École Centrale de Lyon
TFE 2014

DIPF - UKP
Frankfurt am Main

End-of-study internship report

Identification of Argumentative Texts in User-Generated Content on Educational Controversies

Anil Narassiguin

Tutors :

Option : Computer Science

ECL :
Alexandre Saidi

Speciality : Multimedia and Communication

Company :
Ivan Habernal

Profession : Research, Innovation and Development

Abstract

The goal of this work is the identification of persuasive and argumentative documents in user generated content, namely discussion forums and article comments. We have at our disposal a human-annotated corpus of size 990 documents enriched with metadata (topic, source, creation date) and classified as either persuasive (class P1) or non-persuasive (class P2).

We clearly have a binary classification problem. It will be the opportunity to use natural language processing (NLP) tools to process and extract useful knowledge from textual data, engineer fancy features to describe the data and finally utilize machine learning tools and evaluate our model predictions. Most of the tasks, from data processing to binary classification were performed thank to DKPro, a NLP software developed by the UKP laboratory.

Keywords

Natural language processing, argumentation mining, feature engineering, machine learning, SVM

Résumé

Ce travail vise à identifier automatiquement des documents persuasifs et argumentatifs dans des contenus générés par l'utilisateur, c'est-à-dire des forum en ligne et des commentaires d'articles. Nous avons à notre disposition un corpus de 990 documents enrichi de métadonnées telles le thème, la source et la date de création. Chaque document est classé en tant que persuasif (classe P1) ou non persuasif (classe P2).

Nous avons donc un problème de classification binaire, qui sera une opportunité d'utiliser des techniques de traitement automatique du langage naturel (TAL) pour extraire des connaissances utiles sur le corpus, de construire des descripteurs complexes et de découvrir des outils de machine learning. La plupart des tâches réalisées, du traitement des données jusqu'à la classification ont été réalisées grâce à DKPro, un logiciel de TAL développé par le laboratoire UKP.

Keywords

Traitement automatique du langage naturel, fouille de texte, argumentation mining, feature engineering, machine learning, SVM

Acknowledgements

I would like to thank my tutor at DIPF, Mr Ivan Habernal for his availability and advises during this internship. I would like also to thank my tutor at École Centrale de Lyon, Mr Alexandre Saidi for his advises during the past year for both the final year at ECL and the double degree at Université Lyon 1.
Finally, I would address a thank to all the people working at UKP in Darmstadt and DIPF in Frankfurt with one thought for my co-workers Tahir, Neil, and Clément and my office neighbours Ahcène and Daniel.

Contents

Introduction

UKP and DIPF

Ubiquitous Knowledge Processing Lab (also UKP Lab) is a research lab in the Department of Computer Science at the Technische Universität Darmstadt founded in 2006 by Prof. Dr. Iryna Gurevych. UKP Lab develops natural language processing techniques for automatically understanding written text and applies them to information management like information retrieval, question answering, and structuring information in Wikis. Its major realization is the Darmstadt Knowledge Processing Software Repository (DKPro) that offers robust, ready to use NLP components which are built on top of IBMs Unstructured Information Management Architecture (UIMA) as a common and open framework.

UKP has a partnership with the German Institute for International Educational Research (or DIPF, the German acronym) located in Frankfurt am Main. During my internship I worked with Dr. Habernal in the UKP team in DIPF offices. Our team focus on applying text mining and natural language processing tools to problems related to education.

The Project Schedule

As you can see in appendix A, the project was divided into several work packages and further sub-tasks. At the beginning, I started with reading some literature about argumentation mining and discovering the annotated corpus. Then, I had to use DKPro and create basic features. While gaining experience in programming and NLP, I went on building fancier features related to sentiment analysis and dependencies. Then I had to evaluate the results of models and perform an error analysis.

CHAPTER 1

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 classification, DKPro Text Classification Framework, will be introduced and its functionalities will be explained.

1.1 Argumentation

1.1.1 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. ([7], 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. ([7], p. 43)

Argumentation Reason giving in communicative situations by people whose purpose is the justification of acts, beliefs, attitudes, and values. ([7], p. 2)

Persuasion Communication intended to influence the acts, beliefs, attitudes, and values of others. ([7], p. 2)

1.1.2 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

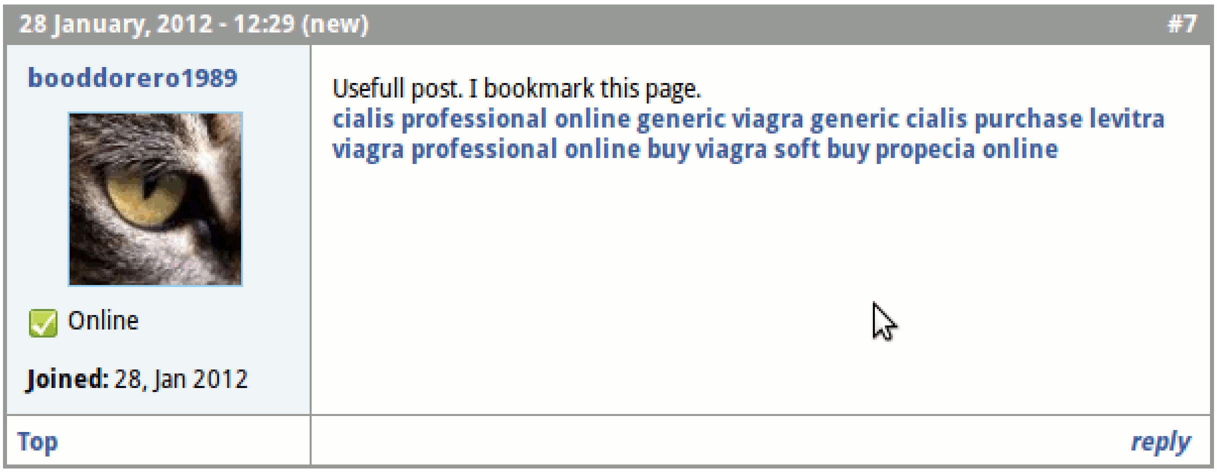
The author does not say explicitly that he/she is against home-schooling. However, he/she provides some examples (necessity of social interaction, total power of parents) and expresses feelings (sorry for home-schooled kids), so we can infer out the *implicit* message. This example is thus also *persuasive*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

1.1.3 Argumentation Mining

With the emergence of information technologies and especially social media, the availability of argumentative textual data is always bigger. People tend to express their point of view and their feelings in article comments, on forum posts, blogging platforms, ect... As most of the data available online, they can be very messy and a lot of comments or posts are indeed off topic, forum spams[1] or *trolls*[2].

[1] Forum spam consists of posts on Internet forums that contains related or unrelated advertisements, links to malicious websites, and abusive or otherwise unwanted information.

[2] In Internet slang, a troll is a person who sows discord on the Internet by starting arguments or upsetting people, by posting inflammatory, extraneous, or off-topic messages in an online community (such as a newsgroup, forum, chat room, or blog) with the deliberate intent of provoking readers into an emotional response or of otherwise disrupting normal on-topic discussion.



28 January, 2012 - 12:29 (new) #7

booddorero1989

Online

Joined: 28, Jan 2012

Usefull post. I bookmark this page.
cialis professional online generic viagra generic cialis purchase levitra viagra professional online buy viagra soft buy propecia online

Top reply

Figure 1.1: Example a the viagra spam in a forum

Even so, it might be not trivial for some cases to judge if the author is completely off-topic or if he uses *sarcasm* or *irony* to support his opinion. One of the main step of *Argumentation Mining*, as you can see on the dark grey rectangle of 1.2 is the detection of relevant documents, means the one that clearly show their argument but also the one which are persuasive but in an *implicit* way.

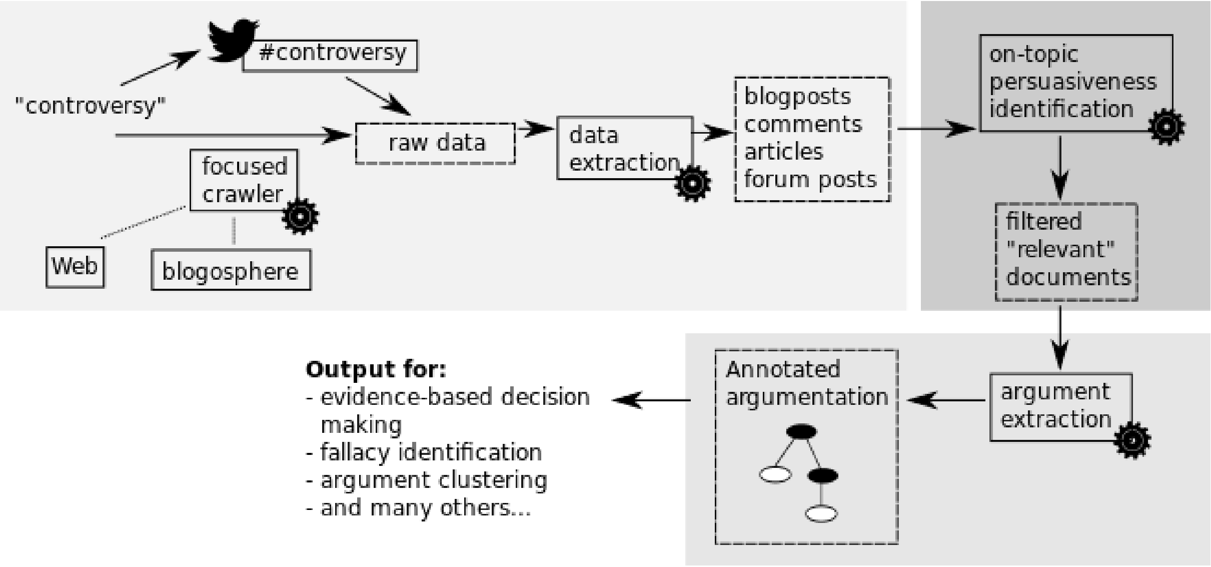


Figure 1.2: The process of argumentation mining

1.2 NLP and the DKPro Framework

1.2.1 Natural Language Processing

Natural Language Processing or *NLP* is a multidisciplinary field that combines Linguistics, Computer Science, Artificial Intelligence and its modern approach, Machine learning.

NLP was actually the main goal of Computer Science's pioneers: write programs that would be able to understand human speech as it is written and spoken. This Fantasy which is becoming more and more true gives the actual definition of NLP: ability of a computer program to understand humans natural language.

The development of NLP applications is challenging since computers and programming languages are highly structured contrary to human language which is not always precise. A lot of misunderstands come from ambiguities in intonation, context (a certain sentence can be serious or sarcastic in different contexts), the social background or the dialect used. Even so, a lot of progress have been made, and here is a non-exhaustive list of fields that emerged from NLP:

- Sentence segmentation, part-of-speech tagging and parsing.
- Speech Recognition
- Automated Translation
- Automatic Summaries

In this work, we'll see how NLP can help us in Argumentation Mining and Persuasiveness detection.

