

CONTENTS

	Page
INTRODUCTION	1
CHAPTER 1 LITERATURE REVIEW	13
1.1 Concept extraction systems and methods	13
1.1.1 Generic keywords extraction methods	13
1.1.2 Software engineering centered extraction systems	15
1.2 Multiword expression extraction	17
1.3 Acronym detection for candidate	19
CHAPTER 2 ARTICLE I: CONCEPT EXTRACTION FROM BUSINESS DOCUMENTS FOR SOFTWARE ENGINEERING PROJECTS	23
2.1 Introduction	24
2.1.1 Motivation	25
2.1.2 Context of application	26
2.2 Automated concept extraction	28
2.3 Proposed approach	30
2.3.1 Gold standard definition	32
2.3.2 Evaluation metrics	36
2.4 Detailed extraction process	41
2.4.1 Implementation overview	41
2.4.2 Candidate extraction	42
2.4.3 Inconsistency filter	44
2.4.4 Stop list exclusion	44
2.4.5 Acronym resolution	46
2.4.6 Complex multiword expressions detection	47
2.4.7 Dictionary validation	48
2.4.8 Web validation	49
2.4.9 Structure detection and analysis	50
2.4.10 Relevance ordering algorithm	52
2.5 Evaluation	55
2.5.1 Experimental setup	55
2.5.2 Baseline comparison	56
2.6 Results	58
2.6.1 Candidate extraction	58
2.6.2 Candidates ordering	59
2.7 Conclusion	61
CHAPTER 3 ARTICLE II: HYBRID EXTRACTION METHOD FOR FRENCH COMPLEX NOMINAL MULTIWORD EXPRESSIONS	63
3.1 Introduction	64

3.2	Overview	65
3.2.1	Context	65
3.2.2	Examples	66
3.2.3	Challenges and issues	67
3.2.4	Uses and proposed method	69
3.3	Details of the proposed method	70
3.3.1	Acquisition of ngrams	70
3.3.2	Semilattice construction	71
3.3.3	Reduction of semilattices	73
3.4	Evaluation	75
3.4.1	Corpora	75
3.4.2	Baseline	78
3.4.3	Results	79
3.5	Conclusion	83
CHAPTER 4 ARTICLE III: CLASSIFIER-BASED ACRONYM EXTRACTION FOR BUSINESS DOCUMENTS		85
4.1	Introduction	86
4.2	Background	87
4.2.1	Definition	87
4.2.2	Morphology	88
4.2.3	Context of use	90
4.3	Specific properties of business documents	92
4.3.1	Main differences	93
4.3.2	Extraction challenges	95
4.3.3	Special cases	96
4.4	Methodology	99
4.4.1	Overview	99
4.4.2	Preparation step	99
4.4.3	Generic acronym repository	101
4.4.4	Short-form identification	104
4.4.5	SF-LF candidate extraction	105
4.5	Candidate SF-LF evaluation	107
4.5.1	Structural features	107
4.5.2	Similarity features	110
4.5.3	Cleanup step	112
4.6	Experiment	113
4.6.1	Parameter and score template definition	113
4.6.2	Corpora	114
4.6.3	Classifier training	115
4.7	Results	117
4.8	Future work	121
4.9	Conclusion	121

CHAPTER 5 GENERAL DISCUSSION	123
5.1 Concept extraction from business documents for software engineering projects	123
5.1.1 Performance and usage	123
5.1.2 Extraction issues	124
5.1.3 Evaluation method	125
5.1.4 Gold standard	125
5.2 Hybrid extraction method for French complex nominal multiword expressions	126
5.2.1 Performance	126
5.2.2 Sources of errors	127
5.3 Classifier-based acronym extraction for business documents	128
5.3.1 Performance and features	128
5.3.2 Limitations	130
GENERAL CONCLUSION	133
APPENDIX I REVISED MULTIWORD EXPRESSION GOLD STANDARD FOR THE FRENCH TREEBANK AND LAPORTE CORPORA	137
REFERENCES	143
BIBLIOGRAPHY	144

LIST OF TABLES

	Page	
Table 1.1	Concept extraction method summary	14
Table 2.1	Summary of the gold standard definition effort by number of selected relevant concepts per annotator	34
Table 2.2	MDR calculation example	37
Table 2.3	Generalized exclusion example	45
Table 2.4	Number of noun-based multiword expressions by length in the DELA	49
Table 2.5	Examples of list structure	52
Table 2.6	MDR measure for the reordering process	61
Table 3.1	Total number of expressions extracted by each method on the French Treebank	83
Table 3.2	Average results for expression of length ≥ 5 for the original and revised gold standards	84
Table 4.1	Features used to create acronyms	89
Table 4.2	Generalized patterns for presentation cases	92
Table 4.3	Attributes of major corpora's types	93
Table 4.4	Acronym's type ranking	97
Table 4.5	Number of non trivial feature acronyms by language	102
Table 4.6	Identification rules for an SF	105
Table 4.7	Subcomponent of the potential score feature	108
Table 4.8	Potential score ratio example	109
Table 4.9	Score template definition	114
Table 4.10	Size and valid entries for each dataset	116
Table 4.11	Average algorithm's performance on the evaluation corpus	118

Table 4.12	Algorithm performance per search space	119
Table 4.13	Feature's relevance rank for search space 1.....	120
Table 4.14	Feature's relevance rank for search space 6.....	120

LIST OF FIGURES

	Page	
Figure 0.1	Disrupted sentence structure example	8
Figure 2.1	Scope and relation of knowledge containers	27
Figure 2.2	Distribution of concepts per number of annotators.....	35
Figure 2.3	Cumulative gold and system from Table 2.2	39
Figure 2.4	Monotonic cumulative gold and reverse cumulative system.....	40
Figure 2.5	Modular composition of the extraction process	41
Figure 2.6	Example of contextual distribution of weights to surrounding concepts	54
Figure 2.7	Concept-based performance for each cleaning step	59
Figure 2.8	Ranked system (RS) curves compared to ranked gold (RG) on various document sets	60
Figure 3.1	Complete semilattice with calculated T_k growth ratio (on edges) and the ngram frequencies (within nodes)	70
Figure 3.2	Example of semilattices construction step D	73
Figure 3.3	Semilattices reduction steps	73
Figure 3.4	Term count for gold standard of processed corpora	77
Figure 3.5	Results of extraction process against the original gold reference.....	80
Figure 3.6	Results of extraction process against the revised gold reference	81
Figure 4.1	Processing steps and resources	100
Figure 4.2	Acronym length profile of the generic database	102
Figure 4.3	Average ratio of LF to SF length	103
Figure 4.4	Match-generation example	107
Figure 4.5	Coverage calculation examples	111

LIST OF ABREVIATIONS

CMWE	Complex multiword expression
LF	Acronym's long form or descriptive form
F-1	F-measure
MDR	Mean discovery ratio
MWE	Multiword expression
RUP	Rational unified process
SF	Acronym's short form
SRS	Software requirements specification
UML	Unified modeling language

INTRODUCTION

Overview

Software engineering teams starting a software development project face many challenges of technological, political, organizational and human nature. Amongst them, the domain knowledge acquisition and understanding task is one that has many impacts on the success of the project as it helps define many aspects like the scope, vision, needs and requirements. These items help to better orient and focus the development effort throughout the whole project duration.

While some software engineering experts have acquired other domain specialization by working on projects from various business domains (banking, manufacturing, retail, transportation, etc.), most of them are not experts in all their client's field of activity. As such, these experts have to acquire all the specialized knowledge by using various methods like interviews, on-site observations, trainings, group meetings and analysis of textual work documents. Teams sometimes hire a domain expert from one of their previous or current clients to act as a non-technical domain or business analyst in order to shorten the knowledge acquisition and understanding effort. But most of these methods are dependent on the client's availability which is often problematic, as time, budget, and dedication to a project is not a given. In these cases, as well as for less problematic projects, the internal business documents can be a valuable source of domain information about the client's organization, its inner workings and on valuable domain concepts. This knowledge can be reused downstream in the development project.

In fact, it is well known that most of the information of an organization is mostly contained within unstructured textual documents traditionally on paper but more currently in electronic format (Blumberg and Aire, 2003). They can be found in various formats: internal process document, emails, help desk notes, memos, corporate web site, white papers, technical papers, reports and so on. A study from the Meryll-Lynch reports that 80% of business knowledge is contained in unstructured documents and that this data doubles every three months.

Extracting domain knowledge from these sources is far from trivial as these documents are not clearly structured or optimized for this task. They are usually written in natural languages (English, French, Spanish, etc.) which are informal by nature, offers a plethora of unsolved challenges. For example, authors might paraphrase the same information in a large variety of styles, complexity levels or structures, the same sentence might take on wildly different meanings depending on the context, the authors' conceptual representation, etc. In consequence, many software engineering experts prefer to avoid these documents, missing useful information by doing so. This research project aims to alleviate this extraction effort by applying natural language processing techniques to the task at hand.

Systems have been developed for many years, mainly for English documents, to support the software engineering experts during the domain modeling effort. Alas, these systems never gained much popularity outside research groups. Indeed, very few commercial software modeling products offer textual analysis tools to help the user extract useful concepts. While a robust and targeted method might still be far away because of the many unsolved challenges, the current state of the natural language processing methods might be improved to better support and promote advances in this field.

Context of the thesis

The first knowledge intensive task usually performed by a software engineer expert at the start of a software development project is to extract the domain model from the organizational information sources. This model is defined in the Rational Unified Process (Kruchten, 1999) as “a standalone subset of the Business Analysis Model that focuses on concepts, products, deliverables, and events that are important to the business domain”. Booch *et al.* (1999) also defines it as capturing “... the most important types of objects in the context of the business. The domain model represents the “things” that exist or events that transpire in the business environment”. Others, as reported in Pressman (2001), consider that domain analysis “involves modeling the domain so that software engineers and other stakeholders can better learn about it... not all domain classes necessarily result in the development of reusable classes”.

Regardless of the exact definition, the emphasis put on the concept's provenance, the business domain, shows that using business documents as a source of knowledge is founded. These documents usually explain the inner workings and definitions of business concepts intervening in the client's project. They can be thesaurus or lexicons containing concept's definitions, memos about new directives to follow regarding specific events or products, directives describing the proper procedure to follow in certain scenarios, forms that specifies the information needed to start or end an event, a work procedure that details the necessary steps to successfully conduct a business oriented activity, a published manual about the partnership program of a company, and so on. These documents can differ in their level of structure, ranging from unstructured free or narrative text (emails, memos, product descriptions), or more structured like detailed step-by-step work procedures or forms, but they are never considered structured as a database can be. Nonetheless, many of the concepts used in these documents are of interest to the domain model definition in order to cover the full scope of activity of the organization at hand.

To extract these relevant concepts, the software engineering experts must read and search within multiple documents that make up the complete corpus of business documentation of an organization. But searching within this mass of documents, which can measure up from a few dozens to many thousand units, is non-trivial and time consuming as they are usually not optimized for such task. It was shown in past research (Butler Group, 2005) that as much as 10% of an organization's total salary is spent in staff searching for information and that knowledge workers spend as much as 40% of their time looking for and managing information, which supports this point. Also, as shown in Meadow *et al.* (2007), a hierarchy of focus applied to information retrieval tasks can be differentiated to suit the needs of knowledge workers :

- a. Search for known items
- b. Search for specific information
- c. Search for general information
- d. Exploration

The first two types, in the current context, would apply when looking for documents concerning known concepts in order to investigate their context and relationship to others. Type 3 and 4 are more in direct alignment with the domain modeling effort when dealing with a previously unknown domain. The general information search targets the discovery of instances of known types of information identified before-hand, like actors, documents, products, artifacts, etc. On the other hand, exploration deals with previously unknown information items and thus requires full scoped research of the corpus in order to enable relevant discoveries. In this light, discovery of relevant concepts from textual documents is considered far from trivial.

The exploration phase is critical in the current context. As the relevant business documentation associated to a software project can be very large, there is a need to browse a wide spectrum of concepts. At the same time, in order for the expert to be confident enough about the output quality, documents must be effectively processed to extract the maximum number of relevant concepts. If it is not the case, he would be required to read the documentation in order to double check the output. In this context, an extraction system should prioritize a high recall of relevant process over a high precision.

In the context of domain modeling, an additional cognitive layer is added to the information research activities as the experts need to sort the potential impact or reuse value of each newly found concepts. They must not only retrieve the concepts that can be used directly in the down-stream modeling and specification activities (use cases, class or design diagrams, requirements, etc.) but also identify those that may have an indirect impact on these previous concepts or on the project as a whole. Some cases are straightforward like a client's profile details (name, address, age, etc.) for a retail web site which could probably be reused directly in the design model. Other are more subtle like a law that forces a strict backup and conservation policy on the system's data which must be later defined in the non-functional requirements. A supervising organization's policy could require a strict periodic publication of reports of a specific format and manner which must be planned in the use-cases. Or even an old but active partnership association deal which must be taken into account as a positive or negative stakeholder in the project's scope definition.

The multiple provenance and nature of these concepts increases the search and discovery effort need by the experts at the beginning of a project. Moreover, missing some of these more influencing concepts might have a widespread negative effect on the time frame, budget and success of the project.

Problem statement

As knowledge extraction from textual documents is a non trivial task, many problems must be overcome in order to extract relevant concepts. These issues emanate from the nature of the documents used, their type of content and from the state of the text processing tools. Combined together, these issues worsen the effectiveness of extraction process and limit their application.

The overall issue regarding the extraction of relevant content is their identification. As business documents are created for internal organizational usage, they are not designed to point out important concepts with future software projects in mind. Relevant concepts therefore float in an ocean of irrelevant terms and expressions. These irrelevant terms and expressions are valid by themselves and can also be considered, sometimes, as part of the domain knowledge, without the need to extract them. As an example, in the sentence “The test technician must then sign the approval form and leave it on his manager’s desk” which describes part of a work procedure, five concepts can be considered for extraction :

- a. Test technician
- b. Approval form
- c. Manager
- d. Desk

While all these concepts can be considered domain knowledge, probably only the first three are useful to extract, and to know, at the start of a project. The manager and test technician can be seen as stakeholders or potential users of the future management system. But the fact

that the test technician's manager has a desk is probably useless in this context. While it constitutes only one fifth of the relevant concepts in this sentence, superfluous or noisy terms and expressions usually outnumber relevant concepts by an order of magnitude.

Moreover, many concepts, domain specific or not, are not always easily identifiable in a sentence. In cases similar to the expression “le formulaire d'approbation du directeur du département des plaintes, communication et du marketing” (“the complaint, communication and marketing department director's approval form”), different individuals could see two or more concepts :

- a. An approval form
- b. A form
- c. A director
- d. A department
- e. A complaint
- f. The complaints, communication and marketing department
- g. The complaints, communication and marketing department's director

Concepts from A to E are more generic in nature while the last two can be seen as named entities, which denotes a name for a specific instance of a concept within the organization. A software engineering expert could include any, none or all the previous concepts in a model of the problem domain of a project. Items A, B and E could be seen as business products, item C and D could be reused as actors in a use-case model while the two last concepts could be tagged as stakeholders. One of the issues in this example is that most pattern based extraction methods would probably miss the two last concepts and segment them down to single (department, complaints, communication, marketing, director) or two-words expressions (department of complaints, director of department).

To make matters worse, when applying a deep parser tool to analyze the constituents and their relationships, these complex expressions produce many syntactic trees giving rise to an exponential augmentation of potential interpretation, usually lowering the quality of the extracted concepts to the point of uselessness. Frequency-based statistical tools which produce candidate expressions are also impacted as they generate and evaluate, using restriction patterns, all the possible sequences of terms : “directeur” (“director”), “directeur du département” (“department director”), “directeur du département de plaintes” (“complaints department director”), “directeur du département de plaintes, communication” (communication, complaints department director”), and so on. Many of these tools will also miss the last two expressions as they usually stop at commas to avoid overgenerating less plausible candidates. They will also, most of the time, favor smaller concepts and excludes large ones that contains them.

Regarding the nature of business documents, many different types can be denoted. Some contain free flowing text as emails, blog entries, mission statements or general directives. This type is highly unstructured as it often contains no indications of change of subject or items of interest. Other unstructured documents contain some implicit or explicit structural hints like main title, titles, subtitles, footnotes, detailed lists and tables. Documents can also be found with a semi-structured format, like lexicon or forms. While not explicit like in a database, these documents provide a more detailed segmentation of a document’s informational content. But more often than not, text analysis of these documents fails as they are composed of incomplete text fragment, unusual text positioning and other phenomenon that hinder generic methods.

Of course, all these content types are not analyzed or parsed by text processing tools with the same output quality. A sentence tokenizer and parser will work well on free flowing text but will encounter difficulties while parsing a list of actions (verbal phrases) introduced by a half-sentence which must be applied to each items as in the text extract in Figure 0.1.

This disrupted sentence model is often seen in internal documents and can contain relevant and detailed knowledge about a project’s target domain. This problem is known and stated in Tanguy and Tulechki (2009) on information retrieval: “[...] an increased complexity of

To complete the new transaction, the manager must then:

- enter his password,
- type in the transaction id,
- confirm the form,
- transmit the approval number to the client.

Figure 0.1 Disrupted sentence structure example

the indexed documents or the queries can be associated with a decrease in the performance of the system. Accurately predicting document or query complexity in order to selectively trigger further processing of more complex zones may therefore have an impact on overall performance.”

Finally, one non-technical issue which is often problematic is knowing when to stop extracting new concepts. In a way, it is much like the “undiscovered ruins syndrome” described in Leffingwell and Widrig (2000) who tell that the more are found, the more are to be found because you never really feel as though you have found them all, and perhaps you never will. As a system digs up relevant concepts, it also produces a lot of noise, irrelevant concepts or invalid expressions, and it has to evaluate if it should keep digging for undiscovered concepts or if it should produce a non-significant amount of relevance compared to noise. Statistic-based tools often have this problem as huge amount of noise is generated. They usually use an empirically adjusted threshold level, like a minimum term frequency or other measures, in order to limit the size of their output. Alas, such thresholds must often be fine-tuned in order to provide quality results over a specific text corpus.

Contributions

The global goal of this thesis is to define a pipeline of text analysis modules available in French to support the knowledge extraction effort in the context of a software development project. French documents were specifically chosen to test the approach on languages where tools and

resources are less mature and readily available, thus enabling the developed method to be applied on a wider range of languages. This pipeline should target a high recall over precision in order to support the concept exploration phase of knowledge acquisition. The base hypothesis of this goal is that structural hints in a business document can be used to boost the relevance ordering of noisy concepts list better than statistic based methods. This intuition of the observations that authors often put emphasis on relevant concepts by integrating them as part of structures in documents. In order to attain this goal, more specific objectives are formulated as part of this thesis.

Objective 1: design of an extraction pipeline based on positive and negative filters

The first specific objective is to define the pipeline for an extraction method which can reorder noisy output from large (high recall) candidate lists using positive (to certify valid concepts) or negative (to remove noisy and invalid expressions) filters. A pipeline architecture was adopted over a monolithic one (where components are strongly coupled together as to form a inseparable whole) as they are easier to extend with new tools and update existing ones to reduce noise and boost performance level. These filters should be based on accessible or newly created methods and should not rely on hypothetical perfect results from non-existing tools which are not available not yet exist but produce a near perfect output¹. They should also minimize the loss of relevant concepts (that is keeping the recall as high as possible) while helping to raise the precision gradually.

