOT REMAIN LIKE THIS/ do you hear me?

P2.4: Non-persuasive, personal worries

> **#5024 (artcomment, redshirting)** Oh boy. . . oh my little (but very tall) girl. Ive chosen to put her into a second year of preschool next year (5 days instead of 3) because I feel that's what is right for her. She's a late October baby, but I'm not sure she's ready for kindergarten. But I worry. Will she be the giant of her class every year? Will there be an opportunity to skip her a grade? She's quite bright, but socially still a little awkward. I don't feel I'm "holding her back", yet if she has brand new twin sisters arriving in July, should I totally turn her world upside down and ship her off to another school with mostly older kids? I'm torn (and totally on the fence) both ways. I want her to excel academically, but I don't want to throw too many changes at her at once. I'm with you, Erica. I'm torn, and I chose the now unpopular "redshirting", but not so she can be a hockey superstar. . . :) I just thought this was a better pace for her. In ten years, I'm sure the "experts" will be telling me I should have held her back, because all the young kids are struggling. . . You can't win.

In #5024, the author only expresses her worries about her child, but she neither takes stance on the topic nor argues about that.

P2.5: Non-persuasive, story-sharing without intentions to persuade

> **#5030 (artcomment, redshirting)** Born in November, my youngest sister was among the oldest children in her peer group until she skipped a grade (I believe she skipped grade one but it may have been grade two). My other sister, also born in November and two years older, showed my youngest sister her homework and my youngest sister proved such a quick learner the teacher had no choice but to recommend she be moved up. Shes still achieving plenty and has never been intimidated by anyone older. She has a competitive drive and enjoys pushing herself forward.

The purpose of #5030 was to share the story without taking stance towards the topic or persuading others (the story of her sister skipping a grade and doing well could is also too far from the redshirting topic).

P2.7: Non-persuasive, impossible to decide about persuasiveness without deep back-ground knowledge

> **#164 (artcomment, homeschooling)**: Child abuse in the name of religious freedom. Just like parents who refuse medical treatment for their children. It makes me wish there was a hell.

In #164, without knowing the context that homeschooling and religion education are somehow related issues in some communities, it is not possible to decide about persuasiveness of this document.

3.1.1.4 Annotation Process

Three annotators were charged of labelling the data as P1 and P2 (if P2, he was asked to specify the subclass). For every annotations, the annotator could write a comment about why he thinks the text can be considered as persuasive or not. This comment can be useful especially if there's a conflict among the 3 annotators.

Once the annotations are performed, a discussion takes place with the annotators in order to solve issues and conflict annotations. If all annotators agree on the class (P1 or P2) of a text, the class will be set as the *gold label* of this text. But if after the discussion, there's still a conflict, the text will be labelled according to majority.

To evaluate how well were the annotations, we compare statistical metrics that are described in appendix C such as Recall, Precision, Accuracy and Macro F_1 measure. The comparison will be performed on 4 scenario:

- A_1 **vs** A_2

- A_1 **vs** A_3

- A_2 **vs** A_3

- 3 Annotators **vs** Gold data

A_1, A_2, A_3 stand for Annotator number 1, 2 and 3.

3.1.1.5 Results of the Manual Annotations

We measure the performances of the annotations on 3 batches of text data, and we aggregate the results:

	Docs	Macro F_1	Acc.	Persuasive			Non-Persuasive		
				P	R	F_1	P	R	F_1
Batch 1									
A1	100	0.879	0.880	0.942	0.845	0.891	0.813	0.929	0.867
A2	100	0.895	0.900	0.875	0.966	0.918	0.944	0.810	0.872
A3	100	0.849	0.850	0.922	0.810	0.862	0.776	0.905	0.835
Batch 2									
A1	200	0.855	0.855	0.909	0.792	0.847	0.813	0.919	0.863
A2	200	0.910	0.910	0.919	0.901	0.910	0.901	0.919	0.910
A3	200	0.874	0.875	0.839	0.931	0.883	0.920	0.818	0.866
Batch 3									
A1	509	0.927	0.927	0.953	0.906	0.929	0.902	0.950	0.926
A2	502	0.879	0.884	0.836	0.986	0.905	0.977	0.757	0.853
A3	511	0.907	0.908	0.977	0.835	0.900	0.857	0.981	0.915
All data									
A1	809	0.904	0.904	0.942	0.871	0.905	0.867	0.940	0.902
A2	802	0.890	0.893	0.858	0.964	0.908	0.948	0.807	0.872
A3	811	0.893	0.893	0.929	0.855	0.890	0.861	0.932	0.895

Table 3.1: Human performance on *gold data persuasive*.