1.2.2 UIMA

DKPro stands for *Darmstadt Knowledge Processing* [8] 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:

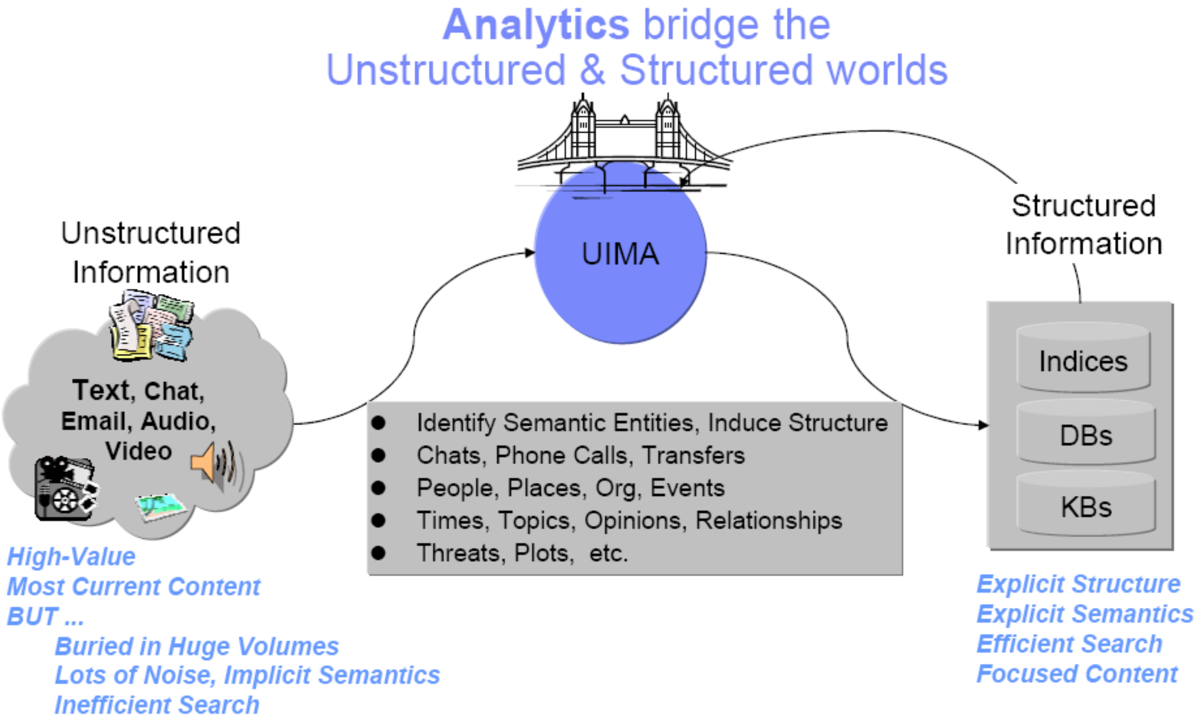


Figure 1.3: UIMA: Order unstructured data, from UIMA website [1]

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).

1.2.3 DKPro Core

Many NLP tools are already freely available in the NLP research community. DKPro Core [6] provides UIMA components wrapping these 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, parts-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.

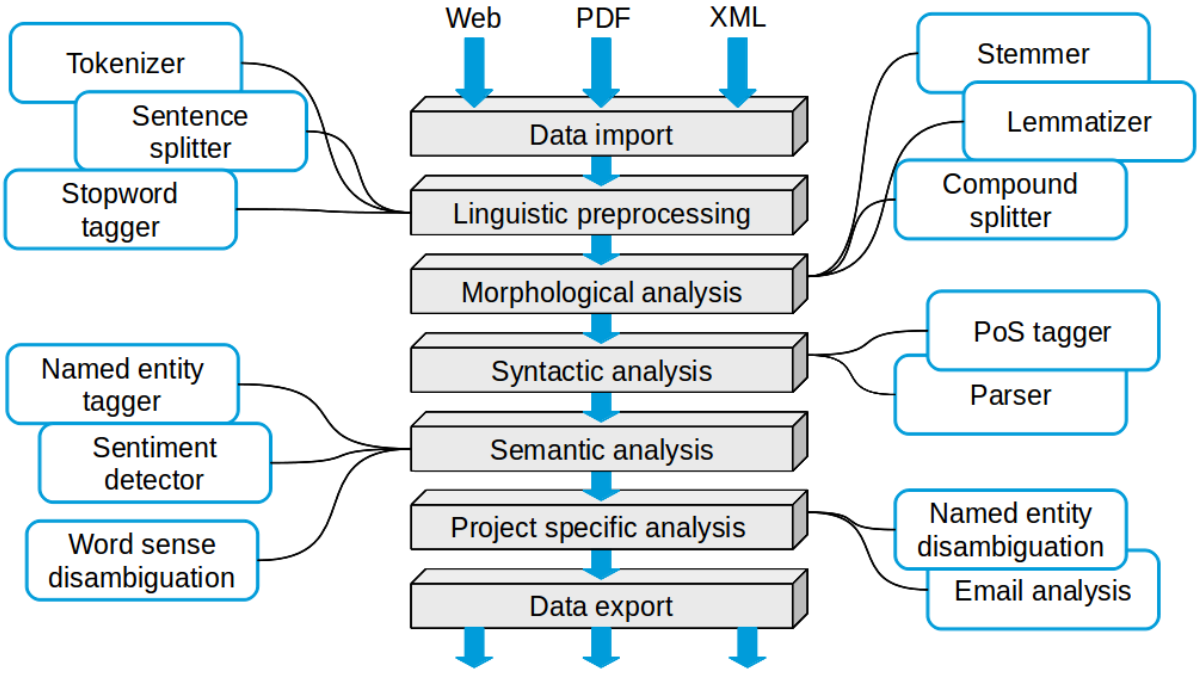


Figure 1.4: DKPro Core Pipeline

Core has several annotators either developed in-house or wrapped[3] 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...

1.2.4 DKPro Text Classification

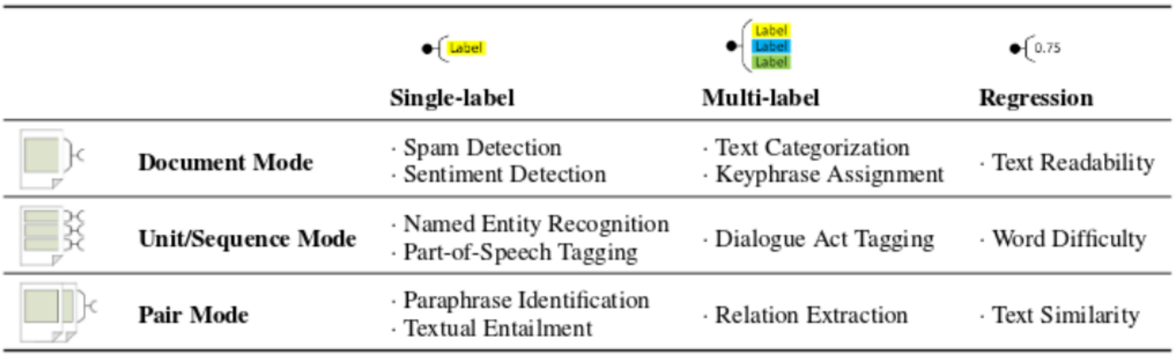
The aim of DKPro TC (commonly called TC) is to allow the user to apply machine learning algorithms easily on the extracted annotations. This framework was built in order to execute the following tasks:

- 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 classification, multi-label classification classification and regression.

[3] 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[4] (*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 communicates with 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* [5] file, which corresponds to Weka's format files.


Label
Label
Label
Label
0.75

		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 1.5: 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.
- Extraction of the features from the annotations.
- Data Processing and Cross Validation (appendix D)
- Report of the results (Accuracy, Macro F-Measure, ect...)

The TC developers give regularly some helpful tutorials on their website[6].

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).
The SVN[7] commits of the different modules are available through Google code and the

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

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

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

[7] 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.

integration of new functionalities in a project is made possible by Maven[8]. 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.

[8] Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

CHAPTER 2

The Research Work

In this part, we'll see how the theoretical knowledge in Argumentation Theory and NLP presented previously 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. We'll also have a brief look at the classifier algorithm we're using to perform our classification.

2.1 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:

174_P2_artcomment_homeschooling.txt

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 annotator comments, the conflicts, ect... (appendix C)

2.1.1 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 [9] in order for the annotators to understand the task. In this section, we'll discuss about the general ideas of this guideline.

2.1.1.1 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
```

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

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

2.1.1.3 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.

2.1.1.4 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 D 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.

2.1.1.5 Results of the Manual Annotations

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

				Persuasive			Non-Persuasive		
	Docs	Macro F_1	Acc.	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 2.1: Human performance on *gold data persuasive.*

The Macro F-measure (here called F_1) and the accuracy are high enough to consider an automatic classification based on machine learning.

2.1.2 Corpus Statistics

Now that we have the *gold labels* for the texts, we can sum the relevant information in a table:
Abbreviations:
`RS = redshirting`

		RS	PIS	HS	SSE	MS	PPS	Total
P1	artcomment	24	60	64	17	1	278	444
	forumpost	14	17	22	9	9	9	80
	all	38	77	86	26	10	287	524
P2	artcomment	15	43	93	16	2	174	343
	forumpost	15	23	45	8	17	15	123
	all	30	66	138	24	19	189	466
P1 + P2	Total	68	143	224	50	29	476	990
	Percentage	6.9	14.4	22.6	5.1	2.9	48.1	100

```
PIS = prayer in school
HS = homeschooling
SSE = single sex education
MS = mainstreaming
PPS = public private schools
```

2.1.3 Conflict Annotations

When an annotator doesn't agree with the others about the class of a certain text, a discussion takes place between the three annotators to try to conclude about the class. If after the discussion, the annotator still doesn't agree with his colleagues, the *gold class*[2] is set as the majority class among the annotators but in the metadata of the text, it will be specified that the text was conflicting. Knowing if the text was conflicting will help us to perform the error analysis in the last parts of this report.

Here are some statistics about conflict annotations:

	Non Conflict	Conflict
P1	393	131
P2	346	120
Total	739	251

2.2 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*).

[2] The class given to a text after the annotation process

2.2.1 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[3], punctuations, ect...

2.2.1.1 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 a text (*Note:* For more information about units in linguistics such as words, tokens, lemma and stemma, have a look at the glossary). The study of n-grams distribution in a corpus is an ancient technique [16] 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 if the text contains the n-gram, 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.

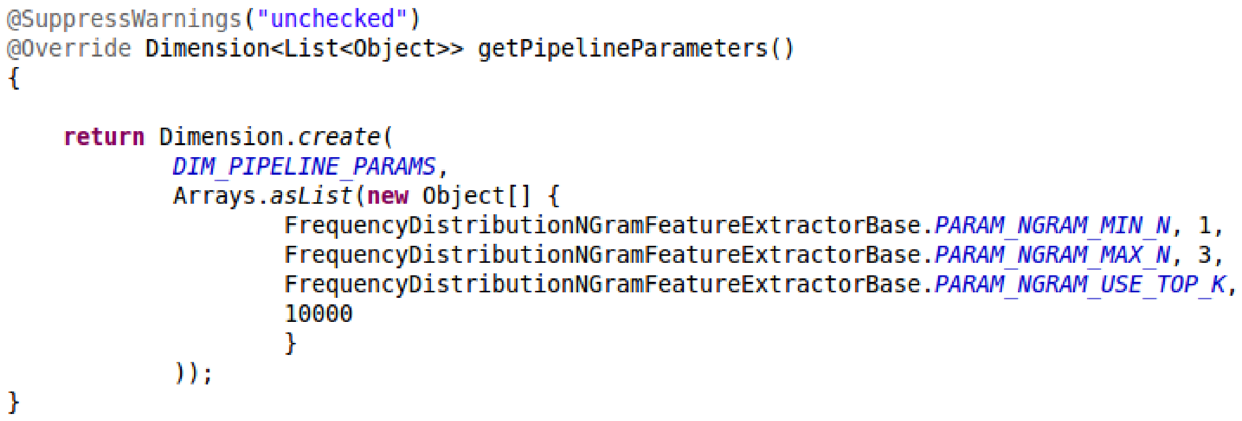
As displayed on 2.1, the n-gram feature extractor takes three parameters: `PARAM_NGRAM_MIN_N` for the minimal value of n, `PARAM_NGRAM_MAX_N` for the maximal value of n and `PARAM_NGRAM_USE_TOP_K` which is the number of common n-grams retained. Since we want the 10000 most common 1,2 and 3-grams, the parameters are set such as follow:

```
PARAM_NGRAM_MIN_N = 1

PARAM_NGRAM_MAX_N = 3

PARAM_NGRAM_USE_TOP_K = 10000
```

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



```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 2.1: NGram Feature in DKPro Java code

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.