Objective 2: definition of a gold standard and a measurement method

The second objective is to define a gold standard and a measurement method adapted to the evaluation of software-oriented extraction systems, as neither currently exist that could be used in the context of this research. The gold standard should take into account not only the reusable concepts that can be retrieved for later use but also those that matter to software experts during the knowledge acquisition phase. This gold standard and its creation methodology should be

¹i.e. a module based on the theoretical availability of 100% correct tree structures provided from a syntactic analyzer which is not yet available

able to serve as a stepping stone to larger and more specialized gold standard. The measurement method should take into account both the high recall requirement and the relevance-optimized presentation order of the output of any system. These two objectives are developed in Chapter 2 and are the subject of our first paper.

Objective 3: new extraction method of large multiword expressions

The third objective is to delve into the problem of large multiword expressions and propose an extraction method. These expressions can be found in many business documents and are one of the major source of noise which hinders concept extraction. They are seldom dealt with in the scientific literature as they tend to appear only in business corpora. The extraction method should be implementable in the processing pipeline as a positive module which can certify large and unusual expressions and thus remove some of the underlying noisy candidates. This objective is developed in Chapter 3 and is the subject of our second article.

Objective 4: extraction method for complex and implicit acronyms

The last objective deals with a specific class of multiword expressions, namely the acronyms. An approach should be devised to enable the extraction of acronyms with either a clear and explicit link between the short and long form, or an implicit link where the short and long form are used in proximity but not explicitly associated. The goal is to ascertain the soundness of large noun-based expressions and to link the short and long form. This last point enable the system to define an equivalence link between the two forms which can further reduce the amount of noise. This last objective is detailed in Chapter 4 and is the subject of the third article.

Thesis outline

This thesis is organized as follows: the current chapter introduces the context of our work, the related problems relevant to this research and the intended objectives.

The next three chapters present the methods and results developed in this thesis. Chapter 2 presents the first journal article. It describes the general framework envisioned to support the

knowledge exploration effort as well as the gold standard definition and performance measure associated with the suggested approach. Chapter 3 presents the second journal article. It describes one specific module developed for the above mentioned framework in order to reduce the noise level of the extracted concept list. Chapter 4 presents the third journal article. It details a framework module designed to extract acronyms definition from business documents. Finally, a general discussion is given in Chapter 5 followed by a general conclusion that summarizes a the work accomplished during this research and some prospects for the future.

CHAPTER 1

LITERATURE REVIEW

This section presents, in the first part, recent articles that are related to the concept extraction processes, both generic and specific to the software engineering domain. These articles are related to the two first objectives from the introduction section to put in contrast the issues with the current research effort and the current state of the art in concept extraction for software engineering. The second section is linked to the third objective which targets the extraction of multiword expressions. The third and final section illustrates the recent development in acronym identification and extraction which is the focus of the last objective from Section .

1.1 Concept extraction systems and methods

While there are many articles which propose extraction methods, few are related to the context of software engineering. As such, the first part of this section present recent methods for the extraction of concepts or keywords from text with no specific usage in mind. The second part review the keyword extraction process which target the software domain. All these research efforts are summarized in Table 1.1 which illustrate the generic method used for extraction, the type of document used as the extraction source, the scope of usage of the keywords, the published performance of the method and the language of application.

Research targeting generic or non-software concept extraction will be detailed in the next section. The second section details work which centers around concept extraction for or from the software engineering domain.

1.1.1 Generic keywords extraction methods

Relevant term detection usually uses external resources to give hints to the different tools used in the extraction process about what constitutes an expression or a term worth retaining. Applications working in domain traditionally used in information extraction (IE) like news, finance,

Table 1.1 Concept extraction method summary

Reference	Method	Document type	Scope	Performance	Target language
Ittoo <i>et al.</i> (2010)	Pipeline	Complaints	Generic	91.7% (Prec.)	English
Maynard <i>et al.</i> (2007)	Rule-based	Business	Economy	84% (F-1)	English
Rose <i>et al.</i> (2010)	Statistical	Abstracts	Generic	37.2% (F-1)	Independent
Deeptimahanti and Sanyal (2011)	Patterns	SRS	Software	N/A	English
Popescu <i>et al.</i> (2008)	Pipeline	Software manual	Software	87.61% (F-1)	English
Kof (2010)	Heuristics	SRS	Software	95% (F-1)	English

health and biology may use some of the numerous datasets, ontologies and lexical and semantic resources available for these specific domains. Lexico-semantic repository like Wordnet (Fellbaum, 1998) and Eurowordnet (Vossen, 1998), which contain polysemous terms with their respective contextual examples, with may be used to detect, disambiguate and link textual manifestation of relevant concepts while higher and mid level ontologies like OpenCyc, Sumo and BFO (basic formal ontology) can either be used for term categorization, semantic linking and context identification.

This opportunity is not a given in more specialized multi-domain organizations for which no adapted resources are available or adaptable. They have to rely on linguistic and statistical methods to be able to differentiate between high value knowledge and “filling”. One such application is Textractor (Ittoo *et al.*, 2010) which uses a pipeline of algorithms and techniques for language detection, noise reduction, candidate concept identification and extraction. Their source documents were irregular text description of customer complaints from help desks and repair action of service engineers. They evaluate their resulting concept list using domain experts and achieve a 91.7% precision score.

Other systems like the International Enterprise Intelligence prototype application (Maynard *et al.*, 2007) developed for the European Musing project rely on the repetition of predicted link manifestation in text as their base for robust and high-confidence results. They use pattern-

based extractors which target specific knowledge items like company contact information, managerial structures, activity domain, imported and exported products, shareholders and so on. This type of extraction method expects a precise type of knowledge and structure, so that any piece of information which doesn't fall into the predefined pattern are not recovered. They use domain ontologies to mine business documents for numerical data from various sources as well as table extraction process to obtain these information from semi-structured documents. This is done to support business intelligence which need to assess financial risk, operational risk factors, follow trends or perform credit risk management. The extraction pipeline achieves a f-measure of 84% on twelve types of specific knowledge units. The precision range from 50.0% to 100.0% while the recall score fluctuates between 66.7% and 100.0% for various types.

The single document extraction RAKE system (Rose *et al.*, 2010) is described as an unsupervised, domain-independent, and language-independent method for extracting keywords from individual documents which makes use of stop words to delimit either single-word or multiwords concepts. They use stop lists based on term frequency and keyword adjacency to boost the extraction power of their process. Applied to a set of abstracts from scientific articles, they achieve domain keyword extraction precision of 33.7% with a recall of 41.5% which amounts to 37.2% for the f-measure. Albeit the claim of language independence, languages like French, Spanish or Italian, makes extensive use of stop words in multiwords concepts which limits the keyword extraction potential of this system to single word concepts and truncated multiwords concepts. Some parallel research has been made precisely on multiword expression extraction in French because of the inherent difficulty to detect them (Green *et al.*, 2010).

1.1.2 Software engineering centered extraction systems

While these previous systems and methods are relevant for knowledge extraction from unstructured and semi-structured document sources, few research is made with these methods applied on the software engineering side. Nonetheless, some systems target natural language documents from software projects to extract domain knowledge and help generate software

artefacts or models. The document source are, for most research, software requirements documents which have been created previously by a human agent. These documents, compared to internal business document, feature limited and less ambiguous domain terminology, hand picked domain concept directly relevant to the software project and usually a more structured and hierarchical logical flow throughout the documents.

One of these systems have been designed to take in software requirements written in natural language to generate UML diagrams (Deeptimahanti and Sanyal, 2011) to cover the semi-automatic generation of class diagram, collaboration diagram and use-case diagram. The goal of their contribution is to limit the need of expert interaction and knowledge in the modeling process. Wordnet, anaphora resolution and sentence rewriting were applied to simplify the sentences in software requirements document to ease the downstream parsing. They then use sentence structures patterns to extract actors, classes, attributes, methods and associations; a sentence subjects are considered sender objects for collaboration diagrams, objects are the receivers, etc. While the system seems complete from the diagram generation point-of-view, no evaluation of the output potential performance was done on a gold standard.

Another approach, the Dowser tool of Popescu *et al.* (2008), applies knowledge extraction methods to reduce ambiguities in requirement specifications in order to help the review process of software requirement specification documents. It is designed to help the software expert detect inconsistencies and contradictions by automatically producing an object-oriented model from the natural language specifications. The system extract the domain specific terms by noun phrase chunking obtained after the execution of a part-of-speech tagging step. They consider that having a human agent in the loop for the model evaluation removes the need to obtain a high recall and precision from the extraction process. Nonetheless, they evaluate their approach on the intro man page of the Cygwin environment for domain specific term extraction and achieved a recall score of 78.8% on the exact document and a 88.46% score after rewriting some of the sentences with a precision of 86.79% which translate into a combined f-measure score of 87.61%. Two other use cases were used to test their system but no evaluation score were published.

Finally in Kof (2010), different term extraction heuristics are compared as part of the process of creating modeling message sequence charts in UML. They explore the influence of named entity recognition in isolation and with sentence structure extraction on two case studies detailing specifications of an instrument cluster from a car dashboard and a steam boiler controller. A precision and recall score of 95% was attained on the extraction of 20 concepts on the steam boiler specifications using named entity recognition, providing one wrong concepts and missing only one.

These combined research efforts presents the following limitations:

- Limitation 1: The methods are applied on specialized documents
- Limitation 2: Make use of many advanced linguistic tools, resources or human intervention
- Limitation 3: There is no annotated corpus publicly available to be used as a gold standard, so methods cannot be compared on the basis of their performance when applied on the same resource.

Our research differs from these previous systems on many aspects. First, the concept extraction process is applied on business documents which were written as part of the software project which increase considerably the extraction task. The current process is also evaluated using a gold standard targeted specially for knowledge extraction for software engineering. The evaluation is also done using the combined views of multiple experts instead of a single expert approach.

1.2 Multiword expression extraction

It is generally accepted that one's personal lexicon includes almost as many multiword concepts than single word concepts (Jackendoff, 1997). Multiword expressions (MWE) found in WordNet 1.7 compose 41% of the terms in the database (Fellbaum, 1998). And while many automatic term extraction techniques exist for single word terms, fewer explore multiword

expressions and very few study the very specific problem of complex multiword expressions (CMWE) in French (Green *et al.*, 2010). Still, a wide array of efforts is focused on MWE extraction. They can be split into either linguistic, statistical or hybrid approaches.

Linguistic techniques include the use shallow-parsers (Bourigault, 1992) based on syntactic information to extract noun phrases from text, the definition of patterns in HPSG grammar to identify structured MWE (Sag *et al.*, 2002) and the alignment of English and French terms (Daille *et al.*, 1994) with a statistical method to verify a cooccurrence of similar terms in the two languages. While this last method yields good results, it is not applicable in the context of this project because targeted corpora are only available in French and most complex multiword expressions are used by a single organization, so they cannot be found in another bilingual organization's corpora.

Statistical techniques can exploit various types of frequency analysis (Justeson and Katz, 1995; Enguehard and Pantera, 1994) or cooccurrence of expressions (Church and Gale, 1991) to find mostly fixed or semi-fixed MWEs in a corpus. Their incremental versions (Justeson and Katz, 1995) tend to bias extraction towards longer chains or provoke an overgeneration of terms without offering methods of elimination. Latent semantic analysis can be used (Baldwin *et al.*, 2003; Schone and Jurafsky, 2001) to make comparisons between the context of the MWE and its atomic parts using Wordnet as a base or to perform rescoring on MWEs extracted from a corpus.

Hybrid strategies make use of both statistic data and linguistic principles; bigram frequency analysis provided by syntactic parser (Goldman *et al.*, 2001), using the web as a lexical repository, etc. Part of the few researches on French MWEs in French, Green *et al.* (2010) uses the French Treebank (Abeillé *et al.*, 2003) functional annotations to instantiate parse trees using tree substitution grammars. Although they did not provide a per-length analysis, their overall performance improved by 36.4% compared to a surface statistics baseline. They test their baselines and algorithm using sentences with 40 words or less.

These methods, while offering good performances, cannot be applied to our research project as they have some shortcomings because of the following limitations :

- Limitation 1: Only target the lower spectrum of length for multiword expressions
- Limitation 2: Mostly restricted to English documents and expressions
- Limitation 3: Make use of heavy linguistics ressources (Wordnet, deep parsers, etc.)
- Limitation 4: Consider only short sentences

Our method is considered a hybrid one as it identifies CMWEs by solving frequency-based formulas derived from syntactic tokens and uses parts of the statistical local maxima method in the elimination phase.

1.3 Acronym detection for candidate

In the last decade, many studies have been carried out on acronym extraction from text. A fair share of them target corpora in the biomedical field, for which extraction performance is high. However, it seems that minimal effort, in terms of quantity of publications, has been expended on acronym extraction for other types of documents.

In many of the following studies, heuristics, strict rule definitions, scoring techniques, data mining, or a machine learning scheme based on linguistic or statistical data are used. Most also implement a “stop word” list to eliminate frequent problematic cases. A large number use the Medstract gold standard evaluation, which contains 173 SF-LF (short form-long form) pairs from 100 Medline abstracts.

The Three Letter Acronym system was introduced by Yeates (1999) for acronym extraction in a digital library context. Heuristics are used to verify that the all-uppercase SF (short form) can be associated with a matching nearby expression based on ratios derived from the potential

SF-LF association. A recall of 93% with a expected precision of 88% was reported in the article.

Larkey *et al.* (2000) extracted Web pages to feed an online acronym list with four heuristic-based methods: simple canonical, canonical, contextual, and canonical/contextual. The last of these yielded a recall of 88% at 92% precision for SFs 2 to 9 letters long and positioned at a maximum distance of 20 tokens from the corresponding LF (long form).

Pustejovsky *et al.* (2001) presented the Acromed system in 2001, which is based on regular expressions and syntactical data to associate LFs with candidate SFs. These associations are filtered with a formula which takes into account relevant words in the LF versus the SF's length. They achieve a recall of 72% with a precision of 98% on their own gold standard, which is based on abstracts extracted from the Medline database.

Park and Byrd (2001) softens the identification rules for acronyms by using linguistics hints and text markers, and also integrates the detection of acronyms containing digits. The right LF candidate is chosen according to the highest score based on extracted features like textual cures, rule priority, distance between the forms, etc. They used published technical texts, small books from the pharmaceutical and automotive engineering fields, and NASA press releases for evaluation purposes. The performance of the approach ranged from a recall of 93.95% and a precision of 96.9% on 32 acronyms in automotive engineering, to a recall of 95.2% at 100% precision on 63 acronyms from the pharmaceutical texts.

Chang *et al.* (2002) identified candidate SF-LF pairs with an alignment function, and evaluated them using logistical regression based on previously identified acronyms. Using the Medstract gold standard, they achieved a maximum of 84% recall at an 81% precision.

The Biotext algorithm developed by Schwartz and Hearst (2003) uses strictly defined rules to search for linked forms. It assumes that SFs and LFs are presented with either one in parentheses and without any offset between the two forms. They achieved a recall of 82% and

a precision of 96% on the Medstract gold standard evaluation corpus containing 168 SF-LF pairs.

Using patterns similar to Larkey *et al.* (2000), Zahariev (2004a) extracted acronyms from the initial paragraphs of 193 abstracts. These abstracts were taken from the Internet Engineering Task Force’s Request for Comments (RFC), published before November 2001, which targets a technical domain. Acronyms with numbers or symbols in the SF are ignored. This approach resulted in a recall of 93.01% with a precision of 95.52%.

Supervised learning was used by Nadeau and Turney (2010) with 17 attributes extracted from both the SF and the LF. They achieved a precision of 92.5% with an 84.4% recall using an SMO kernel-based SVM classifier on the Medstract gold standard corpus. Part-of-speech (POS) tagging data are also exploited in one of their rules for candidate space reduction heuristics. The article presents a clear summary of the major constraints and definition restrictions associated with the major publications in this field. Ni and Huang (2008) also used supervised learning to rank candidates. They used various models with a ranking SVM kernel on a randomly selected sample of TREC W3C documents.

Xu *et al.* (2009) recently presented their MBA system, which uses an alignment algorithm to define a score for each LF candidate. They use the Park and Byrd (2001) offset definition for the search space. The optimal LF is chosen based on the one with the highest score, minus some penalty points for missing words or non matching letters. With a user-defined cutoff score, they then split candidates according to their acronym or non acronym nature, and selected the final definition using statistical methods. The MBA system achieves a precision of 91% and a recall of 85% using the Medstract gold standard corpus. Using that document’s source type, Sohn *et al.* (2008) used a hard coded strategy ordering to find the best way to extract the correct LF for a candidate SF. A total of 1,250 biomedical samples (title and abstract) were used to evaluate their approach, with a recall of 83.2% and a precision of 96.5%.

When compared to the context of our research project, the previous methods shows the following limitations:

- Limitation 1: Trained and evaluated almost exclusively for the biomedical domain
- Limitation 2: Targets only English documents and expressions
- Limitation 3: The extraction methods are tailored for corpora composed of published scientific literature

Unfortunately, comparison between these studies can be difficult because they use different restrictions on SF length, search space length, and the definition of the acronym. Also, they do not always use common or published evaluation corpora. Many of the older studies do not take into account newer phenomena, which may not have been in common use at the time of the study. Our approach attempts to mix several successful techniques in new ways to help achieve sustainable performance on documents (mostly written in French) that differ in many aspects, as we will see later.

CHAPTER 2

ARTICLE I: CONCEPT EXTRACTION FROM BUSINESS DOCUMENTS FOR SOFTWARE ENGINEERING PROJECTS

Pierre André Ménard and Sylvie Ratté

Département de génie logiciel et des TI, École de Technologie Supérieure
1100 Notre-Dame Ouest, Montréal, Québec, Canada, H3C 1K3.

This chapter has been submitted for publication in the
Automated Software Engineering journal on April 1st, 2014.

Abstract

Acquiring relevant business concepts is a crucial first step for any software project for which the software experts are not also domain experts. The wealth of information buried in an enterprise written documentation is a precious source of concepts, relationships and attributes which can be used to model the enterprise's domain. Perusing manually through this type of resource can be a lengthy and costly process because of the lack of targeted extraction tool. We propose a domain model focused extraction process aimed at the rapid discovery of knowledge relevant to software expert. To avoid undesirable noise from high level linguistic tools, the process is mainly composed of positive and negative base filters which are less error prone and more robust. The extracted candidates are then reordered using a weight propagation algorithm based on structural hints from source documents. When tested on French text corpora from public organizations, our process performs 2.7 times better than a statistical baseline for relevant concept discovery. We introduce a new metric to assess the performance discovery speed of relevant concepts. We also present the annotation effort for a gold standard definition of software engineering oriented concepts for knowledge extraction tasks.

