A Machine Learning Approach for Adaptive Information Extraction

Edgar Gonzàlez i Pellicer
egonzalez@lsi.upc.edu

Director
Jordi Turmo Borràs

Memòria del DEA i Projecte de Tesi
Programa de Doctorat en Intel·ligència Artificial
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya

Maig 2006

Contents

Chapter 1

Introduction

1.1 Information Extraction

Information Extraction (IE) is a Natural Language Processing (NLP) area whose goal is to automatically generate structured pieces of information from the relevant information contained in text documents.

The type of documents used for IE can vary. Whereas some documents have been produced for computer use and are hence very structured, most of the documents are produced for human use and their text is *free*: they consist of Natural Language (NL) and lack an explicit structure. In some cases, for instance in many on-line documents, documents are semi-structured: chunks of NL text and structured pieces of information (e.g. metadata, tags) appear together.

Text documents in IE are usually from a restricted domain, and the set of concepts that are considered relevant forms the so-called *scenario of extraction*. This set of concepts is defined *a priori*, and the goal of IE is to extract instances of these concepts occurring in the documents. A sample scenario of extraction in an *aircraft crash* domain is the following:

> An **aircraft crash** is an event happening in a certain **site** and **date**. It involves an aircraft, which is of a **model**, belongs to an **airline** and is made by a **manufacturer**. The aircraft covers a flight, taking off from a **departure** and landing in a **destination**

The concepts defined in the scenario of extraction differ enormously from scenario to scenario. Scenarios consist of different types of elements, including entities, binary or n-ary relations between them and events in which these entities are involved. Properties of each one of these elements can be extracted from text.

As an example, Figure 1.1 shows a free text document from the *aircraft crash* domain. A template about each aircraft crash event referred in the text has to be filled. These templates contain a series of slots, such as crash site and date, crashed aircraft (model), airline, manufacturer and departure and destination of the flight. The templates of this type to be filled from the sample text are shown in Figure 1.2.

Another example, in this case a semi-structured text coming from the *seminar announcement* domain, is shown in Figure 1.3. The template in this scenario contains the slots speaker, location and start and end times of each announced seminar. The template filled from the information in this text is that in Figure 1.4.

A number of variations on IE tasks have been defined through the years, and new challenges for researchers in IE appear every year.

IE has become specially important in our times, as the amount of information available in natural language increases day by day, for instance in Internet. However, the origins of IE go back to the early eighties (or even before) when Text-Based Intelligent (TBI) systems began to

```
<DOC>
<DOCID> nyt960207.0722 </DOCID>
<TEXT> SEATTLE - It's the phone call no one wants to get, but everyone
knows might come one day. It came late Tuesday when Boeing got word
that a chartered 757 aircraft crashed shortly after takeoff from the
Dominican Republic. All 189 passengers are feared dead. The crash,
only the second in the history of the Boeing 757, came less than two
months after an American Airlines 757 slammed into a mountain as it
approached Cali, Colombia. Four people survived the Dec. 20 crash that
killed 160 people. The cause has not yet been determined. After
hearing the news of Alas Nacionales Flight 301 Tuesday night, members
of Boeing's Air Safety Investigation Group monitored the situation
throughout the night and quickly assembled a team of safety experts to
be on standby in case they were needed at the crash scene. One Boeing
air safety investigator was expected to arrive Thursday in Puerto
Plata to assist a team from the National Transportation Safety Board
and the Dominican Republic in trying to determine why the two-engine
jet crashed. More Boeing engineers will be called in if needed.
...
</TEXT>
</DOC>
```

Figure 1.1: Sample document from the aircraft crash domain

```
* Crash_Event1:
  - Crash_Site:   Dominican Republic
  - Crash_Date:   06-02-1996
  - Aircraft:     757
  - Airline:      Alas Nacionales
  - Manufacturer: Boeing
  - Departure:    Dominican Republic
  - Destination:  -

* Crash_Event2:
  - Crash_Site:   Cali (Colombia)
  - Crash_Date:   20-12-1995
  - Aircraft:     757
  - Airline:      American Airlines
  - Manufacturer: Boeing
  - Departure:    -
  - Destination:  Cali (Colombia)
```

Figure 1.2: Event templates extracted from document in Figure 1.1

```
<0.21.3.95.14.12.11.ed47+@andrew.cmu.edu.0>

Type: cmu.andrew.official.cmu-news
Topic: ECE Seminar
Dates: 30-Mar-95
Time: 4:00 - 5:00 PM
Place: Scaife Hall Auditorium
PostedBy: Edmund J. Delaney on 21-Mar-95 at 14:12 from andrew.cmu.edu
Abstract:


  COMPUTERIZED TESTING AND SIMULATION OF CONCRETE CONSTRUCTION

                    FARRO F. RADJY, PH.D.

                    President and Founder
                  Digital Site Systems, Inc.
                       Pittsburgh, PA

DATE: Thursday, March 30, 1995
TIME: 4:00 - 5:00 P.M.
PLACE: Scaife Hall Auditorium
REFRESHMENTS at 3:45 P.M.

-----------------------
```

Figure 1.3: Sample document from the seminar announcement domain

```
* Seminar_1:
  - Speaker:    Farro F. Radjy
  - Location:   Scaife Hall Auditorium
  - Start_Time: 4:00 p.m.
  - End_Time:   5:00 p.m.
```

Figure 1.4: Event template extracted from document in Figure 1.3

be used to automatically obtain information by manipulating documents, instead of relying on manual introduction of knowledge by human experts.

At that time roughly two major TBI areas were considered and distinguished: Information Retrieval (IR) and Information Extraction (IE). IR techniques aim at retrieving documents from a collection that accomplish a set of given restrictions, such as containing certain keywords. These documents are judged relevant according to the query that defines the restrictions, and may be used for an information acquisition process. IE technology, on the contrary, aims at a deeper understanding of the texts, to identify the relevant content within the relevant documents and extract it in a structured way, suitable for human inspection, for insertion in a structured database or for further automatic processing.

Whereas the use of NLP technology in IR is marginal and controversial, IE uses it widely. In order to detect the relevant information within the relevant documents, IE systems require significant amounts of linguistic knowledge. At the core of most IE systems lie a set of linguistic patterns which are used to extract the concepts present in the text: entities, relations, events... These extraction patterns are highly domain specific, an even within the same domain a change of writing style or language makes them ineffective. The acquisition of these patterns and the other required linguistic knowledge can become highly expensive.

The specificity of knowledge required for IE hence supposes a drawback on the portability of IE systems, as a change of document domain, writing style or language requires a costly process of new specific knowledge acquisition.

1.2 Machine Learning for Adaptive IE

The application of Machine Learning (ML) and corpus based approaches has produced competitive systems in many NLP tasks at a lower cost than their hand-crafted equivalents. The aforementioned portability bottleneck of IE has encouraged research in the application of ML techniques to the construction of adaptive IE systems, which could be easily ported to other domains, styles and languages.

ML approaches allow the acquisition of knowledge for IE at a lower cost than those approaches based on manual introduction of the knowledge by a human expert. However, in many cases a significant fraction of the burden of the acquisition task still lies on the user, as she is responsible for feeding the ML system the examples from which to learn. This is the case of supervised systems, which need hand-annotated corpora, or interactively demand examples to the user as needed (active learning).

As the cost of annotating the data or looking for the examples for these supervised approaches can also become large, efforts have been devoted to reduce the elements of supervision in the process of learning extraction patterns. This has given birth to a number of lightly supervised methods for IE pattern acquisition. In a world where the availability of large collections of unannotated data increases every day, lightly supervised methods offer a way to reduce the costs of porting IE systems.

1.3 Our Proposal

Even though research in lightly supervised approaches is giving promising results, the remaining elements of supervision can represent a significant human effort as the size of document collections grows. Lightly supervised methods may require from the user:

- The classification of the documents into the different domains present in the collection. In the case of a single domain, the classification of the documents into relevant or not to the target domain.
- The classification of a small set of documents (the *seed documents*) into the different domains present in the collection.

- A small set of patterns representative of the different domains present in the collection (*seed patterns*).
- The number of domains present in the collection.

All these elements require the user browsing and exploring through the unannotated corpus. In larger corpora, manual exploration becomes harder, and it will eventually become unfeasible by a human user. In addition, by giving supervision the user is introducing a bias into the learning process. In this context, the utility of clustering techniques as an automatical tool for exploratory analysis of data collections is well known [Hartigan, 1975, Dimitriadou, 2003].

The proposal of this thesis project is to enhance pattern learning with document clustering techniques to reduce the elements of human supervision required for the pattern learning process. The goal is the development of a methodology that from a completely unannotated collection of documents and without the need of any kind of user-defined seed documents or patterns produces good quality patterns useful for IE and other NLP tasks.

A baseline methodology for this combination has already been developed. Oncoming research in this thesis project will look for improving the quality of the learned patterns.

1.4 Project CHIL and Evaluation through Text Mining

The research in the thesis project is carried out within the framework of project CHIL (Computers in the Human Interaction Loop)[1], Integrated Project of the 6th Frame Project of the European Union (IST-2004-506909).

The aim of CHIL is to develop technologies giving support to human-to-human communication in a non-intrusive, implicit and indirect way. One of the research areas involved in this problem is NLP, and examples of NLP technologies useful in this context are those related to Text Mining, concretely Question Answering (QA) and Automatic Summarization.

Recently, research in the area is orienting towards the use of IE technology to improve the results of QA and Summarization. Linguistic patterns which allow to extract events and relations between the entities appearing in the document can be useful to the extraction of answers or summaries. An example of research in this direction is the work of Girju [2003].

However, as said before, linguistic patterns depend on the semantic domain and on the characteristics of documents. Within CHIL, we find that:

- Human-to-human communication happens in the most varied scenarios, for instance in meetings or in presentations about different topics. The world knowledge and the linguistic knowledge implied in each of them differs dramatically, as the concepts involved in each situation and their linguistic realizations are different.
- The documents to treat in each scenario can be of diverse nature: oral transcripts, written text, web pages, slides... Each type of document presents particular characteristics: the well-formedness of written text against the ill-formedness of oral transcripts or structured web pages; the lack of punctuation and capitalization of oral transcripts and the high word error rate in some transcripts due to low precision of Automatic Speech Recognizers (ASR)... As a consequence, the linguistic knowledge useful for Text Mining tasks also depends of the kind of document.

In this scenario, it is clear that adaptive techniques to learn linguistic patterns from unannotated text are useful. The diversity of document formats, domains and styles requires flexible and adaptive approaches to pattern acquisition. We believe that clustering-enhanced pattern learning is useful for this task. Besides, the Text Mining tasks of QA and Summarization provide an excellent framework for extrinsic evaluation of the learned patterns.

[1] `http://chil.server.de`

1.5 Overview of this Document

The rest of this thesis project is organized as follows: Chapter 2 gives a review of the state of the art and the development of IE systems. Chapter 3 presents our approach and the state of the work that has been carried so far. Our experiments and their results are contained in chapter 4. Chapter 5 presents the work plan for our oncoming research. Last, chapter 6 lists the publications that our work has produced so far, and locates them in the context of our project.

Chapter 2

State of the Art

We will present an overview of the development of IE (Section 2.1), the architecture of IE systems (Section 2.2) and the application of ML for IE, focusing on the research towards unsupervised approaches (Section 2.3). A sketch of the evolution of evaluation in IE is also presented (Section 2.4).

2.1 Development of Information Extraction

One of the first reported IE systems operating on texts of unrestricted topic was FRUMP, implemented by DeJong [1979, 1982]. FRUMP monitored a newswire using simple scripts to cover news stories. Sager [1981] mentions an even earlier project, before 1970, directed by Naomi Sager herself, from the Linguistic String Project Group at New York University. Sponsored by the American Medical Association, this work sought to convert patient discharge summaries (filled out in English) into a structure for a traditional database management system.

Nevertheless, the development of IE is clearly tied to the MUC (Message Understanding Conference) conferences[1], held from 1987 to 1998. The main objective of the MUC conferences was to promote research in IE and to provide IE systems with a quantitative evaluation method.

The first MUC conference was started by the US Navy (the Naval Ocean Systems Center, San Diego) and they were subsequently sponsored by the United States Defense Advanced Research Projects Agency (DARPA)[2]. In 1990, DARPA launched the TIPSTER[3] text program to fund the research efforts of several of the MUC participants.

Seven MUC conferences were held:

MUC-1 (1987) It was mainly exploratory. The organization did not define any task or evaluation criteria, but selected *Naval Tactical Operations* as document domain. Each group designed its own format to record the extracted information.

MUC-2 (1989) The same domain was selected but a task was defined: *Template Filling*. A 10-slot template was defined, but the evaluation was still done by the participants themselves.

MUC-3 (1991) The domain changed to *Latin American Terrorism*, and the given template consisted of 18 slots. Common evaluation measures (precision and recall) were defined, and the document collection was split into training and test sets. This framework allowed a quantitative comparison of the different competing systems.

MUC-4 (1992) The domain did not change from MUC-3, but the template was modified and increased to 24 slots. The F measure, an harmonic mean of precision and recall, was introduced to allow global comparisons among the systems.

[1] `http://www.itl.nist.gov/iaui/894.02/relatedprojects/muc/`
[2] `http://www.darpa.mil/`
[3] `http://www.fas.org/irp/program/process/tipster.htm`

MUC-5 (1993) Two domains, *Joint Ventures* and *Microelectronics* were proposed, and documents in Japanese were included in the collections. Two different sets of object-oriented templates were used, and the evaluation included *error-based metrics*, as well as the MUC-3 and MUC-4 *recall-precision-based metrics*.

MUC-6 (1995) The *Financial* domain was tested. Moreover, a series of IE subtasks were identified, following three main goals that were proposed:

- Identifying domain-independent components from the whole set of components being developed. To meet this goal, the organisers proposed the *Named Entity* (NE) subtask, which involved recognizing and classifying entities (organizations, people, locations), times (dates, times) and quantities (monetary values, percentages).
- Focusing on the portability of the IE task to different event classes. The organization proposed to standardize low-level objects (people, organizations...) since they were involved in many different types of events. The *Template Element* (TE) subtask was proposed with this aim. The old-style MUC task of detecting the events in which the template elements were involved was named *Scenario Template* (ST) task.
- Encouraging work on deeper understanding. Three more subtasks were proposed with this goal, namely *Coreference Resolution* (CO), *Word Sense Disambiguation* and *Predicate-Argument Syntactic Structuring*. Finally, only the first one was evaluated.

MUC-7 (1998) The English, Japanese and Chinese languages were used, and the new task of *Template Relation* (TR) was evaluated. This task required the identification of relations such as *location-of*, *employee-of* and *product-of* holding between template elements. The *Airline Crashes* domain was used for training, and the *Launch Events* domain was used for testing.

In parallel with the MUC conferences, the European Commission funded under the LRE (Linguistic Research and Engineering)[4] program a number of projects devoted to developing tools and components for IE (also for IR), such as automatically or semi-automatically acquiring and tuning lexicons from corpora, extracting entities or parsing in a flexible and robust way. These projects included ECRAN[5], SPARKLE[6], FACILE[7], and AVENTINUS[8].

Beyond the MUCs, research on IE technology has been included in the TIDES (Translingual Information Detection, Extraction and Summarization)[9] program funded by DARPA in 1999. It is an initiative on fast machine translation and information access, including translingual IE technology. Some of the sponsored projects have been RIPTIDES[10], PROTEUS[11], CREST[12], Coreference.com[13], and the UMass system[14].

The primary IE evaluation framework used by the TIDES program is the ACE (Automatic Content Extraction) program[15]. The ACE program was started in 1999 with the aim of developing automatic content extraction technology to extract information from human language in textual form. The ACE evaluations present greater challenges than MUC evaluations:

- The corpus includes stories from three different sources: newswire (text), broadcast news (speech - ASR transcribed) and newspaper (image - OCR transcribed). From 2005 on,

[4] http://www2.echo.lu/langeng/en/lehome.html
[5] http://www.dcs.shef.ac.uk/research/ilash/Ecran/
[6] http://www.informatics.susx.ac.uk/research/nlp/sparkle/sparkle.html
[7] http://tcc.itc.it/research/textec/projects/facile/facile.html
[8] http://www.dcs.shef.ac.uk/nlp/funded/aventinus.html
[9] http://www.darpa.mil/ipto/programs/tides/
[10] http://www.cs.cornell.edu/home/cardie/tides/
[11] http://nlp.cs.nyu.edu/
[12] http://crl.nmsu.edu/Research/Projects/Crest/
[13] http://www.coreference.com/lingpipe/
[14] http://ciir.cs.umass.edu/research/tides.html
[15] http://www.nist.gov/speech/tests/ace/

documents coming from conversational telephone speech, Usenet newsgroups, discussion forums and weblogs are also included (although not for all languages).