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.

2.2.1.2 Tokens and Sentences

We compute some statistics regarding tokens 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 often 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.

2.2.1.3 Other Lexical Statistics

In the article *Stance Classification of Ideological Debates*[10], 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.

2.2.1.4 Punctuation Related Features

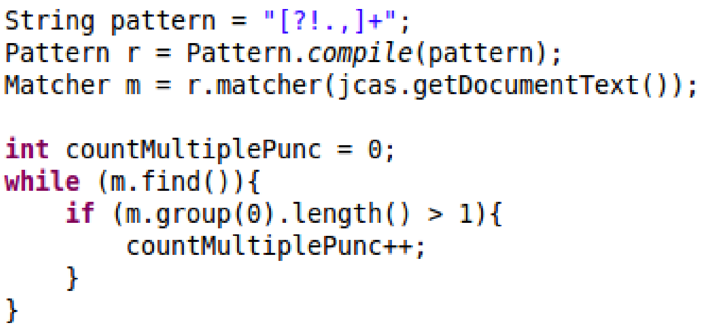
First, we defined a set of features which simply compute the ratio per token of these 6 punctuation marks full stop, comma, question mark, exclamation mark, colon and quotation mark (fig. 2.2). This would tell us if the author is caring about his style of writing.

`. , ? ! : "`

Figure 2.2: 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*[4]: `[?!.,]+`

[4] 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.



```
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 2.3: 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:

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 disagreeable. 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.

2.2.1.5 Multiple Capital Letters

Another feature that can reflect the lack of seriousness is the number of words with multiple 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!!

2.2.2 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.

2.2.2.1 Ratio on common POS

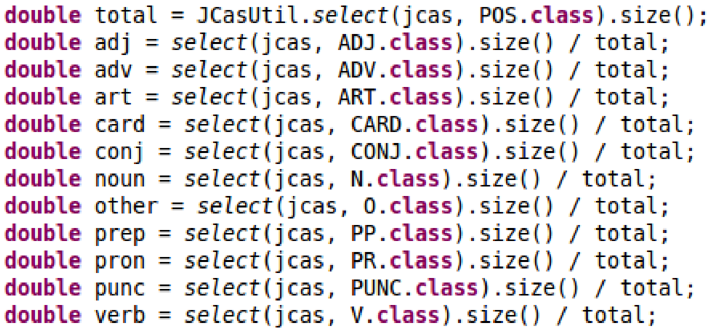
One simple feature, already implemented in DKPro[5], computes 11 ratios of 11 different over the total number of POS:

The DKPro functions allow to compute the ratios and thus create the features very easily as seen on fig. 2.4.

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

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 2.2: The 11 POS we consider for the Ratio POS feature



```
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 2.4: Piece of code for the multiple punctuation feature

2.2.2.2 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.

2.2.2.3 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.

2.2.2.4 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.

2.2.3 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.

2.2.3.1 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 and arduous and we leave to the reader the study of dependency trees. Nevertheless, we give in the following figure (2.5), a simple example of a dependency tree.

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

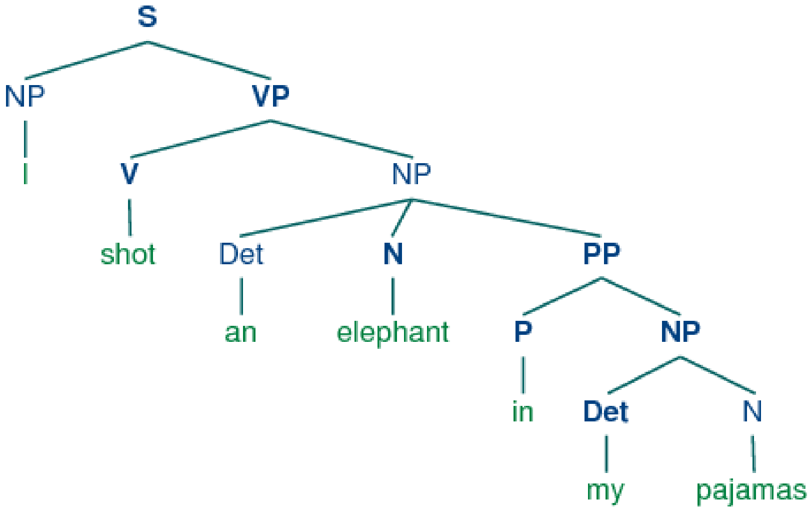


Figure 2.5: A dependency tree

To quantify how complex a sentence can be, we took inspiration on Christian Stab work on Argumentative Discourse [15] 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. 2.5, the depth of the tree is 5.

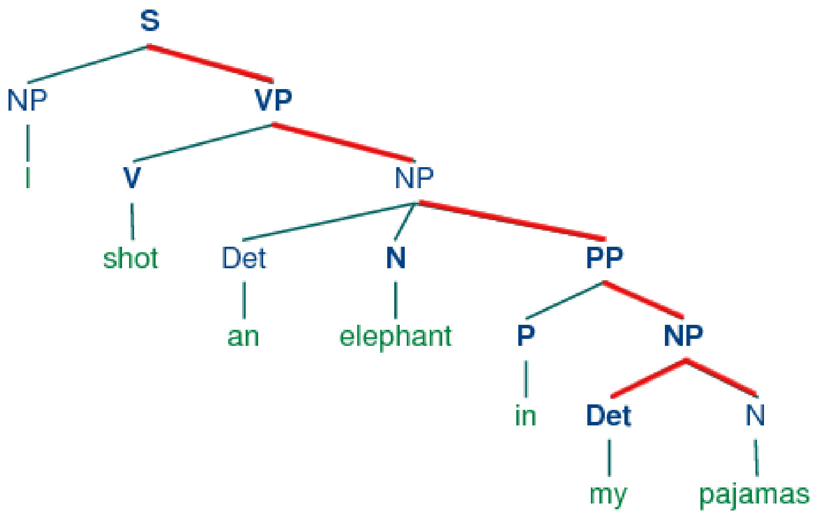


Figure 2.6: 5 edges between the summit S and the extremities Det and N

Building features with this metric

The dependency tree is available on DKPro with the *MaltParser*[12]. 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 2.7: 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:

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

2.2.3.2 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 [15]. As an example, some of the dependency rules from the previous tree (fig. 2.5) 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.

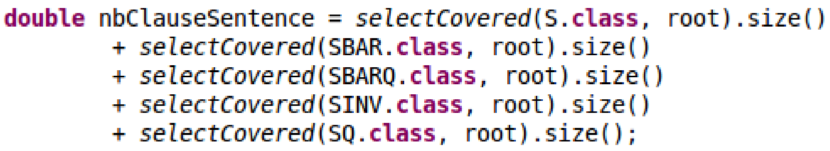
2.2.3.3 Subordinate clauses

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*[6] 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.

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 [15].



```
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 2.8: Clause Ratio Feature

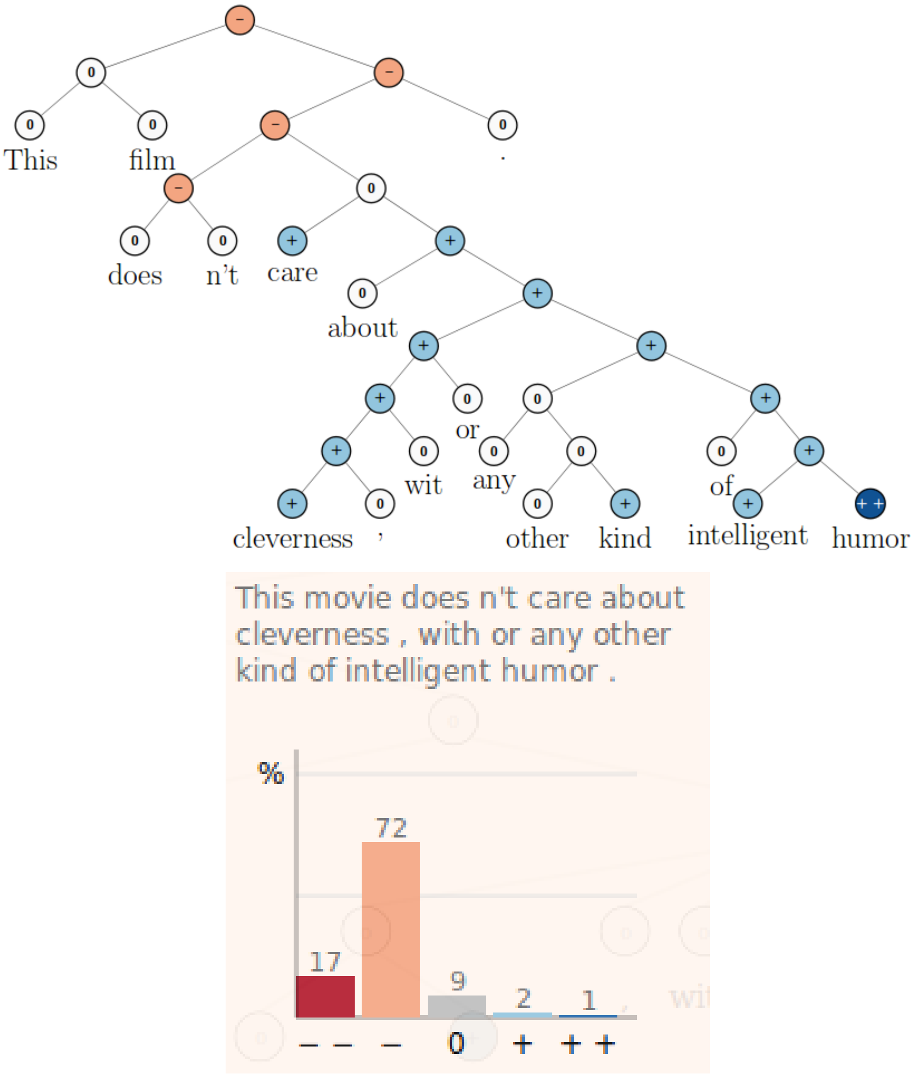
2.2.4 Sentiment Analysis Feature

Sentiment Analysis, or Opinion Mining refer to NLP techniques of detecting subjective information out of textual data. The research work related to the field really exploded over the past decade, especially when social media and its corollary, the availability of high

[6] interrogative word or question word

subjective and sentimental data, emerged. In this work, we use the standard state-of-the-art tool for researchers which is *GPL Stanford Deep Learning for Sentiment Analysis*[7].

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 [14].



This movie does n't care about cleverness , with or any other kind of intelligent humor .

Figure 2.9: Recursive Neural Tensor Network and the resulting sentiment coefficients

Stanford's Sentiment Analysis tool was not available in DKPro, and we had to partially[8] integrate it in the pipeline.

2.2.4.1 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.

[7] http://nlp.stanford.edu/sentiment/code.html

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

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 distribution in our text:

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

2.2.4.2 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. If we consider as input the two following sentences:

```
It is a choice.
Independent choice is what makes American values so precious.
```

The corresponding sentiment fluctuation after using the sentiment analysis tool is: $0 \rightarrow +$

2.2.5 LDA

The Latent Dirichlet Allocation is a generative model[9] that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. Intuitively, its a way of automatically discovering topics that these sentences contain. Suppose you have the following set of sentences:

1) I like to eat mango and apples.
2) I ate a mango and carrot smoothie for breakfast.
3) Horses and cats are cute.
4) My sister adopted a cat yesterday.
5) Look at this cute hamster munching on a piece of carrot.

Intuitively, The LDA will produce those rules:

- Sentences 1 and 2: 100% Topic α
- Sentences 3 and 4: 100% Topic β
- Sentence 5: 60% Topic α, 40% Topic β

[9] A generative model is a model for randomly generating observable data, typically given some hidden parameters.

- Topic α: 30% mango, 15% apple, 10% breakfast, 10% munching... (food and eating)
- Topic β: 20% cat, 20% horses, 20% cute, 15% hamster... (animals)

LDA performs its discovery by representing documents as mixture of topics that spit out words with certain probabilities. In this section, we introduce LDA in a intuitive way but please have a look at Blei, Jordan and Ng work[4] on it.

2.3 The Classifier and Performances

In Data Mining, once we have defined all the features that describe our data, we need to train a model on those features and evaluate the performances of the model. Even if the results can vary a lot from one classifier to another, in this study we're more interested in how perform the feature rather than how perform a certain classifier. Thus, we'll use a common state-of-the-art classifier called *Support Vector Machine* or *SVM.*

2.3.1 SVM Classifier

In the field of machine learning, SVMs are supervised learning models that perform classification (also regression, but it's not the purpose of our study) by finding the hyperplane that maximizes the margin between the two classes. The vectors that define the hyperplane are called *support vectors* [5].

The main advantage of SVM is that, as we'll see later, it can separate non linearly separable data thank to its *kernel* by adding dimensions and transform the data into linearly separable data, as it can be seen on the following figures:

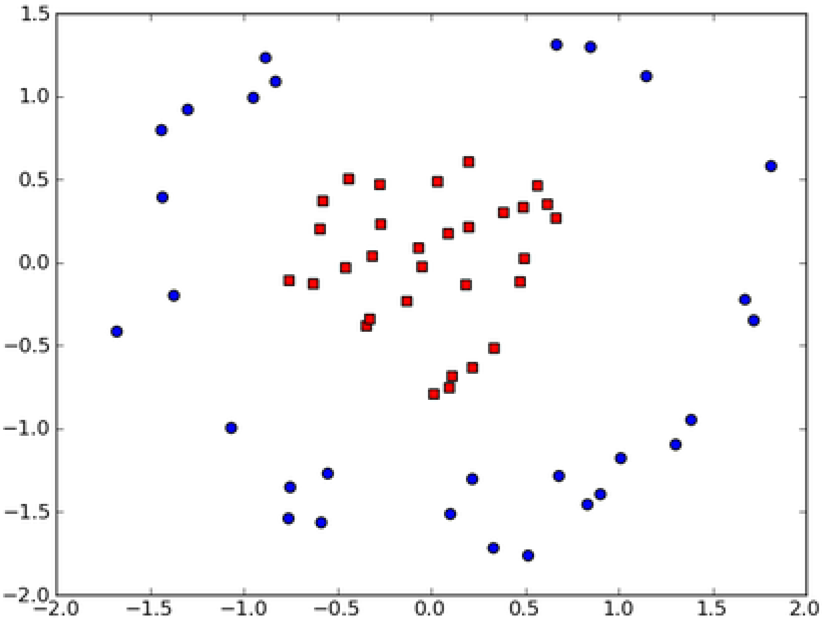


Figure 2.10: In this 2D representation, the blue and red classes are not linearly separable

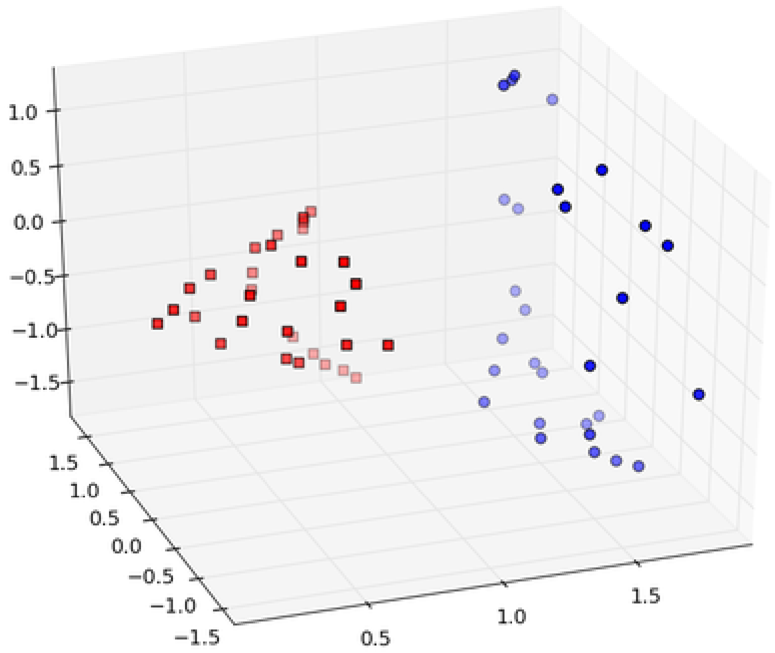


Figure 2.11: By transforming the data and adding one dimension, the classes are linearly separable

We shortly outline the different steps of the SVM algorithm:

Algorithm

- Define an optimal hyperplane: maximize margin
- Extend the above definition for non-linearly separable problems: have a penalty term for misclassifications.
- Map data to high dimensional space where it is easier to classify with linear decision surfaces: reformulate problem so that data is mapped implicitly to this space.

To specify in more details, here is how the optimal hyperplane and the margin are defined:

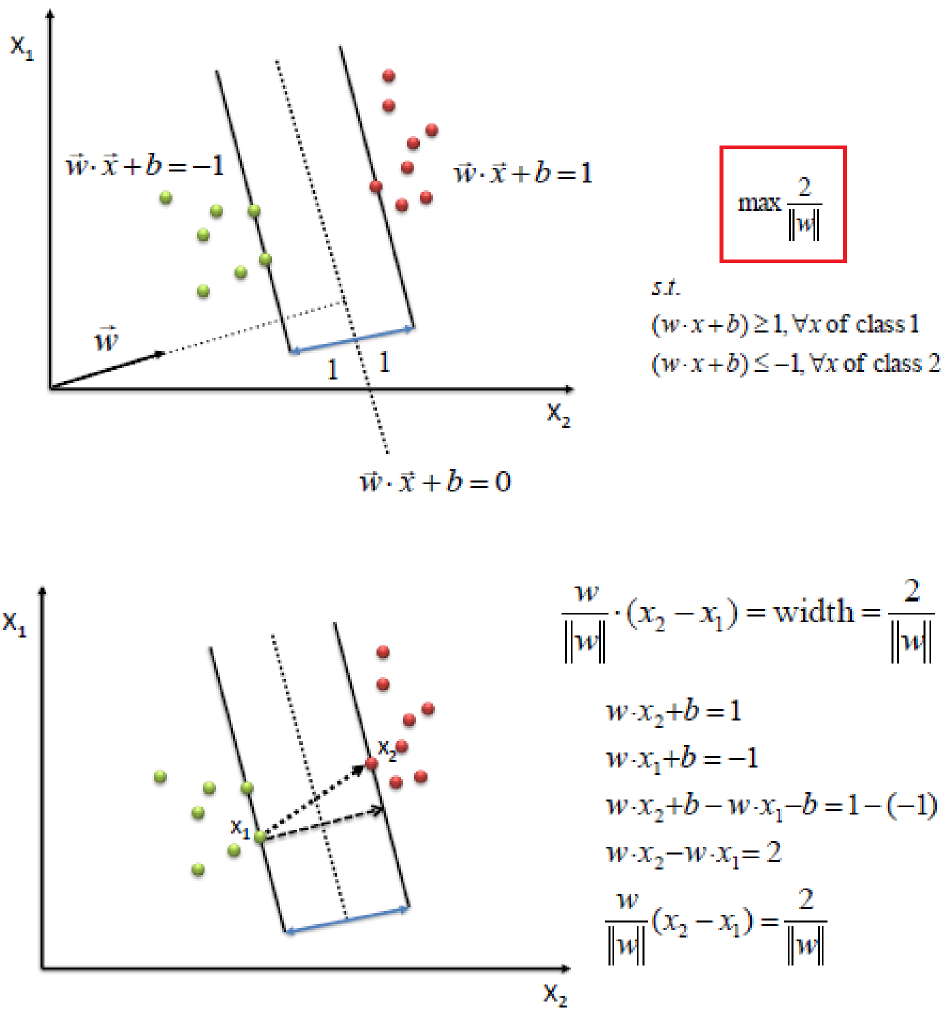

X1
$\vec{w}\cdot\vec{x}+b=-1$
$\vec{w}\cdot\vec{x}+b=1$
$\max\frac{2}{\|w\|}$
s.t.
$(w\cdot x+b)\geq 1, \forall x$ of class 1
$(w\cdot x+b)\leq -1, \forall x$ of class 2
$\vec{w}$
1
1
X2
$\vec{w}\cdot\vec{x}+b=0$
X1
$\frac{w}{\|w\|}\cdot(x_2-x_1)=\text{width}=\frac{2}{\|w\|}$
x2
x1
$w\cdot x_2+b=1$
$w\cdot x_1+b=-1$
$w\cdot x_2+b-w\cdot x_1-b=1-(-1)$
$w\cdot x_2-w\cdot x_1=2$
$\frac{w}{\|w\|}(x_2-x_1)=\frac{2}{\|w\|}$
X2

Figure 2.12: Margin and Hyperplane Optimization for SVM

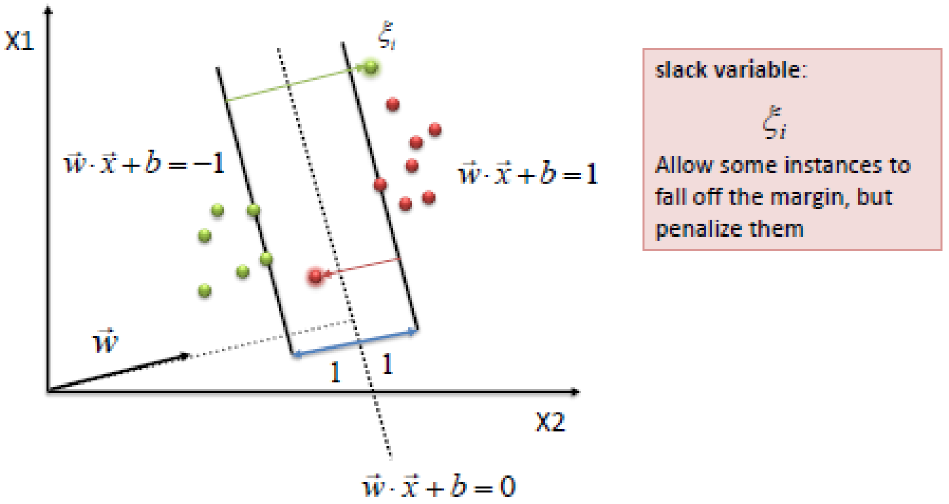
The geometric parameters b and w are found using *Quadratic Programming*[10] on this following function:

$$min\ \frac{1}{2}||w||^2$$

$$s.t. y_i\ (w\ x_i + b) \geq 1, \forall x_i$$

If the data is linearly separable, the solver is supposed to find a unique minimum. Ideally SVM analysis should create an hyperplane that completely separates the classes into two non-overlapping groups. However, in practice, perfect separation may not be possible, or it may result in a model with so many cases that the model does not classify correctly. In this situation SVM finds the hyperplane that maximizes the margin and minimizes the misclassifications. Thus the slack variable is introduced: It correspond to the ratio of variables that are allowed to fall off the margin.

[10] Quadratic programming (QP) is a special type of mathematical optimization problem. It is the problem of optimizing (minimizing or maximizing) a quadratic function of several variables subject to linear constraints on these variables.


X1
ξ_i
$\vec{w}\cdot\vec{x}+b=-1$
$\vec{w}\cdot\vec{x}+b=1$
$\vec{w}$
1
1
X2
$\vec{w}\cdot\vec{x}+b=0$

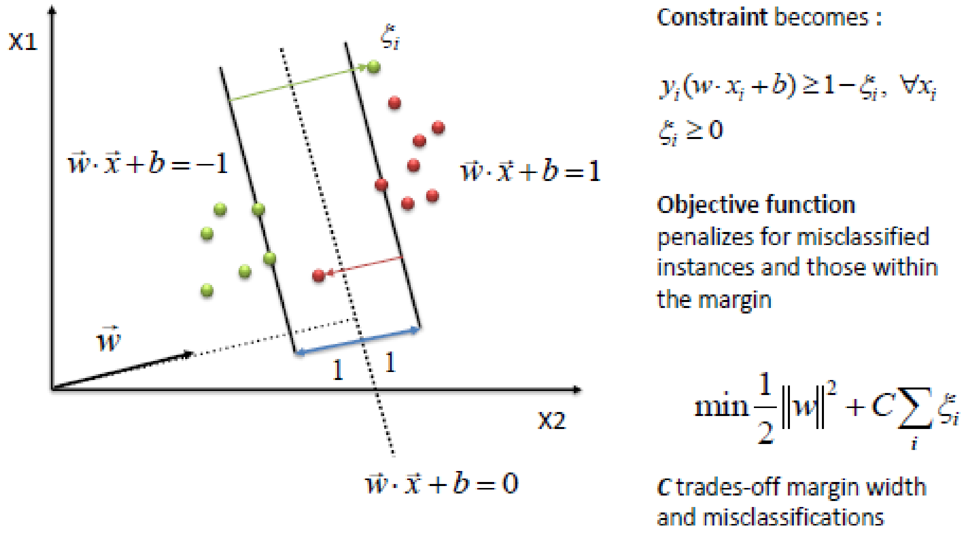
slack variable:

ξ_i

Allow some instances to fall off the margin, but penalize them

Figure 2.13: Slack variable: In this example, two instances are misclassified

SVM tries to maintain the slack variable as close to zero as possible while maximizing margin. However, it does not minimize the number of misclassifications (In computational complexity theory, NP-complete problem[11]) but the sum of distances from the margin hyperplanes. Besides, with the introduction of the new variables, the objective function and the constraints change:


X1
ξ_i
$\vec{w}\cdot\vec{x}+b=-1$
$\vec{w}\cdot\vec{x}+b=1$
$\vec{w}$
1
1
X2
$\vec{w}\cdot\vec{x}+b=0$

Constraint becomes :

$y_i(w\cdot x_i+b)\geq 1-\xi_i,\ \forall x_i$

$\xi_i \geq 0$

Objective function penalizes for misclassified instances and those within the margin

$$\min \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

C trades-off margin width and misclassifications

Figure 2.14: New optimization problem

The simplest way to separate two groups of data is with a straight line (1 dimension), flat plane (2 dimensions) or an N-dimensional hyperplane. However, there are situations where a non-linear region can separate the groups more efficiently. SVM handles this by using a kernel function (non-linear) to map the data into a different space where a

[11] NP-complete problems are in NP, the set of all decision problems whose solutions can be verified in polynomial time

hyperplane (linear) cannot be used to do the separation. It means a non-linear function is learned by a linear learning machine in a high-dimensional feature space while the capacity of the system is controlled by a parameter that does not depend on the dimensionality of the space. This is called kernel trick which means the kernel function transform the data into a higher dimensional feature space to make it possible to perform the linear separation.