Keyword Concept extraction, Domain model, Relevance evaluation, Knowledge extraction

2.1 Introduction

Software experts starting a project in a new domain have to understand the inner working of their client's organization. To do so, they can proceed to interviews, workshops, reading written documents and studying work artifacts and on-site work activities (Pfleeger and Atlee, 2009). Estimation commonly accepted by the business and scientific community suggests that as many as 75% to 85% of organizational knowledge appears in unstructured written documents (Grimes, 2008). It would be a logical choice to exploit these valuable resources even if they cannot be used as the only source for domain knowledge elicitation. Alas, reading business documents and manually extracting knowledge from them is a long and strenuous process which is seldom used as an exploratory method in software projects.

Many issues may prevent software engineers from providing the needed level and quality of analysis at the start of a software project. Among them are such challenges as unfamiliarity with the client's domain, inaccessible or non-existent knowledge resources (work process description, ontologies, data dictionaries, etc.) about the domain or the client's business, non-cooperative or unavailable domain experts for relevant knowledge elicitation, insufficient budgeted time for the analysis phase and so on. The failure to produce a quality analysis may impact the following project's activities as it is a crucial element throughout the whole software development effort.

To better equip software engineers for this project step, we propose a text mining process to extract domain concepts from internal business documents. Because producing a highly structured domain model is a high level cognitive and semantic challenge, we propose a pragmatic, but still challenging, approach of creating an ordered concept list. We further suggest that this process should promote a high recall to avoid the need for a complete reading and verification of the targeted documentation. Even if not all the required knowledge can be found in written form or extracted from available documents, we hypothesize that our method could be used as a more productive and effective way to jump start an analysis phase than other more time consuming techniques as it can be done prior to the analysis phase without human intervention.

The motivation and use cases relating to the problem domain are presented in the remaining introduction sections. The next section discusses other knowledge extraction systems and methods published in the scientific literature. The third section presents an overview of the proposed method as well as the gold standard definition and the performance metrics to use in the current context. The fully detailed process is then explained in section 2.4 followed by the evaluation methodology, performance assessment and the discussion.

2.1.1 Motivation

The extracted knowledge available early in the project start up phases can be used to perfect the downstream stages and to validate the completeness of major analysis artefacts. While the natural way of thinking about domain model or knowledge is to materialize it in a conceptual model for the development phase of a software project, its potential impact spans, directly or indirectly, the complete software engineering process. Aside from the concepts that will be directly managed by the software solution outputted by the project, other knowledge may be relevant to learn about: unknown stakeholders, missing constraints in requirements, outside source of influence by regulating bodies on the supported process and so on. This idea is supported by the Swebok, or software engineering body of knowledge (Abran *et al.*, 2004), which specifies that “software is often required to support a business process, the selection of which may be conditioned by the structure, culture, and internal politics of the organization. The software engineer needs to be sensitive to these, since, in general, new software should not force unplanned changes on the business process.” So relevant knowledge in most software projects can also be found outside the traditional scope of “programmable concepts”.

The SWEBOK also corroborate the interest of mining the text corpora of an organization for knowledge as software engineers need “to infer tacit knowledge that the stakeholders do not articulate, assess the trade-offs that will be necessary between conflicting requirements, and, sometimes, to act as a “user” champion.” So it’s common knowledge in the software engineering domain that clients, stakeholders and end-users does not have, or share, all the information about their business, processes and their needs. In other research (Kotonya and Sommerville,

1998), domain knowledge understanding is seen as the first step of the requirements elicitation. Application domain, problem and business understanding are proposed as activities to adequately start a software project. Prieto-Diaz (1991) cites technical literature as the first source of domain knowledge along with existing applications, customer surveys, expert advice, current/future requirements as inputs to the domain analysis activity.

In some specific applications of software engineering, knowledge acquisition is also of utmost importance. Batini *et al.* (1992) report that conceptual modeling “is by far the most critical phase of database design and further development of database technology is not likely to change this situation” which is also supported by others like Borgida (2007).

For some, like Pressman (2001), “the goal of domain analysis is straightforward: to find or create those analysis classes and/or common functions and features that are broadly applicable, so that they may be reused.” On the other hand, some ¹ considers that domain analysis “involves modeling the domain so that software engineers and others stakeholders can better learn about it... not all domain classes necessarily result in the development of reusable classes”.

2.1.2 Context of application

For the vast majority, business management software in a company make use of at least one type of database in which data, either transactional or operational, is stocked to be used and analyzed by internal or external workers. These workers take part in internal processes which generate or access this data to realize the organization’s main or marginal activities. From this perspective, our extraction process is useful when the business workflow is part of the problem domain i.e. for a project that aims to solve a problem in the documented work process of the business. It wouldn’t be as useful to a software project which is developed for a new undocumented work process other than for the marginal interaction with existing processes.

¹Attributed to Timothy Lethbridge in personal communication on domain analysis, May 2003 from Pressman (2001)

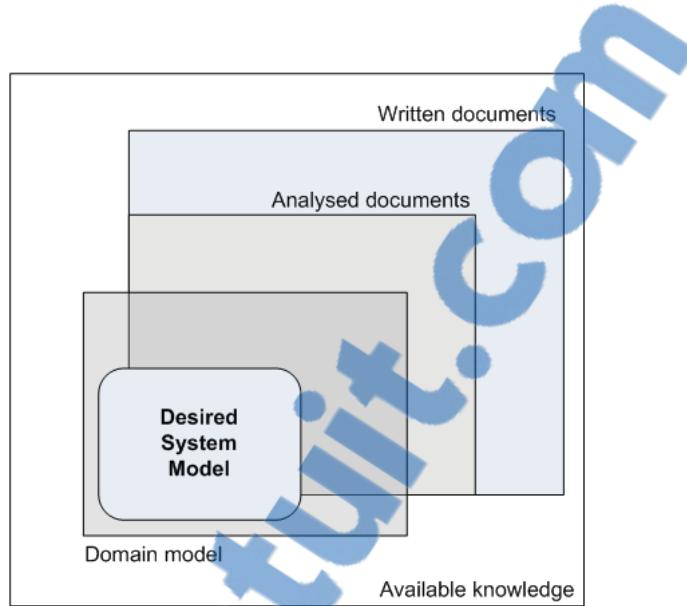


Figure 2.1 Scope and relation of knowledge containers

As shown in Figure 2.1, a part of all the knowledge used in a company is available in textual document format. From these documents, a smaller set must be analyzed to extract knowledge relevant to the domain model. The desired system model and the domain model are probably not fully documented either because of incomplete documentation effort or because the system is expected to use new concepts not readily implemented in the organization's work activities. Thus, the usefulness of the present process is dependent on the two following criteria :

- The amount of domain model knowledge contained within the analyzed corpus of document
- The amount of the desired system model described within the analyzed documents

Thus, the usefulness is optimal when these two criteria are at their best; when both the domain model and the desired system are fully described in the analyzed document. These criteria allow for an automatized text mining process to potentially extract all the relevant knowledge needed for the domain modeling phase. But as these criteria are seldom perfectly fulfilled in any chosen software engineering project, this approach should be viewed complementary to any existing knowledge extraction processes.

2.2 Automated concept extraction

This research project, Binocle, orchestrates the domain knowledge extraction effort on the concepts that are relevant to software engineering projects. The traditional way of doing text analysis for a software project consist in reading documents pertaining to the system goals to identify noteworthy information items, evaluating their relevance towards the task at hand and adding them, if needed, to the current knowledge model. To add them to the model, the human agent must check for redundancy (and possible polysemous terms), determine a role for the concept and so on. Small documents can be processed quite rapidly by experienced experts, but as they advance in the process, more validation must be done at each step to check the current model before integrating new knowledge.

This is evidently a highly cognitive task which relies on both the expertise of the agent with software engineering skills and his knowledge of the targeted organization domain. Early researches by Chen (1983) proposed a set of rules to facilitate the conversion of information requirements into entity-relationship models by the software experts. The first few rules are presented below:

- a. A common noun is an entity type
- b. A transitive verb is a relationship type
- c. An adjective is an entity's attribute
- d. An adverb is an attribute of a relationship
- e. “There are .. X in Y” means “Y has ... X”
- f. “The X of Y is Z” and Z proper, then X links Y and Z
- g. “The X of Y is Z” and Z common, then Z is a value of attribute X of concept Y

While some of these rules might be easy to follow by a human agent, their automation to produce significant output when applied on business document is not straightforward. Obviously,

business documents seldom contains convenient retrievable patterns as authors usually write in a free form style which does not ease the extraction of patterns. Sentences can be much longer and more complex than any foreseen simple patterns or express ideas in a completely different form, thus making the retrieval process tortuous and less efficient.

The current state of the art shows that many systems and methods exists to extract concepts from generic documents or from software requirements, but none establishes the direct link between business documents and software related concepts. As such, this knowledge modeling task suffers from the same hindrances as many other text mining tasks :

- Polysemous terms and expressions
- Multiword expressions extraction
- Named entity resolution
- Difficulty to evaluate relevance of concepts
- Anaphora resolution
- Domain knowledge identification
- Concept hierarchicalization

In addition to these concerning challenges, which are shared with tasks like automated ontology creation and automatic summarization, internal business documents also suffers from disrupted sentence model caused by a highly structured layout which contains many logical segmentation. A typical example is a starting sentence which introduce a list of items which are the ending part of the introductory sentence. Some proposed solutions to those challenges, like anaphora resolution, are seldom designed to cope with these type of cases. Other observations can be made on the level of linguistic quality of some non-published internal business document or the writing style of some authors which delight in excessively long and complex sentences. These phenomenon makes the use of advanced tools more problematic, or at least

less predictable in quality of output, for text analysis as they are typically designed and trained on simpler or higher quality document which have a flatter logical structure.

This section explains the global approach suggested to support the automation of concept extraction. We also present both a gold standard for the evaluation of the extraction pipeline and a measure to evaluate the quality of the reordering applied on the outputted list. These tools are usually required for consistent benchmark so that further researches can be evaluated on solid grounds.

2.3 Proposed approach

In this research, we propose to extract concepts from business documents which are valid and relevant for software engineering experts using a pipeline of positive and negative low-level filters. Our hypothesis is two-fold in the context of the targeted goal of knowledge discovery for software engineering projects. First, we want to verify that filters can effectively be used to narrow down a list of candidate concepts which can be relevant to experts. Secondly, we hypothesize that the logical structure of business documents can be used as hints from the authors to reorder the previous unordered concept enumeration in a way that enables the experts to learn relevant concepts when presented with a noisy list.

The pipeline is designed to be applied on a very noisy lists of concepts produced by a generative process that focus on recall. This type of process creates a wide selection of potentially useful concepts as opposed to a precision-focused process which would try to output a narrow list of concepts with high confidence regarding their usefulness. This approach is also used to avoid applying more complex linguistic analysis tools that would raise the noise level when documents are not adapted to some of their needs which would degrade the performances of the next steps. For example, a like deep sentence structure analyzer might need short well-formed sentences to produce a good analysis, so applying it on long and complex sentences might induce noise caused by incoherent output of this tool. While more complex tools, like deep sentence structure analyzer, have been developed to very different levels for various languages,

less complex filters seems to be a promising path especially for non-English languages as they can produce good results with low requirements on the input documents. Thus, the pipeline should be an assembly of lower level (if possible) modules that can be viewed as positive or negative filter. The role of negative filters is to dismiss irrelevant or invalid expressions that should not make it to the final list. On the opposite side, the positive filters aim to detect and certify relevant concepts on solid assertions. These positive validation modules were added to the pipeline to improve the detection of specialized linguistic phenomena (like acronym resolution) and limit the overproduction of candidate concepts by detecting a wider array of multiword expressions. As some of the concepts can be very long and sometimes contain smaller one, their detection enables to validate the larger expression over some of the shorter overproduced and irrelevant ones.

We apply the filtering pipeline on a per-document basis instead of a corpus like statistic-based term extraction methods. Consolidating extracted lists from different documents might provide a broader contextual view of the problem at hand, but some relevant concepts might be missed because aggregation and relevance are typically done with statistical methods (term frequency, inverse document frequency, etc) that may diminish their importance. The definition of robustness is often associated with precision-focused because the goal is usually to reuse the output of the processing of the developed technique, as integrating a noise filter in downstream module can be complicated. While we agree with these definitions, we argue in the context of this research that a recall-focused robustness is more relevant. This is mainly because the effort of looking for missing concepts throughout the documentation far exceed the one needed by a human agent to filter out irrelevant concepts from a noisy list of expressions.

Because a high recall output list will contain many irrelevant concepts, a neutral (in terms of concepts filtering) structure detection step was added as the last module to enable the final re-ordering technique. The goal, as mentioned earlier, is to help the experts learn the relevant one and thus lower the cognitive effort needed for the modeling task. It exploits the document's logical structures which is commonly used in internal business documents to divide topics, detail specific pieces of information, decompose complex or composite concepts, etc. As such, it

was considered a good basis for transmitting concepts of importance from the business domain to the software project activities.

2.3.1 Gold standard definition

While the relevance to a specific domain knowledge item can be considered a matter of personal opinion, some patterns may emerge from the combined modeling effort of many experts. It thus can be considered a viable gold standard to evaluate a system's performance against such combined expert opinions. But as research has demonstrated, when interacting with an information retrieval system, a searcher's judgments of relevance may not match system determinations of relevance (Anderson, 2006; Schamber, 1994). Taking these observations into consideration, a multiple annotator relevance standard needed to be used to better evaluate the performance of the proposed method while enabling some flexibility in the process. As no gold corpus for manual conceptual modeling from textual sources was available as a comparison source, a group of software experts was polled to create this type of useful reference.

Creating models of knowledge from text documents is a strenuous task, like other textual-based tasks like summary creation. Campaigns of summaries evaluation, like Document Understanding conference 2005², uses 4 annotators to create the reference gold corpus by which every system is scored. We view the selection of concepts (the first step of modeling) as being similar to document summarization; annotators must select units of knowledge which they consider relevant, each unit might be found in different form (noun, adjective, adverb, etc.) in the source documents, only the knowledge and expressions found in the text are to be used in the summary, no one summary can be deemed the best over the others but some quality classification can be made nonetheless.

We used a multiple annotators approach similar to other corpus building tasks in natural language processing projects. We employed eight experts which processed manually each of the document from the two corpora. Each of these expert were from different backgrounds

²http://www-nlpir.nist.gov/projects/duc/data/2005_data.html

of software development environments like banking, industrial, research, telecommunication, etc. They also had various levels of experience, from two to fifteen years, to provide different views of the task. These selection features for the annotators were made to insure that no specialization effect would take place in the experiment. Some gold standard definition efforts, like Abeillé *et al.* (2003), split their corpus and give each part to two annotators who check them sequentially. They used a group of fifteen experts to act as annotators which consulted in group to keep the annotations consistent. For our research, we kept annotators separated from others to avoid cross contamination of the produced artefacts as the personal view was more important than a shared definition of relevance.

Each expert was given five documents in raw text format; only line skip and indentation were retained while text formatting features like underlining, bold, italic or variable font size were removed. This was done to avoid hints given by the document to influence the expert. The instructions given to the experts before the start of the annotation phase to standardize the task were the following:

- Identify which information you consider relevant to know before starting a development project that will create a software system to manage the workflow and data described by the provided internal business documents.
- Relevant concepts can be identified as such in one of the following formats:
 - Annotated directly in the text
 - Transcribed onto a separate list
 - Modeled as a UML class diagram
- Annotate each document in isolation from other documents, which means that similar concepts must be annotated each time they are encountered if considered relevant in more than one document.
- If not needed by the format, the type of concept (like class, attribute, role) can be unspecified.

- If using UML or list representation, concepts' names should be the same as in the documents. This means that generalization should only be noted if presented as such in the text.
- Documents can be annotated in any order.

Experts were given no hint or clue on how the hypothetical system should be built or which part or the processes or concepts it would manage. They only used their past experiences and knowledge of software development and software engineering project management to evaluate the value of each concept found in the documents. The training factor could not be considered in this case as the skills and knowledge to perform the task needed to be acquired beforehand. This is different from many researches in which annotators showed less errors as they progress through the annotation effort.

Table 2.1 Summary of the gold standard definition effort by number of selected relevant concepts per annotator

ID	Representation	Doc #1	Doc #2	Doc #3	Doc #4	Doc #5	Total
1	List	29	88	76	51	26	270
2	UML	28	60	63	40	14	205
3	List	30	82	78	47	29	266
4	Text annotation	28	83	86	62	45	304
5	List	27	82	84	68	41	302
6	UML	34	79	68	46	17	244
7	Text annotation	28	74	85	64	42	293
8	Text annotation	36	83	80	63	36	298
	Average:	30.0	78.9	77.5	55.1	31.3	272.75
	Standard deviation:	3.3	8.6	8.3	10.3	11.7	34.5

The evaluation corpus consisted of 517 sentences totaling 12,444 words from two different public organizations, one of academic nature and the other from the governmental public works department. Each of these sentences were processed manually by the annotators using a method of their choice. Two of them created UML representations of the concepts, two generated a list of concepts, and the rest annotated the relevant expressions directly in each document.

All sentences were unique so that no section was repeated in any documents from the same corpus. This was not a restriction on the choice of the documents, but it enabled a larger coverage of relevant concepts for the same number of sentences (i.e. using documents with a lot of repeated sentences may have yielded a lower number of relevant concepts). The annotations effort yielded a total of 365 unique terms and expressions considered relevant concepts by different numbers of annotators.

As shown in Table 2.1, the annotators that chose to use the UML diagram representation generally selected fewer relevant expressions than any other methods, while the text annotation method seems to be on the higher end of the spectrum. One of the reasons may be that putting concepts in relation with each other changes the perception of relevance during the process. New expressions must be linked to existing ones in the diagram so that the final product is consistent. Nonetheless, the selected concepts from these annotators were mainly shared with others using other representational methods.

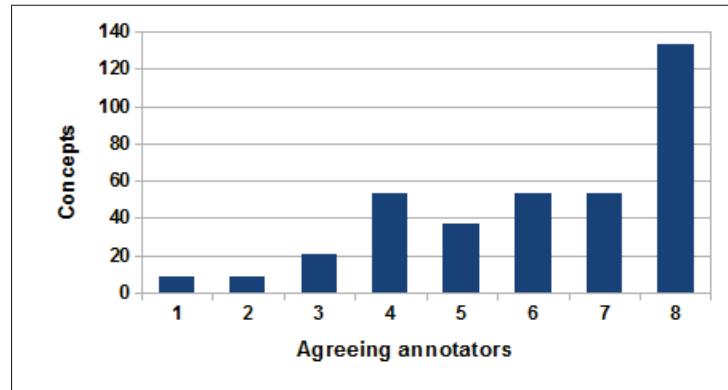


Figure 2.2 Distribution of concepts per number of annotators

Our annotated corpus is comparable in term of the number of processed sentences as well to the amount of extracted items. For example, Leroy *et al.* (2003) employed three experts to verify 26 abstracts for a total effort of 237 sentences in a relation extraction task for cancer research in biomedical texts. 330 relations were extracted by the experts during the gold standard creation. The higher number of annotators employed in our effort enables better validation of

the selected concepts as well as a wider take on the divergence of the relevance definition from the annotators.

