LEARNING FOR TEXT MINING: TACKLING THE COST OF FEATURE AND KNOWLEDGE ENGINEERING

JOSÉ IRIA

February 2012

PhD in Computer Science

Department of Computer Science
The University of Sheffield
Sheffield, UK

Supervisor: Prof. Fabio Ciravegna

BLANK PAGE
IN
ORIGINAL

Dedicated to the loving memory of my grandparents

Adelina and Victor

BLANK PAGE
IN
ORIGINAL

ABSTRACT

Over the last decade, the state-of-the-art in text mining has moved towards the adoption of machine learning as the main paradigm at the heart of approaches. Despite significant advances, machine learning-based text mining solutions remain costly to design, develop and maintain for real world problems. An important component of such cost (feature engineering) concerns the effort required to understand which features or characteristics of the data can be successfully exploited in inducing a predictive model of the data. Another important component of the cost (knowledge engineering) has to do with the effort in creating labelled data, and in eliciting knowledge about the mining systems and the data itself.

I present a series of approaches, methods and findings aimed at reducing the cost of creating and maintaining document classification and information extraction systems. They address the following questions: Which classes of features lead to an improved classification accuracy in the document classification and entity extraction tasks? How to reduce the amount of labelled examples needed to train machine learning-based document classification and information extraction systems, so as to relieve domain experts from this costly task? How to effectively represent knowledge about these systems and the data that they manipulate, in order to make systems interoperable and results replicable?

I provide the reader with the background information necessary to understand the above questions and the contributions to the state-of-the-art contained herein. The contributions include: the identification of novel classes of features for the document classification task which exploit the multimedia nature of documents and lead to improved classification accuracy; a novel approach to domain adaptation for text categorization which outperforms standard supervised and semi-supervised methods while requiring considerably less supervision; and a well-founded formalism for declaratively specifying text and multimedia mining systems.

BLANK PAGE
IN
ORIGINAL

ACKNOWLEDGMENTS

First I would like to thank my thesis supervisor, Fabio Ciravegna, who introduced me to the captivating worlds of Text Mining and Knowledge Management. He not only gave me valuable advice, but also, and perhaps more importantly, taught me by example a number of skills that will accompany me for the rest of my life. Christopher Brewster provided me with countless hours of interesting discussions and reflection, which ultimately led to my personal growth. João Magalhães and Cristina Kadar were great collaborators, with whom I had deep technical discussions. I enjoyed working with them, and greatly benefited from our interactions.

I spent seven years at The University of Sheffield and two years at IBM Research - Zurich. I would like to thank the following collaborators, faculties, staffs, fellow students and friends, who made that period of my life a memorable experience: Rodrigo Carvalho, Ziqi Zhang, Ajay Chakravarthy, Lei Xia, Anna Lisa Gentile, Aba-Sah Dadzie, Matthew Rowe, Neil Ireson, Elizabeth Cano, Mark Greenwood, Laura Smith, Ravish Bhagdev, Victoria Uren, Mario Lučić, Vit Prajzler, Daniel Svonava, Michal Lohnicky, Hamish Cunningham, Dorothea Wiesmann, Michail Vlachos, Niraj Aswani, Danica Damljanovic, Rob Gaizauskas, Abdel Labbi, Dick Husemann, David Guthrie, Ben Allison, Sanaz Jabbari, David Martinez, Sinziana Mazilu, Suvodeep Mazumdar, Lucy Moffatt, Felix Müller, Horacio Saggion, Jose San Pedro, Mark Sanderson, Elisa Sottile, Lucia Specia, Mark Stevenson, Gill Wells, Yorick Wilks, Elena Montañés, Beat Gfeller, Alexiei Dingli, Angelo Dalli, Phil Moore, Wim Peters, Cristian Ursu, and Valentin Tablan. During this period, I was also fortunate to participate in several extremely interesting research projects and activities. I would like to thank the following fellow researchers for their valuable discussions and insights: Steffen Staab, Philipp Cimiano, Spiros Nikolopoulos, Oscar Corcho, Javier Artiles, Sebastian Blohm, John Domingue, Vanessa Lopez, Steve Fullerton, Luca Gilardoni, Siegfried Handschuh, Pierre Grenon, Alberto Lavelli, Lorenza Romano, Agnieszka Lawrynowicz, Abraham Bernstein, Andriy Nikolov, Enrico Motta, Marta Sabou, Luis Sarmento, Hugo Zaragoza and Claudio Giuliano.

Finally I wish to thank my family: my parents José António and Maria Augusta, who always encouraged me to go one step further and supported me in doing so; and my dear fiancée Nathalie, who fills my life with so much love, happiness and well-being. Without them, this thesis would not have come to fruition.

CONTENTS

I THE DOMAIN OF TEXT MINING 1

1 INTRODUCTION 3

1.1 Research Questions 9

1.2 Contributions 10

1.3 Impact 12

1.4 Summary 14

2 FROM UNSTRUCTURED TO STRUCTURED DATA 17

2.1 Anatomy of an Unstructured Information Mining System 18

2.2 Text Mining from a Machine Learning Viewpoint 21

2.2.1 Text Categorization as a Classification Task 21

2.2.2 Entity Extraction as a Classification Task 22

2.3 Evaluating Text Mining 25

2.4 Models for Text Classification 28

2.4.1 Generalised Linear Models 28

2.4.2 Margin Classifiers 30

2.4.3 Feature Selection 33

2.5 Models for Entity Extraction 35

2.5.1 Rule Induction Approaches 35

2.5.2 Single-Object Labelling Approaches 39

2.5.3 Sequential Labeling Approaches 41

2.6 Supervision Requirements 44

2.6.1 Semi-supervised Learning 44

2.6.2 Transductive Learning 45

2.6.3 Feature Labels versus Instance Labels 46

2.7 Ontology and Semantic Annotation 47

2.7.1 Ontology Languages 48

2.7.2 Foundational Ontologies 50

2.7.3 Core Ontologies 53

3 OPEN PROBLEMS IN TEXT MINING 59

3.1 Problem 1: The Cost of Engineering Features 59

3.2 Problem 2: The Cost of Obtaining Labelled Data 63

3.3 Further Problems 65

3.3.1 Problem 3: The Systemic Problem 66

3.3.2 Problem 4: The Communication Problem 67

3.3.3 Problem 5: The Replicability Problem 68

II TACKLING THE COST OF FEATURE AND KNOWLEDGE ENGINEERING 69

4 FEATURE ENGINEERING FOR TEXT MINING 71

4.1 Successful Features for Entity Extraction 71

4.1.1 Datasets 72

4.1.2 A Detailed Study 73

4.1.3 Results 77

4.1.4 Discussion 82

4.1.5 Analysis of the Results 82

4.1.6 Analysis of the Errors 84

4.1.7 Validation 86

4.2 Exploiting Cross-Media Correlations 87

4.2.1 Image Analysis 88

4.2.2 Proposed Approach 90

4.2.3 Multimedia Document Representation 91

4.2.4 Multimedia Document Processing 91

4.2.5 Multimedia Document Categorization 95

4.2.6 Experiments 96

4.2.7 Results and Discussion 97

4.3 Related Work 99

4.4 Conclusions 103

5 OVERCOMING THE SCARCENESS OF LABELLED DATA 107

5.1 Semi-supervised Entity Extraction on a Real-World Use-Case 108

5.1.1 The Use Case 109

5.1.2 Learning Algorithms 110

5.1.3 Dataset 113

5.1.4 Labelling Data 113

5.1.5 Dataset Generation 115

5.1.6 Learning Approach 116

5.1.7 Experimental Setup 116

5.1.8 Results 117

5.1.9 Discussion 118

5.2 Adapting to Different Domains through Feature Labeling 120

5.2.1 Proposed Approach 121

5.2.2 Datasets 123

5.2.3 Evaluation Methodology 124

5.2.4 Labelling Data 124
5.2.5 Optimiser and LDA implementations 126
5.2.6 Results and Discussion 126
5.3 Related Work 129
5.4 Conclusions 132
6 A FORMALISATION OF MULTIMEDIA MINING 135
6.1 Requirements 136
6.2 Proposed Ontology 137
6.2.1 Information Extraction Tasks and Systems 138
6.2.2 Information Extraction Subsystems 139
6.2.3 Text Decomposition 140
6.2.4 Text Tagging 142
6.2.5 Models for Classification and Extraction 142
6.2.6 Semantic Annotation 144
6.3 Discussion 144
6.4 Related Work 148
6.5 Conclusions 151
7 CONCLUSIONS 153

III APPENDICES 157
A OPEN-SOURCE SOFTWARE AUTHORED 159
A.1 The Runes Framework 159
A.2 The Aleph Library 159
A.3 The T-Rex Systems 160
B PAIRING START AND END OF BOUNDARY PREDICTIONS 161

BIBLIOGRAPHY 163

LIST OF FIGURES

Figure 1 Relative sizes of the Library of Congress, the surface and deep Web, email and data on hard disks. 4

Figure 2 The basic anatomy of an unstructured information mining system. 20

Figure 3 An excerpt of a seminar announcement. Which rules would extract the speaker "Ralph Hill" in the above text, but also other speakers in other texts like this one? 23

Figure 4 Spectrum of leniency in what is considered an entity mention match in Entity Extraction. 28

Figure 5 An illustration of the basic principle behind Support Vector Machines 31

Figure 6 Search space produced by rule specialisation. 37

Figure 7 Rule specialisation illustrated. 38

Figure 8 The relationship between Naïve Bayes, Hidden Markov Model, Maximum Entropy Markov Models and Conditional Random Fields. 44

Figure 9 The DOLCE top-level ontology. 50

Figure 10 The Descriptions & Situations ontology design pattern. 51

Figure 11 The Ontology of Information Objects pattern. 52

Figure 12 The Ontology of Plans pattern. 53

Figure 13 The main concepts in the Core Software Ontology. 55

Figure 14 The Decomposition pattern in the COMM ontology. 56

Figure 15 The Content & Media Annotation pattern in the COMM ontology. 57

Figure 16 The Semantic Annotation pattern in the COMM ontology. 57

Figure 17 Example of a Web news article. 62

Figure 18 Service discovery in service-oriented architectures. 67

Figure 19 An excerpt of a workshop call for papers. 74

Figure 20 The effect of token window length on the Seminar Announcements corpus. 78

Figure 21 The effect of token window length on the Workshop Call for Papers corpus. 81

Figure 22 The effect of feature selection on the Seminar Announcements corpus. 81

Figure 23 The effect of feature selection on the Workshop Call for Papers corpus. 82

Figure 24 Retrieval results for the individual media, multimedia and cross-media configurations. 97

Figure 25 Precision-recall graphs for the individual media, multimedia and cross-media configurations. 100

Figure 26 Learning curves for the first dataset generated from 20-newsgroups. From left to right descending: *Cars vs Games, Cars vs Hardware, Cars vs Games vs Hardware vs OS,* and *Cars vs Games vs Hardware vs OS vs Politics vs Religion.* 129

Figure 27 Learning curves for the second dataset generated from 20-newsgroups. From left to right descending: *Comp vs Sci, Rec vs Talk, Comp vs Rec,* and *Comp vs Talk.* 130

Figure 28 Learning curves for the dataset generated from SRAA. From left to right: *Auto vs Aviation* and *Real vs Simulated.* 130

Figure 29 Positioning the Ontology of Information Extraction (ONIX) in the framework of existing ontologies. 137

Figure 30 Ontology pattern in ONIX for describing core top-level Information Extraction concepts. 138

Figure 31 Ontology pattern in ONIX for describing Information Extraction subsystems. 139

Figure 32 Ontology pattern in ONIX for characterising text segmentation. 141

Figure 33 Specialising COMM locators in ONIX for text data. 141

Figure 34 Ontology pattern in ONIX for characterising building and applying data models. 143

Figure 35 Sample semantic description of an IE system using the Ontology of Information Extraction. 146

LIST OF TABLES

Table 1 Contingency table. 25

Table 2 Details on the entity classes in the Seminar Announcements corpus. 73

Table 3 Details on the entity classes in the Workshop Call for Papers corpus. 74

Table 4 The effect of different classes of basic features on the Seminar Announcements dataset. 78

Table 5 The effect of different classes of basic features on the Workshop Call for Papers dataset. 79

Table 6 The effect of adding classes of advanced features on the Seminar Announcements dataset. 79

Table 7 The effect of adding classes of advanced features on the Workshop Call for Papers dataset. 80

Table 8 The impact of space and newline tokens on the Seminar Announcements dataset. 80

Table 9 The impact of space and newline tokens on the Workshop Call for Papers dataset. 80

Table 10 Comparing the T-Rex Entity Extraction system with other state-of-the-art systems on the Seminar Announcements dataset. 86

Table 11 Comparing the T-Rex Entity Extraction system with other state-of-the-art systems on the Workshop Call for Papers dataset. 87

Table 12 Detailed retrieval results for the cross-media configuration: mean precision at 10, mean precision at 30 and mean average precision. Note that FP < FN since low confidence predictions are discarded, in other words, the classifier may not output a prediction for every document. 98

Table 13 Summarized retrieval results for the individual media, multimedia and cross-media configurations: mean precision at 10, mean precision at 30 and mean average precision. 99

Table 14 The list of ontology concepts used in the evaluation. 114

Table 15 Detailed results per class for the supervised configuration. 117

Table 16 Detailed results per class for the graph-based semi-supervised configuration. 118

Table 17 Detailed results per class for the large-scale configuration. 119

Table 18 Characteristics of the datasets used for evaluating the proposed approach. 125

Table 19 Initial labelled features and discovered zLDA features for *Cars vs Hardware*. 126

Table 20 Classification accuracies and the amount of labelled information (either instances or features) used in different sets of experiments. Note that for the *TransferzLDALF* method, the reported results correspond to selecting a fixed number of 18 features per topic (cf. learning curves), but the features outputted by zLDA can overlap and thus the size of the feature set used is smaller when merged. 127

Table 21 Performance comparison with [35]. 128

Table 22 Performance comparison with [169]. 128

ACRONYMS

ONIX	Ontology of Information Extraction
SOA	Service-Oriented Architecture
NLP	Natural Language Processing
IE	Information Extraction
IR	Information Retrieval
DC	Document Classification
TC	Text Classification
RE	Relation Extraction
EE	Entity Extraction
NER	Named Entity Recognition
ML	Machine Learning
SVM	Support Vector Machine
XML	Extensible Markup Language
UML	Unified Modeling Language
ODF	Open Document Format
HTML	Hypertext Mark-up Language
PDF	Portable Document Format
DOC	Microsoft's Word document
OWL	Web Ontology Language
RDF	Resource Description Framework
RDFS	RDF Schema
DOLCE	Descriptive Ontology for Linguistic and Cognitive Engineering
CSO	Core Software Ontology
COMM	Core Ontology of MultiMedia

W3C	World Wide Web Consortium
URI	Universal Resource Identifier
URL	Uniform Resource Locator
DnS	Descriptions & Situations
OIO	Ontology of Information Objects
OoP	Ontology of Plans
API	Application Programming Interface
MPEG	Moving Picture Experts Group
KB	Knowledge Base
IA	Image Analysis
UIMA	Unstructured Information Management Architecture
GATE	General Architecture for Text Engineering
OO	Object Oriented
IG	Information Gain
CE	Cross Entropy
SA	Seminar Announcements
WCFP	Workshop Call for Papers
HMM	Hidden Markov Model
EM	Expectation Maximization
MEMM	Maximum Entropy Markov Models
CRF	Conditional Random Fields
NB	Naïve Bayes
RRM	Regularised Risk Minimization
GLM	Generalised Linear Model
RSS	Really Simple Syndication
LSI	Latent Semantic Indexing
PLSA	Probabilistic Latent Semantic Analysis
GE	Generalised Expectation
LDA	Latent Dirichlet Allocation
WWW	World Wide Web

BLANK PAGE
IN
ORIGINAL

Part I

THE DOMAIN OF TEXT MINING

1

INTRODUCTION

Stepping into a local library or bookstore for the first time, it is hard not to feel at the same time empowered and overwhelmed by the sheer amount of knowledge stored there, knowledge that would take a single individual the span of several lifetimes to acquire, if that was possible or desirable. That kind of feeling is magnified when typing a query into a Web search engine for the first time, and seeing anywhere between thousands and hundreds of millions of natural language documents, images and video become instantly accessible. Figure 1 shows the relative sizes of the Library of Congress and the Web in 2009.

With the barriers, physical or otherwise, of access to information gone, the scale of the individual is replaced by the scale of the collective. After a mere two decades of existence, the Web now empowers communities, social cliques, companies, governments, and many other kinds of collective entities, which search for, consume, combine and release information in digital form on a daily basis. Information that, even if in digital form, is a direct product of human communication and thus authored by humans for humans to understand, not for machines to process. This kind of information is called *unstructured* because its intended meaning is only loosely implied by its form. Unstructured information lacks explicit semantics (structure), which, given the state-of-the-art, is required for machines to interpret it as intended by the human author or needed by the end-user application.

Unstructured information

Unstructured information is by far the largest, most current and fastest growing source of knowledge available [85]. In 2008, more than 30 million websites were created, adding to the already existing 200 million websites (aprox.). However, the Web is just the tip of the iceberg. It is estimated that around 80% of the unstructured information generated is stored out of Web reach. Corporate, scientific, social and technical documentation including best practices, manuals, research reports, medical abstracts, problem reports, customer communications, contracts and emails abound within organization's Intranets. These text and multimedia artefacts contain pieces of knowledge that may be

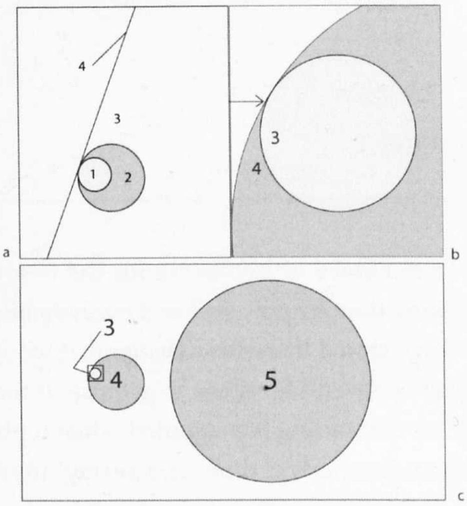


Figure 1: A diagram representing the relative sizes, in 2009, of 1) the Library of Congress, 2) the surface Web, 3) the surface + deep Web, 4) the surface + deep Web + e-mail traffic and 5) the size of text data on hard disks sold in that year. Source: [122].

critical to solve problems, analyse trends, identify opportunities and take decisions.

Because machines cannot, as yet, reliably interpret unstructured information, the admirable networking and indexing infrastructures currently in place on the Internet and in organization's Intranets to facilitate access to such large volumes of information is missing one crucial piece. In fact, only relatively simple and popular information needs can be satisfactorily fulfilled by current technology, namely those that can be successfully expressed through a collection of keywords and/or for which popular results exist. For more complex or atypical information needs, the retrieval experience often resembles that of finding a needle in a haystack: determining the right way of expressing the information need via often limited query mechanisms becomes a methodical and repetitive trial and error task; the analysis of the returned results becomes a time-consuming sifting through an overwhelming amount of irrelevant documents; and, afterwards, once potentially suitable information is discovered, it needs to be checked, understood, manually extracted from inside documents and collated with other information coming from different sources.

Clearly, the usefulness of unstructured information would increase significantly if it could be used to reliably answer queries about entities

(e. g., people, organizations) relevant to the problem at hand, and their relationships (e. g., owner, director, employee). In other words, querying unstructured information repositories should be made as reliable as querying structured ones.

Structured information

The canonical example of structured information is a relational database table. Structured information may be defined as information whose intended meaning is explicitly represented in the structure or format of the data, and is therefore unambiguous. For instance, the information stored in classic relational databases has the intended interpretation for every field data explicitly encoded in the database via column headings. Another example is the information stored in an Extensible Markup Language (XML) document, where some of the data is wrapped by tags which provide explicit semantics about how those data should be interpreted. Unstructured information, in contrast, is structure-free and therefore requires an external interpretation act in order to approximate and extract its original intended meaning or semantics.

From unstructured to structured

How can unstructured natural language and multimedia information be handled as if it was structured, and according to the end user or application needs? A natural solution is to endow unstructured information with some form of "added" structure, which explicitly provides the semantics required to interpret it correctly. An example of assigning semantics would be labelling regions of text in a text document with appropriate XML tags that, for example, might identify the names of organizations or products. Another example would be extracting elements of a document and inserting them in the appropriate fields of a relational database, or using them to create instances of concepts in a knowledge base. But who is able or willing to do it? What kind of technology can be used as support? How much would it cost? At what scale can it be done? These are some of the questions that arise when starting to think about the problem.

The need for automation

In specialised domains, such as engineering, legal or medical domains, experts are needed to interpret the information. However, experts are rarely available or willing to spend their time doing so. Therefore, even though the amount of information can typically be kept fairly manageable, the cost of adding semantics per unit of information is very high in specialised domains. On non-specialised domains, such as (most of) the Web, the converse is true: anyone can interpret the information, but, expectedly, the amount of information generated per second on the Web immediately outpaces any attempt to perform the task manually. In fact, even if/when adding semantics to newly created content was/is second nature to webpage authors, the *caveat* is that semantics is "in the eye of the beholder", that is, different end users or applications will want to interpret the information according to their own needs, which cannot be specified *a priori*. Thus, be it due to high cost or unmanageable scale, adding semantics to unstructured information is not something easily achieved with humans.

In the past three decades, there has been a considerable amount of research on automating the processes that deal with "structuring unstructured information". Within the domain of Text Mining, in particular, Document Classification and Information Extraction are two very active research fields which are addressing the problem.

Document Classification

Document Classification (DC) [150], also known as document categorization or topic spotting, is the task of labelling documents with thematic categories, or classes, from a pre-defined set. When restricting it to natural language text, the task is better known as text categorization. Document classification provides structure to a document collection by attaching a small amount of semantics to each document (a topic label). It has been applied in many contexts, ranging from document indexing based on a controlled vocabulary, to document filtering, automated metadata generation, word sense disambiguation, population of hierarchical catalogues of Web resources, and in general any application requiring document organization or selective and adaptive document dispatching.

Information Extraction (IE) [6] is the task of identifying mentions of entities and their relationships in documents. It identifies, classifies, and structures into entity and relationship classes, segments of interest in unstructured data sources, such as natural language text. Information Extraction subsumes two main subtasks: Entity Extraction (EE) and Relation Extraction (RE). Both are usually customized and targeted at a particular application domain.

Entity Extraction

Entity Extraction addresses the problem of locating mentions of predefined types of entities, where the entity classes can be very diverse, ranging from people and companies in business applications to cells and proteins in biomedical applications. An important subclass of the problem is to identify *named entities*, in which case the technique is called Named Entity Recognition (NER). For example, when given the sentence "The Times graphically describes Queen Victoria's last visit to

Sheffield in May 1897", a name entity recognizer specialized in people, organizations, locations and dates should identify the named entities "The Times" of type `Organization`, "Queen Victoria" of type `Person`, "Sheffield" of type `Location` and "May 1897" of type `Date`.

Relation Extraction

Relation Extraction works on the output of the former, that is, assuming that the relevant entities have been correctly identified, the task of RE is to find pre-defined relationships between them. Again, the set of relevant relationships to consider depends on the type of narrative, ranging from corporate acquisitions mentioned in newspaper corpora to protein interactions described in biomedical literature. For example, given the sentence mentioned earlier, a relation extraction system that is designed to identify relationships between people and locations should extract a `VisitedCity` relation between "Queen Victoria" and "Sheffield".

Knowledge Engineering vs. Machine Learning approaches

Prior to the last decade, the most actively researched approaches to document classification and information extraction can be classified as Knowledge Engineering [156] approaches. In essence, this type of approach consists in manually defining a set of rules encoding expert knowledge on how to classify documents or extract entities and relations from them. In the last decade, research on this family of approaches has increasingly declined in favour of the Machine Learning (ML) paradigm [121]. The ML paradigm has at its heart a general inductive process that statistically determines the relevant characteristics of the classes of interest and builds an automatic classifier, in this case of documents, entities or relationships, from that. Machine learning-based approaches generally achieve an accuracy comparable to that achieved by human experts, but offer considerable savings in expert labour, since no intervention from either knowledge engineers or domain experts is needed for the construction of the classifier or for its porting to a different set of classes. It is the machine learning approach to document classification and information extraction that this thesis concentrates on.

Orthogonally to the task being performed, machine learning approaches can be divided into three categories, according to the assumptions they make about the availability of labelled examples and unlabelled examples.

Unsupervised approaches to document classification, also known as document clustering [140], require no labelled examples and instead rely on the hypothesis that documents having similar contents are rele-

Unsupervised approaches

vant to the same query. The similarity between documents is usually measured with associative coefficients from the vector space model, e. g., the cosine coefficient. Unsupervised approaches to information extraction [41, 31, 79] typically exploit the redundancy of information in large scale repositories such as the Web in order to extract frequent facts, starting by applying a few simple pre-defined patterns, and later inducing more complex patterns from the way these facts appear expressed in a large number of documents. Thus, unsupervised approaches make certain assumptions about the structure of the data (e. g., redundancy in the way facts are expressed) and are designed to exploit them in some way (e. g., clustering of frequent patterns).

Supervised approaches

When labelled examples are available, either by consultation with domain experts or by relying on a safe heuristic to derive labels for some of the examples (as done in chapter 4), supervised learning approaches can be adopted, typically leading to improved accuracy with respect to unsupervised ones. A number of learning algorithms have been used to train classifiers for document classification, including the popular Naïve Bayes [115] and Support Vector Machine (SVM) [44] algorithms. The latter have also been successfully applied in named entity recognition [110] and relation extraction [172, 45], rivalling the performance obtained by maximum entropy models [23] and conditional random fields [102], two other state-of-the-art algorithms applied to these tasks. Thus, supervised approaches are able to exploit the information provided by the labels to guide the inductive process.

Semi-supervised approaches

In many domains, the cost of obtaining labelled examples far surpasses the cost of obtaining unlabelled examples. In fact, unlabelled examples can often be obtained for free. This is the case with (generic) text and images, which can be obtained by simply crawling the Web or a repository of interest, for example. Given the availability of these large amounts of "cheap" unlabelled examples, recently there has been a trend to exploit them in order to mitigate the effect of insufficient labelled examples on classifier accuracy. This family of approaches is designated semi-supervised learning. Semi-supervised approaches have been successfully applied to document classification [127], named entity recognition [125] and relation extraction [36]. The learning scheme lies somewhere between supervised and unsupervised: the class information is learned from the labelled examples and the underlying structure of the data from the whole of the examples, including the unlabelled ones. When labelled examples are hard to obtain, making the right assumptions about how these two aspects relate, e. g., similar examples yield similar classes, has been shown to lead to improved accuracy [34].

Research on the application of machine learning methods to document classification and information extraction is fundamental, for it deals with improving systems' accuracy in performing the tasks in place of humans and with reducing the amount of labelled examples needed. These problems are far from solved, except when considering "easy" domains and tasks, such as that of identifying people's names in an English text. This leads us to the the main motivation behind the work presented in this document.

1.1 RESEARCH QUESTIONS

Goal of this Thesis

The goal of this thesis is to provide a set of approaches, methods and findings aimed at reducing the cost of creating and maintaining document classification and information extraction systems.

Several components to that cost constitute open problems in text mining, as we shall see in chapter 3. The two main components addressed in this thesis are:

The two challenges addressed

FEATURE ENGINEERING: the cost of feature engineering is related to the effort put into understanding which characteristics or features of the data should be considered in the process of inducing a classification model.

KNOWLEDGE ENGINEERING: the cost of knowledge engineering[1] concerns the effort in creating or revising labels for documents or mentions of entities or relations in the documents. It also relates to the effort in eliciting knowledge about systems and data in order to make systems interoperable and results replicable.

The three research questions addressed throughout this thesis stem from these two challenges.

The first research question addressed in this thesis is the following:

First research question

> Which classes of features lead to an improved classification accuracy in the document classification and entity extraction tasks?

The second research question addressed in this thesis is the following:

[1] Knowledge Engineering is an engineering discipline that involves integrating knowledge into computer systems in order to solve complex problems normally requiring a high level of human expertise.[55]

Second research question

> How to reduce the amount of labelled examples needed to train machine learning-based document classification and information extraction systems, so as to relieve domain experts from this costly task?

The third research question addressed in this thesis is the following:

Third research question

> How to effectively represent knowledge about text mining systems and the data that they manipulate?

I tackle the cost of engineering features and knowledge for machine learning-based text mining in Part II, addressing each of the above questions. Some novel findings regarding feature engineering for entity extraction and multimedia document classification are presented in chapter 4, addressing the first research question. Innovative applications of advanced machine learning methods to text mining, which tackle the cost in obtaining and maintaining labelled examples are presented in chapter 5, addressing the second research question. Finally, a formalism for representing multimedia mining is proposed in chapter 6, addressing the third research question.

In the remainder of Part I, chapter 2 will provide the reader with the necessary background to understand the technical aspects required to address the research questions, while chapter 3 will introduce the reader to the above challenges in more detail.

1.2 CONTRIBUTIONS

This thesis is the culmination of several years of work on addressing the aforementioned research questions, which have materialised into the following contributions to the scientific community:

1. (*Feature Engineering*) A proposal of novel features for the document classification task which exploit the multimedia nature of the documents, together with a study that shows a consistent improvement in system accuracy when using those features, over several baselines. These ideas were partially disseminated via the following publications:

   a) J. Iria, F. Ciravegna and J. Magalhães. Web News Categorization using a Cross-Media Document Graph. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, Santorini, Greece, July 2009.

   b) J. Iria and J. Magalhães. Exploiting Cross-Media Correlations in the Categorization of Multimedia Web Documents. In *Proceedings of the IJCAI'09 Workshop on Cross-Media Information Access and Mining,* Pasadena, USA, July 2009.

2. (*Feature Engineering*) An exhaustive study on the impact of different general-purpose feature types in the entity extraction task, showing which features perform better in terms of overall system accuracy. The study also reveals that the use of rich external resources greatly contributes to the performance of IE systems and it is more likely to explain the differences in performance reported by several systems than the design decisions relative to the learning model. These ideas were partially disseminated via the following publications:

   a) J. Iria. Relation Extraction for Mining the Semantic Web. In *Semantic Web – Concepts And Applications,* Ravi Kumar Jain (editor), Icfai University Press, 2008.

   b) J. Iria, N. Ireson and F. Ciravegna. An Experimental Study on Boundary Classification Algorithms for Information Extraction using SVM. In *Proceedings of the EACL 2006 Workshop on Adaptive Text Extraction and Mining,* Trento, Italy, April 2006.

3. (*Knowledge Engineering*) A study on the use semi-supervised learning in the entity extraction task, which shows that, by exploiting unlabeled data, less labeled data is needed to achieve the same accuracy as supervised learning. These ideas were partially disseminated via the following publications:

   a) J. Iria. Automating Knowledge Capture in the Aerospace Domain. In *Proceedings of the 5th ACM International Conference on Knowledge Capture,* Redondo Beach, California, September 2009.

4. (*Knowledge Engineering*) A novel approach to domain adaptation for text categorization, which merely requires that the source domain data are weakly annotated in the form of labeled features, and an empirical study that shows that the proposed approach outperforms standard supervised and semi-supervised methods, and obtains results competitive to those reported by state-of-the-art domain adaptation methods, while requiring considerably less supervision. These ideas were partially disseminated via the following publications:

   a) C. Kadar and J. Iria. Domain Adaptation for Text Categorization by Feature Labeling. In *Proceedings of the 33rd European Conference on Information Retrieval,* Dublin, Ireland, April 2011. (runner-up to best paper award)

5. (*Knowledge Engineering*) A formalism for declaratively specifying unstructured multimedia information mining systems, their subsystems and components. A declarative specification of such a mining system is also an unambiguous description of the system that can be used to document experiments and help lowering the entry barrier to novice developers. These ideas were partially disseminated via the following publications:

   a) J. Iria. Formally Describing Unstructured Multimedia Information Mining. Under review at *Journal of Web Semantics.*

   b) J. Iria. A Core Ontology of Knowledge Acquisition. In *Proceedings of the 6th European Semantic Web Conference,* Heraklion, Crete, June 2009.

6. Three open-source software frameworks and libraries made available to the community, allowing to replicate the experiments and build on top of the existing systems (see Appendix A):

   T-REX a library for text classification, entity and relation extraction, available at `http://t-rex.sourceforge.net`.

   RUNES a plugin-based data processing framework, available at `http://runes.sourceforge.net`.

   ALEPH a machine learning framework and library, available at `http://aleph-ml.sourceforge.net` and `http://www.mloss.org/software/view/172/`.

7. Participation in the international academic competition *PASCAL Challenge on Evaluating Machine Learning for Information Extraction*[2].

1.3 IMPACT

At the time of writing, more than thirty publications cited the aforementioned publications derived from this thesis, including:

[2] `http://nlp.shef.ac.uk/pascal/`

1. T. Berners-Lee, W. Hall, J. Hendler, K. O'Hara, N. Shadbolt, and D. Weitzner. A Framework for Web Science. *Foundations and Trends in Web Science*, 1 (1). pp. 1–130, 2006.
2. A. Lavelli, M. E. Califf, F. Ciravegna, D. Freitag, C. Giuliano, N. Kushmerick, L. Romano and N. Ireson. Evaluation of machine learning-based information extraction algorithms: criticisms and recommendations. *Language Resources and Evaluation*, 42 (4). pp. 361–393, 2008.
3. N. Aussenac-Gilles and D. Sörgel. Text analysis for ontology and terminology engineering. *Applied Ontology*, 1 (1). pp. 35–46, 2005.
4. F. Suchanek, G. Ifrim and G. Weikum. Combining linguistic and statistical analysis to extract relations from web documents. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Philadelphia, PA, USA, August 2006.
5. M. Tchalakova, B. Popov, M. Yankova. Methodology for Bootstrapping Relation Extraction for the Semantic Web. In *Proceedings of The Twelfth International Conference on Artificial Intelligence: Methodology, Systems, Applications*, Varna, Bulgaria, September 2006.
6. M. Chen, X. Liu, and J. Qin. Semantic relation extraction from socially-generated tags: a methodology for metadata generation. In *Proceedings of the 2008 International Conference on Dublin Core and Metadata Applications*, Berlin, Germany, September 2008.
7. R. Pasley, P. Clough, and M. Sanderson. Geo-tagging for imprecise regions of different sizes. In *Proceedings of the 4th ACM Workshop on Geographical Information Retrieval*, Lisbon, Portugal, November 2007.
8. L. Specia and E. Motta. A hybrid approach for extracting semantic relations from texts. In *Proceedings of the COLING-ACL 2006 Workshop on Ontology Learning and Population*, Sydney, Australia, 2006.

Furthermore, the work described in this thesis has given origin to other work, published with colleagues:

1. Z. Zhang and J. Iria. A Novel Approach to Automatic Gazetteer Generation using Wikipedia. In *Proceedings of the ACL 2009 Work-*

shop on Collaboratively Constructed Semantic Resources, Singapore, August 2009.

2. L. Xia and J. Iria. An Approach to Modeling Heterogeneous Resources for Information Extraction. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC-08)*, Marrakesh, Morocco, May 2008.
3. M. A. Greenwood and J. Iria. Saxon: An Extensible Multimedia Annotator. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC-08)*, Marrakesh, Morocco, May 2008.

The aforementioned open-source projects have had several hundred downloads each so far, and have attracted the attention of several companies.

1.4 SUMMARY

The goal of this chapter was to familiarize the reader with the general problem of providing structure to unstructured data, to introduce the tasks of document classification and information extraction, to present the research questions addressed in the thesis, and to summarize the contributions made to the state-of-the-art in the area.

The following is a summary of the thesis structure.

PART I – THE DOMAIN OF TEXT MINING

CHAPTER 2, FROM STRUCTURED TO UNSTRUCTURED DATA: provides the reader with the necessary background to understand the technical aspects of the work presented in the remainder of the thesis.

CHAPTER 3, OPEN PROBLEMS IN TEXT MINING: explains in detail the open problems in text mining that are relevant to the work presented in Partt II of this thesis.

PART II – TACKLING THE COST OF FEATURE AND KNOWLEDGE ENGINEERING

CHAPTER 4, FEATURE ENGINEERING FOR TEXT MINING: investigates the impact of incorporating diverse feature types into boundary classification algorithms for Entity Extraction, and proposes novel

Document Classification features able to generically exploit information from across different parts of multimedia documents.

CHAPTER 5, OVERCOMING THE SCARCENESS OF LABELLED DATA: investigates the impact of using semi-supervised learning in the task of Entity Extraction from highly technical documents, and proposes using feature labelling, as opposed to instance labelling, in adapting an existing Text Classification model to different domains.

CHAPTER 6, A FORMALISATION OF MULTIMEDIA MINING: proposes a novel formalism for describing unstructured multimedia information mining systems, extending existing state-of-the-art formalisms for describing software and multimedia content.

CHAPTER 7, CONCLUSIONS: concludes with a review of the main contributions of this thesis.

BLANK PAGE
IN
ORIGINAL

2 FROM UNSTRUCTURED TO STRUCTURED DATA

Motivating example: document categorisation

Imagine that we would like to automatically categorise multimedia documents obtained from a popular news website according to a set of pre-defined topics. Available is a set of examples of such categorisation, and therefore a natural decision is to adopt a supervised machine learning-based approach to the problem, to exploit the availability of labelled data. Moreover, the system would benefit from making use of both text and image features, as both are potentially valuable. For example, the presence of images with certain characteristics (e. g., vivid colours) increases the likelihood that the surrounding text concerns a given category (e. g., sports).

Motivating example: entity extraction

Additionally, we would like to automatically recognise certain classes of entities in text, e. g., the judge in a competition, the venue of a conference. The output of the system consists in populating a database table with all the entities found together with metadata about exactly where in the documents they were found. Available is a set of examples of how to recognise the entities in question, in the form of manually marked-up text by a human annotator, which again leads to the decision of employing a supervised learning approach. This system would benefit from using external resources to aid the classification of candidate text segments. Concretely, the use of gazetteers of people and location names, and World Wide Web (WWW) resources, holds the potential of increasing the accuracy of the classifier.

The above are examples of the Text Classification (TC) and Entity Extraction (EE) tasks, and constitute a good starting point for thinking about the core problems addressed in this thesis. In the remainder of this chapter I will introduce the reader to the technical background required to understand the contributions to the state-of-the-art in text mining set forth in Part II of the thesis.

2.1 ANATOMY OF AN UNSTRUCTURED INFORMATION MINING SYSTEM

In the last decade, most of the approaches that were proposed to solve text mining problems have followed the general trend in Information Retrieval (IR) and Natural Language Processing (NLP) of moving towards the adoption of machine learning as the main paradigm at the heart of the approaches. From a bird's eye perspective, the design of a ML-based approach to most current text mining problems can be regarded as following three main steps:'

1. Cast the problem as a classification task: identify the target classification object, and how to obtain labelled examples for it, to train a classifier
2. Select an off-the-shelf learning algorithm with the appropriate characteristics, e. g., the right time complexity
3. Determine which features in the text characterise the object, extract them and pass them to the learning algorithm

Accordingly, systems built to deliver text (or, more generally, multimedia) mining functionality address at least four major *functional* concerns:

DECOMPOSITION: Both text and images need to be decomposed into finer-grained media segments, which constitute the building blocks for deriving Information Extraction patterns and models. Text is typically decomposed into sentences, phrases and tokens. Images are typically segmented into regions of interest.

SEGMENT ANALYSIS: Media segments need to be processed and, in many cases, tagged, that is, extra information, obtained through some form of analysis and/or use of external resources, is attached to the segment, with the purpose of enriching the patterns and models mentioned in the previous point. Typical text tagging tools are part-of-speech and orthography taggers [139], which attach tags at the token level, or chunkers [1], which attach tags at the sentence level. Typical image analysis tools are colour [160] and texture analysers [163], and edge detectors [124].

DATA MODELLING: Predictive classification/extraction models, capable of classifying segments into classes of interest given the decomposed and tagged input media, need to be constructed. The classes of interest come from an understanding of the problem domain, which is ideally encoded in the form of a domain ontology. *Manually* built models consist mainly of text patterns, carefully created, tested and maintained by domain and linguistic experts. *Automatically* induced models are created by machine learning-based systems, which require the availability of at least a few labeled media segments, and tap into a wealth of machine learning literature for a choice of algorithms and meta-algorithms [121]. Those useful for the understanding of the ideas in this thesis will be introduced in the following sections.

SEMANTIC ANNOTATION: Extraction models need to be applied over unseen media to infer new information, typically outputted in the form of semantic annotations [77]. In the simplest case, which we will restrict ourselves to here for the sake of exposition, the application of the models directly yields the target information of interest. In other cases, though, some form of validation, consistency checking, and merging of intermediate information may be required, which would add further concerns to this list.

Figure 2 illustrates the basic anatomy of an information extraction system. There is a wealth of literature on automating the several types of tasks delimited by the above concerns, and methods and software systems exist for each of them, featuring varying degrees of capability, correctness and performance.

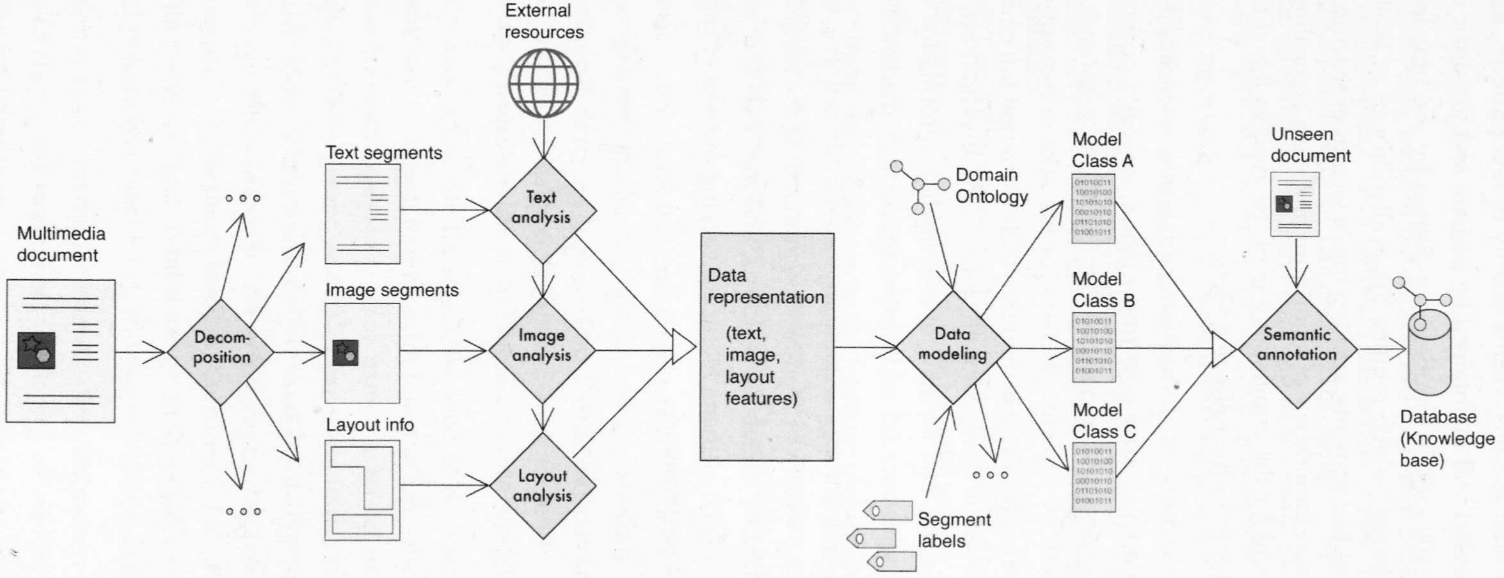


Figure 2: The basic anatomy of an unstructured information mining system. It comprises at least four major processing steps, addressing different concerns: decomposition, segment analysis, data modelling and semantic annotation.