??help: Find a way to say the results are good that's why we want to do a text classification.

3.1.2 Corpus Statistics

Now that we have the *gold labels* for the texts, we can sum the relevant information in a table: ??Find a ways for the color and the size

		redshirting	prayer-in-schools	homeschooling	single-sex-education	mai...
P1	artcomment	24	60	64	17	1
	forumpost	14	17	22	9	9
	all	38	77	86	26	10
P2	artcomment	15	43	93	16	2
	forumpost	15	23	45	8	17
	all	30	66	138	24	19
P1 + P2	Total	68	143	224	50	29
	Percentage	6.9	14.4	22.6	5.1	2.9

3.1.3 Conflict Annotations

??Mention the conflict annotations + some statistics

3.2 Feature Engineering

Now that we have all our data annotated, we have to extract relevant information from it in order to perform the classification. The problem of identifying persuasion in a text is a relatively new question in NLP and we don't have any straightforward methodology to find relevant features. Thus, we'll implement the NLP standard features that are used widely in other sub-fields of language processing and then we'll use the state-of-the-art features discovered in Argumentation Mining. A lot of modules of DKPro helped us to create our features but we had to extend somehow the software for certain functionalities that were not in DKPro Core and TC (for example the *Sentiment Analysis*).

3.2.1 Lexical Features

The adjective lexical refers to the words and the vocabulary of a corpus. This part will deal with the extraction of meaningful features related to words, sentences, tokens[2], punctuations, ect...

3.2.1.1 Tokens N-Grams

In NLP, an n-gram is a contiguous sequence of n (with n integer) items from a given sequence of text or speech. As a result of, tokens n-grams are sequences of tokens from

---

[2]Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens.

a text (*Note:* For more information about units in linguistics such as words, tokens, lemma and stemma, have a look at appendix D). The study of n-grams distribution in a corpus is an ancient technique [12] in language processing and it's usage is common in the field.

In this study, we use a DKPro available feature that extracts the 10.000 most common 1,2 and 3-grams in all the corpus and returns for each text a 10000-dimensional binary vector. Each vector element corresponds to one of the 10000 extracted n-gram, its value is 1 the text contains the n-grams, 0 otherwise.

To better understand this feature, let's take a simple example. We consider that the set of 1-grams contains 10 elements such as follow:

$$E = \{2, 1990, a, born, dog, Frankfurt, in, I, was, zoo\}$$

If the text input is:

                            I was born in 1990

DKPro will return the following binary 10-dimensional vector:

2	1990	a	born	dog	Frankfurt	in	I	was	zoo
0	1	0	1	0	0	1	1	1	0

As mentioned before, this feature is already implemented in DKPro and it's easy to use it in a pipeline.

```java
@SuppressWarnings("unchecked")
@Override Dimension<List<Object>> getPipelineParameters()
{

    return Dimension.create(
            DIM_PIPELINE_PARAMS,
            Arrays.asList(new Object[] {
                    FrequencyDistributionNGramFeatureExtractorBase.PARAM_NGRAM_MIN_N, 1,
                    FrequencyDistributionNGramFeatureExtractorBase.PARAM_NGRAM_MAX_N, 3,
                    FrequencyDistributionNGramFeatureExtractorBase.PARAM_NGRAM_USE_TOP_K,
                    10000
                    }
            ));
}
```

Figure 3.1: NGram Feature in DKPro Java code

As displayed on 3.1, the n-gram feature extractor takes three parameters: *PARAM_NGRAM_MIN_N* for the minimal value of n, *PARAM_NGRAM_MAX_N* for the maximal value of n and *PARAM_NGRAM_USE_TOP_K* which is the number of common n-grams retained. Since we want the 10.000 most common 1,2 and 3-grams, the parameters are set such as follow:

    PARAM_NGRAM_MIN_N = 1

    PARAM_NGRAM_MAX_N = 3

```
PARAM_NGRAM_USE_TOP_K = 10000
```

We also implemented a subclass from the n-grams extractors to have the lemma instead of the tokens. We thought that it would give us better results since lemma refer to a general form of a word (ex: *be* instead of *are*, *was*, *is*, ect...) but in *a posteriori* analyses, we don't see any improvements in our results. At the result of, we'll only consider tokens n-grams in our features.

The Tokens N-Grams feature will give us results for our *Baseline Analysis*: more complex models and pipelines will systematically be compared to this "simple" model.

3.2.1.2 Tokens and Sentences

We compute some statistics regarding sentences and sentences in a text:

- Number of sentences and tokens in a text.

- Maximum size (in character) of a token and a sentence in a text.

- Minimum size (in character) of a token and a sentence in a text.

- Average size (in character) of tokens and sentences in a text.

As an example, if we consider the following text as input:

208_P2_artcomment_homeschooling.txt "Not having read any of the standard high school literature, people make references I dont get." Got news for you. They're not making references to required high school readings. More likely Internet and pop culture. I hope you succeed getting some accountability in to the system. What is this issue, gun ownership?

The outputs are the following descriptive statistics:

- **6** sentences in this text.

- The minimal sentence is "Got news for you" and its size is **18** characters.

- The maximal sentence is "Not having read any of the standard high school literature, people make references I dont get." and its size is **100** characters.

- The average size for a sentence is **53.3** characters.

and besides:

- **67** tokens.

- The minimal token is "I" and its size is **1** characters.

- The maximal sentence is "accountability" and its size is **14** characters.

- The average size for a token is **4.0** characters.

The features related to minimal sizes wouldn't give us much information about a text since a words like *a* or *I* are usually used. On the other hand, the maximal size, total number and average size related features might be very useful since they quantify somehow the interest of an author in a conversation/debate.

3.2.1.3   Other Lexical Statistics

In the article *Stance Classification of Ideological Debates*[8], Hasan defines 3 simple features which help to perform stance classification:

- Length in characters of a text.

- Ratio of tokens with more than 6 characters.

- Average number of tokens per sentence.

3.2.1.4   Punctuation Related Features

First, we defined a set of features which simply compute the ratio per token of these 6 punctuation marks full stop, comma, question mark, exclamation mark, colon and quotation mark (fig. 3.2). This would tell us if the author is caring about his style of writing.

.   , ? ! : "

Figure 3.2: 6 punctuation marks

Another punctuation feature inspired by *An* and *Walker* [3] is the repeated punctuation feature that computes the number of repeated punctuation (such as "!!!!!" or "!??!") in a text. In practice, this feature requires the following *regular expression*[3]: [?!.,]+

```java
String pattern = "[?!.,]+";
Pattern r = Pattern.compile(pattern);
Matcher m = r.matcher(jcas.getDocumentText());

int countMultiplePunc = 0;
while (m.find()){
    if (m.group(0).length() > 1){
        countMultiplePunc++;
    }
}
```

Figure 3.3: Piece of code for the multiple punctuation feature

This feature can be a good representation of aggressiveness and poor argumentation in a debate, as it can be seen in the following example:

---

[3]In theoretical computer science and formal language theory, a regular expression (abbreviated regex or regexp) is a sequence of characters that forms a search pattern, mainly for use in pattern matching with strings, or string matching, i.e. "find and replace"-like operations.

> **208_P2_artcomment_homeschooling.txt** smarmy bastard 2011/09/19 at 6:00 PM
> ”ok so obviously , there are more free thinkers who would agree with you, ... and
> because there are more than one free thinker(s) , it becomes a group of free thinkers
> ...who all agree ... hmmm (head hits floor)” ‗‗ Smarmy: now you’re just being disagree-
> able. If many people independently think for themselves, without being told what, when,
> and how to think, it does not follow that they all think the same thing. ”Following” is
> an attribute reserved for religion. Me thinks your head may have hit the floor too hard
> this time.