The distribution of concepts selected by a specific number of annotators is detailed in Figure 2.2, which shows that the number of commonly agreed concepts are relatively high. For example, twenty concepts were chosen by exactly three annotators, but these annotators were not the same for all twenty concepts ³. More than a third of all the annotated concepts, 133 out of 365, were agreed upon by all the eight annotators and large majority (89.4%) of all the concepts were selected by at least half the experts. While no two concept selections were identical between any experts, these numbers show that there was a relatively strong agreement between them. This confirms that a common view, however flexible, of relevance is shared among software engineering experts which enables a gold standard to be used as a benchmark for concept extraction systems.

2.3.2 Evaluation metrics

In the context of a composite knowledge extraction process, we used two different methods to evaluate performances; one for the candidate concepts extraction task and another for the evaluation of the reordering stage aimed at optimizing knowledge exploration for the modelers.

The first evaluation metric is used to evaluate the raw potential of the presented process chain to extract candidate concepts and remove irrelevant one while keeping the relevant terms and expressions. The F-measure (or F1) is used for this intermediate point in the process as it gives a joint view of the recall and precision potential of a system. It is calculated as the harmonic mean of the recall and precision so that $F1=2*(R*P)/(R+P)$. In this context, the complete list of all concepts annotated by the experts serves as the aim for a ideal recall score. The precision is thus the ratio between the number of elements from that list and the total number of the extracted candidates.

³The detailed selection for each annotators can be consulted on <https://sites.google.com/a/etsmtl.net/lincs-pa/ressources/gold-corpus>

Of course, a perfect score on both recall and precision in this case would not mean a perfect score regarding any individual annotator, except if the list of one of them would span the complete set of concepts of all the others, which is unlikely. It would rather be considered perfect from the point of view of the next step of reordering the concept list. A low F-measure would either mean a low precision or low recall (or both) which in turn would translate respectively as a higher challenge to place the top relevant concepts at the beginning of the list or as an impossibility to do so if the top concepts were not available for the reordering step.

The second metric was defined in order to evaluate if the reordered concept list would be relevant to an expert for the analysis step of a software project. This metric was named Mean Discovery Ratio (MDR) as it approximates the average rate at which an expert (or any other human or non-human agent) reading the ordered list would learn relevant concepts extracted from a specific document. As we viewed the extraction of domain concepts from documents as similar to a text summarization task, this metric is inspired by the Pyramid method (Nenkova and Pas-sonneau, 2004) for evaluating an automated summarization system against a gold standard defined by multiple experts.

Table 2.2 MDR calculation example

Position :	1	2	3	4	5	6
Ranked Gold	4	4	3	2	2	1
Cumulative Gold	4	8	11	13	15	16
Ranked System	2	0	4	3	1	0
Cumulative System	2	2	6	9	10	10
Ratio CS(CG	0.5	0.25	0.545	0.692	0.667	0.625

As in the Pyramid evaluation method, the first step is to calculate the weight of each concept annotated by the experts. Each candidate concept gains one unit of weight for each annotator which tagged it as being relevant. In this context, the candidates can be either a concept or an attribute, if annotated as such by any expert. This is the grounding step for relevance as we consider that a higher weight provides an increased knowledge value for domain modeling task than a low weight candidate. So for a document annotated by N experts, the maximum

weight for any given concept would be N and values would decrease to 1 for concepts that were selected only by one expert. The resulting concept-weight pairs are then ranked in a decreasing order of weight which gives the “ranked gold” standard for the performance metric. The ranked gold standard is viewed as the best learning value a given system could provide to an expert to enable him to quickly learn about the most relevant concepts pertaining to the domain at hand as explained in a specific document.

Once the ranked gold standard is available for a given text, the unordered output of a system, for the same text segment, can be tagged with the corresponding weights. It is important to note that neither concept list, the gold standard or the system output, are in the reading order of the documents, other than by mere chance. Applying the weights from the gold standard to the corresponding text provides an ordered list where each concept also found in the ranked gold have the same weight. The remaining concepts have a weight value of zero. Both lists are then aligned as depicted in the example in Table 2.2 with the lines titled “Ranked Gold” (RG) and “Ranked System” (RS).

$$MDR_N = \frac{1}{n} \sum \frac{CSW_i}{CGW_i} \quad (2.1)$$

As shown in equation 2.1, the MDR of a system is then calculated as the average of the cumulative system weight (CSW) divided by the cumulative gold weight (CGW) at each of the i th position in the interval $[1, N]$, N being the total number of concepts in the ranked gold standard. The cumulative values of the example is shown in the “Cumulative Gold” and “Cumulative System” lines in Table 2.2 and drawn visually in Figure 2.3. The ranked gold line should practically always drawn like an asymptotic curve because the added weights diminish as the number of added concept increases. Only in the unlikely event that every expert would create the exact same annotation list would the gold be drawn as a straight positively sloped line. The ratio line in the same table show the performance value of the accumulated relevant concept for each new concept position. For the presented theoretical example, the MDR of the system would be approximately 0.547 compared to the ranked gold.

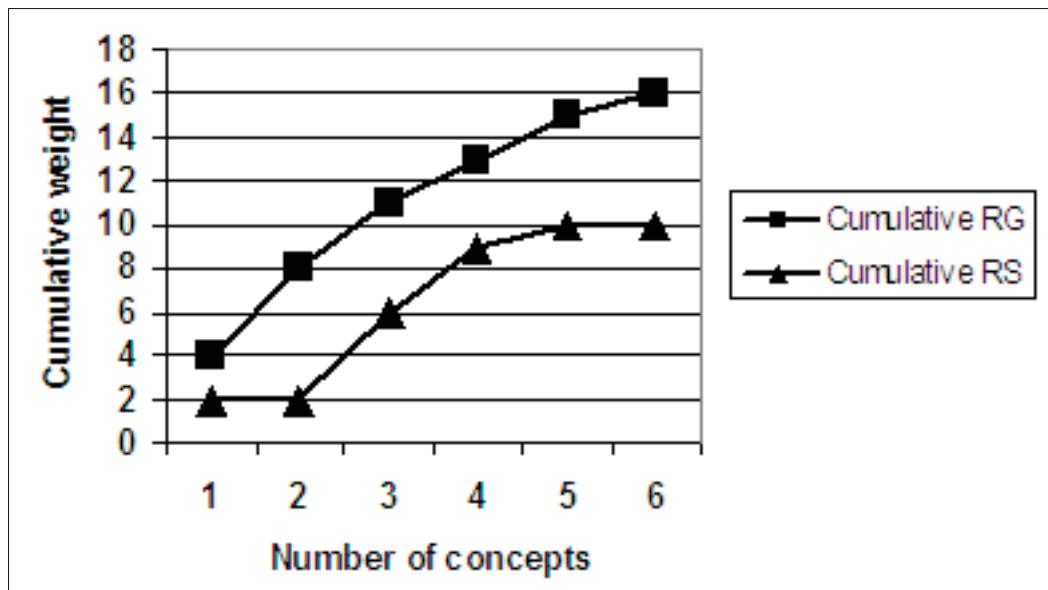


Figure 2.3 Cumulative gold and system from Table 2.2

Following the formula, the MDR can take on values in a normalized interval of $[0,1]$. A MDR value of zero indicates that the studied system could not present any valuable concepts within the range of the ranked gold span. Evidently, a value of 1 illustrates the capability of the system to not only put all the relevant concepts within that boundary of the ranked gold span, but also put the most relevant at the top of the list.

To show that the MDR is a metric which consider both the completeness of the concept list and its order, another theoretical example can be built with a system providing the complete set of concepts but in the exact opposite ranking. The Figure 2.4 shows a ranked gold standard curve (the top one) of a theoretical list of ten concepts created by ten annotators and in which every concepts have been annotated by a different number of annotators so that the ranked gold weights are 10, 9, 8, 7 and so on. The ranked gold system is presented as the bottom line in which all concepts from the gold are present but in a increasing order. The final MDR of this system would only be 0.471 even if all the concepts are within the gold boundary.

Finally, it is important to mention that the MDR metric can be measured on any number of concepts. As some systems would probably provide a greater number of concepts, because of the noise, than in the gold standard, the cumulative weight of the ranked gold should be

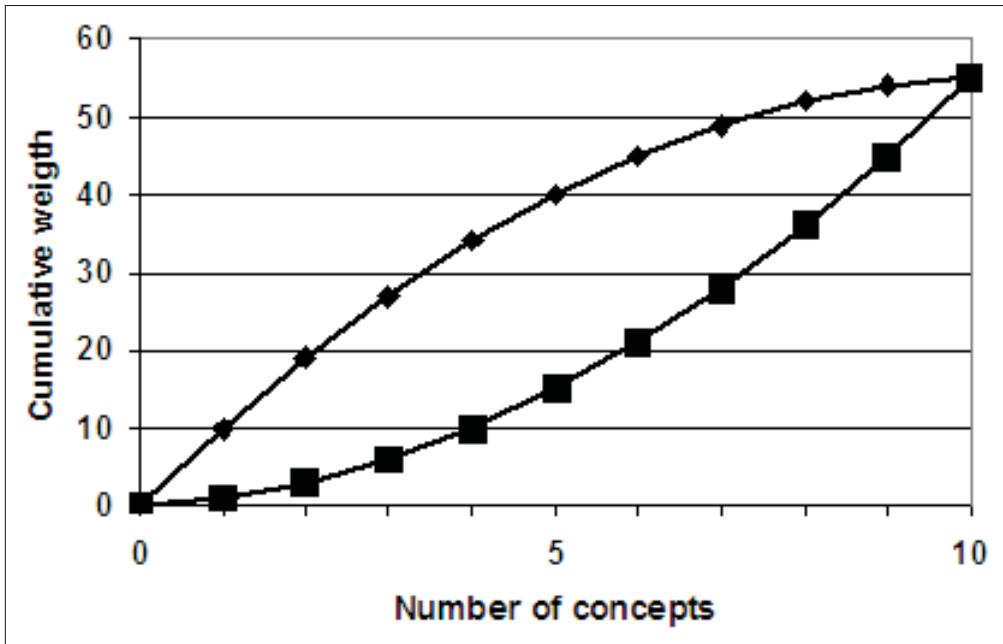


Figure 2.4 Monotonic cumulative gold and reverse cumulative system

stretched to match the number of concepts from the system's list. This would create a flat line at level of the highest weight of the cumulative gold and the compared system would then add new concepts (and hopefully weights) to bridge the gap to the non-increasing gold roof. Although it can be measured in this way, we consider that measuring the MDR at the gold standard's end gives a better estimation of a system's extraction potential, especially when compared to a baseline or other systems applied to the same document. The main issue is that this metric does not take into account the length of the list, so that system providing too many concepts would be at an advantage compared to good but conservative ones. Thus the MDR calculated at the gold's end gives each system an equal chance.

2.4 Detailed extraction process

2.4.1 Implementation overview

As a high recall extraction method is needed to cover as much scope as it is relevant to do so, a loose and broad extraction method was designed based on a pipeline of positive and negative filters. The terms and expressions of this list are called candidate concepts.

The high-recall/low precision list of extracted candidate concepts is then ranked using document and semantic related weights to enable a software engineering oriented exploration of the concept list.

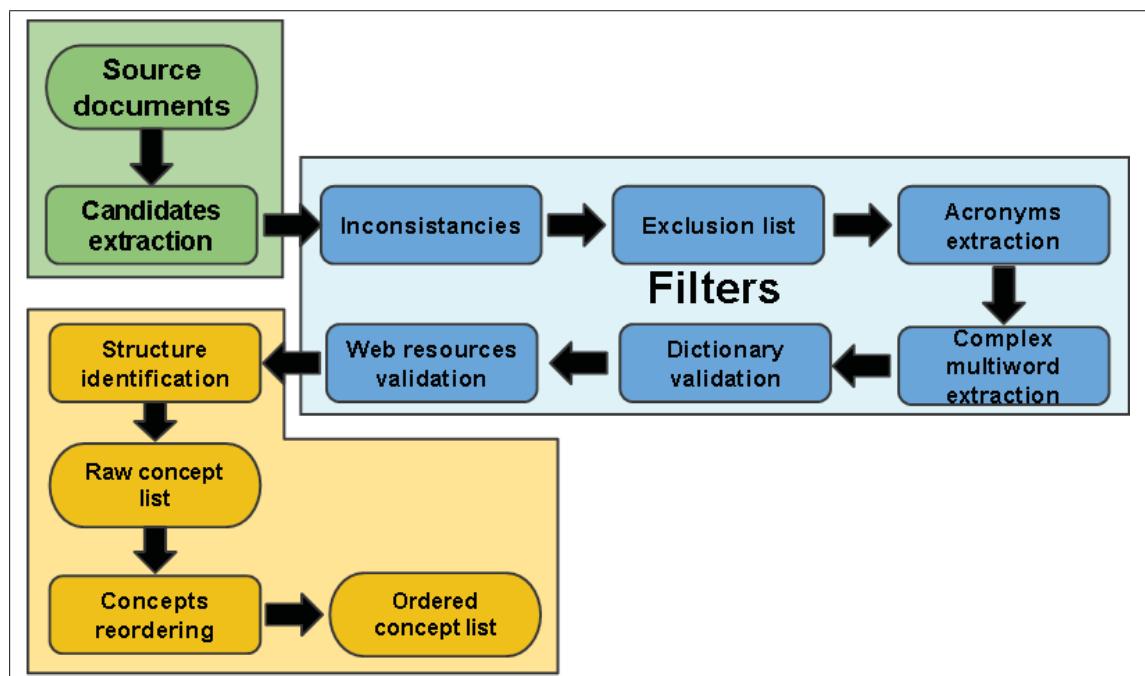


Figure 2.5 Modular composition of the extraction process

All the modules that needed training or textual evidences from document sources were applied on a separate development corpus. It was taken from a different public organization than those of the introduced gold standard and was hand annotated by the authors. Nonetheless, the

attributes of the documents (semi-structured, type and depth of logical document structure, level of language, etc.) were similar to those of the gold standard.

At the end of each module's execution, a cleanup sub step takes place to certify or remove concepts or occurrences of concepts before starting the next module. It was done to avoid collision between modules as some may have contradictory decision. They instead make conservative assumption and only certify or remove the items they are confident about. In the case of positive detection module, they will only approve expression validated during their processing but will not remove the ones for which they found no information. So for a case where a concept is represented in the text by three occurrences, if a negative module eliminates all occurrences on the basis that they are part of common irrelevant expressions, no other downstream module will be able to process them. On the other hand, if one of the occurrence was detected to be highly relevant by a positive module, the two others would have made it to the next step to be processed (certified, removed or ignored) by the others modules.

The final output of the pipeline is an unordered list of certified candidate concepts, called the certified list, and a list of associations between occurrences of certified expressions and occurrences of logical structures of documents.

2.4.2 Candidate extraction

The first module is not a filter but a set of tools built to extract the candidate concepts. Using the Gate (Cunningham *et al.*, 2011) platform, a short pipeline was created to take each document and apply the following tools: a French tokenizer, a sentence splitter, a part-of-speech tagger named TreeTagger (Schmid, 1994) with the basic French language model and a gender and number annotator for noun, adjectives and verbs. An expression extractor was then applied to export the occurrence of syntactic patterns like:

- noun
- noun + noun

- noun + det + noun
- noun + det + noun + adj
- noun + verb
- adj + noun
- noun + adj + det + noun
- noun + adj
- adj + noun + adj

where “det” is any single or group of determinants and articles and “adj” is any adjective or set of adjectives. The “verb” token denotes present or past participle. The linguistic information (part-of-speech tag, gender, number, lemma, etc) about each part of an expression was exported to be available to any module in the presented process.

As the previous patterns only covers standard expressions found in documents, another parallel extraction process was necessary to export longer and more complex expressions. In fact, artefacts, departments, external regulatory bodies or items for which a specific name have been given are often expressed using longer than usual expressions which falls outside the above-mentioned patterns. They sometimes can be viewed as domain-specific named entity. As French language uses determinant and articles to separate most nouns, it can produce very long expressions like “Loi sur l'accès aux documents des organismes publics et sur la protection des renseignements personnels du Québec” (“An act respecting access to documents held by public bodies and the protection of personal information”) which is 18 words long.

This parallel extraction process generates any expression longer than the four unit accessible from the basic pattern for which a frequency of two or more was found. The process starts at five units and increments gradually the extraction length by one word until no repeated expression is found for the next size. These expressions are limited in size by the sentence

boundaries and conjugated verbs except for participles or infinitives. Incomplete expressions starting or ending with a determinant, conjunction or article (e.g. "of immigration and border protection", "department of immigration and") were removed, as a shorter version of the sub-expression would already be available. The frequency of each expression was retained as supplementary information.

2.4.3 Inconsistency filter

This module targets the instances detected by some of the basic patterns and inspects the linguistic information associated with each word to see if any error could be used to dismiss the expression. The patterns are typically those including adjectives or participles (adj + noun, noun + adj, adj + noun + adj, etc), as they do not necessarily agree with the previous contiguous noun as they can be related to another noun earlier in a sentence. It checks if the dependent tokens (adjective and some participles) respect the language rules in conjunction with the head noun. For example, in a noun+adj pattern, the gender and number of both words would be checked for consistency; if the information doesn't match, like "sections accessible" ("accessible sections" where the adjective number doesn't match with the noun), all the occurrences matching this expression are removed from the list. This would be the case if a plural noun was paired with a singular adjective, as the adjective takes on a plural inflection in French. No action is taken for ambiguous dependencies, like adjectives which are written the same way in either gender.

2.4.4 Stop list exclusion

Instead of the usual static list of words and expressions used by most researches, a generalized expression-based grammar was implemented. It contains mostly generic and commonplace verbal locutions or semi-fixed expressions which are not domain related. I.e. "as a matter of fact", "ruled out", etc. They were taken from public sources like Wikipedia, Wiktionary and other online resources.

As many of these expressions can be used at various tenses or modified by adverbs or adjectives (e.g. “passer un très mauvais quart d’heure” (having a very rough time), “porter rapidement plainte” (to rapidly lodge a complain), “porter sa lourde croix” (to carry his heavy cross), etc.), defining an exhaustive list of all the variations was ruled out. Instead, each part (verb, noun, determinant, etc) of the verbal locutions were generalized using their lemma. Between each of these lemmatized parts, a placeholder span was added to account for potential modifiers (adverbs, adjectives, etc) as well as non-modifying elements like appositions and other punctuations. Each span can accommodate from zero to three elements which are not explicitly a part of the original expression. Table 2.3 shows the lemmatized version of a sample verbal locution and lists some of the potentially detected modified expressions.