Throughout this thesis, the focus will be mostly on novel, more effective methods for data modelling (chapter 4 and chapter 5). A contribution to solving the semantic annotation problem is also presented in chapter 6.

2.2 TEXT MINING FROM A MACHINE LEARNING VIEWPOINT

When adopting a Machine Learning (ML) approach to text mining, most tasks are adequately modelled as classification tasks, as mentioned in the previous section. *Classification* can be described as the task of assigning a class y to an observation $\mathbf{x}$. More formally, a learning algorithm (in the *supervised* setting) takes a set of labeled training examples, $(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_n}, y_n)$ as input, where $\mathbf{x_i} \in \mathcal{X}$ is typically a feature vector that characterises some object in the data $\mathcal{X}$, and the corresponding label y_i belongs to a finite set of classes denoted as $\mathcal{Y}$. The goal of classification is to form a hypothesis $h : \mathcal{X} \mapsto \mathcal{Y}$ which maps an input $\mathbf{x}$ to an output y.

2.2.1 *Text Categorization as a Classification Task*

Document classification is the task of assigning documents to one or more pre-defined categories, based on their content [150]. More formally, the task consists of assigning a Boolean value to each pair $(d_j, c_i) \in \mathcal{D} \times \mathcal{C}$, $j = 1 \ldots |\mathcal{D}|$, $i = 1 \ldots |\mathcal{C}|$, where $\mathcal{D}$ is the set of documents to classify and $\mathcal{C}$ is the set of pre-defined categories that those documents may belong to. Thus, a value of *True* attributed to (d_j, c_i) means that d_j may be "filed under" c_i, while, conversely, a value of *False* means that it may not.

Document classification

From a ML perspective, the task is that of approximating the unknown *target* function $\check{h} : \mathcal{D} \times \mathcal{C} \rightarrow \{True, False\}$, by a function of the same form $h : \mathcal{D} \times \mathcal{C} \rightarrow \{True, False\}$, called the *classifier*, such that $\check{h}$ and h provide identical outputs given identical inputs (as much as possible).

When the task is to assign exactly one category to each $d_j \in \mathcal{D}$, this is called *single-label* classification. When multiple categories may be assigned, the task is termed *multi-label* classification. Moreover, in the most common variants of the task, the categories or classes $\mathcal{C}$ are treated as pure string labels, since no additional knowledge about their meaning is utilised.

Text Classification (TC), also known as text categorization, is the variant of document classification that deals with documents that contain only text, or, equivalently, assumes that the textual part of the document content is the only part that carries valuable information for discriminating between documents. The most widely used feature representation is the *bag of words*, where each document is represented as a vector of words and their frequency in the corpus.

2.2.2 *Entity Extraction as a Classification Task*

Entity Extraction

As mentioned in chapter 1, EE is the task of identifying mentions of entities of interest in documents. The mentions are typically noun phrases and consist of one to a few tokens in the text. Examples of EE tasks include identifying the speaker featured in a talk announcement, as is done in this chapter, or finding the proteins mentioned in a biomedical journal article, as is done in [54], but there are many more application domains: the MUC [70], ACE [2] and CoNLL [52] academic competitions have, for several years, formally evaluated participant's EE systems on corpora that have covered different domains.

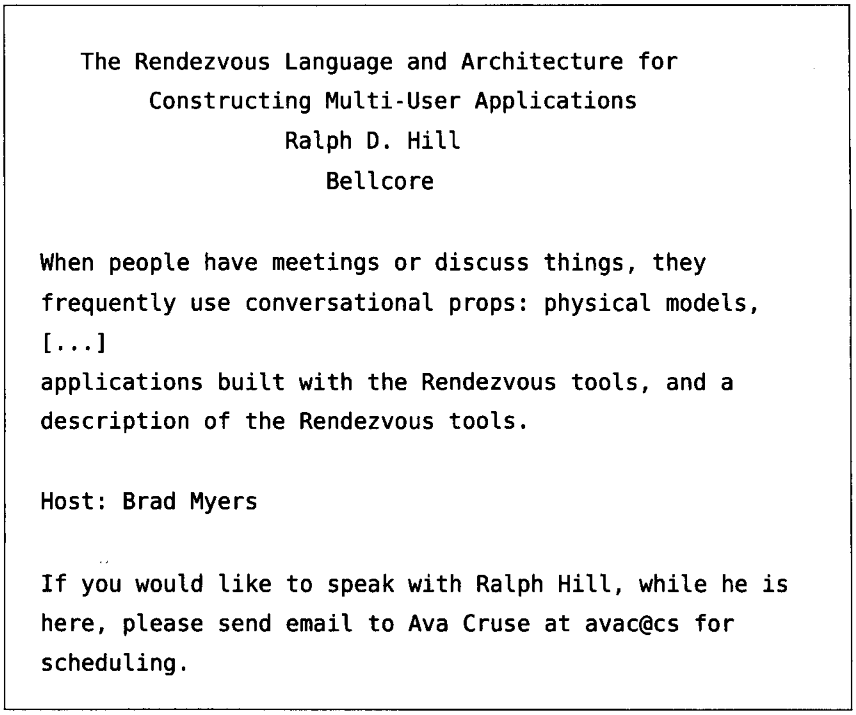
Example

Figure 3 shows an excerpt of a seminar announcement of the kind that can be found very commonly in any University e-mail system, circulating amongst faculty[1]. The aim of EE is to enable the creation of systems capable of extracting target information from documents. For example, from the excerpt above one might be interested in extracting the name of the speaker at the seminar. Judging from the example shown, the task is not a straightforward one, since the name of the speaker may appear in different forms, e. g., "Ralph D. Hill" and "Ralph Hill", and also because the names of several other people, who do not have the role of speaker at the seminar, appear in the text. Moreover, it should be made clear that the goal is not to enable extracting the name of the speaker in *this* particular announcement, but rather to find rules or models of the data that allow extracting the name of the speaker in *any* announcement that shares similar characteristics with this one.

Models for Entity Extraction

Modeling Entity Extraction (EE) as a classification task involves deciding on the target classification object, how to obtain features that characterise it, and how to obtain labeled examples. The most widely used models are the following:

[1] This example is taken from the *Seminar Announcements* corpus, a standard dataset for the EE task, which will be described in more detail in subsection 4.1.1



```
     The Rendezvous Language and Architecture for
          Constructing Multi-User Applications
                    Ralph D. Hill
                       Bellcore

When people have meetings or discuss things, they
frequently use conversational props: physical models,
[...]
applications built with the Rendezvous tools, and a
description of the Rendezvous tools.

Host: Brad Myers

If you would like to speak with Ralph Hill, while he is
here, please send email to Ava Cruse at avac@cs for
scheduling.
```

Figure 3: An excerpt of a seminar announcement. Which rules would extract the speaker "Ralph Hill" in the above text, but also other speakers in other texts like this one?

SEGMENT MODELS In this type of model, segments of text, composed of one or more tokens, constitute the target classification object. The labels are attached to each segment, and the features describe a segment and its constituent tokens. More formally, a segmentation s of input text of length n is a sequence of segments $\mathbf{s} = s_1 \ldots s_{|s|}$ such that the first segment starts at 1, the last segment ends at n, and segment s_{j+1} begins right after segment s_j ends. Each segment is composed of a sequence of tokens $\mathbf{t} = t_1 \ldots t_{|t|}$ and has a label $y \in \mathcal{Y}$ attached to it. Segment-level features can be, for example, the similarity of the segment to an entity in a database, or the length of the segment. Thus, segment-level features capture joint properties of the tokens and can potentially be more powerful than token-level features alone. However, a segment model is computationally more complex than a token model, because all possible segmentations of the text need to be tried. Examples of systems that employ this type of model include [32, 62, 145].

TOKEN MODELS Here tokens constitute the target classification object, labels are attached to each token, and features describe a token. More formally, a sequence of tokens $\mathbf{t} = t_1 \ldots t_{|\mathbf{t}|}$ has an associated label sequence $\mathbf{y} = y_1 \ldots y_{|\mathbf{y}|}$, and the problem is to determine the correct label sequence. Naturally, since entity mentions may span several tokens, there needs to be a way to re-assemble the tokens into a segment. The way to do that is to consider that each token plays a role in the segment, and assign it the label corresponding to that role. Hence, with the "IOB" scheme, a token gets a "B" (begin) label if it is the first token of an entity mention, an "I" (inside) label if it is part of an entity mention but it is not its first token, and an "O" (outside) label if it is not part of any entity mention. The "BIE" scheme additionally uses the "E" (end) label to mark the final token of an entity mention. Token-level features can be, for example, the word itself, the orthography, or the part-of-speech. Examples of systems that employ this type of model include [102, 37].

BOUNDARY MODELS In this type of model, the boundary (or virtual separator) between two tokens in the text is the target classification object. Labels are attached to the boundary. The features are the same as in token models. In practice, this model is very similar to the token model, the main difference being the labeling scheme. Here, typically two independent binary classifiers label each boundary as being the start of an entity (or not), and as being the end of an entity (or not). Re-assembling the entity mentions from a sequence of "start" and "end" labels requires solving the problem of how to pair them — which ones to discard and which ones to keep. Examples of systems that employ this type of model include [61, 38, 59].

CHARACTER MODELS Here characters or character *n-grams* constitute the target classification object, labels are attached to each character, and features describe a character and their neighbouring characters. When using character-level models for word-evaluated tasks like EE, care needs to be taken to ensure that multiple characters inside a single word do not receive different labels. An example of a system that explores this type of model can be found in [96].

Does object x_i belong to class c_i?		Oracle	
		Yes	No
Classifier	Yes	tp	fp
	No	fn	tn

Table 1: Contingency table, which derives the quantities tp, fp, fn, and tn from answering the question"does object x_i belong to class c_i?".

2.3 EVALUATING TEXT MINING

Before delving into concrete ML methods and techniques for text mining, some concepts related to the empirical evaluation of ML-based approaches need to be introduced. Rather than concentrating on issues of computational efficiency, the evaluation of text mining systems typically focuses on determining how *effective* the system is, measuring its capability of making predictions on unseen data. In particular, classification effectiveness will be measured in this thesis using the classic Information Retrieval measures of precision and recall, as explained in what follows.

In the context of learning theory, an *oracle* is an entity that is able to provide the correct answer to the question "does object x_i belong to class c_i?". For example, an oracle can be a human domain expert interacting with an application that wraps a learning algorithm. For example, in ML-based EE, the oracle consists of a method that consults previously collected entity labels in the *gold standard* corpora, i. e., the corpora that are reserved for estimating the precision and recall of the system.

Oracle

Gold Standard

Precision of a classifier is defined as the probability of a random candidate object being correctly classified by the classifier under class c_i. *Recall* is defined as the probability of a random candidate object that ought to be classified under class c_i by the classifier actually being classified. In other words, precision may be viewed as the "degree of soundness" of the classifier, whilst recall may be viewed as its "degree of completeness". These probabilities may be estimated in terms of a contingency table, which defines the elementary quantities tp, fp, fn, tn. See Table 1.

The quantity tp denotes the "true positives", that is, the number of candidate entity mentions that the classifier has predicted to belong to class c_i, and the oracle agreed. Conversely, fp denotes the "false positives", the number of candidate mentions that were classified in-

True positives, true negatives, false positives and false negatives correctly by the classifier, according to the oracle. Those candidate entities that the classifier classifies as negative can be either fn, "false negatives", or tn, "true negatives", again according to the verdict given by the oracle. In the context of EE, fp are many times informally called the system "mistakes", while fn the system "misses".

Accuracy, precision and recall

Based on the above unit terms it is now possible to define accuracy (Acc), precision (Pre) and recall (Rec), as :

$$\mathrm{Acc} = \frac{\mathrm{tp} + \mathrm{tn}}{\mathrm{tp} + \mathrm{fp} + \mathrm{fn} + \mathrm{tn}} \quad . \tag{2.1}$$

,

$$\mathrm{Pre} = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fp}} \tag{2.2}$$

and

$$\mathrm{Rec} = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fn}} \tag{2.3}$$

Micro- and macro-averaging

When computing the average precision and recall for several classes $c_i \in C$, two different methods may be adopted :

MACRO-AVERAGING : precision and recall are first computed "locally" for each category, and then "globally" by simply averaging over the results of the different classes, i. e., for precision:

$$\mathrm{Pre}_{macro} = \frac{1}{|C|} \sum_{c_i \in C} \mathrm{Pre}_{c_i}$$

MICRO-AVERAGING : precision and recall are computed by directly summing the statistics tp, fp, and fn from each class, i. e., for precision:

$$\mathrm{Pre}_{micro} = \frac{\sum_{c_i \in C} \mathrm{tp}_{c_i}}{\sum_{c_i \in C} \mathrm{tp}_{c_i} + \sum_{c_i \in C} \mathrm{fp}_{c_i}}$$

F-measure

In many cases it is useful to have a single measure to compare systems. The most commonly used way of combining precision and recall is the F_1-measure (or, simply, *F-measure*), their harmonic mean.

Concretely, F-measure is defined as:

$$F_1 = \frac{2 \times \text{Pre} \times \text{Rec}}{\text{Pre} + \text{Rec}} \tag{2.4}$$

The above formula is a special case of the more general F_β formula (setting $\beta = 1$):

$$F_\beta = (1 + \beta^2) \times \frac{\text{Pre} \times \text{Rec}}{(\beta^2 \times \text{Pre}) + \text{Rec}} \tag{2.5}$$

Measuring the accuracy of an EE system in terms of precision, recall and F-measure involves setting up an experiment that takes the initial corpus of documents $D = \{d_1, \ldots, d_{|D|}\}$ and splits it into sets, not necessarily of equal size. Some sets, referred to as *training data* or *training corpus*, are used for training the system, that is, the system is allowed to consult the oracle for those documents, in order to obtain the labels for the underlying learning algorithm; while the remainder of the sets, referred to as *test data* or *test corpus*, are reserved to testing the accuracy of the system, that is, the oracle cannot be consulted for the documents in the test corpus.

There are numerous methods to split a corpus into training and test sets. Two of the most popular are *random splitting* and *n-fold cross-validation*. The former method simply draws documents at random from D, drawing a document with probability p into the training set and probability $1 - p$ into the test set, where p is a user-defined parameter. The latter method divides the corpus into k sets, uses one of the sets as the test set and the other $k - 1$ sets as the training set, and then runs k trials so that each of the sets is used as the test set once. The advantage of this method is that it reduces the influence of the way the corpus is split on the results, since every document is assigned to a training set $k - 1$ times, leading to decreasing variance in the F-measure estimate as k increases. The disadvantage of this method is that the amount of training data used increases as k increases, which can result in model overfitting. A variant of these methods is to perform the random splitting method k different times, in which case it is possibly to independently choose how large each training set is and how many trials to average over.

Cross-validation

Because of the nature of text as object for classification, for the purpose of computing tp, fp and fn there are several ways to determine what constitutes a match between the output of the system and the gold

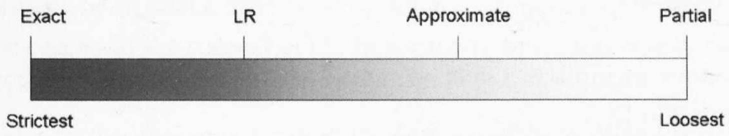


Figure 4: Spectrum of leniency in what is considered an entity mention match in Entity Extraction.

Types of entity matches

standard. The different ways vary with respect to leniency, forming a spectrum (see Figure 4). A *strict* match requires the entity mention to be perfectly recognised, while a *partial* match allows any substring of the entity mention outputted by the system to match the one in the gold standard, or vice-versa. In between these two extremes, *Left-Right* (LR) matches require that either the left (start) or the right (end) of the entity mention is correctly recognised, while *approximate* matches allow a substring of the entity mention to be recognised only (but not vice-versa).

2.4 MODELS FOR TEXT CLASSIFICATION

Text Classification is a well studied problem with a large body of literature spanning several decades — see Sebastiani [151] for a high-level comprehensive survey of approaches to the problem. This section introduces selected machine learning topics required to understand the ideas in the remainder of the thesis.

2.4.1 *Generalised Linear Models*

Recall from subsection 2.2.1, that a learning algorithm takes a set of *training examples* $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ as input, where n is the number of examples, x_i represents a given object i from domain X, and the corresponding labels y_i belong to a finite set of k classes denoted as Y. The goal of the algorithm is to form a hypothesis $f : X \mapsto Y$ which maps an input $x \in X$ to an output $y \in Y$. The nature of the output y varies with the type of classification task: in a binary classification

Multi-class multi-label classification setting $y \in \{-1, +1\}$, in a multi-class single-label setting $y \in \{1, \dots, k\}$ and in the multi-class multi-label setting $y \in \mathcal{P}(\{1, \dots, k\})$, where $\mathcal{P}(\cdot)$ denotes the power set.

Linear models are a class of ML models that operate under the assumption that the relationship between the input variables $\mathbf{x} = [x_1, \dots, x_{|\mathbf{x}|}]$ and the output variable y is linear, or can be approximated by a hyperplane with little error. In the TC task, $\mathbf{x}$ represents the text features of a document, and y the topic to predict. The output variable y can be regarded as a random variable with an associated probability distribution. If so, for each document topic the goal is thus to compute $P(y|x)$.

A linear relationship between $\mathbf{x}$ and y, modelled by $\mathbf{w}$, can be written as:

$$E[y|\mathbf{x}, \mathbf{w}] = \mathbf{w} \cdot \mathbf{x} = \omega_0 + \omega_1 x_1 + \dots + \omega_{|\mathbf{w}|} x_{|\mathbf{x}|}$$

where $\omega_1, \dots, \omega_{|\mathbf{w}|}$ are called the *regression coefficients* and can be interpreted as measuring the importance or weight of the several corresponding model components $x_1, \dots, x_{|\mathbf{x}|}$, and ω_0 is usually called the intercept or *bias* of the model.

Generalised linear models

The linearity assumption can become inadequate when trying to model more complex data. As an extension to the simple linear model, Generalised Linear Models (GLMs) [117] introduce a link function $g : \mathfrak{R} \to \mathfrak{R}$ to model non-linear relations between the input and the output. A GLM has the form:

$$E[y|\mathbf{x}, \mathbf{w}] = g^{-1}(\omega_0 + \omega_1 x_1 + \dots + \omega_{|\mathbf{w}|} x_{|\mathbf{x}|})$$

where g is a monotonic differentiable function that should be chosen according to the problem at hand. The most common choices of g instantiate the GLM framework into the well-studied models of linear regression, logistic regression and log-linear regression.

Linear regression

To recover the linear regression model, it suffices to define the link function as $g(x) = x$. The random variable y is assumed to follow a normal distribution with mean μ and variance σ^2, i. e.:

$$y \sim N(\mu, \sigma^2)$$

This model is applicable to regression settings where y and a certain transformation of X results in a Normal distribution. Note that in many cases of interest the variable y is not continuous but discrete (i.e., expresses "belongs to class or not") so the Normal linear model is not adequate.

Logistic regression

A widely adopted linear model in a variety of domains is *logistic regression* [4]. To obtain logistic regression under the GLM framework, we define the link function to be $g(x) = \text{logit}(\mu)$, where the *logit* function is defined as:

$$\text{logit}(x) = \log\left(\frac{x}{1-x}\right) \quad .$$

In this model the random variable y is assumed to follow either a binomial distribution

$$y \sim \text{Bin}(n, p),$$

or a multinomial distribution

$$y \sim \text{Mul}(n, p_1, \ldots, p_k),$$

corresponding to the single- and multi-label classification problems introduced in subsection 2.2.1, respectively.

2.4.2 *Margin Classifiers*

Separating hyperplane

Margin

A margin classifier finds an optimal separating hyperplane between data points of different classes in a high dimensional feature space induced by a mapping of the data. The notion of *margin* reflects the distance between the hyperplane and the support vectors. The notion of *optimality* of the hyperplane is deeply related to the notion of margin: the best hyperplane, from a generalisation viewpoint, is the one that maximises the margin, that is, the one that maximally separates the points belonging to different classes. This is not only intuitively appealing, but has also been explained formally in [165].

Support Vector Machines (SVMs) are a widely used method belonging to the family of *margin classifiers*. SVMs have been applied, for more than one decade, to a wide range of domains and are well-known for their robustness to noisy and sparse data. The method gets its name from *support vectors*, which are the data points, that determine the position and orientation of the separating hyperplane in that space. The hyperplane is often called the *decision boundary*.

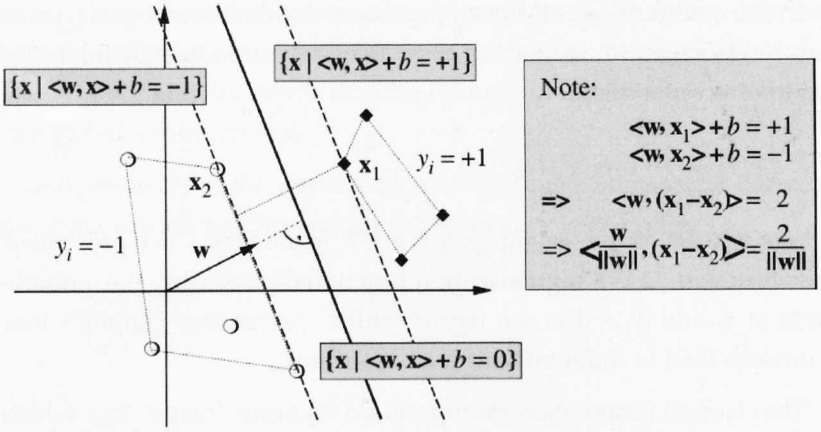

{x | <w, x> + b = −1}
{x | <w, x> + b = +1}
{x | <w, x> + b = 0}
x_1
x_2
$y_i = +1$
$y_i = -1$
w
Note:
$\langle w, x_1\rangle + b = +1$
$\langle w, x_2\rangle + b = -1$
$\Rightarrow \langle w, (x_1 - x_2)\rangle = 2$
$\Rightarrow \langle \frac{w}{\|w\|}, (x_1 - x_2)\rangle = \frac{2}{\|w\|}$

Figure 5: An illustration of the basic principle behind Support Vector Machines. Circles and diamonds denote two different classes. The solid line represents the maximally separating linear decision boundary (hyperplane). $\mathbf{x_1}$ and $\mathbf{x_2}$ are support vectors. Source: [149].

To illustrate with a simple example, consider the binary classification toy problem depicted in Figure 5. For linearly separable problems, there exists a separating hyperplane satisfying $y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) > 0, \forall i \in \{1, 2, \ldots, n\}$, where $y_i \in \{-1, +1\}$ (the class label), $\mathbf{w} \in \mathbb{R}^d$ (the weight vector), $\mathbf{x}_i \in \mathbb{R}^d$ (a data point), $b \in \mathbb{R}$ (the threshold), n is the dataset size, and $\langle \cdot, \cdot \rangle$ denotes the dot product. The optimal hyperplane, the one that maximizes the margin (shown as a solid line in the figure), can be found by solving an optimisation problem. Concretely, SVMs find an *approximate* of the optimal hyperplane as linear combination of the support vectors, by solving the following *quadratic minimisation* problem:

Optimal hyperplane

$$\underset{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}}{\text{maximize}} \quad \min\{\|\mathbf{x} - \mathbf{x}_i\| : \mathbf{x} \in \mathbb{R}^d,\ \langle \mathbf{w}, \mathbf{x}\rangle + b = 0,\ i = 1, \ldots, n\}$$

Rescaling $\mathbf{w}$ and b such that the data points closest to the hyperplane satisfy $|\langle \mathbf{w}, \mathbf{x}_i\rangle + b| = 1$, the canonical form $(\mathbf{w}, b)$ of the hyperplane is obtained, which satisfies $y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geqslant 1, \forall i \in \{1, 2, \ldots, n\}$. The margin[2] in this case equals to $\frac{1}{\|\mathbf{w}\|}$.

[2] This can be seen by considering two closest points x_1 and x_2 on opposite sides of the margin, and projecting them onto the hyperplane normal vector $\frac{\mathbf{w}}{\|\mathbf{w}\|}$.

In the common case of linear classifiers, the decision boundary can be seen as a linear function $f(x) = \mathbf{w}^T + b$ that minimizes the following regularised cost function:

$$E(f) = \sum L(\mathbf{w}, \mathbf{x}) + \lambda\Omega(f),$$

where $\mathbf{w} \in \mathbb{R}^d$ is the weight vector, $b \in \mathbb{R}$ is the bias, L is a prescribed loss function, Ω is a regularization functional measuring the smoothness of f, and $\lambda > 0$ is the regularization parameter. Different loss functions lead to different learning algorithms.

The classical formulation for supervised SVM uses "hinge" loss, which is defined as $L(\mathbf{w}, \mathbf{x}) = \max(0, 1 - y_i\langle\mathbf{w}, \mathbf{x}_i\rangle)$, and "L2-norm" regularisation, which is defined as $\Omega(f) = \frac{1}{2}||\mathbf{w}||^2$, leading to the following primal formulation:

$$\underset{\mathbf{w} \in \mathbb{R}^d}{\text{minimize}} \quad \frac{\lambda}{2}||\mathbf{w}||^2 + \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i\langle\mathbf{w}, \mathbf{x}_i\rangle)$$

The dual formulation for the above problem is given by [44]:

$$\begin{aligned} \underset{\alpha \in \mathbb{R}}{\text{maximize}} \quad & \textstyle\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle \\ & \text{subject to } 0 \leqslant \alpha_i \leqslant \frac{1}{\lambda n}. \end{aligned}$$

In this formulation, if we replace $\langle x_i, x_j\rangle$ by K_{ij}, yielding

$$\begin{aligned} \underset{\alpha \in \mathbb{R}}{\text{maximize}} \quad & \textstyle\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j K(x_i, x_j) \\ & \text{subject to } 0 \leqslant \alpha_i \leqslant \frac{1}{\lambda n}. \end{aligned}$$

then the above method is called a *kernel method*, and $K(\mathbf{x}_i, \mathbf{x}_j) \equiv \langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle$ is called the *kernel function*. The *linear kernel* maps the space onto itself, i.e., $K_{linear}(\mathbf{x}_i, \mathbf{x}_j) = \langle\mathbf{x}_i, \mathbf{x}_j\rangle$. Using the kernel, the $\mathbf{x}_i$ are mapped into a higher (potentially infinite) dimensional space by function $\phi : \mathbb{R}^d \mapsto \mathbb{R}^k$, where typically $d \ll k$. SVM finds a linear separating hyperplane with maximal margin in this higher dimensional space. This enables applying SVMs to problem in which the data is not linearly separable, by defining an appropriate ϕ. Ensuring that the kernel function is positive semidefinite guarantees the convexity of the search space and, therefore, that the optimisation procedure converges to a global minimum [44].

Kernel

Independently of the way in which the optimal hyperplane is derived, the SVM decision function for an input vector $\mathbf{x}$ is given by

SVM decision function

$$\operatorname{sgn}\left(\sum_i y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b\right). \tag{2.6}$$

2.4.3 *Feature Selection*

A technique that has enjoyed success in improving accuracy while at the same time reducing learning algorithm running times in the context of the document classification task is *feature selection* [60]. Feature selection consists in choosing a subset of relevant features from the set of possible features, throwing away irrelevant and redundant features which may contribute negatively to the generalisation ability of the learning algorithm. As a side effect, feature selection speeds up the learning process and improves the interpretability of the learned model, as algorithm, in the former, and humans, in the latter, have to deal with less features.

Impact of feature selection

Feature selection algorithms typically fall into two classes: *feature ranking* and *subset selection*. While subset selection methods search the set of possible features for the optimal subset using cross-validation, feature ranking methods rank the features by a metric, and eliminate all features that do not achieve an adequate score.

Feature ranking

The following feature ranking metrics are used in this thesis:

FREQUENCY This method simply ranks features according to the number of times they occur in the dataset, that is

$$\mathrm{Freq}(F) = tp + fp. \tag{2.7}$$

INFORMATION GAIN The Information Gain (IG) of a feature is defined intuitively as the reduction in uncertainty about the class of a learning instance once the value of that feature for that instance is known. Uncertainty is measured in terms of *entropy*. The entropy of a class Y is given by:

$$H(Y) = -\sum_{y_i \in Y} p(y_i) \log_2 p(y_i).$$

The *conditional entropy* measures the entropy associated with feature F:

$$H(Y|F) = \sum_{f_i \in F} p(f_i) H(Y|F = f_i),$$

where

$$H(Y|F = f_j) = -\sum_{y_i \in Y} p(y_i|f_j) \log_2 p(y_i|f_j).$$

The Information Gain is the difference between the entropy and the conditional entropy for the feature:

$$IG_Y(F) = H(Y) - H(Y|F).$$

Because $H(Y)$ is fixed, a feature F_1 has a higher information gain than feature F_2 if $H(Y|F_1) < H(Y|F_2)$. Thus, for the purposes of ranking, $H(Y)$ can be omitted.

When the learning model is formulated with binary classes and binary features, the equation can be re-written in terms of tp, fp, fn and tn as:

$$IG_{Y=\{-1,+1\}}(F = \{f, \bar{f}\}) = -p(f)e(tp, fp) - p(\bar{f})e(fn, tn), \quad (2.8)$$

where $p(f) = (tp + fp)/(pos + neg)$, $p(\bar{f}) = 1 - p(f)$, and

$$e(x, y) = -\frac{x}{x+y} \log_2 \frac{x}{x+y} - \frac{y}{x+y} \log_2 \frac{y}{x+y}.$$

From its definition, it is clear that the IG measure is a supervised strategy, because it uses the entity mention labels in its choice of features to retain. In contrast, the frequency measure is unsupervised.

CROSS ENTROPY Cross Entropy (CE) is similar to IG, the difference being that, with binary-valued features, IG considers both the presence and absence of the feature while CE only considers the presence of the feature. In fact, IG can be defined in terms of CE as:

$$IG_Y(F) = CE_Y(F = f) + CE_Y(F = \bar{f}).$$

The CE equation is re-written in terms of tp, fp, fn and tn as:

$$CE_{Y=\{-1,+1\}}(F = \{f, \bar{f}\}) = -p(f)e(tp, fp), \tag{2.9}$$

where e was defined above.

2.5 MODELS FOR ENTITY EXTRACTION

Entity Extraction is a well studied problem with a large body of literature spanning several decades — see Moens [122] for a high-level comprehensive survey of approaches to the problem. This section adds to the machine learning methods and techniques introduced in subsection 2.2.1 further selected ML topics required to understand the ideas related to EE in the remainder of the thesis.

2.5.1 *Rule Induction Approaches*

Most of the research on EE prior to the last decade made use of symbolic techniques, as opposed to the statistical methods that will be discussed later. Starting with manually built symbolic rules, approaches then moved to pursue the automatic induction of the rules, while still keeping them human-understandable. A good way to illustrate this class of approaches is to take an advanced one as representative of its class and review it in some detail. The $(LP)^2$ algorithm was chosen for this purpose. Other successful early systems include Rapier [32], SLIPPER [40], Snow-IE [145], BWI [61] and SRV [62].

The $(LP)^2$ *algorithm*

The $(LP)^2$ algorithm [38] is a supervised rule induction algorithm that follows the boundary classification model. It has been successfully applied to the EE task. The algorithm is similar to the CN2 algorithm described in [121].

$(LP)^2$ induces two types of symbolic rules:

1. rules that spot entity mentions in the text;
2. rules that correct the mistakes made by the previous rules.

Rules are induced by generalising over a set of examples of entity mentions marked with XML tags in a training corpus and taken as positive examples. The rest of the corpus is considered to be a pool of negative examples.

Tagging rules

A *tagging rule* is composed of a left hand side, which consists of a pattern of conditions over a "window" of token positions relative to the boundary to which the rule applies, and a right hand side that is an action of inserting an XML tag in the text. Each rule inserts a single tag, which can be a open or close tag, e. g., `</speaker>`. For example, the following rule would insert a `<speaker>` tag in the text when the sequence of tokens "Speaker:" is detected in the text:

$$\mathtt{Word}_{-2} = \text{Speaker} \wedge \mathtt{Orthography}_{-1} = \text{punct.} \longrightarrow \texttt{<speaker>},$$

Contextual rules

where "punct." is a class of features for punctuation characters in the text. One other type of rules in $(\mathtt{LP})^2$, called *contextual rules*, complement the tagging rules. They rely on the tags inserted by the tagging rules in order to induce rules that have high precision only in the presence of a another tag. For example, contextual rules are able to close open tags, i. e., they will use the presence of a `<speaker>` to insert a missing `</speaker>`.

Shift rules

Finally, *shift rules* are rules that are induced by an analysis of the mistakes made by the tagging rules and the contextual rules. They simply shift the predictions made by other rules to some relative position. For example, if `</speaker>` is regularly predicted one token to the left of its correct position in the text, shift rules would be able to simply shift all predictions one position to the right.

Rule induction

Induction of the tagging rules is achieved as follows. For each positive example the algorithm builds a set of *initial rules* from it, generalises (or specialises) those rules and keeps the k best generalisations (specialisations) derived in this way. In $(\mathtt{LP})^2$, top-down specialisation is used. To build the initial rules, a tag in the training corpus is selected and a text window of w tokens to the left and w tokens to the right is extracted, which includes additional information about the tokens (typically added by running a sequence of NLP tools, as seen in chapter 6). Each piece of information stored in the $2w$ token window is transformed into a condition of the type `[feature class = feature value]` in the rule left-hand side, like in the example rule above. The initial rules derived from this window consist of rules with a single condition. An iterative cycle starts with the initial rules at iteration 0

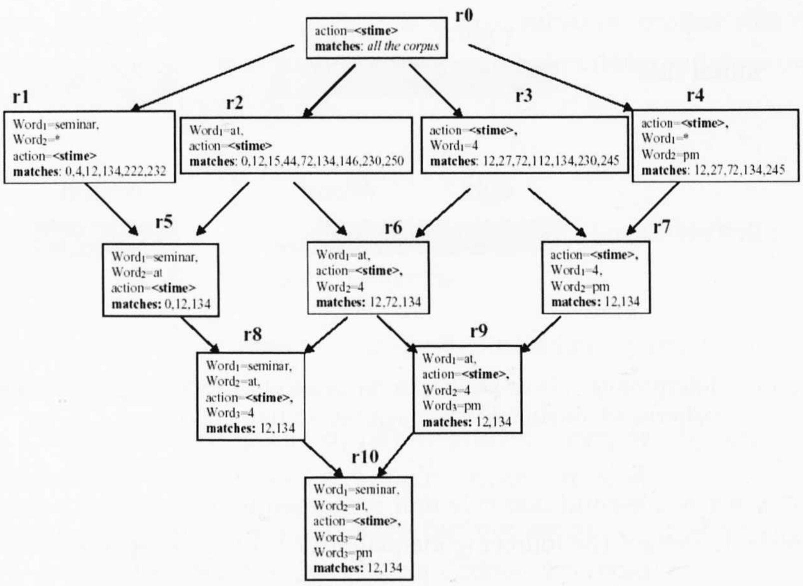


Figure 6: The search space produced by rule specialisation forms a *galois* lattice. Each node in the lattice is a candidate rule that is scored against the positive/negative examples in the training corpus.

and specialises rules at iteration i by introducing one single additional condition to generate the rules for iteration $i+1$. At the end of each iteration, the k best specialisations, according to some rule scoring criterium (in $(LP)^2$, *precision* is used), are kept and the retained rules become part of the best rules pool R. Furthermore, when a rule enters the best rules pool, all of the examples covered by the rule are removed from the positive examples set, i. e., they will no longer be used to derive new rules. $(LP)^2$ is thus a sequential *covering* algorithm. Rule induction proceeds iteratively until the set of positive examples is empty.

The search space explored by $(LP)^2$ for one given tag in the text is illustrated in Figure 6. All possible combinations of specialisations of the set of conditions in a rule are tried in order to form the candidate rules. The size of the full space is 2^{f2w}, where $2w$ is the size of the token window, and f is the total number of feature classes attached to those tokens, e. g., lemma, gazetteer, orthography. Clearly, considering all possible candidate rules is computationally intractable even for modest values of f and w. Thus, pruning of the search space plays a very fundamental role in the algorithm's specialisation procedure.

Pruning the search space

Let $i \in R$ be the initial rule and $d = s(i, c) \in R$ a rule that is derived from i by applying some specialisation function $s : R \times R \to R$, where

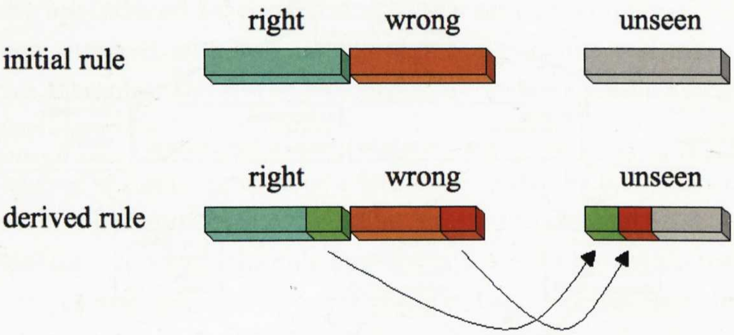


Figure 7: Interpreting rule specialisation in terms of right, wrong and unseen patterns of the rule matched against the training corpus.

$c \in R$ is a single-condition rule that is merged with the initial rule to restrict it further. The following inequalities hold for rule specialisation:

$$tp_d \leqslant \min(tp_i, tp_c)$$

$$fp_d \leqslant \min(fp_i, fp_c)$$

$$fn_d \geqslant \max(fn_i, fn_c)$$

These can be interpreted as shown in Figure 7, in terms of tp (right), fp (wrong) and fn (unseen) patterns applied over the corpus. Essentially, a derived rule d moves some of the true positives and some of the false positives of i into the the set of false negatives. What is hoped thus is that less true positives are moved than false positives, leading to a rule that discriminates better.

Pruning based on a quick check

Based on this observation, $(LP)^2$ prunes the search space by checking whether the derived rule d reduces the set of false positives :

- if $fp_d = fp_i \wedge tp_d < tp_i$ then d does not improve i with respect to false positives while actually reducing the set of true positives. Rule d is thus discarded immediately, meaning that its whole search sub-tree is pruned.
- if $fp_d = fp_i \wedge tp_d = tp_i$ then d does not improve nor worsens i. Depending on the bias of the algorithm, it is possible to remove i, remove d, or to maintain both. In the case of $(LP)^2$, d is discarded.
- otherwise, accept d into the set of accepted rules R.

The search space can be further pruned by performing subsumption tests between the rules in the set of accepted rules R and a candidate derived rule d. That is, if $\exists r \in R : \text{subsumes}(r, d)$, then d is eliminated from the search space[3]. A rule r_1 is said to subsume another rule r_2 when r_2 matches the same entity mentions that r_1 does, and does not add to the false positives, i. e., :

Pruning using subsumption tests

$$\text{subsumes}(r_1, r_2) = \begin{cases} \text{true,} & \text{iff } fp_{r_1} \leqslant fp_{r_2} \wedge \\ & \{p : \text{matches}(r_1, p)\} \supseteq \{p : \text{matches}(r_2, p)\} \\ \text{false,} & \text{otherwise} \end{cases}$$

where matches(r, p) returns true if rule r matches a pattern p in the corpus and false otherwise.

Finally, the search space may be also pruned "unsafely", by resorting to heuristics. In $(LP)^2$, the following heuristics are used:

Pruning using heuristics

SCORE THRESHOLD a rule is kept, but not specialised further, if its score (precision) is above a user-defined threshold.

MINIMUM MATCHES a rule is eliminated immediately if it fails to match at least a user-defined amount of entity mentions in the corpus. This is similar to the the classic *apriori* principle [3] in itemset mining.

2.5.2 *Single-Object Labelling Approaches*

A simple statistical classification model for EE consists in independently assigning an entity class y_i to each token x_i according to features derived from x_i and its neighbours in x, disregarding potential correlations between labels attached to nearby tokens. The set of possible feature functions $f \in \mathcal{F}$ forms a *feature space* in $\mathbb{R}^d$, where d is the dimensionality of the space, and normally $d = |\mathcal{F}|$. Each object to classify is represented in this space as a vector $x \in \mathbb{R}^d$. This simple model can be carried out with off-the-shelf state-of-the-art classifiers, such as a logistic classifier or a Support Vector Machine (both used in chapter 4).

Under token or boundary models (recall subsection 2.2.2), the extraction of features that characterise a token or boundary at position i in the text can be viewed as a function $f : (i, t, y_i) \mapsto \mathbb{R}$ that takes as argument the position i, a sequence of tokens t and a label y_i attached to the token at position i. The function is thus allowed to look at neighbouring tokens, taking into account the position where the features occur

Features generation

[3] Note that the reverse should also be performed — any rule $r \in R$ subsumed by d should also be removed. In practice, r may already have been specialised, so the test is not performed in $(LP)^2$.

relative to position i. The features generated in this way consist in the combination of their value with their position. For instance, considering the 2-neighbourhood around the token "Ralph" in the sentence from the example in Figure 3

> [...] speak with Ralph Hill, while he is [...]

would generate the following feature encoding (considering word, orthography and part-of-speech features):

```
token.-2.speak token.-1.with token.1.Ralph token.2.Hill
pos.-2.VB pos.-1.IN pos.1.NNP pos.2.NNP
orth.-2.low orth.-1.low orth.1.firstcap orth.2.firstcap
```

Several applications of SVM in the context of the Entity Extraction task can be found in the literature. Of particular relevance to the work in this thesis are the state-of-the-art SVM-based systems designed by Finn and Li et al., both of which follow the boundary model.