- From 2003 on, documents in Chinese and Arabic are included in the corpus, in addition to documents in English.
- The Entity Detection and Tracking (EDT) subtask requires detecting mentions of entities and chaining them together by identifying their coreference links. In addition, there is a hierarchy of types, subtypes and classes of entities, as well as levels of mentions (names, nominal expressions or pronouns).
- From 2001 on, the Relation Detection and Characterization (RDC) task requires the identification of explicit and implicit relations between a previously identified set of entities. There is also a variety of types and subtypes of relations.
- From 2004 on, the TIMEX[16] task requires the recognition and normalization of date and time expressions.
- In 2004, the Event Detection and Recognition (VDR) task was first proposed, even if in the end it was postponed until ACE-05. The concept of an ACE event is simpler than MUC's, being an event involving zero or more ACE entities, values and time expressions. Five different types of events were defined.
- The evaluation is similar in spirit to MUC's, relative to the output given by a reference model.

More recently, and in parallel with the TIDES program, the European Commission has been funding the Pascal[17] Network of Excellence. This organization, together with the Dot.Kom[18] European project has launched the Challenge on Evaluation of Machine Learning for Information Extraction from Documents, which involves three tasks (namely a full scenario task, an active learning task and an enriched scenario task) in the domain of a Workshop Call for Papers semi-structured texts. The first edition of this challenge started in June 2004, and the formal evaluation took place in November 2004. However, the framework of comparison is ML oriented rather than IE oriented. The evaluation framework, including the measures, and the type of documents also differ from those in ACE.

2.2 General Architecture of an Information Extraction System

The first IE systems taking part in MUC used traditional Natural Language Understanding architectures, based on full parsing, a semantic interpretation of the resulting in-depth syntactic structure, and discourse analysis. However, in MUC-3 the best scores were obtained by Lehnert et al. [1991], which used a simpler approach named *selective concept extraction*. Traditional parsing, interpretation and discourse analysis were replaced by a simple phrasal parser, an event pattern matcher and a template merging procedure.

In MUC-4 Appelt et al. [1992] presented a similar approach, but in terms of a more flexible model, based on finite-state transducers. This lead to the definition by Hobbs [1993] at MUC-5 of a generic architecture for IE systems as:

> An information extraction system is a cascade of transducers or modules that at each step add structure and often lose information, hopefully irrelevant, by applying rules that are acquired manually and/or automatically.

[16] TIDES Standard for the Annotation of Temporal Expressions (`http://timex2.mitre.org`)
[17] `http://www.pascal-network.org/`
[18] `http://www.dot-kom.org`

The modules that Hobbs [1993] considered were:

1. Text Zoner, which turns a text into a set of text segments.
2. Preprocessor, which turns a text or text segment into a sequence of sentences, each of which is a sequence of lexical items, where a lexical item is a word together with its lexical attributes.
3. Filter, which turns a set of sentences into a smaller set of sentences by filtering out the irrelevant ones.
4. Preparser, which takes a sequence of lexical items and tries to identify various reliably determinable, small-scale structures.
5. Parser, whose input is a sequence of lexical items and perhaps small-scale structures (phrases) and whose output is a set of parse tree fragments, possibly complete.
6. Fragment Combiner, which tries to turn a set of parse tree or logical form fragments into a parse tree or logical form for the whole sentence.
7. Semantic Interpreter, which generates a semantic structure or logical form from a parse tree or from parse tree fragments.
8. Lexical Disambiguation, which turns a semantic structure with general or ambiguous predicates into a semantic structure with specific, unambiguous predicates.
9. Coreference Resolution, or Discourse Processing, which turns a tree-like structure into a network-like structure by identifying different descriptions of the same entity in different parts of the text.
10. Template Generator, which derives the templates from the semantic structures.

This simplification of the understanding process was widely adopted by all the IE community, specially due to the drawbacks of full parsing in terms of cost, robustness and coverage. Nowadays, the usual structure of an IE system is that depicted in Figure 2.1, although specific systems are characterized by their own set of modules. At the core of the system lies a set of Extraction Patterns, highly domain-specific and which capture the entities, relations and events present in the text.

2.3 Machine Learning for Information Extraction

Many IE systems have their knowledge hand-coded by human experts. This implies that the porting of an existing system to a different domain, writing style or language is expensive in terms of human effort. Even if some components are domain-independent, knowledge such as concept hierarchies or extraction patterns depends heavily on the domain. Porting thus requires a complete reengineering process, usually by someone with in-depth familiarity with the system.

To reduce this cost, two different strategies have been considered:

- Development of tools to aid the human experts in the task of adapting the system to new scenarios, and to decrease the required level of familiarity with the system. An example of such tools is that in Yangarber and Grishman [1997]. An interface is proposed that allows the user to easily annotate examples of relevant events, which are automatically translated into the appropriate specific patterns and generalized to cover syntactic variants (passive, relative clause...). The user can also generalize the patterns semantically. In this way, it is expected that users with shallow knowledge of the system can port it to different scenarios with lower effort. However, the user still has to browse the corpus to find the appropriate set of examples, and to decide when to stop the acquisition process.
- Use of ML techniques to acquire the required knowledge.

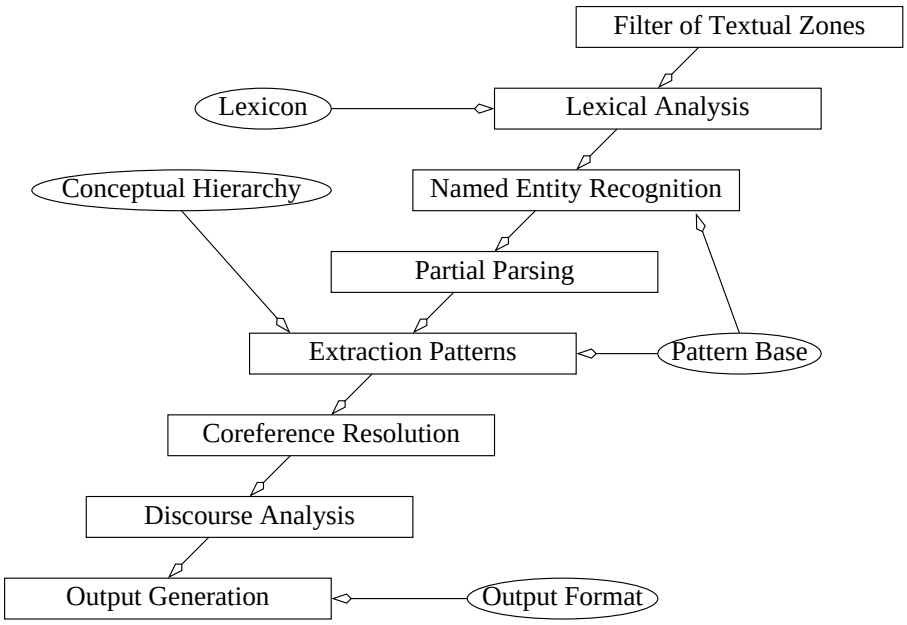


Figure 2.1: Usual architecture of an IE system

We will focus here on the second approach. Research on ML for IE has been encouraged by the success of corpus-based approached in other NLP tasks [Young and Bloothoft, 1997, Manning and Schütze, 1999]. Surveys of the application of ML methods to IE can be found in [Cardie, 1997, Yangarber and Grishman, 2000, Turmo et al., 2006].

Many different learning algorithms and kinds of knowledge have been used for IE, including:

- Hidden Markov Models [Freitag and McCallum, 2000, Ray and Craven, 2001, Skounakis et al., 2003],
- Relational Markov Networks [Bunescu and Mooney, 2004],
- Probabilistic Context-Free Grammars with Head Rules [Miller et al., 1998, 2000],
- Maximum Entropy [Chieu and Ng, 2002, Kambhatla, 2004],
- Dynamic Bayesian Networks [Peshkin and Pfeffer, 2003],
- Conditional Random Fields [McCallum and Jensen, 2003, Cox et al., 2005]
- Hyperplane Separators, [Roth and Yih, 2001, Sun et al., 2003, Chieu et al., 2003, Zelenko et al., 2003, Finn and Kushmerick, 2004, Zhao and Grishman, 2005].

However, the most usual approach, and the most interesting for the purposes of this thesis project, is the acquisition of knowledge in form of rules. [Muslea, 1999] and [Glickman and Jones, 1999] review the different types of rules used for IE. Rules can be *single-slot*, if they allow to extract a single slot of the templates, or *multi-slot*, if they allow the extraction of several slots at a time. Both propositional learning and relational learning have been used as learning paradigms to acquire the rules.

An example of a single-slot rule for IE is that in Figure 2.2, learned by RAPIER [Califf, 1998], to extract the *city* slot in a *job offering* domain. The rule expresses that a list of at most 2 proper nouns (NNP) preceeded by preposition (IN) *in* and followed by a comma and a proper noun of a state can be extracted as the *city* of the *job offering*. This rule extracts *Atlanta* from the sentence *located in Atlante, Georgia* and extracts *Kansas City* from *offices in Kansas City, Missouri*.

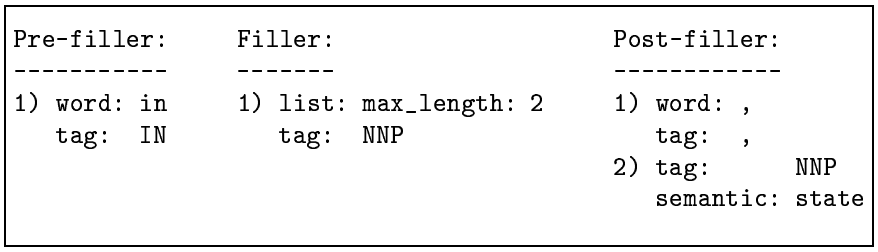


Figure 2.2: IE rule learned by RAPIER

As mentioned in the introduction, some approaches for learning are supervised: the relevant concepts present in the corpus must be previously annotated by a human user to allow the learning process; or it is the system which interactively demands the examples to the user, as in the case of active learning. As said, the cost of this supervision has encouraged research in lightly supervised approaches, in which human participation is reduced.

Tables 2.1 and 2.2 contain an overview of the most relevant ML systems learning rules for IE from free or semistructured text.

Table 2.1 contains supervised approaches. For each one of them it gives the learning paradigm **LP** (either propositional **P** or relational **R** learning), the strategy used by the system, whether it can learn single-slot **S** or multi-slot **M** rules, and the type of document from which it learns: structured **ST**, semistructured **S** or free **F** text.

Table 2.2 contains lightly supervised approaches. In this case, column **S/M** has been removed, as it is usually the user who has to annotate the slot-fillers within the patterns learned by the system to create the extraction rules. We can find instead a column which indicates which elements of supervision the approach requires.

A brief exposition of the listed systems follows.

2.3.1 Supervised Approaches for IE Rule Learning

Supervised methods are those where the learning process requires a corpus where relevant concepts are annotated. The applicability of these approaches hence depends on the availability of such resources. Most evaluations, like MUC or ACE, provide this kind of corpora to groups taking part in the evaluation to encourage the research on IE. Supervision can also be online, if the system requests examples to the user as the learning progresses.

2.3.1.1 Propositional Learning

AutoSlog One of the first systems to incorporate the use of ML for IE was AutoSlog, developed by Riloff [1993]. AutoSlog uses a heuristic-driven specialization strategy to generate extraction rules (called *concept nodes*) from a corpus where the slot fillers for each document have been annotated. The method works as shown in Algorithm 1.

The predefined heuristics are the most important part of AutoSlog, and are a set of linguistic patterns which determine a *trigger word* and a set of *enabling conditions*. For instance, a possible heuristic is `<subject> passive-verb`. If we are dealing with a document from the terrorism domain, whose **physical target** slot is *public buildings*, and find the sentence in Figure 2.3, the sentence would be generalized using the heuristic to a *concept node* whose trigger word would be *bombed*, whose enabling condition would be the passive form of the verb, and which would extract the subject of the clause to fill the **physical target** slot.

The rules have to be reviewed manually because the process can generate dubious or very specific slot filler definitions.

Approach	LP	Strategy	S/M	DT
AutoSlog [Riloff, 1993]	P	Heuristic driven specialization	S	F
PALKA [Kim and Moldovan, 1995]		Candidate elimination	S/M	
CRYSTAL [Sodeland et al., 1995]		Bottom-up covering		
[Chai and Biermann, 1997]		Brute force		
TIMES [Chai et al., 1999]			S	
LIEP [Huffman, 1995]	R	Bottom-up covering	M	
RAPIER [Califf, 1998]		Bottom-up compression	S	S
SRV [Freitag, 1998a]		Top-down covering		
WHISK [Soderland, 1999]			S/M	ST/S/F
EVIUS [Turmo and Rodríguez, 2002]				S/F

Table 2.1: Supervised ML approaches learning rules for IE

Approach	LP	Strategy	Supervision Elements	DT
AutoSlog-TS [Riloff, 1996]	P	Heuristic driven specialization	Classified document collection	F
[Basili et al., 2000]		Heuristic driven generalization		
[Harabagiu and Maiorano, 2000]		Heuristic driven specialization	Keywords	
QDIE [Sudo, 2004]		Exhaustive Search		
ESSENCE [Català et al., 2003]		Bottom-up covering	Keywords Extraction Synsets	
[Riloff and Jones, 1999]		Bootstrapping	Seed lexicon words	
DIPRE [Brin, 1998]			Seed relations	S
SnowBall [Agichtein and Gravano, 2000]				F
ExDISCO [Yangarber et al., 2000] [Yangarber, 2003]			Seed patterns	
[Stevenson and Greenwood, 2005]				
[Surdeanu et al., 2006]			Seed documents	

Table 2.2: Lightly supervised ML approaches learning rules for IE

Algorithm 1 AutoSlog

```
for all documents and slot fillers do
    Find the clause c where the first appearance of the slot filler string in the document occurs
    repeat
        Try to generalize the clause c using the set of heuristics to obtain a concept node
        if could not generalize then
            Find clause c where next appearance of the slot filler string occurs
        end if
    until could generalize ∨ reached end of document
    if could generalize ∧ concept node not already proposed then
        Propose to user for review and acceptance
    end if
end for
Return accepted concept nodes
```

PALKA PALKA [Kim and Moldovan, 1995] is based on a candidate elimination algorithm and an ad-hoc semantic hierarchy, defined for each domain. The rules learned are named Frame-phrasal Pattern Structures (FP-Structures), and include lexical and semantical constraints. The user must provide a frame for each event to be captured, defining its position in the event hierarchy, the semantic classes of all the involved entities, and a set of keywords which will be used by the system to select the training examples from the corpus. The process then follows the steps in Algorithm 2.

Algorithm 2 PALKA

```
Given a frame to learn
repeat
    Retrieve the set of candidate clauses C, using the given keywords
    for all c ∈ C do
        Apply FP-structures learned so far to clause c
        if non-existent event found then
            Specialize matching FP-structure within hierarchy, to prevent matching
        else if existent event not found then
            Create new FP-structure
            Generalize new FP-structure within hierarchy
        end if
    end for
until found FP-structures cover all initial specific ones
Return the found FP-structures
```

The use of the semantic hierarchy allows PALKA to learn rules that are more general than AutoSlog's. However, no generalizations are made on the trigger words.

CRYSTAL The rules learned by CRYSTAL [Sodeland et al., 1995] are also in the form of concept nodes, but they consist of a set of features related to the slot-fillers and the trigger. Values for these features can be terms, heads, semantic classes, syntactic relations (subject, direct

> in la oroya, junin department, in the central peruvian mountain range, *public buildings* were bombed and a car-bomb was detonated.

Figure 2.3: Example sentence from the terrorism domain

or indirect object), verbal modes, etc. CRYSTAL learns an initial set of specific rules, one for every example in the training corpus, and then uses a bottom-up covering algorithm in order to relax such features. This relaxation is achieved by means of dropping out irrelevant features and generalizing semantic constraints by using an ad-hoc semantic hierarchy. The process is detailed in Algorithm 3

Algorithm 3 CRYSTAL

```
1: for all examples e do
2:    Insert initial specific rule covering e into rule dictionary
3: end for
4: while dictionary contains initial rules do
5:    δ ← initial rule extracted from dictionary
6:    loop
7:       δ' ← the rule most similar to δ
8:       if ∄ δ' then
9:          Exit loop
10:      end if
11:      ω ← the unification of δ and δ'
12:      if error of ω on training > tolerance then
13:         Exit loop
14:      end if
15:      Remove all rules covered by ω from dictionary
16:      δ ← ω
17:   end loop
18:   Insert δ into dictionary
19: end while
20: Return rule dictionary
```

TIMES Chai and Biermann [1997] use WordNet[19] [Miller et al., 1990, Fellbaum, 1998], a broad coverage semantic hierarchy. The basic element of WordNet is a synset, which corresponds to a group of word senses referring to the same concept, as related by a loose kind of synonymy. The system of Chai and Biermann [1997] starts by assigning to each word its most frequent sense in WordNet (though the user can change it manually) and then generalizing specific rules by moving the senses to the top of the hierarchy. Recall is expected to be as high as possible by this method. In a second phase, these top synsets are specialized to improve precision, through interaction with the user and an analysis of the training set. TIMES [Chai et al., 1999] is a more recent version of this approach.

WAVE WAVE [Aseltine, 1999] is an incremental learning approach similar to CRYSTAL, in which a reusable hierarchy of partially learned rules is maintained along the process.