3.2.1.5   Multiple Capital Letters

Another feature that can reflect the lack of seriousness is the number of words with mul-
tiple capital letters. In the following example, there are 3 words with multiple capital
letters:

> **3144_P2_artcomment_public-private-schools.txt** WRONG - NO !! Perhaps bad
> psychology, bad child rearing, perhaps. They are paying for the public schools, its called
> TAXES!!

3.2.2   Part Of Speech Features

In grammar, parts of speech (abbreviation: POS) are the linguistic categories of words
such as verb, noun, ect... In DKPro the POS are modelled as subclasses of the class *POS*,
which is an annotation.

3.2.2.1   Ratio on common POS

One simple feature, already implemented in DKPro[4], computes 11 ratios of 11 different
over the total number of POS:

POS	Abbreviation	Examples
Adjective	ADJ	good, tall
Adverb	ADV	quickly, lightly
Article	ART	a, the
Cardinal Number	CARD	one, eighty-two
Conjunction	CONJ	for, and
Noun	N	cat, Germany
Exclamation	O	O, oh!
Preposition	PP	above, within
Pronoun	PR	I, she
Punctuation	PUNC	“.”, “;”
Verb	V	to be, had

Table 3.2: The 11 POS we consider for the Ratio POS feature

??TODO: Investigate why no all the black lines

---

[4]In TC Google code, have a look at de/tudarmstadt/ukp/dkpro/tc/features/syntax/POSRatioFeatureExtractor.java

The DKPro functions allow to compute the ratios and thus create the features very easily as seen on fig. 3.4.

```
double total = JCasUtil.select(jcas, POS.class).size();
double adj = select(jcas, ADJ.class).size() / total;
double adv = select(jcas, ADV.class).size() / total;
double art = select(jcas, ART.class).size() / total;
double card = select(jcas, CARD.class).size() / total;
double conj = select(jcas, CONJ.class).size() / total;
double noun = select(jcas, N.class).size() / total;
double other = select(jcas, O.class).size() / total;
double prep = select(jcas, PP.class).size() / total;
double pron = select(jcas, PR.class).size() / total;
double punc = select(jcas, PUNC.class).size() / total;
double verb = select(jcas, V.class).size() / total;
```

Figure 3.4: Piece of code for the multiple punctuation feature

3.2.2.2 Comparative and Superlative

The previous features don't consider the ratios of comparative and superlative (for adverbs and adjectives) in a text but they are relevant in debates since opponents usually keep on comparing the different point of views.

3.2.2.3 Modal Verbs

Again, in a more granulate analysis, we can evaluate the ratios for the 9 common ratios in English: can, could, may, might, must, shall, should, will and would.

3.2.2.4 POS N-Grams

By analogy with the Token NGrams feature, DKPro has a POS NGrams feature. As an example, if you consider the sentence:

This	Virginia	law	is	insane	.
ART	NP	NN	V	ADJ	PUNC

The POS 1-grams are: `ART, NP, NN, V, ADJ, PUNC`
The POS 2-grams are: `ART_NP, NP_NN, NN_V, V_ADJ, ADJ_PUNC`
and so on ...
Unfortunately, the POS n-grams tend to introduce some noise and redundancy in our classification, so we won't use them much.

3.2.3 Syntactic features

The *syntax* is the study of how languages are constructed in a certain language. We'll define in this part the syntax related. Syntax is very descriptive, most of the syntactic representations of sentences, texts are often graphs, tables, ect... We'll see how we came with the quantitative features needed to perform the classification.

3.2.3.1 Depth of the Dependency Tree

Dependency Tree

In V Ágel's works [2], the dependency tree is a graph that maps the relations between the different grammar units in a sentence. The theory behind is long long and arduous and we leave to the reader the study of dependency trees. Nevertheless, we give in the following figure (3.5), a simple example of a dependency tree.