SVM applied to ER

Finn [59] introduced a variant to the usual boundary classification approach which makes use of a two-level ensemble of classifiers, which play a role similar to tagging and contextual rules in the $(LP)^2$ algorithm described earlier. The approach takes advantage of the fact that high-confidence predictions for the start of an entity are an indication of its end in the nearby text, and vice-versa. In the first level, their approach uses high-precision classifiers so as to spot individual start or end of fragments. On the second level, their approach uses high-recall classifiers, but restricted to the vicinity of the individual start/ end already predicted by the first level classifiers. Their SVM-based system implementing the multi-level approach to boundary classification, called ELIE, shows state-of-the-art performance on the standard datasets overviewed in subsection 4.1.1.

Li et al. [110] describe in great detail their system, called GATE-SVM. One distinctive feature of the system is that it uses a variant of SVM, the SVM with *uneven margins*, which the authors show to be particularly helpful for imbalanced datasets, that is, datasets whose distribution of labeled examples among classes is far from uniform. Another interesting feature of GATE-SVM is that it uses a weighing scheme for the token features according to distance of the token to the boundary in the text. The system also obtained state-of-the-art results on the same standard datasets.

2.5.3 *Sequential Labeling Approaches*

The approaches to EE reviewed so far do not incorporate into the learning model the potential interdependencies between the labels of nearby tokens. $(LP)^2$ proposed contextual rules to achieve that, while in the case of SVM previous label(s) can be added as a feature of the learning example, but there is no account in the model itself for the sequence of labels. This has led to a number of different models for sequence labeling, starting with Hidden Markov Model and evolving onto Conditional Random Fields. These models use probability theory, specifying a probability distribution to select the most likely class y for a given observation x.

Sequence labeling

An Hidden Markov Model (HMM) [138] is a probabilistic finite state automaton for modeling sequential data. It defines a set of (hidden) states, with transitions between them. Associated with each state is a probability distribution over the possible transitions from that state to another state. What is more, states can output symbols, one at a time, based on a symbol emission probability distribution. In the context of EE, the states of an HMM represent the entity classes to be extracted from the text — namely, there would be states for each entity class, and a background state for "no class" —, while the symbols would be the tokens in the text. The state transition and symbol emission probabilities of an HMM, termed the model parameters, can be estimated from the training data.

Hidden Markov Models

Given a sequence of symbols, e. g., a text sentence, and the HMM that is assumed to have produced it, the typical question that an HMM inference algorithm is required to answer is to find the sequence of states that is most likely to have generated that sequence of symbols. Since for EE the states are the entity classes, this is the same as saying that, given an input sentence, the HMM finds the most likely sequence of entity labels that generated the sentence. Such sequence of labels will be the background state everywhere except where the entity mentions lie, in which case the state becomes the entity label that corresponds to the mention. Entities can be then extracted by assembling contiguous sequences of tokens which are labeled with the same entity class. This is precisely what is done in [63].

HMM applied to ER

What makes HMMs attractive is their solid mathematical foundation and the fact that the problem of finding the most likely sequence can be solved using an algorithm that runs in time linear with the number

Inference with an HMM

of observed symbols (i. e., size of the text in EE), the Viterbi algorithm [138]. Given m labels and a sequence x of length n, there can be $O(m^n)$ possible labelings of $\mathbf{x}$. This exponential complexity is cut down to $O(nm^2)$ with the Viterbi algorithm.

Discriminative vs. Generative approaches

HMMs belong to a class of approaches called *generative*, as opposed to *discriminative*, to which SVMs belong. HMMs can be seen as an extension to the well-known Naïve Bayes generative model [115] (mostly used in document categorisation) for sequentially structured data. Generative approaches try to model the underlying (unknown) joint probability distribution that generates the data rather than simply trying to model the way in which the classes can be discriminated (e. g., with an hyperplane as in SVMs). The joint probability of text tokens (observed symbols $\mathbf{x}$) and sequence of entity labels (hidden states $\mathbf{y}$) modeled by an HMM is given by

$$\Pr(\mathbf{x}, \mathbf{y}) = \prod_{i=0}^{n} p(y_i | y_{i-1}) p(x_i | y_i). \tag{2.10}$$

However, in EE and many other tasks, the problem "only" lies in finding the hidden state sequence, not both symbols and states. Therefore, a major drawback of generative models when compared to discriminative models is that generative models try to solve a harder problem than what is actually required by many applications, which unfortunately tends to lead to intractable problem formulations that require introducing simplifying assumptions that, in turn, lead to worse results. In fact, to keep inference tractable, HMMs use the "output independence assumption", which stipulates that the current observation, given the current state, is independent of previous observations. Since in EE observations correspond to tokens, this assumption is unrealistic most of the time.

Maximum Entropy Markov Models

Maximum Entropy Markov Models (MEMM) [116] were introduced to solve the limitations of HMMs. An MEMM is a discriminative approach, because it models the conditional probability of a state given an observation sequence rather than the joint probability. It allows for arbitrary, non-independent features on the observation sequence, and for the transition probability to depend on past and future observations. MEMM uses a *per*-state exponential model for the conditional probabilities of next states given the current state, which is given by

$$P(y|x) = \frac{1}{Z(x)} e^{\sum_{i=1}^{m} \lambda_i f_i(x,y)}, \tag{2.11}$$

where f_i are binary valued functions that test features in $\mathbf{x}$, λ_i are Lagrange multipliers introduced for the purposes of numerical optimisation (see [142] for details), and Z is a normalising constant, defined as $Z(x) = \sum_{y \in \mathcal{Y}} e^{\sum_{i=1}^{m} \lambda_i f_i(x,y)}$.

Conditional Random Fields

Linear Chain Conditional Random Fields (CRF) [102] can be considered to be the state-of-the-art approach to sequence labelling. CRFs have been proven to be very successful in EE, for example see Sutton and McCallum[159]; and also when applied to EE on documents from the biomedical domain [152, 118]. Just as a HMM is a sequential extension to the Naïve Bayes (NB) model, CRF can be understood as a sequential extension to MEMM model. While CRFs make the same assumptions as HMM on the dependencies among the class variables, no assumptions on the dependencies among observation variables need to be made, like in the case of the MEMM. Figure 8 illustrates the relationship between the four types of models overviewed in this section.

CRFs were motivated by a well-known problem with MEMMs, that of biasing toward states with fewer outgoing transitions. The reason for this behavior stems from the fact that the same probability mass is allocated for modeling the labeling decision at each position in the sequence of labels. This problem was solved in CRF by considering a single probability distribution that models the joint probability of a label sequence, conditioned on a sequence of observations. Therefore, some transitions may contribute more than others to the overall score, depending on the corresponding observations. The single exponential model for the joint probability of the entire sequence of labels given the observation sequence is given by

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} e^{\sum_{j=1}^{n} \sum_{i=1}^{m} \lambda_i f_i(y_{j-1}, y_j, x, j)}, \tag{2.12}$$

where $Z(x) = \sum_{y \in \mathcal{Y}} e^{\sum_{j=1}^{n} \sum_{i=1}^{m} \lambda_i f_i(y_{j-1}, y_j, x, j)}$.

Learning the parameters of a Conditional Random Fields can be cast as an optimisation problem. The function being optimised is concave, like for SVM, thus a global maximum can be found efficiently using standard procedures, such as gradient-based methods.

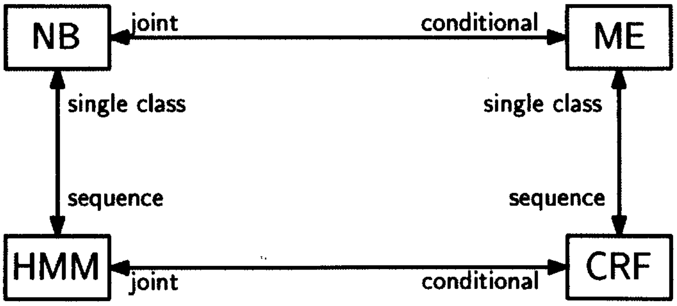


Figure 8: The relationship between Naïve Bayes, Hidden Markov Model, Maximum Entropy Markov Models and Conditional Random Fields. Source: [142].

2.6 SUPERVISION REQUIREMENTS

Labelled examples generation

In order to obtain labelled examples required to develop text mining models (and also to compute their performance against gold standards), domain experts who are able to understand the corpora are typically commissioned to annotate it, i. e., attach the aforementioned labels to the text documents, text segments, tokens or boundaries, as discussed earlier. This is done with the aid of dedicated software tools, such as the one described in [39], and semantic annotation formalisms, such as the one presented in chapter 6.

The ML methods introduced in the previous sections are intended to model the TC and IE problems in such a way as to account for different assumptions about, and representations for, the input documents. In addition to that, an orthogonal aspect when choosing or designing ML methods concerns its supervision requirements, i. e., how capable is the method to deal with the availability of labelled and unlabelled data (or lack thereof).

2.6.1 *Semi-supervised Learning*

Unlabelled data is often available at the time of classifier induction, depending on the problem domain. If this is the case, a wealth of literature [175] on semi-supervised learning approaches is available that focuses on exploiting such unlabelled data in various tasks. In particular, several variants of semi-supervised SVM were derived from the supervised formulation, see for example [14].

Various machine learning techniques try to solve the so-called *cost-sensitive learning* problems by making strong simplifying assumptions. In active learning [162], for instance, labels are assumed to be expensive, but the learner may ask an oracle to reveal a label for selected unlabelled examples. Active feature acquisition [119] makes the assumption that obtaining features is expensive, but the learner may identify examples for which complete information is most informative in order to classify a given test instance. Inductive transfer learning and domain adaptation methods, such as the one studied in section 5.2, work under the assumption that training data for a particular task or domain is expensive, but data from other related domains may be cheaper.

Cost-sensitive learning

In the case of semi-supervised learning, the assumption made is that class labels are expensive to obtain, while features are implicitly assumed to have zero cost. Semi-supervised learning consists of a family of algorithms which can effectively combine unlabelled data with labelled data in the learning process by exploiting the manifold structure (also called cluster structure) in data [12, 19, 176, 177]. This is achieved by assuming that

Cluster assumption

- nearby points are likely to have the same label; and
- points on the same structure (such as a cluster or a submanifold) are likely to have the same label.

Note that the first assumption is local, while the second one is global. The cluster assumption therefore implies considering both local and global information contained in the dataset during learning.

2.6.2 *Transductive Learning*

Semi-supervised methods can be categorised into two types: transductive and inductive. The goal of *transductive learning* is merely to estimate the labels for the unlabelled data, whereas *inductive learning*, which was adopted in chapter 4, attempts to induce a decision function which has a low error rate on the whole sample space.

Transductive vs. inductive learning

In other words, in a transductive learning setting, a single optimization problem is set up involving all training and test instances, and the solution of that optimization problem yields labels for the test instances. In such setting, the test instances provide evidence about the distribution of the data, which may be useful when the labelled data is limited and the distribution of unlabelled data is informative about the location of the decision boundary.

As a note, these two kinds of learning relate to two different high-level patterns of interaction between the domain expert and the IE systems:

- transductive learning supports well on-demand patterns of interaction – for example, domain experts quickly assemble a set of objects (documents, entities, relations, and so on), give a few classification examples and expect the system to decide on-the-fly about the labels for the remaining objects; whereas
- inductive learning is more suited to support a regular pattern of interaction – for example, domain experts classify objects of interest in the context of their normal workflow, and expect the system to help spotting new objects of interest in the future.

2.6.3 *Feature Labels versus Instance Labels*

Feature labelling

In some problem domains, labelled data are available in the form of labelled features rather than labelled instances (which have been considered so far in this chapter). This is designated the *feature labelling paradigm*. The main advantage of this paradigm resides in the fact that labelling features is less expensive than labelling documents. It is both easier and quicker for domain experts to identify a small set of features that are globally expected to be positively correlated with a given class, instead of examining a set of instances in detail.

Generalised expectation

In this thesis, the *generalised expectation criteria* method [114, 50] is adopted in chapter section 5.2 to translate the knowledge about which features from the source domain are expected to apply also in the target domain into constraints on model expectations for certain word-class combinations. A Generalised Expectation (GE) criterion is a term in a parameter estimation objective function that assigns scores to values of a model expectation. Let x be the input, y the output, and θ the parameters for a given model. Given a set of unlabelled data $\mathcal{U} = \{x\}$ and a conditional model $p(y|x;\theta)$, a GE criterion $G(\theta;\mathcal{U})$ is defined by a score function V and a constraint function $G(x,y)$:

$$G(\theta;\mathcal{U}) = V(E_{\mathcal{U}}[E_{p(y|x;\theta)}[G(x,y)]]).$$

The GE formulation is generic enough to enable exploring many different choices of score functions and constraint functions. For example, in the work presented in this thesis I maximize the GE term together with an entropy regularization term in the objective function, although this can be easily combined with an empirical loss term to form a composite objective function that takes into account labelled instances as well.

2.7 ONTOLOGY AND SEMANTIC ANNOTATION

A good part of the exposition in this thesis related to semantic annotation (and, equivalently, labelled data) relies on the concept of *ontology*. Additionally, the work presented in chapter 6 relies on basic knowledge about RDF and OWL, two ontology languages adopted as standards for encoding modern ontologies, and some knowledge about DOLCE, CSO and COMM, three top-level ontologies that provide the backbone for the semantic annotation formalism proposed in this thesis. This section briefly reviews these concepts and artefacts. It may be skipped by the reader familiar with them.

Notation used

The Unified Modeling Language (UML) [89] will be used throughout to depict the main concepts and associations in the ontologies used in the thesis. In the text itself, concepts and associations will be written in `sans serif` and will be labelled in a namespace-like manner. Namespace prefixes indicate the ontology where those concepts and associations are defined. If no namespace is used, they are defined in ONIX, the semantic annotation formalism presented in chapter 6.

Ontology

In philosophy, ontology studies the nature of being and existence. The term "ontology" is derived from the Greek words "onto", which means "being", and "logia", which means "written or spoken discourse". Computer scientists extended previous notions of ontology into a new interpretation, which can be best expressed as "a specification of a conceptualisation" [72]. In computer science and information science, knowledge reuse is facilitated by the use of explicit ontology, i. e., knowledge encoded into software systems [164] in some language.

2.7.1 *Ontology Languages*

Ontology languages have been adopted as standards in the past few years and increasingly more ontologies characterising various domains are becoming available in those languages. The ontologies presented in this thesis is encoded in OWL. Let us briefly review the main concepts behind these languages.

The Resource Description Framework (RDF) [99] is a W3C standard that defines a simple model for describing resources and the relations between those resources. RDF makes no a priori assumptions about a particular application domain or the associated semantics.

Resource Description Framework

The RDF model consists of resources, properties and statements. A *resource* is anything that can be named by a Universal Resource Identifier (URI) [15], which includes not just things on the Web (such as pages, parts of pages or collections of pages) but also tangible things, provided that an URI scheme can be associated to them. Systems can define some concept and each use a different (unique) URI to name it to avoid clashes. However, systems agreeing on a common concept will use the same URI and effectively share semantics. Note that by adopting URIs, RDF avoids the problem of polysemy, i. e., using different terms to denote the same resource.

A *statement* is basically a resource-property-value triple, which defines a binary relationship between two resources (the *subject* and the *object*) using a resource *property* (the *predicate* in RDF terminology). The triple notation is commonly written as $A(O, V)$, meaning that an object O has an attribute A with value V. The RDF model is also equivalently represented as a labelled directed graph — where the nodes represent resources and the arcs represent the properties of those resources — and in XML serialized form.

In RDF, statements are also resources. The RDF model offers the predefined resource `rdf:statement` and the predefined properties `rdf:subject`, `rdf:predicate`, and `rdf:object` to allow *reifying* a statement as a resource. Any RDF statement can be the object or value of a triple, which in the graph representation means that graphs can be nested as well as chained (forming an *hypergraph*). The RDF model also defines some other meta-level constructs, such as container types for describing collections of resources — bags, sequences, and alternatives. A bag is unordered, a sequence is ordered, and an alternative is a set of choices.

To allow for the creation of controlled, sharable, extensible vocabularies (often named *schemas*), RDF Schema (RDFS) [27] was later introduced. RDFS adds an additional layer on top of RDF to integrate some simple notion of *classes*, class *inheritance*, properties and property inheritance. RDFS lets developers define a particular vocabulary for RDF data and specify the kinds of objects these attributes can be applied to. Phrasing the role of RDFS using knowledge engineering terminology: RDFS defines a simple ontology that particular RDF documents may be checked against to determine consistency.

RDF Schema

The RDFS specification defines a number of classes and properties that have specific semantics. The `rdfs:Class` and `rdfs:Property` classes allow a resource to be typed as a class or property respectively, and properties can be used to describe these classes and properties. RDF objects can be defined as *instances* of one or more classes using the `rdf:type` property. The property `rdfs:subClassOf` essentially states that one class is a subset of another — allowing schema designers to build taxonomies of classes for organising their resources — while the property `rdfs:subPropertyOf` does the same for properties. Both `rdfs:subClassOf` and `rdfs:subPropertyOf` are transitive and both are supposed to be cycle-free (i. e., a class can neither be a subclass of itself nor a subclass of its own subclasses). Constraints on properties can also be specified using the `rdfs:domain` and `rdfs:range` constructs.

The Web Ontology Language (OWL) [11] extends the basic fact-stating ability of RDF and the class- and property-structuring capabilities of RDFS. OWL is more expressive than RDFS. Besides declaring classes and organising them in a subsumption hierarchy, with OWL the ontology engineer can additionally specify classes as logical combinations — *intersections, unions,* or *complements* — of other classes, or as *enumerations* of specified objects; and besides declaring properties and organising them in a subproperty hierarchy, the ontology engineer can additionally state that a property is *transitive, symmetric, functional,* or is the *inverse* of another property. Moreover, *equivalence* statements can be made on classes and on properties, *disjointness* statements can be made on classes, and *equality* and *inequality* can be asserted between objects. Finally, OWL allows declaring restrictions on properties that are local to a class. A common use of this feature is to define classes where a particular property is restricted so that all the values for the property in instances of the class must belong to a certain class or value range.

Web Ontology Language

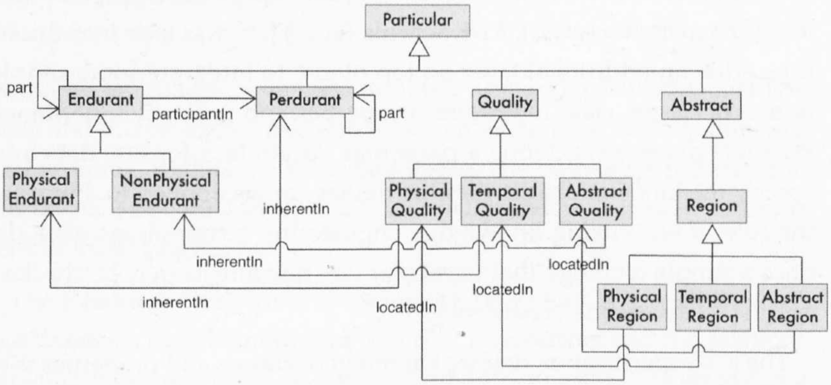


Figure 9: A UML diagram view of the DOLCE top-level ontology. Source: [66].

2.7.2 *Foundational Ontologies*

The Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [66] is the foundational ontology used as a modeling basis for the ONIX ontology presented in this chapter 6. A *foundational* ontology is, in essence, an axiomatic theory about high-level domain-independent classes, such as *object*, *attribute*, *event*, *spatial* and *temporal* connections, and so on. DOLCE belongs to the WonderWeb library of foundational ontologies[4] and has been successfully applied in different domains, for example in biomedicine [64].

The DOLCE Foundational Ontology

Since foundational ontologies provide a predefined set of well-founded, general-purpose, and reusable semantic concepts and their relations, they constitute a good starting point for building new ontologies. Notably, a foundational ontology defines ontology *design patterns*. Ontology patterns capture, in the form of ontology, re-occurring modeling needs, e. g., location in space and time. These design patterns can be applied to achieve high quality design and modeling consistency, and also enhanced interoperability, particularly with ontologies based on the same foundational ontology. Other examples of foundational ontologies include the Suggested Upper Merged Ontology (SUMO) [128] and Cyc [107].

DOLCE categorises entities into four classes: Endurants, Perdurants, Qualities and Abstracts. A crucial distinction in DOLCE is the one between enduring and perduring entities. *Endurants* are entities which

[4] `http://wonderweb.semanticweb.org/`

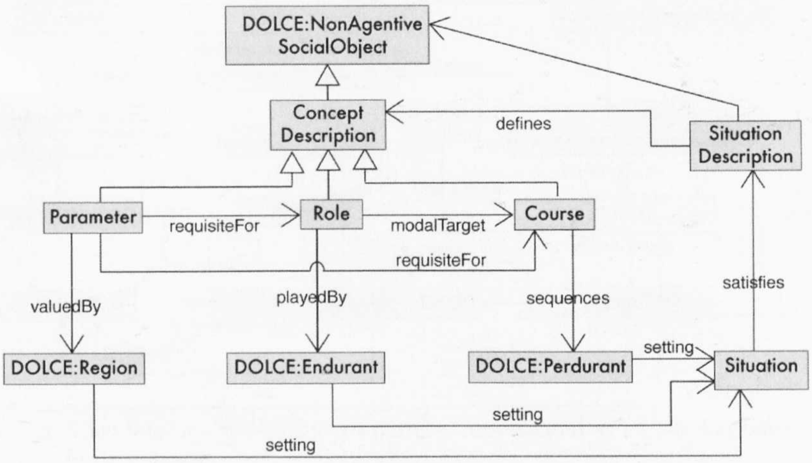


Figure 10: The Descriptions & Situations (DnS) ontology design pattern, represented as a UML diagram. Source: [129].

exist in time, e. g., person, house, theory, while *perdurants* are entities which happen in time, e. g.a seminar, a party, a football match. The main relation between endurants and perdurants is that of *participation*: an endurant "lives" in time by participating in a perdurant, e. g.a person "participates" in his or her life. All entities have *qualities* such as colour, shape, size, and so on. *Spatial* locations, a special kind of physical quality, and *temporal* qualities encode the spatio-temporal attributes of objects or events. Finally, *abstracts* do not have spatial or temporal qualities and they are not qualities themselves. An example of this are *regions*, which are used to encode the measurement of qualities in some metric or conceptual space. The UML diagram view of DOLCE is sketched in Figure 9.

Extensions to DOLCE

As extensions to DOLCE, and included in the same library of foundational ontologies, there are three ontological design patterns, which were used extensively in the design of the ontology presented in chapter 6:

DESCRIPTIONS & SITUATIONS A common modeling need in many domains is to formalize contextual knowledge. The DnS ontology [65] is, in essence, an ontology of *contexts*, providing a principled approach to context reification through a clear separation of *states-of-affairs* and their interpretation based on a non-physical context, called a "description". DnS thus defines a *situation description* template and reification rules for the main classes in

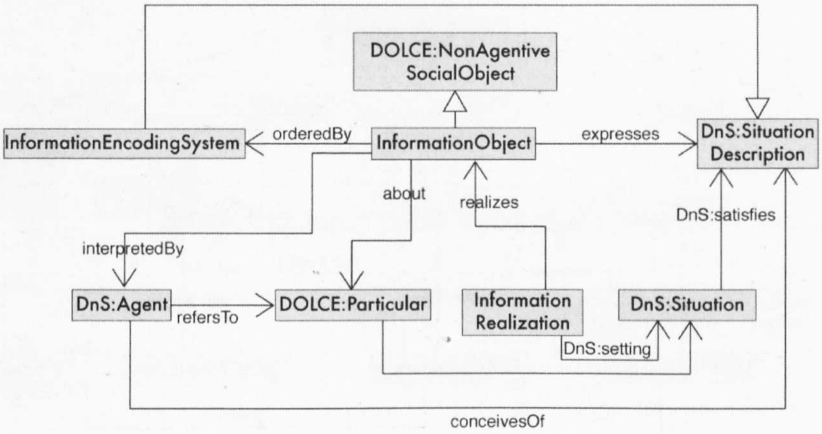


Figure 11: The Ontology of Information Objects (OIO) pattern, represented as a UML diagram. Source: [129].

the DOLCE foundational ontology. This is shown in Figure 10. DnS results to be a theory of ontological contexts because it is capable of describing various notions of context — physical and non-physical situations, topics, provisions, plans, assessments, beliefs, and so on — as first-order entities. For the purposes of this chapter, we are particularly interested in using context to model different *views on data and computational processes*, e. g., input/output data, intermediate process. DnS builds on DOLCE in the following way: a `DnS:Situation` is a newly introduced top class; a `DnS:Description` is a non-physical *endurant* that may be satisfied by a `DnS:Situation` (and is disjoint from it); the *setting* for the `DnS:Situation` is an entity in DOLCE (in grey), be it a *region*, an *endurant* or a *perdurant*.

ONTOLOGY OF INFORMATION OBJECTS DOLCE includes an ontology design pattern that enables distinguishing between entities in the real world and entities that live in *information systems*, called OIO [66]. The main entities defined by OIO are `OIO:InformationObject` and its respective `OIO:InformationRealisation`. An information object is a spatio-temporal entity of abstract information as described in Shannon's communication theory (e. g., a computer program), while an information realisation is an entity that realizes the information object (e. g., the program binary code stored in a DVD between some specified sectors). OIO builds on DOLCE and DnS in the following way: an `OIO:InformationObject` is a non-physical `DOLCE:Endurant` that expresses a `DnS:Description`.

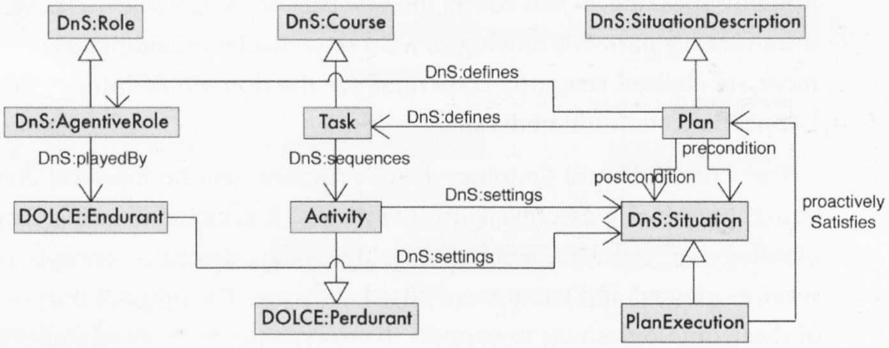


Figure 12: The Ontology of Plans (OoP) pattern, represented as a UML diagram. Source: [129].

The description satisfies a `DnS:Situation`, namely the situation that works as the setting for the `OIO:InformationRealisation` that realizes that `OIO:InformationObject`. The information object holds information about some `DOLCE:Particular`. In Figure 11, the Ontology of Information Objects is depicted.

ONTOLOGY OF PLANS The Ontology of Plans (OoP) [66] characterises *planning* concepts. It is an ontology design pattern intended to model plans at an abstract level, providing a framework for more specific ontologies that characterise particular kinds of plans (e. g., personal, social, computational), and it forms the basis for modeling software component/service workflows, which will be needed later. The main entities defined by OoP are `OoP:Task`, `OoP:Plan`, and `OoP:PlanExecution`, all appropriately grounded on DOLCE and DnS entities. OoP builds on DOLCE and DnS in the following way: an `OoP:Plan` is a special kind of `DnS:Description`, which defines *roles* to be played by `DOLCE:Endurants` in an `OoP:Task`; the `OoP:Task` entity sequences an `OoP:Activity` (a specialisation of a `DOLCE:Perdurant`), whose setting is the `OoP:Plan`. Figure 12 shows the Ontology of Plans.

2.7.3 *Core Ontologies*

In the design and development of the Ontology of Information Extraction, two core ontologies were used as a modeling basis: the Core Ontology of MultiMedia (COMM) and the Core Software Ontology (CSO).

Roughly speaking, COMM covers the *data* part of ONIX, while CSO covers the *processing* part. The ontologies were extended by providing new and more specialised semantic constructs for the domain of Information Extraction from multimedia data.

The Core Software Ontology

The Core Software Ontology [129] provides generic top-level constructs to formally describe software systems. It is included in the Core Ontology of Software, a set of ontologies that describe concepts of service-oriented and component-based software. The original purpose of these ontologies was to support the maintenance of server applications, but their generality translated into their adoption as a modeling basis for many other ontologies [103, 8, 153]. The Core Ontology of Software is modularized into three major sub-ontologies — the Core Software Ontology, Core Ontology of Software Components and the Core Ontology of Services — and rooted in the upper-level ontologies overviewed in the previous subsection. For the purposes of the work presented in this thesis, we are mainly interested in the Core Software Ontology (CSO).

The most fundamental concepts required to model both software components and Web services are formalised in CSO, including concepts such as *software, data, users, access rights* and *interfaces,* that is, entities that live in the *computational domain.* In CSO, `CSO:Software` is characterised as an `OIO:InformationObject` and is said to express an `OoP:Plan`. The plan that it expresses consists of an arbitrary number of `CSO:ComputationalTasks` that sequence `CSO:ComputationalActivities`, specialising `OoP:Task` and `OoP:Activity`, respectively. The ontology additionally introduces the concept of `CSO:ComputationalObject`, a specialisation of the concept `OIO:InformationRealisation`. The computational object can be, for example, the realisation of `CSO:Software`, whose execution leads to `CSO:ComputationalActivities`. The other main concept modeled by CSO is that of `CSO:Data`. Like `CSO:Software`, `CSO:Data` is viewed as a special kind of `OIO:InformationObject`, with the difference that they do not express an `OoP:Plan`.

The Core Ontology of MultiMedia [8] is an ontology and API that provides constructs to semantically describe *multimedia* artifacts available on the Web. The ontology was originally created to tackle a number of aspects that make the MPEG-7 standard [71] cumbersome and ineffective. For example, MPEG-7 does not support machine-processable semantic annotations of the image subject matter, since the lightweight MPEG-7 proprietary controlled vocabularies used to tag the images are incom-

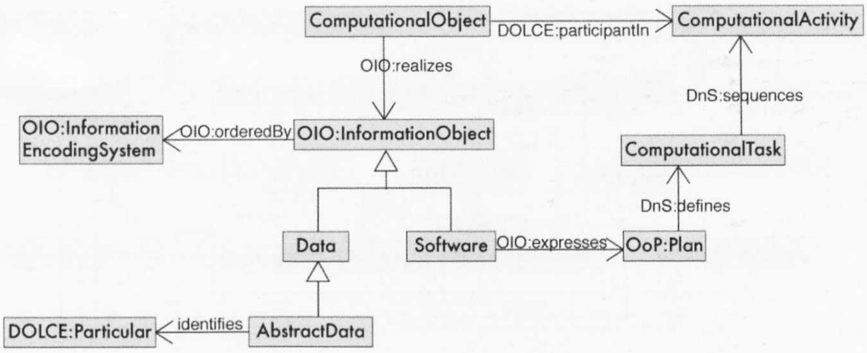


Figure 13: The main concepts in the Core Software Ontology (CSO), represented as a UML diagram. CSO introduces fundamental concepts for modeling computational systems, particularly the concepts of *software* and *data*. Source: [129].

The Core Ontology of Multimedia

patible with Semantic Web ontologies. COMM, on the other hand, is modeled using a sound ontological engineering approach, which builds upon the foundational ontologies reviewed in the previous subsection, and is formalised in OWL (described earlier). To leverage annotators' experience with MPEG-7, COMM covers all the MPEG-7 descriptors and uses the same naming conventions as in that standard.

The ontology formalises the basic concepts of `COMM:DigitalData`, `COMM:MultimediaData` and `COMM:Algorithm`, all of which rely on the Ontology of Information Objects. `COMM:DigitalData` is viewed as a kind of `OIO:InformationObject` used for communication between machines. Specialising the DnS and OIO ontologies, COMM introduces the concept of `COMM:StructuredDataDescription`, a kind of `DnS:Description`, that defines meaningful labels for the information in `COMM:DigitalData`. Such information is characterised by `DOLCE:AbstractRegions` (encompassing scalars, matrices, strings, rectangles, polygons, and so on). Since in DOLCE regions are described by parameters, COMM defines `COMM:StructuredDataParameters`, a kind of `DOLCE:Parameter`, to which `DOLCE:AbstractRegions` assign values.

The `COMM:MultimediaData` concept is a specialisation of the concept `COMM:DigitalData`, and it is in turn expected to be specialised to specific media types, e. g., `ImageData` or `TextData` — in fact, this is precisely what has been done in the design of ONIX. Finally, a `COMM:Algorithm` is seen as a method applied to solving computational problems, and as such it is a specialisation of `OIO:Method`.

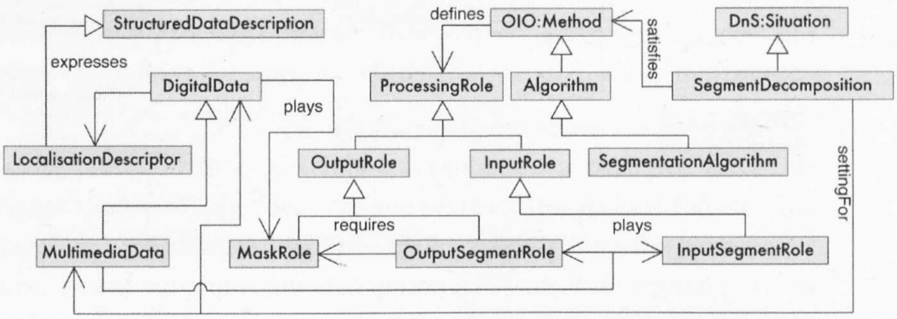


Figure 14: The *Decomposition* pattern in the COMM ontology, represented as a UML diagram. Source: [8].

The ontology design patterns in COMM

Besides the basic concepts just introduced, three ontology design patterns are included in the COMM ontology. These are:

DECOMPOSITION The decomposition pattern, depicted in Figure 14, handles the description of a multimedia document's structure. COMM views a decomposition of a `COMM:MultimediaData` entity as a `DnS:Situation` that satisfies a `DnS:Description`. For example, the situation can be "splitting a text into sentences" and the description can be the algorithm that performs that. An important specialisation of the concept `DnS:Role` in COMM consists in the `COMM:InputSegmentRole` and `COMM:OutputSegmentRole` concepts. Concretely, a `COMM:InputSegmentRole` expresses that some `COMM:MultimediaData` entity plays the role of an input segment in some situation. The need localise segments within the input media leads to the definition of a `COMM:MaskRole`, to be played by one or more `COMM:DigitalData` entities which express one `COMM:LocalisationDescriptor` (the localisation descriptors are ontological versions of MPEG-7 region locators for defining regions in an image).

CONTENT & MEDIA ANNOTATION This pattern, depicted in Figure 15, allows expressing the attachment of annotations to `COMM:Media` or to `COMM:MultimediaData` (the media are realised by multimedia data). In COMM, a `COMM:Annotation` is a `DnS:Situation` that represents the "state-of-affairs" of the `COMM:MultimediaData` and all the metadata attached to them. Metadata are modeled as `COMM:DigitalData` that play a `COMM:AnnotationRole`. There is a corresponding `COMM:AnnotatedMediaRole` for `COMM:Media`.

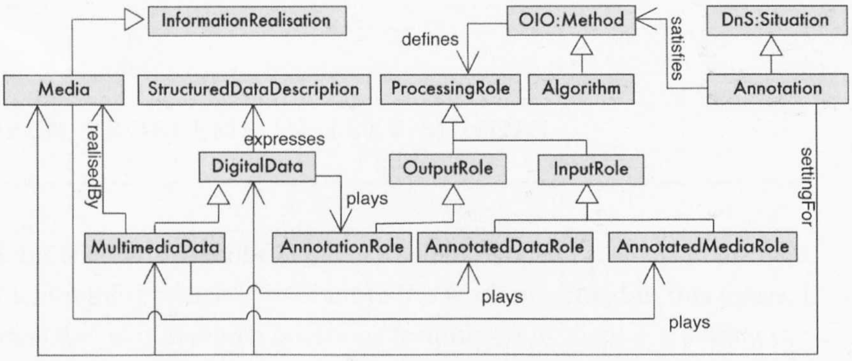


Figure 15: The *Content & Media Annotation* pattern in the COMM ontology, represented as a UML diagram. Source: [8].

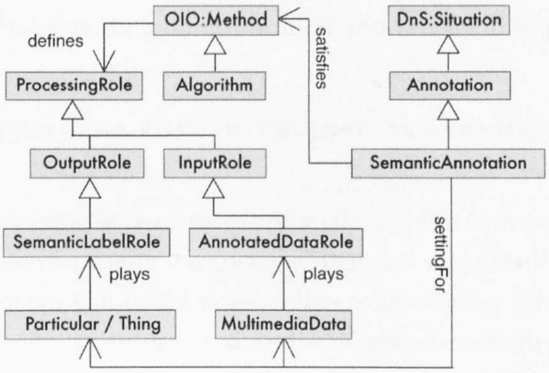


Figure 16: The *Semantic Annotation* pattern in the COMM ontology, represented as a UML diagram. Source: [8].

SEMANTIC ANNOTATION The semantic annotation pattern, depicted in Figure 16, provides support in COMM for associating multimedia descriptors with domain concepts defined in some domain ontology. The range of the association is generically expressed by a `OWL:Thing` or a `DOLCE:Particular` in the pattern. The association is done through a `COMM:SemanticAnnotation` (a `DnS:Description` that satisfies the `OIO:Method` applied to generate it), which specifies that the annotated `COMM:MultimediaData` plays an `COMM:AnnotatedDataRole` and the referred `OWL:Thing` a `COMM:SemanticLabelRole`.

BLANK PAGE
IN
ORIGINAL

3

OPEN PROBLEMS IN TEXT MINING

In this chapter, I describe in detail a subset of open problems in the field of text mining which is relevant to the work presented in this thesis. I refine the main research questions formulated in chapter 1, setting out a set of subproblems to tackle in Part II.

I argue that text mining approaches are currently still hard to design, develop and maintain due to the cost of engineering features, the cost of obtaining labelled data and three other non-technical problems, which I designate the systemic, communication and replicability problems.

3.1 PROBLEM 1: THE COST OF ENGINEERING FEATURES

Feature engineering is the process of analysing the learning examples where the classifier made the costliest mistakes and identifying novel classes of features that could potentially enable a better discrimination of those examples, in order to reduce the overall classification error. The repeated application of this method can be termed a *feature engineering cycle*.

Feature engineering

The outcome of the feature engineering cycle are novel sets of features, which are typically derived via a number of hand-crafted atomic observational tests, e. g., word is capitalized, or word is "speaker", or word appears in lexicon of city names. A large collection of features for learning is then formed by making conjunctions of the atomic tests in certain pre-defined patterns, for example the conjunctions consisting of all tests at the current sequence position conjoined with all tests at the position one step ahead, e. g.current word is capitalized and next word is "plc".

At first glance, it would seem plausible to take the classes of features reported to work well in the literature and use them all in order to build the "best" performing text mining system. Unfortunately, this naïve approach does not work for several reasons. First, the systems that underlie the reported results in the different publications are not the same. This is because authors rarely make their systems available, leading the unfortunate reality that each author implements their own system, which will inevitably differ in some detail from another author's system. Thus, classes of features that work well in a given system are many times irrelevant when tried in another system, due to those unreported differences. To make matters worse, results are usually reported on different corpora. Even when they are reported on the same corpus, different splits of the corpus for training and testing the systems may be used[1]. Second, despite the robustness of modern learning algorithms, simply merging two successful classes of features does not necessarily mean that better results will be obtained — in some cases, the accuracy of the system drops slightly with respect to using just one or the other class of features. Third, introducing redundant and/or irrelevant features into a system does come at a cost, as it increases the running time of the learning algorithm, and it may require running more input pre-processing tools, which also add to the total running time of the system.

Reported results not comparable

Redundant and irrelevant features

In spite of the apparent simplicity of the process, the feature engineering cycle is, in fact, a very time consuming part of building a ML-based text mining system. Part of the problem stems from the sheer volume of data and results involved, the quantity of which defies manual organization and analysis. Moreover, it is a process characterized by many frustrated attempts to improve classifier accuracy, as the majority of the features tried tend not to make a difference. When a set of features is found that does improve accuracy, the improvement tends to be a small increment. The process ends when a given target accuracy is achieved, or when time runs out to keep pursuing it.

Time consuming iterative process

Therefore, because the cost of the feature engineering cycle is typically prohibitively high in terms of time and effort, the results of ablation studies, showing which classes of features have been successfully utilized in which settings, constitute valuable lessons for text mining practitioners to jumpstart the process of designing and developing new TC and IE systems.

Feature engineering is, indeed, one of the most promising means of improving system performance in a variety of statistical natural language processing and machine learning tasks. While automatic feature selection [155], ensemble methods [49], and innovative statistical models each offer the possibility of significant accuracy gains, their potential contributions are necessarily constrained by the quality of the features

[1] A more detailed discussion on the issues in evaluating IE systems can be found in [106].

extracted from the text and images. As experience with machine learning for solving the document classification and information extraction tasks accumulates in the field, practitioners have found that feature engineering is as critical as the choice of machine learning algorithm, given that feature design does significantly affect the performance of systems in practice.

Impact of statistical feature selection methods

A related question, which arises from the fact that statistical feature selection is a successful technique in document classification, is whether feature selection can similarly contribute to an improvement in the accuracy of EE approaches. In chapter 4, I present, as part of the ablation study, empirical evidence showing that it does not.

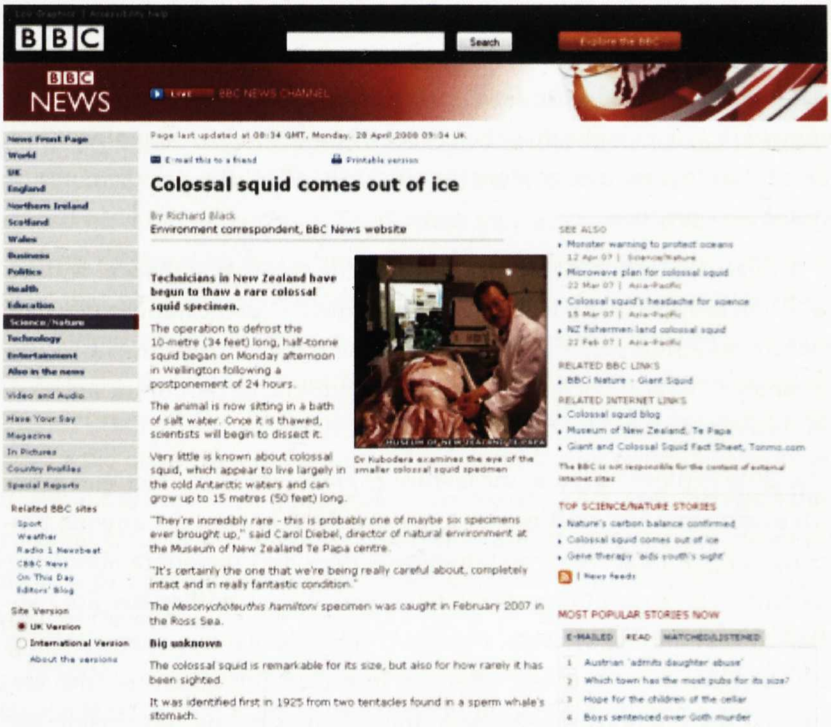
Features arising from multimedia nature of documents

A little explored avenue for feature engineering is the fact that modern documents are multimedia in nature. Multimedia documents typically carry a mixture of text, images, tables and metadata about the content (e. g., style information), all of which are traditionally not handled by text mining systems. However, valuable features characterising the document content can often be found in the structure, the layout and the relationships between the different media comprising the document.