Table 2.3 Generalized exclusion example

Source expression :	gagner du terrain (to gain ground)
Generalized grammar :	[verb.infinitive=gagner] [placeholders 0-3] [det] [placeholders 0-3] [noun=terrain] ([verb.infinitive=gain] [placeholders 0-3] [det] [placeholders 0-3] [noun=ground])
Detected examples :	gagner du terrain (to gain ground) gagner, comme précédemment, du terrain (gaining, as before, ground) gagnait rapidement du terrain (quickly gaining ground) gagnant ainsi un peu de terrain (gaining as such some ground)

The current list⁴ used in this project contains 6,030 French expressions taken from publicly available common stop list expressions. As previously explained, each item in this list can potentially detect many different variations of verbal locutions and commonplace expressions because of the flexible patterns approach. While it is not possible to give an exact number, it is plausible to assume that this technique can potentially detect a greater number of semi-flexible stop expressions compared to the usual method of using a list of fixed expressions. Any occurrence of an expression that matched one of the variation of these patterns was thus

⁴The complete list can be found at the following address: <https://sites.google.com/a/etsmtl.net/lincs-pa/ressources/stolist-locution>

removed from the candidate list. Longer expressions which contained one of these patterns were also removed from further processing.

2.4.5 Acronym resolution

Performing acronym detection and extraction on business documents present different types of challenge than for more studied sources like abstract or full text from scientific articles in biology, medicine, chemistry, etc. The main issue in this case is the implicit use of the short form of acronyms without explicitly linking it to the expanded form as in other types of document used in researches. This may be due to a looser review process prior to publication or the assumption that the target reader have prior knowledge to deduce the link. This results in the alternative use of both forms without a clear way to link them together. Of course, the use of acronym is different from one organization to another. Some may only use a small amount external and well-known acronym (USA, UN, FBI, CIA, etc.) while others may make heavy use of internally created acronyms to denote many types of concepts such as employee's role, document type, equipment, system, tool and so on. State of the art algorithm designed for other types of source document (abstract or full text from scientific articles in biology, medicine, chemistry, etc.) perform modestly on business documents as they usually contain an undefined mix of explicit and implicit acronym use.

To perform the extraction for this study, we used a classifier-based technique applied to candidates found in localized search scope Ménard and Ratté (2010) which focused on business document like those used in the current research. In overview, the process starts by identifying short form in a part-of-speech tagged text and generating potential matches from the N preceding and following sentences. For each of these candidates extracted from each localized search space, a set of features are extracted. Three structural features, potential score, level and letter's triviality concerns the link between the two forms. Three similarity features, ratio of function words, ratio of content words and percentage of coverage, tries to discriminate the frequency of letters, and finally a distance which indicates the absolute number of tokens between the

short form and the candidate. These features are then used to train a model to classify new and unencountered instances.

The training set was extracted from documents obtained from diverse publicly accessible government web sites. No acronym in the training set were in common with the ones used in the targeted documents. Their original experiment provided a final F-measure of 83.72% compared to two baselines of 61.83% and 62.91% on the business corpus. The order of usefulness of the features for the classifier was the potential score, relevant words, coverage, distance, function words and triviality level.

The details the algorithm and experiments for this module is explained in-depth in Chapter 4.

2.4.6 Complex multiword expressions detection

During the candidate extraction phase (first step of the process), expressions that fell outside the reach of the basic patterns (see Section 2.4.2) in size and complexity were extracted using an incremental generative approach. This extraction method overgenerates a lot of intermediate, irrelevant and incomplete expressions before producing the correct one. For example, a valid ten words long expression would also produce two expressions of nine words, three of eight word, four of seven words and so on. These expressions are thus eliminated with a statistical approach. It works by creating a tree starting with one of the longest expression available. The tree is constructed with all the contained expressions, with the frequency information added at every node in the tree, in such way that each level of the tree contains expression one unit shorter than the upper level. Each node is connected with each containing parent so that almost all nodes except for the root are linked to two parents. For example, the expression “B C D” would be linked to upper level with “A B C D” and “B C D E”. The tree is then explored to check, for each node, if a significant increase of frequency is detected when compared to both of his parents. The minimal growth ratio of frequency is optimized pragmatically, so a 30% threshold was used to eliminate non significant expressions. If any node is still available at the end of the process, the highest ranking expression, in term of length, is elected valid.

The process of tree building and node elimination then starts over with the remaining longest expressions. Expressions used in a previous tree, either as root or sub-level, are removed from the potential root expressions. Once all the expressions are processed and a list of complex multiword expressions is available, the module scans the concept's occurrences from both the long expression list and the basic pattern extraction list and remove any sub-length expressions. For example, the complex multiword expression “A B C D E F” would eliminate an occurrence of “B C D E” only if contained in the longer expression. Other occurrences of “B C D” found in the text but not part of the longer expression would be untouched by this module. This step helps to reduce a large amount of the overgenerated candidates in both extraction processes.

This filter is explained more precisely in Chapter 3 of this thesis.

2.4.7 Dictionary validation

While filters like acronym resolution and complex multiword expression detection can help detect domain-specific long multiword expressions in a corpus, existing static resources like dictionaries or high-level ontologies can provide approved commonplace expressions which may span multiple domains. These resources typically contain single word terms and short length multiword expressions which does not overlap those found by the two previously mentioned filters. The presence of a term or an expression in these resources does not mean that it is relevant as part of the domain knowledge. Nonetheless, these multiword expressions can be used to better define the candidates in the output list, using it as a positive filter to approve matching expressions.

A typical example would be “produit à valeur ajoutée” (value-added product) which can be found in businesses from any field of activity for which products are created or sold. As for the two previous filters, these expressions are used to eliminates sub-length occurrences, like “value-added” in the previous example, when it was used as part of the full expression. Of course, for this filter, it was assumed that no resources of those types were available for the domain covered by the corpus used in this research.

Table 2.4 Number of noun-based multiword expressions by length in the DELA

Length	Count	Ratio
2	56395	62%
3	28687	31.5%
4	4516	5%
5	1031	1.1%
6	286	0.3%
7	85	< 0.1%
8	21	< 0.1%
9	6	< 0.1%
10	11	< 0.1%
Total	91038	100%

As our research was done on a French corpus, the DELA French dictionary of inflected forms⁵ was used. In its original form, it contains 683,824 inflected single word terms based on 102,073 different lemmas, as well as 108,436 multiword expressions based on 83,604 different lemmas. It contains named entities, like “Blanche Neige et les sept nains” (“Snow White and the Seven Dwarfs”) as well as common multiword expressions like “président de la chambre criminelle de la cour de cassation” (president of the criminal division of the Court of Cassation”). For this filter, only the multiword expressions used as noun phrases were used from this dictionary. All other types of expressions or terms (verbal phrases, adverbs, etc.) were removed. As shown in Table 2.4, the contribution of this resource for multiword expressions is greater for the shorter expressions, between two and four words long, which make up approximately 98.5% of all expressions, than for longer expressions above 5 words which compose the remaining 1.5% of the list.

2.4.8 Web validation

On the same theme of shared knowledge as the previous module, the web validation module detects multiword expressions using internet search resources. This gives access to unstructured knowledge which is not available in traditional and maintained resources like dictionaries, on-

⁵Dictionnaire électronique du LADL: <http://infolingu.univ-mlv.fr/>

tologies or taxonomies. For any given expression composed of two words or more, it launches a search on various search engines to extract the titles of the top 20 returned pages. It then checks if any of these titles match the queried expression based on these criteria:

- The title match exactly the expression
- The title match the expression if the lemmatized form of the starting or ending word is used
- The expression is found between non-character separators

The separators from the last criteria can be the title's starting or ending positions, parenthesis, a dash, a comma, a vertical pipe, etc. This criteria helps validates expressions found in titles like "Value-added product - Definition". Such expressions are then used to remove spanned sub-expressions in the documents. For consistency of the research methodology, all web sites from the top-level domain of any organization for which we used the documents in the development or evaluation corpora were removed from the returned pages before the criteria-based search was executed. It was also done to check if the expression was really shared with other domains of activity. A multiword expression found in many other domain would improve the confidence of its relevance.

2.4.9 Structure detection and analysis

A final neutral module was added to analyze documents to detect specific structures and identify their type. Its position in the pipeline is not critical as it only adds information for the re-ordering step. This detection is based on the hypothesis that authors uses the logical structures of business documents to convey hints about relevant concepts from their domain of activity. One of the difficulty of this step is that most language analysis tools remove all document formatting (font emphasis like italic or bold, font size, format type, non-tabular indentation, bullet type, etc) when extracting text for processing. As such, the structure detection take place of raw text files without any formatting, which is an added difficulty.

On a test sample from a separate experiment on 6,976 sentences taken from various types of business documents (tables of content, indexes, forms and highly structured documents were excluded), over 45% of the sentences were part of either a title, a subtitle or a list. This high ratio indicates that while business documents are not considered structured data, like a database; the ideas and concepts they express can be highly organized. But all these sentences present a challenge from a text processing's point of view, mostly because they cannot be parsed and analyzed like a complete sentence. This disrupted sentence model is not taken into consideration by many complex linguistic analysis tool, which is one of the main reason for their lacking presence in the presented process.

Many textual and non-textual artifacts can be found in free text documents. Aside from images and graphs, which are not suited for text mining, tables and lists can be used to organize relevant concepts. While tables may sometimes be similar to a dictionary presentation format, lists are more closely related to free text. They can be used at the top level of the document, to define titles' and sections' order and depth, or in a more local fashion to emphasize relevant items. The Table 2.5 illustrates some generic list types that can be encountered in real-life documents like those found in our development corpus. While the simple structure is the most widely found, many documents used mixed indices at different level and a few used the interleaved structures. The interleaved format can be most problematic when trying to associate items from the same level to analysis logical series of knowledge within a document. These formats can be found at the section-level of the document as well as for the list definitions deeper in a document logical structure.

Based on the most current structure format like the simple and multi-level models, the current module targeted the main title, section titles, introductory sentence presenting a list and the list items per se. It then extracted the starting and ending position so it can be easily matched with the multiple occurrence of the candidate list.

Table 2.5 Examples of list structure

Simple	Multi level	Mixed multi level	Interleaved mixed multi level
1.	1.	1.1	1.
2.	1.1	i.	A-
3.	1.2.1	ii.	B-
OR	1.2.2	1.2	2.
*	2.	a)	C-
*	2.1	b)	3.
*	2.2	c)	4.

2.4.10 Relevance ordering algorithm

Once all the candidate concepts have been extracted and the clean up and the verification process has taken place, the final list can be reordered to optimize concept learning by placing the most relevant ones at the top of the list and all the others in a decreasing order of relevance.

The technique used to reorder the candidate list uses a weight distribution schema coupled with a density analysis to assign a relevance score to each candidate. The steps of this technique are defined as follows :

- a. Type-based structural weight attribution
- b. Structural weight propagation
- c. Context segmentation
- d. Conceptual weight distribution
- e. Weight repartition
- f. Reordering

The starting point of this technique requires both an unordered candidate concepts list $C = \{C_1, C_2, C_3, \dots\}$ and a list of all the structural items in a document. The four structural items'

type suggested are the main title, the section titles (of any depth), the introduction or header sentence of the document's lists and each individual items of the lists. The set of types can be generalized as $T = \{T_1, T_2, T_3, \dots\}$. These specific structure's types were selected as they usually carry a higher semantic payload for domain knowledge; titles may be used to denote global or high-level concepts while lists introduce concepts which are relevant enough to expose their composition or relationship. The set of occurrences of structure within a document can be defined as the list of typed elements $S = \{S_{1T_n}, S_{2T_n}, S_{3T_n}, \dots\}$ where T_n can be any of the four types defined earlier (or any other for that need).

The first step is to define a weight vector $W = \{w_1, w_2, w_3, \dots\}$ of scalar values, each corresponding to a type in the T set, so that $|T| = |W|$. These values should reflect the relative importance between elements of T as well as the inherent value ($w_i > 0$) or non-value ($w_i = 0$) given to each structure type.

Once the weight set W have been defined, the next step consists of propagating each structure's weight to the contained candidates. This is done by accumulating, for each candidate concept, the total weight of the individual structures which contains at least one occurrence of the candidate. Single structures which span multiple occurrences of the same candidate are only added once. On the other hand, individual structures of the same type or textually identical individual structures (i.e. a section title and a list item with the same text) are all added up for a specific concept. This can be expressed as $W(C_i) = \sum_{i=1}^s \sum_{j=1}^c W_i * \text{Bin}(C_j, S_i)$ where W_i is the weight attributed the structure instance i , C_j is the instance j in the set of concept C and S_i is the instance i of the structure set S . The function $\text{Bin}(C_j, S_i)$ returns 1 if S_i contains C_j and 0 otherwise. This formula, applied to each concept and each structure, give the global weight of each concept for a document.

For each concept that have $W(C_i) > 0$, a surrounding context is defined for the next step. This context can be a proximity zone (i.e. the N words written before and after each occurrence of the concept) or a linguistically delimited zone, like the containing sentence or paragraph. The goal of this step is to find a relevant segmentation within which an occurrence of a concept

may spread its cumulative weight to other occurring concepts which are not represented in the document structures.

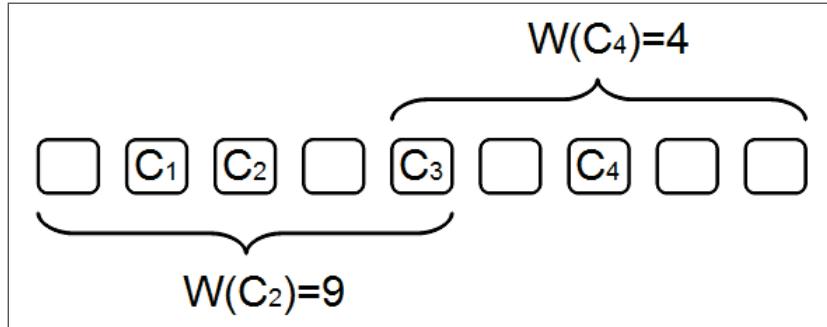


Figure 2.6 Example of contextual distribution of weights to surrounding concepts

Once the context has been defined, the fourth step is to define the total weight density of each area of the document. For each context area surrounding weighted concepts ($W(C_i) > 0$) in a text, weights are divided equally between surrounding concepts. This is done in an incremental manner such that weight accumulates in areas of overlapping context.

In the example shown in Figure 2.6, we define a context segmentation (in the previous step) of ± 2 surrounding tokens, with starting values of $W(C_1)=0$, $W(C_2)=9$, $W(C_3)=0$ and $W(C_4)=4$. In the first step, the 9 units of weight from C_2 would be divided between all the the concepts contained within its surrounding (including C_2 as well), which would give $C_1=3$, $C_2=3$ and $C_3=3$. The weight from C_4 (4 units) would then be split equally with C_3 , adding 2 units to the 3 already accumulated by C_3 (from the propagation of C_2). Following these calculations, the final values would be $C_1=3$, $C_2=3$, $C_3=5$ and $C_4=2$. If the context segmentation was defined as the containing sentence and all four concepts were in the same sentence, then each concept would inherit $(9+4)/4$ units of weight. This step is applied under the hypothesis that concepts appearing in the surroundings of structure-based concepts are more likely to be relevant. The second to last step can then be realized, which is to add up, for each concept, the weight of each of its occurrences within a document.

Finally, the concepts are reordered by descending cumulative weights with zero valued concepts trailing at the bottom of the list. Concepts with equal weights are not sorted in any meaningful way between them. It provides the final ranked system output list which can be compared to the ranked gold standard as described in the performance evaluation section.

2.5 Evaluation

2.5.1 Experimental setup

This research used two text corpora: a *development corpus* and an *evaluation corpus*. The first was used to develop, test and optimize the filters of the pipeline. It contains ten French documents taken from public government websites (mainly treasury and health department). In addition to being used to develop the filters, some of them were manually tagged by the author to test the final relevance reordering step of the pipeline. To prevent a training effect on the evaluation corpus, the development and fine-tuning steps of each of the implemented modules in the pipeline were done using this set of documents which were different from those of the evaluation corpus regarding the subject matter, the source organizations, and the documents' structure.

The second corpus, called the evaluation corpus, is composed of 227 French documents taken from public organizations (academic and public works) which were from different domains than the documents from the development corpus. The academic subcorpus contained 152 documents which were provided by the archive service of the École de technologie supérieure⁶. The public works subcorpus contained 75 documents consulted from the public works and government services Canada⁷. From this evaluation corpus, five documents of various lengths were taken to be annotated manually by the group of software engineering experts. Documents were randomly taken from the subcorpora: three from the academic subcorpus and two from the public works subcorpus. This combined subset of five tagged documents taken from the

⁶<http://www.etsmtl.ca>

⁷<http://www.tpsgc-pwgsc.gc.ca>

evaluation corpus is called the *gold corpus*. It can be further specified that the three tagged documents taken from the academic subcorpus form the *academic gold corpus* and the two remaining tagged documents are called the *public works gold corpus*.

Evaluation was realized on the sentences contained in the gold corpus on a per-document basis which means that each document was analyzed in isolation from the others to prevent a training effect from some filters. For example, an implicit acronym found in the first document of the academic gold corpus cannot be applied to the other two documents; they must be detected in the other documents as well to be selected for the candidate list.

The pipeline filters were thus applied on each document individually in order to generate an unordered candidate list. Each concept was then automatically assigned with the corresponding weight as described in Section 2.4.10 and reordered accordingly, producing the ordered concept list. For the set of structure types $T=($ main title, section title, introduction list sentence, list item $)$, the set of weight $W=(4, 3, 2, 1)$ was used for the algorithm described in Section 2.4.10.

The ordered concept list was then outputted to a text format to be checked manually as some expressions could not exactly match the concepts from the gold standard. One example is when the pos-tagging tool used during the candidate extraction step produces ambiguous lemmas that are taken without correction by the pipeline and then integrated into the final expressions . For example, the word “taux” (rate or rates) is assigned a lemma string for “taultaux” (the written name of symbol tau or rate or rates) as the tagging tool cannot disambiguate between the two meanings. In this case, the final expression of “taux de taxe” (“tax rate”) would be “taultaux de taxe” (“tax taulrate”) which cannot be compared to the correct expression.

2.5.2 Baseline comparison

The previous extraction and ranking methods were compared to a baseline method to show the potential improvement. The baseline implemented for comparison uses the tf-idf term weighting measure for relevance ranking. It is, of course, the log reduced normalization (Manning and Schütze, 1999) form (see equation 2.2).

$$tf - idf_{t,d} = tf_{t,d} \times idf_t = tf \times \log \frac{N}{df_t} \quad (2.2)$$

This measure is known to provide a high score for domain-specific knowledge and a downward effect to expressions which are commonly used throughout a document corpus. The candidates expressions were generated using a n-gram model restricted by sentence boundary, conjugated verbs (except for participle and infinitives) and separators like parentheses and commas. Given that the longest concept in the the gold corpus was eight words long and that we aimed for a high recall for the extraction process, we extracted all the possible n-grams for a length (the value of N) of eight words in order to enable the baseline to extract concepts for every length of expression found in the gold corpus. The manual limitation on the length of the extracted expressions was done to prevent the baseline from overproducing irrelevant expressions. As there is no natural threshold associated with the tf-idf method to assess the relevance of an expression, there is no standard way to know when to stop generating larger expressions. As a final step, all expressions starting or ending with a determinant or an article were removed from the list as they would not match any of the expressions found in the gold corpus. This was done to put the baseline on a even level as the presented process which also removed the expressions during the candidate extraction phase.