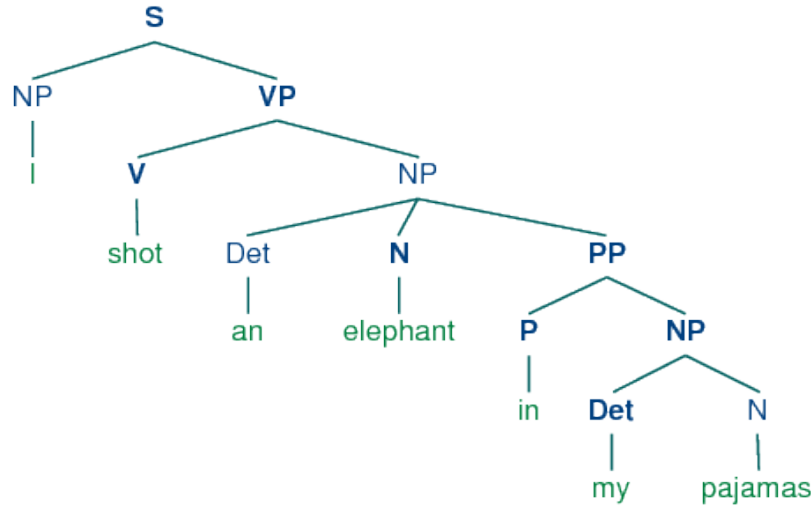For the sentence `I shot an elephant in my pajamas`, we get the following tree:

Figure 3.5: A dependency tree

To quantify how complex a sentence can be, we took inspiration on Christian Stab work on Argumentative Discourse [11] by calculating the depth of the dependency tree for every sentence. The depth of a tree is the number of edges between the first node and the furthest extremity in the tree. In fig. 3.5, the depth of the tree is 5.
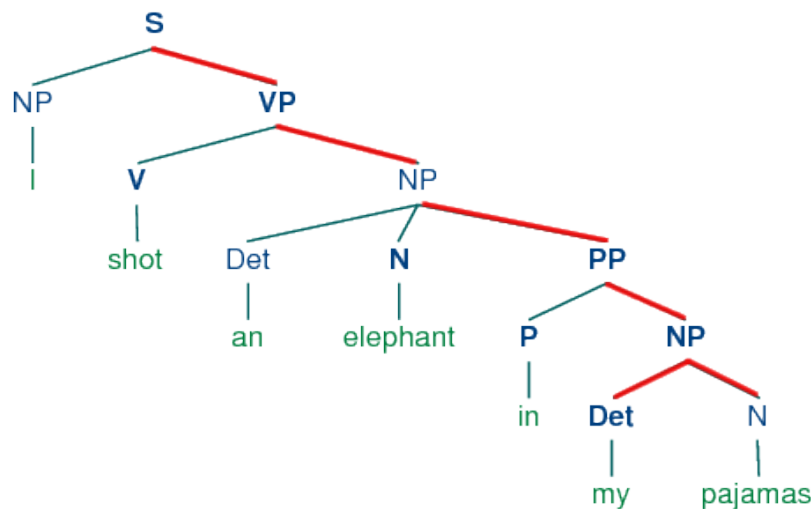

Figure 3.6: 5 edges between the summit S and the extremities Det and N

Building features with this metric

The dependency tree is available on DKPro with the *MaltParser*[9]. For every input sentence, it returns the corresponding dependency tree, such as follow: Certain functions

```
(S (NP I) (VP (V shot) (NP (Det an) (N elephant) (PP (P in) (NP (Det my)
                        (N pajamas))))))
```

Figure 3.7: MaltParser's output tree

allow to evaluate the depth of this kind of tree. Even so, our base unit for the classification is the text (and not the sentence), so we need to compute certain statistics over the sentences:

Maximal Tree

If *depth* is a function that returns the depth of a tree, the biggest tree in a text has a size of:

$$\max_{s \in text} depth(s)$$

Where the dummy variable s corresponds here to a sentence.

Average length of a tree

Here we simply calculate the tree depth average on all the sentences:

$$\frac{1}{|s|} \sum_{s \in text} depth(s)$$

Bla bla ?