Unfortunately, text mining systems tend to simply strip documents down to their core single-medium formats, for example by treating the document as a set or sequence of words plus a separate set or sequence of images. As a consequence, cross-media features that would otherwise be valuable to the mining algorithm [112, 26] are unfortunately ignored. In fact, these traditional simplifying assumptions about document content pre-processing have been reported not to be adequate for more demanding real-world applications such as information retrieval from patient diagnostic reports [51, 174] or jet-engine maintenance reports [46].

Exploit correlations across media

Instead of treating each medium separately, it would thus arguably be advantageous to exploit potential correlations across the media elements. This poses the problem of how to effectively model the relationship between those media objects, in other words, to identify with novel sets of features via feature engineering. For example, in news articles, like the one in Figure 17, typically one or more images are included with the story that illustrate the events. Features such as their caption or their disposition with respect to the text flow are not random, but rather choices made by the editor according to conventions



Page last updated at 08:34 GMT, Monday, 28 April 2008 09:34 UK

Colossal squid comes out of ice

By Richard Black
Environment correspondent, BBC News website

Technicians in New Zealand have begun to thaw a rare colossal squid specimen.

The operation to defrost the 10-metre (34 feet) long, half-tonne squid began on Monday afternoon in Wellington following a postponement of 24 hours.

The animal is now sitting in a bath of salt water. Once it is thawed, scientists will begin to dissect it.

Dr Kubodera examines the eye of the smaller colossal squid specimen

Very little is known about colossal squid, which appear to live largely in the cold Antarctic waters and can grow up to 15 metres (50 feet) long.

"They're incredibly rare - this is probably one of maybe six specimens ever brought up," said Carol Diebel, director of natural environment at the Museum of New Zealand Te Papa centre.

"It's certainly the one that we're being really careful about, completely intact and in really fantastic condition."

The *Mesonychoteuthis hamiltoni* specimen was caught in February 2007 in the Ross Sea.

Big unknown

The colossal squid is remarkable for its size, but also for how rarely it has been sighted.

It was identified first in 1925 from two tentacles found in a sperm whale's stomach.

SEE ALSO
- Monster warning to protect oceans
  12 Apr 07 | Science/Nature
- Microwave plan for colossal squid
  22 Mar 07 | Asia-Pacific
- Colossal squid's headache for science
  15 Mar 07 | Asia-Pacific
- NZ fishermen land colossal squid
  22 Feb 07 | Asia-Pacific

RELATED BBC LINKS
- BBCi Nature - Giant Squid

RELATED INTERNET LINKS
- Colossal squid blog
- Museum of New Zealand, Te Papa
- Giant and Colossal Squid Fact Sheet, Tonmo.com

The BBC is not responsible for the content of external internet sites

TOP SCIENCE/NATURE STORIES
- Nature's carbon balance confirmed
- Colossal squid comes out of ice
- Gene therapy 'aids youth's sight'

MOST POPULAR STORIES NOW

1 Austrian 'admits daughter abuse'
2 Which town has the most pubs for its size?
3 Hope for the children of the cellar
4 Boys sentenced over Goth murder

Figure 17: Example of a Web news article.

defined by professional and/or web communities, whether explicitly or implicitly.

Following such rationale, in chapter 4 I present an approach to DC that takes advantage of cross-media correlations in documents, and show that it outperforms baseline text mining and simple multimedia mining approaches in terms of the measured accuracy in classifying documents. In order to achieve that, it is fundamental to identify effective ways to use cross-media features.

To summarise, my research concerning the cost of engineering features addresses the following sub-questions:

- Which classes of features contribute more significantly to the accuracy obtained in EE approaches?
- How does accuracy vary with the several parameters introduced by the EE learning model?
- Do statistical feature selection methods have a positive impact in the EE task as they do in the TC task?

- Can cross-media features help improve the accuracy of classifiers in the DC task?

The answers to the above questions are given in chapter 4.

3.2 PROBLEM 2: THE COST OF OBTAINING LABELLED DATA

Even though the whole point of constructing automated systems is to circumvent the need for domain experts to manually analyse documents, the experts are still needed in the initial stages of development, or during maintenance, to provide enough training material to enable the application of machine learning techniques. Concretely, the role of the expert is to perform semantic annotation on some documents representative of the domain of interest. For example, a mention of a gene can be marked in the documents where it appears, as can the mentions of relationships between genes, and these marks can be linked to entity and relation classes in the ontology that models the gene domain, thereby granting the appropriate semantics to the annotations. From a machine learning viewpoint, the annotations work as labels for the examples passed to the learning algorithm.

Expensive to obtain labelled examples

Unfortunately, experts are rarely available or willing to spend their time annotating. During the content creation phase, annotating adds a burden that is very often seen as "a waste of time", and it is therefore difficult to convince experts to make the extra effort of providing the semantics of newly created content. For legacy content, typically a document repository exists before any ontology has been defined for the domain in question, and so semantic annotations need to be created at the time the ontology is defined. This suffers from the same problems as before, aggravated by the fact that potentially a large amount of semantic annotations need to be produced in a short period of time.

Limited labelled data availability

Therefore, in real-world settings, such as the one described in chapter 5, labelled data availability for adopting ML-based text mining approaches will depend on a number of factors, including willingness and/or capability of domain experts to annotate, the complexity of the natural language analysis that needs to be performed – and, closely related, the complexity of the semantic annotation process –, the methodology employed and tool availability to obtain the labelled data, and so on. When working with real-world data outside controlled lab environments where annotation quality can be tightly monitored and enforced, it is not safe to make assumptions about either the amount or the quality of labelled data that will be available to the learning methods.

However, it is relatively safe to assume that in many real-world tasks it will be cheap to find unlabelled data. Large databases of text and multimedia documents are generally available for a given domain, where only a very small portion of them is hand-classified. It is also prudent to be conservative about the expectation of the semantic annotation quality. In particular, the assumption should not be made, contrary to some academic settings, that, when annotating a given document, the domain expert will make a complete annotation for that document – some annotations may be missed, either intentionally or unintentionally. This has an impact on the way the information extraction tasks should be modelled. Concretely, the absence of annotation cannot be modelled as a negative example for the classifier. Semi-supervised methods provide an answer to this problem, and for that reason I explore their use in chapter 5.

The benefit of semi-supervised approaches

If creating semantic annotations for one domain is difficult already, maintaining and porting them across domains is an even more daunting task. It is crucial to devise methods to minimize the experts' effort in providing labelled data during the creation and maintenance phases. One way to achieve this is through the use of domain adaptation methods.

Domain adaptation [13] is a fundamental learning problem where one wishes to use labelled data from one or several source domains to learn a hypothesis performing well on a different, yet related, domain for which no labelled data is available. The task of domain adaptation is particularly relevant to real-world TC problems, because the simplifying assumption, often made, that documents in the training set are drawn from the same underlying distribution as documents in the test set rarely holds in practice. As a consequence, statistical models derived from training data drawn from the "source" domain typically do not perform well on test data drawn from the "target" domain. For example, [126] report that a text classification model trained on a Yahoo! directory performed poorly on a Weblog classification problem, since the distribution of terms differed significantly.

Domain adaptation

The feature labelling paradigm introduced in section 2.6 is particularly appealing for the domain adaptation task because it is often possible for domain experts to tell which features from the source domain are expected to apply robustly also in the target domain. This is easier and less time consuming than labelling documents. Unfortunately, approaches to domain adaptation have not considered the use of the feature labelling paradigm so far. This is what will be explored in this thesis, in chapter 5.

Feature labelling for domain adaptation

To summarise, my research concerning the cost of obtaining labelled data addresses the following sub-questions:

- What is the impact, in a real-world setting, of using semi-supervised methods for the task of EE, both in system accuracy and training time?
- Can the accuracy obtained by a feature labelling approach to domain adaptation be comparable with that of state-of-the-art approaches that use a more costly instance labelling approach?

The answers to the above questions are given in chapter 5.

3.3 FURTHER PROBLEMS

I have argued so far that designing, developing and maintaining text mining systems that work on real-world tasks and data is currently complex and costly. There are further components to that cost beyond the cost of engineering features and the cost in obtaining labelled data, which are not tied to technical ML aspects.

The multidisciplinary set of skills required to understand all the concepts involved, and the sheer number of software subsystems and components that need to be integrated, translates into an extremely high cost of entry for novice developers of IE systems. It also means that maintenance costs are considerable, even for expert developers. I will call this the *systemic* problem. Further, there is often ambiguity in the terminology employed by engineers and researchers who, coming from areas as diverse as Natural Language Processing, Image Analysis, Machine Learning and the Semantic Web, work together to develop IE solutions. I will call this the *communication* problem. On top of this, results reported in the scientific literature are hard to replicate, because IE systems' descriptions are almost invariably incomplete due to the difficulty in covering all the details in the limited space allocated for publication. I will call this the *replicability* problem.

3.3.1 *Problem 3: The Systemic Problem*

Besides the functional concerns identified in chapter 2 (i.e., decomposition, segment analysis, data modelling and semantic annotation), it is not uncommon for a designer of text mining systems to have to take into account a few or all of the following *non-functional* concerns:

Scalability: to be able to handle large amounts of documents or cope with domains in which a large number of entities and relations amongst those entities exist.

Expressiveness: to understand which data structures are expressive enough to hold the models of the data and their intermediate representations.

Portability: to seek a modular design of the system so that porting it to other tasks and domains can be done quickly by mere parameterisation of functional components.

Data Integration: to merge data outputted by several pre-processors (e. g., part-of-speech tagger, image texture analyser) working on multiple documents, and data from external resources (e. g., ontologies, knowledge bases, language resources), into a uniform data representation that supports integrated analysis.

Tools Orchestration: to wrap a number of existing tools and determine the dependencies amongst them in order to process the data.

Formats Handling: to conform to given file formats for the purposes of input/output.

The complexity associated with developing text mining systems translates into an extremely high cost of entry for novice developers. It also means that the maintenance costs are very considerable, even for expert developers. In order to reduce the complexity associated with developing and maintaining text mining systems, the state-of-the-art is moving towards the adoption of Service-Oriented Architecture (SOA) [53] principles. To enable *automated discovery and composition*, services need to be declaratively described, and their descriptions published, so that both users and other services can learn about their capabilities (see Figure 18). Service discovery plays an important role in speeding up or automating the construction of text mining systems from their individual components. And, arguably, service discovery plays an even more important role in making distributed systems robust to the failure

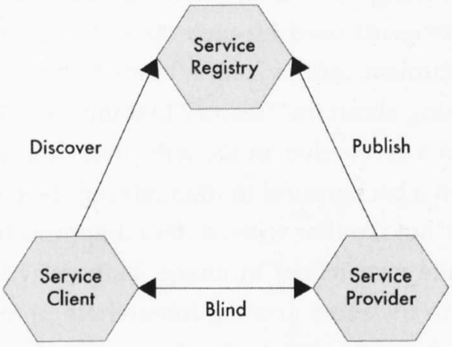


Figure 18: Service discovery in service-oriented architectures. Service repositories aggregate service descriptions published by their respective providers, and reply to queries from users or other services.

of any of the subsystems, as equivalent components can be located automatically and can replace the faulty ones on-the-fly.

The quality of the descriptions (of tasks, subsystems, components, and inputs/outputs) essentially determines how well automated discovery and composition will work. The finer-grained the descriptions are, the richer the queries against the service repository can be. For example, describing the entity recognition system in our example down to the level of the learning algorithm parameters would enable discovering, amongst the available systems, those systems that allow tuning the precision/recall balance of the system, for example. In addition, the richer the semantics of the description, the more accurate the queries can be. For instance, if the system uses a stochastic gradient descent learning algorithm, and the algorithm is described as having sub-quadratic average complexity, then, with the right semantics in the Knowledge Base (KB), it would be possible to automatically infer that the system is suitable for working over large datasets.

3.3.2 *Problem 4: The Communication Problem*

The complexity of the domain of unstructured information mining is not only reflected in the complexity of its systems, but also in the considerable amount of concepts to master and in the variability with which those concepts are named and referred to. The simplest form of the communication problem is a terminology problem: engineers and researchers coming from different backgrounds, working together to develop text mining solutions, are often faced with difficulty in understanding the terms used by each other. For example, someone specialised in document categorization tends to simply use the term "word" when talking about the "feature that indicates the presence of a word", whereas a researcher in ML will prefer the term "feature", and someone with a background in data mining the term "attribute"; researchers in NLP not familiar with ML-based approaches will use the term "rules", while researchers in image analysis will use the term "model" to refer to the same artifact; researchers in NLP tend to talk about "annotation" and "annotator", whilst researchers in ML tend to talk about "label" and "oracle"; and so on.

The communication problem is exacerbated by the need to communicate complex ideas related to the structure of the data and/or the system. Therefore, making available an agreed-upon means of referring to and describing text mining tasks and systems, their inputs and outputs and their internal constituents (components, subsystems, auxiliary resources, and so on), can be invaluable as a communication framework for a team or community to share domain knowledge about unstructured information mining.

3.3.3 *Problem 5: The Replicability Problem*

Another important problem addressed in this thesis is that of enhancing *replicability* of empirical research — a concern shared recently in several areas of research, see [135, 146, 157, 106]. As mentioned, results reported in the scientific literature are hard to replicate, because text mining systems' descriptions are almost invariably incomplete due to the difficulty in covering all the details in the limited space allocated for publication.

There is therefore the need for mechanisms to enable providing detailed information both about the IE system and about the *provenance* [134] of extracted facts. Enabling researchers to accompany published results with a unambiguous (formal) description of the text mining methods and systems used in the experimental tasks that they report about, would expectedly lead to advances in the sharing and evaluation of systems and resources in the scientific community.

BLANK PAGE
IN
ORIGINAL

Part II

TACKLING THE COST OF FEATURE AND KNOWLEDGE ENGINEERING

4 FEATURE ENGINEERING FOR TEXT MINING

Identifying the right features for a given text mining problem is a very time consuming, yet extremely important, task in the development and maintenance of ML-based text mining solutions. This chapter addresses the first research question in this thesis, recall from chapter 1:

First research question

> Which classes of features lead to an improved classification accuracy in the document classification and entity recognition tasks?

As mentioned in chapter 3, the cost of the feature engineering cycle is usually prohibitively high in terms of time and effort. For that reason, the results of ablation studies constitute valuable lessons for text mining practitioners. In this chapter, I begin by providing one such comprehensive ablation study for the task of Entity Extraction, deriving conclusions from carefully designed experimental conditions. The outcome of the feature engineering conducted during the ablation study consists in successful sets of features for Entity Extraction.

Later in the chapter I also propose a novel class of features for mining multimedia documents. The method to extract the features exploits document layout and the relationship between the different media comprising the document, and it does so in a generic way, making this class of features extractable from any multimedia document. I show that it is possible to improve accuracy of text mining systems by processing not just text but also images and the cross-media correlations between the elements in a multimedia document.

4.1 SUCCESSFUL FEATURES FOR ENTITY EXTRACTION

In this section I investigate the impact of incorporating diverse classes of features into *boundary classification* approaches (recall from subsection 2.2.2) to the Entity Extraction (EE) task. I also measures the impact of several other typical model parameters that arise in this kind of approach. In addition, a comparative study of the impact of *feature selection* metrics is presented.

Impact of features and parameters

The goal of this *ablation* study is to determine which features and learning model parameters are the real contributors to the success of boundary model EE approaches in the literature, and determine the sensitiveness of the experimental results to each of the several feature types and parameters. For that reason, the experiments are run under the exact same experimental conditions, just varying one feature type or parameter at a time. Because the focus is on feature engineering, the adopted learning model is a simple and fixed one, so as to minimize the influence it might have on the results and conclusions. Two well-known corpora are used in this empirical study.

Lessons learned from the study enabled building an entity recognition system comparable to the state-of-the-art, but which uses a much simpler learning model than other state-of-the-art systems in the literature. The system is briefly described in section A.3 and made publicly available on the Web. Moreover, the successful features that constitute the outcome of this study were adopted in a real-world use case in the design of the EE approach presented in chapter 5.

4.1.1 *Datasets*

The experiments in this section were performed over two standard benchmark datasets for EE: the Seminar Announcements (SA) corpus [62] and the Workshop Call for Papers (WCFP) corpus [87].

Seminar Announcements dataset

The Seminar Announcements corpus consists of a set of 486 emails announcing seminars collected at Carnegie Mellon University. An excerpt of such an announcement is shown in Figure 3. The announcements consist of free text and are thus generally unstructured. In some cases the author provided some kind of structure, which is in any case dependent on the author and not consistent across documents.

Each seminar announcement is annotated with `speaker`, `location`, `stime` (start time) and `etime` (end time). The speaker is the name of the person giving the seminar, the location is where the seminar will take place, and the other two are the time when the seminar will start and end, respectively. Table 2 shows the details about the entity classes. There can be more than one speaker at a seminar, but in this dataset only one speaker has been annotated. Unlike `speaker`, the other classes always have one possible value per seminar, even if it may each

CLASS	FREQUENCY	EXAMPLES
speaker	759	Ralph Hill, Mr. Kurtz, Dr. David Evans
location	645	Student Center Room 207, 7500 Wean Hall
stime	984	4:15, 4.30 pm, 10am
etime	435	5:30, 5.15 pm, 11am

Table 2: Details on the entity classes in the Seminar Announcements corpus.

occur several times in different surface forms throughout the seminar announcement document.

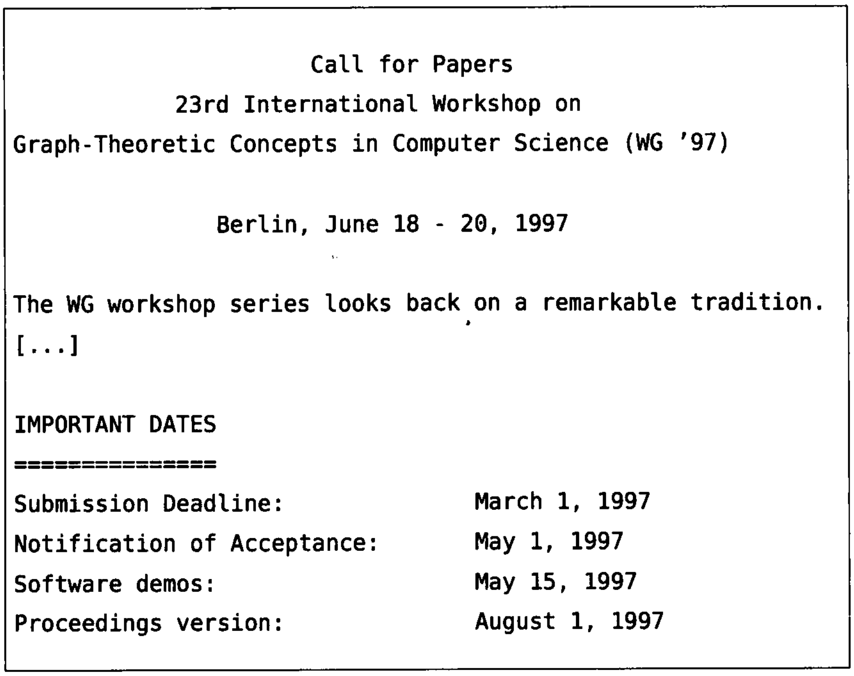
The Workshop Call for Papers corpus was created for the International Challenge entitled "Evaluating Machine Learning for Information Extraction" organised by the PASCAL Network of Excellence, The University of Sheffield, ITC-IRST, U. Illinois at Urbana-Champaign, U. College Dublin and Fair Isaac Corporation. It consists of 1100 workshop call for papers, 600 of which were annotated. From those, 200 were reserved to assess the performance of the systems, and were thus never released by the organisers of the challenge. Most of the text is from the domain of computer science. Moreover, the training and test sets are temporally separate. Figure 19 shows an excerpt of a document in the WCFP corpus.

Workshop Call for Papers dataset

The WCFP corpus is annotated with 11 entity classes, such as workshop name, acronym, homepage, location, date, the deadlines for paper submission, notification of acceptance and camera-ready version, and the name, location and date of the conference associated with the workshop. Every class contain exactly one entity of its type per document. Table 3 shows the details about the entity classes. The similarity between the values for the several date classes, and also between the values for the workshop vs. conference classes, means that it is only possible to discriminate between the values by using the context surrounding the entity mentions.

4.1.2 *A Detailed Study*

The experimental ablation study presented in this section investigates several instantiations of the boundary classification model. The study is important as it intends to clarify the contribution of different classes



```
                    Call for Papers
            23rd International Workshop on
Graph-Theoretic Concepts in Computer Science (WG '97)

                Berlin, June 18 - 20, 1997

The WG workshop series looks back on a remarkable tradition.
[...]

IMPORTANT DATES
===============
Submission Deadline:                March 1, 1997
Notification of Acceptance:         May 1, 1997
Software demos:                     May 15, 1997
Proceedings version:                August 1, 1997
```

Figure 19: An excerpt of a workshop call for papers.

CLASS	FREQUENCY	EXAMPLES
wname	788	AAAI-95 Fall Symposium
wacronym	809	ZobIS 96, RTSS'98
wdate	912	June 8-9 2000, September 1st-2nd
whomepage	582	http://www.cs.virginia.edu/wecwis2000
wlocation	681	Pisa, Italy ; Cottbus ; Cottbus, Germany
wsubmissiondate	906	March 1 ; June 3, 1996
wnotificationdate	581	Friday 27th March 1998
wcamerareadydate	518	Mar. 24, 2000
cname	294	15th International Conference on Conceptual Modelling
cacronym	607	ACL/COLING '98, ECAI-2000
chomepage	179	www.acm.org/sigs/sigmm/MM99

Table 3: Details on the entity classes in the Workshop Call for Papers corpus.

of features and model parameters to the overall performance of this approach to EE.

As introduced in subsection 2.2.2, boundary classification models make use of two independent binary classification tasks: classifying a boundary in the text as to whether it is the start of an entity mention, and classifying it as to whether it is the end of an entity mention. Thus, a learning example describes a boundary in the training corpus. The boundary becomes the center of a window of tokens to its left and right, of a given fixed size. The boundaries that start an entity mention labeled with "tag" are the positive examples for the classifier of `start-<tag>`, while all the other boundaries in the corpus become negative examples for this classifier. Conversely, the positive examples for the classifier of `end-<tag>` are the boundaries that end an entity mention labeled with that "tag", and all the other examples constitute negative examples.

In the study, the following classes of features and model parameters are used in the experiments:

EFFECT OF COMBINING CLASSES OF FEATURES Typical external data resources and processors used in EE include sentence splitters, tokenisers, parts-of-speech taggers and gazetteers. This experiment will show the effect of combining these resources and their contribution to the different entity classes. The experiment will combine four kinds of token-related features: the *token string*, the *token part-of-speech*, the *token orthography*, and categories for the token looked up in a *gazetteer*. These classes of features are denoted by `S`, `P`, `O`, and `G`, respectively. The data resources used for this experiment are the default ones provided by the NLP tools chosen (see below).

EFFECT OF QUALITY OF THE FEATURES This experiment takes the resources of the previous experiment and tries to improve them. The parts-of-speech are organised in a *tree structure* where a part-of-speech tag can have a parent tag, e. g., `VBD`, `VBN` and `VBZ` tags have a more generic `VB` parent tag. When a tag is inserted as a feature, its ancestors up to the root of the tree are also inserted. This potentially helps the learning machine to generalise better.

The orthographic categories for this experiment were augmented with specific categories for one and two letter words, words containing special characters and acronyms, inspired by what is done in the $(\mathsf{LP})^2$ algorithm. Moreover, the orthography is also organised hierarchically in a similar manner to the parts-of-speech.

The gazetteer used in this experiment is the gazetteer used in [59]. This gazetteer includes roughly the same categories as the one used in the previous experiment, but contains many more entries, particularly related to first and last names. Concretely, it contains several tenths of thousand entries for first and last name, whereas the previous gazetteer only contained a few hundred entries for first name. In contrast, the gazetteer used in this experiment contains fewer categories for date and time than the previous one. The classes of features for this experiment are denoted by `P'`, `O'`, and `G'` respectively.

EFFECT OF SPACE AND NEWLINE TOKENS The effect of space and newline tokens is dependent on the nature of the dataset and of the entity class to extract. This experiment explores three variants in the way the corpus is preprocessed: removing all space and newline tokens; removing just space tokens, keeping newline tokens; and keeping all token types.

EFFECT OF TOKEN WINDOW LENGTH As explained in subsection 2.2.2, boundary classification models take tokens in the vicinity of the boundary to generate features, forming a so-called "window" of tokens around the boundary. This experiment analyses the impact of the chosen window length in the performance of the system.

EFFECT OF FEATURE SELECTION [67] have shown that instance selection is technique able to greatly reduce the complexity of the learning problem while maintaining accuracy. Inspired by their work, this experiment addresses a related question. It takes standard feature selection metrics widely used in text categorisation and applies them in the context of the boundary classification approach to EE. In contrast with the text categorisation field where feature selection has been widely studied, little is known about the effects of using feature selection in EE. The feature selection metrics used in this experiment are the ones described in subsection 2.4.3: *cross-entropy, information gain, frequency* and a *random baseline* metric.

All experiments use a base system and modify it, a single change at a time, for determining the effect in the results obtained by running an otherwise identical experiment.

The base system pre-processes the corpus using the default ANNIE components of General Architecture for Text Engineering (GATE)[1],

[1] http://www.gate.ac.uk, see chapter 6 for a brief description

namely the default tokeniser, parts-of-speech tagger and gazetteer. In the base system, no tokens are discarded, not even space tokens. The features are encoded as described in section 2.5. The default window length is 5 tokens to each side of the boundary. The SVM implementation used in all experiments is SVMLight [92], using a linear kernel with parameters j=2, c=0.075 for the SA corpus and j=10, c=0.05 for the WCFP corpus, optimised by cross-validation (grid search). Feature selection is performed on each binary classifier's datasets separately.

Base system

At the end of the classification process, the predictions for the start and end of the entity mentions coming from both classifiers are paired by (a) recursively enumerating all possible pairs for each document (b) calculating a score for each possible subset of the superset of pairs, based on the sum of classifier confidence measures for the individual predictions and (c) selecting the set of pairs that maximizes the confidence score . The pseudo-code for the pairing algorithm can be found in Appendix B.

Pairing open and close tags

Regarding validation, all experiments were run over ten random 50:50 splits of the SA dataset and ten random 75:25 splits of the WCFP dataset. The assessment of the system performance considers strict matches only (see section 2.3). The computed precision, recall and f-measure values were macro-averaged, mainly because the micro-averaged values were not reported in the literature for some of the systems, making result comparison only possible with macro-average. All F-measures report on the ten random split runs.

Validation methodology used

4.1.3 *Results*

The results obtained in the ablation study on the several classes of basic features — token string, orthography, part-of-speech, and gazetteer lookup — are shown in Table 4 and Table 5, for the SA corpus and the WCFP corpus, respectively. Note that for the sake of clarity only a few of the slots for WCFP are shown — those which exhibit higher sensitivity to changing the features classes.

Table 6 and Table 7 show the results obtained concerning the use of several classes of more advanced features, as introduced earlier: hierarchical part-of-speech, enhanced and hierarchical orthography, and enhanced gazetteers. The reported results consist of the difference

FEATURE	LOCATION	ETIME	STIME	SPEAKER	MACRO
O	0	12.3	52.1	0	16.1
G	0	76.5	71.6	7	38.78
P	37	84.7	82.6	42.78	61.77
S	82.3	95.9	**94.8**	53.7	81.7
POG	72.8	95.4	91.9	70.54	82.7
SG	83.2	94.7	94.5	72.6	86.2
SO	85.9	**96.4**	94.3	69.1	86.4
SP	86.2	96.2	94.2	70.9	86.9
SPO	**86.7**	96.2	94.2	72	87.3
SOG	85.7	94.9	94.7	77.8	88.3
SPG	86	95.9	94.5	78.5	88.7
SPOG	86.2	95.9	94.5	**78.8**	**88.9**

Table 4: The effect of different classes of basic features on the results for the Seminar Announcements dataset, measured using F-measure. The best results for each class are highlighted in bold font.

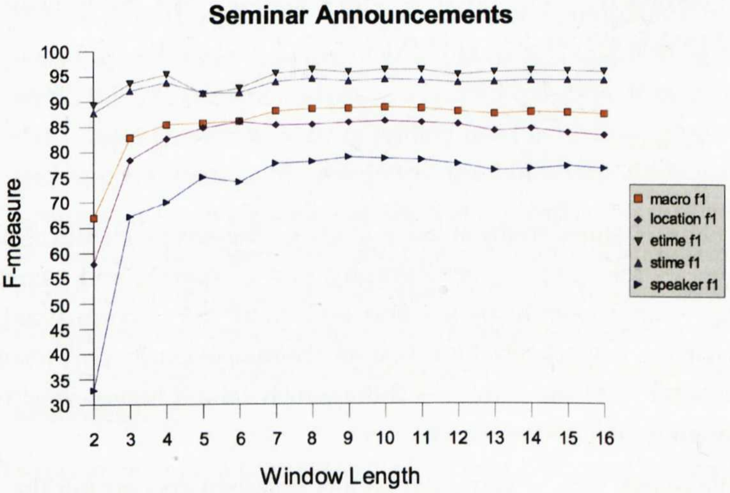


Figure 20: The effect of the token window length parameter on the results averaged over all classes in the Seminar Announcements corpus. The x axis shows the length of the window in number of tokens to either side of the boundary, while the y axis shows the F-measure obtained.

FEATURE	WACRON	WLOCAT	WDATE	WHOME	MACRO
O	29.6	0	0	4.1	6.9
G	0	42.1	59.2	1.3	19.2
P	53.2	38.1	53.3	49.4	41.3
POG	57.3	53.3	66.5	54.7	50.3
SP	69.3	60.3	67.4	58.5	60.4
S	69.6	55.3	69.7	60.5	60.6
SG	67.7	63.6	70.4	60.4	60.8
SOG	70.5	62.6	71.1	60	60.9
SPO	**71.7**	58.1	68.5	58.1	61
SPOG	69.6	63.2	**71.5**	58.2	61.1
SPG	71.6	58.9	70.3	**61.3**	61.4
SPG	71.6	**63.9**	70.6	58.4	**62.1**

Table 5: The effect of different classes of basic features on the results for the Workshop Call for Papers dataset, measured using F-measure. Results for the workshop acronym, location, date and homepage classes are shown.

FEATURE	LOCATION	ETIME	STIME	SPEAKER	MACRO
O′ – O	**54.2**	**76.9**	**33.8**	**41.3**	**51.5**
P′ – P	8.4	3.3	2.1	3.9	4.4
G′ – G	0	**-16.2**	**-25.2**	**61.8**	5
SP′ – SP	-0.4	0.3	-0.7	-0.3	-0.3
SO′ – SO	1.1	-1.5	-1	0.9	-0.2
SP′O′G – SPOG	-1.3	-1.8	-0.8	-2.5	**-1.6**
SPOG′ – SPOG	0.8	1	0.2	**6.82**	**2.2**

Table 6: The effect of adding classes of advanced features on the results for the Seminar Announcements dataset, measuring the difference in F-measures obtained. Selected differences for each class are highlighted in bold font.

FEATURE	WACRON	WLOCAT	WDATE	WHOME	MACRO
O′ – O	**22.1**	**40**	**54.3**	**52.6**	**33.4**
P′ – P	-1	-1.6	-0.9	1	-0.2
G′ – G	0	-3	**-39.9**	-1.3	**-7.4**
SP′ – SP	0.5	-0.3	0.9	0.3	0.4
SO′ – SO	1	**2.5**	0	-1	0.5
SP′O′G – SPOG	0.7	**2.2**	-0.9	-1.4	0.5
SPOG′ – SPOG	1.1	-1	**2.9**	**-3.9**	0.2

Table 7: The effect of adding classes of advanced features on the results for the Workshop Call for Papers dataset, measuring the difference in F-measures obtained. Selected differences for each class are highlighted in bold font.

SETUP	LOCATION	ETIME	STIME	SPEAKER	MACRO
A	82.9	96.1	93	72.3	86.1
B	85.8	**97.4**	94.1	**79.4**	**89.2**
C	**86.3**	95.9	**94.6**	78.8	88.9

Table 8: The impact of space and newline tokens on the results for the Seminar Announcements dataset. Configurations: A. removed spaces and newlines, B. removed spaces only, C. nothing removed.

SETUP	WLOCAT	WDATE	WHOME	WNAME	MACRO
A	63.9	72.2	**60.7**	55.7	62.6
B	**71.8**	**74.4**	57.6	**65.9**	**65**
C	63.2	71.5	58.2	59.9	61.1

Table 9: The impact of space and newline tokens on the results for the Workshop Call for Papers dataset. Configurations: A. removed spaces and newlines, B. removed spaces only, C. nothing removed.

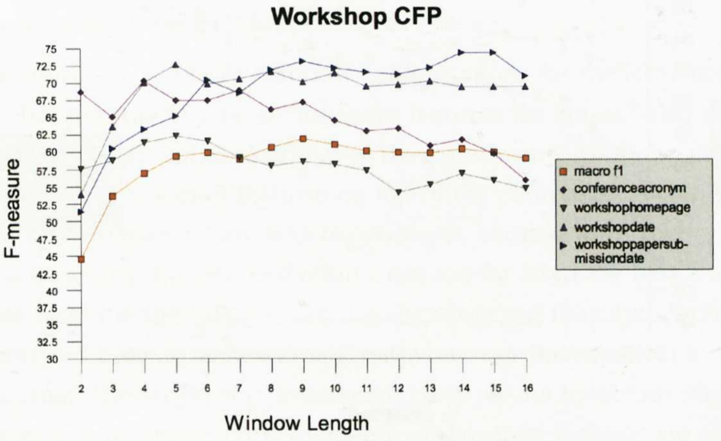


Figure 21: The effect of the token window length parameter on the results averaged over all classes in the Workshop Call for Papers corpus. The x axis shows the length of the window in number of tokens to either side of the boundary, while the y axis shows the F-measure obtained.

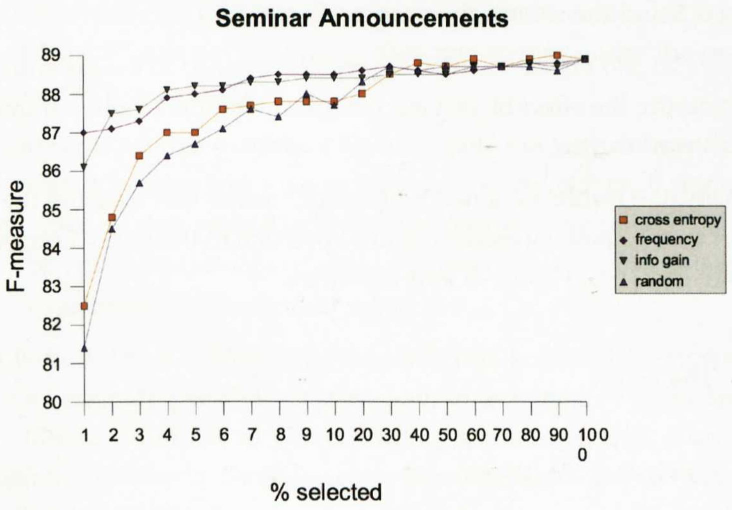


Figure 22: The effect of feature selection on the results averaged over all classes in the Seminar Announcements corpus. The x axis shows the percentage of features selected, while the y axis shows the F-measure obtained.

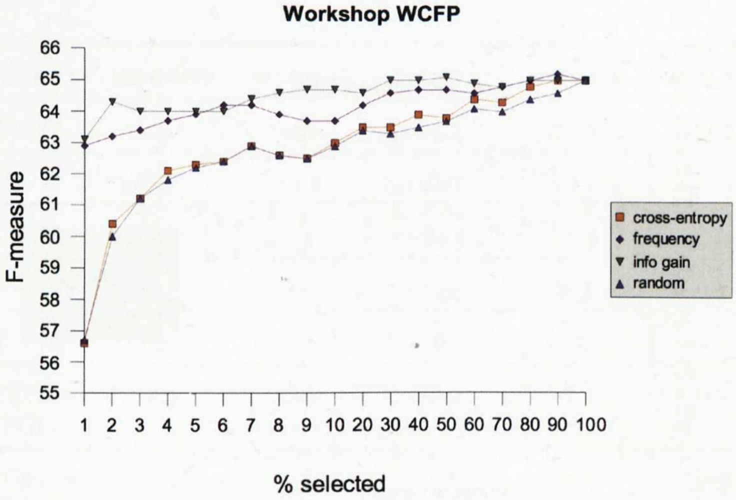


Figure 23: The effect of feature selection on the results averaged over all classes in the Workshop Call for Papers corpus. The x axis shows the percentage of features selected, while the y axis shows the F-measure obtained.

in the F-measure obtained between a given configuration of feature classes and their respective enhanced counterparts.

The results obtained in the experiments that measure the effect of newline and space tokens are presented in Table 8 and Table 9, for the SA corpus and the WCFP corpus, respectively.

Figure 20 and Figure 21 show the results obtained in the experiments that measure the effect of varying the token window length parameter, as discussed earlier.

Finally, the results obtained in the experiments that measure the impact of feature ranking metrics in this approach to the Entity Extraction task are shown in Figure 22 and Figure 23.

4.1.4 *Discussion*

4.1.5 *Analysis of the Results*

It can be observed that, for the SA corpus, *in general the more classes of basic features are used, the better the results*, confirming the rule of thumb "the more features the better", which can be explained by SVMs' robustness to noisy and redundant data. The inclusion of gazetteer lookups boosts particularly the results of `speaker`, but for `location`, the addition of gazetteer seems to always have a slightly negative effect. Remarkably, `stime` and `etime` achieve very good results with nothing else but the plain token string features.

The results on the basic features' configurations for the WCFP corpus show that the rule of thumb "the more features the better" also seems to apply, although not as clearly as in the experiments for the SA corpus. The additional classes of features on top of the plain token string seem be less discriminative than in SA experiments, because *using token string alone achieves impressively good results*, not too far from the best feature classes combination (`SPG`). It can also be observed that the workshop location and date classes score highest whenever the gazetteer is used. In contrast, the workshop homepage class seems to obtain slightly worse results whenever parts-of-speech or gazetteer lookups are added.

On both datasets, the improvements on the speaker, workshop location and workshop date classes through the use of a gazetteer are as expected. The exception is the location class in the SA dataset. This may be explained by the fact that most seminar locations consist of room names/numbers rather than city/country names.

Regarding the use of enhanced features, clearly, `O'` and `P'` perform much better in isolation than their counterparts `O` and `P`. However, when used together with other classes of features their discriminative value does not actually contribute much. Overall, it can be concluded that *the use of* `O'` *and* `P'` *does not constitute a clear improvement*, since the results differ on the two datasets.

It can be noted that gazetteers `G` and `G'` perform very differently. On slots related to date and time, e. g., `etime`, `stime`, `wdate`, `G` performs better, while on slots related to people's names, namely `speaker`, `G'` performs better. This is easily explained by considering the characteristics of the gazetteers, as described in subsection 4.1.2.

In both datasets a clear improvement was achieved by removing space tokens only. For these datasets and, in fact, most datasets, spaces yield little discriminative value. *Treating spaces as separate tokens can thus adversely influence the ability of the learning machine to generalise.* On the other hand, the presence of *newline tokens seems to have significant positive influence* on the results for both datasets.

There seems to be an optimal token window length for each entity class in each dataset. Evidently, the token window should be large enough in order to capture useful patterns in the text. But it is somewhat surprising to learn that for windows that are "too large" there is a constant small drop in the F-measure obtained. This can be seen as a sign of over-fitting. There are even some classes, for instance the workshop homepage or the conference acronym, that reveal a significant drop in F-measure as the window length increases. In contrast with comments about the previous experiments, this goes against the rule of thumb "the more features the better". Roughly, the *optimal average window length seems to be around* $l = 9$ to each side of the boundary, for both datasets.

Concerning the use of feature ranking in this task, it can be observed that the behaviour of the several metrics is similar in both datasets: in general, cross-entropy does not improve much over the random baseline. Information gain and frequency metrics are able to effectively reduce the number of features used to between 5% and 10% of the total features, with little harm to the overall results. For a few ranges of the percentage of selected features parameter there was a slight improvement over the F-measure obtained by the system configuration that does not employ feature selection. However, it can be concluded that *the use of feature ranking methods does not provide a clear improvement* in any of the datasets.

4.1.6 *Analysis of the Errors*

In order to understand what is happening "under the hood" and beyond the statistics of the previous section, an analysis of the kind of errors that the EE system makes was conducted. The following types of common errors, illustrated with examples below, were identified:

PARTIAL MATCHES In this type of error, the system recognises the entity mention correctly, but its predicted boundaries differ from the ones provided by the human annotators. For example:

> `<location>`Student Center Room 207 (CMU)`</location>`

vs.

> `<location>`Student Center Room 207`</location>` (CMU)

This error is quite acceptable, and actually only considered an error due to the matching scheme adopted (strict). Many evaluation methods in the literature would count it as a true positive.

TWO-FOR-ONE This kind of error sees the system propose two different entity mentions for a single class. However, the assumption in both datasets is that only one entity per document exists. For instance:

> fluids, <stime>4:30 p.m.</stime>, Coffee at <stime>4:15</stime>

vs.

> fluids, <stime>4:30 p.m.</stime>, Coffee at 4:15

A simple solution to this class of errors would involve a post-processing procedure to remove proposed entity mentions according to domain-specific contraints. A more complex solution would involve taking into account such constraints during learning.

MISS THE OBVIOUS Here, the system is able to match an entity mention for a given entity but not another mention for the same entity in the same document. For instance:

> <speaker>Dr. Hill</speaker> is an expert [...] speak with Ralph Hill

vs.

> <speaker>Dr. Hill</speaker> is an expert [...] speak with <speaker>Ralph Hill<speaker>

This happens due to not being possible to capture in the model infrequent contexts. A simple solution to this class of error would involve using string similarity to match entity mentions similar to the one recognised by the system. This would probably only work for simple named entities though.