2.3.1.2 Relational Learning

Given that IE deals with entities, their properties and relations among them, relational learning seems a natural choice to learn knowledge useful for the task. The examples and concepts are represented in this framework in terms of first order logic.

LIEP An example of this kind of systems is LIEP [Huffman, 1995]. LIEP automatically infers multi-slot rules from training examples of events occurring in free text by using a bottom-up covering algorithm. If a given example is not matched by any learned rule, LIEP tries to find a rule to generalize. If this is not possible, LIEP builds a new specific rule from the example. Specific

[19] http://wordnet.princeton.edu/

rules used by LIEP consist of a feature set, similar to that used by CRYSTAL. However, LIEP has no prior information about the syntactic relations between chunks. LIEP learns such relations (`subject(A,B)`, `object(A,B)`, `prep_object(A,B)`, etc.) by using a form of explanation-based learning with an over-generated and incomplete theory. As opposed to most approaches, LIEP requires the user to interactively annotate training examples within sentences until she considers there is enough coverage.

SRV Other approaches are based on general relational learning systems, and more specifically, on Inductive Logic Programming (ILP) systems well known by the ML community (e.g., FOIL [Quinlan, 1990, Quinlan and Cameron-Jones, 1993], CIGOL [Muggleton and Buntine, 1988], GOLEM [Muggleton and Feng, 1992], CHILLIN [Zelle and Mooney, 1994] and PROGOL [Muggleton, 1995]).

SRV [Freitag, 1998a], for instance, is an ILP system based on FOIL. SRV transforms the problem of learning IE rules into a classification problem: deciding whether a document fragment is a possible slot value. The input of this system is a training set of documents, and a set of attributive and relational features related to tokens T (e.g., `capitalized(T)`, `next(T1, T2)`) that control the generalization process. Introducing domain-specific linguistics or any other information is a separate task from the central algorithm, which is invariable and domain independent. SRV uses a top-down covering algorithm to learn IE rules from positive and negative examples. The annotated slot-filler fragments within training documents are the positive examples. Negative examples are automatically generated taking into account empirical observations related to the number of tokens of positive examples: if positive examples are sequences of between MIN to MAX tokens, negative examples are the rest of sequences of between MIN to MAX tokens in the training corpus.

Freitag [1998b] proposes a multi-strategy approach, combining the results of SRV with a simple rote memorization and the term-space Naive Bayes approach of Freitag [1997]. For each classifier, a regression model mapping confidence to probability is trained using hold-out data removed from the training corpus. The probabilities given by each classifier are then combined using different strategies. The combination is shown to outperform its individual components.

RAPIER RAPIER [Califf, 1998] is based on GOLEM, CHILLIN and PROGOL. It uses a bottom-up compression algorithm: training examples are considered specific rules. At each iteration, two rules (specific or not) are selected to be compressed into a new one. Rules used in this process are discarded and the resulting rule is added to the set of possible ones. The input documents are represented as token sequences, optionally POS tagged. No parsing process is required for them. Each training example consists of three text fragments, where one of them is the slot-filler value and the other two are the unbounded contexts to the left and right of the value. In order to learn rules from such examples, RAPIER takes into account the implicit token succession relation and some token generalizations: token sets, POS tags or semantics derived from the WordNet hierarchy. Thompson et al. [1999] propose an incremental version of RAPIER using active learning.

WHISK A more flexible approach within the relational learning paradigm is WHISK [Soderland, 1999]. In addition to free text, WHISK can deal with structured and semistructured documents. Following a different approach, WHISK represents documents as sequences of tokens, some of them tags representing meta-data (HTML tags, delimiters of parsed chunks, features of heads,...) and allows learning of both single-slot and multi-slot rules to extract exact slot values. Rules are represented as pairs `< pattern; output >`, in which `pattern` is a regular expression meant to be matched by documents and `output` specifies the output template to be given when a match occurs. These rules are learned in a top-down fashion from a training set of positive examples.

An unusual selective sampling approach is used by WHISK. Initially, a set of unannotated documents is randomly selected as training input out of those satisfying a set of keywords. These documents are presented to the user who tags the slot-fillers. WHISK starts by learning a rule from the most general pattern. The growth of the rule proceeds one slot at a time. This is done

by adding tokens just within the slot-filler boundaries as well as outside them. The growth of a rule continues until it covers at least the training set. After a rule set has been created, a new set of unannotated documents can be selected as a new training input from those satisfying the rule set. Although WHISK is the most flexible state-of-the-art approach, it cannot generalize on semantics when learning from free text, and no negative constraints can be learned.

EVIUS EVIUS [Turmo and Rodríguez, 2002, Turmo, 2002] uses a multi-strategy approach within the relational learning paradigm to learn single-slot and multi-slot IE rules from free text and semi-structured documents. The learning systems explained so far learn single concept extractions. They learn knowledge useful to extract instances of each concept in the scenario of extraction independently. Instead, EVIUS assumes the fact that the scenario of extraction imposes some dependencies among concepts to be dealt with. When one concept depends on another one, knowledge about the former is useful for learning to extract instances of the target. EVIUS is a supervised multi-concept learning system based on a multi-strategy constructive learning approach [Michalski, 1993] that integrates closed-loop learning, deductive restructuring [Ko, 1998] and constructive induction. Closed-loop learning allows EVIUS to incrementally learn IE rules similar to Horn clauses for the whole scenario of extraction. This is done by means of determining which concept to learn at each step. Within this incremental process, the learning of IE rules for each concept is basically accomplished using FOIL, which requires positive and negative examples. Positive examples are annotated in the training data using an interface, while negative examples are automatically generated. Once IE rules for a concept have been learned, the learning space is updated using deductive restructuring and constructive induction. These techniques assimilate knowledge which may be useful for further learning: the training examples of learned concepts and new predicates related to these concepts.

2.3.2 Lightly Supervised Approaches

As we have already mentioned, the main bottleneck of supervised pattern learning methods is the cost of providing the system with the examples to learn from, for instance, by annotating a corpus. Lightly supervised approaches reduce the required supervision from the user. In addition to supervision previous to the process of learning, the user usually has to annotate the slot-fillers in the learned extraction patterns. Nevertheless, the overall cost is reduced, and most of these approaches can benefit from completely unannotated data. For these reasons, they are attractive for adaptive IE.

2.3.2.1 Heuristic-driven Approaches

AutoSlog-TS One of the first steps toward the reduction of supervision in learning methods for IE was AutoSlog-TS [Riloff, 1996]. Being an less supervised version of AutoSlog, it uses a similar approach but instead of requiring an annotated corpus, it only requires the classification of the documents in the collection as either relevant or unrelevant to the task. The method works as described in Algorithm 4.

The heuristics used in AutoSlog-TS are the same proposed for AutoSlog plus three new ones. On the MUC-4 corpus in the terrorism domain, AutoSlog-TS obtained higher precision but slightly lower recall than AutoSlog. However, the rule dictionary constructed by AutoSlog-TS was less than half the size of that of AutoSlog.

[**Basili et al., 2000**] The system of Basili et al. [2000] requires a general corpus with the documents in it classified into a set of specific domains. Verbs are used as event triggers, and a statistical χ^2 test is applied to all verbs in the corpus. The assumption of Basili et al. [2000] is that those verbs whose distribution in a domain is significantly different from that in the whole corpus work as event triggers.

For each one of these verbs a set of verb sub-categorization structures is extracted applying a conceptual clustering algorithm. Each occurrence of the verb in the parsed sentences of

Algorithm 4 AutoSlog-TS

1: Given a corpus divided into relevant and irrelevant documents $\mathcal{D} = \{d_1 \ldots d_n\} = \mathcal{R} \cup \bar{\mathcal{R}}$
2: Generate all possible patterns π, using a set of heuristics
3: Find relevance rate of each pattern π, defined as the probability that a document is relevant if it contains the pattern

$$rel(\pi) = P(d \in \mathcal{R} \mid \pi \in d) = \frac{\|d \mid d \in \mathcal{R} \wedge \pi \in d\|}{\|d \mid d \in \mathcal{D} \wedge \pi \in d\|}$$

4: Sort patterns by **RlogF** score:

$$\begin{aligned} rlogf(\pi) &= rel(\pi) \cdot \log_2 freq(\pi) \\ freq(\pi) &= \|d \mid d \in \mathcal{D} \wedge \pi \in d\| \end{aligned}$$

5: Return those patterns π with $rlogf(\pi) > 0.5$

the corpus is translated into a vector of binary `<attribute> : <value>` features in the form `<syntactic_relation> : <argument_head>` in order to be clustered. The resulting clusters represent the different verb sub-categorization structures, each one with its corresponding specific patterns.

The heads of the structures are semantically tagged using WordNet synsets and the resulting specific patterns are generalized by using the following heuristics:

- Synsets of noun heads are semantically generalized using a measure of conceptual density [Agirre and Rigau, 1996].
- Patterns are expanded via linguistically principled transformations (e.g., passivization, potential alternations).

Finally, multi-slot IE rules are built from these generalized event patterns by manually marking the argument that fills each slot of a pre-defined event template. Validation from the user is necessary to eliminate possible noisy verbs and overly specific patterns obtained during the learning process.

[**Harabagiu and Maiorano, 2000**] The starting point of Harabagiu and Maiorano [2000] is the observation that less than 8% of the relations in typical IE scenarios are present in WordNet. The method proposed first requests from the user a set of keywords which define the domain of extraction. References to these keywords are searched in WordNet (synsets, taxonomic relations, occurrences in glosses...) and, using a series of heuristics to detect implicit relations, an enriched semantic space of the domain is built. The corpus is scanned to find collocations of domain concepts in subject, verb and object position of clauses, and a set of linguistic patterns is generated. The last step is to select only the most general linguistic patterns. However, no automatic method to deal with this selection is suggested by the authors, and no evaluation of the system is presented.

2.3.2.2 Covering Approaches

ESSENCE ESSENCE [Català et al., 2003] extracts patterns from observations instead of examples. The required input is a *Task Definition*: for each slot of the output template, the set of *Extracting Synsets* (semantic values the slot-filler can take) and the set of *Context Keywords* (words that usually appear near the information that is to be extracted). The process works as shown in Algorithm 5.

The process of generalization uses a bottom-up covering algorithm and the semantic relations in WordNet. After the learning, the user has to validate the patterns and identify the slot-fillers actually occurring in them. The process may be repeated using the already learned patterns and a new set of observations found with new keywords.

Algorithm 5 ESSENCE

1: **for all** slots in the template **do**
2: Request the sets of *Extracting Synsets* and *Context Keywords* for the slot
3: Retrieve sentences s from the document collection $\mathcal{D}$ which contain some of the keywords
4: Create observations o taking windows of fixed size surrounding the keywords in each sentence
5: Semantically annotate each observation with WordNet synsets
6: Filter those observations which do not contain any of the *Extracting Synsets*
7: Construct a specific pattern from each observation
8: Find the set of generalized patterns covering all specific patterns
9: **end for**

2.3.2.3 Exhaustive Search Approaches

QDIE Query-Driven Information Extraction (QDIE) is the proposal of Sudo [2004], aimed towards cross-lingual IE. The system relies on an Information Retrieval system to recover a set of relevant documents given a set of keywords. The patterns present in these documents are ranked using $tf \cdot idf$ and the best patterns are selected. The user has then to annotate the slot-fillers present in the patterns.

Three different pattern models are tested: predicate-argument (direct syntactic relation between two nodes), chain (chain of syntactic relations between two nodes) and subtree (syntactic subtrees). To avoid the combinatorial explosion of an exhaustive search in the syntactic subtree space, the right-most expansion base subtree discovery algorithm of Abe et al. [2002] is used.

Sudo [2004] uses this algorithm as a component for cross-lingual IE. A series of keywords given in the target language are translated to the source language, where the QDIE monolingual system is applied. The extracted information is then translated to the target language. The performance of such approach is shown to be better than that of translation-based IE: translating the source documents to the target language and then applying a monolingual IE system on them.

2.3.2.4 Bootstrapping Based Approaches

Bootstrapping [Yarowsky, 1995, Abney, 2004] is a learning approach which has been shown useful for several NLP tasks, and which has also been applied to the specific task of pattern learning for IE. All the proposed approaches are based on the use of a starting set of seed examples or seed patterns from which context conditions can be learned. These conditions allow to hypothesize new positive examples, which in turn allow to learn new context conditions, and so on.

[**Riloff and Jones, 1999**] Riloff and Jones [1999] use a multi-level bootstrapping approach to acquire a lexicon and a dictionary of extraction patterns for a semantic category. The user provides a set of seed words for the lexicon and the rest of the process is completely unsupervised. There is an inner loop of joint acquisition of lexical entries and extraction patterns and an outer loop which keeps only the highest scored learned entries to prevent the degradation of the performance produced when a wrong word is introduced in the learned set. AutoSlog is used to obtain all candidate patterns, and its heuristic-driven approach is hence conserved. A sketch of the process is shown in Algorithm 6.

Snowball Agichtein and Gravano [2000]'s work is based on DIPRE [Brin, 1998], a system to extract relations from web pages. Snowball extracts simple binary relations such as `<company>` - `<location>` from free text, given a set of starting seed pairs. The patterns for extracting such relations include five fields: the named entity classes for the two entities involved in the relation, and the left, middle and right contexts of the pattern. Contexts are represented using a vector-space model.

The learning process starts finding occurrences of the known pairs in the text. A specific pattern for each occurrence is found and then the patterns are clustered using a single-pass clustering

Algorithm 6 [Riloff and Jones, 1999]

1: Initialize the lexicon $\mathcal{L}$ to the set of seeds
2: **repeat**
3: Generate all candidate extraction patterns π from the corpus using AutoSlog heuristics
4: Apply the candidate extraction patterns to the corpus to obtain a set of candidate lexical entries
5: Initialize the learned pattern set $\Pi_L = \emptyset$
6: Initialize the temporary lexicon $\mathcal{L}' = \mathcal{L}$
7: **repeat**
8: Sort candidate patterns according to **RlogF** score:

$$rlogf(\pi) = \frac{F(\pi)}{N(\pi)} \cdot \log_2 F(\pi)$$

where $N(\pi)$ is the number of unique entries extracted by π, and $F(\pi)$ is the number of unique entries already in the temporary lexicon $\mathcal{L}'$ extracted by π.
9: Add best scored pattern $\hat{\pi}$ to Π_L
10: Add the entities extracted by $\hat{\pi}$ to $\mathcal{L}'$
11: **until** (maximum iteration number reached $\vee$ $rlogf(\hat{\pi}) < 0.7$) $\wedge \neg(rlogf(\hat{\pi}) > 1.8)$
12: Score learned entries ω in the temporary lexicon $\mathcal{L}'$

$$score(\omega) = \sum_{\pi | extracts(\pi,\omega)} 1 + 0.01 \cdot rlogf(\pi)$$

13: Add N best scored learned entries in $\mathcal{L}'$ to the learned lexicon $\mathcal{L}$
14: **until** maximum iteration number reached
15: Return the learned lexicon $\mathcal{L}$

algorithm. The similarity between patterns is based on the cosine in vector-space representation. The centroid of each cluster is used as a generalized pattern, and then these patterns are used to extract new pairs of related entities. The process is repeated iteratively.

A drawback of the system is that the quality functions used require one of the elements in the pair to be a key for the relation, this is, it does not support N-to-M relations.

ExDISCO Yangarber et al. [2000] propose ExDISCO, a bootstrapping approach to learn patterns in the form of subject-verb-object tuples. The initial set of seed patterns defines a partition of the learning corpus into relevant and irrelevant documents. The candidate patterns are ranked according to how their distribution correlates with the relevance of the documents, and the highest ranked pattern is selected. The addition of this pattern induces a new split of the corpus, which in turn gives a new ranking of the candidate patterns, and so on. In addition, the system includes a procedure to induce semantic classes for the slots of the patterns in parallel with the learning of the patterns.

[**Yangarber, 2003**] The problem of stopping bootstrapping is usually solved using manual thresholds. The process stops when the quality of the learned elements decreases below a value, or the number of iterations exceeds a maximum. However, the quality of the final solutions can depend on these threshold values. Yangarber [2003] proposes an extension to [Yangarber et al., 2000] to learn patters for several scenarios in parallel and uses the competition between the different learners as an indication to stop the learning, in a framework which is named *Counter-training*.

The user must provide a set of seed patterns for each one of the scenarios, and the learning process works in the same way for each one of them as in ExDISCO. One learner is started for each scenario. When patterns are scored by each individual learner, the progress of the other learners is taken into account: a pattern that occurs in documents that have been found relevant

by other learners is penalized. When a learner cannot incorporate any new pattern because all the candidates have been rejected, its learning stops. The overall learning process continues with the other learners until there is only one left.

[**Stevenson and Greenwood, 2005**] Stevenson and Greenwood [2005] propose a method which integrates WordNet into a vector-space model. Patterns consist of triples representing the subject, the verb and the object of a clause. A set of seed patterns is required to the user, and the process detailed in Algorithm 7 is carried.