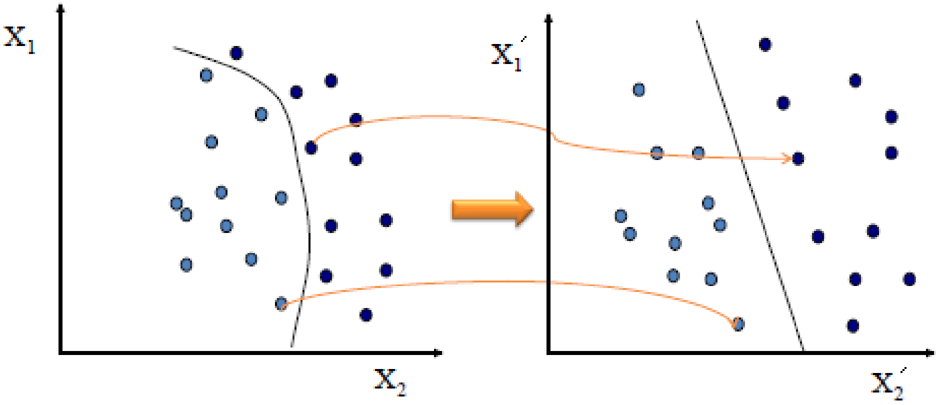


Figure 2.15: Kernel Function: Non-linear separation

The two kernel functions used in practice are the following:

Polynomial Kernel

$$\forall x_i, x_j \ instances, \ \ k(x_i, x_j) = (x_i.x_j)^d, d \in \mathbb{N}$$

Gaussian Kernel

$$\forall x_i, x_j \ instances, \ \ k(x_i, x_j) = exp(-\frac{||x_i - x_j||^2}{2\sigma^2})$$

2.3.2 Weka's SMO Classifier

The machine learning tools available on DKPro TC are based on *Weka* components. The implementation we'll use for SVM is John C. Platt's *Sequential Minimal Optimization* (SMO) algorithm[13]. This implementation globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes by default. We'll use the polynomial kernel in our experiments since it was *a posteriori* the most efficient one.

2.3.3 Evaluate the performances of our classification

To evaluate the performances of our classification, we should define a bunch of metrics that quantify the prediction capacities of our system (features + algorithm). We'll first see the different scenarios of prediction and then the measures we use to quantify the performances of our prediction. With the type of data we have, we can define 4 types of scenarios. If certain terms are unknown for you, please have a look at appendix D on

Binary Classification.

The In-Domain Cross-Validation We perform a 10 folds cross-validation (appendix D) for all the data in one of the six domains we previously defined. At the result of, for every experiment we run, we get 6 in-domain cross-validation. The reported results are the *Accuracy* and teh *Macro F-measure* as defined in appendix D.
For the domain homeschooling, we'll perform a *leave-one-out* cross-validation[12]

The Full Cross-Validation We perform a 10 folds cross-validationD for all the data in one of the six domains we previously defined. At the result of, for every experiment we run, we get 6 in-domain cross-validation.

The Cross-Domain Validation For each domain, the data related to the domain is considered as a test set and the rest of the data (5 other domains) forms the test set. This evaluation scenario is somehow the most important since it reveals how *general* are the features and how well they can predict the persuasiveness in general.

The Full Cross-Domain Validation We aggregate the results of the 6 cross-domain validations such as follow:
If *Dom* is a domain in D = {homeschooling, redshirting, prayer-in-schools, public-private-schools, mainstreaming, single-sex-education} and TP_{Dom}, TN_{Dom}, FP_{Dom} and FN_{Dom} are the statistical measures for the performances of the binary classification (appendix D), the confusion matrix of the Full Cross-Domain Validation scenario is defined as followed:

	Prediction	
Actual Value	TP_{FCD}	FP_{FCD}
	FN_{FCD}	TN_{FCD}

With:

$$TP_{FCD} = \sum\nolimits_{Dom \in D} TP_{Dom}$$

$$TN_{FCD} = \sum\nolimits_{Dom \in D} TN_{Dom}$$

$$FP_{FCD} = \sum\nolimits_{Dom \in D} FP_{Dom}$$

$$FN_{FCD} = \sum\nolimits_{Dom \in D} FN_{Dom}$$

The *Accuracy* and the *F-measure* for this scenario directly come from the previous measures.

[12] Leave-one-out cross-validation (LOOCV) is a particular case of n-fold cross-validation with n = number of instances. Since we have 29 text data for homeschooling, we perform a 29 folds cross-validation.

CHAPTER 3

Results and Perspective

Once that the features used and the classification method (SVM algorithm) are defined, we run the pipeline on our *gold data*. In this part we'll show the results returned by DKPro and we'll try to comment them. Then with the error analysis, we'll try to understand why our classification method has still some issues to recognize certain instances (a machine learning method would have some issue to understand *irony* or *sarcasm*). We'll also determine the features that contain high information thank to feature selection methods. Finally we'll have a look at extra tasks that should be done in the future.

3.1 Performance Evaluation

In this section, we display the macro F-measure for 11 different sets of features. Except if it's indicated, for every row, we consider the feature that is written in the cell in addition with the previous ones (for example for *POS Ratio*, we consider *POS Ratio* features and *Baseline* features).

Full Cross-Validation and Full Cross-Domain

Features	Full CV	Fm Full CD
Baseline	68.92	60.07
Only Lemma NGrams	63.81	58.32
POS Ratio	68.83	60.75
1-Skip 2-grams	65.55	55.52
POS NGrams + Sub-clause Ratio	64.90	57.91
Sentiment Analysis	67.80	60.32
Only Baseline + Sentiment Analysis	67.70	60.30
Sentiment Fluctuation	67.61	60.21
30 Dependency Rules	67.60	58.51
5000 Dependency Rules	66.73	60.90
LDA	68.12	60.22

The best result is highlighted in dark grey and the second best result is highlighted in light grey.

This table shows us that the different configurations are more or less equivalent (indeed, the confidence intervals overlap in most of the cases) which mean that the Baseline performs as good as fancier methods. We need to investigate the results for the in-domain cross-validation and the cross-domain validation to see if this trend is going on.

In-Domain Cross-Validation

It should be reminded that for the mainstreaming domain, a leave-one-out cross-validation is performed instead of a 10-folds CV.

Features	RS	PIS	HS	SSE	MS	PPS
Baseline	51.37	73.94	70.53	65.32	39.58	68.5
Lemma NGrams	55.89	71.23	67.35	74.0	43.50	61.71
POS Ratio	52.78	74.62	70.13	65.32	39.58	70.25
1-Skip 2-grams	45.29	64.34	70.10	75.37	39.58	68.57
POS NGrams + Sub-clause Ratio	59.52	69.78	70.09	77.92	39.58	66.90
Sentiment Analysis	54.16	73.32	71.55	67.53	39.58	70.57
Only Baseline + Sentiment Analysis	52.78	72.64	71.55	65.32	39.58	70.00
Sentiment Fluctuation	52.78	75.35	72.93	69.70	39.58	71.26
5000 Dependency Rules	49.60	73.79	68.11	73.91	39.58	68.85
LDA	50.95	60.22	69.73	73.91	39.58	68.66

Cross-Domain Validation

Features	RS	PIS	HS	SSE	MS	PPS
Baseline	54.16	52.62	61.02	62.50	61.84	65.90
Lemma NGrams	51.38	50.10	60.36	60.73	57.35	62.00
POS Ratio	54.16	53.23	62.67	60.07	61.84	66.27
1-Skip 2-grams	48.89	51.79	58.34	47.92	38.75	62.44
POS NGrams + Sub-clause Ratio	51.21	48.91	55.60	65.65	63.39	62.58
Sentiment Analysis	55.53	53.02	61.17	60.07	61.84	64.83
Only Baseline + Sentiment Analysis	55.53	52.39	60.91	60.07	58.95	65.81
Sentiment Fluctuation	51.20	53.44	62.16	62.50	48.73	64.10