3.2.3.2 Dependency Rules

Similarly to the tokens n-grams and the POS n-grams, it's possible to define dependencies n-grams, simply called dependency rules [11]. As an example, some of the dependency rules from the previous tree (fig. 3.5) are : VP → NP, NP → PP → NP, ect...

In our study, we extract 5000 dependency rules from the all corpus and compute binary vectors that show the presence or not of a dependency rule.

3.2.3.3 Subordinate clauses

Clause

In grammar, a clause is the smallest grammatical unit that can express a complete proposition.

Subordinate clause

Subordination as a concept of syntactic organization is associated closely with the distinction between coordinate and subordinate clauses. One clause is subordinate to another, if it depends on it. The dependent clause is called a subordinate clause and the independent clause is called the main clause.

We can distinguish 5 kind of subordinate clauses:

- **Clause S** - simple declarative clause, i.e. one that is not introduced by a (possible empty) subordinating conjunction or a *wh-word*[5] and that does not exhibit subject-verb inversion.

- **Clause SBAR** - Clause introduced by a (possibly empty) subordinating conjunction.

- **Clause SBARQ** - Direct question introduced by a wh-word or a wh-phrase. Indirect questions and relative clauses should be bracketed as SBAR, not SBARQ.

- **Clause SINV** - Inverted declarative sentence, i.e. one in which the subject follows the tensed verb or modal.

- **Clause SQ** - Inverted yes/no question, or main clause of a wh-question, following the wh-phrase in SBARQ.

??TRY to find some examples....

To evaluate the importance of those clauses in the text, we built a feature that calculate the maximum number of clauses per sentence. This feature was also inspired by Stab work [11].

```
double nbClauseSentence = selectCovered(S.class, root).size()
        + selectCovered(SBAR.class, root).size()
        + selectCovered(SBARQ.class, root).size()
        + selectCovered(SINV.class, root).size()
        + selectCovered(SQ.class, root).size();
```

Figure 3.8: Clause Ratio Feature

3.2.4 Sentiment Analysis Feature

Sentiment Analysis, or Opinion Mining refer to NLP techniques of detecting subjective information out of textual data. The research work related to this work really exploded over the past decade, especially when social media emerged and the availability of high subjective and sentimental data. In this work, we use the standard state-of-the-art tool for researchers which is *GPL Stanford Deep Learning for Sentiment Analysis*[6].

This tool assign to each sentence 5 percentage coefficients labelled `Very Negative`, `Negative`, `Neutral`, `Positive` and `Very Positive`. Those coefficients are calculated by recursive deep models that are detailed in Socher and Perelygin article [10].