Algorithm 7 [Stevenson and Greenwood, 2005]

1: Process the corpus to find the set Π_C of all candidate patterns π
2: Initialize the set of accepted patterns Π_A to the given seeds
3: **repeat**
4: Sort patterns in Π_C according to their similarity to those in Π_A
5: Move the highest ranked patterns from Π_C to Π_A
6: **until** Learning is complete
7: Return the accepted patterns Π_A

To compute the similarity of two patterns, they are viewed as vectors of binary `<position>` : `<word>` features, and the measure of Jiang and Conrath [1997] is used to determine the semantic similarity between different words in the same position (subject, verb or object).

[**Surdeanu et al., 2006**] Surdeanu et al. [2006] apply co-training [Blum and Mitchell, 1998] to simultaneously learn two text classifiers:

- one using Expectation-Maximization and Naive Bayes with the words in the documents as features (as in Nigam et al. [2000]),
- one using a decision list with the patterns in the documents as features.

Co-training starts with a set of seed documents, labeled as belonging to several categories. Iteratively, one classifier learns from the already labeled documents and makes predictions about the unlabeled ones. These newly labeled documents are fed to the other classifier, which learns from them and makes new predictions which are fed back to the first one, and so on. The patterns selected by the decision list classifier are used as IE extraction patterns, as they are relevant and specific to each domain.

The kind of patterns learned are tuples of the form subject-verb-object, subject-verb, verb-object... Several criteria to score the patterns are proposed and evaluated.

Drawbacks of Bootstrapping Even if bootstrapping approaches are very appealing due to its reduction in handcrafting, they present some problems. The main disadvantage of bootstrapping approaches is that, although the initial set of seed examples can be very reliable for the task in hand, the accuracy of the learned patterns quickly decreases if any wrong patterns are accepted in a single round. Systems based on bootstrapping techniques must hence incorporate statistical or confidence measures for patterns in order to limit this problem (such as those of Agichtein and Gravano [2000] and Yangarber [2003]). Another drawback of the bootstrapping techniques is that they need a large corpus (on the order of several thousand texts), which is not feasible in some domains. Finally, the bootstrapping approach is also dependent on the set of seed examples that are provided by the user. A bad set of seed examples can lead to a poor set of extraction patterns.

2.4 Evaluation of Information Extraction

2.4.1 MUC Conferences

The first formulae proposed for the evaluation of IE tasks, and still ones of the most widely used, were those used in MUC-3. Given the response templates filled by the system and the set of key templates constructed by the judges, each slot in them is classified as:

- Correct (COR), if the response and the key are deemed to be equivalent,
- Partial (PAR), if the response and the key are judged to be a near match,
- Incorrect (INC), if the key and response do not match,
- Spurious (SPU), if the response has a fill which has no corresponding fill in the key,
- Missing (MIS), if the key has a fill and the response has no corresponding fill.

Two measures were adopted from IR, recall (R) and precision (P). They measure the completeness and accuracy of the system, respectively, according to the counts of each type of slot:

$$R = \frac{COR + (0.5 \cdot PAR)}{COR + PAR + INC + MISS} \qquad P = \frac{COR + (0.5 \cdot PAR)}{COR + PAR + INC + SPUR}$$

The F-measure introduced in MUC-4 was intended for global comparison between systems. In fact, a family of measures F_β was defined, where the parameter β expresses how much preponderance is given to precision over recall:

$$F_\beta = \frac{(\beta^2 + 1.0) \cdot P \cdot R}{\beta^2 \cdot P + R}$$

The most used F-measure was F_1, which gives the same relevance to both factors. It is equal to the harmonic mean of precision and recall. A detailed description of the metrics used for evaluation in MUC-4 can be found in Chinchor [1992].

From MUC-6, no partial credit was given any longer for partially extracted slots. Partially extracted slots were from then on considered incorrect.

2.4.2 ACE Conferences

The evaluation of the ACE conferences follows the same model as that of MUC: there is a key reference constructed by the judges, the solution proposed by the systems is compared to the key, and the score depends on how similar both solutions are. However, the ACE evaluation has particular features, with respect to MUC:

- There is a distinction between entities, relations and events (elements) and their mentions. The score for each element includes the detection of both the element and its mentions.
- Different attributes, as well as a position in a hierarchy, are defined for each kind of element and mention. The score depends not only on the detection of the textual span of the elements, but also on the right prediction of the values of their attributes. Some attributes are judged more important than others, and thus represent a larger part in the total score.
- The correct identification of all mentions that corefer to the same element is also evaluated.
- The mapping between the elements extracted by the system and those in the key which gives the highest score is the one that is selected for evaluation.

The detailed formulae for the evaluation scores used in ACE are complex, but can be found in [ACE05]

2.4.3 Evaluation of Extraction Patterns

The evaluation of extraction patterns for IE obtained from ML approaches has been carried out by researchers in several ways. However, two main types of evaluation can be distinguished: direct and indirect.

2.4.3.1 Direct or Intrinsic Evaluation

In this case the patterns are directly evaluated using metrics related to the quality of the patterns *per se*.

A usual form is the evaluation of the usability of the patterns. The patterns are used in a complete IE system and then the performance obtained by the system on a given task over a test corpus is measured. In general, this is the approach used by supervised learning approaches. In lightly supervised approaches, the patterns must be extended manually after the learning process by including the slot-fillers to be evaluated in this manner, as in Català et al. [2003], Yangarber et al. [2000].

It is also possible to measure the relevance of the patterns to their domain or scenario of extraction. Riloff [1996], Surdeanu et al. [2006] evaluate the learned patterns with this method. A human judge reviews all patterns and for each one of them she decides whether it is relevant or not for its domain. However, this is a costly process, and it becomes impractical as the size of the learning corpora and the number of learned patterns increases. In addition, the task of deciding whether a pattern is relevant or not for a given domain is not trivial, mainly due to the ambiguity of the patterns. Thus, the process must be carried by more than one judge, so that the relevance of the ambiguous patterns can be agreed upon.

In systems where the system outputs a list of members of a semantic class or a list of elements accomplishing a relation, measuring the quality of these lists is another form of direct evaluation of the patterns. The IR metrics of precision and recall are generally used in this case. An example of this approach is Riloff and Jones [1999]. However, this form of evaluation requires an exhaustive list of the elements in the class or relation present in the corpus, which is not feasible for large corpora. Intermediate solutions must be adopted, sometimes resorting to extern hand-crafted resources like gazetteers, as in Agichtein and Gravano [2000].

2.4.3.2 Indirect or Extrinsic Evaluation

In extrinsic evaluations the patterns are incorporated in a system for a task different to IE, and their quality is measured by the performance of the system, using the metrics proper of the task.

A common approach is to use the detected patterns for text filtering, and evaluate the classification of the documents in a collection which is induced by the patterns. Standard classification and clustering evaluation metrics like precision/recall or purity/inverse purity can be used. Yangarber et al. [2000], Yangarber [2003], Surdeanu et al. [2006] evaluate their patterns by filtering. Stevenson and Greenwood [2005] extend this document filtering approach to sentence filtering, using a version of the MUC-6 corpus which had been annotated with events at sentence level by Soderland [1999].

Lastly, as mentioned, other Text Mining tasks such as QA and Summarization provide a framework in which IE technology and knowledge can be incorporated. Extrinsic evaluations measure the performance of these Text Mining systems enhanced with IE knowledge and use it as an indication of the quality of the incorporated knowledge. Standard measures such as Mean Reciprocal Rang (MRR) for QA or ROUGE [Lin, 2004] for Summarization can be used.

Chapter 3

Our Approach

In the previous chapter, we have reviewed several approaches of ML for IE. State-of-the-art lightly supervised approaches depending on seeds give interesting results, and the cost of manually providing the seeds is significantly lower than those involved in supervised approaches or other lightly supervised approaches which depend on a complete classification of all the documents in the collection. However, as stated in the introduction, seed-based approaches have drawbacks:

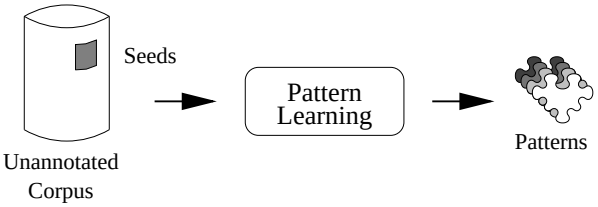
- The supervision requires the user browsing and exploring through an unannotated corpus. As the size of corpora grows, manual exploration becomes harder, and it will eventually become unfeasible by a human user.
- By giving this kind of supervision the user is introducing a strong bias into the learning process, as described in Section 2.3.2.4.
- The selection of seed documents relevant to different domains requires an a priori definition of the domains, and also requires the availability of collections of documents relevant to each IE task.

We believe that removing this human supervision and guiding the whole process of pattern learning by an automatic analysis of the data will avoid the inconveniences of seed-based approaches. Our goal is to develop a methodology that, from a completely unannotated collection of documents, and without the need of any kind of user-given seed documents or patterns, produces good quality patterns useful for IE tasks. And our proposal to achieve this goal is to enhance the pattern generation process with clustering techniques.

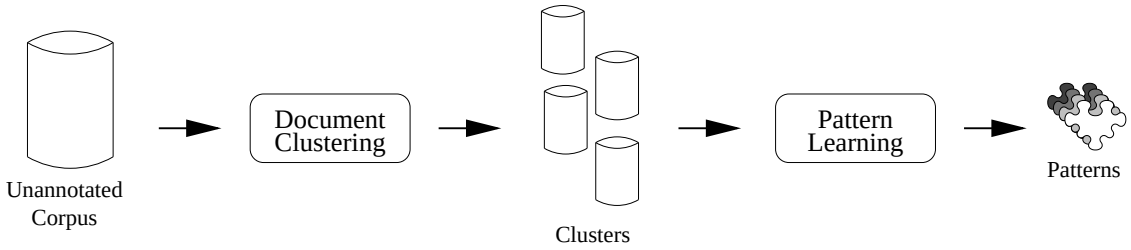
A graphical representation of the differences between existing approaches and the one we propose is depicted in Figure 3.1. Lightly supervised approaches based in bootstrapping require manual selection of a set of seeds (either seed documents or seed patterns) from which to start the pattern learning process (Figure 3.1a). Our proposal is to remove this process of manual seeding with the help of *clustering* techniques.

We conceive at least three different ways in which document clustering and pattern learning can be combined:

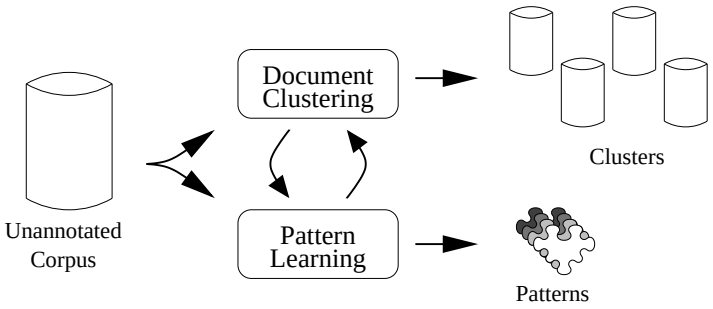
- Figure 3.1b shows clustering and pattern learning as independent processes in a sequential combination, the latter taking as input the output of the former. This is the simplest approach.
- Figure 3.1c shows clustering and pattern learning as independent collaborative processes, each one receiving input from and giving output to the other.
- Figure 3.1c shows clustering and pattern learning as a unique joint process. Clusters and patterns are learned at the same time by a single learner.



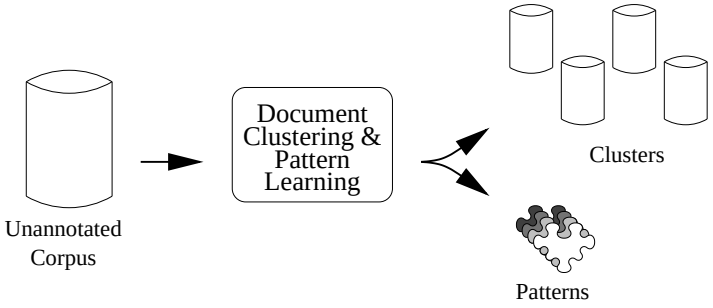
a) Pattern learning with manual seed selection



b) Sequential clustering and pattern learning



c) Collaborative clustering and pattern learning



d) Joint clustering and pattern learning

Figure 3.1: Approaches for IE pattern learning

The feasibility of enhancing pattern learning by clustering depends on the availability of suitable document clustering methodologies. For this reason, in a first step our primary focus of research has been document clustering. Thorough details of the research carried out are given in Section 3.1. Our research focus is now shifting towards the application of this newly developed clustering methodologies to enhance the pattern learning process. The description of the ongoing and future work on combination of clustering and pattern learning is given in Section 3.2. Last Section of the chapter, 3.3, contains some comments about the semantics of the patterns obtained by an approach like the one we propose, and the meaning of *scenario of extraction* in these cases.

3.1 Document Clustering

The task of clustering can be defined as *the process of partitioning a set of patterns into disjoint and homogeneous meaningful groups, called clusters* [Tasoulis and Vrahatis, 2004]. Research on clustering is active in the field of pattern recognition. A survey of clustering methods in general can be found in Jain et al. [1999].

Even if document clustering is mainly an unsupervised task, there are still elements of supervision in many clustering methods. The number of clusters k or a set of seed documents from which to start the clustering process must be provided by the user or by external sources, and the performance of the method can be sensitive to these choices. Unsupervised document clustering can then be defined as the process of grouping similar documents without knowing *a priori* the number of document categories. Given that the aim of this thesis project is to reduce the elements of supervision in the process of pattern learning, our work has focused in this unsupervised setting of the clustering problem.

Classical generic unsupervised clustering methods are often used for unsupervised document clustering. These methods are based on two steps:

Clustering Candidate Generation In the first step, a set of clusterings π_j is generated, each one consisting of a different number k_j of groups of elements or *clusters*. Hierarchical algorithms generate the set of clusterings by building a tree representation of the clusters, or dendrogram, without supervision [Hatzivassiloglou et al., 2000]. Other approaches are based on the use of supervised clustering algorithms, such as iterative refinement [Zhao and Karypis, 2004] or matrix factorization [Xu et al., 2003] among others. These approaches repeatedly apply the supervised clustering algorithm for an increasing number of clusters k_j.

Best Candidate Selection In the second step, the best clustering is selected by means of a criterion function, such as the C score of Calinski and Harabasz [1974] or Minimum Description Length (MDL) of Rissanen [1978].

Other unsupervised methods use a hybrid strategy in which a supervised clustering method is used to improve an initial solution found by means of an unsupervised method [Surdeanu et al., 2005].

However, each one of these methods has an intrinsic and particular bias, uses a certain document representation, and depends on a document similarity measure. All these assumptions guide the clustering process, and lead it to a particular solution that may not be the optimal clustering. To overcome this limitation, recent research has focused on clustering combination. From a general point of view, the problem of clustering combination can be defined as: *Given multiple clusterings of the data set, find a combined clustering with better quality* [Topchy et al., 2005]. The most popular methods in the state of the art are graph partitioning based [Strehl and Ghosh, 2002] and probability based [Topchy et al., 2005].

Probability based clustering combination has already been applied to document collections, as in Topchy et al. [2005], based on Expectation-Maximization (EM) [Dempster et al., 1977], and in Siersdorfer and Sizov [2004], based on voting following a probabilistic model. These combination approaches give the same relevance to each individual clustering. However, different clustering methods may be more or less suitable for different data collections, according to how the collections

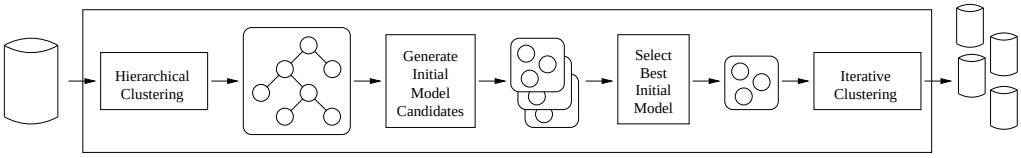


Figure 3.2: The geometric hybrid clustering method

accomplish the assumptions of the method. It is hence sensible to think of a weighted combination of clusterings, in which *better* clusterings contribute more to the final result. This makes necessary to find a strategy to determine the weights of each clustering in the combination.

Our work has included research on individual clustering methods, which is detailed in Section 3.1.1, as well as the development of a generic approach for weighted clustering combination, described in Section 3.1.2. Two unweighted clustering combination methods have been adapted to be used in this framework. The resulting weighted methods are explained in Section 3.1.3. Sections 4.1 and 4.2 in the next chapter contain a walkthrough of the experiments we have carried out on document clustering.

All this work is described in detail in Gonzàlez and Turmo [2005, 2006].

3.1.1 Unsupervised Individual Clustering

As mentioned before, the particular bias of the individual clustering methods, as well as the type of document representation and similarity measure they use, imply a different point of view of the documents. In our research, we have worked with a heterogeneous set of unsupervised individual clustering methods, which will be detailed here.