5000 Dependency Rules	59.87	53.84	62.40	67.79	48.73	64.21
LDA	61.46	53.22	62.40	63.47	51.48	63.39

Some significant improvements can be seen for the domains redshirting, single sex education, and public private schools, but not for the same feature sets configurations. For example, if we consider the in-domain cross-validation, sentiment analysis works pretty well for prayer in schools and homeschooling. Those debates are very sensible and lead to highly sentimental data, such as follow:

> **1279_P2_artcomment_prayer-in-schools.txt** I love the posters that always claim religion is the route of all evil, violence and strife. Yet socialism/communism are responsible for up to 200 million dead leading up to the 21st century, and still counting. All the religious wars throughout human history don't even come close to these atheist murderers! The one parameter that all the accusers fail to recognize, is that they are all run by Humans. There is something seriously wrong with Homo Sapiens.

Regarding the cross-domain scenarios, it seems that *LDA* and dependencies rules are the two techniques which work the best and thus fit the best features which can detect persuasiveness among different domains. This is due to the fact that argumentation structures tend to be similar and thus dependency rules and the graphs of topics (for the LDA) tend to be the same.

To better understand our actual results, we'll perform an analysis on the feature by selecting the ones which gather most of the information: it's *Feature Selection*.

3.2 Feature Selection

We consider the case where we take all the features into account (except the POS N-Grams and the Lemmas). Here is an overview of the types of features we have and their distribution.

Type of Feature	Number
Baseline	10000
Part of Speech (POS)	16
Lexical	15
Syntactic	5010
Sentiment	45
LDA	30
TOTAL	15116

Even before the selection, we see a clear domination of baseline features which corresponds to n-grams (with n=1,2,3) presence in a text. The high number of syntactic features is due to the 5000 dependency rules that are extracted from the corpus.

Information Gain
Information Gain is a measure that evaluates the worth of an attribute by measuring an entropy difference. Information Gain is calculated such as follows:

$$InfoGain(Class, Feature) = H(Class) - H(Class|Feature)$$

Where H is the information entropy which in our case (two classes P1 and P2) is written as:

$$H(Class) = -p(P1)\, log\, p(P1) - p(P2)\, log\, p(P2)$$

$$H(Class|Feature) = p(P1, Feature) log \frac{p(Feature)}{p(P1, Feature)} - p(P2, Feature) log \frac{p(Feature)}{p(P2, Feature)}$$

p is here the probability. *Feature* corresponds here to any of the 15116 features we extracted from the text corpus. Thank to the function *InfoGainAttributeEval* available in Weka, 15116 information gains are computed and the features are ranked from the highest information gain to the lowest.

For the full cross-validation scenario, here are the 20 “best” features we obtain:

Topic related features (ngrams and LDA) are here obviously dominant. Indeed, the available data mostly deal with school, education and religion. Even so, some general features (syntactic, lexical, sentiment) are also present. Regarding lexical features, the 2

Rank	Feature	Information Gain
1	post length	0.10607
2	number of sentences	0.07802
3	ngram **in**	0.07338
4	ngram **school**	0.06966
5	ngram **public**	0.06615
6	ngram **private**	0.06578
7	LDA school private send public bid afford sacrifice wealthy rich crappy	0.06558
8	LDA education system public change improve quality educational voucher author product	0.06377
9	sentiment very positive standard deviation	0.06094
10	LDA year start back hold grade age kindergarten ready son turn	0.06089
11	max number of Sub-Clauses	0.05945
12	sentiment very negative standard deviation	0.05933
13	LDA boy girl learn sex single woman classroom young experience gender	0.0559
14	sentiment negative standard deviation	0.0554
15	cardinal ratio	0.05313
16	LDA teacher teach school union job classroom good run hand ca	0.05309
17	dependency rule S $\rightarrow NP \rightarrow VP$	0.05282
18	ngram **schools**	0.05165
19	sentiment positive standard deviation	0.05127
20	ngram **to**	0.05026

first best are related to the length of a post (global length in characters and number of sentences). For sentiments, the one that count are the standard deviations.

Now we remove all the features with an information gain equals to 0. There are 1738 remaining features, with the following distribution:

Type of Feature	Number
Baseline	1476
Part of Speech (POS)	11
Lexical	10
Syntactic	185
Sentiment	31
LDA	25
TOTAL	1738

The features that we engineered are quite pertinent since 70% of the POS, 67% of the lexical, 68% of the sentiment and 80% of the LDA features have an information gain strictly higher than 0. Even so, the baseline n-grams features still represent 85% which means that 85% of the information that tell us if a text is P1 or P2 is contained in simple topic-related features and uneasily interpretable features.

3.3 Error Analysis

In the section 2.1.3, we described how in the annotation process, some of the posts are declared as *conflicting* since the annotators didn't unanimously agree on its class. In following example, for one of the annotator, the post is persuasive, the author implicitly supports homeschooling in the sentence *many are willing to ignore several centuries of improvements in organized education in our part of the world*, but the two other annotators wrote that they were not able to conclude without any extra knowledge.

329_P2_artcomment_homeschooling.txt The comments are more fascinating than the article. Many folks, some of them with impressive intellectual accomplishments, would have us believe that the current era is different, that we, especially in the industrialized world, have moved beyond superstition into truly rational ways of thinking. The comments prove quite the opposite. Many people firmly believe things obviously at odds with reality. In particular, many are willing to ignore several centuries of improvements in organized education in our part of the world. In addition, many people refuse to accept that we can do things as a group, "society" that is, that we can't do as individuals. This does not bode well for our future, since in a very real sense we are all stuck with each other.

To find out if our classifier also "hesitates" on conflicting posts, we define 4 coefficients. By analogy with the binary classification, we call them TP, TN, FP and FN:

```
TP: Correctly classified and no conflict
TN: Misclassified and conflict
FP: Correctly classified and conflict
FN: Misclassified but no conflict
```

Again, the coefficients are named this way just by analogy with binary classification, but here they don't record the performances of a classification. Similarly, they can be represented in a confusion matrix:

	No Conflict	Conflict
Classified	TP	FP
Misclassified	FN	TN

The FN cell, highlighted in gray, is the most problematic one. It corresponds to the cases where the annotators unanimously agreed on the class but the classifier couldnt recognize it. To quantify how much the classifier can't recognize those instances we decided to compute the following coefficient:

$$\frac{TP + TN + FP}{TP + FN + FP + \mathbf{FN}}$$

We obtain the following results for the different configurations:

In-Domain	Full	RS	PIS	HS	SSE	MS	PPS
Coefficient	73.64	57.35	81.82	77.68	74.00	68.97	78.57

Cross-Domain	RS	PIS	HS	SSE	MS	PPS
Coefficient	61.76	62.24	70.54	70.00	62.07	78.57

If you want to check all the confusion matrices, please have a look at appendix E.

To conclude this error analysis, we'll pick-up randomly 5 *FN* cases out of the 261 ones of the full cross-validation and try to analyse why they were misclassified.

3073_P2_artcomment_public-private-schools.txt Parenting styles decisions, including where to send your children to school, is a very subjective matter. Saying that someone is a bad person for sending their children to private/parochial school is no different than someone telling me that I'm not a whole or complete woman because I have chosen not to have any children. Ridiculous.

It's hard to assign the label P1 to this post without deep knowledge. that said, the annotators considered it as P2. But here, the length of the sentences (quite long), the diversity of the vocabulary and a sentiment tone which stays more or less neutral, made the classifier to conclude P1 for the predicted class.

323_P1_artcomment_homeschooling.txt There are the rare exceptions where homeschooling is great - but for most it is child abuse. Regular testing should be required and if they fail it's back to public school.

Here it's totally the opposite. The author clearly shows his views about homeschooling - *There are the rare exceptions where homeschooling is great* - but the post is relatively short and there's high sentiment variation (*is great* and *child abuse*), that's the reason why the classifier classified it as non persuasive.

1617_P2_forumpost_prayer-in-schools.txt I don't believe that anyone is saying that students shouldn't be allowed to pray in school... in the form of an individual saying grace over lunch or before the start of a test... The issue is with official prayers read over the loudspeaker or otherwise directly sanctioned by school authorities.

Here, there's quite a lot of topic related vocabulary: students, pray, school, ect... and besides some argumentative forms are present, *I don't believe* and *students shouldn't.* This could explain why the classifier considered it as P1.

4891_P2_forumpost_mainstreaming.txt Hi,, For the past 2 years my dd has been at trinity independent school, I. Rochester. It's a social school for kiddies with asd, dyslexia, dyscalclia and motor probs. Well we were self funding her due to her not being able to be statmented due to lack of funding * rolls eyes * . We got into some financial probs and the school have asked us not to return her in Jan due to the amount that's owed by us. We have tried 17 schools in Medway to try and see if they will take my dd, all have refused due to her not being statemented and them not being able to meet her needs. The lea have told me to retry for a statement and to keep dd at home until it comes thru, however this takes 6 months lol!! I really don't know what to do as I'm a full time uni student and work, I can't homeschool her. Plus her sensory needs need addressing in a strict school routine and m scared for her being taken out of this. To top things off the school knew damn well we would struggle to find her an appropriate school due to her needs, but still said they would happily have her back once statemented. I'm lost,, can anyone help me with what to do?? Sam xx

The author is asking for advice and clearly don't show any aim to persuade. Nevertheless, the post is very long, so are the sentences and despite the grammar mistakes, there's a lot of topics vocabulary.

> **5145_P1_forumpost_redshirting.txt** I know several people who said that they wish their parents had waited a year to send them to kindergarten. DH was 4 when he started school and was only 17 when he graduated from HS. He hated that everyone was doing everything a year before he could do it. He was very good at sports though, even though he was a year younger than anyone else on his team. I honestly made my decision because I had a choice and I just did not feel right personally about sending my DS to Kindergarten at age 4. Last year the cut off was October 9th I believe and this year it has been changed to July 1st. DS's birthday is August 30th. I did not wait to send DS because I wanted him to have an edge over anyone. Age has nothing to do with ability and if he were the type who would struggle, he would struggle regardless of when he started school. I do not compare my children with other people's children. I do what is best for my children and assume other parents do what is best for their children. Age wise I was right in the middle of my class at school and I was far ahead of my class. I don't think age really dictates how well you will do in school or in sports.

Lot of sentiment fluctuation but above all lots of occurrences of the first person.

Here are the results of the cross-validations if we take out the conflicting instances:

In Domain CV	Full	RS	PIS	HS	SSE	MS	PPS
Conflicts removed	75.24	65.22	80.17	82.66	61.36	68.42	77.71
With conflict (reminder)	68.18	51.48	74.13	73.21	74.00	65.52	70.59

Here are shown the accuracies. There's a significant increase for all the domains expect homeschooling and single sex education that can be explained by the low quantity of data available for those two domains.

3.4 Future Work and General Conclusion

The different results we obtained lead us to think that other techniques can be applied to this classification problem. Obviously, more lexical and syntactic features should be implemented. For example, if I had more time, I could have integrated the Ruby-based Penn Discourse Tree Bank[11] parser[1] in DKPro which would have given us discourse related feature sets.

Besides, in this study, meaningful features are highly topic-related and this because the domains distribution is not equal (476 texts for public private schools debate and only 29 for mainstreaming). If we want to gain in generality regarding our features, we should extend the quantity of domains. Moreover, 990 texts might be considered as low for a classification problem, but how is it possible to extend the dataset without launching another annotation phase which is time time-consuming? We came with the idea of *bootstrapping* our training set by parsing some debate portal and consider all the posts as P1 (if the amount of data extracted is high enough, this approximation is acceptable since

[1] http://wing.comp.nus.edu.sg/ linzihen/parser

most of the posts of a debate portal should contain persuasiveness). We already have a content extractor written in Java using *jsoup*[2] that can extracts posts from the debate portal createdebate[3]. This portal has a *Creative Commons License* so we can freely use its content. As of today, bootstrapping was not launched, but it should be possible to do it soon.

3.5 Personal Conclusion

This internship in the field of NLP and text mining gave me an opportunity to understand the way research works. Working in this environment taught me how to organize myself (in terms of programming, research, ect...) and how to collaborate with others.

[2] jsoup is a Java library for working with real-world HTML, http://jsoup.org/

[3] http://www.createdebate.com/

APPENDIX A

Work Packages

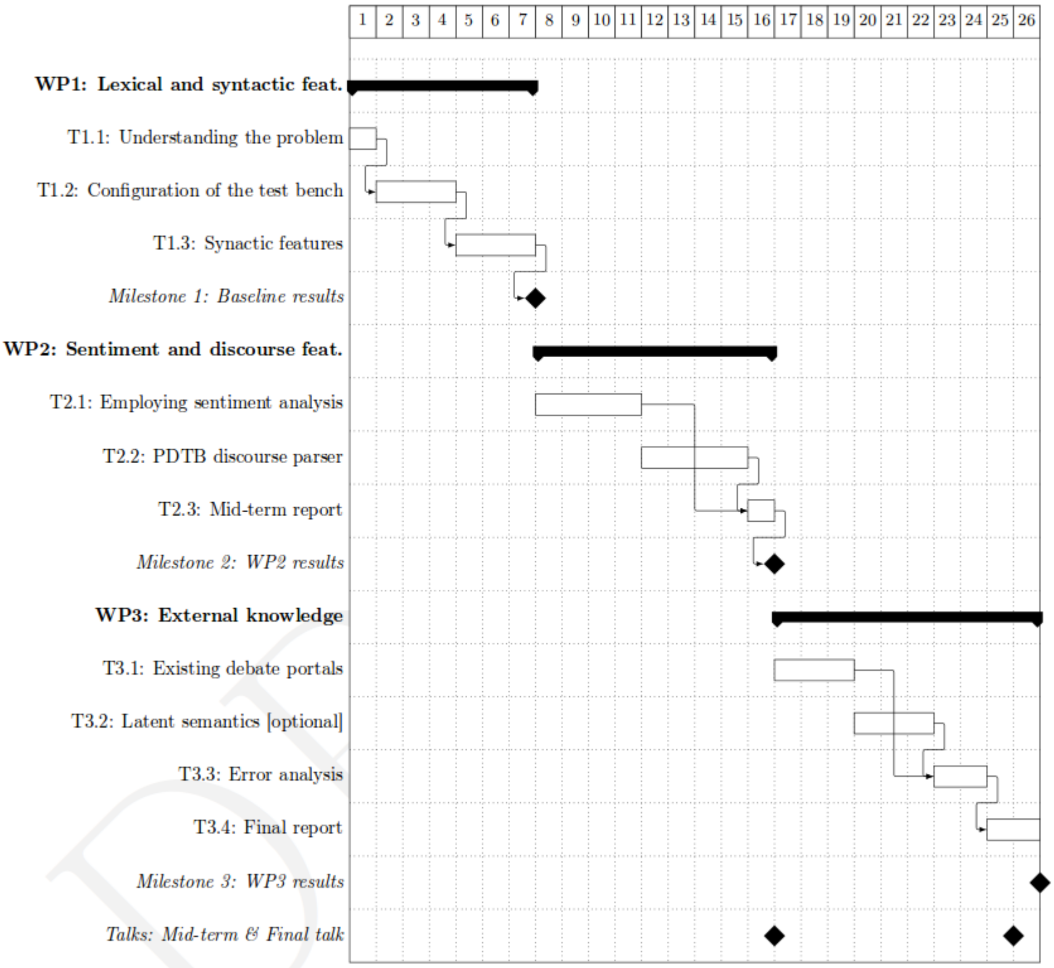


Figure 1: Project schedule

Figure A.1: Project schedule

APPENDIX B

DKPro Core Overview

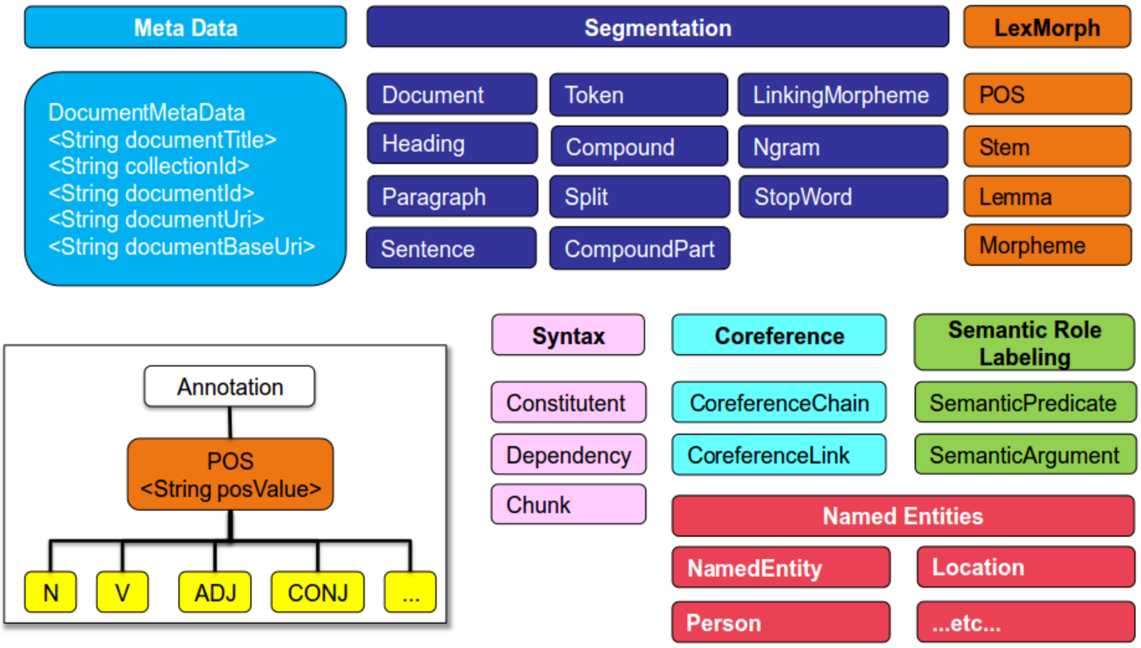
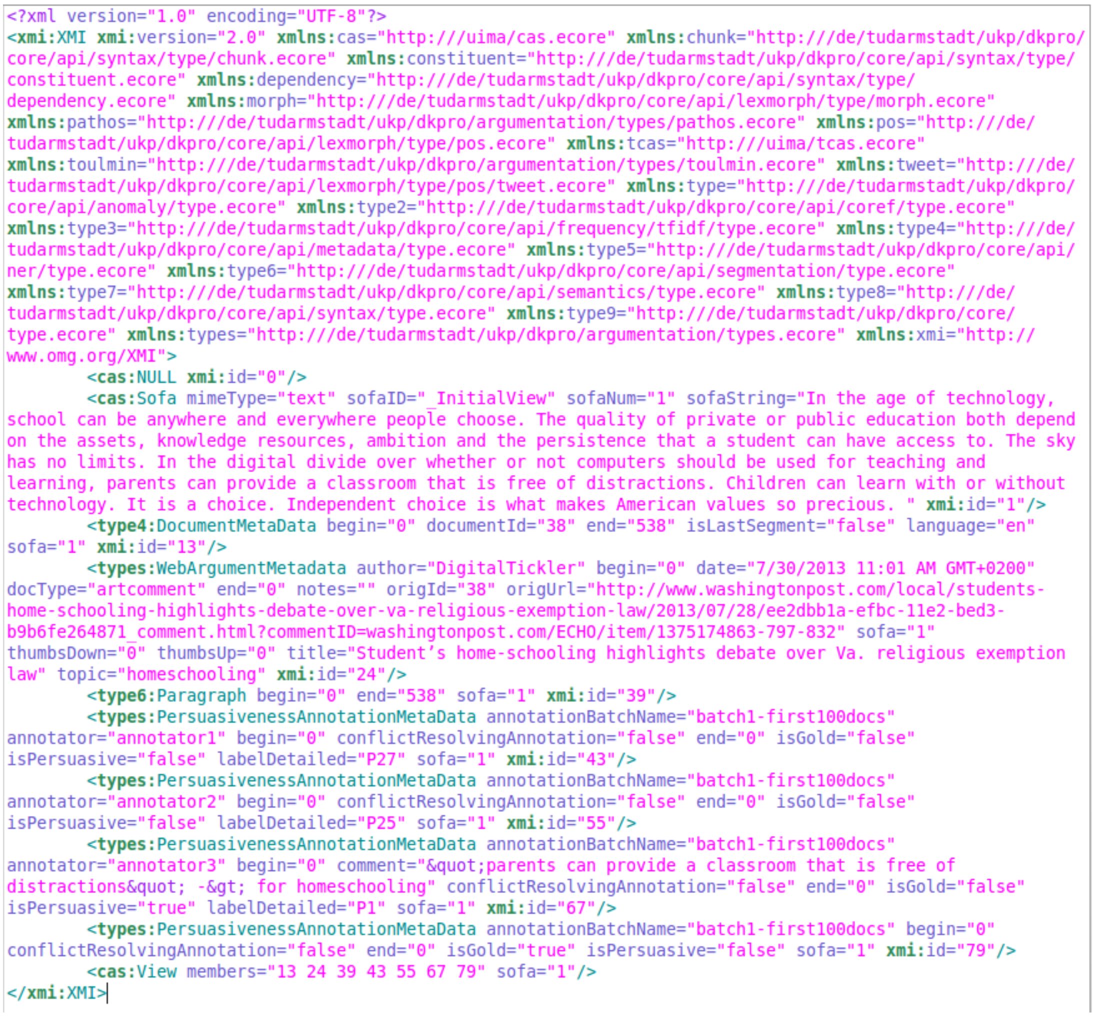


Figure B.1: DKPro Type System

APPENDIX C

Xmi Files - Example