---

[5]interrogative word or question word
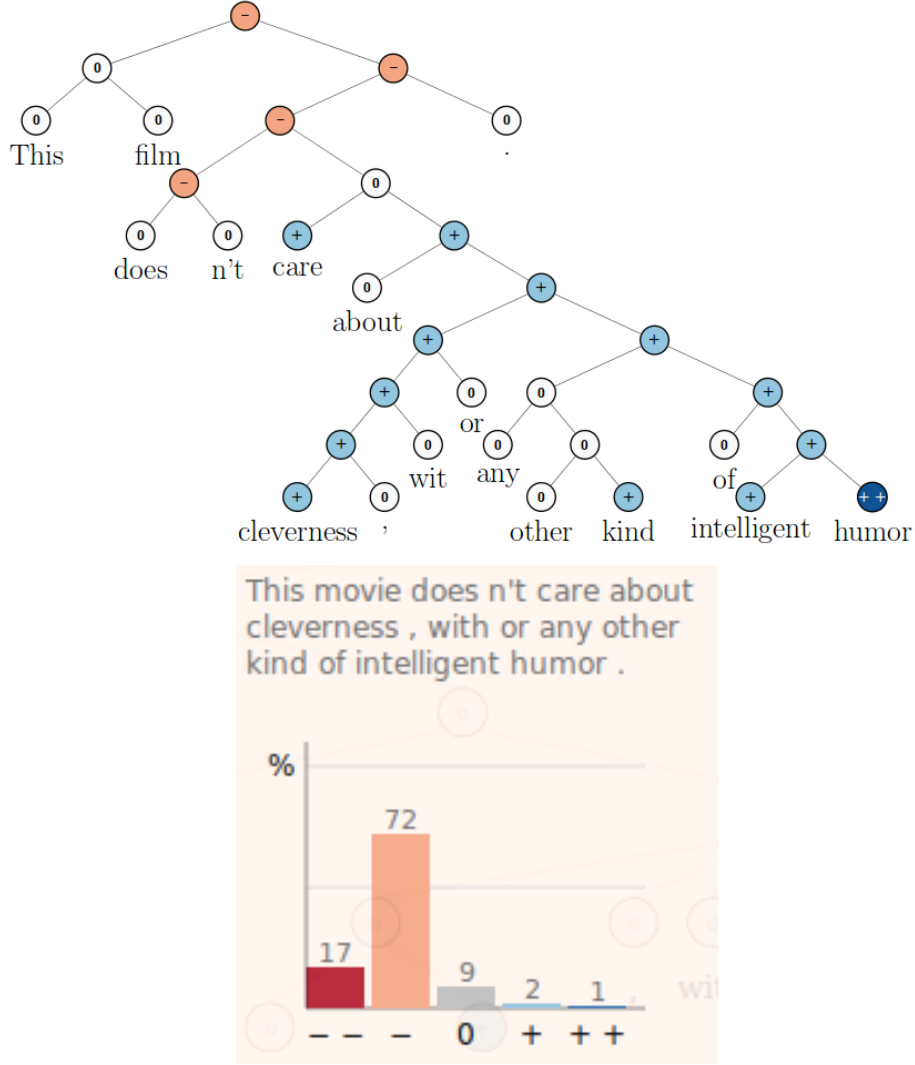[6]http://nlp.stanford.edu/sentiment/code.html

Figure 3.9: Recursive Neural Tensor Network and the resulting sentiment coefficients

Stanford's Sentiment Analysis tool was not available in DKPro, and we had to partially[7] integrate it in the pipeline.

3.2.4.1 Sentiment Coefficients

We call *sentiment coefficients* the 5 output coefficients returned by Stanford's Sentiment Analysis tool. Again, those coefficients are calculated on the sentences and thus, we have to perform a statistical analysis on the sentences.

We denote the 5 sentiment coefficients with symbols as follows: (-- - 0 + ++) and f_c is the function that given one sentiment c returns the corresponding coefficients in the sentence. Thus, we compute the minimum, the maximum, the average and the standard deviation of those 5 coefficients which gives us 20 metrics to evaluate the sentiment

---

[7]The two software don't work on the same annotations

distribution in our text:

$$\forall c \in \{\texttt{--}, \ \texttt{-}, \ \texttt{0}, \ \texttt{+}, \ \texttt{++}\}, \ \begin{cases} min_c = \min_{s \in text} \ f_c(s) \\ max_c = \max_{s \in text} \ f_c(s) \\ \mu_c = \frac{1}{|s|} \sum_{s \in text} f_c(s) \\ \sigma_c = \sqrt{\dfrac{\sum_{s \in text} (f_c(s) - \mu_c)^2}{|s|}} \end{cases}$$

3.2.4.2 Sentiment Fluctuation

The sentiments may vary slightly or significantly from a sentence to another. We can then define the *sentiment rules* which model the transition from one state to another. Since we have 5 type of coefficients, it results in 25 rules (ex: $- \rightarrow +, 0 \rightarrow ++, - \rightarrow -$).

In comparison with token n-grams and dependency rules, 25-dimensional binary vectors are built and represent the absence and the presence of a rule.