The first one of them is a geometric hybrid method [Surdeanu et al., 2005], which has been shown to give good performance for unsupervised document clustering of different real-world collections. The second one is an information-theoretical hybrid method, which is a version of the previous one with a different bias, a different document representation and a different similarity measure. The third one is a classical method consisting of a hierarchical algorithm and a criterion function to determine the best clustering. A description of each one of them follows.

3.1.1.1 Geometric Hybrid Method

An outline of the method presented in Surdeanu et al. [2005] is shown in Figure 3.2, and is described below:

The process starts finding a good initial clustering for an iterative refinement algorithm. This kind of algorithm requires the number of clusters to be provided, and is sensitive to this choice. This is why a good estimation of the number of clusters is mandatory for a good initial clustering, even if some documents remain uncovered. The initial clustering is found by applying a classical method:

1. A hierarchical algorithm is used to find a dendrogram.

2. A set of initial clustering candidates is generated for different degrees of document coverage and different cluster quality measures. Each one of the candidates consists of the list of the best clusters from the whole set present in the dendrogram, ranked taking into account a specific degree of document coverage and a specific cluster quality measure[1]. This implies that some documents may not occur in any cluster of the list.

3. The best clustering candidate is selected by applying a global quality measure.

[1] For simplicity, the set of cluster quality measures has been elided here. For more details, see Surdeanu et al. [2005]

In Surdeanu et al. [2005], the method is specified using a geometric point of view:

- Documents are represented as $tf \cdot idf$ vectors of words [Spärck-Jones, 1972].
- The distance between two documents is computed by the cosine distance.
- The hierarchical algorithm used is Hierarchical Agglomerative Clustering (HAC) [Murty and Krishna, 1980].
- The global quality function is computed by Calinski and Harabasz's C score.
- The iterative refinement algorithm applied is EM, with parameters estimated using Naive Bayes [Nigam et al., 2000].

We will refer henceforth to this method as **Geo**.

3.1.1.2 Information-Theoretical Hybrid Method

The field of Information Theory goes back to the seminal work of Shannon [1948]. Recently, there has been an interest in applying information theoretical measures to the task of document clustering [Dhillon and Guan, 2003, Slonim, 2003]. For this reason, and to find a view of the data different from **Geo**, we have adapted the aforementioned hybrid method to use information-theoretical measures, as follows:

- Documents are represented as probability distributions of words.
- The distance between two documents is computed by Jensen-Shannon divergence [Lin, 1991]. There are other measures coming from information theory that could be useful to define a document distance, such as Kullback-Leibler divergence [Kullback and Leibler, 1951] or mutual information. However, on the contrary of Jensen-Shannon divergence, they are not symmetric or require absolute continuity of one distribution with respect to the other.
- The hierarchical algorithm used is Agglomerative Information Bottleneck method (aIB) [Slonim and Tishby, 1999].
- The global quality function is computed as the length of the message encoding the documents in the clustering candidate, as described in Gonzàlez and Turmo [2006]. Classical information-theoretical selection criteria, such as MDL or Minimum Message Length (MML) [Boulton and Wallace, 1969], require a probability distribution, which cannot be inferred from the dendrogram.
- The iterative refinement algorithm applied is divisive information-theoretical clustering (DITC) [Dhillon and Guan, 2003].

We will refer to this method as **IT**.

3.1.1.3 Hierarchical Method

The third clustering method is a classical method based on a hierarchical algorithm. A dendrogram is built using the aIB algorithm, and the first local maximum of the Calinski and Harabasz's C score as the number of cluster increases is found, as it is the level of the dendrogram at which the best clustering is expected to occur.

We will refer to this method as **HiIT**.

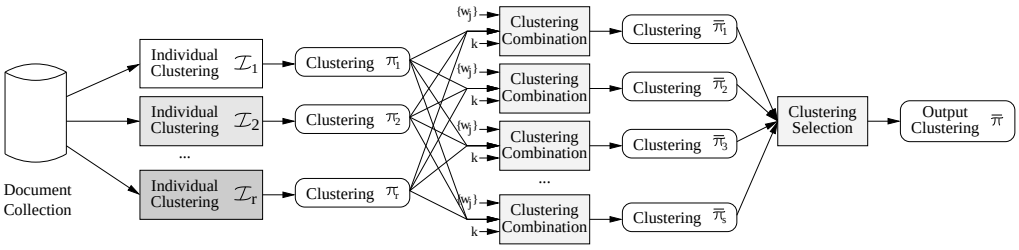


Figure 3.3: The weighted clustering approach

3.1.2 Generic Approach for Clustering Combination

The approach we propose for weighted clustering combination is depicted in Figure 3.3, and proceeds in three steps:

1. Generate the initial clusterings, $\{\pi_j\}$, each one with number of clusters k_j, applying different individual unsupervised clustering methods, $\{\mathcal{I}_j\}$, to the input document collection.

2. Generate weighted combination clusterings, $\bar{\pi}_\alpha$, from the initial ones, $\{\pi_j\}$, using the clustering combination method with different sets of weights, $\{w_j\}_\alpha$, and numbers of clusters, $\bar{k}_\alpha$. Following the usual cluster ensemble problem statement [Strehl and Ghosh, 2002], the combination method does not access the document representation used by the individual clustering methods.

3. Select the best weighted combination clustering $\bar{\pi}$ from those generated in step 2.

This approach defines a family of weighted combination schemes.

3.1.3 Weighted Combination

We have adapted two non-weighted clustering combination methods to deal with a weighting of the initial clusterings. The first one is based on solving graph partition problems, whereas the second one is based on EM.

To describe these methods, we will use a formalization of clustering combination. Having $\mathcal{D} = \{d_1 \dots d_n\}$ a set of documents, a clustering π_j of this set is a partition of $\mathcal{D}$ into a set $\{C_j^1 \dots C_j^{k_j}\}$ of k_j disjoint clusters C_j^i. Each cluster C_j^i can be identified by its numerical label i. The clustering π_j can also be viewed as a function mapping documents to labels:

$$\pi_j : \mathcal{D} \to \{1 \cdots k_j\} \tag{3.1}$$

The aim of clustering combination is to find a clustering $\bar{\pi}$, of $\bar{k}$ clusters, which is the consensus of r clusterings $\{\pi_j\} = \{\pi_1 \dots \pi_r\}$, by means of a consensus function Γ. If we consider a weighted combination of clusterings, for each initial clustering π_j a weight w_j is defined.

Through all this discussion, $\|\mathcal{S}\|$ refers to the cardinality of set $\mathcal{S}$.

3.1.3.1 Graph based Method

Strehl and Ghosh [2002] propose several methods to combine a cluster ensemble and produce a single output clustering, based on solving graph partition problems. In addition, it offers a criterion to select which method to use in each case, using a normalized mutual information measure.

Given initial clusterings $\{\pi_j\}$, and the number of desired clusters $\bar{k}$, three graph partition problems are solved to obtain three possible combinations $\bar{\pi}$:

CSPA (Cluster based Similarity Partitioning Algorithm) A graph is built in which every document $d_a \in \mathcal{D}$ is a vertex. The weight of the edge between d_a and d_b is the number of clusterings π_j in which the documents lie in the same cluster, $\|\pi_j \mid \pi_j(d_a) = \pi_j(d_b)\|$. A partition of this graph into $\bar{k}$ clusters is found, and the combination is the induced clustering on $\mathcal{D}$.

HGPA (HyperGraph Partitioning Algorithm) A hypergraph is built in which every document $d_a \in \mathcal{D}$ is a vertex. Each cluster C_j^i in each initial clustering π_j is a hyperedge, and the hypergraph is partitioned into $\bar{k}$ clusters. The combination is the induced clustering on $\mathcal{D}$.

MCLA (Meta-CLustering Algorithm) A graph is built in which every cluster C_j^i in each initial clustering π_j is a vertex. The weight of an edge between clusters C_j^i and $C_{j'}^{i'}$ is the Jaccard measure of the two sets: $\|C_j^i \cap C_{j'}^{i'}\| / \|C_j^i \cup C_{j'}^{i'}\|$. This graph is partitioned into $\bar{k}$ so-called meta-clusters γ_q. In the combination, each document d_l is assigned to the meta-cluster to which it contributes the most, this is $\arg\max_{\gamma_q} \|C_j^i \in \gamma_q \mid d_l \in C_j^i\|$.

To decide which of the three combinations $\bar{\pi}$ is the best, the measure of normalized mutual information (NMI) between two clusterings is defined as:

$$NMI(\pi_j, \pi_{j'}) = \frac{I(\pi_j, \pi_{j'})}{\sqrt{H(\pi_j) \cdot H(\pi_{j'})}} \tag{3.2}$$

where I and H are the usual mutual information and entropy, respectively. For each one of the three obtained combinations $\bar{\pi}$, the average normalized mutual information (ANMI) with respect to the initial clusterings $\{\pi_j\}$ is:

$$ANMI(\bar{\pi}, \{\pi_j\}) = \frac{\sum NMI(\bar{\pi}, \pi_j)}{\|\{\pi_j\}\|} \tag{3.3}$$

The obtained $\bar{\pi}$ with the highest ANMI is selected as the best combination.

Weighted Version To incorporate weighting into this method, the graphs produced by the three methods have been modified:

CSPA The weight of the edge between documents d_a and d_b is the sum of the weights of the clusterings π_j in which the documents lie in the same cluster, $\sum w_j \mid \pi_j(d_a) = \pi_j(d_b)$.

HGPA The weight of the hyperedge representing cluster C_j^i is the weight w_j of the clustering π_j to which the cluster belongs.

MCLA The weight of the edge between clusters C_j^i and $C_{j'}^{i'}$ is the Jaccard measure of the two sets, multiplied by the weights of the clusterings π_j and π_p to which the clusters belong: $\|C_j^i \cap C_{j'}^{i'}\| / \|C_j^i \cup C_{j'}^{i'}\| \cdot w_j \cdot w_{j'}$

The same ANMI function is used to select the best model.

In all cases, the non-weighted version is equivalent to assign a weight of $w_j = 1$ to all clusterings π_j.

As in Strehl and Ghosh [2002], we have used the freely available[2] packages METIS and HMETIS of Karypis and Kumar [1998], Karypis et al. [1997] to solve the graph and hypergraph partition problems.

[2] `http://glaros.dtc.umn.edu/gkhome/views/metis/`

3.1.3.2 Probability based Method

Topchy et al. [2005] introduce a probabilistic view of combination, which is solved using EM.

Given initial clusterings $\{\pi_j\}$, and the number of desired clusters $\bar{k}$, a matrix Y can be defined with as many rows as documents in the clusterings, and as many columns as initial clusterings. Each entry y_{lj} stands for the label of the cluster to which document d_l belongs in clustering π_j, as computed with Equation 3.1. These labels are seen as random variables drawn from a probability distribution described as a mixture of $\bar{k}$ multi-variate component densities. A document d_l can be represented by its labels $y_l = (y_{l1} \dots y_{lr})$. If the naive Bayes assumption is taken, with respect to the independence of the labels given the class, and each label y_{lj} is considered to be drawn from a multinomial distribution, the probability of y_l is:

$$P(y_l \mid \Theta) = \sum_{m=1}^{\bar{k}} \alpha_m P(y_l \mid \Theta_m) \tag{3.4}$$

$$P(y_l \mid \Theta_m) = \prod_{j=1}^{r} P(y_{lj} \mid \Theta_{mj}) \tag{3.5}$$

$$P(y_{lj} \mid \Theta_{mj}) = \prod_{k=1}^{k_j} \vartheta_{mjk}{}^{\delta(y_{lj},k)} \tag{3.6}$$

where α_m is the probability of each mixture *a priori*, ϑ_{mjk} is the probability of feature j in mixture m taking value k, and δ stands for the function evaluating to 1 if its arguments are equal and to 0 otherwise.

The model parameters, Θ, are estimated using EM and the combination clustering $\bar{\pi}$ is obtained by assigning each document to the most probable mixture component:

$$\bar{\pi}(d_l) = \arg\max_m P(y_l \mid \Theta_m) \tag{3.7}$$

Weighted Version The most natural weighted extension to the previous approach is to use weighted naive Bayes models [Ferreira et al., 2001]. Equations 3.4 and 3.6 remain the same, and weights are introduced into Equation 3.5, as follows:

$$P(y_l \mid \Theta_m) = \prod_{j=1}^{r} P(y_{lj} \mid \theta_{mj})^{w'_j} \tag{3.8}$$

These weights w'_j must be normalized so they add up to the number of initial clusterings r. EM is also used to train this model.

3.1.3.3 Clustering Selection

The set of weighted combination clusterings $\{\bar{\pi}_\alpha\}$ is virtually infinite. Selecting the best combination can be seen as a search problem.

To select the best combination from all the possible weighted combination clusterings, $\bar{\pi}_\alpha$, a scoring function η to be maximized is used:

- For graph based combination, η is the average normalized mutual information, as defined in Equation 3.3.
- For EM based combination, η is the log-likelihood of the clustering given the probabilistic model found, $LL(\bar{\pi}) = \sum_{d_l} logP(y_l \mid \Theta)$, where P is defined in Equation 3.4.

However, a strategy must be defined to explore the possible values of $\bar{k}$ and w_j. In our first experiments, we have used a constrained global search approach. All combinations with $\bar{k}$ and w_j satisfying:

$$\begin{aligned} \bar{k} &\in \{\max(2, \lfloor \min k_j - \sigma_k \rfloor) \ldots \min(\|\mathcal{D}\|, \lceil \max k_j + \sigma_k \rceil)\} \\ w_j &\in \{1 \ldots G\} \end{aligned}$$

are found, where k_j is the number of clusters in initial clustering π_j, σ_k is the standard deviation of k_j, and G is a parameter. The η function is found for each one of the combinations, and the one with the maximum value is selected.

The number $\bar{k}$ of clusters that a combination clustering can contain is thus limited to values in the interval defined by the minimum and maximum number of clusters found by the individual clustering methods, extended by the standard deviation σ_k in both directions. This number must be greater or equal than 2, and lower or equal than the number of documents in the clusterings, $\|\mathcal{D}\|$. If there is good agreement between all individual methods, σ_k will be small and the best $\bar{k}$ will be close to the individual k_j. On the contrary, if the individual k_j differ considerably, σ_k will be large, and the best $\bar{k}$ can be outside the interval of k_j. Parameter G allows to prune the combinations by limiting the weights to natural values ranging from 1 to G. The larger the value of G, the finer the tuning of the weights.

The cost of this simple approach, however, grows exponentially as the number of clusterings in the combination increases. In addition, the search space may be too constricted by the fact of w_j being integer values in a range, and good solutions can be missed. For these reasons, we are investigating on the application of other search approaches, such as hill-climbing or local beam search.

3.2 Enhancing Pattern Learning with Document Clustering

Our starting work on pattern learning is based on that of Surdeanu et al. [2006]. As explained in Chapter 2, this is a co-training bootstrapping approach which simultaneously learns a Naive Bayes word-based text classifier through EM and a set of patterns for a decision list classifier.

The first approach we have proposed for the combination of pattern learning and document clustering is a sequential setup (Figure 3.1b). The process of manual selection of the required set of seed documents has been substituted by a document clustering process. The differences in performance when feeding the pattern learning loop automatically obtained seeds have been evaluated, and compared to the reference value given by manual seeding. The following automatically found sets of seeds have been used:

Geo, IT, HiIT The complete clustering solutions given by the three individual clustering methods in Section 3.1.1

Combi The complete clustering solution given by the weighted graph-based combination method in Section 3.1.3.1.

Geo.Seeds, IT.Seeds The set of seed documents selected by the **Geo** and **IT** methods, before applying the iterative clustering algorithm (thicker box in Figure 3.4). These are not complete clustering solutions, and there will hence be documents not covered by any cluster.

More details about this first evaluation are given in Section 4.3. We will take this sequential approach as a baseline for oncoming work.

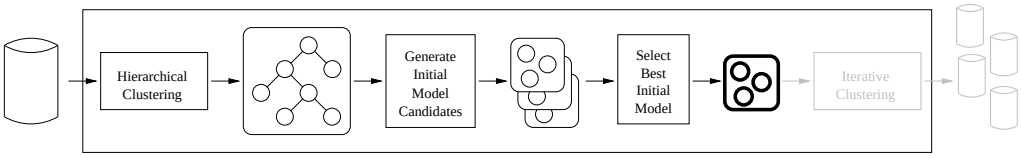


Figure 3.4: The **Geo.Seeds** and **IT.Seeds** methods

3.3 Semantics of the Obtained Patterns

In lightly supervised approaches it is the user's task to annotate *a posteriori* the slot-fillers present in the patterns learned by the system to afterwards use these patterns in a complete IE system. However, in manually seeded approaches there is still a notion of scenario of extraction, as the seed patterns represent textual instances of the relevant concepts and the learned patterns are hence also expected to deal with relevant concepts.

Nevertheless, when human supervision is removed and replaced by unsupervised clustering, and pattern learning becomes an exploratory process of an unannotated document collection, the concept of scenario seems not to be applicable. There is no implicit certainty about the semantics of each pattern being related to any particular scenario of extraction. We can say that the patterns found in this case are **open domain**, since they are not restricted to any particular semantic domain.