As this is a statistical method, applying it to a single document, like for our method, would not have produced a useful output. Thus, it was applied once on the academic evaluation corpus of 152 documents and once on the 75 documents of the public works evaluation corpus. This means that the tf-idf scores were calculated on the total span of documents from each domain. Then for each of the five tagged documents of the gold corpus, expressions found both in the document and the corresponding ranked tf-idf list were taken to create the document's specific ordered concept list. For example, the expressions from the tf-idf scored list produced with the public works evaluation corpus were checked if they appeared in the first document from the public works gold corpus. Those found in this document were selected, in order, to create this

document's final concept list for the baseline. This step was considered to give an additional edge to this method compared to applying it directly on the gold standard.

The choice of this baseline is also related to the fact that this method of relevance evaluation is known to be language independent. This was an important factor as no adaptation was needed to apply it to French documents.

2.6 Results

2.6.1 Candidate extraction

The complete pipeline was applied to the gold corpus to evaluate the performance using the recall and precision of the output. The performances at each step of the process can be studied in Figure 2.7. The center line is the resulting f-measure score after the application of each filter. At this point, it is important to note that the performances are calculated on the unordered list of concepts provided by the processing pipeline. This output is compared to the list of all the annotated concepts from the gold standard to compute the recall and precision score, without regards to the number of annotators that selected each concept. This is done in order to provide a comparison to traditional, unordered, extraction processes using typical performance measures.

As the performance lines show, the recall slowly degrades from 93.55% to 90.77% as the precision improves from 42.65% to 53.85%. While not perfect, the recall level stays in the proximity of the 90% mark, which is a small drop from the starting 93.55%. It can also be seen that a few of the consecutive improvements are neglectable when compared to the precision score of the previous step. This is mainly because some modules filter, in or out, the same concept as their predecessor. The recall-focused process provides a 67.60% F-measure score to the reordering final step.

In comparison, the baseline applied on same the documents produced a recall of 45.75% with precision of 0.98%. The combined f-measure of 1.92% for the baseline is much lower than our

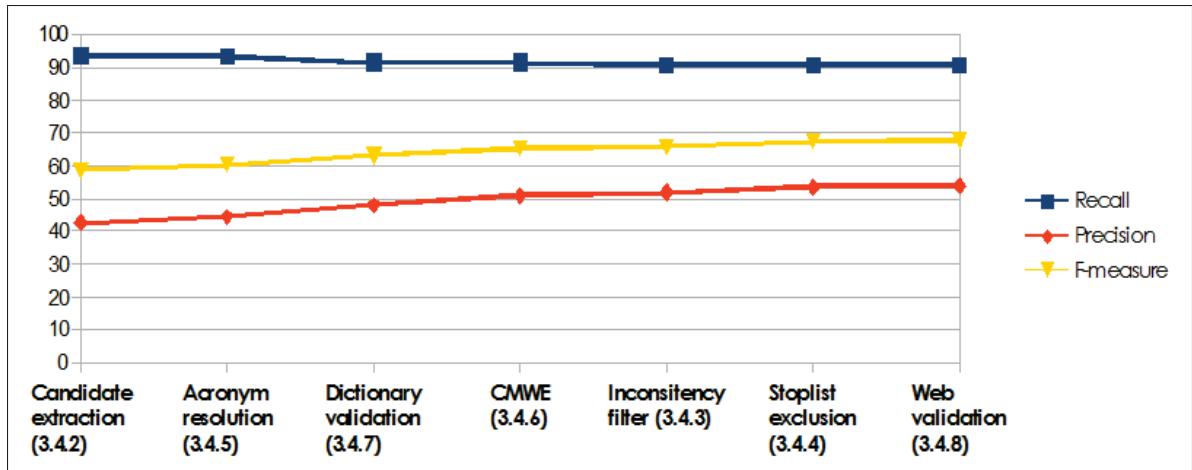


Figure 2.7 Concept-based performance for each cleaning step

approach as statistical approaches tends to overgenerate candidates, even after many candidates are filtered out.

While not shown, the exact order of the filtering modules did not have much effect on the final performance. This is due mainly to the overlap between some of the module affected candidate. For example, many dictionary entries were also found by the web validation module. As a consequence, this last module did not have much influence on the overall performance of the pipeline. But if placed before the formers, the effect would be inverted to a certain degree. Still, the performance curve shows that the proposed complex multiword expression (Section 3) and the acronym resolution (Section 4) filters give an appreciable rise on the precision while lowering slightly the recall performance. The dictionary filter offers the highest increase of recall but also the highest reduction for precision. Others, like the web validation and the inconsistency filter did not influence the performances in a significant way but did not hinder them either.

2.6.2 Candidates ordering

Following the candidate filtering, the neutral (in terms of removing or certifying candidates) reordering module was applied. As shown in Figure 2.8a, the global reordering performance was

distinctively more effective when using the hints from the logical structure of documents than from the statistical inference of relevance provided by the baseline. Corpus-specific performances, illustrated in Figure 2.8b and Figure 2.8c, show similar level which is a good indicator that the approach is relatively stable when applied to corpus from different organizations.

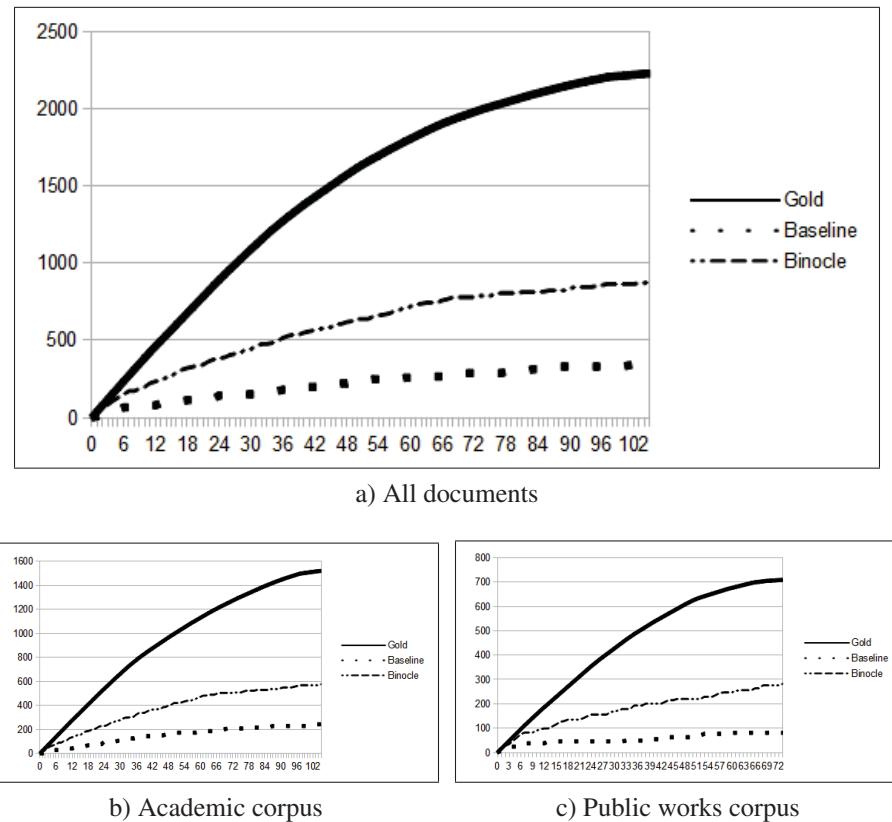


Figure 2.8 Ranked system (RS) curves compared to ranked gold (RG) on various document sets

While the Binocle's performance curves are lower than the gold standard, the calculated corresponding MDR scores presented in Table 2.6 show a clear improvement. Baseline MDR scores, while stables, are approximately 35% to 40% lower than our method. Nonetheless, the MDR score is of the same order than the inverse performance curve show in Figure 2.4 which included all the gold concepts in reverse order. This means that will all the needed items were

not included within the scope of the gold standard length, the method can still provide steady flow of relevant concepts for its output list.

Of course, the performances were different depending on the length of the analyzed documents versus the quantity of logical structures that could be capitalized upon. The shortest document was scored with a MDR of 59.85% for Binocle compared to 24.37% for the baseline. The baseline similarly performed at 23.42% for the longest document and the proposed system MDR lowered to 46.64%.

Table 2.6 MDR measure for the reordering process

<i>Gold corpus</i>	<i>Baseline</i>	<i>Binocle</i>
Complete	0.1593	0.4300
Academic	0.1693	0.4265
Public works	0.1591	0.4466

2.7 Conclusion

This research was aimed at improving the speed of extraction of relevant concepts from business documents in the context of a software project. It was implemented as a pipeline including a loose candidate generation step followed by filtering module which certified or removed candidates. As this first step produces a low precision output, it is followed by a reordering of the concepts list to optimize it for faster learning by a human or software agent. To better compute the impact of the process, a gold standard was defined for the need of evaluation and a clear improvement was achieved when compared to the language independent baseline for relevance evaluation.

A measure of relevance, the mean discovery ratio, was also introduced to better gauge the current and future systems and their improvement regarding the need of the users during the domain analysis phase of a software project. Using this measure, our system performed at 43% compared to the baseline of 15.93%. This combined score was stable even when applied to two corpora from different organization.

The next logical phase of the project will be to model specific diagrams adapted to software engineering project from the extracted and ordered list and to evaluate the effect of new filtering module on the pipeline and output. Another research path will be to explore specialized filtering using small user input to bootstrap the relevance evaluation process. This would require a small exploration phase from the user on a few documents to denote his view of relevance through selection of concepts which would then be used by the process chain's modules to adapt their definition of relevance. This will help boost the usefulness of this approach as well as better way to support software expert in their acquisition and exploration of domain knowledge.

CHAPTER 3

ARTICLE II: HYBRID EXTRACTION METHOD FOR FRENCH COMPLEX NOMINAL MULTIWORD EXPRESSIONS

Pierre André Ménard and Sylvie Ratté

Département de génie logiciel et des TI, École de Technologie Supérieure
1100 Notre-Dame Ouest, Montréal, Québec, Canada, H3C 1K3.

This chapter has been submitted for publication in the
Computational Intelligence journal on April 3rd, 2014.

Abstract

Many documents contain specialized concepts and named entities formulated with long nominal multiword expressions. These expressions are essential to the full comprehension of business corpora but are seldom extracted by existing methods because of their form, the sparseness of occurrences and the fact that they are usually excluded by the candidate generation step. Current extraction methods usually does not target these types of expression and perform poorly on their length range. This article describes a hybrid method based on the local maxima technique with added linguistic data to help the frequency count and the filtering. It uses loose candidate generation rules aimed at long and complex expressions which are then filtered using ngrams semilattices constructed with root lemma of multiword expressions. Relevant expressions are chosen using a statistical approach based on the global growth factor of ngram frequency. Two annotated corpora were used to produce baseline performance data using a modified statistical approach. The results indicate an increased in average F1 performance by 23.4% on the larger corpora and by 22.2% on the smaller one.

3.1 Introduction

Exploration of a large corpus of text documents, like the internal work document used in a company or organization, can be facilitated by concept mapping (Osada *et al.*, 2007). The concept extraction process needed to create the maps must rely on terminology extraction methods which are widely studied and still offer a rough challenge to most natural language processing applications.

These concepts occur either as single word or as multiword expressions. Multiword terms found in those texts can be either generic (non domain-specific) or specialized (domain-specific). Some of these domain-specific expressions can be much longer than their average counterpart because of complexity of domain structure and the need to clearly identify a specialized entity. The efficient automatic extraction of these long domain-specific terms (coined “complex multiword expression” or CMWE in this article) faces many challenges: unusually long expression, non-standard words and elements, embedded smaller named entities, low document and corpus frequency, occurrences restricted to specific documents in the corpus, limited to an organization or a department, slightly variable expressions denoting the same concept, fixed and truncated forms used for the same concept and so on. These characteristics make readily accessible MWE extraction tools far less effective for CMWE identification.

Our project focus on the extraction of multiword expression business documents written in French with a clear aim on CMWEs which are rarer and elusive by nature. The particular intent of this project pressures us to prioritize recall over precision whenever possible. This was logically adopted from contextual usage of the project: it is far simpler for an expert to discover an out-of-scope concept in a conceptual model than to find a missing one in a thousand page corpus or to guess that one is actually missing. This method tries to circumvent the elusive nature of CMWEs in French by using a generalized form of the expressions, loose candidate generation to improve the recall factor and a frequency-based elimination step to filter non relevant expressions for an improved precision level.

3.2 Overview

3.2.1 Context

Concepts found in texts can be illustrated by single common nouns (*table, process, canvas, etc.*) or by multiword expressions (*barcode scanner, bill of lading, etc.*) corresponding to a set of recognized patterns (Laporte *et al.*, 2008): Noun, Noun+Noun, Noun+dett+Noun, Noun+Adj, etc. Other longer nominal expressions, still relevant to the analysis, are outside this set of patterns. They often represent important business actors or artifacts that must be modeled in order to better understand the business processes. These concepts can contain inflected verbs, conjunctions, and other elements not normally encountered in single or traditional multiword concepts. These expressions are therefore considered “complex” because they do not necessarily follow patterns mentioned earlier, are difficult to automatically identify in a text and can consist in idiosyncratic aggregation of smaller stand-alone MWEs.

These concepts mostly start with a noun phrase (NP): Noun, Noun+Adjective, Noun+Noun, Adjective+Noun, Adverb+Adjective+Noun, Adjective+Adjective+Noun and so on. These NPs, when used as the head of the expression, indicate the basic nature of the entity. Attached to this head are other structures (NP, VP, etc) which further specify the uniqueness or specialization of the entity or one of its modifiers.

They can designate, for example, the name of a department, an employee’s job title, a type or name of internal or official document, a standard, a named software system, a law’s functional name, etc. They are more often found in documents of organizations that are highly structured, such as large companies, multinationals and government’s departments or agencies. Of course, some domain don’t use this type of expression, while on the other hand, domains around the legislative sphere usually use massively CMWEs. This study also targets CMWEs that may be named entities (NE) or includes one or more of them as modifiers or head. While Grishman (1996) denotes three types of named entity, organization (name corporate or governmental), person (name person or family) and location (name of politically or geographically defined

location like cities, provinces, countries, international regions, etc.), other types of named occurrences may exist within an organization like a specific department, a system, a norm, etc. They can also be defined using a complex multiword expression, they are therefore considered part of the scope of this effort. Also in the scope are expressions that can be tagged as semi-fixed expressions by the classification suggested in Sag *et al.* (2002). Because they generally keep the same word order while being able to be slightly modified at different levels, they can be considered as a word complex with a single part of speech. They falls more frequently in the compound nominal and the proper name categories.

3.2.2 Examples

The following examples are real world instances of CMWEs ranging from 7 to 30 words for French and from 7 to 24 for their translation in English. The three first examples are taken from private company documents and represent respectively an organizational structure, a document's title from a work procedure and a position's title. The last three items are concepts from governmental sources (from Québec, the United-States and France) and are titles for a job role and two laws.

- a. Direction de la politique des biens immobiliers et du matériel (*Management of the policy for real estates and material*)
- b. Manuel des standards de paramétrisation pour le broyage par attrition (*Manual of parametrization standards for attrition grinding*)
- c. Délégué aux ressources humaines et relations syndicales (*Human resources and labor-union relations representative*)
- d. Ministre délégué à la sécurité sociale aux personnes âgées aux personnes handicapées et à la famille (*Minister for Social Security for elderly people with disabilities and family*)
- e. Loi sur le programme d'aide aux Inuit bénéficiaires de la Convention de la Baie James et du Nord québécois pour leurs activités de chasse, de pêche et de piégeage (*An Act*

respecting the support program for Inuit beneficiaries of the James Bay and Northern Québec agreement for their hunting, fishing and trapping activities)

- f. Acte uniforme pour garantir la comparution de témoins à partir de l'extérieur d'un état dans les procédures pénales (*Uniform act to secure the attendance of witnesses from without a state in criminal proceedings*)

When used in a sentence, these examples may play the role of subject or object like any single-word concepts. Furthermore, in the case of the longer examples, they may in fact cover most of the length of the containing sentence. Some even include verbs or punctuation which are often usually excluded by multiword extraction techniques when searching for shorter expressions. Some other, like example e., contains many nested smaller multiword expressions and named entities: "programme d'aide" (*support program*), "Convention de la Baie James et du Nord québécois" (*James Bay and Northern Québec agreement*), "Baie-James" (*James Bay*), "activités de chasse" (*hunting activity*), etc. These might also be relevant individual concepts that have to be extracted in addition to the larger CMWE.

3.2.3 Challenges and issues

In spite of their potentially important role in the business environment, extraction of these expressions faces many challenges. One major issue is their relative sparseness both in existence and usage. They may be found in a few documents across the entire corpus or mentioned only a few times in each text. In shallow corpora, that is to say containing a large variety of subjects but with low repetitiveness of concepts, they may prove harder to extract as their occurrence rate might be at the same level as non relevant expressions such as the repetitions of a sentence in different parts of a document (i.e. the legal warning at the end of a document). Even in large corpus, or at least in less shallow ones, some CMWEs are often replaced with a smaller substitutive term to ease the writing and reading effort. This smaller version of the CMWE could be an acronym, a truncated version of the full expression or an alias. This phenomena

would keep the full-length occurrences of the CMWE lower than their actual usage, making them more evasive for general term extraction methods.

These expressions are also difficult to extract because they're often made up from smaller valid MWEs, NE and single word concepts. Semantically speaking, they are not simply an aggregate of shorter expressions since their clustering create new expressions representing a concept distinct in nature from its parts. Those can be stand-alone relevant concepts which must also be identified independently. Linguistic and statistical methods often break up the CMWE and overproduce syntactically valid (but often semantically invalid) shorter MWEs or CMWEs.

Generally speaking, French multiword expressions are longer in word count than their English equivalent, mostly due to function words in between content words. These structures are, most of the time, number and gender specific determinants and articles which are needed to link NPs and other phrases together. These NPs, when used as modifiers in a MWE, are not necessarily linked to the immediate previous NP as it can skip link, for example, to an earlier NP or to the head NP at the start of an expression. Moreover, feminine nouns may add more tokens than a masculine one, so CMWEs containing many feminine nouns might end up quite longer than masculine ones. For example, the expression "Direction de la surveillance et de la protection de la santé" ("Surveillance management and Health Protection") contains three feminine nouns ("surveillance", "protection" and "santé") which requires the bigrams "de la" ("of") in front of them. If these were masculine nouns, it could be contracted as "du" ("of"), which would make the entire expression 27% shorter. As a consequence, the link between "direction" and "protection" is longer, which can hinder some local search space based noun phrase resolution method. These added tokens and long distance links both make MWEs in French more pervasive than in English.