??SHOW an example

3.2.5 LDA

??BETTER title and explain

DKPro Core Overview

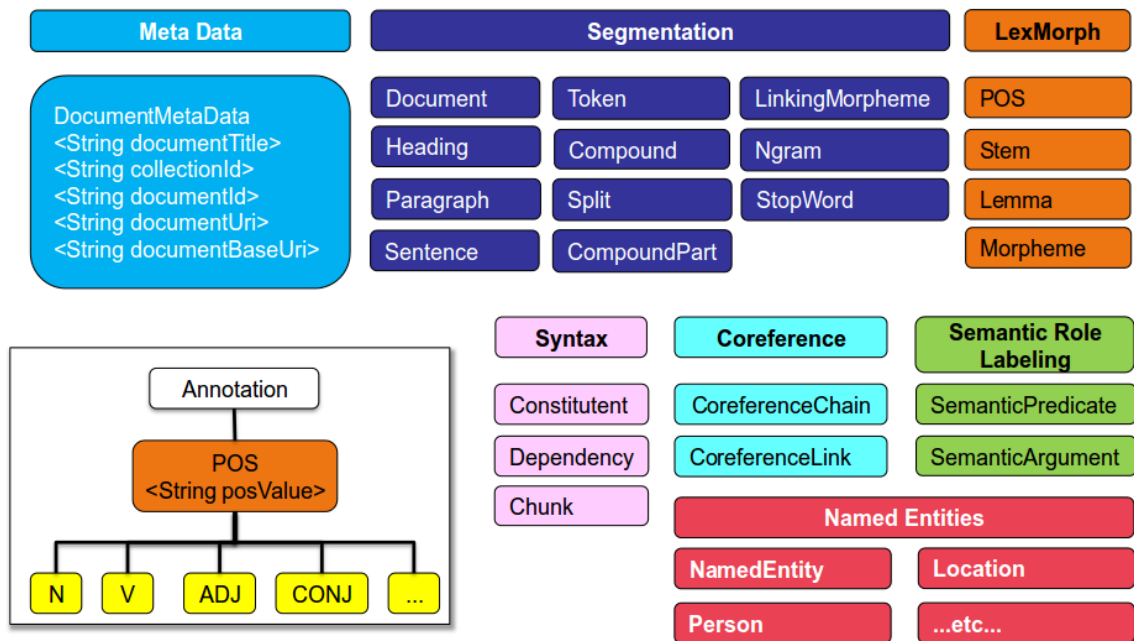Figure A.1: DKPro Type System

APPENDIX B

Maven

TODO: Explain the functionalities of Maven.

APPENDIX C

Statistical Analysis of Binary Classification

TODO: TP, FP, R, P, Accuracy, F-mes, ect...

Units of linguistic morphology

TODO: Words, Tokens, Lemma, Stemma

Parts Of Speech

TODO: Part of Speech description

Glossary

gold data Bla bla. 11

supervised learning Supervised learning is the machine learning task of inferring a function from labelled training data. 4

token Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens.. 12

Bibliography

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

[2] V. Ágel. *Dependency and valency: an international handbook of contemporary research*. Dependenz und Valenz: ein internationales Handbuch der zeitgenössischen Forschung. de Gruyter, 2006.

[3] Pranav Anand, Marilyn Walker, Rob Abbott, Jean E. Fox Tree, Robeson Bowmani, and Michael Minor. Cats Rule and Dogs Drool!: Classifying Stance in Online Debate. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, pages 1–9, Portland, Oregon, June 2011. Association for Computational Linguistics.

[4] 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.

[5] A.J. Freeley and D.L. Steinberg. *Argumentation and Debate: Critical Thinking for Reasoned Decision Making*. Wadsworth series in speech communication. Wadsworth/Thomson Learning, 2000.

[6] 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.

[7] Ivan Habernal, Judith Eckle-Kohler, and Iryna Gurevych. Argumentation mining on the web from information seeking perspective. In *Frontiers and Connections between Argumentation Theory and Natural Language Processing*, page (to appear), July 2014.

[8] Kazi Saidul Hasan and Vincent Ng. Stance classification of ideological debates: Data, models, features, and constraints. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1348–1356, 2013.

[9] Joakim Nivre and Johan Hall. A quick guide to maltparser optimization.

[10] Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank.

[11] Christian Stab and Iryna Gurevych. Identifying argumentative discourse structures in persuasive essays. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, page (to appear), Stroudsburg, PA, USA, October 2014. Association for Computational Linguistics.

[12] Julian R. Ullmann. A binary n-gram technique for automatic correction of substitution, deletion, insertion and reversal errors in words. *The Computer Journal*, 20(2):141–147, 1977.