In this context, the similarity measure and the document representation by which the clustering process is guided are the key to defining the semantics. Research on the Text Classification area has shown that word-based vector space and bag-of-words models are able to capture the semantic similarity of documents in terms of topic. So clustering using a word-based document representation will tend to group documents by domain, and the learned patterns will be relevant to different domains. However, other document representations yield different clusterings, and thus different learned patterns. The process of clustering hence determines the output of the pattern learning. In some way, the chosen document representation and clustering algorithms convey the *a priori* semantics of the extracted patterns.

It can be argued that *crisp* clustering, in which every document belongs to a single cluster, is too restrictive for real-world uses. Domains from real-world collections will overlap, and events typical of one domain will happen in texts that would be judged as from another one (for instance, a management succession event may happen in a sports piece of news if it refers to the succession of the president of a football club). *Fuzzy* clustering techniques are the alternative to *crisp* clustering. We think frameworks which combine several clusterings in a fuzzy way, such as that of Dimitriadou [2003], are suitable for pattern learning, as they give a way to combine a number of clusterings using not only different clustering algorithms but also different document representations, and give a less biased view of the similarity between documents. We expect our work to eventually explore this line of research.

Chapter 4

Experiments and Results

We have given a small review of the evolution of IE pattern evaluation in Section 2.4.3. Mainly, we have appointed two different ways of evaluation:

- Intrinsic evaluation, in which the output of methods is evaluated *per se*, using measures related to the target problem,
- Extrinsic evaluation, in which the output of methods is used incorporated into a system for a task different to IE. The performance of the methods is then evaluated by how these systems perform in this task.

We present here the results of our experiments in document clustering (Sections 4.1 and 4.2), and of the evaluation of the baseline approach for the combination of pattern learning and document clustering (Section 4.3). The evaluation of clustering is intrinsic, whereas that of patterns is extrinsic and accomplished through the Text Classification task. Section 4.4 gives details of our work on Text Mining systems, as a step towards future extrinsic evaluations of IE patterns on QA and Summarization.

4.1 Clustering Spontaneous Speech Transcripts

The first experiments carried aimed at checking the suitability of unsupervised document clustering methods for clustering spontaneous speech transcripts. Gonzàlez and Turmo [2005] contains all the details for this research. A brief overview will be given here.

The geometric hybrid method **Geo** in Section 3.1.1.1 was tested on Switchboard [Godfrey et al., 1992], a collection of 2438 spontaneous telephone conversations among 543 speakers (302 male, 241 female) for which manual transcripts are available[1]. The participants in every conversation were chosen by a computer operator, who also introduced a topic to talk about from a set of 67. For our purposes, we used these topics as categories.

4.1.1 Evaluation Data

We used two different collections extracted from Switchboard Corpus:

SWB , consisting of all the documents in the corpus, a total of 4876, one for each side of the conversations.

SWB-22 , consisting of only those documents belonging to those categories from 100 documents up. This makes a collection of 22 categories and 2682 documents, coming from 1341 conversations. The aim of this second collection is to allow comparisons of the behavior of the

[1] `http://www.isip.msstate.edu/projects/switchboard/index.html`

different methods when working on data sets with less categories and a more similar number of documents in all of them: in SWB the number of documents in each category ranges from 156 to 8, whereas in SWB-22 the range is from 156 to 100.

4.1.2 Evaluation Metrics

In order to measure the quality of the clustering solutions resulting from different approaches, we compare the obtained clusters $\{C^1 \dots C^k\}$ to the set of categories $\Lambda = \{\lambda^1 \dots \lambda^q\}$ in the data. We then use the following four measures:

Purity (Pur) evaluates the degree to which each cluster contains documents from a single category λ^j. The purity of a cluster C^i is the ratio of the cluster size, $\|C^i\|$, that the largest category of documents assigned to C^i represents [Zhao and Karypis, 2004]. The overall purity is the weighted average of all cluster purities:

$$Pur = \sum_{i=1}^{k} \frac{\|C^i\|}{\|\mathcal{D}\|} \cdot Pur(C^i) = \sum_{i=1}^{k} \frac{\|C^i\|}{\|\mathcal{D}\|} \cdot \frac{1}{\|C^i\|} \cdot \max_j \|C^i \cap \lambda^j\| \tag{4.1}$$

Intuitively, the larger the purity value, the better the clustering solution is.

Inverse purity (IPur) evaluates the degree to which the documents in a category are grouped in a single cluster. The inverse purity of a category λ^j is the ratio of the category size, $\|\lambda^j\|$, that the cluster with the largest number of documents in that category represents. The overall inverse purity is the weighted average of all category inverse purities:

$$IPur = \sum_{j=1}^{q} \frac{\|\lambda^j\|}{\|\mathcal{D}\|} \cdot IP(\lambda^j) = \sum_{j=1}^{q} \frac{\|\lambda^j\|}{\|\mathcal{D}\|} \cdot \frac{1}{\|\lambda^j\|} \cdot \max_i \|C^i \cap \lambda^j\| \tag{4.2}$$

F1 is the harmonic mean of purity and inverse purity.

$$F_1 = \frac{2 \cdot Pur \cdot IPur}{Pur + IPur} \tag{4.3}$$

Estimated number of clusters (k) . The closer k is to the number of categories in the collection q, the better the initialization algorithm is.

4.1.3 Evaluation Procedure

To obtain statistically significant measures, we used a layout similar to a stratified 10-fold cross validation: we randomly split every collection in 10 folds, keeping in each fold the same distribution of categories as in the whole collection. We then use every group of 9 of these folds. The fact that the task of document clustering does not use separate training and test sets accounts for the fact that we do not use the spare fold.

In each experiment, **Geo** is compared to its two constituent algorithms: **HAC** and randomly initialized EM, average across five runs (**EM5**).

4.1.4 Results

Two series of experiments were carried[2]:

[2]The results shown here differ slightly from those in Gonzàlez and Turmo [2005]. The experiments have been repeated to use the presented set of evaluation metrics, and the randomness in the partition for cross-validation has caused this minor differences.

Collection	Method	Pur	IPur	F_1	k
SWB	**Geo**	**53.93 (3.15)**	**87.44 (0.96)**	**66.65 (2.30)**	**26.90 (1.64)**
	HAC	34.51 (1.16)	92.66 (0.21)	50.28 (1.23)	*id*
	EM5	16.12 (1.02)	87.58 (0.66)	27.04 (1.42)	*id*
SBW-22	**Geo**	**80.84 (4.34)**	**92.08 (0.70)**	**86.03 (2.47)**	**18.00 (1.00)**
	HAC	52.25 (3.43)	94.88 (0.59)	67.32 (2.74)	*id*
	EM5	31.74 (2.16)	89.27 (1.83)	46.18 (2.43)	*id*

Table 4.1: Comparison of method performances

Collection	Method	Pur	IPur	F_1	k
SWB	**Geo**	**53.93 (3.15)**	**87.44 (0.96)**	**66.65 (2.30)**	**26.90 (1.64)**
	HAC + C	7.49 (4.25)	99.01 (1.34)	13.63 (6.99)	6.30 (2.83)
	EM5 + C	10.45 (2.46)	91.48 (2.90)	18.64 (3.78)	3.90 (1.13)
SBW-22	**Geo**	**80.84 (4.34)**	**92.08 (0.70)**	**86.03 (2.47)**	**18.00 (1.00)**
	HAC + C	20.79 (8.11)	98.09 (1.20)	33.49 (11.27)	5.50 (1.75)
	EM5 + C	18.53 (2.28)	95.24 (1.56)	30.95 (3.17)	3.60 (0.49)

Table 4.2: Comparison of model dimension detection

4.1.4.1 Method Performance

Table 4.1 shows the result for the first experiments: we apply **Geo** on the collection and compare its performance to that of **HAC** and **EM5** using the k found by **Geo**. For each measure, its mean and its standard deviation are shown.

Comparing the average values among all the folds, we can see that the quality of the solutions proposed by **Geo** clearly outperforms those proposed by its constituents **HAC** and **EM5**, when using the same k found by **Geo**. This is a promising result considering that our approach is unsupervised.

There is a statistically significant difference in terms of purity and F_1 between **Geo** and the other two methods **HAC** and **EM5** in both collections. In terms of inverse purity **HAC** is the best choice, yet the difference in purity makes the F_1 measure be quite lower than that of **Geo**. The performance of **Geo** and **EM5** is similar in inverse purity: there is a significant difference favoring **Geo** in SWB-22 but not in SWB.

Examining the k value found by **Geo**, we can see that in the case of SWB-22 it is quite close to the real q. In SWB there is an underestimation. The fact that SWB has several minor categories with very few documents accounts for this.

4.1.4.2 Model Dimension Detection

Table 4.2 shows the results for the second set of experiments, aimed at comparing the ability to detect a good estimation of the number of clusters. **HAC** and **EM5** are enhanced with the method of Calinski and Harabasz [1974] to allow the detection of k. Here the differences between **Geo** and the other methods are even larger. **HAC** and **EM5** dramatically underestimate k with respect to **Geo**. This produces solutions with high inverse purity but very low purity. Such unbalanced Pur/IPur metrics produce very low F_1 measures, which indicate globally poor clustering solutions.

4.2 Clustering Combination

The next stage of our research was testing the proposed weighted combination approach. This research is contained in Gonzàlez and Turmo [2006].

The performance of each one of the individual clustering methods in Section 3.1.1, and of

Collection	Documents	Categories	Terms
APW	5000	11	27366
LATIMES	5000	8	31960
Reuters	3019	93	7846
Reuters10	2545	10	6734
SMART	5467	4	11950

Table 4.3: Evaluation data sets

weighted and unweighted versions of the combination methods in Section 3.1.3 using these individual methods, is evaluated in corpora coming from several sources.

4.2.1 Evaluation Data

We used five document collections:

APW The Associated Press (year 1999) subset of the AQUAINT collection. The document category assignment is indicated by a CATEGORY tag.

LATIMES The Los Angeles Times subset of the TREC-5 collection. The categories correspond to the newspaper desk that generated the article [Zhao and Karypis, 2004].

Reuters The by now classic Reuters-21578 text categorization collection [Rose et al., 2002]. Similarly to previous work, we use the ModApte split [Nigam et al., 2000], but, since our algorithms are unsupervised, we use the test partition directly.

Reuters10 A subset of the above ModApte test partition that includes only the ten most frequent categories.

SMART A collection previously developed and used for the evaluation of the SMART information retrieval system.

Table 4.3 lists the collection characteristics after preprocessing: number of documents, categories and terms.

4.2.2 Evaluation Metrics

The same metrics as in the previous experiments were used: Purity, Inverse Purity, F_1 and k.

4.2.3 Evaluation Procedure

In this case, a single run was performed on the whole collection, and the measures were calculated on the proposed solutions.

4.2.4 Results

The results for the individual clustering methods can be found in Table 4.4. The **Geo** method is the best in Reuters, Reuters10 and SMART collections, and usually gives the most balanced results ($Pur \approx IPur$). **HiIT** stands out in APW, but often underestimates k. This is why it performs well in terms of inverse purity, yet suffers from low purity. **IT** is the best in LATIMES, because although its F_1 is the same as **HiIT**'s, its estimated k is better.

Table 4.5 shows the results for the combination methods. It presents the results obtained by the graph based method (**Gr**) and the probability based method (**EM**). For both, we give the results obtained so by non-weighted combination (**Eq**) as by weighted combination, estimating the best weights using the procedure in Section 3.1.3.3 (**W**).

Collection	Method	Pur	IPur	F_1	k
APW	Geo	0.74	0.72	0.73	10
	IT	0.72	0.56	0.63	8
	HiIT	**0.63**	**0.88**	**0.74**	**3**
LATIMES	Geo	0.75	0.56	0.64	14
	IT	**0.75**	**0.61**	**0.67**	**7**
	HiIT	0.66	0.68	0.67	6
Reuters	**Geo**	**0.68**	**0.83**	**0.75**	**8**
	IT	0.68	0.76	0.71	7
	HiIT	0.57	0.82	0.67	4
Reuters10	**Geo**	**0.77**	**0.85**	**0.81**	**6**
	IT	0.77	0.76	0.76	6
	HiIT	0.73	0.86	0.79	4
SMART	**Geo**	**0.91**	**0.77**	**0.83**	**6**
	IT	0.89	0.58	0.71	9
	HiIT	0.71	0.97	0.82	3

Table 4.4: Results of the individual methods

The graph based weighted method is the one that achieves the best F_1 in all collections but in APW, where **EM.W** is better. Moreover, in the other four collections the graph based combination methods perform better or equal than the individual clustering methods in all four considered measures.

Regarding the difference between weighted and non-weighted versions of combination, **Gr.W** is better than **Gr.Eq** in all cases. Only in the determination of k in the SMART collection the latter works better than the former ($k = 5$ for **Gr.W**, while $k = 4 = q$ for **Gr.Eq**). Although the differences between **Gr.Eq** and **Gr.W** are small, we believe that the addition of weighting represents an improvement, specially because the weights are found automatically.

On the contrary, the results obtained by the probability based methods **EM** are not so satisfactory. In all but the **APW** collection, the results are lower than using the graph based ones **Gr**, and also lower than the results obtained by some individual methods (e.g., **Geo** and **IT** in the Reuters collection). For the EM methods the results are variable and it cannot be concluded that the weighted combination strategy outperforms its individual components, as occurs with **Gr** methods.

4.3 Enhancing Pattern Learning

The next experiments were aimed at establishing a baseline for the combination of document clustering and pattern learning. The six different automatical sets of seed documents described in Section 3.2 are used as input to the reportedly best method of Surdeanu et al. [2006]: the co-training of a Naive Bayes word-based model using EM and a pattern-based decision list classifier, using Collins criterion to select the best patterns at each step. The performance of the method given the different sets of seeds is compared to that obtained using manually selected seeds.

As in Surdeanu et al. [2006], the evaluation is done indirectly through a text classification task. The decision list classifier containing the set of learned patterns is used to classify the documents in a test partition, and the quality of the produced classification is used as an indication of the quality of the patterns.

4.3.1 Evaluation Data

The APW, LATIMES, Reuters10 and SMART collections are used for this evaluation.

Collection	Method	w_j	Pur	IPur	F_1	k
APW	Gr.Eq	-	0.71	0.73	0.72	7
	Gr.W	2+1+3	0.72	0.72	0.72	7
	EM.Eq	-	0.73	0.64	0.68	11
	EM.W	**1+1+4**	**0.63**	**0.88**	**0.74**	**3**
LATIMES	Gr.Eq	-	0.75	0.68	0.72	7
	Gr.W	**1+3+1**	**0.76**	**0.68**	**0.72**	**8**
	EM.Eq	-	0.78	0.53	0.63	16
	EM.W	1+4+1	0.75	0.61	0.67	7
Reuters	Gr.Eq	-	0.70	0.86	0.77	7
	Gr.W	**3+3+4**	**0.71**	**0.86**	**0.78**	**9**
	EM.Eq	-	0.72	0.71	0.71	10
	EM.W	1+1+4	0.57	0.82	0.67	4
Reuters10	Gr.Eq	-	0.81	0.84	0.83	7
	Gr.W	**4+3+4**	**0.82**	**0.85**	**0.83**	**7**
	EM.Eq	-	0.82	0.82	0.82	8
	EM.W	1+1+4	0.73	0.86	0.79	4
SMART	Gr.Eq	-	0.91	0.91	0.91	4
	Gr.W	**1+3+3**	**0.92**	**0.91**	**0.92**	**5**
	EM.Eq	-	0.91	0.68	0.78	11
	EM.W	1+1+4	0.71	0.97	0.82	3

Table 4.5: Results of the combination methods

4.3.2 Evaluation Metrics

The measures used for evaluation of the text classification task are those of microaveraged precision and recall. Every cluster C^i is mapped to the category λ^j with which it overlaps the most, through a mapping function ϕ. The clusters mapped to category λ^j are $\phi^{-1}(\lambda^j)$.

Then microaveraged precision (P) and recall (R) are defined as:

$$P = \frac{\sum_{j=1}^{q} \|\lambda^j \cap \phi^{-1}(\lambda^j)\|}{\sum_{j=1}^{q} \|\phi^{-1}(\lambda^j)\|} \tag{4.4}$$

$$R = \frac{\sum_{j=1}^{q} \|\lambda^j \cap \phi^{-1}(\lambda^j)\|}{\sum_{j=1}^{q} \|\lambda^j\|} \tag{4.5}$$

$$\tag{4.6}$$

4.3.3 Evaluation Procedure

A 5-fold cross validation setup is used. The collection is split in 5 parts. The classifier using the learned patterns is evaluated on each one of them, after being trained in the other four. The results considered are the average of the 5 folds.

The performance of the patterns is evaluated in an incremental way: first the microaveraged precision and recall obtained with the highest rated 100 patterns are measured. Then the following 100 patterns are added and precision and recall are measured again, and so on. As the number of included patterns grows, recall is expected to increase and precision to decrease.

4.3.4 Results

Figures 4.1 and 4.2 show the precision-recall curves for the four different collections.