3.2.4 Uses and proposed method

Difficult as they may be to extract, these expressions can be beneficial to many text mining applications. From a terminology stand point, they can enrich term dictionaries as they are often considered relevant concepts. They also help prevent the overgeneration of non relevant concepts as the nested terms might be a lot less relevant than their container when they always appears as part of a CMWEs. For deep parser tools, the substitution of a long expression by a temporary placeholder might have a positive effect by simplifying the sentence and thus minimizing the structure analysis effort. Text mining tools using a candidate generation step for relevance estimation of terms might benefit from the identification of CMWEs by limiting the overgeneration of candidates by giving an accurate frequency analysis of included concepts. In the context of a concept exploration project such as this one, leaving out the CMWEs of the final concept list (leaving in only the smaller expressions), be it a textual list or a visual concept graph, might not provide enough hints to the observer on the existence of missing larger expressions. This might lead to lost knowledge or erroneous assertions depending on the scope and objectives of the exploration task.

The global approach begins with a preparation step that generates ngrams from each document in the corpus. The ngrams are then simplified using their root lemma to obtain a corpus frequency. The frequency is then analyzed, taking into account only ngrams presenting a significant rise in frequency in comparison to their parent ngrams. A parent-child relationship is defined between two ngrams when the larger expression (the parent) contains a smaller expression (the child). A stop list is used in the preparation step to define the limits of the ngrams. A stop list is a list of frequent words which have little independent semantic content, such as prepositions and determiners Yeates (1999). In this case, the stop list contains verbs of specific tenses, pronouns and other words that are not typically used in CMWE. This choice reflects the need to extract nominal CMWE.

For each CMWE, and for each ngram listed by our technique, a set of words that are potentially significant (nouns, adjectives and verbs) are extracted to simplify their comparison. Singular

masculine flexions, for names and adjectives, and infinitive tense for verbs, are kept to create these labels named “root lemma” of the expression. Therefore, the ngrams “*le chef d'équipe*” (“the team leader”), “*des chefs des équipes*” (“leaders of the teams”) and “*le chef de l'équipe*” (“the leader of the team”) all possess the same root lemma: “*chef équipe*” (“leader team”). The goal was to simplify comparison between ngrams and allow significant frequency calculations.

3.3 Details of the proposed method

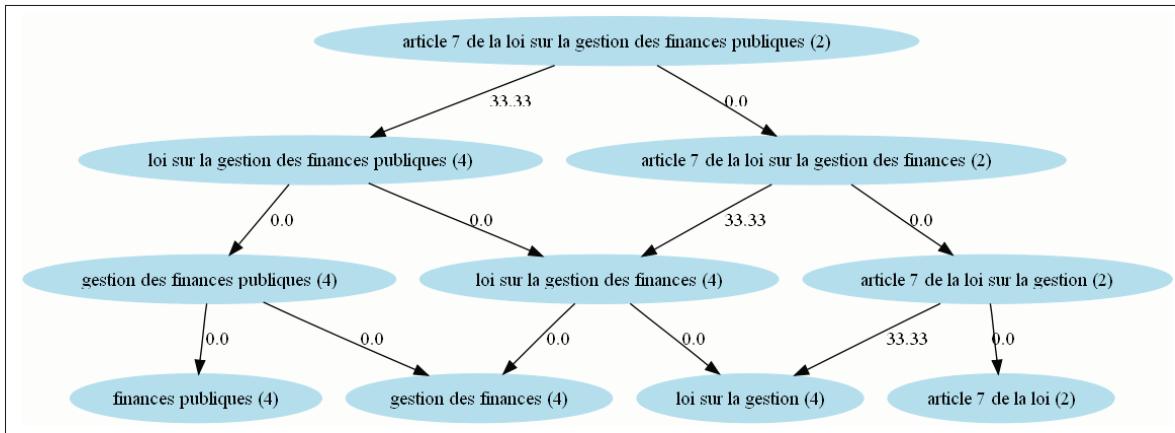


Figure 3.1 Complete semilattice with calculated T_k growth ratio (on edges) and the ngram frequencies (within nodes)

This technique is applied in three logical steps: the ngrams’ acquisition, the construction of semilattices and the cleaning of these semilattices. The first step is applied once for the entire corpus and then the two remaining steps are applied iteratively for each documents and their extracted ngrams.

3.3.1 Acquisition of ngrams

In order to apply the technique, it is necessary to collect the set of all possible ngrams in each documents in a corpus. The Gate Cunningham *et al.* (2002) platform is used to manage the corpus and run a sequence of tools (tokenizer, sentence splitter, etc) on each document. Tokens extracted from each document are labelled with the TreeTagger part-of-speech tagger

to identify the lexical category of each word. A home-made plugin for Gate then delimits each possible ngram, summarizes their frequency in each document and stores them for processing. The ngrams length can be bounded by punctuation signs, sentence boundary or stop words.

Basic rules were defined in order to delimit ngrams, while supplying a great flexibility to avoid losing complex elements. The rules are the following:

R1 The ngrams contain no sentence or end-of-line punctuation signs, possessive pronouns, subordinate conjunctions or relative pronouns.

R2 ngrams partially or completely overlapping institutionalized¹ expressions or common locutions are rejected (ex: “*at every moment*”, “*acting as*”, *etc.*”)

R3 ngrams not starting with a noun, adverb or adjective or not finishing with a noun, adverb, adjective or verb, are excluded.

Rule R1 helps to maintain a sort of syntactic coherence, assuring us that we are not overlapping other context specific group (see section 3.4.1 for the full explanation). The list used in R2 is a non-exhaustive list of locutions and expressions that is constantly updated. It now contains more than 550 elements. The idea behind rule R3 is to insure that we will process only syntactically valid nominal candidates.

The root lemma of each ngram is kept and the frequency of each term, always based on its root lemma, is calculated for the entire corpus. Ngrams with a corpus frequency of 1 are therefore eliminated.

3.3.2 Semilattice construction

The construction of semilattices is based on a tree structure in which each node can possess one or two parents. This structure is needed to represent the source contexts of smaller ngrams.

The technique is defined by the following algorithm:

¹(Sag *et al.*, 2002) : “Institutionalized phrases are syntactically and semantically compositional, but occur with markedly high frequency (in a given context).”

- a. Extract the longest unprocessed ngram having a corpus frequency of at least 2.
- b. Add it as a semilattice root.
- c. Extract the P list of unprocessed embedded ngrams included in the semilattice root expression in decreasing order of the size of the root lemma. Preserve their corpus frequency for the reduction stage. For example, if the chosen root is “*loi sur la gestion des finances publiques*”, the P list could include “*gestion des finances publiques*” at the first element of the list and then “*finances publiques*” in second position.
- d. Submit each sub ngram (NG_p) from the P list to the root; each node N_R receiving it must then :
 - a. Transmit it to each of its child node (N_{C_1}, N_{C_2}, \dots) to verify if it contains NG_p .
 - b. If no N_{C_i} contains the NG_p being processed, add it as a $N_{C_{i+1}}$ of the current node.
 - c. If NG_p is included in N_{C_i} , it accepts NG_p ; the verification process of step 4 is then reapplied with N_{C_i} as the new N_R .
- e. Mark all the ngrams in list P as being treated
- f. Repeat step 1 if non-treated ngrams remain.

The Figure 3.2 shows an example to illustrate step D where an incomplete lattice receives a new N_R expression “D E” at its root. The root node give it to his first left child (“A B C D”) which reject it as it is not contained in its expression. Node “B C D E” then receives it and accepts it. The verification continue with “B C D E” which sends it to both children nodes. The first child (“B C D”) rejects it while the second accepts it. At the last step, the node “C D E” has no children to send the new “D E” node, to it is added as one of its child node as specified in the second action of step D.

Figure 3.1 shows the final result of the semilattice construction of a ngram taken from a test corpus. Different levels of the semilattice indicate the size of the root lemma of each ngram.

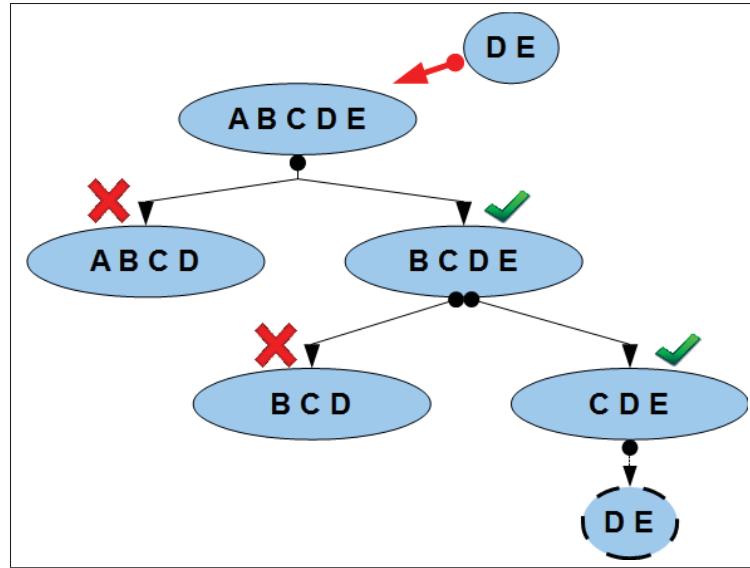


Figure 3.2 Example of semilattices construction step D

3.3.3 Reduction of semilattices

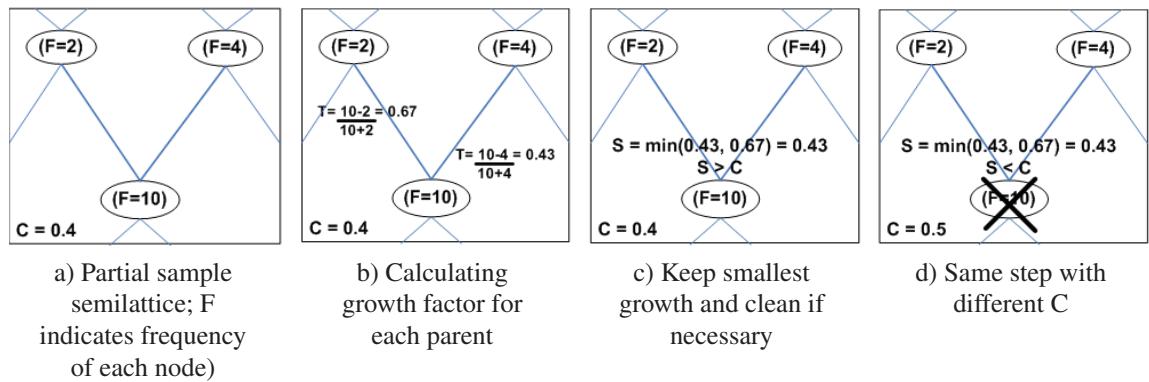


Figure 3.3 Semilattices reduction steps

At this stage, each semilattice is treated individually to eliminate nodes containing terms that are considered irrelevant. To do this, a recursive call is launched for each node, which calculates if the T_k rate increases:

$$T_k = \frac{F_{node} - F_{parent_k}}{F_{node} + F_{parent_k}}$$

where k is the total number of parents of a node, F_{node} is the frequency of the active node and F_{parent_k} is the frequency of the k parent of the node. Therefore T_k represents the increase rate between a studied node and a parent node k . The interval of possible values for T_k is then $[0, 1]$. The numbers in parenthesis in Figure 3.1 are the corpus frequency for each root lemma of the ngrams.

The global reduction method of each semilattice (see Figure 3.3a) is defined as follows:

- a. For each node, the T_k factor of each parent is calculated according the preceding formula (Figure 3.3b). For the root node, a virtual parent with a frequency of 2 is created to assure that very large candidates with very low frequency are not automatically retained because of a high T_k growth ratio.
- b. The S criterion of a node is calculated as being the minimal value of all T_k of its parents.

$$S = \min_1^k (T_k)$$

- c. A threshold value C is empirically defined; nodes for which $S > C$ are considered valid. All others are eliminated from the semilattice (Figure 3.3c and 3.3d).
- d. All parent ngrams initially valid for which their valid child-nodes completely cover the parent expression, are eliminated.
- e. All non-complex expressions that can be extracted by other techniques are removed.

The goal of this last step is to avoid overlapping basic techniques that are more suited for the extraction of simpler terms.

To illustrate step 4, let's consider the following case (with $C = 0.4$), where three nodes conform to $S > C$: “*manuel de définition du calendrier de conservation*” ($S=0.6$), “*manuel de définition*” ($S=0.57$) and “*calendrier de conservation*” ($S=0.8$). The first node (the parent) would then be rejected to the advantage of its two child-nodes.

If we suppose $C < 33.33$ to reduce the semilattice in figure 3.1, only the ngram “*loi sur la gestion des finances publiques*” would be kept. All other nodes would be eliminated by the method because their growth ratio is lower than C .

3.4 Evaluation

3.4.1 Corpora

To evaluate this algorithm, two French corpora used in previous researches for multiword extraction were chosen. The larger one is the French Treebank Abeillé *et al.* (2003) which is a compilation of 21,568 news snippets from the *Le Monde* newspaper containing over 1 million words. The large majority of them are one sentence long, but some contains two or three sentences, either in citation or narrative mode. The second one is from Laporte *et al.* (2008) and contains the transcription of three consecutive days of debates from the French National Assembly and the French novel “*Le tour du monde en quatre-vingts jours*” from Jule Vernes. The full corpus contains 166 000 words distributed in 8600 sentences. The original annotation effort tagged a total of 5057 occurrences of multiword expressions.

Applying the proposed methods and the baselines (described in the next section), we observed that the resulting term lists contain many valid expressions which were absent from the gold reference of the tested corpora. This is understandable since the goal of our study have a larger definition of MWE (as explained in section 3.2.1). Because of these missing terms, which were to be considered in the global performance evaluation, the list of expressions making up the gold standard of each corpus had to be revised to reflect this change of definition. To do so, the list of candidate CMWEs of each algorithm was verified to see if the originally rejected terms

were in fact valid, as it was out-of-scope to repeat the full annotation effort of both corpora. For each candidate expression in these lists, a manual verification was done to check three properties: containment, completeness and atomicity.

Containment is the property of each term modifiers to be linked to the head of the CMWE, or to one of its modifier, and not to a term outside of the expression boundaries. This verification had to be made in the textual context of each occurrence of the term. This property was introduced mainly because modifiers at the end of long CMWEs, especially those containing verb phrase, may be linked with an external noun phrase. For example, “taux d’intérêt *à court terme*” (“short term interest rate”) would not be kept if “*à court terme*” (“short term”) doesn’t qualify the “interest rate” as in “pour augmenter les taux d’intérêt *à court terme*” (“to raise in the short term the interest rate”).

Completeness verifies that all modifiers (related to the head term) expressed in the text are included in the candidate expression. If an algorithm extracts “cour d’appel” (“court of appeal”) from the three occurrences “cour d’appel de Paris” (“Paris court of appeal”), “cour d’appel du Québec” (“Québec’s court of appeal”) and “cour d’appel de Washington” (Washington court of appeal”), it would not be considered complete because of the truncated location modifiers which referred in each case to the head of the expression. Another example of excluded candidate expression is when the head NP is part of an out-of-scope verbal locution, like in “mis à la [disposition dans le cadre d’un prêt de main-d’œuvre]” where the head NP “disposition” is in fact linked to an external verbal phrase.

Candidate expressions are finally tested for atomicity which checks that it contains exactly one head noun phrase. Thus, the term “directeur du département d’ingénierie et adjoint” (“engineering department manager and assistant”) would not be valid, but the candidate “département de génie logiciel et des technologies de l’information” (“Software engineering and information technology department”) would be considered as a valid expression because only one concept (here a business role) is defined by the whole expression.

If these three properties (containment, completeness and atomicity) occurred simultaneously in any instance of a non-gold candidate term, it was added to its respective gold standard list. Non-nominal expressions were also removed from both the original and revised gold reference².

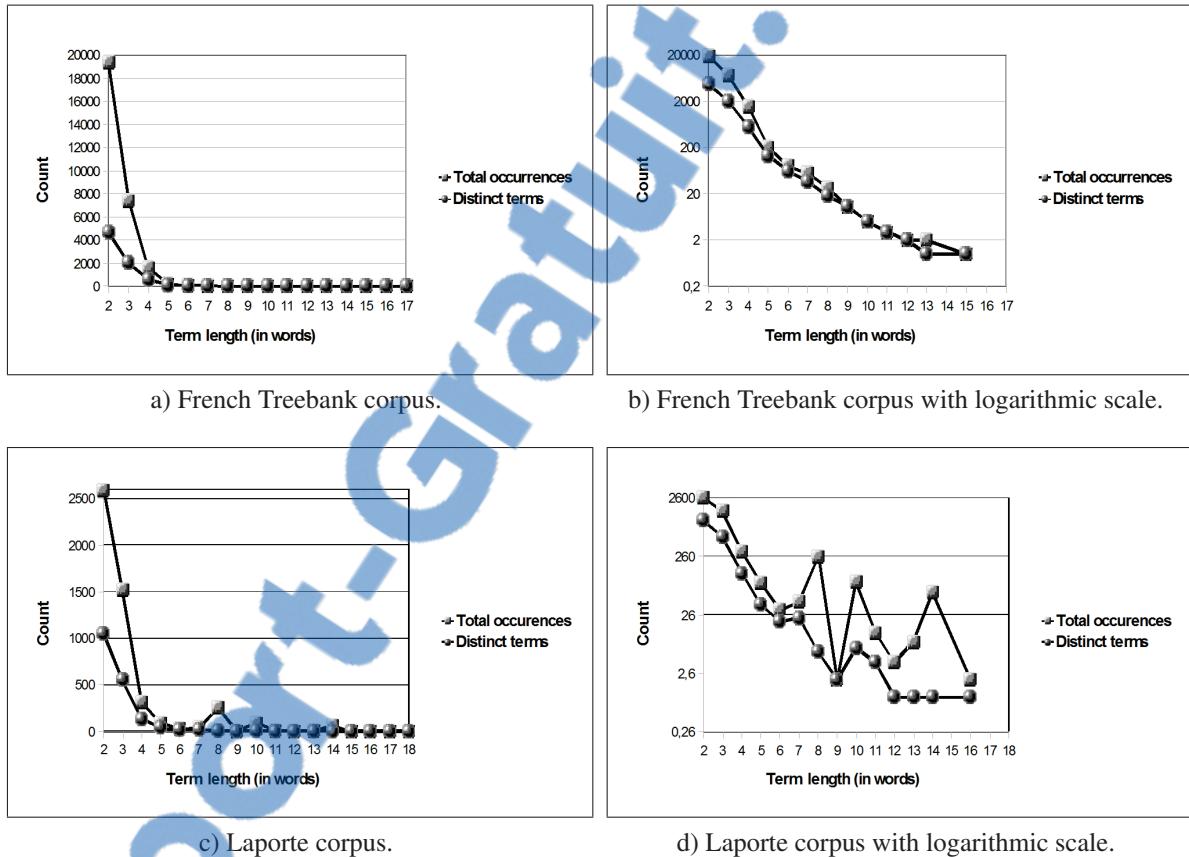


Figure 3.4 Term count for gold standard of processed corpora

Both corpora, as shown in Figure 3.4, present a similar curve in distinct terms and their total occurrences in text. For example, there are 135 unique expressions of length 5 in the Laporte corpus and a total of 207 occurrences of these unique expressions can be found in the corpus. The longest expression in the French Treebank corpus is 15 tokens long and the Laporte corpus include a 16 tokens long CMWE. Generally speaking, the longer the expressions are, the less distinct cases are present and the less frequently they occur in the text. As Figure 3.4a and

²Both corpora can be downloaded at the Binocle page of <http://lincs.etsmtl.ca>

Figure 3.4c show, multiword expression of length 5 and higher are much less frequent in both corpora compared to shorter expressions. Figure 3.4b and Figure 3.4d present the same data but with logarithmic scale for the frequency count which shows the curves of longer MWE with greater details. The French Treebank corpus shows a stable and synchronized decrease in both distinct MWEs and occurrences. The Laporte corpus, on the order hand, contains a few long MWEs which are more often repeated. This can be explained by the nature of each corpus; the first contains sentences from news articles about various events, the second is a transcript from a debate session in the Parliament. The few longer expressions in the Laporte have a greater probability of being repeated by the various speakers as they talk about the same subjects.