FALSE FALSE NEGATIVE There are missing annotations in the gold standard, causing the scorer to flag a false negative that should actually not exist. For example:

> Lecture Demonstration by <speaker>Ravi Kiran</speaker>

is correct but missing in file `cmu.andrew.org.che.chegsa-242_0` of the SA corpus. For the purpose of comparison with results in the literature, these errors should not be corrected, limiting the maximum achievable accuracy obtained by systems on these corpora.

Dealing with most of the above types of errors is outside the scope of this chapter, which focuses on feature engineering with as little as possible influence from the algorithmic side of EE systems.

ENTITY CLASS	T-REX	ELIE	GATE-SVM	CRF
location	84.9	**85.9**	81.3	85.3
stime	93.1	90.2	94.8	**99.1**
etime	93.6	94.6	92.7	**96.0**
speaker	**85.9**	84.9	69	76.3
macro-avg	**89.4**	88.9	84.5	88.7

Table 10: Comparing the T-Rex Entity Extraction system with other state-of-the-art systems on the Seminar Announcements dataset. Macro-averaged F-measures over all classes are presented.

4.1.7 *Validation*

Final system configuration

Drawing from the lessons learned from the empirical ablation study, the T-Rex EE system (see Appendix C) was designed and compared with the state-of-the-art, namely with the approaches by Finn[59], Li et al.[110] and Sutton and McCallum[159], which were reviewed in section 2.5. For the SA dataset, the base system used in the experiments was modified to remove space tokens in preprocessing and use the gazetteer used by Finn[59]. For the WCFP dataset, the base system used in the experiments was modified to remove space tokens in preprocessing. For both datasets the token window length was adjusted to $l = 9$. No feature ranking method was adopted.

Experimental setup

Care was taken to ensure the experiments were reproduced exactly as the original authors described them - see concerns about the comparability of experiments in IE in [106]. Therefore, for the SA dataset, we used the same random 50:50 splits repeated ten times and the exactly the same gazetteer as used by Finn in their experiments. For the WCFP dataset, we used the same standard classes of features and the same 4-fold cross-validation splits imposed by Ireson et al.[87] for evaluation of the system that participated in the PASCAL international competition.

Comparing against the state-of-the-art

Table 10 compares T-Rex with other state-of-the-art systems on the SA dataset, while Table 11 does the same for the WCFP dataset. On the SA dataset, T-Rex achieves an improvement over the previously best-reported results for the other two systems. Note that speaker is usually considered the most difficult class to extract in this dataset.

ENTITY CLASS	T-REX	ELIE	GATE-SVM
wname	58.1	55.5	**60.6**
wacronym	66.6	68.3	**69.7**
wdate	**78.2**	70.9	76.8
whomepage	63.8	62.8	**68.5**
wlocation	**67**	55.5	66.9
wsubmission	77.4	70.5	**79.3**
wnotification	78.9	71.9	**80.9**
wcamera	72.8	68.7	**75.9**
cname	62.3	**66.5**	66
cacronym	60.6	**69.1**	66.1
chomepage	29.7	**43.3**	33.1
macro-avg	65	63.9	**67.6**

Table 11: Comparing the T-Rex Entity Extraction system with other state-of-the-art systems on the Workshop Call for Papers dataset. Macro-averaged F-measures over all classes are presented.

T-Rex also compares well against the other two SVM-based systems on the WCFP corpus, obtaining better results than those obtained by ELIE. Most importantly, note that T-Rex is considerably simpler than ELIE in the sense that it does not require a multi-level classification approach in order to achieve better results on the two corpora, and also simpler than GATE-SVM because it uses a standard SVM implementation and does not use feature weighing according to distance to the boundary as GATE-SVM does.

4.2 EXPLOITING CROSS-MEDIA CORRELATIONS

In this section[2], I propose a novel way to derive features for the Document Classification (DC) task, and empirically measure its effectiveness. The new model exploits document layout and the relationship between the different media comprising the document. To support the design and development of approaches to the problem akin to the one adopted

[2] Joint work with J. Magalhães, who, given his expertise in image processing, handled the generation of features from the images in the multimedia documents, for the experiments.

in this section, I also introduce a multimedia categorisation framework that accounts for exploiting features from across the different media present in a multimedia document.

The experimental results reported in this section show that, by preserving not just text and images but also the cross-media correlations between text elements and the images in a multimedia document, it is possible to improve system accuracy, with respect to traditional approaches that ignore cross-media correlations. Plus, an advantage of the proposed approach is that it makes almost no assumptions about the way multimedia content is modelled, and is thus widely applicable.

4.2.1 *Image Analysis*

Recently, given the increasing abundance of multimedia content, there has been interest in exploiting not just the text but the several media present in a document, in particular images, with the aim of improving classification. From a ML viewpoint, this translates into not just deriving features from text, such as those studied previously in this chapter, but also into using *image analysis* methods to derive features from the images.

Image Analysis

Image Analysis (IA) is the field of computer science that deals with the quantitative and/or qualitative characterisation of two-dimensional and three-dimensional digital images, through methods mainly based on pattern recognition and signal processing. The analysis involves exploiting several types of low-level image features, which capture different aspects or dimensions of an image. The MPEG-7 standard, introduced in chapter 2, acts as a guideline for low-level audio and visual feature extraction, and, for the case of images, provides sets of visual colour, texture, and shape descriptors. These are briefly reviewed in what follows:

Low-level image descriptors

Colour histograms

COLOUR DESCRIPTORS Colour is a useful and well-studied feature in many image analysis tasks. MPEG-7 divides colour descriptors into several categories [71] and, in practice, four descriptors are commonly used: the colour histogram, colour moments, the colour coherence vector, and the colour correlogram. Colour histograms and moments, as the names suggest, capture the global colour distribution in an image through histograms and low-order moments (e. g., mean, variance), respectively. The colour coherence vector [133] is an extension of colour histograms, which further divides pixels falling in each colour histogram bin into coherent pixels and non-coherent pixels. The colour correlogram [84] characterises how the spatial correlation of pairs of colours is changing with distance: a correlogram is a square matrix in which entry (i, j) specifies the probability of finding a pixel of colour c_j at a fixed distance from a given pixel of colour c_i.

TEXTURE DESCRIPTORS Texture can be thought of as local arrangements of image signals in the spatial domain, or, alternatively, in the frequency domain (which can be "accessed" via spectral transforms). Studies in psychophysical research (e. g., [141]) have long suggested that the brain performs a multi-channel frequency and orientation analysis of the visual image formed on the retina, and this has motivated computer vision researchers to apply multi-channel filtering approaches to texture analysis. In this line of work, Tamura et al. [161] identified the following properties as playing an important role in describing texture: uniformity, density, coarseness, roughness, regularity, linearity, directionality, direction, frequency, and phase. The Tamura features for image analysis attempt to quantify intuitive information such as roughness, presence of orientation, and picture quality in terms of factors like sharpness of edges and period of repeating patterns. Gabor filters [47] are another important visual primitive in the same line of work, which have been widely applied in tasks like invariant object recognition and edge detection. Two-dimensional Gabor filters are defined as a series of multi-scale and multi-orientation cosine modulated Gaussian kernels, and the Gabor texture representation of images is derived by convolving the image with the Gabor filters (using Fast Fourier Transform). Gabor features enable the detection of spatially local patterns such as oriented lines, edges and blobs. MPEG-7 provides three texture descriptors, two of which are based on the Gabor features.

Tamura and Gabor features

SHAPE DESCRIPTORS The shape of objects plays a critical role in image analysis. Ideally, shape features should be invariant to scaling, rotation, and translation of the object. Unfortunately, this inherent complexity in representing shapes is responsible for shape features to be less developed than their colour and texture counterparts. MPEG-7 does support region-based and contour-based shape descriptors [71]. However, the quality of shape feature extraction processes is often insufficient for their incorporation into higher-level tasks, and for that reason they will not be considered in the work presented in this section. A recent survey of shape feature extraction techniques can be found in [120].

In [111], Ma and Zhang present a detailed comparison of a number of commonly used colour and texture features, using a large and diverse collection of image data. The investigated colour features include colour histograms, colour moments, colour coherence vectors and colour correlogram, with respect to different colour spaces and quantizations. Texture features used in the comparison included Tamura features, edge histograms and Gabor texture features. The choice of image features in subsection 4.2.4 was influenced by this study.

4.2.2 *Proposed Approach*

In this section, a novel approach to exploiting cross-media correlations from multimedia documents is presented. Reiterating, the goal is to improve unstructured multimedia information mining systems' accuracy in classifying multimedia documents.

The approach should make weak assumptions about the way multimedia content is expected to be modelled, so as to enable its application on a wide range of problem domains. The hypothesis explored here is that a minimal set of assumptions suffices to obtain significant improvements. To test this hypothesis, document classification experiment over a large collection of news stories collected from the Web is conducted.

To support the proposed method, a framework that processes the content in a series of steps is introduced, as follows:

Multi-step approach

"DOCUMENT-GRAPH" REPRESENTATION Format-specific parsers convert a multimedia document into a canonical graph representation.

COMPUTATION OF CROSS-MEDIA CORRELATIONS A simple, yet informative, structural analysis algorithm to detect correlations between the different media elements.

INFERENCE Given a set of training document-graphs, a model is estimated for each category. The learned model may then be used on new documents to infer their category.

4.2.3 *Multimedia Document Representation*

The "document-graph" represents a multimedia document using text nodes, image nodes and cross-media edges. Formally, each document is defined as

$$d_n = \{T_n, I_n, X_n\}, \tag{4.1}$$

where its elements are:

Types of graph nodes and edges

- a set $T_n = \{T_{n,1}, \ldots, T_{n,|T|}\}$ of (non-nested) text data nodes, where each node contains a meaningful text block and the corresponding feature vectors derived from its content;
- a set $I_n = \{I_{n,1}, \ldots, I_{n,|I|}\}$ of image data nodes, where each node contains an image and the corresponding feature vectors derived from its visual characteristics; and
- a function $X_n : I \times T \rightarrow \Re$ of cross-media edges, where each edge quantifies the relation between a text node and an image node. They contain a correlation value that expresses the likelihood that both referred nodes concern the same information.

This data representation model captures the essential information to perform cross-media classification and is independent of the way multimedia documents are modelled when stored in their original formats.

4.2.4 *Multimedia Document Processing*

The first step in the creation of the document-graph consists in parsing the documents. Although I developed parsers for several formats, namely OpenDocument, Portable Document Format (PDF) and Hypertext Mark-up Language (HTML), here I discuss processing of the latter format only, noting that, for our purposes, the principles are identical regardless of format.

A set of heuristics are used to perform web content cleaning [109], that is, to filter out irrelevant content from the web pages, e.g., advertisements and navigational links in Web pages. The extraction of the relevant parts of the documents starts by converting the document from HTML into a well-formed XHTML. The following rules strip the document of unwanted content:

MAIN BODY IDENTIFICATION The pre-processor parses the XML tree to locate the tree branch containing the main body of the content. This involves hand-crafting simple corpus-specific patterns over the XML tree.

Processing steps

NOISY STRUCTURES REMOVAL Content such as videos, comments, navigational links, adverts, and so on, remaining in the main body section, is removed, again using corpus-specific patterns.

NOISY IMAGES REMOVAL Some images in the corpus are too small to be processed or are just stylistic images (e.g., an icon). Images with less than 200 pixels are ignored and images with a URL pointing to a specific location (e.g., location where all formatting images are stored) are ignored as well.

This process generates a clean document that serves as the basis for the creation of the document-graph, by extracting the text and image nodes and their relations, as will be detailed in the next sections.

Text nodes are generated by analysing the layout structure of the document, parsing the textual content to extract sentences and processing text data with standard text processing techniques, as follows:

FORMATTING-BASED ANALYSIS Style and layout information define the structure of a document. In the case of XHTML, I use standard formatting tags to guide the extraction of the text, section titles (tags `<h1>`, `<h2>` and `<title>`), alternative text for an image (`<img alt-text="...">`) and image captions (via corpus-specific patterns). This creates the text nodes for titles, captions and "alt-text".

Obtaining the text nodes

TEXT BODY ANALYSIS Textual cues like punctuation provide further information to segment the text. This step creates text nodes corresponding to sentences in the document, keeping the information about their sequence (in the form of "next-sentence" edges).

TEXT PROCESSING Standard text processing techniques [170] are applied: stop words and infrequent words are removed from the text corpus, to avoid over-fitting. After this, I apply the Porter stemmer [137] to reduce words to their morphological root. The resulting text nodes contain a histogram of the vocabulary V of terms, i.e., the vector $T_{n,s} = \{t_1, \ldots, t_{|}V|\}$ represents node s of document n, where each component is the frequency of the corresponding term.

For each image in the document, an image node is created in the document-graph. Information about the image sequence is kept in the form of "next-image" edges between image nodes, and information about immediately adjacent text nodes in the form of "previous-sentence" and "next-sentence" edges. For the case of XHTML, this is done via the analysis of the DOM tree, e.g. an image node is created for each <img> element.

The contents of the image node i of document n are represented as the feature vector

$$I_{n,i} = \{I_{n,i}^{HSV}, I_{n,i}^{Gabor}, I_{n,i}^{Tamura}\} \tag{4.2}$$

where each component corresponds to the following configuration of the visual features introduced in subsection 4.2.1:

Obtaining the image nodes

- Colour features: images are split into 9 equal tiles and an HSV histogram per colour channel with 256 bins is computed.
- Gabor texture features: are computed with a bank of filters in 8 directions and 6 scales for the entire image. I consider the mean and the variance of the output of each filter.
- Tamura texture features: images are split into 9 equal tiles and the three Tamura texture features are computed (contrast, directionality and coarseness).

The above features were obtained using standard algorithms from the Open Computer Vision library [3]. Cross-media edges store the likelihood that images and text nodes in the graph concern the same information. The rationale is as follows. When a message or idea is conveyed through the different media in a multimedia document, each text paragraph and image offers support to different parts or aspects of the message. Because a full text typically expresses several ideas, the problem resides in trying to understand which media elements refer to the same information. In this work, I propose a method to do this in an unsupervised way and making very general assumptions about the document model. Namely, I make the rather generic assumption that it is possible to obtain the sequence of text and images, and that images may have associated text like captions.

Assumptions made

The sequence information in the document-graph provides proximity information that can be used to infer layout-based cross-media associa-

[3] http://sourceforge.net/projects/opencvlibrary/

tions. This is achieved by superposing a window over the text nodes centred on a given image node.

Formally, the layout distance between an image I_i and a text block T_s is defined as

$$f_L(I_i, T_s) = 1 - \frac{\mathrm{NodeDist}(I_i, T_s)}{MaxWindowSize + 1}, \tag{4.3}$$

where $\mathrm{NodeDist}(I_i, T_s)$ is the number of nodes between an image node I_i and a text node T_s (for simplicity, I dropped the index n corresponding to the document), and $MaxWindowSize$ is the maximum window size covering the text nodes around the considered image node. In the experiments reported below, I set the window length to be equal to the document size, thereby computing the correlation between all sentences in the document and the given image.

Cross-reference information is another way of establishing the relation between a sentence and an image. They complement layout-based correlations, capturing those sentences that refer to an image not placed nearby (in terms of the document layout). To detect these associations, techniques for measuring text similarity can determine the level of relatedness between a sentence T_s and an image I_i through its text caption $T_{i,c}$ and its alternative text $T_{i,a}$ (for simplicity, I dropped the index n corresponding to the document).

Formally, the correlation between a sentence and an image is measured as the cosine distance between the sentence T_s and the image's associated text $T_i = T_{i,c} + T_{i,a}$, given by

$$f_T(T_i, T_s) = 1 - \frac{T_i}{\parallel T_i \parallel} \times \frac{T_s}{\parallel T_s \parallel} \tag{4.4}$$

By merging the two methods above for detecting cross-media associations between an image node and a text node, it is possile to quantify the degree of correlation between elements in the two media. Formally, the weight of a cross-media edge in the document-graph is given by the average of the two quantities defined above:

$$\gamma_{i,s} = X(I_i, T_s) = \frac{f_L(I_i, T_s) + f_T(T_i, T_s)}{2}. \tag{4.5}$$

4.2.5 *Multimedia Document Categorization*

Our aim is now to infer the category of a given multimedia document given the cross-media document-graph. To complete this task I follow a probabilistic approach:

$$p(c_l|d_n) = p(\{T_n, I_n, X_n\}, \beta_l), \tag{4.6}$$

where β_l corresponds to the model for the document category c_l from the set $\mathcal{C} = \{c_1, \ldots, c_L\}$ of L categories. In this setting, I define a collection $\mathcal{D} = \{d_1, \ldots, d_N\}$ of N multimedia documents, split into a training set in order to learn the category models, and a test set for evaluation. To simplify the exposition, I shall assume multimedia documents have only one image and extend the method to the general case a posteriori. In this probabilistic setting, a document d_n is represented by the vector

Cross-media document vector

$$\overrightarrow{d_n} = \left[\sum_s \gamma_{1,s} \times T_{n,s}; I_{n,1}\right], \tag{4.7}$$

which includes all text content (the sum over all T nodes), an image feature vector, and the cross-media correlation $\gamma_{1,s}$ which weighs sentences according to their relevance to image $I_{n,1}$. The probabilistic framework of Equation 4.6 was formally implemented as

Inference

$$\log \frac{p(c_l|\overrightarrow{d_n})}{p(\overline{c_l}|\overrightarrow{d_n})} = \log \frac{p(c_l)}{p(\overline{c_l})} + \sum_i p(d_{n,i})\beta_{l,i} \tag{4.8}$$

where $\overline{c_l}$ indicates the non-presence of the category c_l, $d_{n,i}$ is the i^{th} dimension of the document vector $\overrightarrow{d_n}$, and $\beta_{l,i}$ is the i^{th} dimension of the linear model for category c_l. The latter is given by

Model estimation

$$\beta_{l,i} = \log \frac{E\left[d_i|c_l\right]}{E\left[d_i|\overline{c_l}\right]} \tag{4.9}$$

The interpretation of this equation is straightforward: the dimension $\beta_{l,i}$ is close to zero if the i^{th} dimension of d is irrelevant for the category, positive if it is frequent, and negative if it is rare. This way, when evaluating unseen samples each dimension will have a low or high contribution to the detection of the category. Finally, to recover the more general case where documents have more than one image, I simply average the output of $p(c_l|\overrightarrow{d_n}, \beta_l)$ for each image in the document.

The decision function for the proposed multimedia document classifier is simply

$$\arg\max_{c_l} \sum_i \beta_{l,i} d_i, \tag{4.10}$$

with a caveat: a confidence threshold is utilised to filter the prediction output of the classifier above – if the prediction confidence is below 0.5, the prediction is discarded.

4.2.6 *Experiments*

To evaluate the proposed framework, I conducted a categorization experiment on BBC Web news articles that were obtained via a RSS feed. The experiment uses news articles that were obtained between the 2nd of May 2008 and the 4th of June 2008. The category of each news article is obtained via the news category assigned by BBC journalists. On the BBC website, news are organised according to category, and it is possible to extract the category from the article's Uniform Resource Locator (URL). There are a total of 44 categories, which are listed in Table 12. All results were assessed in both a information retrieval setting, using traditional IR measures.

BBC Web News dataset

I collected a total of 6,732 news articles, randomly split into 10 random sets of 4,577 training documents and 2,155 test documents for cross-validation. Each news article belongs to just one category and most articles have at least one image. It is worth reiterating that this dataset is different from other news datasets such as Reuters-RCV1 [108]. The latter contains plain text documents only, while the BBC Web news dataset used here consists of multimedia documents with images and structure.

Evaluation methodology

Documents are first transformed into the document-graph and their cross-media correlations are computed according to Equation 4.5. Category models are learned from the training documents vectors, Equation 4.7, and computed according to Equation 4.9. Once the system is trained, I followed the typical evaluation methodology for document retrieval by category: for each category I ranked the test documents according to equation Equation 4.8 and evaluated the rank with precision-recall curves, average precision, precision after 10 retrieved documents and precision after 30 retrieved documents. The means across all queries are computed from the results per query: mean av-

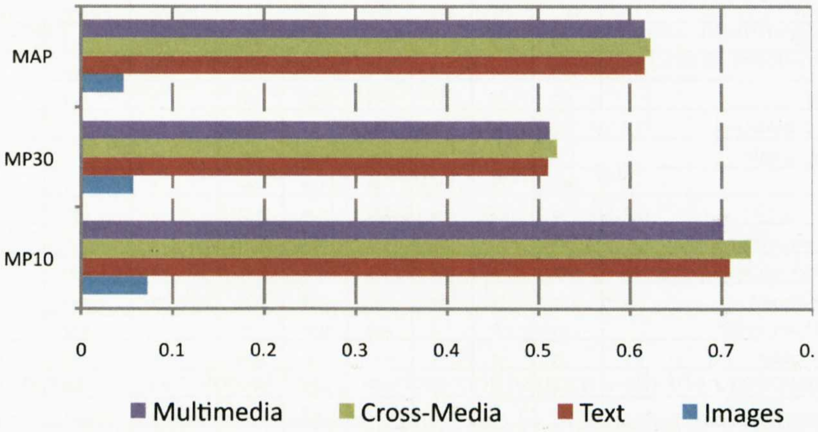


Figure 24: Retrieval results for the individual media, multimedia and cross-media configurations.

erage precision, mean precision at 10 and mean precision at 30. This procedure was carried out for (i) only text data, (ii) only image data, (iii) a simple concatenation of text and image features (multimedia), and (iv) text, image and cross-media correlation features.

4.2.7 *Results and Discussion*

Table 12 presents the detailed results comparing multimedia retrieval by category in the four settings. Table 13 summarizes the information in the aforementioned table, by taking the mean values over the categories. Taking into account the cross-media correlations yields better results with respect to the other configurations, for all the three measures considered. Figure 24 presents the same results in a bar chart for easier comparison.

Not surprisingly, image results are always much lower than the other settings. This observation is justified by the fact that some categories cannot be actually discriminated from just images. For example, there is virtually little difference among the pictures of the categories "/England", "/England/London", "/Scotland", "/Northern_Ireland", "/England/Manchester", and "/Wales". Thus, images only contain information to discriminate between categories like "/sports" and "/uk_politics", i.e., broader categories.

Another interesting observation is that the simple concatenation of text and image features generally does not perform much better than

Category	TP	FP	FN	Pre	Rec	F1	Pre@10	Pre@30	AvgPre
/sci/tech	9	7	8	0.56	0.53	0.54	0.6	0.3	0.47
/sci	8	4	9	0.67	0.47	0.55	0.7	0.33	0.49
/world	183	12	61	0.94	0.75	0.83	1	1	0.89
/world/middle_east	28	1	12	0.97	0.7	0.81	1	1	0.86
/sport/football	156	16	9	0.91	0.95	0.93	1	1	0.96
/sport	269	14	12	0.95	0.96	0.95	1	1	0.98
/uk	30	2	72	0.94	0.29	0.44	1	0.77	0.55
/northern_ireland	56	0	45	1	0.55	0.71	1	1	0.84
/world/south_asia	19	2	12	0.9	0.61	0.73	0.9	0.66	0.69
/entertainment	37	1	60	0.97	0.38	0.55	1	0.93	0.76
/world/asia-pacific	23	8	9	0.74	0.72	0.73	0.9	0.74	0.75
/sport/cricket	23	2	2	0.92	0.92	0.92	0.9	0.77	0.96
/wales	14	7	53	0.67	0.21	0.32	0.9	0.48	0.48
/business	142	14	35	0.91	0.8	0.85	1	0.97	0.88
/england	65	28	81	0.7	0.45	0.55	1	0.97	0.63
/world/europe	20	1	33	0.95	0.38	0.54	1	0.77	0.57
/sport/olympics	6	1	8	0.86	0.43	0.57	0.8	0.4	0.61
/sport/rugby_union	21	0	9	1	0.7	0.82	0.9	0.73	0.93
/uk_politics	24	11	35	0.69	0.41	0.51	0.6	0.7	0.55
/world/americas	11	1	32	0.92	0.26	0.41	1	0.6	0.49
/education	16	1	19	0.94	0.46	0.62	1	0.77	0.72
/world/africa	27	2	12	0.93	0.69	0.79	0.9	0.94	0.87
/scotland	39	3	64	0.93	0.38	0.54	1	0.97	0.65
/england/london	16	2	14	0.89	0.53	0.66	1	0.56	0.59
/magazine	0	0	6	0	0	0	0	0	0
/sport/football/internationals	3	2	15	0.6	0.17	0.26	0.6	0.27	0.5
/health	18	1	23	0.95	0.44	0.6	1	0.82	0.71
/sport/motorsport	11	0	0	1	1	1	1	0.37	1
/sport/motorsport/formula_one	8	4	0	0.67	1	0.8	0.7	0.23	0.78
/england/manchester	0	0	9	0	0	0	0	0	0.02
/technology	8	2	19	0.8	0.3	0.44	0.8	0.37	0.48
/sport/football/eng_div_1	2	0	3	1	0.4	0.57	0.4	0.17	0.65
/sport/cricket/england	2	4	9	0.33	0.18	0.23	0.5	0.23	0.44
/sport/football/europe	0	2	9	0	0	0	0	0.07	0.05
/sport/rugby_league	1	0	5	0	0.17	0	0.1	0.06	0.28
/sport/football/eng_prem	4	1	7	0.8	0.36	0.5	0.4	0.2	0.48
/sport/tennis	13	0	2	1	0.87	0.93	1	0.45	0.87
/sport/other_sports	7	1	3	0.88	0.7	0.78	0.8	0.3	0.84
/sport/other_sports/cycling	4	0	0	1	1	1	0.4	0.13	1
/sport/athletics	0	0	2	0	0	0	0	0	0.01
/sport/cricket/counties	0	1	3	0	0	0	0.2	0.07	0.16
/sport/motorsport/motorbikes	3	0	0	1	1	1	0.3	0.1	1
/sport/boxing	3	1	3	0.75	0.5	0.6	0.6	0.31	0.66
/sport/other_sports/horse_racing	0	0	2	0	0	0	0	0.03	0.04
(Totals)	1329	159	826	0.72	0.49	0.56	0.70	0.51	0.62

Table 12: Detailed retrieval results for the cross-media configuration: mean precision at 10, mean precision at 30 and mean average precision. Note that FP < FN since low confidence predictions are discarded, in other words, the classifier may not output a prediction for every document.

text alone, and can actually result in lower precision, as is the case with

Table 13: Summarized retrieval results for the individual media, multimedia and cross-media configurations: mean precision at 10, mean precision at 30 and mean average precision.

	MP@10	MP@30	MAP
Images	7.27	5.70	4.68
Text	70.90	51.07	61.66
Multimedia	70.23	51.22	61.68
Cross-Media	73.18	52.02	62.30

measuring precision at ten. This does not happen with the proposed cross-media correlations method, which can be interpreted as a feature-weighing scheme that boosts the importance of text related to the image for learning the models. The precision-recall graph on Figure 25 provides another view on how the cross-media approach performs. It is interesting to see that cross-media is much better at the beginning of the rank (recall < 10%), which is where most users look at [93]. Also, at the mid-range of the rank, it can be seen that cross-media is better than just text and the concatenation of text and images. Moreover, it is very encouraging to see how image data contribute to the cross-media results, despite the fact that image-only results are so low when compared to the other results. This strengthens the hypothesis and gives evidence that cross-media correlation is an important aspect to take into account when building classification models for multimedia documents.

4.3 RELATED WORK

Several systematic studies comparing different systems and classes of features for the Entity Extraction task can be found in the literature. In fields related to EE, such as Relation Extraction, Word Sense Disambiguation and Semantic Role Labeling, other similar comparative studies exist. In this section, the studies that are the most relevant to the work presented in this chapter are reviewed.

Borthwick et al. [24] present a study that combines orthography, lexical, and dictionary-based features into a maximum entropy approach to person name recognition. It was one of the first purely statistical systems to make use of features from different knowledge sources. Zhang and Johnson [173] investigate the impact of various local linguistic features for named entity recognition on the CoNLL-2003 [52]

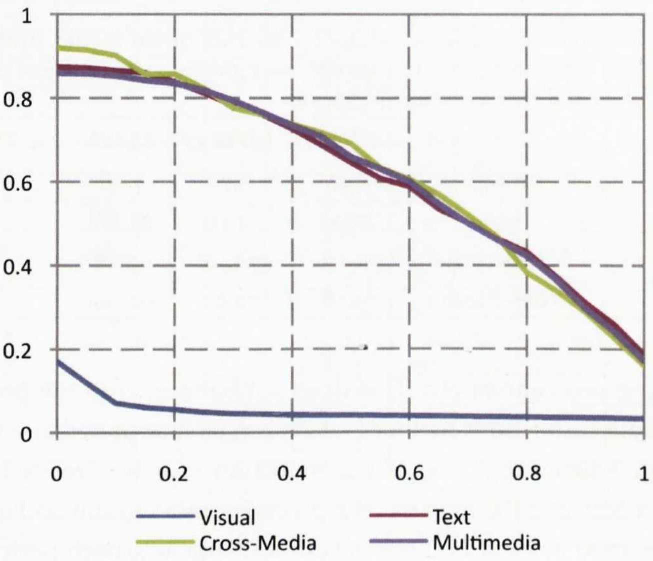


Figure 25: Precision-recall graphs for the individual media, multimedia and cross-media configurations.

Feature engineering studies for Entity Recognition

datasets. The authors showed that system accuracy could be significantly improved by using some relative simple token-based features that are available for many languages. They concluded that although more sophisticated linguistic features will also be helpful, they provide much less improvement than might be expected. This is in line with the conclusions drawn in this chapter from the results presented in subsection 4.1.3.

The task of entity recognition from research papers was tackled by Peng and McCallum [136] using Conditional Random Fields. The work consists of an empirical exploration of several factors, including variations on Gaussian, Laplace and hyperbolic-L1 priors for improved regularisation, and several classes of features. The authors motivate the need for this study by arguing that "while the basic theory of CRFs is becoming well-understood, best-practices for applying them to real-world data require additional exploration", which is a similar motivation for the work presented in this chapter and also for the work presented in chapter 5.

Rosenfeld et al. [143] describes a systematic comparison of feature-rich probabilistic classifiers for NER tasks. The authors motivate their work by stating: "[...] performance of a feature-rich classifier strongly depends upon the feature sets it uses. Since systems developed by different researchers are bound to use different feature sets, the differences in performance of complete systems cannot reliably teach us about the qualities of the underlying algorithms. [...]". This is in many ways the converse of the motivation for the work in this chapter, cf. chapter 3 — in their work, Rosenfeld et al. make features invariant and compare the models. They compare the performances of three models: MEMM, CRF, both reviewed in subsection 2.2.2, and the Regularised Risk Minimization (RRM) model [173], within the same platform and using exactly the same set of features. They also study the effects of different training sizes, different choice of parameters, and different classes of features on the performance of the systems. The experiments indicate that CRF outperforms MEMM for all datasets and classes of features, which expected since CRF is a better model of sequence labeling, but, surprisingly, the RRM model performs at the same level or even better than CRF, despite being a local model like MEMM, and being significantly simpler to build than both CRF and MEMM. Likewise, the system in subsection 4.1.7 is simpler than the systems it was compared against, but, in spite of that, obtained competitive results.

Comparison of learning models

In the closely related field of Relation Extraction (RE), Jiang and Zhai [90] systematically explored a large space of features for the RE task by evaluating the effectiveness of different feature subspaces. The authors make an argument similar to that of chapter 3, that "there has not been any systematic exploration of the feature space for relation extraction, and the choices of features in existing work are somewhat arbitrary.". And, analogously to the experimental results presented in subsection 4.1.3, their results show that using basic unit features only is generally sufficient to achieve state-of-the-art performance, while over-inclusion of complex features may decrease system accuracy. A combination of features from different levels of complexity, coupled with task-oriented feature pruning, yields the best accuracy in their case.

Feature engineering studies for Relation Extraction

Similar work is described in Fayruzov et al. [54], in the biomedical domain. The authors study the effect of various lexical and syntactic features in the protein interaction extraction task. Again, they argue in a similar fashion that "most approaches for protein interaction mining from biomedical texts use both lexical and syntactic features. However, the individual impact of these two kinds of features on the effectiveness of the mining process has not yet been thoroughly studied". A SVM-based system is employed to evaluate the different features on five benchmark datasets. The results of the study indicated that using a method that exploits a very rich feature set is not significantly better than — in fact, it does not even consistently outperform — a stripped-down version of the same method, which uses basic features only. This is similar to the conclusions arrived at in subsection 4.1.4.

The paper by GuoDong et al. [75] presents a study on incorporating diverse lexical, syntactic and semantic knowledge in feature-based relation extraction using SVM. Like what was done in section subsection 4.1.7, their study enabled the design of a system that incorporated diverse features and outperformed previously best-reported systems on the ACE [2] relation extraction tasks. Related work in the same vein is that of Buyko et al. [30] and Kambhatla [95]

Feature engineering studies in other related fields

Finally, feature engineering has also been object of attention in other fields. For example, Wang et al. [168] explore the effect of various features in the Semantic Role Labeling task, while Mohammad and Pedersen [123] study the effect of combining several lexical and syntactic features in the Word Sense Disambiguation task.

Several approaches to extract structural information from PDF, HTML and other structured multimedia document types can be found in the literature, see [101] for an overview. The approaches by [7][43][144] are based on templates that characterize each part of the document. These templates are either extracted manually or semi-automatically.

Approaches based on structural extraction

Rosenfeld et al. [144] devised a learning algorithm to extract information (author, title, date, etc.) that relies on a general procedure for structural extraction. Their proposed technique enables the automatic extraction of entities from the document based on their visual characteristics and relative position in the document layout. They ignore text content and only use features such as fonts, physical positioning and other graphical characteristics to provide additional context to the information. Like the latter, the approach here presented is based on a set of heuristics that extract and preserve all structure information. But, in contrast to these approaches, I implement an additional cross-media analysis step aimed at discovering associations between images and paragraphs in the text.

Approaches to combining classifiers

Another problem that I tackle in this chapter is how to build a single classifier from the low-level features originating from the different single-medium elements of the multimedia document. Previous approaches use co-training [20] or ensemble algorithms [94] that train different classifiers on single-medium feature vectors and combine the classifiers through a voting scheme to produce a single classifier with better accuracy. More recent approaches concatenate the single-medium feature vectors into a single cross-media vector, see [113]. Although I also concatenate single-medium features into a single feature vector, the proposed approach differs from the previous ones because I construct the cross-media feature vector by also taking into account the confidence that a given text-image pair are associated, in the form of weights affecting the text tokens that participate in the identified relation.

Tasks other than classification

The idea of using features from text and images has also been applied to tasks other than classification. For example, [10] use a generative hierarchical model for clustering image collections which integrates semantic information provided by associated text and visual information provided by image features. The data is modeled as being generated by a fixed hierarchy of nodes, with leafs of the hierarchy corresponding to clusters. The work in [33] combines textual and visual statistics in a single index vector for content-based search of a web image database. Textual statistics are captured in vector form using Latent Semantic Indexing (LSI) based on text in the containing HTML document, while visual statistics are captured in vector form using colour and orientation histograms. The authors show that the combined approach allows improved performance in conducting content-based search. In addition to text and image features, this work makes use of cross-media correlations.

Classifying images using multimedia features

Other related classification tasks include classifying images with the help of text. For example, [56] develop an image annotation model on a dataset of pictures naturally embedded into news articles and show that using captions as a proxy for annotation keywords can remove the overhead of manual annotation, and also demonstrate that the news article associated with the picture can be used to boost image annotation performance. The task here presented is the inverse, that is, I classify documents with the help of images.

4.4 CONCLUSIONS

I presented a thorough study that measured the impact in system accuracy of the several classes of features and model parameters used in the boundary classification approach to the Entity Extraction task. The study, presented in detail in subsection 4.1.2, is a valuable guide to other researchers in itself, contributing to a clarification of what constitute successful features for EE. But identifying these successful features and parameters also enabled the design of a system competitive with the state-of-the-art in this chapter. The system was described in subsection 4.1.7. Despite being significantly simpler than the systems against which it was compared, the new system was shown to be very competitive.

As explained in chapter 3, the focus of the study is on the feature engineering aspect of EE research, adopting a simple formulation of the learning model and a simple learning algorithm — in chapter 5, a study that uses more advanced algorithms for EE is presented. The study was motivated by the need to run the several systems under the same experimental conditions, since reported results in the literature are seldom comparable. To help the reader understand what is involved in ensuring invariant experimental conditions, in section 2.3 a number of concepts were introduced: gold standard, F-measure, micro- and macro-averaging, cross-validation methods, leniency in what constitutes an entity match, and a few other relevant concepts.

In subsection 4.1.2, the details about the conducted experiments were given. Experiments were designed for measuring the effect of combining classes of features, the effect of enhancing them, the effect of space and newline tokens, the effect of the length of the token window and the effect of using feature ranking metrics. The systems ran over two standard datasets for EE, which were described in subsection 4.1.1. The experimental results were presented in subsection 4.1.3.

According to the discussion and an analysis of the errors made by the system, presented in subsection 4.1.4, the study reveals that slightly different combinations of feature types provide the best results on each of the datasets, but that, in general, the use of very simple features is surprisingly enough to account for most of the accuracy obtained. It also shows that the use of rich data resources, such as gazetteers, greatly contributes to the best observed results and it is more likely to explain the differences in the results reported by several systems in the literature than the design decisions relative to the learning model. Further, the high regularities in the document formatting of the SA corpus determine that newline tokens should be used, while the difficulty in generalising patterns over the WCFP corpus means that spaces should particularly be avoided. The various entity classes related to dates and times in WCFP benefit from a gazetteer that discriminates well in those categories, while a class like speaker boosts the results on SA when a gazetteer that contains substantial data about person names is used.

The results obtained for feature selection show that even the simple frequency metric can greatly reduce the number of features with no significant loss in terms of accuracy. However, in contrast with the application of feature selection to the TC task, where the use of some of the metrics is known to consistently improve accuracy, there was no observed significant improvement in the accuracy by any of the metrics in the EE task.

The contribution to the state-of-the-art in this chapter consisted of a novel class of features for the task of classifying multimedia documents. The main advantage of the proposed approach is that it makes almost no assumptions about the way multimedia content is modelled, and is thus widely applicable. I introduced the proposed multimedia document categorization framework, which exploits document layout and the relationship between the different media comprising the document, and described how the cross-media correlation nodes can be computed. I showed that, by preserving not just text and images but also the cross-media correlations between text elements and the images in a multimedia document, it is possible to improve system accuracy, with respect to traditional approaches that ignore cross-media correlations.

The main contributions in this chapter can thus be summarized as follows:

1. Through a thorough feature engineering cycle, I designed a EE system competitive with the state-of-the-art, which, despite being significantly simpler than the systems against which it was compared, achieved comparable or better accuracy.
2. I introduced a canonical document representation graph that integrates data coming from heterogeneous formats and media. This representation is designed such that it is able to accommodate, for every supported document format, just enough information to allow the inference algorithm(s) to run.
3. I proposed a novel method for detecting cross-media associations and quantifying the level of image and text block correlation.
4. I reported on experimental results on a Web news dataset, which show that the proposed cross-media approach, by exploiting features from more than one media, yields an improvement over the results obtained by the corresponding single-medium tasks.

BLANK PAGE
IN
ORIGINAL

5

OVERCOMING THE SCARCENESS OF LABELLED DATA

Annotating data is a task that adds a burden to the development and maintenance of ML-based text mining solutions, and is very often perceived as "low priority" or even "a waste of time" by domain experts. It introduces the analogous of the "the chicken or the egg" dilemma in this domain, i. e., poor models will hardly convince experts of the value of ML-based text mining solutions, but the models will not improve without the invaluable support of those very same experts.

In this chapter, I propose two novel methods to overcome the scarceness of labelled data and turn the development and maintenance of ML-based text mining less demanding a process for domain experts. The chapter addresses the second research question in this thesis, recall from chapter 1:

> How to reduce the amount of labelled examples needed to train machine learning-based document classification and information extraction systems, both when creating them as well as maintaining them?

Second research question

I begin by presenting an experimental study designed to take advantage of unlabelled data often plentifully available for the EE task. The method used in the study is borrowed from the TC task (see section 5.3). Unfortunately, while there are plenty of studies on the use semi-supervised learning techniques in the TC task, little is documented in the literature concerning their use in the EE task. Moreover, equally difficult to find are studies on real-world use cases, such as the one presented in this chapter, where I investigate the impact of using weaker supervision requirements in the task of recognizing entities from highly technical documents in the jet engine manufacturing domain. An interesting question is thus whether the application of semi-supervised learning in EE and for this domain leads to comparable accuracy to standard techniques while using less labelled examples.

The caveat with using semi-supervised learning in general is that, in order to make it possible for domain experts to spend less time in

Trade-off

labelling the training data, text mining practitioners and system designers need to spend a greater amount of effort in studying successful features and models, because, as we saw in section 2.6, semi-supervised approaches make certain assumptions about the data. For the experiments conducted for the tasks presented in this chapter, the solid feature engineering groundwork is already laid by the lessons learned in chapter 4.

Later in the chapter I propose two novel methods that use feature labelling, as opposed to instance labelling, in adapting an existing model to a related domain. Recall from section 2.6 that the feature labelling paradigm is particularly appealing for the domain adaptation task because it is often possible for domain experts to tell which features from the source domain are expected to apply robustly also in the target domain, and doing so is considerably less time consuming than labelling instances.

5.1 SEMI-SUPERVISED ENTITY EXTRACTION ON A REAL-WORLD USE-CASE

In this section, I present a comparison between several machine learning approaches to extracting knowledge from reports about jet engines. The problem is approached as an EE task similar to that of chapter 4, but here the focus is on studying the effect of the learning algorithm rather than the features used. The goal is to understand the impact of the chosen learning algorithm in the EE system's accuracy and training time.

Furthermore, the work here presented reports on a complete IE workflow applied to a real-world problem. The workflow consists of: a design stage, in which the target knowledge to extract is identified; several interviewing sessions with domain experts in order to obtain labelled data; the implementation of a suitable IE system; validation of the first results; and refining the system accuracy through several iterations. As such, this chapter illustrates well the difficulty in obtaining labelled data when working in real-world settings.

I show that the application of a semi-supervised approach provides an increase in accuracy in this domain, and that the application of a large-scale approach considerably reduces training time while keeping accuracy comparable to the standard supervised approach. I conclude that both are good choices for this class of application scenarios, offering a trade-off between computational requirements and accuracy.

5.1.1 *The Use Case*

As discussed in chapter 1, in large organisations data regarding activities and tasks are routinely stored in an unstructured manner, in the form of images and natural language used in e-mails, word-processed documents, spreadsheets and presentations. A aerospace manufacturer in the UK, for example, periodically strips jet engines for maintenance and checks for potential issues, each time creating a so-called *strip report* which describes the analysis of the technical issues eventually found on the stripped engine. Over time, large unstructured data repositories are formed that enclose valuable knowledge for the organisation.