We can see how, even if the **Manual** seeds give the best results in all collections but APW, in many cases the performance obtained with automatical seeds is comparable to that obtained with **Manual**. The behavior of **Combi** and **IT** is in the four collections only slightly lower than **Manual**. Only in SMART the curves are more than 5% far from **Manual**, specially in terms of precision. **Geo** and **Geo.Seeds** give good results, in fact better than **Manual**, in APW. They keep up with **Manual** in LATIMES, but in Reuters10 and SMART their performance is lower than **Manual** and other automatical approaches as **Combi** and **IT**. Lastly, it can be seen that the use of **IT.Seeds** or **HiIT** produces a considerable decrease in the performance of the pattern learning process in all collections.

Even if these are only baseline results, the direction appointed seems promising. We can see how, using a simple sequential combination of document clustering and pattern learning, we can learn patterns of use for text classification. These patterns will also be presumably useful for IE. Oncoming research will focus on raising the performance of automatical methods based on combination of pattern learning and document clustering, as well as on the application of the acquired knowledge to IE and eventually to other Text Mining tasks.

4.4 Evaluation through Text Mining

As mentioned in the introduction, there has been a recent interest in the application of IE techniques to Text Mining Tasks such as QA and Summarization.

Within the work of his thesis project, the author takes part in the teams in his research group, TALP Research Center, that are developing QA and Summarization systems. The author has collaborated in the development of:

- A multilingual single-document summarizer for newswire documents [Fuentes et al., 2004],
- A single-document summarizer for spontaneous speech transcripts, within project CHIL [Fuentes et al., 2005c,b],
- A multi-document query-driven summarizer based on QA [Fuentes et al., 2005a],
- A metric to automatically evaluate summary performance [Fuentes et al., 2005a] that is shown to correlate with human metrics, based on the pyramid method of [Nenkova and Passonneau, 2004],
- A multilingual factoid QA system for Spanish [Ferrés et al., 2004a, 2005b] and English [Ferrés et al., 2004b, 2005a].

We refer to the given references for more details on the architecture of the systems and evaluation results.

The availability of QA and Summarization systems will allow research on the application of the IE patterns learned by our approaches on Text Mining tasks. The comparison of the system performances with and without this additional source of knowledge will be compared and conclusions on the quality of the learned patterns drawn.

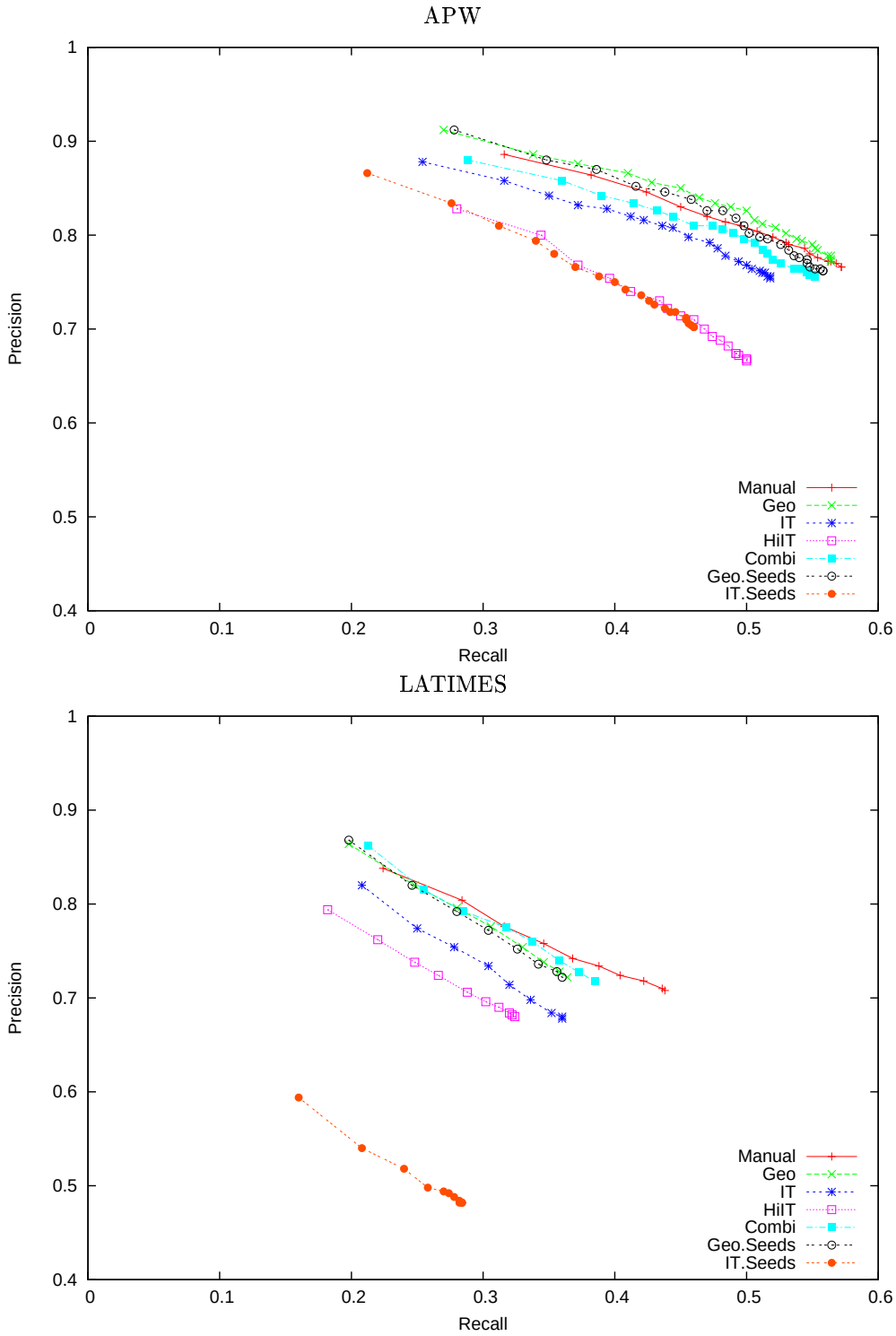


Figure 4.1: Results of pattern learning for APW and LATIMES collections

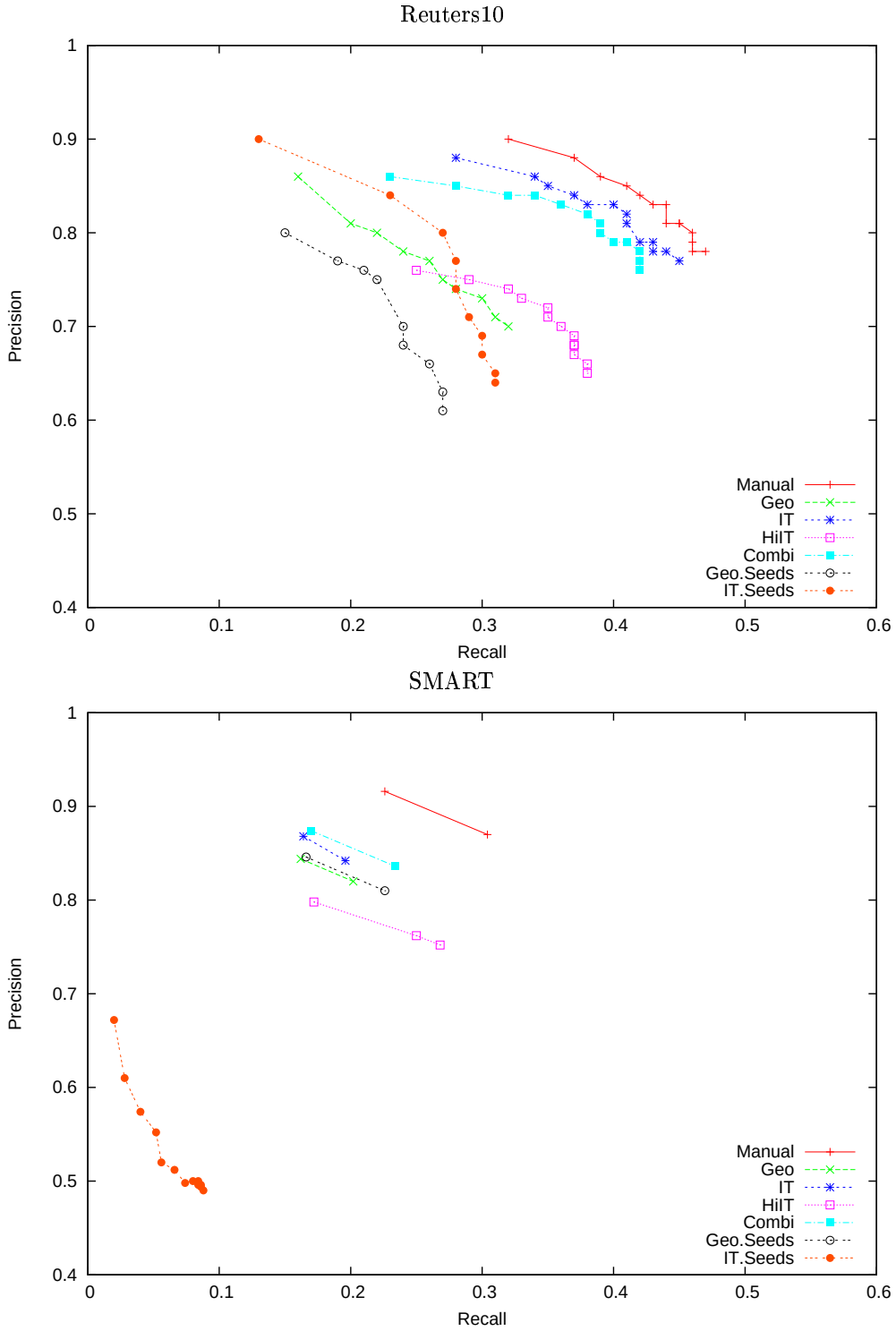


Figure 4.2: Results of pattern learning for Reuters10 and SMART collections

Chapter 5

Work Plan

In this work we have presented an approach for lightly supervised learning of IE patterns which reduces the elements of supervision present in many other approaches by considering document clustering.

In a first step, we have focused in unsupervised document clustering techniques. We have developed a generic approach for unsupervised document clustering by weighted combination. Several individual clustering methods have been considered and tested, and two unweighted combination methods have been adapted to deal with weights in the combination. Individual and combination clustering techniques have been tested on real world document collections.

Next step has been the combination of clustering and pattern learning. The manual selection of seeds has been replaced by automatical selection through a clustering process. This has produced a baseline sequential approach which has been evaluated. Even if the performance is lower than using manually given seeds, the results are promising taking into account that it is a simple sequential approach.

In the meantime, we have collaborated in the development of QA and Automatic Summarization systems. These systems will provide a platform for extrinsic evaluation of the learned patterns.

We now have a baseline for the combination of document clustering and pattern learning. Our research will continue from here to explore the three possible approaches we have sketched in Chapter 3: sequential, collaborative and joint combinations of clustering and pattern learning.

The research lines we will follow will be:

1. In the sequential approach, improve the existing procedure to detect the best weights. As the constrained global search method of Section 3.1.3.3 has many limitations, we will replace it with hill-climbing or local beam search.

2. Add *fuzziness* to clustering. This *fuzziness* will be added in both the individual clustering methods and the combination methods.

   As mentioned, *fuzzy* clustering methods are popular, specially in contexts where there is a combination of knowledge from several sources. Moving from *crisp* to *fuzzy* clustering can be a better approach to reality, as domains often overlap and documents can talk about more than one topic.

   Even though *fuzzy* clustering will be used for the three combination approaches, we believe that *fuzziness* is specially important for collaborative combination, as a *crisp* notion of belonging associates documents to a single cluster and makes the process of collaborative learning harder.

3. Testing probabilistic learning algorithms. Learning a probability of belonging to a domain to each pattern instead of a *crisp* membership will give a model closer to reality. The shift towards probability in learning and the shift towards *fuzziness* in clustering are movements in the same direction.

This will be tested on the three combination approaches. However, we want to mention our interest in applying co-clustering to words, patterns and documents to obtain a joint approach. Co-clustering of more than 2 variables poses additional theoretical problems which will have to be solved.

4. In parallel to the mentioned research, analyzing the features used for representation of documents and for patterns. The incorporation of semantic features will be considered, specially for patterns, as it would allow a semantic generalization of the learned patterns, which is not possible now. This will be used for the three combination approaches.

5. Relating to evaluation, intrinsic evaluations of domain relevance of the patterns will be carried out, using human judges.

6. Intrinsic evaluations of the patterns will also be carried out on full IE tasks, using annotated corpora such as those from ACE and MUC evaluations.

7. Extrinsic evaluations will be carried on QA and Summarization tasks. Written text documents will be used in the TREC, CLEF and DUC evaluations, but we also intend to use heterogeneous data coming from the CHIL project.

5.1 Thesis Project Scheduling

This thesis work is expected to conclude in two years from now. The Gantt chart [Gantt, 1910] in Figure 5.1 shows the devised project scheduling.

During the first month we will perform the improvement of the weight detection procedure for our existing sequential approach (item 1). The following 7 months will be devoted to research in *fuzzy* clustering (item 2), and the next 10 months, to probabilistic learning algorithms (item 3). In parallel, we will try different feature sets for documents and patterns (item 4). We expect to eventually come to a definitive set in a point during the experiments on probabilistic learning, and then finish our research with this fixed set.

In the meantime, the different evaluations (items 5, 6 and 7) will be carried out.

The last 6 months of our work will be spent in the writing of the doctoral thesis dissertation.

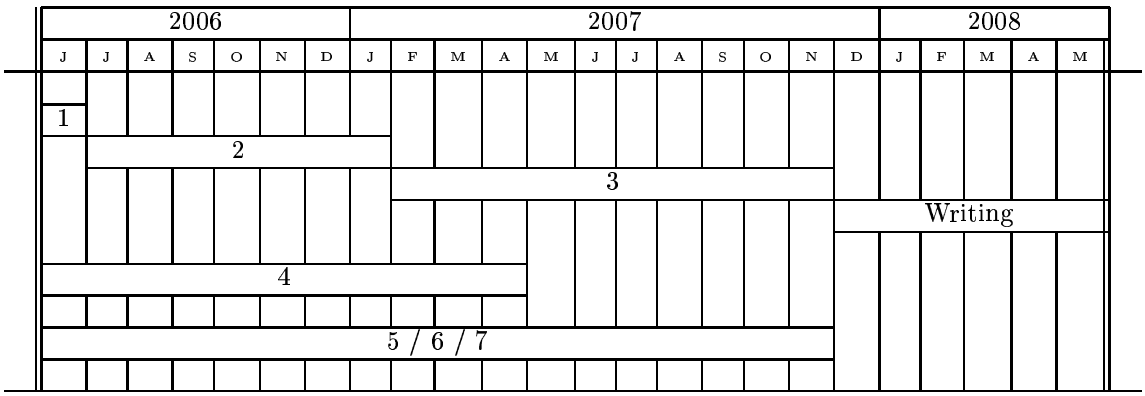


Figure 5.1: Thesis Project Scheduling

Chapter 6

Publications

Here are listed the publications that the research carried so far has produced, classified by which of the three main areas that the author's work has embraced they cover: document clustering, question answering and summarization.

6.1 Document Clustering

- [Gonzàlez and Turmo, 2005]
  Edgar Gonzàlez, Jordi Turmo
  Unsupervised Clustering of Spontaneous Speech Documents
  Proceedings of the 9th European Conference on Speech Communication and Technology (InterSpeech), 2005.