```
<?xml version="1.0" encoding="UTF-8"?>
<xmi:XMI xmi:version="2.0" xmlns:cas="http:///uima/cas.ecore" xmlns:chunk="http:///de/tudarmstadt/ukp/dkpro/
core/api/syntax/type/chunk.ecore" xmlns:constituent="http:///de/tudarmstadt/ukp/dkpro/core/api/syntax/type/
constituent.ecore" xmlns:dependency="http:///de/tudarmstadt/ukp/dkpro/core/api/syntax/type/
dependency.ecore" xmlns:morph="http:///de/tudarmstadt/ukp/dkpro/core/api/lexmorph/type/morph.ecore"
xmlns:pathos="http:///de/tudarmstadt/ukp/dkpro/argumentation/types/pathos.ecore" xmlns:pos="http:///de/
tudarmstadt/ukp/dkpro/core/api/lexmorph/type/pos.ecore" xmlns:tcas="http:///uima/tcas.ecore"
xmlns:toulmin="http:///de/tudarmstadt/ukp/dkpro/argumentation/types/toulmin.ecore" xmlns:tweet="http:///de/
tudarmstadt/ukp/dkpro/core/api/lexmorph/type/pos/tweet.ecore" xmlns:type="http:///de/tudarmstadt/ukp/dkpro/
core/api/anomaly/type.ecore" xmlns:type2="http:///de/tudarmstadt/ukp/dkpro/core/api/coref/type.ecore"
xmlns:type3="http:///de/tudarmstadt/ukp/dkpro/core/api/frequency/tfidf/type.ecore" xmlns:type4="http:///de/
tudarmstadt/ukp/dkpro/core/api/metadata/type.ecore" xmlns:type5="http:///de/tudarmstadt/ukp/dkpro/core/api/
ner/type.ecore" xmlns:type6="http:///de/tudarmstadt/ukp/dkpro/core/api/segmentation/type.ecore"
xmlns:type7="http:///de/tudarmstadt/ukp/dkpro/core/api/semantics/type.ecore" xmlns:type8="http:///de/
tudarmstadt/ukp/dkpro/core/api/syntax/type.ecore" xmlns:type9="http:///de/tudarmstadt/ukp/dkpro/core/
type.ecore" xmlns:types="http:///de/tudarmstadt/ukp/dkpro/argumentation/types.ecore" xmlns:xmi="http://
www.omg.org/XMI">
        <cas:NULL xmi:id="0"/>
        <cas:Sofa mimeType="text" sofaID="_InitialView" sofaNum="1" sofaString="In the age of technology,
school can be anywhere and everywhere people choose. The quality of private or public education both depend
on the assets, knowledge resources, ambition and the persistence that a student can have access to. The sky
has no limits. In the digital divide over whether or not computers should be used for teaching and
learning, parents can provide a classroom that is free of distractions. Children can learn with or without
technology. It is a choice. Independent choice is what makes American values so precious. " xmi:id="1"/>
        <type4:DocumentMetaData begin="0" documentId="38" end="538" isLastSegment="false" language="en"
sofa="1" xmi:id="13"/>
        <types:WebArgumentMetadata author="DigitalTickler" begin="0" date="7/30/2013 11:01 AM GMT+0200"
docType="artcomment" end="0" notes="" origId="38" origUrl="http://www.washingtonpost.com/local/students-
home-schooling-highlights-debate-over-va-religious-exemption-law/2013/07/28/ee2dbb1a-efbc-11e2-bed3-
b9b6fe264871_comment.html?commentID=washingtonpost.com/ECHO/item/1375174863-797-832" sofa="1"
thumbsDown="0" thumbsUp="0" title="Student’s home-schooling highlights debate over Va. religious exemption
law" topic="homeschooling" xmi:id="24"/>
        <type6:Paragraph begin="0" end="538" sofa="1" xmi:id="39"/>
        <types:PersuasivenessAnnotationMetaData annotationBatchName="batch1-first100docs"
annotator="annotator1" begin="0" conflictResolvingAnnotation="false" end="0" isGold="false"
isPersuasive="false" labelDetailed="P27" sofa="1" xmi:id="43"/>
        <types:PersuasivenessAnnotationMetaData annotationBatchName="batch1-first100docs"
annotator="annotator2" begin="0" conflictResolvingAnnotation="false" end="0" isGold="false"
isPersuasive="false" labelDetailed="P25" sofa="1" xmi:id="55"/>
        <types:PersuasivenessAnnotationMetaData annotationBatchName="batch1-first100docs"
annotator="annotator3" begin="0" comment="&quot;parents can provide a classroom that is free of
distractions&quot; -&gt; for homeschooling" conflictResolvingAnnotation="false" end="0" isGold="false"
isPersuasive="true" labelDetailed="P1" sofa="1" xmi:id="67"/>
        <types:PersuasivenessAnnotationMetaData annotationBatchName="batch1-first100docs" begin="0"
conflictResolvingAnnotation="false" end="0" isGold="true" isPersuasive="false" sofa="1" xmi:id="79"/>
        <cas:View members="13 24 39 43 55 67 79" sofa="1"/>
</xmi:XMI>
```

Figure C.1: Example of XMI file

APPENDIX D

Binary Classification

Cross-Validation and Train-Test Scenarios

First, we should keep in mind that those two scenarios can also be evaluated for other problems than binary classification. For the train-test scenario, we simply have two sets, the *training set* that will be used to train our model and the test set for making the predictions. In the n folds cross-validation scenario we split our data in n folds, one of the fold will be the test set and the n-1 others will form the training set. The predictions are evaluated n times.

Basic definitions

Suppose we have to classes (to keep the report example, P1 and P2), one called *positive* (here P1) and the other *negative*[1]. Then we run our algorithm (with a cross-validation or a train-test scenario) and we can report four cases:

- *True Positive (TP)*: A P1 instance was recognized as P1.
- *True Negative (TN)*: A P2 instance was recognized as P2.
- *False Positive (FP)*: A P2 instance was recognized as P1.
- *False Negative (FN)*: A P1 instance was recognized as P2.

Confusion Matrix

It's a 2 by 2 matrix that summarizes the prediction results as follow:

[1] This designation come from the fact that binary classification is usually used in epidemiology (The patient is sick or not)

		Prediction outcome		
		P1	**P2**	**total**
actual value	**P1**	True Positive	False Negative	P′
	P2	False Positive	True Negative	N′
	total	P	N	

Statistical measures calculated from the Confusion Matrix

Sensitivity: Sensitivity (also called the true positive rate, or the recall rate in some fields) measures the proportion of actual positives which are correctly identified as such (here the percentage of P1 instances which are correctly identified as persuasive). The formula is:

$$Sensitivity = \frac{TP}{TP + FN}$$

Specificity: Specificity (sometimes called the true negative rate) measures the proportion of negatives which are correctly identified as such (here the percentage of P2 instances which are correctly identified as non persuasive). The formula is:

$$Specificity = \frac{TN}{TN + FP}$$

Accuracy: In binary classification, the accuracy is the proportion of true results (both true positives and true negatives). The formula is:

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP}$$

Macro F-measure: In case of high biased in our data (for example 99% of negative instances and 1% of positive instances), the accuracy is not a relevant metric anymore. Indeed a classifier that foolishly predict all the instances as negative will have, an accuracy of 99%. Thus, we introduce the macro F-measure which can be seen as the harmonic mean of precision and sensitivity. The formula is:

$$Fm = \frac{2TP}{2TP + FN + FP}$$

APPENDIX E

Error Analysis - Confusion Matrices

Glossary

argumentation Reason giving in communicative situations by people whose purpose is the justification of acts, beliefs, attitudes, and values. 2

gold data Bla bla. 14

irony The use of words to express something different from and often opposite to their literal meaning. 4

lemma In morphology and lexicography, a lemma (plural lemmas or lemmata) is the canonical form, dictionary form, or citation form of a set of words (headword). In English, for example, run, runs, ran and running are forms of the same lexeme, with run as the lemma. 16

machine learning Machine learning is a subfield of computer science and artificial intelligence that deals with the construction and study of systems that can learn from data, rather than follow only explicitly programmed instructions. 4

parts-of-speech A term in traditional grammar for the eight categories into which words are classified according to their functions in sentences: noun, pronoun, verb, adjective, adverb, preposition, conjunction, interjection. 6

persuasion Communication intended to influence the acts, beliefs, attitudes, and values of others. 2

sarcasm A cutting, often ironic remark intended to wound. 4

single-label classification Each instance is assigned to one and only one class. 7

stemma ?. 16

supervised learning Supervised learning is the machine learning task of inferring a function from labelled training data. 7

token Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. 16

tokenization Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens.. 6

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] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[5] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] ZIHENG LIN, HWEE TOU NG, and MIN-YEN KAN. A pdtb-styled end-to-end discourse parser. *Natural Language Engineering*, 20:151–184, 4 2014.

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

[13] John C. Platt. Advances in kernel methods. chapter Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pages 185–208. MIT Press, Cambridge, MA, USA, 1999.

[14] 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.

[15] 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.

[16] 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.