Real-world motivating scenario

Throughout the life cycle of a product such as a jet engine, the manufacturer is interested in validating the level of performance of the product, identifying design shortfalls and understanding, evaluating and, where required, taking corrective action to address the potential problems. The main need in this scenario is to be able to access historical data contained in the aforementioned repositories in order to find solutions that have been applied to previous design issues. The issues identified on one product are also a valuable source of lessons for the design phase of subsequent products and the operational phase of other existing products. A challenging research issue is thus to consider how the knowledge is spread across numerous sources, and how it can be captured and retrieved in an efficient manner.

Unstructured information repositories

Unfortunately, traditional IR techniques not only tend to underperform on the kinds of domain-specific queries that are typically issued against these unstructured repositories, but they are also often inadequate. Keyword-based search does not work particularly well because relevant data tend to be distributed geographically, and, consequently, organisational and language (terminology) differences magnify the barriers to retrieval. For example, some of the terminology employed to describe certain components and phenomena varies across the organization's several engineering sites around the globe, and, even within the same team, the use of abbreviations, misspellings and the adoption of slightly different alternative names poses challenges for retrieval.

The need for IE

Furthermore, the inadequacy of standard IR stems mainly from the fact that most systems return documents in response to a query, but engineers are usually looking for knowledge on a particular issue which crosses document boundaries. Discovering that knowledge is an expensive, error-prone, inefficient and time consuming manual process, aggravated by the need to repeat large portions of it for every new issue that arises. For example, upon the observation of an uncommon type of damage to the leading edge of the engine blades, an engineer may want to know if there is a correlation between the observed damage and the engine model. Currently, she needs to try several queries using the possible terms (that come to mind) to express that particular type of damage and model, inevitably forgetting or not knowing about some of those terms, and obtaining in return documents that need to be further browsed for obtaining a confirmation of the answer.

The ability to query unstructured repositories as if they were structured would thus effect significant improvements to the way teams work in scenarios such as these, which are becoming commonplace as organisations more and more regard knowledge as their most valuable asset.

5.1.2 *Learning Algorithms*

The experiments in this chapter keep the choice of features constant and vary only the learning algorithm in the EE system setup. Compared are a standard SVM implementation (as representative of supervised approaches), a graph label propagation algorithm (as representative of semi-supervised approaches) and a stochastic gradient descent variant of SVM (as representative of what I call "large-scale" approaches[1]). The classical SVM formulation was already presented in section 2.4. In what follows, I briefly describe the label propagation and the stochastic gradient descent algorithms.

Graph-based semi-supervised methods

The label propagation algorithm is a simple and effective semi-supervised algorithm that belongs to the family of graph-based approaches. Graph-based semi-supervised methods define a graph where the nodes are both the labelled and unlabelled examples in the dataset, and edges (may be weighted) reflect the similarity of examples. Graph methods are non-parametric, discriminative, and transductive in nature.

[1] By "large-scale" approaches I mean the family of sublinear complexity learning algorithms, designed to work over large amounts of data.

Label Propagation

In the label propagation approach, known labels are used to propagate information through the graph in order to label all nodes. The goal is to learn a labelling function satisfying two constraints at the same time: 1) it should be close to the given labels on the labelled nodes, and 2) it should be smooth on the whole graph. The geometry of the data is thus captured by an empirical graph $G = (V, E)$ where nodes $V = \{1, \dots, n\}$ represent the training data and edges E represent similarities between them. Similarities are given by a weight or kernel matrix K such that K_{ij} is non-zero iff $(i, j) \in E$.

Algorithm 1 Graph label propagation algorithm

```
Compute kernel matrix $K$
Compute the diagonal degree matrix $D$ by $D_{ii} \leftarrow \sum_j K_{ij}$
Initialize $Y^0 \leftarrow (y_0, \dots, y_l, 0, \dots, 0)$
Iterate
1. $Y^{(t+1)} \leftarrow D^{-1} K Y^t$
2. $Y^{(t+1)} \leftarrow Y_l$
until convergence to $Y^{(\infty)}$
Label point $x_i$ by the sign of $y_i^{(\infty)}$
```

Starting with nodes $1, 2, ..., l$ labelled with their known label (1 or -1) and nodes $l + 1, ..., n$ labelled with 0, each node starts to propagate its label to its neighbours, and the process is repeated until convergence. An algorithm of this kind has been proposed by [176], which I reproduce above. There, estimated labels on both labelled and unlabelled data are denoted by $Y = (Y_l, Y_u)$. Estimated labels may be allowed to differ from the actual labels.

Complexity

The algorithm is a particular form of the power iteration eigenvector algorithm, where Y is the stationary vector of the matrix K. The algorithm requires on the order of $O(kn^2)$ time, for a sparse graph where each data point has k neighbours, which can be prohibitive when working over large amounts of data.

Obtaining an inductive classifier

Even though the label propagation algorithm intrinsically works in a transductive setting, it is possible to easily obtain an inductive learner from the transductive learner [34]. Assuming that labels $y_1, \dots, y_n$ have already been computed by the algorithm above, the label of a new point x is given by:

$$\hat{y} = \frac{\sum_j K(x, x_j) \hat{y}_j}{\sum_j K(x, x_j)},$$

a simple inductive formula whose computational requirements scale linearly with the number of samples already seen. It is interesting to note that, if $K(x_i, x_j)$ is the k-nearest neighbour function, the algorithm reduces to k-nearest neighbour classification.

Stochastic gradient descent

Stochastic (or "on-line") gradient descent is an optimization method for minimizing an objective function, in which the true gradient, instead of considering all examples, is approximated by a gradient at a single example[158]. By contrast, the classical SVM algorithm of section 2.4 can be seen as optimizing a cost function which can be expressed as an average over all the training examples. Essentially, the loss function measures how well the learning system performs on each example. Computing such an average takes a time at least proportional to the number of examples.

Complexity

Stochastic gradient descent instead updates the learning model via the loss function measured for a single example. The complexity of stochastic learning descent-based algorithms is thus $O(n)$. This works because the averaged effect of these updates is identical when learning from a large amount of examples. Although the convergence is much more noisy, the sublinear computing cost for the gradient is a huge advantage for large-scale problems.

Recall from section 2.4 that in the SVM primal formulation we seek to minimize

$$\frac{\lambda}{2}||f||_H^2 + \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i f(x_i)).$$

The stochastic gradient descent approach minimizes the empirical risk by selecting a random instance at each iteration and updating f in the following way:

$$f_{t+1} = (1 - \eta\lambda)f_t - \eta l'(f_t(x_t), y_t)K(x_t, \cdot),$$

where η_t is the learning rate, λ is the regularization constant and l is the loss function. The representer theorem [44] gives us the necessary tools to move from the above function update to a vector update, by expressing the update in terms of the α_i variables of the dual, yielding

$$\alpha_t = -\eta_t l'(f_t(x_t), y_t)$$

$$\alpha_{t'} = (1 - \eta\lambda)\alpha_{t'} \text{ for } t' < t.$$

For the classical case of SVMs using "hinge" loss, as presented in section 2.4, the update function is finally

$$\alpha_t = -\eta_t y_t \mathbb{1}[y_t f_t(x_t) - 1 \leqslant 0]$$

5.1.3 *Dataset*

I gathered samples consisting of 70 "strip reports", containing 661,117 tokens in total. The reports are written by jet engine engineers using Microsoft Word. Processing this type of documents requires conversion into an open XML-based file format, the Open Document Format (ODF) format. This, unfortunately, introduces some noise in the data, since the conversion is not completely reliable. Fortunately, token sequences are preserved, essential for the EE task.

Jet engine reports

The ODF data is in turn represented into a canonical data format defined by the Runes data processing framework I have created (see Appendix A), in order to make it compatible with the NLP pre-processing toolset (tokeniser, part-of-speech tagger, etc.), wrapped as a set of plug-ins in the same framework.

Pre-processing

5.1.4 *Labelling Data*

Labelled data for this task consists of annotations of token sequences (of arbitrary size) from the documents. The annotations associate the token sequences to one or more entity types defined in some ontology. The multimedia annotation tool AktiveMedia[2] was extended to support working with ODF documents. In AktiveMedia, users can annotate by

Annotation tool

[2] http://sourceforge.net/projects/aktivemedia/

selecting text or a region of an image and assign a given concept or relation from the ontology to their selection.

Domain ontology

An ontology of the problem domain was defined together with the domain experts. The so-called "jet engine domain ontology" contains roughly 400 concepts characterizing the problem domain, organized in a hierarchical fashion and featuring some relations between concepts. The list of classes to use for our EE classification task correspond to the concepts in this ontology.

First annotation effort

With the goal of producing a lasting metadata resource associated with the strip reports corpus, which may potentially be re-used in the future in the context of other tasks, I conducted two annotation efforts. In the first annotation effort, I asked non-experts to mark the text against the full ontology. However, in doing so I faced two major problems: the annotators lacked the required expertise to fully comprehend the highly specialised contents of the documents; and the sheer size of the ontology made it very hard to keep in mind all possible concepts when analysing a piece of text.

Second annotation effort

Given the lack of confidence in the metadata resource produced during the first annotation effort, a second iteration addressed both issues by involving expert users in the process and by restricting the ontology to a subset of the concepts, determined in a series of meetings with the experts. As a result, a total of 34 concepts were deemed as the most useful for this task.

Concepts annotated

In an analysis carried out after the second annotation effort, I verified that some of these concepts did not actually provide enough support in the corpus so as to enable learning. For that reason, any ontology concept with less than 10 occurrences in the corpus was removed. A more compact list of 15 ontology concepts became the final list of classes for evaluation purposes. The list is shown in Table 14.

Engine	Tube
Engine Module	Groove
Engine Serial Number	Ring
Module Serial Number	HP Compressor
HP Turbine	IP Turbine
LP Turbine	Observed Damage
Document Title	Date
Customer Number	

Table 14: The list of ontology concepts used in the evaluation.

The metadata obtained from the annotation sessions amounted to a total of 2498 annotations. Of these, 259 were detected and discarded as invalid by the system, as they span parts of tokens only. The adopted token-level model (as opposed to a, e. g., character-level model) does not support such kind of annotations. On manual inspection, the majority of those annotations were due to mistakes made during the annotation process; concretely, they resulted from dragging the mouse over the text but not covering the whole of the token as intended while using AktiveMedia.

Semantic annotations acquired

5.1.5 *Dataset Generation*

I take the approach commonly found in the literature of decomposing the multi-class classification problem in this EE task into multiple independent binary classification problems using a one-vs-all approach. Thus I generate one dataset per class.

One-vs-all approach

The generation of examples (nodes in the graph) for the semi-supervised algorithm differs from the other two in that only tokens within the window around a positive example are labelled as negative, while all other tokens are kept unlabelled. This results in model with less assumptions – concretely, in this model tokens that have mistakenly not been labelled are not considered to be negative examples, which intuitively seems better.

Generation of dataset instances

The dataset used to run the experiments for the graph-based semi-supervised learning algorithm therefore contains only a few thousand labelled examples out of the several hundred thousand unlabelled ones, whereas the datasets used to run the experiments for the supervised and large-scale learning algorithms follows the traditional EE problem formalisation as described in chapter 4, and therefore contain the full several hundred thousand instances as labelled ones. Another notable difference between the algorithms' datasets is that the graph-based algorithm runs in a transductive setting, and thus needs to have access to the (unlabelled) test data together with the training data in order to run.

I follow the feature generation procedure described in chapter 4. For each token I consider a window of size 5 to each side, to capture its context. In the experiments here reported I used the token string itself, plus the token stem, the part-of-speech tag, and the orthography of the

Features

token as features. The NLP tools employed come from the OpenNLP project[3].

5.1.6 *Learning Approach*

I designed three system configurations, which vary only in the learning algorithm employed, for comparison purposes:

SUPERVISED LEARNING : using the widely used off-the-shelf `libsvm`[4] support vector machine toolkit, selecting the linear kernel and setting the penalty parameter of the error term C to 10 (determined via cross-validation).

SEMI-SUPERVISED LEARNING : using my implementation of the graph label propagation algorithm described in [177]. I set the number of neighbours k to 5.

LARGE-SCALE LEARNING : using my own port to Java of the stochastic gradient descent SVM learner originally written by Leon Bottou[5], using the default parameters – 5 iterations, a lambda of 10^{-4}, and the hinge loss function. The lambda parameter was chosen through cross-validation.

Due to the high dimensionality of the feature space, I use the linear kernel $K(x_i, x_j) = x_i^T x_j$ in all the configurations.

The rationale is to understand whether it is possible to achieve gains, either in terms of accuracy or speed, or both, by using alternative learning algorithms.

5.1.7 *Experimental Setup*

Splits

All experiments were performed adopting a 5-fold cross validation over the 70 strip reports, therefore selecting 56 reports for training and leaving 14 reports for testing on each fold. Moreover, all experiments were conducted on a 2Ghz laptop with 1Gb of RAM running Java SDK 1.5. The running times reported for comparison pertain to this setting.

For the training phase, I report the total time to learn the models only, on all classes and all documents, that is, not considering the time

[3] http://opennlp.sourceforge.net/

[4] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[5] http://leon.bottou.org/projects/sgd

taken by any other pre- or post-processing steps, in order to emphasize on the comparison of learning algorithm performance.

For the testing phase, I instead report the average time taken by the whole process to run over one document, including all processing steps, in order to give an idea of how fast the system runs when invoked to extract information from a document.

5.1.8 *Results*

I measured the accuracy of the system using standard measures for information extraction, namely precision, recall and F-measure. Recall their definitions from section 2.3. As in chapter 4, for this task a true positive occurs strictly when there is a perfect match between the predicted tokens and the solution tokens, otherwise a false positive is counted. All the quantities presented for the total are micro-averaged.

In the supervised configuration, the learning algorithm trained in 4m 58s, and the system tested in a total of 924.14s over the 5 folds, which means it took 13.2s on average per document. Table 15 presents the detailed results per entity type.

Entity Type	Precision	Recall	F_1
Engine Module	0.78	0.71	0.74
Engine Serial Number	0.79	0.63	0.70
Customer Number	0.72	0.67	0.69
Module Serial Number	0.81	0.71	0.76
Date	0.69	0.69	0.69
Document Title	0.70	0.58	0.63
Observed Damage	0.81	0.58	0.68
HP Compressor	0.71	0.54	0.61
IP Turbine	0.93	0.62	0.74
LP Turbine	0.68	0.80	0.74
Tube	0.71	0.68	0.69
HP Turbine	0.64	0.67	0.65
Ring	0.47	0.74	0.57
Groove	0.71	0.38	0.5
Engine	0.25	0.08	0.12
Total	0.75	0.65	0.70

Table 15: Detailed results per class for the supervised configuration.

In the semi-supervised configuration, the system run in a total of 2h 11m. The system was tested in a transductive setting, thus it does not make sense to speak of training and testing phases in this case. This does not affect the conclusions. Table 16 presents the detailed results per entity type.

Entity Type	Precision	Recall	F_1
Engine Module	0.80	0.74	0.77
Engine Serial Number	0.80	0.63	0.71
Customer Number	0.72	0.70	0.71
Module Serial Number	0.81	0.81	0.81
Date	0.86	0.67	0.75
Document Title	0.80	0.52	0.63
Observed Damage	0.73	0.66	0.69
HP Compressor	0.62	0.59	0.61
IP Turbine	0.89	0.57	0.69
LP Turbine	0.75	0.67	0.71
Tube	0.81	0.61	0.70
HP Turbine	0.52	0.88	0.65
Ring	0.40	0.82	0.54
Groove	0.55	0.50	0.52
Engine	0.60	0.21	0.32
Total	0.77	0.66	0.72

Table 16: Detailed results per class for the graph-based semi-supervised configuration.

In the large-scale configuration, the learning algorithm trained in 28.4s, and the system tested in a total of 160.16s over the 5 folds, which means it took 2.28s on average per document. Table 17 presents the detailed results per entity type.

5.1.9 *Discussion*

It is interesting to observe that all three configurations obtain comparable F-measure values.

The semi-supervised configuration, using the graph-based label propagation algorithm, shows a slight improvement over the standard supervised configuration, which uses SVM. This empirically observed improvement is presumably due to the fact that the semi-supervised algorithm makes use of the cluster assumption on the unlabelled test data, and confirms the merits of the semi-supervised approach as reported in other works using artificial datasets [176] or real-world datasets but different tasks, such as document classification [167] or relation extraction [36]. However, the graph-based semi-supervised algorithm

Entity Type	Precision	Recall	F_1
Engine Module	0.78	0.75	0.76
Engine Serial Number	0.77	0.64	0.70
Customer Number	0.70	0.55	0.62
Module Serial Number	0.82	0.67	0.74
Date	0.68	0.65	0.66
Document Title	0.75	0.45	0.56
Observed Damage	0.79	0.55	0.65
HP Compressor	0.67	0.50	0.57
IP Turbine	0.95	0.56	0.7
LP Turbine	0.66	0.77	0.71
Tube	0.67	0.65	0.66
HP Turbine	0.58	0.58	0.58
Ring	0.38	0.58	0.46
Groove	0.60	0.23	0.33
Engine	1	0.08	0.15
Total	0.75	0.62	0.68

Table 17: Detailed results per class for the large-scale configuration.

presents two disadvantages with respect to the other two: it works in a transductive setting, therefore requiring the test data to be available together with the training data, which may not be possible in all application scenarios; and its running time exceeds by far the one of the standard configuration.

The large-scale configuration, on the other hand, which uses a stochastic gradient descent algorithm, shows very little loss in accuracy with respect to the standard configuration, but runs an order of magnitude faster than the latter. Again, this also empirically confirms the conclusions obtained in previous related work on the document classification task, namely that the stochastic gradient descent approach can be competitive with standard SVMs when the size of the training set is large. This makes it very attractive for application scenarios in which a on-the-fly analysis of the document is required.

Error analysis

Further, I decided to further investigate on what kind of errors the system was making. For that, I replaced the learning algorithm with a rote learner, that is, a learning algorithm that simply memorizes every positive instance of a class that it is presented with, producing a classifier that simply looks up the input feature vector in its "memory" in order to determine the instance class. I used the rote learner to both train and test on the whole corpus. The average results obtained were 0.79 precision, 1.0 recall, and 0.88 F-measure.

Surprisingly, the results obtained with the rote learner were far below the expected value of 100% precision. The low precision value may have at least two explanations: one might be due to human annotators failure in spotting and marking up some mentions of entities even when they have identical contexts to other rightly annotated mentions, which turns true positives into false positives, decreasing precision; a second explanation might be that the window size is not enough to discriminate between the training examples, which would also affect false positives. In an attempt to quantify which of the cases were most frequent, I checked a sample of 30 false positives, and verified that 73% of those turned out to be incorrectly assigned as false positives, that is, the system annotated correctly but there was an annotation by the human annotator missing.

Given the 79% upper limit on the precision value, the results obtained by the EE system under the three configurations do not appear to be so low as on first impression. These findings hint at the difficulty in producing high quality semantic annotations in such highly specialised technical domains. Moreover, due to the high cost of the domain experts' time, it was unfortunately not possible to measure the inter-annotator agreement, because every human annotator was given a different set of documents to annotate.

5.2 ADAPTING TO DIFFERENT DOMAINS THROUGH FEATURE LABELING

In this section[6], a novel approach to domain adaptation for text categorization is presented, which merely requires that the source domain data are weakly annotated in the form of labelled features. The main advantage of the approach resides in the fact that labelling words is less expensive than labelling documents. Two methods are proposed, the first of which seeks to minimize the divergence between the distributions of the source domain, which contains labelled features, and the target domain, which contains only unlabelled data. The second method augments the labelled features set in an unsupervised way, via the discovery of a shared latent concept space between source and target.

[6] Joint work with C. Kadar, who performed the software modifications to MALLET and carried out the experiments.

The approach proposed here outperforms standard supervised and semi-supervised methods, as will be shown, and obtains results competitive to those reported by state-of-the-art domain adaptation methods, while requiring considerably less supervision.

5.2.1 *Proposed Approach*

Rather than requiring documents in the source and target domains to be examined and labelled, the proposed approach to the domain adaptation problem leverages a small set of words that domain experts indicate to be positively correlated with each class – the labelled features. It is founded on the general GE method introduced in section 2.6.

Label regularization

Concretely, the *label regularization* technique introduced in [114] is employed. With label regularization, the constraints are expectations of model marginal distributions on the expected output labels. As such, estimated label marginal distributions $\tilde{g}_{x,y} = \tilde{p}(y)$ are used and constraints of the form $G(x, y) = \vec{1}(y)$ are considered. Model divergence from these constraints can be computed by using, for example, KL-divergence [100]:

$$G(\theta; \mathcal{U}) = -D(\tilde{p}(y) || E_{\mathcal{U}}[\vec{1}(y)p(y|x;\theta)]).$$

GE criteria for domain adaptation

In order to use GE for domain adaptation, criteria are derived that encourage agreement between the source and target expectations. Let $\mathcal{S}$ be source domain data and $\mathcal{T}$ be target domain data, both unlabelled. The model divergence for the task of domain adaptation is computed by:

$$G(\theta; \mathcal{S}, \mathcal{T}) = -\sum_{i \in F(\mathcal{S} \cup \mathcal{T})} D\left(\hat{p}(y|x_i > 0) || \tilde{p}_\theta(y|x_i > 0)\right), \qquad (5.1)$$

where F is a function that returns the set of features in the input data, $p(y|x_i > 0) = \frac{1}{C_i}\vec{1}(y)\vec{1}(x_i > 0)$ is an indicator of the presence of feature i in x times an indicator vector with 1 at the index corresponding to label y and zero elsewhere, and $C_i = \sum_x \vec{1}(x_i > 0)$ is a normalizing constant; $\tilde{p}_\theta$ denotes the predicted label distribution on the set of instances that contain feature i and $\hat{p}$ are reference distributions derived from the labelled features.

These reference distributions are estimated using the method proposed by [148]: let there be n classes associated with a given feature out of L total classes; then each associated class will have probability q_{maj}/n and each non-associated class has probability $(1 - q_{maj})/(L - n)$, where q_{maj} is set by the domain experts to indicate the correlation between the feature and the class.

Choice of regularizer

To encourage the model to have non-zero values on parameters for unlabelled features that co-occur often with a labelled feature, the Gaussian prior on parameters is selected as regulariser, since it prefers parameter settings with many small values over settings with a few large values. The combined objective function is finally:

$$\mathcal{O} = - \sum_{i \in F(\mathcal{S} \cup \mathcal{T})} D\left(\hat{p}(y|x_i > 0) \| \tilde{p}_\theta(y|x_i > 0)\right) - \sum_j \frac{\theta_j^2}{2\sigma^2}, \qquad (5.2)$$

consisting of a GE term for each for each labelled feature i, and a zero-mean σ^2-variance Gaussian prior on parameters.

First method

Two methods are now presented, which follow the proposed feature labelling approach to text categorization and the GE formulation above. As per Equation 5.1, both methods are multi-class and semi-supervised (in that they make use of the unlabelled target domain data). The first method, which we will designate as *TransferLF*, directly uses the input labelled features to derive the reference distributions $\hat{p}$ (in the way described earlier). Then, given the latter and unlabelled source and target domain datasets, it estimates the classification model parameters by using an optimization algorithm, taking Equation 5.2 as the objective function.

The second method, which we will designate as *TransferzLDALF*, is similar to the first one, but additionally aims at augmenting the set of input labelled features with new labelled features derived from the target domain data. To discover and label new features the idea is to find a shared latent concept space that captures the relation between the two domains and bridges source and target features. This can be achieved in an unsupervised manner by using latent topic models such as Latent Dirichlet Allocation (LDA) [17]; however, we are interested in encouraging the recovery of topics that are more relevant to the domain expert's modelling goals, as expressed by the labelled features provided, than the topics which would otherwise be recovered in an unsupervised way. Weak supervision in LDA was recently introduced in works such as [5, 73]. With this goal in mind, the approach in [5] is adopted and adapted to the purposes of this method: it is possible to add supervision to LDA in the form of so-called *z-labels*, i.e., knowledge that the topic assignment for a given word position is within a subset of topics. Thus, in addition to their role in GE, we use the input labelled features as *z-labels*, in order to obtain feature clusters (containing both source and target features) where each cluster respects to one topic from the set of topics found in the labelled features. It is then possible to augment the original labelled features set with the k most probable target domain features present in each cluster, in hope that the additional GE constraints lead to improved performance.

Second method

The algorithm for inducing a text categorization classifier for both methods is shown below. The first two steps only apply to *TransferzLDALF*.

```
Algorithm 2 TransferLF and TransferzLDALF
Input: labelled features $\mathcal{L}$, unlabelled source $\mathcal{S}$ and target $\mathcal{T}$ domain data
Output: induced classifier $\mathcal{C}$

TransferzLDALF only:
(1) $\mathcal{L}_{\mathcal{LDA}}$ = labelled features from weakly-supervised LDA using input $\mathcal{L}$, $\mathcal{S}$ and $\mathcal{T}$
(2) Augment $\mathcal{L}$ with k target domain features per topic from $\mathcal{L}_{\mathcal{LDA}}$

TransferLF and TransferzLDALF:
(3) Compute reference distributions $\hat{p}(y|x_i > 0)$ from $\mathcal{L}$
(4) Estimate model parameters by running optimization algorithm according to Equation 5.2
(5) return induced classifier $\mathcal{C}$
```

5.2.2 *Datasets*

The first of the datasets chosen for an empirical analysis of the proposed approach to text categorization is K. Lang's original 20-newsgroups[7] dataset [104]. It contains approximately 20,000 documents that correspond to English-language posts to 20 different newsgroups. There are roughly 1000 documents in each category. The topic hierarchy for this

Twenty-Newsgroups corpus

[7] http://www.cs.umass.edu/~mccallum/code-data.html

dataset contains four major groups: *sci* (scientific), *rec* (recreative), *talk* (discussion) and *comp* (computers), with 3 to 5 topics under each group.

SRAA corpus

The second dataset used in the experiments is the SRAA[1] corpus. It contains messages about simulated auto racing, simulated aviation, real autos and real aviation from 4 discussion groups. The first 4,000 documents from each of the classes in this dataset were used.

5.2.3 *Evaluation Methodology*

Splits

For the purposes of evaluating domain adaptation, documents were gathered such that they were drawn from related topics, having different distributions. For example, the newsgroups *rec.autos* and *rec.motorcycles* are both related to cars, whereas the newsgroups *rec.sport.baseball* and *rec.sport.hockey* both describe games. Plus, moving to the first level of the 20-newsgroups taxonomy, broader categories may also be built: recreative, talk, computers and scientific.

The SRAA data set is split in a similar manner into four categories: auto, aviation, real, simulated. Table 18 summarizes the characteristics of the datasets used in the experiments, indicating the source vs. target splits, the initial number of labelled features, and the KL-divergence [100] measuring the distribution gap between the domains[8].

Data preparation

Minimal preprocessing was applied on the data: lowercasing the input and removing a list of English stopwords. Each document is represented as a vector of words and their frequency in the corpus.

Evaluation metric

The results are presented using *accuracy* as the evaluation metric (see section 2.3). In all comparisons, care was taken to reproduce the original authors' experimental setting with rigour.

5.2.4 *Labelling Data*

Human domain expertise is replaced in the experiments by an oracle-labeller – an experimental setup also adopted in, e.g., [50]. Making use of the true instance labels, the oracle computes the mutual information of the features within each class, and, if above a given threshold, labels the feature with the class under which it occurs most often, and also with any other class under which it occurs at least half as often.

[8] It may be noted that the obtained KL-divergence values are considerably larger than if they were to be split randomly, which would yield values close to zero.

Dataset	Source Data	Target Data	KL divergence
Cars vs Games	rec.autos rec.sport.baseball	rec.motorcycles rec.sport.hockey	0.5679
Cars vs. Hardware	rec.autos comp.sys.ibm.pc.hardware	rec.motorcycles comp.sys.mac.hardware	0.4136
Cars vs Games vs Hardware vs OS	rec.autos rec.sport.baseball comp.sys.ibm.pc.hardware comp.windows.x	rec.motorcycles rec.sport.hockey comp.sys.mac.hardware comp.os.ms-windows.misc	0.4579
Cars vs Games vs Hardware vs OS vs Politics vs Religion	rec.autos rec.sport.baseball comp.sys.ibm.pc.hardware comp.windows.x talk.politics.mideast soc.religion.christian	rec.motorcycles rec.sport.hockey comp.sys.mac.hardware comp.os.ms-windows.misc talk.politics.misc talk.religion.misc	0.3701
Comp vs Sci	comp.graphics comp.os.ms-windows.misc sci.crypt sci.electronics	comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x sci.med sci.space	0.3897
Rec vs Talk	rec.autos rec.motorcycles talk.politics.guns talk.politics.misc	rec.sport.baseball rec.sport.hockey talk.politics.mideast talk.religion.misc	0.5101
Comp vs Rec	comp.graphics comp.sys.ibm.pc.hardware comp.sys.mac.hardware rec.motorcycles rec.sport.hockey	comp.os.ms-windows.misc comp.windows.x rec.autos rec.sport.baseball	0.4741
Comp vs Talk	comp.graphics comp.sys.mac.hardware comp.windows.x talk.politics.mideast talk.religion.misc	comp.sys.ibm.pc.hardware comp.sys.mac.hardware talk.politics.guns talk.politics.misc	0.2848
Auto vs Aviation	rec.autos.simulators rec.aviation.simulators	rec.autos.misc rec.aviation.student	0.8152
Real vs Simulated	rec.autos.misc rec.autos.simulators	rec.aviation.student rec.aviation.simulators	0.6532

Table 18: Characteristics of the datasets used for evaluating the proposed approach.

In the experiments, the mean of the mutual information scores of the top 100L most predictive features is used as threshold, where L is the number of classes; and $q_{maj} = 0.9$ as the majority of the probability mass to be distributed among classes associated to a labelled feature. The oracle is very conservative in practice – refer to Table 20 for the actual number of labelled features for each source domain.

5.2.5 *Optimiser and LDA implementations*

The MALLET[9] toolkit was utilized to solve the optimization problem using L-BFGS, a quasi-Newton optimization method that estimates the model parameters.

Finally, zLDA[10] was chosen as an implementation of the semi-supervised LDA method. The original labelled features are used as seeds for their latent topics and run the algorithm in its standard setup, as reported in [5]: $\alpha = .5$, $\beta = .1$, 2000 samples. Table 19 shows an example concerning the *Cars vs Hardware* experiment. The oracle identified and labelled 17 and 40 features, respectively. They all come from the source domains: *rec.autos* and *comp.sys.pc.ibm.hardware*, respectively. With these as input, zLDA identifies new associated features that are specific to the target (e.g. *bike* for *rec.motorcycles* and *apple* for *comp.sys.mac.hardware*).

Class	initial seed words	top 18 words in topic
Cars	article writes car cars wheel miles toyota honda driving engine oil engines ford rear year auto autos	writes article car good **bike** time back people cars make year thing engine **ride** years **road** work front
Hardware	advance windows disk system drives computer dx software bus mode os ibm memory machine monitor dos hardware board chip card cards ram mb pc interface vlb mhz cache ide cpu controller port modem motherboard gateway scsi video isa bios floppy	system drive problem computer work **mac** card mail **apple** software mb good time pc problems disk board bit

Table 19: Initial labelled features and discovered zLDA features for *Cars vs Hardware*.

5.2.6 *Results and Discussion*

Table 20 presents the results obtained from running the experiments on the several configurations shown in Table 18. The results presented compare against two classifiers which are induced from the source

[9] http://www.mallet.cs.umass.edu
[10] http://pages.cs.wisc.edu/~andrzeje/software.html

domain data only: a standard supervised maximum entropy classifier as a baseline, and the proposed *TransferLF* method prevented from looking at the target domain data.

Dataset	*# source labelled instances*	MaxEnt	*# source labelled features*	TransferLF on source	TransferLF	*# zLDA labelled features*	TransferLF with zLDA features
Cars vs Games	*2000*	90.3	*52*	84.7	**96.1**	*29*	92.8
Cars vs. Hardware	*2000*	90.7	*57*	88.2	**94.2**	*32*	88.7
Cars vs Games vs Hardware vs OS	*4000*	76.0	*109*	72.3	**80.9**	*60*	78.8
Cars vs Games vs Hardware vs OS vs Politics vs Religion	*6000*	67.1	*167*	63.0	69	*81*	**70.2**
Comp vs Sci	*4000*	71.8	*59*	76.1	78.4	*30*	**82.2**
Rec vs Talk	*3874*	77.9	*60*	74.3	74.5	*29*	**92.8**
Comp vs Rec	*5000*	87.9	*70*	86.1	**91.3**	*32*	86.7
Comp vs Talk	*5000*	93.3	*67*	91	**94.1**	*33*	94.0
Auto vs Aviation	*8000*	77.2	*48*	78.0	86.9	*29*	**91.6**
Real vs Simulated	*8000*	63.9	*54*	60.4	59.7	*30*	**77.7**

Table 20: Classification accuracies and the amount of labelled information (either instances or features) used in different sets of experiments. Note that for the *TransferzLDALF* method, the reported results correspond to selecting a fixed number of 18 features per topic (cf. learning curves), but the features outputted by zLDA can overlap and thus the size of the feature set used is smaller when merged.

The results show that the feature labelling approach to domain adaptation invariably outperforms the baseline non-domain-adaptation maximum entropy approach, while, in addition, greatly reduces the supervision requirements – compare the number of labelled features against the number of labelled instances used to induce the classifiers. The experiments show that this can be observed not only in the binary classification case, but also in the multi-class classification case.

The results also suggest that the semi-supervised nature of the proposed methods is a differentiating factor, since *TransferLF* using source domain data only consistently underperforms.

Table 21 and Table 22 compare the proposed approach with semi-supervised and latent semantic analysis-based techniques for domain adaptation in the literature. Transductive Support Vector Machines (TSVM) [91] are used as the baseline semi-supervised text classification approach. Refer to section 5.3 for a brief description of MMD[35] and

TPLSA[169]. It can be observed that the performance of the proposed methods is comparable with that of TSVM, which, again, is remarkable given that only a few labelled features are required to achieve that. The state-of-the-art MMD and TPLSA approaches still obtain higher accuracy in general, which is not surprising given that their supervision requirements are much greater, but it is still very interesting to see how the results obtained by the feature labelling approach remain competitive. This is important, since in many application domains the reduction of the annotation effort is an enabling factor, at the expense of a only few accuracy points.

Finally, Figure 26, Figure 27 and Figure 28 show the learning curves obtained by varying the number of labelled features input to the *TransferzLDALF* method. From these curves it is possible to obtain a deeper insight into the supervision requirements of the proposed approach. It is possible to conclude that as little as 5 features per topic are enough to achieve performances close to the plateau of the curve, as seen in some of the experiments, and that, on average, around 18 features per topic are enough to achieve top accuracy for the majority of the experiments.

Dataset	TSVM	MMD	TransferLF	TransferLF with zLDA features
Cars vs Games	87.4	94.5	96.1	92.8
Cars vs Hardware	92.5	94.6	94.2	88.7
Cars vs Games vs Hardware vs OS	75.4	82.4	80.9	78.8

Table 21: Performance comparison with [35].

Dataset	TSVM	TPLSA	TransferLF	TransferLF with zLDA features
Comp vs Sci	81.7	98.9	78.4	82.2
Rec vs Talk	96	97.7	74.5	92.8
Comp vs Rec	90.2	95.1	91.3	86.7
Comp vs Talk	90.3	97.7	94.1	94.0
Auto vs Aviation	89.8	94.7	86.9	91.6
Real vs Simulated	87	88.9	59.7	77.7

Table 22: Performance comparison with [169].

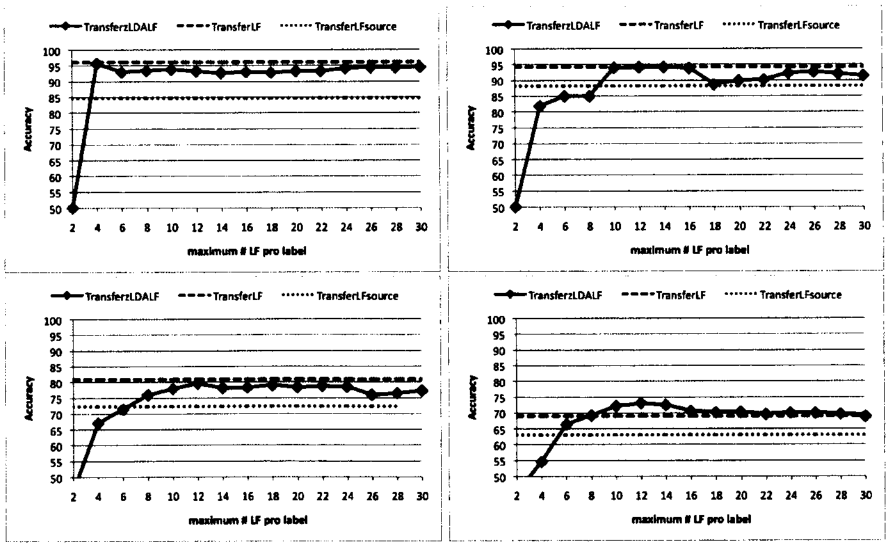


Figure 26: Learning curves for the first dataset generated from 20-newsgroups. From left to right descending: *Cars vs Games*, *Cars vs Hardware*, *Cars vs Games vs Hardware vs OS*, and *Cars vs Games vs Hardware vs OS vs Politics vs Religion*.

5.3 RELATED WORK

In the first part of this chapter, I applied EE technology to a real-world problem. Several previous projects have successfully applied IE in areas such as sale products indexing[11], job advertisement collection[12] and scientific article collection from the Internet, among several others. For example, the information extraction task in the case of sale products indexing consists in identifying the description, price and seller of the product (among other features) within the textual information found in potential product web pages. The approaches behind these systems consist of a mixture of manually-built extraction rules and machine-learning based techniques. The EE work presented in this chapter uses only the latter.

Applications to real-world problems

The experiments in section 5.1 compare a semi-supervised learning approach, namely a graph label propagation algorithm, with a standard supervised SVM approach. Semi-supervised learning methods have recently gained much attention in the machine learning literature. They have been applied to related tasks such as document classification[167] and relation extraction[36], where it has been shown that the use of unlabelled data can improve the accuracy of the learning system. Our

Semi-supervised approaches to related tasks

[11] http://froogle.google.com
[12] http://www.flipdog.com

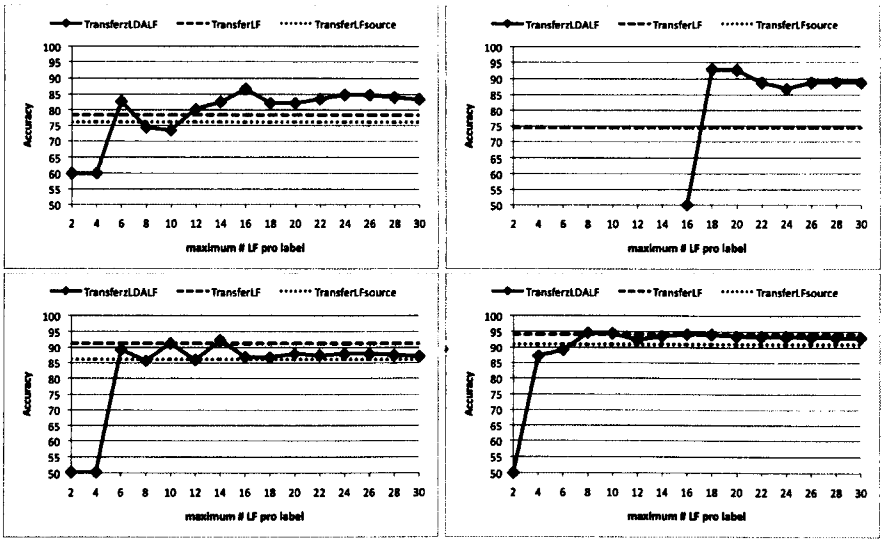


Figure 27: Learning curves for the second dataset generated from 20-newsgroups. From left to right descending: *Comp vs Sci, Rec vs Talk, Comp vs Rec,* and *Comp vs Talk.*

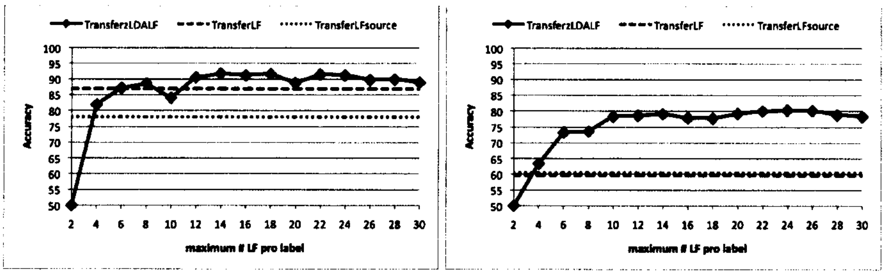


Figure 28: Learning curves for the dataset generated from SRAA. From left to right: *Auto vs Aviation* and *Real vs Simulated.*

work empirically shows that the same is true for the EE task, a conclusion that was missing in the literature, to the best of my knowledge.

Stochastic approaches to related tasks

Also included in section 5.1 is a comparison against a large-scale approach, namely using a stochastic gradient descent variant of SVM. For large data sets, on-line gradient descent can be much faster than batch gradient descent, as we have seen. Several related works in the field of document classification have shown the effectiveness of this family of methods, e. g.[154][22]. Again, the work presented here work empirically shows that this is also true for the EE task.

In the second half of this chapter, the focus turned to the problem of domain adaptation. This problem has been roughly approached in two ways in the literature: the supervised case and the semi-supervised case.

In the former, there are available labelled documents from the source domain, and also a small amount of labelled documents from the target domain. The goal is to take advantage of both labelled datasets to obtain a model that performs well on the target domain. For example, [86, 58] work under this setting. The semi-supervised case differs in that no labelled documents in target exist, therefore the goal is to take advantage of an unannotated target corpus, see, e.g., [18, 90, 169, 35]. The variant addressed in this chapter was the semi-supervised problem.

Variants to the domain adaptation problem

The problem of domain adaptation can be seen as that of finding a shared latent concept space that captures the relation between the two domains [13]. Therefore, several recent approaches sought an appropriate feature representation that is able to encode such shared concept space. [86] uses standard machine learning methods to train classifiers over data projected from both source and target domains into a high-dimensional feature space, via a simple heuristic nonlinear mapping function.

In [131], the authors approach the problem from dimensionality reduction viewpoint. The method finds a low-dimensional latent feature space where the distributions between the source domain data and the target domain data are as close to each other as possible, and project onto this latent feature space the data from both domains. Standard learning algorithms can then be applied over the new space.