- [Gonzàlez and Turmo, 2006]
  Edgar Gonzàlez, Jordi Turmo
  Unsupervised Document Clustering by Weighted Combination
  LSI Research Report, LSI-06-17-R, 2006

6.2 Question Answering

- [Ferrés et al., 2004a]
  Dani Ferrés, Saamir Kanaan, Alicia Ageno, Edgar Gonzàlez, Horacio Rodríguez, Mihai Surdeanu, Jordi Turmo
  TALP-QA System for Spanish at CLEF-2004: Structural and Hierarchical Relaxing over Semantic Constraints
  Cross-Lingual Evaluation Forum (CLEF) Evaluation Campaign, 2004

- [Ferrés et al., 2004b]
  Dani Ferrés, Saamir Kanaan, Edgar Gonzàlez, Alicia Ageno, Horacio Rodríguez, Mihai Surdeanu, Jordi Turmo
  TALP-QA System at TREC-2004: Structural and Hierarchical Relaxing over Semantic Constraints
  Text Retrieval Conference (TREC) Evaluation Campaign, 2004

- [Ferrés et al., 2005b]
  Daniel Ferrés, Samir Kanaan, Edgar Gonzàlez, Alicia Ageno, Horacio Rodríguez, Jordi Turmo
  The TALP-QA System for Spanish at CLEF-2005
  Cross-Lingual Evaluation Forum (CLEF) Evaluation Campaign, 2005

- [Ferrés et al., 2005a]
  Daniel Ferrés, Saamir Kanaan, David Domínguez-Sal, Edgar Gonzàlez, Alicia Ageno, Maria Fuentes, Horacio Rodríguez, Mihai Surdeanu, Jordi Turmo
  TALP-UPC at TREC 2005: Experiments Using Voting Scheme Among Three Hetereogeneous QA Systems
  Text Retrieval Conference (TREC) Evaluation Campaign, 2005

6.3 Summarization

- [Fuentes et al., 2004]
  Maria Fuentes, Edgar Gonzàlez, Horacio Rodríguez
  Resumidor de Notícies en Català del Projecte Hermes
  II Congrés d'Enginyeria en Llengua Catalana, 2004

- [Fuentes et al., 2005c]
  Maria Fuentes, Edgar Gonzàlez, Jordi Turmo
  Baseline Summarization System for Text including Speech Transcripts
  European Project CHIL (IP 506909) Deliverable D5.8, 2005

- [Fuentes et al., 2005b]
  Maria Fuentes, Edgar Gonzàlez, Horacio Rodríguez, Jordi Turmo, Laura Alonso
  Summarizing Spontaneous Speech Using General Text Properties
  Proceedings of the Crossing Barriers in Text Summarization Research Workshop
  Recent Advances in Natural Language Proceessing (RANLP), 2005

- [Fuentes et al., 2005a]
  Maria Fuentes, Edgar Gonzàlez, Daniel Ferrés, Horacio Rodríguez
  QASUM-TALP at DUC 2005 Automatically Evaluated with a Pyramid based Metric
  Document Understanding Conference (DUC) Evaluation Campaign, 2005

Bibliography

K. Abe, S. Kawaose, T. Asai, H. Arimura, and S. Arikawa. Optimized substructure discovery for semi-structured data. In *Proceedings of the 6th European Conference on Principles and Practice of Knowledge in Databases (PKDD)*, 2002.

S. Abney. Understanding the yarowsky algorithm. *Computational Linguistics*, 30(3), 2004.

ACE05. The ace 2005 (ace05) evaluation plan, 2005.

E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM Conference on Digital Libraries (DL)*, pages 85–94, 2000.

E. Agirre and G. Rigau. Word sense disambiguation using conceptual density. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, 1996.

D. Appelt, J. Bear, J. Hobbs, D. Israel, and M. Tyson. Description of the fastus system used for muc-4. In *Proceedings of the 4th Message Understanding Conference (MUC)*, 1992.

J.H. Aseltine. Wave: An incremental algorithm for information extraction. In *Proceedings of the AAAI Workshop on Machine Learning for Information Extraction*, 1999.

R. Basili, M.T. Pazienza, and M. Vindigni. Corpus-driven learning of event recognition rules. In *Proceedings of Machine Learning for Information Extraction Workshop (ECAI)*, 2000.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT)*, pages 92–100, 1998.

D.M. Boulton and C.S. Wallace. The information content of a multistate distribution. *Journal of Theoretical Biology*, 23:269–278, 1969.

S. Brin. Extracting patterns and relations from the world-wide web. In *Proceedings of the 1998 International Workshop on the Web and Databases (WebDB)*, 1998.

R. Bunescu and R.J. Mooney. Collective information extraction with relational markov networks. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2004.

M.E. Califf. *Relational Learning Techniques for Natural Language Information Extraction.* PhD thesis, University of Texas at Austin, 1998.

T. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3:1–27, 1974.

C. Cardie. Empirical methods in information extraction. *AI Magazine*, 18(4):65–79, 1997.

N. Català, N. Castell, and M. Martín. A portable method for acquiring information extraction patterns without annotated corpora. *Natural Language Engineering*, 9(2):151–179, 2003.

J.Y. Chai and A.W. Biermann. The use of lexical semantics in information extraction. In *Proceedings of the ACL Workshop on Natural Language Learning*, 1997.

J.Y. Chai, A.W. Biermann, and C.I. Guinn. Two dimensional generalization in information extraction. In *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI)*, 1999.

H.L. Chieu and H.T. Ng. A maximum entropy approach to information extraction from semi-structured and free text. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI)*, 2002.

H.L. Chieu, H.T. Ng, and Y.K. Lee. Closing the gap: Learning-based information extraction rivaling knowledge-engineering methods. In *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 216–223, 2003.

N. Chinchor. Muc-4 evaluation metrics. In *Proceedings of the 4th Message Understanding Conference (MUC)*, pages 22–29, 1992.

C. Cox, J. Nicolson, J. Finkel, C. Manning, and P. Langley. Template sampling for leveraging domain knowledge in information extraction. In *First PASCAL Challenges Workshop*, 2005.

G.F. DeJong. Prediction and substantiation: A new approach to natural language processing. *Cognitive Science*, 3:251–273, 1979.

G.F. DeJong. An overview of the frump system. In W.G. Lehnert and M.H. Ringle, editors, *Strategies for Natural Language Processing*, pages 146–176. Erlbaum, 1982.

A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Royal Statistical Society, Series B*, 39(1), 1977.

I.S. Dhillon and Y. Guan. Information theoretic clustering of sparse co-occurrence data. In *Proceedings of IEEE International Conference on Data Mining*, 2003.

E. Dimitriadou. *Exploratory Data Analysis and Applications.* PhD thesis, Technische Universität Wien, 2003.

C. Fellbaum, editor. *WordNet: An Electronic Lexical Database.* MIT Press, 1998.

J.T.A.S. Ferreira, D.G.T. Denison, and D.J. Hand. Data mining with products of trees. In *Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis (IDA)*, 2001.

D. Ferrés, S. Kanaan, A. Ageno, E. Gonzàlez, H. Rodríguez, M. Surdeanu, and J. Turmo. Talp-qa system for spanish at clef-2004: Structural and hierarchical relaxing over semantic constraints. In *CLEF 2004 Evaluation Campaign*, 2004a.

D. Ferrés, S. Kanaan, D. Domínguez-Sal, E. Gonzàlez, A. Ageno, M. Fuentes, H. Rodríguez, M. Surdeanu, and J. Turmo. Talp-upc at trec 2005: Experiments using a voting scheme among three hetereogeneous qa systems. In *TREC 2005 Evaluation Campaign*, 2005a.

D. Ferrés, S. Kanaan, E. Gonzàlez, A. Ageno, H. Rodríguez, M. Surdeanu, and J. Turmo. Talp-qa system at trec-2004: Structural and hierarchical relaxing over semantic constraints. In *TREC 2004 Evaluation Campaign*, 2004b.

D. Ferrés, S. Kanaan, E. Gonzàlez, A. Ageno, H. Rodríguez, and J. Turmo. The talp-qa system for spanish at clef-2005. In *CLEF 2005 Evaluation Campaign*, 2005b.

A. Finn and N. Kushmerick. Information extraction by convergent boundary classification. In *Proceedings of the AAAI Workshop on Adaptive Text Extraction and Mining*, 2004.

D. Freitag. Using grammatical inference to improve precision in information extraction. In *Notes of the ICML-97 Workshop on Automata Induction, Grammatical Inference, and Language Acquisition*, 1997.

D. Freitag. *Machine Learning for Information Extraction in Informal Domains.* PhD thesis, Carnegie Mellon University, 1998a.

D. Freitag. Multistrategy learning for information extraction. In *Proceedings of the 15th International Machine Learning Conference*, pages 161–169, 1998b.

D. Freitag and A. McCallum. Information extraction with hmm structures learned by stochastic optimization. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI)*, 2000.

M. Fuentes, E. Gonzàlez, D. Ferrés, and H. Rodríguez. Qasum-talp at duc 2005 automatically evaluated with a pyramid based metric. In *DUC 2005 Evaluation Campaign*, 2005a.

M. Fuentes, E. Gonzàlez, and H. Rodríguez. Resumidor de notícies en català del projecte hermes. In *Proceedings of the II Congrés d'Enginyeria en Llengua Catalana*, 2004.

M. Fuentes, E. Gonzàlez, H. Rodríguez, J. Turmo, and L. Alonso. Summarizing spontaneous speech using general text properties. In *Proceedings of the Crossing Barriers in Text Summarization Research Workshop at RANLP 2005*, 2005b.

M. Fuentes, E. Gonzàlez, and J. Turmo. Baseline summarization system for text including speech transcripts. European Project CHIL (IP 506909) Deliverable D5.8, 2005c.

H.L. Gantt. *Work, Wages and Profit.* The Engineering Magazine, 1910.

R. Girju. Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering*, pages 76–83, 2003.

O. Glickman and R. Jones. Examining machine learning for adaptable end-to-end information extraction systems. In *Proceedings of the AAAI Workshop on Machine Learning for Information Extraction*, 1999.

J.J. Godfrey, E.C. Holliman, and J. McDaniel. Switchboard: Telephone speech corpus for research and development. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1992.

E. Gonzàlez and J. Turmo. Unsupervised clustering of spontaneous speech documents. In *Proceedings of the 9th European Conference on Speech Communication and Technology (EuroSpeech/InterSpeech)*, 2005.

E. Gonzàlez and J. Turmo. Unsupervised document clustering by weighted combination. Technical report, LSI Department, Universitat Politècnica de Catalunya, 2006.

S. Harabagiu and S. Maiorano. Acquisition of linguistic patterns for knowledge-based information extraction. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC)*, 2000.

J.A. Hartigan. *Clustering Algorithms.* John Wiley & Sons, 1975.

V. Hatzivassiloglou, L. Gravano, and A. Maganti. An investigation of linguistic features and clustering algorithms for topical document clustering. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 224–231, 2000.

J. Hobbs. The generic information extraction system. In *Proceedings of the 5th Message Understanding Conference (MUC)*, pages 87–92, 1993.

S. Huffman. Learning information extraction patterns from examples. In *Proceedings of the IJCAI Workshop on New Approaches to Learn for NLP*, 1995.

A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31 (3):264–323, 1999.

J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*, 1997.

N. Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2004.

G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning: Applications in vlsi domain. In *Proceedings of the Design and Automation Conference*, 1997.

G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.

J. Kim and D. Moldovan. Acquisition of linguistic patterns for knowledge-based information extraction. *IEEE Transactions on Knowledge and Data Engineering*, 7(5), 1995.

H. Ko. Empirical sequence assembly planning: A multistrategy constructive learning approach. In I. Bratko, R.S. Michalski, and M. Kubat, editors, *Machine Learning and Data Mining*. John Wiley & Sons, 1998.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

W. Lehnert, C. Cardie, D. Fisher, E. Riloff, and R. Williams. Description of the circus system as used for muc-3. In *Proceedings of the 3rd Message Understanding Conference (MUC)*, 1991.

C-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the ACL-04 Workshop on Text Summarization Branches Out*, 2004.

J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

A. McCallum and D. Jensen. A note on the unification of information extraction and data mining using conditional-probability, relational models. In *Proceedings of the IJCAI-03 Workshop on Learning Statistical Models from Relational Data*, 2003.

R.S. Michalski. Towards a unified theory of learning: Multistrategy task-adaptative learning. In B.G. Buchanan and D. Wilkins, editors, *Readings in Knowledge Acquisition and Learning*. Morgan Kauffman, 1993.

G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Five papers on wordnet. *International Journal of Lexicography*, 3(4), 1990. Special Issue.

S. Miller, M. Crystal, H. Fox, L. Ramshaw, R. Schwartz, R. Stone, and R. Weischedel. Description of the sift system used for muc-7. In *Proceedings of the 7th Message Understanding Conference (MUC)*, 1998.

S. Miller, H. Fox, L. Ramshaw, and R. Weischedel. A novel use of statistical parsing to extract information from text. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, 2000.

S. Muggleton. Inverse entailment and progol. *New Generation Computing*, 13:245–286, 1995.

S. Muggleton and W. Buntine. Machine invention of first-order predicates by inverting resolution. In *Proceedings of the 5th International Conference on Machine Learning (ICML)*, 1988.

S. Muggleton and C. Feng. Efficient induction of logic programs. In S. Muggleton, editor, *Inductive Logic Programming*. Academic Press, 1992.

M.N. Murty and G. Krishna. A computationally efficient technique for data clustering. *Pattern Recognition*, 12:153–158, 1980.

I. Muslea. Extraction patterns for information extraction tasks: A survey. In *Proceedings of AAAI-99 Workshop on Machine Learning for Information Extraction*, 1999.

A. Nenkova and R. Passonneau. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Conference on Human Language Technologies (HLT/NAACL)*, 2004.

K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2/3), 2000.

L. Peshkin and A. Pfeffer. Bayesian information extraction network. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.

R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5(3):239–266, 1990.

R. Quinlan and R.M. Cameron-Jones. Foil: A midterm report. In *Proceedings of the European Conference in Machine Learning (ECML)*, pages 3–20, 1993.

S. Ray and M. Craven. Representing sentence structure in hidden markov models for information extraction. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)*, 2001.

E. Riloff. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the 11th National Conference on Artificial Intelligence (AAAI)*, pages 811–816, 1993.

E. Riloff. Automatically generating extraction patterns from untagged text. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI)*, pages 1044–1049, 1996.

E. Riloff and R. Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI)*, 1999.

J.J. Rissanen. Modelling by shortest data description. *Automatica*, 14:465–471, 1978.

T.G. Rose, M. Stevenson, and M. Whitehead. The reuters corpus volume 1 - from yesterday's news to tomorrow's language resources. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC)*, 2002.

D. Roth and W. Yih. Relational learning via propositional algorithms: An information extraction case study. In *Proceedings of the 15th International Conference On Artificial Intelligence (IJCAI)*, 2001.

N. Sager. *Natural Language Information Processing: A Computer Grammar of English and its Applications*. Addison-Wesley, 1981.

C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27: 279–423, 623–656, 1948.

S. Siersdorfer and S. Sizov. Restrictive clustering and metaclustering for self-organizing document collections. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 226–233, 2004.

M. Skounakis, M. Craven, and S. Ray. Hierarchical hidden markov models for information extraction. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.

N. Slonim. *The Information Bottleneck: Theory and Applications.* PhD thesis, The Hebrew University, 2003.

N. Slonim and N. Tishby. Agglomerative information bottleneck. In *Proceedings of Advances in Neural Information Processing Systems (NIPS-12)*, 1999.

S. Sodeland, D. Fisher, J. Aseltine, and W. Lehnert. Crystal: Inducing a conceptual dictionary. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1314–1321, 1995.

S. Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34:233–272, 1999.

K. Spärck-Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

M. Stevenson and M.A. Greenwood. A semantic approach to ie pattern induction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 379–386, 2005.

A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

K. Sudo. *Unsupervised Discovery of Extraction Patterns for Information Extraction.* PhD thesis, New York University, 2004.

A. Sun, M. Naing, E. Lim, and W. Lam. Using support vector machines for terrorism information extraction. In *Proceedings of the First NSF/NIJ Symposium on Intelligence and Security Informatics (ISI)*, pages 1–12, 2003.

M. Surdeanu, J. Turmo, and A. Ageno. A hybrid unsupervised approach for document clustering. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2005.

M. Surdeanu, J. Turmo, and A. Ageno. A hybrid approach for the acquisition of information extraction patterns. In *Proceedings of the EACL 2006 Workshop on Adaptive Text Extraction and Mining (ATEM)*, pages 49–56, 2006.

D.K. Tasoulis and M.N. Vrahatis. Unsupervised distributed clustering. In M.H. Hamza, editor, *Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN)*, 2004.

C.A. Thompson, M.E. Califf, and R.J. Mooney. Active learning for natural language parsing and information extraction. In *Proceedings of the 16th International Machine Learning Conference*, 1999.

A. Topchy, A.K. Jain, and W. Punch. Clustering ensembles: Models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12), 2005.

J. Turmo. *An Information Extraction System Portable to New Domains.* PhD thesis, Universitat Politècnica de Catalunya, 2002.

J. Turmo, A. Ageno, and N. Català. Adaptive information extraction. *ACM Computing Surveys*, 38, 2006. To appear.

J. Turmo and H. Rodríguez. Learning rules for information extraction. *Natural Language Engineering*, 8:167–191, 2002. Special Issue on Robust Methods in Analysis of Natural Language Data.

W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 267–273, 2003.

R. Yangarber. Counter-training in discovery of semantic patterns. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 343–350, 2003.

R. Yangarber and R. Grishman. Customization of information extraction systems. In P. Velardi, editor, *International Workshop on Lexically Driven Information Extraction*, 1997.

R. Yangarber and R. Grishman. Issues in corpus-trained information extraction. In *Proceedings of the International Symposium on Spontaneous Speech: Toward the Realization of Spontaneous Speech Engineering*, 2000.

R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the Conference on Computational Linguistics (COLING)*, pages 940–946, 2000.

D. Yarowsky. Unsupervised word sense disambiguation rivaling unsupervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 1995.

S. Young and B. Bloothoft, editors. *Corpus-Based Methods in Language and Speech Processing.* Kluwer Academic Press, 1997.

D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. *Journal of Machine Learning Research 3*, pages 1083–1106, 2003.

J.M. Zelle and R.J. Mooney. Inducing deterministic prolog parsers from treebanks: A machine learning approach. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI)*, pages 748–753, 1994.

S. Zhao and R. Grishman. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 419–426, 2005.

Y. Zhao and G. Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3), 2004.