Latent concept space

A probabilistic approach in the same vein can be found in [169], where the authors propose an extension to the traditional Probabilistic Latent Semantic Analysis (PLSA) algorithm [83]. The proposed algorithm is able to integrate the labelled source data and the unlabelled target data under a joint probabilistic model which aims at exploiting the common latent topics between two domains, and thus transfer knowledge across them through a topic-bridge to aid text classification in the target domain.

Other relevant approaches following the same underlying principle include the feature extraction method described in [132], the method based on latent semantic association presented in [74] and the linear transformation method in [35] that takes into account the empirical loss on the source domain and the embedded distribution gap between the source and target domains.

The domain adaptation approach presented in this chapter may also be considered to belong to the above family of approaches in that a shared latent space between the domains is modelled, but with two major differences. First, it requires only labelled features instead of labelled instances. Second, the modelling of the latent space is not unsupervised, but partially supervised instead – by taking advantage of the availability of labelled features.

5.4 CONCLUSIONS

In this chapter, I presented a study and proposed two novel methods to overcome the scarceness of labelled data. I began by presenting a successful application of IE technology to a real-world problem in the aerospace engineering domain. I argued that the use of a semi-supervised approach could yield better results under a metadata constrained setting such as this, a hypothesis which I had not seen validated in the literature for the EE task. Even though obtaining labelled data in this domain is hard due to the high cost of domain experts' time, the application of the machine learning-based technology was still successful, yielding results comparable to the state-of-the-art in other domains.

Next, I presented a novel approach to domain adaptation for text categorization that aims at reducing the effort in porting existing statistical models induced from corpora in one domain to other related domains. The approach is based on a new paradigm of labelling words (as opposed to labelling whole documents), which is less time consuming and more natural to domain experts, as argued in chapter 3.

I proposed two domain adaptation methods under this approach, in subsection 5.2.1. The first method seeks to minimize the divergence between the distributions of the source domain, which contains labelled features, and the target domain, which contains only unlabelled data. The second method is similar to the first one, but can additionally make use of the labelled features to guide the discovery of a latent concept space, which is then used to augment the original labelled features set.

The contributions in this chapter are fourfold:

1. I presented a novel approach to domain adaptation for text categorization that relies on labelled words instead of labelled documents, with the aim of overcoming the scarceness of labelled data;
2. I proposed two different methods in order to analyse the merits of the approach to domain adaptation;

3. I empirically showed that results competitive with the the state-of-the-art can be achieved with the use of semi-supervised methods in the EE task and with a low number of labelled features in the TC task; and
4. I empirically showed that the feature labelling approach, despite only using a weak form of supervision, outperforms standard supervised and semi-supervised methods, and obtains results competitive with those previously reported by state-of-the-art methods that require the classic, more expensive, form of supervision – that of labelling documents.

BLANK PAGE
IN
ORIGINAL

6

A FORMALISATION OF MULTIMEDIA MINING

In this chapter, I propose a novel formalism for declaratively specifying unstructured multimedia information mining tasks, methods and systems, with the aim of tackling the systemic, communication and replicability problems introduced in section 2.7. The chapter addresses the second research question in this thesis, recall from chapter 1:

> How to effectively represent knowledge about text mining systems and the data that they manipulate?

Third research question

The proposed formalism is based on ontology [72], and extends existing state-of-the-art formalisms for describing software and multimedia content. It enables representing DC and IE tasks, the methods that support them and the systems that implement them. It also enables representing natural language processing and machine learning subsystems, which modern DC and IE systems are typically composed of.

The key benefits offered by the proposed formalism are threefold. Firstly, it provides an agreed-upon means of referring to and describing IE tasks and systems, their inputs and outputs and their internal constituents (components, subsystems, auxiliary resources, and so on) — tackling the communication problem. Secondly, the practice of semantically describing IE systems enables cutting system development and maintenance costs down by promoting the reuse of components and supporting discovering the optimal ones for a given task, both at design time and at runtime — tackling the systemic problem. Lastly, it enables researchers to accompany published results with an unambiguous (formal) description of the IE methods and systems used in the experimental tasks that they report about — tackling the replicability problem.

The formalisation is accomplished by taking an information extraction stance on the problem, though the resulting formalism is not strictly restricted to information extraction. Rather, it is more general, since, for example, it is able to formally describe document classification just as well. However, I will refer to "information extraction" throughout this chapter for simplicity, as an alternative term to the far too verbose "unstructured multimedia information mining" expression.

6.1 REQUIREMENTS

How to approach the problem of enabling formal descriptions of IE systems? In other words, going back and re-analysing what was written in the previous section, which requirements should be drawn for the design and development of the Ontology of Information Extraction?

The identified requirements are given below:

COVERAGE Naturally, the ontology should be able to describe at least the parts of a system that correspond to the functional concerns identified earlier. Concretely, it should be able to represent the subsystems, and their respective components, related to decomposition, segment analysis, data modeling and semantic annotation.

MULTIMEDIA The proposed ontology should be able to describe IE systems that work with multimedia data. In particular, it should support text and images, and be designed in such a way that it can be extended to other types of media as well.

SEMANTIC INTEROPERABILITY An ontology of IE, much like any other ontology, must ensure that the intended meaning of the captured semantics can be shared among different systems. Reasoning processes about concepts and relations in different environments can only be guaranteed to yield identical results if the semantics is sufficiently explicitly described.

SYNTACTIC INTEROPERABILITY The semantics of the IE system descriptions are only shareable among different systems if there is some agreed-upon syntax in which to convey it.

SEPARATION OF CONCERNS Domain knowledge should be kept separate from knowledge about the IE system, as they can and should evolve separately. Moreover, as mentioned, the design and development of IE systems addresses at least four functional concerns. These should also be kept separate, if possible, in the design of ONIX.

MODULARITY Modularity is a key engineering principle which arises whenever dealing with large systems, be it software systems or

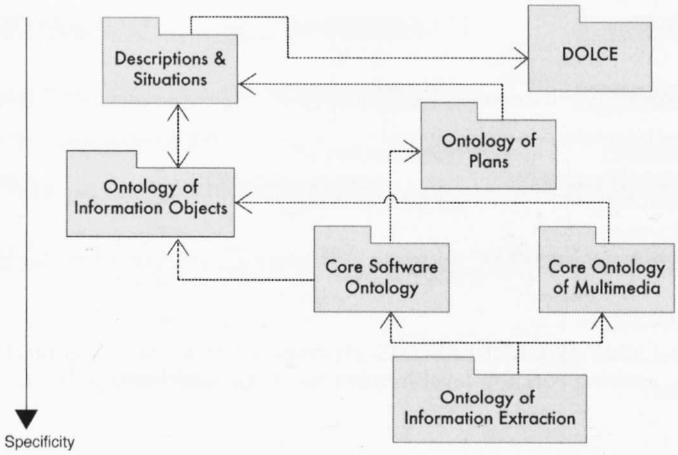

Descriptions & Situations
DOLCE
Ontology of Plans
Ontology of Information Objects
Core Software Ontology
Core Ontology of Multimedia
Ontology of Information Extraction
Specificity

Figure 29: Positioning the Ontology of Information Extraction (ONIX) in the framework of existing foundational and core ontologies. ONIX extends the Core Software Ontology (CSO) and the Core Ontology of MultiMedia (COMM).

ontologies. Since ONIX can become rather large, its design should be made modular from its inception.

EXTENSIBILITY It is expected that ONIX will be eventually extended and adapted to even more specific domains and applications. As ontology development methodologies show, ontologies are inherently incomplete, and for that reason extensibility is a key and pervasive requirement in ontology engineering [68].

6.2 PROPOSED ONTOLOGY

The design of the Ontology of Information Extraction (ONIX) leverages the richness and well-foundedness of the set of DOLCE ontologies — the Descriptions & Situations (DnS), the Ontology of Information Objects (OIO) and the Ontology of Plans (OoP) —, as well as of the Core Ontology of MultiMedia (COMM) and the Core Software Ontology (CSO). The relationship between ONIX and these other ontologies is depicted in Figure 29.

The ontology is organized into several patterns, addressing the different functional concerns identified in section 2.7, and described below with the aid of UML diagrams.

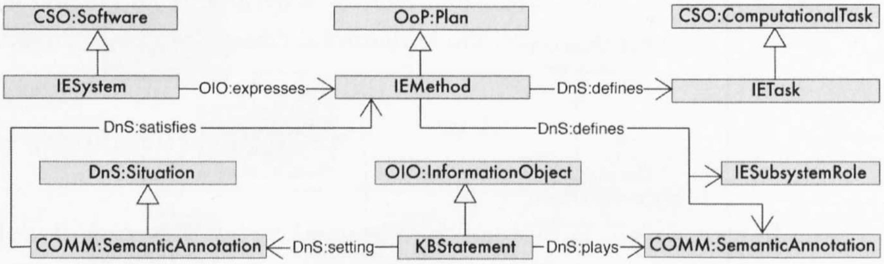


Figure 30: Ontology pattern in ONIX, represented as a UML diagram, for describing core top-level Information Extraction concepts.

6.2.1 *Information Extraction Tasks and Systems*

Let us start by characterising core information extraction concepts by answering the following questions:

> What is an Information Extraction (IE) task/system/method?
>
> What is a Knowledge Base (KB) statement?

The core concepts will be defined in terms of the DnS pattern, which will be providing context primitives to an interplay of COMM and CSO concepts. This is depicted in Figure 30.

Core top-level information extraction concepts

A IESystem is a CSO:Software that expresses a OoP:Plan, namely that of a IEMethod. A IETask is a CSO:ComputationalTask defined by the method. A statement in a knowledge base, KBStatement, is a DnS:Role, namely the COMM:SemanticLabelRole, played by OIO:InformationObject. In other words, the statement is an information object expressing a fact that provides a semantic label to some media segment — the segment which contributed to that fact being extracted or derived. A IEMethod is a DnS:Description of a situation. In the IE domain, the situation it describes is a COMM:SemanticAnnotation. This provides a DnS:setting where the statements and the CSO:ComputationalActivity sequenced by the IETask (not shown in the figure) exist.

Additionally, the IEMethod defines the IESubsystemRole, the meaning of which will become clear in what follows.

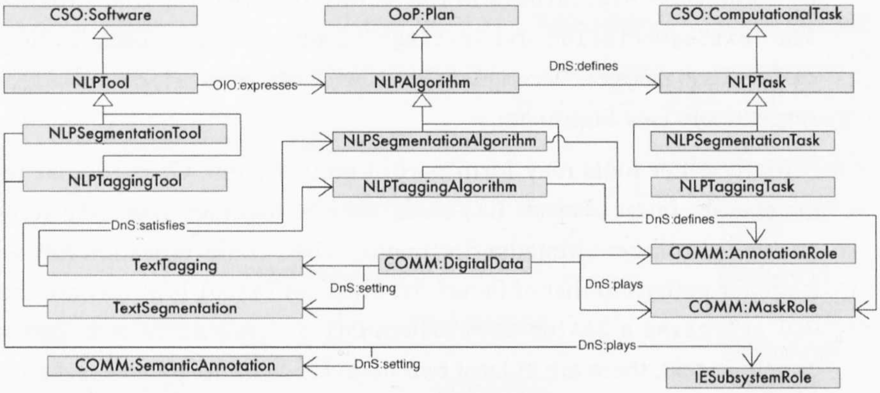


Figure 31: Ontology pattern in ONIX, represented as a UML diagram, for describing one of the possible types of IE subsystems: Natural Language Processing (NLP) tools.

6.2.2 *Information Extraction Subsystems*

Information extraction systems are complex systems composed of many subsystems, as discussed earlier. Thus, the ontology should also capture concepts related to these subsystems. Progressing down the level of detail in the characterisation of IE systems involves answering the following questions:

What is an IE subsystem?

What kinds of IE subsystems are there?

This is depicted in Figure 31 for *natural language processing* (NLP) tools, arguably the most commonly used kind of IE subsystem.

A `NLPTool` is a software that `OIO:expresses` a `NLPAlgorithm`, which `DnS:defines` a `NLPTask`. Following COMM patterns of decomposition and annotation, NLP tools can be specialised into *segmentation* and *tagging* tools. Examples of the former include the *sentence splitter*, the *chunker* and the *tokeniser*, while examples of the latter include tools such as the *part-of-speech tagger* or the *orthography tagger*. Both `NLPTaggingTools` and `NLPSegmentationTools` play the `IESubsystemRole` (defined above) in the setting of a `COMM:SemanticAnnotation`. A `NLPSegmentationTool` expresses a `NLPSegmentationAlgorithm` defining a `NLPSegmentationTask` and the `COMM:MaskRole`, which is played by some `COMM:DigitalData` in the setting of a `TextSegmentation` situation satisfied by the algorithm. Conversely, a `NLPTaggingTool` expresses a `NLPTaggingAlgorithm` which defines a `NLPTaggingTask` and the `COMM:AnnotationRole`, played

Characterising a natural language processing subsystem by some COMM:DigitalData in the setting of a TextTagging situation. The TextSegmentation and TextTagging situations are specialisations of the COMM:SegmentDecomposition and COMM:Annotation situations, respectively (see Figure 32).

Characterising an image analysis subsystem

Many other tools may form part of an IE system. Of particular relevance are *image analysis* (IA) tools, which, together with NLP tools, enable handling multimedia documents. Their characterisation follows a similar pattern to that of Figure 31. Thus, an IATool is a software that OIO:expresses a IAAlgorithm which DnS:defines an IATask. Analogously to text, there are at least two main kinds of image analysis tools, the IADecompositionTool, e.g., a region of interest classifier, and the IAAnnotationTool, e.g., an edge detector. Like NLP tools, both play a IESubsystemRole in the setting of a COMM:SemanticAnnotation.

6.2.3 *Text Decomposition*

The subsystems of an IE system are typically regarded as *blackboxes*. Unfortunately, this lack of detail makes it impossible to choose between them for the purposes of automated discovery and composition. Hence, the ontology would benefit from semantic constructs to declaratively specify the data resources and internal processes used by the subsystems. This is achieved by specialising a number of generic COMM constructs to the IE domain. Let us start by characterising *text decomposition*, by answering the following questions:

> What is a text segment (e.g., document, sentence, phrase, token)?
>
> What is the relationship between the NLP subsystem and text segments?

Characterising text segmentation

Refer to Figure 32. Text is a type of COMM:Media that realizes TextData (a specialisation of COMM:MultimediaData). A NLPAlgorithm defines a TextSegmentRole, which TextData can play in the setting of some situation. In the case of the TextSegmentation situation depicted, TextData also plays the InputTextRole and OutputTextRole roles defined by the NLPSegmentationAlgorithm. In other words, the segmentation algorithm can, for example, take a document and split it into sentences (sentence splitter), or it can take a sentence and split it into tokens (sentence-level tokeniser), or it can take a document and split it into tokens (document-level tokeniser), and so on. Two other types of outputs are

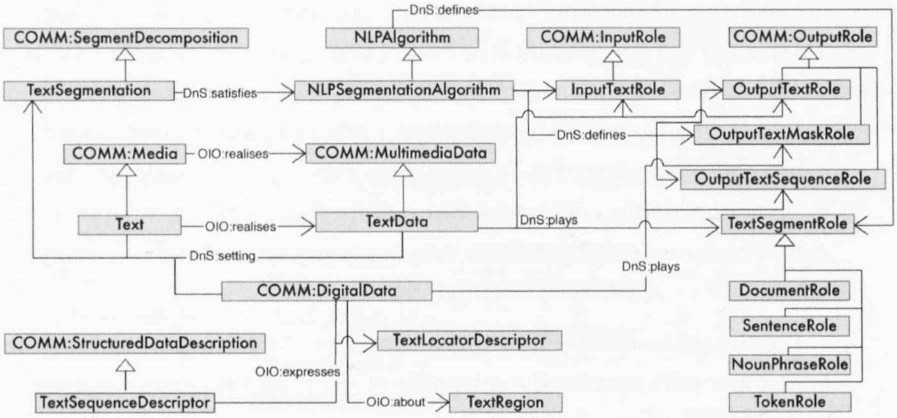


Figure 32: Ontology pattern in ONIX, represented as a UML diagram, for characterising text segmentation. This can be seen as a specialisation of the COMM decomposition pattern to the Information Extraction domain.

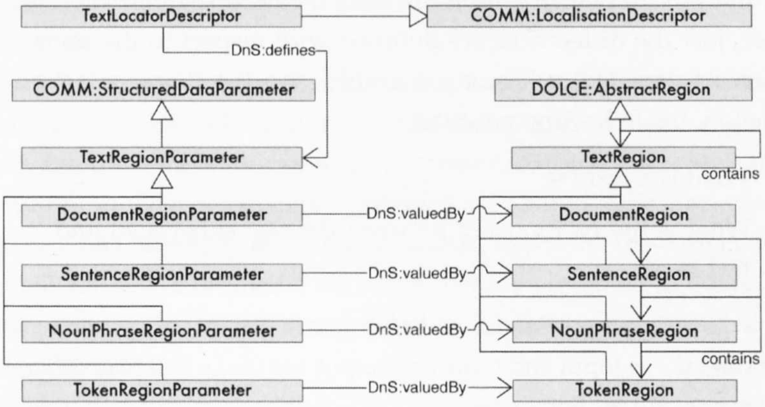


Figure 33: Specialising COMM locators in ONIX, represented as a UML diagram, for text data.

defined by a segmentation algorithm: the OutputTextMaskRole and the OutputTextSequenceRole. The former is played by a COMM:DigitalData object that expresses a TextLocatorDescriptor and is about some TextRegion, e. g.a descriptor about a sentence that provides a means of locating it in the document. The latter is played by a COMM:DigitalData object that expresses a TextSequenceDescriptor, e. g.a descriptor about the order in which tokens appear in the document[1].

[1] In practice, this descriptor is often expressed implicitly in the native programming language. For example, returning a List<Token> object in Java implicitly encodes the sequence information, since Lists are ordered collections.

Figure 33 shows the TextLocatorDescriptor and the TextRegion. The former is a specialisation of COMM:LocalisationDescriptor for text data. This descriptor enables specifying regions in the TextData manipulated by the algorithm. It defines a number of TextRegionParameters valued by their respective TextRegions. This is by no means an exhaustive list of possible subclasses, but merely illustrative of a pattern that suits a number of different concrete implementations and can easily be extended.

Characterising text locators and regions

For the decomposition of image data, the constructs introduced in COMM are sufficient for the purposes of ONIX, and thus do not need to be extended.

6.2.4 *Text Tagging*

Text analysis and tagging tools are similar to text segmentation tools. Hence, just the differences are outlined with respect to the semantic constructs depicted in Figure 32. In this case, the characterisation is guided by the following questions:

> What is a text tag?
>
> What is the relationship between the NLP subsystem and text tags?

In a TextTagging situation satisfying a NLPTaggingAlgorithm, the latter can take as input and return as output TextData that play different TextSegmentRoles. For instance, a part-of-speech tagger tags tokens, while a sentiment classifier may tag sentences. A tagging algorithm defines an InputTextRole and an OutputTextTagRole. The latter is played by a structured COMM:DigitalData object that OIO:expresses a TextTagDescriptor (a subclass of COMM:StructuredDataDescription). This COMM:DigitalData object is OIO:about a DOLCE:Particular.

Characterising text analysis and tagging

As with the decomposition pattern, there was no need to extend the constructs introduced in COMM for the tagging of image data.

6.2.5 *Models for Classification and Extraction*

Information extraction systems internally contain one or several models that guide the process of classification of/extraction from the multimedia artifacts that are passed as input. In particular, machine learning-

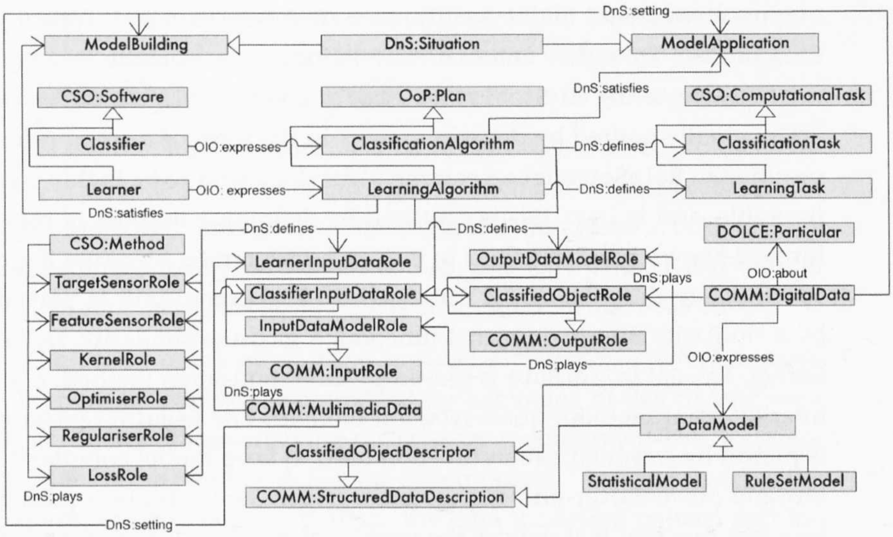


Figure 34: Ontology pattern in ONIX, represented as a UML diagram, for characterising building and applying data models.

based IE systems contain algorithms that are able to *learn* these models automatically by looking at *training data*, that is, multimedia segments with sample annotations of what to classify/extract. Then, with these acquired models of the data IE systems are able to *classify* previously unseen data automatically, in other words, to predict the correct annotations for new multimedia segments that are presented to them. This leads to the following questions which will help characterise what was just described:

What is a model of the data?

What is a learning algorithm/task?

What is a classification algorithm/task?

The ONIX ontology pattern that addresses this concern is depicted in Figure 34. There are two situations to describe in this pattern: `ModelBuilding` and `ModelApplication`. A `Learner` is a `CSO:Software` that expresses a `LearningAlgorithm`, which defines a `LearningTask` and satisfies the `ModelBuilding` situation. Conversely, a `Classifier` is a `CSO:Software` that expresses a `ClassificationAlgorithm`, which defines a `ClassificationTask` and satisfies the `ModelApplication` situation.

For the purposes of information extraction from multimedia documents, a `LearningAlgorithm` takes `COMM:MultimediaData` as input and outputs a DataModel (expressed by some COMM:DigitalData). Typically, data models are either automatically induced via machine learning methods, resulting in StatisticalModels and/or RuleSetModels, or are manually crafted by domain experts in the form of a set of rules, resulting in RuleSetModels. Learning algorithms tend to be highly configurable, and in ONIX this is captured by defining a number of roles for well-known basic functions in the machine learning literature (here semantically described by a CSO:Method). The KernelRole is played by a similarity metric or kernel function, e. g.cosine similarity, radial kernel. The OptimizerRole is played by an optimisation method, e. g., interior point method, quasi-Newton method. The RegularizerRole is played by a regularisation function used to keep model complexity low and prevent over-fitting, e. g., lasso, ridge. The LossRole is played by a loss function that defines the penalty of miss-prediction, e. g., log loss, hinge loss. The FeatureSensorRole is played by a function that extracts features from the data, while the TargetSensorRole is played by a function that determines the class which the data belongs to by consulting some oracle, e. g., manually annotated data.

Characterising a learning algorithm

Characterising a classification algorithm

A ClassificationAlgorithm takes as input COMM:MultimediaData and a DataModel, and outputs COMM:DigitalData that expresses a structured description of the classified object, a ClassifiedObjectDescriptor.

6.2.6 *Semantic Annotation*

Recall that *semantic annotation* can be described in simple terms as the process of associating a semantic *class* defined in some domain ontology with the metadata about some media or media segment. Semantic annotation in the context of information extraction is thus no different from semantic annotation in any other domain — the semantic annotation pattern defined in COMM is both simple and generic, and for that reason it can be adopted *as is* for the purposes of ONIX.

6.3 DISCUSSION

Let us check how the requirements outlined in section 6.1 are satisfied by the proposed Ontology of Information Extraction.

By carefully choosing to model ONIX on top of COMM, we ensure that ONIX *supports multimedia data*. We have shown how to specialise the core COMM constructs to support text, while COMM itself already provided the necessary constructs to support images. ONIX can easily be extended to accommodate more types of media in the same way.

All the ontologies chosen as a basis for ONIX — DOLCE, COMM and CSO — provide a rich axiomatisation of each pattern using first order logic. Moreover, through the semantic annotation pattern, our ontology can be linked to any Web-based domain ontology. These reasons fulfill the *semantic and syntactic interoperability* requirements, respectively.

Fulfillment of the requirements

The use of ontology patterns ensures a clear *separation of concerns*. Moreover, the concerns identified at the beginning of this chapter were all addressed, thus satisfying the *coverage* requirement. The IE system and subsystem patterns define the core concepts. The text segmentation pattern, the text tagging pattern, the data modeling pattern and the semantic annotation pattern, discussed in sections 6.2.3, 6.2.4, 6.2.5, and 6.2.6, respectively, each address the homonymous concerns identified.

The *modularity* requirement is satisfied, since these patterns form modules in the core of the architecture of the IE ontology. The *extensibility* requirement is fulfilled in several ways. Concretely, ONIX allows accommodating further media types, as mentioned above, but it is also straightforward to define new types of IE subsystems, new types of segment roles and regions and new types of learning and classification algorithms. The modularity and extensibility of ONIX are in great part due to the patterns being grounded in the DnS pattern, which enables adding further contextual knowledge in such a way that it will not change the patterns (mainly by defining new roles or parameters), so that legacy descriptions remain valid.

Let us now revisit the motivating examples of chapter 2 and check how the systems designed to support them can be formally described with ONIX.

The developers of the cross-media DC system have decided to use a sentence splitter and a named entity recognizer to split each document into sentences and to tag occurrences of people and location names. Further, a colour analysis tool is used to process the images in the document. Pairs of sentences and images *per* document are then passed to a classifier, whose output is interpreted in order to generates statements of the type *about_topic(document, topic)* as output to the KB. Figure 35 illustrates how to semantically describe this system.

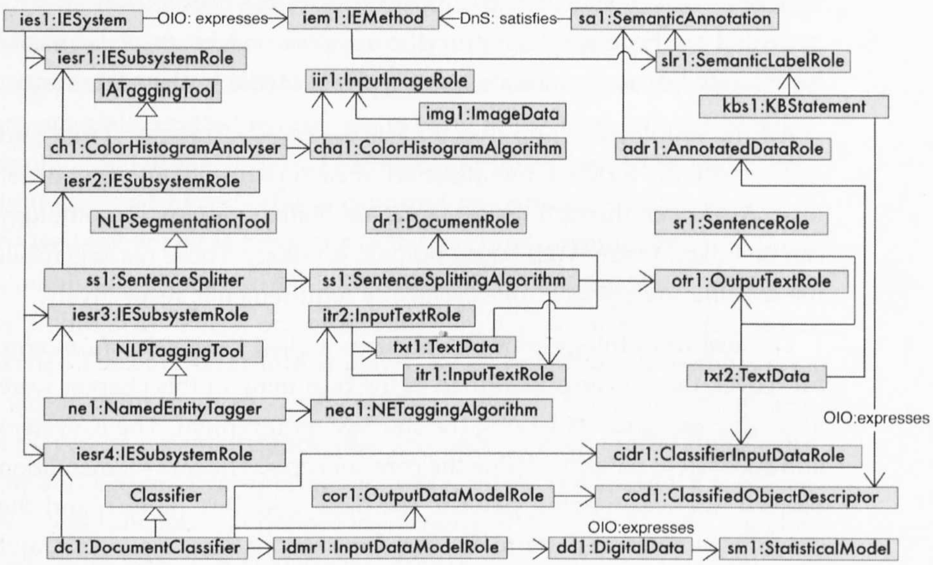


Figure 35: A UML diagram illustrating part of the semantic description, using ONIX, of a cross-media document classification system. Note that all the `DnS:plays`, `DnS:defines` and some `DnS:setting` and `OIO:expresses` labels were omitted for clarity, and, due to space limitations, the outputs of the colour analyser and the named entity tagger were also omitted.

The system defines four subsystem roles, played by a colour histogram analyser (an IA tagging tool), a sentence splitter (a NLP segmentation tool), a named entity tagger (a NLP tagging tool) and an image/sentences pair classifier (a classifier). The image analyser takes the `ImageData` from the multimedia document as input and outputs a colour histogram for the image(s). The input to both the sentence splitter and the named entity tagger is the text from the whole document (hence playing a `DocumentRole`), whereas the output of the sentence splitter is the text from a sentence (hence playing a `SentenceRole`). The sentence text also plays the `ClassifierInputDataRole` defined by the classifier. The classifier loads a previously learned document classification model as input (via the `InputDataModelRole`). The generated KB statements express the `ClassifiedObjectDescriptor` outputted by the classifier. Finally, the multimedia document plays the `COMM:AnnotatedDataRole` and the KB statement plays the `COMM:SemanticLabelRole` in this particular `COMM:SemanticAnnotation` setting, which `DnS:satisfies` the IE method `OIO:expressed` by the system.

Motivating scenarios revisited

The system that implements the entity recognition task features a similar description. It defines six subsystem roles, played by a sentence splitter, a tokeniser, an orthography tagger, a part-of-speech tagger, a named entity tagger, and a classifier of tokens. The part-of-speech tagger works over input sentences, therefore it takes as input `TextData` that plays a `SentenceRole`, and outputs `COMM:DigitalData` that play a `OutputTextTagRole` and express a `TextTagDescriptor`. The orthography tagger can be described in the same way, with the exception that it works over input tokens, and so its input consists of text that plays a `TokenRole`. The tokens, which are the candidate entity mentions, also play the `ClassifierInputDataRole` defined by the classifier. Like in the previous example, the classifier loads a previously learned entity recognition model as input, and the generated KB statements, which are of the form *instance_of(entity_mention, class)*, express a `ClassifiedObjectDescriptor`. Annotated tokens play a `COMM:AnnotatedDataRole`, the KB statement a `COMM:SemanticLabelRole`, and, finally, the metadata that allows locating the annotated tokens in the text consist of `COMM:DigitalData` that play `OutputTextMaskRoles` and express `TextLocatorDescriptors`.

The work presented in this chapter just scratches the surface on what is possible to represent about IE systems. The focus was on those semantic constructs and ontology patterns that are generic enough to constitute the core of IE. To support other concerns, the ontology would need to be extended. For example, knowledge fusion, i. e., when a KB statement is not directly warranted by the media segment but rather receives a indirect contribution from the segment towards its existence and validity (e. g., via merging of several extracted facts), is an important concern that would deserve its own pattern.

Other potential uses of the ontology

The potential uses of the proposed Ontology of Information Extraction go beyond the main motivation in this chapter, that of supporting automated discovery and composition of IE components and services. For example, the formalism enables providing detailed information about the *provenance* [134] of extracted facts, which in some systems is stored as metadata. Another possible use of the formalism would be to contribute to an enhanced *replicability* of empirical research (a concern shared recently in several areas, see [135, 146, 157, 106]), by allowing researchers to accompany published results with an unambiguous (formal) description of the IE methods and systems used in the experimental tasks that they report about.

6.4 RELATED WORK

There are several research areas that are relevant to the work presented in this chapter.

Research on the software engineering aspects of natural language systems has led to the current availability of feature-rich software frameworks to support building and maintaining them. Two frameworks representative of the *state-of-the-art* are GATE and UIMA.

General Architecture for Text Engineering

The GATE [21] is arguably the most widely used academic framework and graphical development environment for NLP tools and applications. It has been extended over the years to include support for ontologies, multimedia data, and machine learning. GATE uses native data structures (the current implementation is in Java) to represent data, and features a plugin-based architecture in which components can be described using so-called CREOLE (XML) descriptor files.

Unstructured Information Management Architecture

The Unstructured Information Management Architecture (UIMA) [57] is an open-source platform for integrating components that analyse unstructured sources such as multimedia documents. Unlike GATE, UIMA adopts a declarative data layer for interoperability. UIMA-based systems define type systems — similar to ontologies with extremely limited semantic commitments, e. g., only supporting single-inheritance type/subtype hierarchies — to specify the kinds of information that they manipulate [69].

Unfortunately, despite offering a great deal of functionality, current frameworks tend to lock developers by not offering any mechanisms to ease the integration of functionality that is required but not already provided. XML-based plugin descriptor files, document annotations stored in native data structures and limited type systems, are just a few examples of framework-specific design decisions that present the following problems:

SCOPE Design decisions tend to mimic the native programming language. For example, component descriptors often merely expose OO class fields as design time and runtime parameters in some configuration file. It is thus impossible to create descriptions that fall outside the scope of what the native programming language can represent. For instance, it would prove difficult to represent that the component input is any "multimedia document that contains exactly one image and no audio or video", since modern OO programming languages are less expressive than e. g., OWL.

Limitations of state-of-the-art IE frameworks

SEMANTICS Framework-specific types and concepts lack formal semantics. Appropriate within the context of the framework only, it is impossible to guarantee that these types can be reused in the context of any other framework, because even if names match syntactically, they are unlikely to match semantically — they mean different things in different frameworks.

WEB INTEROPERABILITY Most framework-specific component/service descriptors are not compatible with existing Web and Semantic Web standards, which limits the interoperability with publically available services on the Web.

Due to the above problems, new directions in software engineering have recently been explored to combat the increasing complexity and rapid pace of change in modern systems development. Among these new paradigms is Semantic Web Enabled Software Engineering (SWESE), which tries to apply Semantic Web technologies (such as ontologies and reasoners) in mainstream software engineering [78]. It hopes to provide stronger logical foundations and precise semantics to software models and other development artifacts. The work presented in this chapter can be regarded as a contribution to this goal, as it provides a Semantic Web-based formalism with application to the domain of IE systems engineering.

Semantic Web Enabled Software Engineering

In recent years, several researchers have reported on the use of ontologies as a means to enable interoperability between text mining, NLP or linguistic tools and resources. Declerck et al. [48] argue the need for a language infrastructure that enables sharing NLP tools and language resources, and suggest that Semantic Web technologies are the most suitable to achieve that goal.

Similarly, Hayashi et al. [81] describe the need for a *global language infrastructure*, an open and web-based software platform to which language resources can be easily plugged, and on which language services can be efficiently composed, disseminated and consumed. They argue that such infrastructure should be ontology-based, and introduce the idea of a *language service ontology* [80]. The top-level of the language service ontology contains concepts such as `LanguageProcessingResource`, `LanguageDataResource`, and `LanguageService`. Further, they propose sub-ontologies to handle linguistic annotation and lexicon modeling. The linguistic annotation sub-ontology is incorporated with the pur-

Language Service Ontology

pose of specifying the input/output data of NLP tools as well as for defining the content of corpora, and includes concepts such as LinguisticExpression, LinguisticMeaning and LinguisticAnnotation. The lexicon modeling sub-ontology specialises the top-level concept LanguageDataResource with concepts such as Corpus and Lexicon. Compared to ONIX, the language service ontology is more geared towards linguistics and language resources, while ONIX is tailored to the domain of IE. Moreover, ONIX views data as multimedia artifacts instead of as language resources, and is also agnostic about whether processing resources are language related, making it more generally applicable.

Ontology of NLP services

A framework for describing and discovering NLP processing resources is proposed by Klein and Potter [97]. The overall motivation of the authors is similar to the one presented in this chapter: interoperability and discoverability of NLP components and services, with a view to seamless composition (be it manual or automated). Their proposed ontology of NLP services is based on the OWL-S semantic service description ontology [29], which serves the same purpose as CSO in ONIX. The top-level class in their ontology is NL-Resource, representing a *natural language resource*, which subsumes two subclasses, NL-StaticResource and NL-ProcessingResource. A NL-StaticResource describes things like corpora, probability models, lexicons and grammars, while a NL-ProcessingResource describes things like tagging and parsing tools. The Document class contains a hasAnnotation property, which enables representing the several layers of annotation generated by the processing resources over the input document. The class NL-Analyzer specialises NL-ProcessingResource by restricting its hasInput and hasOutput properties to those resources that are textual in nature.

Advantages of the proposed ontology

By being grounded on DOLCE, CSO and COMM, ONIX offers several advantages over the ontology proposed by Klein and Potter. First, the DnS ontology pattern makes it possible and, in fact, almost ensures that that the design and extensions to ONIX are performed in a sound way, analogous to the use of the *dependency inversion* principle in software engineering, i. e., by adding descriptions of situations that interlink the respective "involved parties" (classes) but avoid having to modify them. For that reason, in ONIX documents do not "have" annotations, but instead annotated documents are COMM:MultimediaData that play an COMM:AnnotatedDataRole in the context of a COMM:SemanticAnnotation situation that satisfies the particular OIO:Method that generates the annotations. This is intuitively more sensible, since annotations are not intrinsic properties of documents — in fact, in any other domain asserting that the class Document features a hasAnnotation property would not make sense. Second, the use of COMM enables inheriting a number of semantic constructs for describing multimedia data, hence broadening the scope of ONIX beyond a mere formalisation of NLP components and text documents. Finally, ONIX additionally enables representing machine learning methods and algorithms based on CSO constructs, which constitute an important class of components in the domain of IE.

Related work on using semantic descriptions of NLP systems

Halpin [76] proposes the use of Semantic Web ontologies, and in particular the ontology by Klein and Potter, to describe natural language systems together with the performance and accuracy obtained by a given configuration of components. The author argues that storing metadata about how well NLP components perform in different situations can contribute to guiding the composition of components for assembling effective systems according to task and input, and lead to advances in the sharing and evaluation of NLP resources in the scientific community. This view is aligned with what was mentioned in section 6.3 concerning replicability of IE experiments.

Language Grid composite language services

In fact, in line of the above, a primitive version of Hayashi et al.'s vision is already up and running in the form of the Language Grid [88], an infrastructure for coordinating distributed language services on the Web. On the Language Grid, services that provide the functionality of a single language resource are called atomic services, while services composed of atomic services are called composite and described in WS-BPEL [25]. For example, a composite service can be composed of a machine translator, a morphological analyser and a technical term dictionary in order to translate sentences in a specialised domain. Instrumental to enabling discoverability and interoperability of services is the language service ontology of [80], upon which the interfaces of all services on the Language Grid are designed.

6.5 CONCLUSIONS

The contribution in this chapter consisted of a well-founded, modular, extensible and multimedia-aware formalism for describing Information Extraction systems, called the Ontology of Information Extraction. The ontology provides the top-level semantic constructs required to describe the backbone of any IE system: the subsystems, tasks and methods that address the basic concerns of decomposition, segment analysis, data modeling and semantic annotation (cf. section 2.7).

The advantages of using the proposed ontology include enhanced interoperability among systems through an agreed-upon means of referring to and describing IE concepts; and supporting faster and optimal engineering decisions both at design time and at runtime, through automated discovery and composition, contributing to reduced system development and maintenance costs. Additionally, the ontology can be used to record detailed information about the provenance of extracted facts, and it could contribute to improved replicability of IE experiments.

The DC and EE tasks were used as starting point for illustrating the problem addressed. The systems I implemented to test the hypotheses in chapter 4 and chapter 5 were used to identify concrete requirements for the design of the ontology (cf. section 6.1): coverage, multimedia awareness, semantic and syntactic interoperability, separation of concerns, modularity, and extensibility. An explanation of how the requirements were met was given in section 6.3, together with an example of how the proposed ontology is used to formally describe systems that perform these tasks.

In the design of the ontology, existing foundational and core Semantic Web ontologies, reviewed in section 2.7, were reused and extended by specialising concepts defined in those ontologies to the domain of Information Extraction. The Ontology of Information Extraction is, to the best of my knowledge, the first specialised ontology for semantically describing this domain.

The work presented in this chapter just scratches the surface on what is possible to represent about unstructured information mining systems. The focus was on those semantic constructs and ontology patterns that are generic enough to constitute the core of unstructured information mining. To support further concerns, other ontologies will be created that extend the proposed core ontology.

7 CONCLUSIONS

The research presented in this thesis comprised the design and evaluation of machine learning and ontology models to tackle important and current research questions concerning two related tasks in text mining: document classification and information extraction. Throughout the previous chapters a variety of approaches, methods and experimental evidence were provided, all of which contributed to getting us closer to the overall goal we set out to achieve, that of reducing the cost of creating and maintaining DC and IE systems.

The first problem tackled in this thesis was the cost of engineering features, a problem which was described detailedly in section 3.1. I started by measuring the impact (in system accuracy) of the several classes of features and model parameters typically employed in ML-based approaches to the Entity Extraction task. Initially identifying these successful features and parameters was necessary to guide the design of the several systems that I implemented to gather the empirical evidence presented in the rest of the thesis. The study was also motivated by the need to run the several systems under the same experimental conditions, since reported results in the literature are, unfortunately, not often comparable. It is also a valuable guide to other researchers in itself, contributing to a clarification of what constitute successful classes of features for EE.

The design of a first simple system according to the lessons learned via the aforementioned studied warranted some interesting results. One of the main conclusions was that, in general, the use of very simple features is surprisingly enough to account for most of the EE accuracy obtained, and that the use of rich data resources, such as gazetteers, greatly contributes to the best observed results and it is more likely to explain the differences in the results reported by several systems in the literature than the design decisions relative to the learning model. In effect, despite being significantly simpler than the systems against which it was compared, the system was shown to be very competitive. Further, we concluded that feature selection does not contribute to improving the accuracy in the EE task, in contrast with the application of feature selection to the TC task, where the use of some of the metrics is known to consistently improve accuracy. The results showed, however, that even the simple frequency metric can greatly reduce the number of features with no significant loss in terms of accuracy. Knowing that accuracy is not affected enables us to build simpler models in terms of number and classes of features.

Next, attention was turned to extending feature classes to include features that exploit the multimedia nature of modern documents, a little explored avenue in the literature. My contribution to the state-of-the-art here consisted of a novel class of successful features for the task of classifying multimedia documents. To collect the features from corpora, I proposed a novel method for detecting cross-media associations and quantifying the level of image and text correlation. I showed that, by preserving not just text and images but also the cross-media correlations between text elements and the images in a multimedia document, it is possible to improve system accuracy. Plus, the proposed approach has a significant characteristic: it makes almost no assumptions about the way multimedia content is modelled, making it widely applicable to virtually any multimedia document.

Seeking novel approaches to overcoming the scarceness of labelled data (the second problem addressed in this thesis, introduced in section 3.2), I began by presenting a successful application of IE technology to a real-world problem in the aerospace engineering domain characterized by very costly labelled data (annotations of jet engine reports) on the one hand, and the existence of complex patterns, which would be also costly to enunciate and maintain without the aid of some automated approach. In order to alleviate the need for labelled data and at the same time take advantage of the high volume of unlabelled data available, semi-supervised ML methods were employed, yielding results comparable to the state-of-the-art in other text mining tasks. This study prepared the ground for the main contribution in this thesis, and filled a gap in the literature, since little has been reported on the application of semi-supervised methods to real-world EE problems.

Motivated by problems characterized by limited availability of labelled data, such as the one above, I then devised a novel approach to domain adaptation for text categorization that aims at reducing the effort in porting existing statistical models induced from corpora in one domain to other related domains, thus greatly alleviating supervision requirements for tasks of this kind. Key to the approach is a new paradigm of labelling words (as opposed to labelling whole documents), was shown in the literature to be less time consuming and more natural to domain experts. I proposed a method to minimize the divergence between the distributions of the labelled source domain and the unlabelled target domain. Further, I extended the method to make use of the labelled features to guide the discovery of a latent concept space, which is then used to augment the original labelled features set. The results showed that, despite the use of a weak form of supervision, we are able to outperform standard supervised and semi-supervised methods and match the results obtained by the more expensive forms of supervising. Lowering the supervision requirements in this manner constitutes, in my opinion, a significant step towards a more general applicability of ML-based text mining approaches.

Finally, I focused on the issue of the interoperability of systems and comparability of results, still a major contributor to the high cost of creating and maintaining IE technology, as explained in section 3.3. My contribution consisted of a formalism for declaratively specifying unstructured multimedia information mining systems, their subsystems and components, in the form of ontology. The ontology can be used to create unambiguous descriptions of systems or to record detailed information about the provenance of extracted facts, thus leading to improved interoperability and replicability of experiments.

Although document classification and information extraction are no longer young research areas, there are still many open questions and research opportunities. These include:

NOVEL CLASSES OF FEATURES In chapter 4, I augmented the classes of features typically used for DC to include features that capture correlations between multimedia elements in a multimedia document, with minimal assumptions made about the content. The results were encouraging and hint at the new opportunities to, in an analogous manner, identify classes of features that take into account generic properties of classes of documents. Since identifying successful features for a given text mining problem is a very time consuming task, a desirable future scenario for anyone involved in creating and maintaining this kind of systems would be to have at their disposal catalogued knowledge that maps document characteristics to successful features classes. Such knowledge would ideally be readily available to plug-in routinely into new or existing solutions.

NOVEL MACHINE LEARNING METHODS This is the focus of most of the work reported in the literature, at the time of writing: to explore the application of new algorithms arising from advances in ML theory (although there have been also cases in which IE practice precedes more solid theoretical analysis). I expect, however that we are reaching a plateau in what respects the value of the contributions if pursuing this direction, and that any significant contribution yet to come in the short to medium term is going to be less revolutionary than the other topics in this list.

NOVEL PARADIGMS FOR ACQUIRING LABELLED DATA In chapter 5, a new paradigm was employed that constitutes a significant departure from the well-established instance labelling paradigm, with very promising results. It is my expectation that new paradigms for annotating data will revolutionise the way we approach IE and yield significant gains both in terms of accuracy and general portability of IE to a variety of domains. I would like to briefly mention tagging, as performed by web users, as an example of one such paradigm. It can be regarded as a weak form of annotation, and could have a defining role in shaping a bootstrapping process whereby existing tags on the Web enable tagging other Web content automatically, and this new tagged content can in turn be used to tag more content, in a self-sustaining cycle. This is one of my favourite possible avenues for future work.

NOVEL FORMALISMS FOR REPRESENTING IE-RELATED ARTEFACTS The work presented in chapter 6 provides a minimal base upon which to build more extensive and specific representation formalisms able to represent unstructured information mining systems. There is ample room to do so, given the breadth of IE domains and solutions out there, and for that reason I expect the proposed ontology (or an analogous one) to assume a central role in an explosion of metadata describing IE, to happen in the short to medium-term.

I expect advances in research will address these questions. I hope that machine learning-based Information Extraction becomes, in the long-term, a cost-effective class of solutions for practical applications in any domain.

BLANK PAGE
IN
ORIGINAL

Part III

APPENDICES

A OPEN-SOURCE SOFTWARE AUTHORED

A.1 THE RUNES FRAMEWORK

Runes is a framework that handles, on behalf of the designer/implementer, several important aspects related to the representation of multimedia resources for unstructured information mining purposes:

- it adresses scalability by automating the selection of an optimal underlying data structure for holding the data at any time during processing;
- it provides support for expressive data models up to the level of hypergraphs;
- it enhances portability by featuring a plugin framework and encouraging developers to think in terms of small modular processing units;
- it integrates data by providing unique identifiers to stored data and merging those that are identical;
- it orchestrates execution of external tools by running a dependencies resolution algorithm that determines which should run and in which order; and
- it supports processing several data formats and media thanks to the provided plugins that accompany the framework

Runes can be downloaded from the Sourceforge project page:
`http://runes.sourceforge.net/`

A.2 THE ALEPH LIBRARY

Aleph is both a multi-platform machine learning framework aimed at simplicity and performance, and a library of selected state-of-the-art algorithms.

Aleph features:

- semi-supervised algorithms: graph label propagation, discrete regularization, etc.
- large-scale linear algorithms: logistic linear regression, stochastic gradient descent linear SVM, etc.
- wrappers to well-known tools: libsvm, SVMlight, etc.
- graph-based algorithms: random walks, absorbing random walks, etc.
- feature selection statistics: infogain, cross entropy, chi-squared, etc.
- convenience validation utilities: several splitting methods, several scoring functions
- fast vector and matrix implementations: based on matrix toolkits for java, but with a few optimizations on top of it
- fast on-the-fly operations over datasets, instances and features: based on the concept of views over those first-class objects in the framework

Aleph can be downloaded from the Sourceforge project page: `http://aleph-ml.sourceforge.net/`

A.3 THE T-REX SYSTEMS

Trainable Relation Extraction (T-Rex) is a highly configurable machine learning-based Information Extraction from Text framework, which includes tools for document classification, entity extraction and relation extraction.

T-Rex can be downloaded from the Sourceforge project page: `http://t-rex.sourceforge.net/`

B

PAIRING START AND END OF BOUNDARY PREDICTIONS

The boundary classification model for EE requires start and end boundary predictions to be re-conciliated as a post-processing stage. Here I describe, in the form of pseudo-code, the algorithm to pair start and end boundary predictions.

Algorithm 3 Pseudo-code for the algorithm to pair start and end boundary predictions

```
P ← set of predictions from model
A ← new entity annotation set
C ← pruneCandidates(identifyCandidates(P))
While C is not empty, do
    1. c ← next candidate from C
    2. R ← determineConflictRegion(c, C)
    3. B ← determineBestConfiguration(R)
    4. A ← $A \cup B$
    5. C ← $C \setminus R$
Annotate text using A
```

Algorithm 4 Pseudo-code for the function identifyCandidates

```
C ← new entity annotation set
for each entity type t, do
    1. n ← next start boundary prediction
    2. if n exists, then
        2.1. e ← next end boundary prediction before n
        2.2. while e exists, do
            2.2.1. s ← n
            2.2.2. n ← next start boundary prediction
            2.2.3. do
                2.2.3.1. c ← new candidate of type t spanning text (s, e)
                2.2.3.2. C ← $C \cup \{c\}$
                2.2.3.3. e ← next end boundary prediction before n
            2.2.3. while e exists
return C
```

Algorithm 5 Pseudo-code for the function pruneCandidates

```
$C \leftarrow$ candidate entity annotation set (passed as argument)
$P \leftarrow$ new entity annotation set
$A \leftarrow$ candidates sorted by start boundary position in document
for each candidate $c \in C$, do
  1. accept $\leftarrow$ TRUE
  2. $R \leftarrow$ candidates inside text region spanned by c
  3. for each candidate $r \in R$, do
    3.1. if $\text{score}(c) < \text{score}(r)$ then accept $\leftarrow$ FALSE
  4. if accept = TRUE then $P \leftarrow P \cup \{c\}$
return P
```

Algorithm 6 Pseudo-code for the function determineConflictRegion

```
$s \leftarrow$ annotation (passed as argument) $R \leftarrow$ new set of entity annotations
$L \leftarrow$ new stack of entity annotations
$R \leftarrow R \cup \{s\}$
$L \leftarrow \text{push}(L, s)$
while L is not empty
  1. $a \leftarrow \text{pop}(L)$ 2. $O \leftarrow$ candidates overlapping text region spanned by a
  3. for each candidate $o \in O$, do
    3.1. if $o \notin R$ then
      3.1.1. $R \leftarrow R \cup \{o\}$
      3.1.2. $L \leftarrow \text{push}(L, o)$
return R
```

Algorithm 7 Pseudo-code for the function determineBestConfiguration

```
$R \leftarrow$ set of entity annotations (passed as argument)
$m \leftarrow$ new entity annotation configuration (max)
$L \leftarrow$ new stack of entity annotations configurations
$L \leftarrow \text{push}(L, m)$
while L is not empty
  1. $c \leftarrow \text{pop}(L)$
  2. for each $r \in R$
    2.1. if $\text{overlaps}(r, c)$ then
      2.1.1. $a \leftarrow$ non-overlapping configuration from $(c, r)$
      2.1.2. if $a \notin L$ then $\text{push}(L, a)$
    2.2. else $m \leftarrow m \cup \{r\}$
  3. $L \leftarrow L \setminus \{c\}$
  4. if $\text{confidenceScore}(m) < \text{confidenceScore}(c)$ then $m \leftarrow c$
return m
```

BIBLIOGRAPHY

[1] Steven Abney. Parsing by chunks. In *Principle-Based Parsing*, pages 257–278. Kluwer Academic Publishers, 1991. (Cited on page 18.)

[2] ACE. Nist. automatic content extraction (ace) program. 1998-present., 2008. URL `http://www.itl.nist.gov/iad/mig/tests/ace/2008/`. (Cited on pages 22 and 102.)

[3] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499. Morgan Kaufmann Publishers Inc., 1994. ISBN 1-55860-153-8. (Cited on page 39.)

[4] A. Agresti. Building and applying logistic regression models. In *An Introduction to Categorical Data Analysis*, page 138. Wiley, 2007. ISBN 978-0-471-22618-5. (Cited on page 30.)

[5] D. Andrzejewski and X. Zhu. Latent dirichlet allocation with topic-in-set knowledge. In *Proceedings of the Workshop on Semi-supervised Learning for Natural Language Processing at NAACL-HLT'09*. ACL, 2009. (Cited on pages 123 and 126.)

[6] Douglas E. Appelt. Introduction to information extraction. *AI Communications*, 12(3):161–172, 1999. (Cited on page 6.)

[7] Arvind Arasu and Hector Garcia-Molina. Extracting structured data from web pages. In Alon Y. Halevy, Zachary G. Ives, and AnHai Doan, editors, *ACM SIGMOD Conference*, pages 337–348. ACM, 2003. (Cited on page 102.)

[8] Richard Arndt, Raphael Troncy, Steffen Staab, Lynda Hardman, and Miroslav Vacura. COMM: designing a well-founded multimedia ontology for the web. In *ISWC 2007 + ASWC 2007*, pages 30–43, 2007. (Cited on pages 54, 56, and 57.)

[9] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open information extraction from

the web. In *In the Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2670–2676, January 2007.

[10] K. Barnard and D. Forsyth. Learning the semantics of words and pictures. In *International Conference on Computer Vision*, pages II:408–415, 2001. (Cited on page 103.)

[11] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. OWL Web Ontology Language Reference. Website, February 2004. URL `http://www.w3.org/TR/owl-ref`. (Cited on page 49.)

[12] M. Belkin and P. Niyogi. Using manifold structure for partially labeled classification. In *Proceedings of the 16th Annual International Conference on Advances in Neural Information Processing Systems (NIPS'02)*, 2002. (Cited on page 45.)

[13] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *Proceedings of 20th Annual Conference on Neural Information Processing Systems (NIPS-06)*, 2006. (Cited on pages 64 and 131.)

[14] K. P. Bennett and A. Demiriz. Semi-supervised support vector machines. In *Proceedings of the 1998 Conference on Advances in Neural information Processing Systems*, pages 368–374. MIT Press, 1999. (Cited on page 44.)

[15] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifiers (URI): Generic syntax. Website, August 1998. URL `http://www.ietf.org/rfc/rfc2396.txt`. (Cited on page 48.)

[16] Ravish Bhagdev, Sam Chapman, Fabio Ciravegna, Vitaveska Lanfranchi, and Daniela Petrelli. Hybrid search: Effectively combining keywords and ontology-based searches. In Manfred Hauswirth, Manolis Koubarakis, and Sean Bechhofer, editors, *Proceedings of the 5th European Semantic Web Conference*, LNCS, Berlin, Heidelberg, June 2008. Springer Verlag.

[17] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. (Cited on page 122.)

[18] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006*

Conference on Empirical Methods in Natural Language Processing (EMNLP'06). ACL, 2006. (Cited on page 131.)

[19] A. Blum, J. Lafferty, R. Rwebangira, and R. Reddy. Semi-supervised learning using randomized min-cuts. In *Proceedings of the 21st International Conference on Machine Learning (ICML'04)*. Morgan Kaufmann, 2004. (Cited on page 45.)

[20] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *COLT' 98: Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998. (Cited on page 102.)

[21] Kalina Bontcheva, Valentin Tablan, Diana Maynard, and Hamish Cunningham. Evolving gate to meet new challenges in language engineering. *Natural Language Engineering*, 10(3-4):349–373, 2004. (Cited on page 148.)

[22] A. Bordes, L. Bottou, P. Gallinari, and J. Weston. Solving multi-class support vector machines with larank. In *Proceedings of the 24th international conference on Machine learning (ICML'07)*, pages 89–96. ACM, 2007. (Cited on page 130.)

[23] A. Borthwick. A maximum entropy approach to named entity recognition, 1999. (Cited on page 8.)

[24] Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In *IN PROCEEDINGS OF THE SIXTH WORKSHOP ON VERY LARGE CORPORA*, pages 152–160. Association for Computational Linguistics, 1998. (Cited on page 99.)

[25] BPEL. Business process execution language for web services (bpel), version 1.1., 2002. URL `http://www.ibm.com/developerworks/library/ws-bpel/`. (Cited on page 151.)

[26] T. M. Breuel. Information extraction from html document by structural matching. In *In Int'l Workshop on Web Document Analysis*, pages 11–14. ACM, 2003. ISBN 1-58113-057-0. doi: http://doi.acm.org/10.1145/279943.279962. (Cited on page 61.)

[27] Dan Brickley and RV. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. Website, February 2004. URL `http:`

//www.w3.org/TR/2004/REC-rdf-schema-20040210/. (Cited on page 49.)

[28] Bruce G. Buchanan and David C. Wilkins, editors. *Readings in knowledge acquisition and learning: automating the construction and improvement of expert systems*. Morgan Kaufmann Publishers Inc., 1993.

[29] Mark Burstein, Jerry Hobbs, Ora Lassila, Drew Mcdermott, Sheila Mcilraith, Srini Narayanan, Massimo Paolucci, Bijan Parsia, Terry Payne, Evren Sirin, Naveen Srinivasan, and Katia Sycara. Owl-s: Semantic markup for web services. Website, November 2004. URL http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/. (Cited on page 150.)

[30] Ekaterina Buyko, Elena Beisswanger, and Udo Hahn. Testing different ace-style feature sets for the extraction of gene regulation relations from medline abstracts. In *Proceedings of the Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008)*, pages 21–28, June 2008. (Cited on page 102.)

[31] Michael J. Cafarella, Doug Downey, Stephen Soderland, and Oren Etzioni. Knowitnow: fast, scalable information extraction from the web. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 563–570. Association for Computational Linguistics, 2005. (Cited on page 8.)

[32] Mary Elaine Califf and Raymond J. Mooney. Relational learning of pattern-match rules for information extraction. In *AAAI '99/IAAI '99: Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, pages 328–334. American Association for Artificial Intelligence, 1999. ISBN 0-262-51106-1. (Cited on pages 23 and 35.)

[33] M. La Cascia, S. Sethi, and S. Sclaroff. Combining textual and visual cues for content-based image retrieval on the world wide web. In *CBAIVL '98: Proceedings of the IEEE Workshop on Content - Based Access of Image and Video Libraries*. IEEE Computer Society, 1998. (Cited on page 103.)

[34] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning (Adaptive Computation and Machine Learning).* The MIT Press, September 2006. (Cited on pages 9 and 111.)

[35] B. Chen, W. Lam, I. Tsang, and T.L. Wong. Extracting discriminative concepts for domain adaptation in text mining. In *Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2009. (Cited on pages xv, 127, 128, and 131.)

[36] Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. Relation extraction using label propagation based semi-supervised learning. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 129–136. Association for Computational Linguistics, 2006. (Cited on pages 8, 118, and 129.)

[37] Hai Leong Chieu and Hwee Tou Ng. A maximum entropy approach to information extraction from semi-structured and free text. In *In Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pages 786–791, 2002. (Cited on page 24.)

[38] Fabio Ciravegna. Adaptive information extraction from text by rule induction and generalisation. In *Proceedings of 17th International Joint Conference on Artificial Intelligence (IJCAI)*, 2001. (Cited on pages 24 and 35.)

[39] Fabio Ciravegna, Ajay Chakravarthy, and Vitaveska Lanfranchi. Cross-media document annotation and enrichment. In *Proceedings of the 1st Semantic Authoring and Annotation Workshop (SAAW2006)*, 2006. (Cited on page 44.)

[40] William W. Cohen and Yoram Singer. A simple, fast, and effective rule learner. In *In Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 335–342. AAAI Press, 1999. (Cited on page 35.)

[41] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of the Joint SIG-DAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999. (Cited on page 8.)

[42] C. Cortes and V. Vapnik. Support-vector network. *Journal of Machine Learning Research*, 20:273–297, 1995.

[43] Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo. Roadrunner: Towards automatic data extraction from large web sites. In *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*, pages 109–118. Morgan Kaufmann Publishers Inc., 2001. (Cited on page 102.)

[44] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 1 edition, 2000. ISBN 0521780195. (Cited on pages 8, 32, 33, and 112.)

[45] Aron Culotta and Jeffery Sorensen. Dependency tree kernels for relation extraction. In *42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2004. URL `http://www.cs.umass.edu/~culotta/pubs/culotta04dependency.pdf`. (25% accepted; 134 citations in Google Scholar). (Cited on page 8.)

[46] Aba-Sah Dadzie, R. Bhagdev, A. Chakravarthy, S. Chapman, J. Iria, V. Lanfranchi, J. Magalhaes, D. Petrelli, and F. Ciravegna. Applying semantic web technologies to knowledge sharing in aerospace engineering. *Journal of Intelligent Manufacturing, Special issue on Knowledge Discovery and Management in Engineering Design and Manufacturing*, 2008. (Cited on page 61.)

[47] J.G. Daugman. Complete discrete 2-d gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(7):169–1179, 1988. ISSN 0096-3518. (Cited on page 89.)

[48] Thierry Declerck, Paul Buitelaar, N. Calzolari, and A. Lenci. Towards a language infrastructure for the semantic web. In *International Conference on Language Resources and Evaluation (LREC-04), Lisbon, Portugal*. ELRA/ELDA, 2004. (Cited on page 149.)

[49] Thomas G. Dietterich. Ensemble methods in machine learning. In *MCS '00: Proceedings of the First International Workshop on Multiple Classifier Systems*, pages 1–15. Springer-Verlag, 2000. ISBN 3-540-67704-6. (Cited on page 60.)

[50] G. Druck, G. Mann, and A. McCallum. Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st Annual International SIGIR Conference*. ACM, 2008. (Cited on pages 46 and 124.)

[51] S. Ebadollahi, Lexing Xie, Shih-Fu Chang, and J.R. Smith. Visual event detection using multi-dimensional concept dynamics. In *2006 IEEE International Conference on Multimedia and Expo*, pages 881–884, 2006. (Cited on page 61.)

[52] Erik and Fien De Meulder. Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 142–147. Association for Computational Linguistics, 2003. (Cited on pages 22 and 99.)

[53] Thomas Erl. *SOA Design Patterns*. Prentice Hall PTR, 2009. ISBN 0136135161, 9780136135166. (Cited on page 66.)

[54] Timur Fayruzov, Martine De Cock, Chris Cornelis, and Veronique Hoste. The role of syntactic features in protein interaction extraction. In *DTMBIO '08: Proceeding of the 2nd international workshop on Data and text mining in bioinformatics*, pages 61–68, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-251-1. doi: http://doi.acm.org/10.1145/1458449.1458463. (Cited on pages 22 and 101.)

[55] Edward A. Feigenbaum and Pamela McCorduck. *The fifth generation*. Addison-Wesley, 1983. (Cited on page 9.)

[56] Yansong Feng and Mirella Lapata. Automatic image annotation using auxiliary text information. In *Proceedings of ACL-08: HLT*, pages 272–280. Association for Computational Linguistics, June 2008. (Cited on page 103.)

[57] David Ferrucci and Adam Lally. Uima: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348, 2004. (Cited on page 148.)

[58] Jenny Rose Finkel and Christopher D. Manning. Hierarchical bayesian domain adaptation. In *Proceedings of the 2009 NAACL-HLT Conference*. ACL, 2009. (Cited on page 131.)

[59] Aidan Finn. A multi-level boundary classification approach to information extraction, 2006. (Cited on pages 24, 40, 76, and 86.)

[60] George Forman. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3:1289–1305, 2003. ISSN 1533-7928. (Cited on page 33.)

[61] D. Freitag and N. Kushmerick. Boosted Wrapper Induction. In *Proceedings of AAAI 2000*, 2000. (Cited on pages 24 and 35.)

[62] Dayne Freitag. Information extraction from html: Application of a general machine learning approach. In *In Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 517–523, 1998. (Cited on pages 23, 35, and 72.)

[63] Dayne Freitag and Andrew Kachites Mccallum. Information extraction with hmms and shrinkage. In *In Proceedings of the AAAI-99 Workshop on Machine Learning for Information Extraction*, pages 31–36, 1999. (Cited on page 41.)

[64] A. Gangemi, C. Catenacci, and M. Battaglia. Inflammation ontology design pattern: an exercise in building a core biomedical ontology with descriptions and situations. In *Ontologies in Medicine*, pages 64–80. IOS Press, 2004. (Cited on page 50.)

[65] Aldo Gangemi and Peter Mika. Understanding the semantic web through descriptions and situations. In *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, pages 689–706. Springer-Verlag, 2003. (Cited on page 51.)

[66] Aldo Gangemi, Stefano Borgo, Carola Catenacci, and Jos Lehmann. Task taxonomies for knowledge content d07. Technical report, Metokis Project, 2004. (Cited on pages 50, 52, and 53.)

[67] Alfio Massimiliano Gliozzo, Claudio Giuliano, and Raffaella Rinaldi. Instance filtering for entity recognition. *SIGKDD Explor. Newsl.*, 7(1):11–18, 2005. ISSN 1931-0145. doi: http://doi.acm.org/10.1145/1089815.1089818. (Cited on page 76.)

[68] Asuncion Gomez-Perez, Oscar Corcho, and Mariano Fernandez-Lopez. *Ontological Engineering : with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. First Edition (Advanced Information and Knowledge Processing)*. Springer, 2004. ISBN 1852335513. (Cited on page 137.)

[69] Thilo Gotz and Oliver Suhre. Design and implementation of the uima common analysis system. *IBM Systems Journal*, 43(3): 476–489, 2004. (Cited on page 148.)

[70] Ralph Grishman and Beth Sundheim. Message understanding conference 6: a brief history. In *Proceedings of the 16th conference on Computational linguistics*, pages 466–471. Association for Computational Linguistics, 1996. (Cited on page 22.)

[71] MPEG Working Group. Mpeg-7: The generic multimedia content description standard, part 1. *IEEE MultiMedia*, 9(2):78–87, 2002. ISSN 1070-986X. (Cited on pages 54, 88, and 89.)

[72] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993. (Cited on pages 47 and 135.)

[73] Li H. Gu Xu G., Yang S.-H. Named entity mining from click-through data using weakly supervised latent dirichlet allocation. In *Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2009. (Cited on page 123.)

[74] H. Guo, H. Zhu, Z. Guo, X. Zhang, X. Wu, and Zr. Su. Domain adaptation with latent semantic association for named entity recognition. In *Proceedings of the 2009 NAACL-HLT Conference*. ACL, 2009. (Cited on page 131.)

[75] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 427–434. Association for Computational Linguistics, 2005. (Cited on page 102.)

[76] Harry Halpin. Automatic evaluation and composition of nlp pipelines with web services. In *Proceedings of Language Resources and Evaluation of Corpora Conference*, 2006. (Cited on page 151.)

[77] Siegfried Handschuh, Steffen Staab, and Fabio Ciravegna. S-cream - semi-automatic creation of metadata. In *EKAW '02: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, pages 358–372. Springer-Verlag, 2002. ISBN 3-540-44268-5. (Cited on page 19.)

[78] Hans-Jorg Happel and Stefan Seedorf. Applications of ontologies in software engineering. In *Second International Workshop on Semantic Web Enabled Software Engineering (SWESE 2006)*, 2006. (Cited on page 149.)

[79] T. Hasegawa, S. Sekine, and R. Grishman. Discovering Relations among Named Entities from Large Corpora. In *Proc. of ACL-2004*, pages 415–422, 2004. (Cited on page 8.)

[80] Yoshihiko Hayashi. A linguistic service ontology for language infrastructures. In *ACL '07: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 145–148. Association for Computational Linguistics, 2007. (Cited on pages 149 and 151.)

[81] Yoshihiko Hayashi, Thierry Declerck, Paul Buitelaar, and Monica Monachini. Ontologies for a global language infrastructure. In Jonathan Webster, Nancy Ide, and Alex Chengyu Fang, editors, *Proceedings of the 1st International Conference on Global Interoperability for Language Resources (ICGL-2008), January 9-11, Hong Kong, China*, pages 105–112, 1 2008. (Cited on pages 149 and 151.)

[82] J. Hendler. Agents and the semantic web. *Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications]*, 16: 30–37, 2001.

[83] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval (SIGIR-99)*. ACM, 1999. (Cited on page 131.)

[84] Jing Huang, S. Ravi Kumar, Mandar Mitra, Wei-Jing Zhu, and Ramin Zabih. Image indexing using color correlograms. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 762, Washington, DC, USA, 1997. IEEE Computer Society. ISBN 0-8186-7822-4. (Cited on page 89.)

[85] IBM Research. IBM Research Website, 2009. URL `http://research.ibm.com`. (Cited on page 3.)

[86] Hal Daume III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*. ACL, 2007. (Cited on page 131.)

[87] Neil Ireson, Fabio Ciravegna, Mary Elaine Califf, Dayne Freitag, Nicholas Kushmerick, and Alberto Lavelli. Evaluating machine learning for information extraction. In Luc De Raedt and Stefan Wrobel, editors, *ICML*, volume 119 of *ACM International Conference Proceeding Series*, pages 345–352. ACM, 2005. (Cited on pages 72 and 86.)

[88] Toru Ishida. Language grid: An infrastructure for intercultural collaboration. In *SAINT '06: Proceedings of the International Symposium on Applications on Internet*, pages 96–100. IEEE Computer Society, 2006. ISBN 0-7695-2508-3. doi: http://dx.doi.org/10.1109/SAINT.2006.40. (Cited on page 151.)

[89] Ivar Jacobson, Grady Booch, and James Rumbaugh. *The unified software development process*. Addison-Wesley Longman Publishing Co., Inc., 1999. ISBN 0-201-57169-2. (Cited on page 47.)

[90] Jing Jiang and ChengXiang Zhai. A systematic exploration of the feature space for relation extraction. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 113–120. Association for Computational Linguistics, April 2007. (Cited on pages 101 and 131.)

[91] T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning*. Morgan Kaufmann, 1999. (Cited on page 127.)

[92] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Machine Learning: ECML-98*, pages 137–142. Morgan Kaufmann, 1998. (Cited on page 77.)

[93] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, (SIGIR'05)*, 2005. (Cited on page 99.)

[94] Dhiraj Joshi, Milind Naphade, and Apostol Natsev. Semantics reinforcement and fusion learning for multimedia streams. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 309–316. ACM, 2007. (Cited on page 102.)

[95] Nanda Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 22. Association for Computational Linguistics, 2004. (Cited on page 102.)

[96] Dan Klein, Joseph Smarr, Huy Nguyen, and Christopher D. Manning. Named entity recognition with character-level models. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 180–183, Morristown, NJ, USA, 2003. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1119176.1119204. (Cited on page 24.)

[97] Ewan Klein and Stephen Potter. An ontology for nlp services. In *International Conference on Language Resources and Evaluation (LREC-04), Lisbon, Portugal*. ELRA/ELDA, 2004. (Cited on pages 150 and 151.)

[98] Jon M. K. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[99] Graham Klyne and Jeremy Carroll. Resource Description Framework (RDF): Concepts and abstract syntax. Website, 2004. URL `http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/`. (Cited on page 48.)

[100] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951. (Cited on pages 121 and 124.)

[101] Alberto H. F. Laender, Berthier A. Ribeiro-neto, Altigran S. da Silva, and Juliana S. Teixeira. A brief survey of web data extraction tools. *SIGMOD Record*, 31:84–93, 2002. (Cited on page 102.)

[102] John Lafferty, Andrew Mccallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001. (Cited on pages 8, 24, and 43.)

[103] Steffen Lamparter, Stefan Luckner, and Sibylle Mutschler. Semi-automated management of web service contracts. *International Journal of Service Sciences (IJSSci)*, 1(3/4), 2008. (Cited on page 54.)

[104] K. Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339. Morgan Kaufmann, 1995. (Cited on page 123.)

[105] Tessa Lau and Eric Horvitz. Patterns of search: analyzing and modeling web query refinement. In *UM '99: Proceedings of the*

seventh international conference on User modeling, pages 119–128, Secaucus, NJ, USA, 1999. Springer-Verlag New York, Inc.

[106] A. Lavelli, M. Califf, F. Ciravegna, D. Freitag, C. Giuliano, L. Romano, and N. Ireson. Evaluation of machine learning-based information extraction algorithms: criticisms and recommendations. *Language Resources and Evaluation*, 42(4):361–393, 2009. (Cited on pages 60, 68, 86, and 147.)

[107] Douglas B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN 0201517523. (Cited on page 50.)

[108] D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004. (Cited on page 96.)

[109] Jing Li and Christie I. Ezeife. Cleaning web pages for effective web content mining. In *Proceedings of the 17th International Conference on Database and Expert Systems Applications, (DEXA 2006)*, pages 560–571, 2006. (Cited on page 91.)

[110] Yaoyong Li, Kalina Bontcheva, and Hamish Cunningham. Using uneven margins SVM and perceptron for information extraction. In *Proceedings of Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, 2005. (Cited on pages 8, 40, and 86.)

[111] W. Y. Ma and H. Zhang. Benchmarking of image features for content-based retrieval. In *Proc. IEEE 32nd Asilomar Conf. Signals, Systems, Computers*, volume 1, pages 253–257, 1998. URL `http://turing.csie.ntu.edu.tw/~bi/docs/IEEE_papers/benchmarking_of_image_features_for_content_based_retrieval.pdf`. (Cited on page 90.)

[112] Gerd Maderlechner and Peter Suda. Information extraction from document images using white space and graphics analysis. In *SSPR '98/SPR '98: Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*, pages 468–474. Springer-Verlag, 1998. (Cited on page 61.)

[113] Joao Magalhaes and Stefan Ruger. Information-theoretic semantic multimedia indexing. In *CIVR '07: Proceedings of the 6th ACM*

international conference on Image and video retrieval, pages 619–626. ACM, 2007. (Cited on page 103.)

[114] G. S. Mann and A. McCallum. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of Machine Learning Research*, 11:955–984, 2010. (Cited on pages 46 and 121.)

[115] Andrew McCallum and Kamal Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, pages 41–48. AAAI Press, 1998. (Cited on pages 8 and 42.)

[116] Andrew McCallum, Dayne Freitag, and Fernando C. N. Pereira. Maximum entropy markov models for information extraction and segmentation. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 591–598. Morgan Kaufmann Publishers Inc., 2000. (Cited on page 42.)

[117] P. Mccullagh and J. Nelder. *Generalized Linear Models, Second Edition*. Chapman & Hall/CRC, August 1989. ISBN 0412317605. (Cited on page 29.)

[118] Ryan McDonald and Fernando Pereira. Identifying gene and protein mentions in text using conditional random fields. *BMC Bioinformatics*, 6(Suppl 1), 2005. (Cited on page 43.)

[119] Prem Melville, Maytal Saar-Tsechansky, Foster Provost, and Raymond Mooney. Active feature-value acquisition for classifier induction. In *Proceedings of the Fourth IEEE International Conference on Data Mining*, pages 483–486. IEEE Computer Society, 2004. (Cited on page 45.)

[120] Yang Mingqiang, Kpalma Kidiyo, and Ronsin Joseph. A survey of shape feature extraction techniques. In Peng-Yeng Yin, editor, *Pattern Recognition Techniques, Technology and Applications*, chapter 3, pages 626–674. I-Tech, Vienna, Austria, 2008. (Cited on page 90.)

[121] T. Mitchell. *Machine Learning*. McGraw-Hill Education (ISE Editions), October 1997. (Cited on pages 7, 19, and 35.)

[122] Marie-Francine Moens. *Information Extraction: Algorithms and Prospects in a Retrieval Context (The Information Retrieval Series)*. Springer, October 2006. (Cited on pages 4 and 35.)

[123] Saif Mohammad and Ted Pedersen. Combining lexical and syntactic features for supervised word sense disambiguation. In *Proceedings of CoNLL-2004*, pages 25–32. Boston, MA, USA, 2004. (Cited on page 102.)

[124] Ugo Montanari. On the optimal detection of curves in noisy pictures. *Commun. ACM*, 14(5):335–345, 1971. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/362588.362594. (Cited on page 18.)

[125] David Nadeau. Semi-supervised named entity recognition: Learning to recognize 100 entity types with little supervision, 2007. (Cited on page 8.)

[126] X. Ni, G.-R. Xue, X. Ling, Y. Yu, and Q. Yang. Exploring in the weblog space by detecting informative and affective articles. In *Proceedings of the 16th International World Wide Web Conference*. ACM, 2007. (Cited on page 64.)

[127] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using em. *Mach. Learn.*, 39(2-3):103–134, 2000. ISSN 0885-6125. doi: http://dx.doi.org/10.1023/A:1007692713085. (Cited on page 8.)

[128] Ian Niles and Adam Pease. Towards a standard upper ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, pages 2–9. ACM Press, 2001. (Cited on page 50.)

[129] Daniel Oberle, Steffen Lamparter, Stephan Grimm, Denny Vrandecic, Steffen Staab, and Aldo Gangemi. Towards ontologies for formalizing modularization and communication in large software systems. *Applied Ontology*, 1(2):163–202, 2006. (Cited on pages 51, 52, 53, 54, and 55.)

[130] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998. URL `http://citeseer.ist.psu.edu/page98pagerank.html`.

[131] S. J. Pan, J. T. Kwok, and Q. Yang. Transfer learning via dimensionality reduction. In *Proceedings of AAAI'08*. AAAI Press, 2008. (Cited on page 131.)

[132] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09)*, 2009. (Cited on page 131.)

[133] Greg Pass, Ramin Zabih, and Justin Miller. Comparing images using color coherence vectors. In *MULTIMEDIA '96: Proceedings of the fourth ACM international conference on Multimedia*, pages 65–73, New York, NY, USA, 1996. ACM. ISBN 0-89791-871-1. doi: http://doi.acm.org/10.1145/244130.244148. (Cited on page 89.)

[134] Rob McCool Paulo Pinheiro da Silva, Deborah L. McGuinness. Knowledge provenance infrastructure. *IEEE Data Engineering Bulletin*, 26(4):26–32, 2003. (Cited on pages 68 and 147.)

[135] Ted Pedersen. Empiricism is not a matter of faith. *Comput. Linguist.*, 34(3):465–470, 2008. ISSN 0891-2017. doi: http://dx.doi.org/10.1162/coli.2008.34.3.465. (Cited on pages 68 and 147.)

[136] Fuchun Peng and Andrew McCallum. Information extraction from research papers using conditional random fields. *Inf. Process. Manage.*, 42(4):963–979, 2006. (Cited on page 100.)

[137] M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3): 130–137, 1980. (Cited on page 92.)

[138] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Readings in speech recognition*, chapter 8, pages 267–296. Morgan Kaufmann Publishers Inc., 1990. ISBN 1-55860-124-4. (Cited on pages 41 and 42.)

[139] Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142, 1996. (Cited on page 18.)

[140] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, 1979. ISBN 0408709294. (Cited on page 7.)

[141] De Valois RL., Albrecht DG., and Thorell LG. Spatial frequency selectivity of cells in macaque visual cortex. *Vision Res.*, 5(22): 545–559, 1982. (Cited on page 89.)

[142] Katrin Tomanek Roman Klinger. Classical probabilistic models and conditional random fields. Technical report, Technical University of Dortmund, 2007. (Cited on pages 43 and 44.)

[143] Benjamin Rosenfeld, Moshe Fresko, and Ronen Feldman. A systematic comparison of feature-rich probabilistic classifiers for ner tasks. *Knowledge Discovery in Databases: PKDD 2005*, pages 217–227, 2005. (Cited on pages 100 and 101.)

[144] Binyamin Rosenfeld, Ronen Feldman, and Yonatan Aumann. Structural extraction from visual layout of documents. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 203–210. ACM, 2002. (Cited on page 102.)

[145] D. Roth and W. Yih. Relational learning via propositional algorithms: An information extraction case study. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1257–1263, 2001. (Cited on pages 23 and 35.)

[146] Markus Ruschhaupt, Wolfgang Huber, Annemarie Poustka, and Ulrich Mansmann. A compendium to ensure computational reproducibility in high-dimensional classification tasks. *Statistical Applications in Genetics and Molecular Biology*, 3(1):37, 2007. (Cited on pages 68 and 147.)

[147] Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288–297, 1990.

[148] R. Schapire, M. Rochery, M. Rahim, and N. Gupta. Incorporating prior knowledge into boosting. In *Proceedings of the 20th International Conference on Machine Learning*. Morgan Kaufmann, 2002. (Cited on page 122.)

[149] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001. ISBN 0262194759. (Cited on page 31.)

[150] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002. ISSN 0360-0300. (Cited on pages 6 and 21.)

[151] Fabrizio Sebastiani. Text categorization. In Alessandro Zanasi, editor, *Text Mining and its Applications to Intelligence, CRM and Knowledge Management*, pages 109–129. WIT Press, Southampton, UK, 2005. URL `http://www.isti.cnr.it/People/F.Sebastiani/Publications/TM05.pdf`. (Cited on page 28.)

[152] Burr Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *JNLPBA '04: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 104–107. Association for Computational Linguistics, 2004. (Cited on page 43.)

[153] Savino Sguera, Armando Stellato, Philippe Ombredanne, and Maria Teresa Pazienza. Software semantic provisioning: Actually reusing software. In *Semantic Web Applications and Perspectives*, 2007. (Cited on page 54.)

[154] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th international conference on Machine learning (ICML'07)*, pages 807–814. ACM, 2007. (Cited on page 130.)

[155] Wojciech Siedlecki and Jack Sklansky. On automatic feature selection. *Handbook of pattern recognition & computer vision*, pages 63–87, 1993. (Cited on page 60.)

[156] Kendal Simon and Creen Malcolm. *An Introduction to Knowledge Engineering*. Springer-Verlag New York, Inc., 2006. (Cited on page 7.)

[157] Sören Sonnenburg, Mikio L. Braun, Cheng Soon Ong, Samy Bengio, Leon Bottou, Geoffrey Holmes, Yann LeCun, Klaus-Robert Müller, Fernando Pereira, Carl Edward Rasmussen, Gunnar Rätsch, Bernhard Schölkopf, Alexander Smola, Pascal Vincent, Jason Weston, and Robert Williamson. The need for open source software in machine learning. *J. Mach. Learn. Res.*, 8:2443–2466, 2007. ISSN 1533-7928. (Cited on pages 68 and 147.)

[158] J. C. Spall. *Introduction to Stochastic Search and Optimization*. John Wiley & Sons, Inc., 2003. (Cited on page 112.)

[159] Charles Sutton and Andrew McCallum. Composition of conditional random fields for transfer learning. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 748–754. Association for Computational Linguistics, 2005. (Cited on pages 43 and 86.)

[160] Michael J. Swain and Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991. (Cited on page 18.)

[161] H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *IEEE Transaction on Systems, Man, and Cybernetics*, 6(8):460–472, 1978. (Cited on page 89.)

[162] K. Tomanek and F. Olsson. A web survey on the use of active learning to support annotation of text data. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning For Natural Language Processing*, 2009. (Cited on page 45.)

[163] Mihran Tuceryan and Anil K. Jain. *Texture analysis*. World Scientific Publishing Co., Inc., 1993. ISBN 981-02-1136-8. (Cited on page 18.)

[164] Mike Uschold and Michael Grüninger. Ontologies: principles, methods, and applications. *Knowledge Engineering Review*, 11(2): 93–155, 1996. (Cited on page 47.)

[165] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995. ISBN 0387945598. (Cited on page 30.)

[166] Luis von Ahn. Human computation. In *K-CAP '07: Proceedings of the 4th international conference on Knowledge capture*, pages 5–6, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-643-1.

[167] F. Wang and C. Zhang. Label propagation through linear neighborhoods. *IEEE Trans. on Knowl. and Data Eng.*, 20(1):55–67, 2008. (Cited on pages 118 and 129.)

[168] Hongling Wang, Guodong Zhou, Qiaoming Zhu, and Peide Qian. Exploring various features in semantic role labeling. *Advanced Language Processing and Web Information Technology, International Conference on*, 0:3–8, 2008. (Cited on page 102.)

[169] G. Xue, W. Dai, Q. Yang, and Y. Yu. Topic-bridged plsa for cross-domain text classification. In *Proceedings of the 31st Annual International SIGIR Conference*. ACM, 2008. (Cited on pages xv, 128, and 131.)

[170] Yiming Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1:67–88, 1997. (Cited on page 92.)

[171] Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. Faceted metadata for image search and browsing. In *CHI '03:*

Proceedings of the SIGCHI conference on Human factors in computing systems, pages 401–408, New York, NY, USA, 2003. ACM. ISBN 1-58113-630-7.

[172] Dmitry Zelenko, Chinatsu Aone, Anthony Richardella, Jaz K, Thomas Hofmann, Tomaso Poggio, and John Shawe-Taylor. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3, 2003. (Cited on page 8.)

[173] Tong Zhang and David Johnson. A robust risk minimization based named entity recognition system. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 204–207. Association for Computational Linguistics, 2003. (Cited on pages 99 and 101.)

[174] Xiang S. Zhou, Sonja Zillner, Manuel Moeller, Michael Sintek, Yiqiang Zhan, Arun Krishnan, and Alok Gupta. Semantics and cbir: a medical imaging perspective. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 571–580. ACM, 2008. (Cited on page 61.)

[175] X. Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin, Madison, 2005. URL `http://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf`. (Cited on page 44.)

[176] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Canergie Melon University, 2002. (Cited on pages 45, 111, and 118.)

[177] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings 20th International Conference on Machine Learning (ICML'03)*. ACM, 2003. (Cited on pages 45 and 116.)