3.4.2 Baseline

As the presented method is based on the local maxima da Silva and Lopes (1999) algorithm, we used it as the baseline for performance comparison. It works by applying a “glue” function separately on each point, called dispersion point, between the ngram’s tokens. There are $N-1$ dispersion points in an expression of N tokens. The glue function uses statistical data from the left and the right parts of the expression. Each dispersion point result is then combined into a global value for the whole expression using the fair dispersion point normalization. If a candidate expression has a higher value than each expression of one token longer or shorter, then this candidate expression is considered a valid multiword expression. This method is useful for applying bigrams based algorithms to multiword expressions of length greater than two tokens.

The original article presents five adapted glue functions : log likelihood, dice, ϕ^2 , mutual information and the Symmetrical Conditional Probability (SCP) developed by the authors. They offer different levels of precision on the test corpus, ranging from 51.66% for log likelihood to 84.90% precision for the SCP measure. We chose to use SCP and Dice as baselines, the former ranked first in the benchmark and the latter being second-to-last. This choice was made to check if performances would fluctuate on the same range than in the original article.

As these algorithms were not meant to specifically target nominal expressions, the output lists of candidate MWEs from each execution were hand-checked to filter out unwanted expressions using R1, R2 and R3 rules (see section 3.3.1).

The multiword expressions of length 5 and higher were kept for performance evaluation against each method. The review method of the gold standard gives an advantage to the baseline algorithms because they generate a lot more candidate terms than our approach; the raw chance of discovering new valid multiword expressions is therefore greater.

3.4.3 Results

To generate results for the proposed method, all ngrams containing 16 words or less were generated for each document in the two corpora. While this length is adequate for those corpora, it must always be adjusted to the largest ngram that can be created within a sentence in the target corpus. No minimal frequency was imposed for the conservation of ngrams in the acquisition phase. All variations of ngrams were then generated for each document. All resulting CMWEs were put in a reference list for each baseline and for the current method. Each reference list was then compared to both original and revised gold standard on a strict comparison basis; in other words, the root lemma of CMWEs returned must match exactly those in respective gold standard. A base threshold of $C = 0$ was used for the semilattice reduction ($SR_{0\%}$) performance evaluation, so that all expressions with a non-null frequency growth were included.

Figure 3.5 presents from top to bottom the recall (3.5a and 3.5b), precision (3.5c and 3.5d) and F1 measure (3.5e and 3.5f) of each tested technique against the original gold standard. All three algorithms scored equivalent levels of recall and precision on the FTB corpus except for expressions of length 12 and 13 because the gold standard contained only two 12-grams and one 13-gram. Precision was low for each length except for $SR_{0\%}$ on 12 and 13 tokens long expressions. On the Laporte corpus, recall basically fluctuated the same way for expression up to length 10. It then dropped for both baselines on greater length, while the $SR_{0\%}$ gave some positive results on length 11, 13 and 14. The gold reference included only one expression for

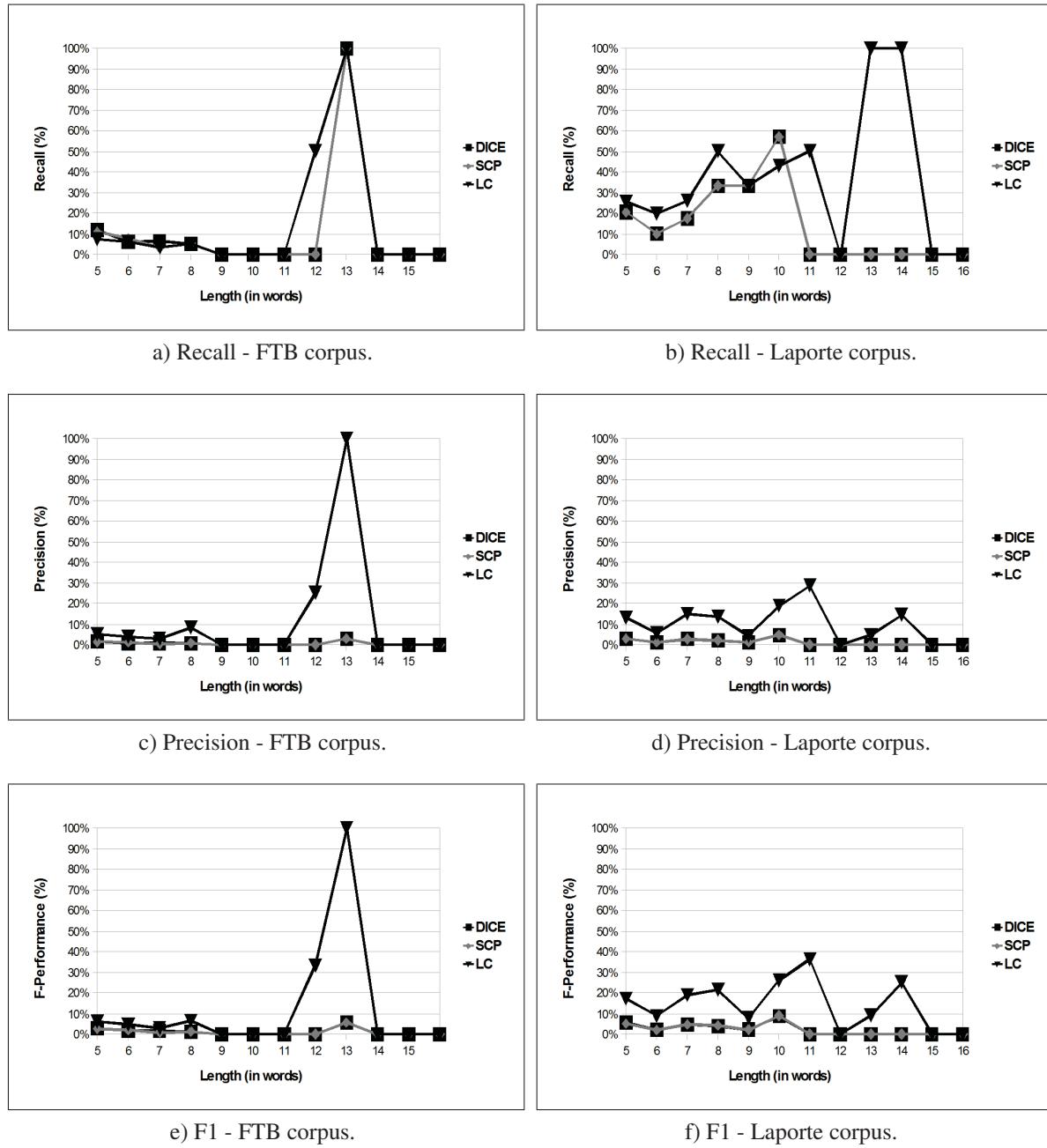


Figure 3.5 Results of extraction process against the original gold reference

lengths 12, 13, 14 and 16 and none for length 15, which explain the wildly fluctuating curves for recall; it's either a complete hit or a total miss. On average, precision level by $SR_0\%$ on Laporte is better than the baselines which produce a lot more results, even after the elimination

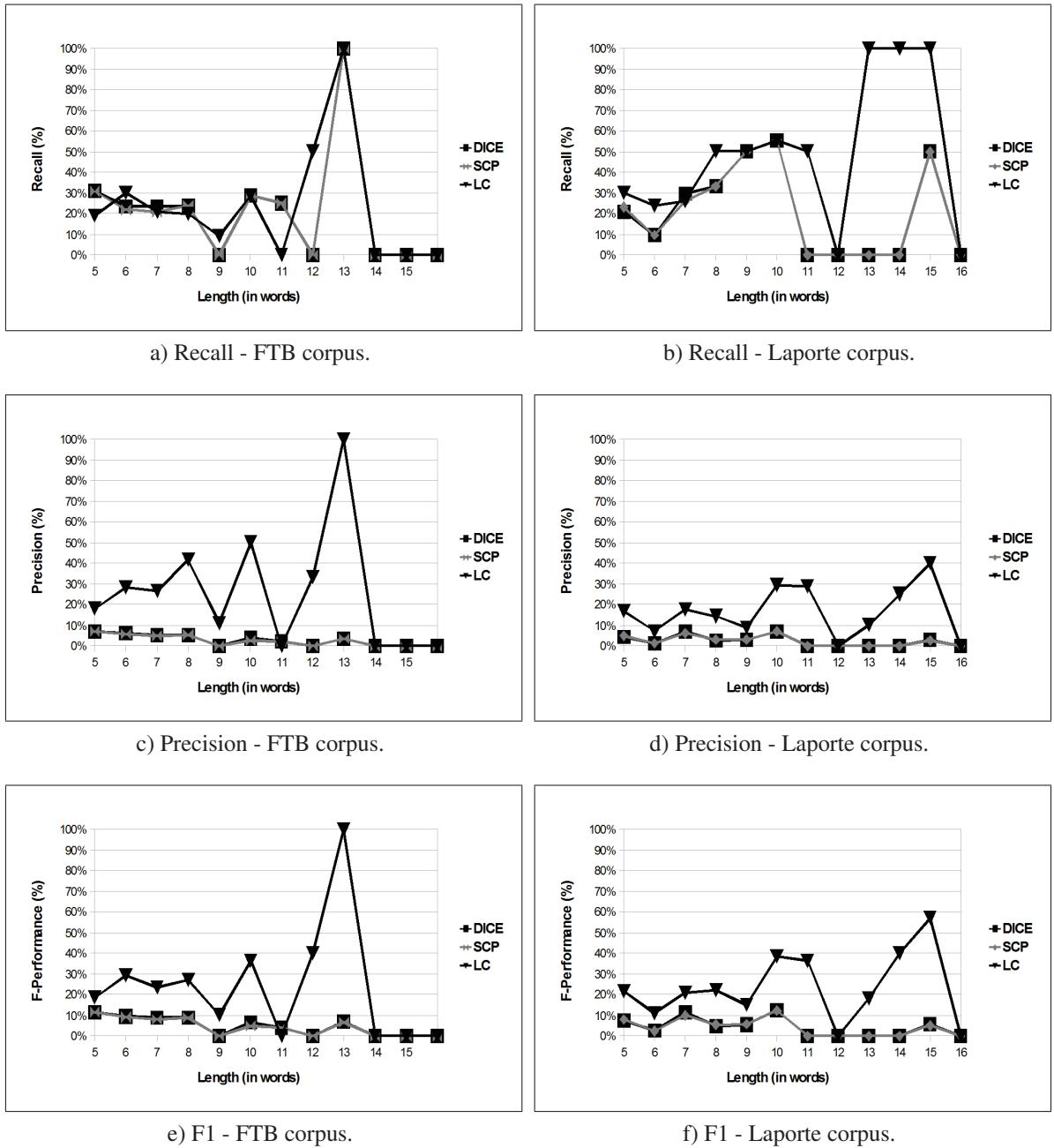


Figure 3.6 Results of extraction process against the revised gold reference

of non-nominal expressions. The resulting F1 measure is similar to the precision curve in both corpora but is slightly higher for the Laporte corpus in comparison to the associated precision graph.

After the complete revision of each gold standard as described in section 3.4.1, results were processed again for each algorithms on each corpus. Figure 3.6 presents these results which are similar in behavior to the original gold standard performances. The recall for the baselines is greatly increased compared to the original gold standard on the FTB corpus while the precision for both show only a slight increase. The recall and precision on this corpus is higher for $SR_{0\%}$, especially for lengths smaller than 11 tokens. For the Laporte corpus, fewer new expressions were added to the gold reference which explains the small difference from the original gold. However this increase pushes $SR_{0\%}$ combined performance a step higher as more relevant expressions are found with less non-valid CMWEs than the baselines.

It is important to note that the amount of noise produced by each algorithm, baseline or proposed, is very different. As presented in Table 3.1, the number of extracted expressions for the baselines is approximately five time greater than the proposed approaches (with thresholds of either $> 0\%$ or $> 25\%$) for lengths of five or six words. As the length increases, the ratio between the baselines and the semilattice reduction methods also increases to around 30 at length 10. For lengths 14 and 15 of the FTB corpus, the two baselines suggested between 30 and 38 invalid CMWEs for length while $SR_{0\%}$ extracted none, which provides much less noise in the output. At this rate, even if the recall was perfect for the Dice baseline for length 5, the precision would still be around 14% which would give a f-measure of 27%. In comparison, the proposed method produced a number of expressions for each length which was approximately the same order as the equivalent gold standard.

The average performance results for all lengths combined is shown in Table 3.2 for each method on each set of corpus and gold standard. While the recall on the FTB corpus with both the revised and the original gold standard are about the same level for all algorithms, the precision drags down both baselines because of the size of their output list, while the precision for $SR_{0\%}$ remains similar to the recall. For the Laporte corpus with both gold standards, the recall of $SR_{0\%}$ is at least twice as high as either baseline while the precision is more than 6 times higher for the proposed algorithm. These levels provide a final increase of the average F1 measure over the baselines of 23.42% for the revised Laporte and 22.24% for the revised FTB gold

Table 3.1 Total number of expressions extracted by each method on the French Treebank

Length	Gold		Baseline		Semilattice reduction	
	Original	Revised	Dice	SCP	$SR_{0\%}$	$SR_{25\%}$
5	94	127	688	698	141	124
6	51	73	348	351	81	72
7	31	43	216	221	34	28
8	19	25	130	133	12	10
9	10	11	102	108	9	8
10	5	7	74	76	4	2
11	3	4	61	62	0	0
12	2	2	55	58	4	3
13	1	1	35	35	1	1
14	0	0	37	38	0	0
15	1	1	30	31	0	0
16	1	1	27	27	0	0
17	0	0	20	22	0	0
18	0	0	29	29	0	0
19	0	0	14	14	0	0
20	0	0	19	21	0	0

reference. The average F1 performance for the $SR_{25\%}$ threshold is about the same as the $SR_{0\%}$ for every corpus and gold set except for the revised Laporte which is slightly less. For every other algorithms, both the recall and precision only fluctuate by a small margin.

3.5 Conclusion

In the context of the our project, we must extract concepts represented by complex multiword expressions to construct a better model for a given body of knowledge. Methods currently available are inadequate for this task. With the help of a simple ngrams model and basic syntactic data, the technique presented in this article allows a better extraction of concepts that can be found in business documents. The main additions from a strictly statistical local maxima approach are the inclusion of the part-of-speech data for the generation of a generalized expression and for the filtering of NP-based terms. The frequency based calculation are also more sensitive to the low frequency profiles of CMWEs.

Table 3.2 Average results for expression of length ≥ 5
for the original and revised gold standards

Corpus	Method	Recall	Precision	F1 measure
$FTB_{original}$	$LocalMax_{SCP}$	12.71%	0.75%	1.42%
	$LocalMax_{Dice}$	12.94%	0.85%	1.59%
	$SR_{0\%}$	17.50%	15.65%	16.53%
	$SR_{25\%}$	16.86%	15.31%	16.05%
$Laporte_{original}$	$LocalMax_{SCP}$	15.61%	1.68%	3.03%
	$LocalMax_{Dice}$	15.61%	1.66%	3.01%
	$SR_{0\%}$	40.72%	10.73%	16.99%
	$SR_{25\%}$	38.56%	10.46%	16.46%
$FTB_{revised}$	$LocalMax_{SCP}$	25.11%	3.12%	5.55%
	$LocalMax_{Dice}$	25.51%	3.28%	5.81%
	$SR_{0\%}$	27.76%	30.87%	29.23%
	$SR_{25\%}$	27.02%	33.09%	29.75%
$Laporte_{revised}$	$LocalMax_{SCP}$	22.51%	2.51%	4.52%
	$LocalMax_{Dice}$	22.63%	2.57%	4.61%
	$SR_{0\%}$	53.23%	17.95%	26.85%
	$SR_{25\%}$	43.63%	14.51%	21.78%

While this technique is sensible to the size of the corpus because of the statistical dimension, it can be easily tuned to provide sufficiently robust results for integration in other text analysis projects which requires a high precision on the provided results. With the appropriate adjustments to the ngrams acquisition rules, this technique may also be applied to other languages.

The next step will be to apply the method on different corpora in order to define an optimal C parameter for the reduction stage in accordance with measurable criteria such as corpus size, documents size, the nature of the documents and other available features.

CHAPTER 4

ARTICLE III: CLASSIFIER-BASED ACRONYM EXTRACTION FOR BUSINESS DOCUMENTS

Pierre André Ménard and Sylvie Ratté

Département de génie logiciel et des TI, École de Technologie Supérieure
1100 Notre-Dame Ouest, Montréal, Québec, Canada, H3C 1K3.

This chapter has been published in the Knowledge and Information Systems journal on August 8th, 2010.

Abstract

Acronym extraction for business documents has been neglected in favor of acronym extraction for biomedical documents. Although there are overlapping challenges, the semi-structured and non predictive nature of business documents hinders the effectiveness of the extraction methods used on biomedical documents, and fail to deliver the expected performance. A classifier-based extraction subsystem is presented as part of the wider project, Binocle, for the analysis of French business corpora. Explicit and implicit acronym presentation cases are identified using textual and syntactical hints. Among the 7 features extracted from each candidate instance, we introduce “similarity” features, which compare a candidate’s characteristics with average length-related values calculated from a generic acronym repository. Commonly used rules for evaluating the candidate (matching first letters, ordered instances, etc.) are scored and aggregated in a single composite feature which permits a supple classification. One hundred and thirty-eight French business documents from 14 public organizations were used for the training and evaluation corpora, yielding a recall of 90.9% at a precision level of 89.1% for a search space size of 3 sentences.

Keywords acronym extraction, classification, business document mining, similarity feature, machine learning, information extraction, natural language processing

4.1 Introduction

Acronyms permeate our everyday life in many ways. We read them daily in newspapers (UN, WTO, UNESCO, US, etc.), advertisements, and official reports. We often use them in business correspondence (FYI, PS, NB, ASAP, IMO, etc.), and even for informal private conversations (CU, LOL, TGIF, etc.) in instant or delayed text messaging. They can simplify and accelerate our reading by reducing the number of long, well-known expressions, or may slow it down in an acronym-laced text when they are unfamiliar to the reader.

Although the acronym phenomenon is quite a recent addition to the linguistic timeline, the high proliferation rate of neologisms of this type provides motivation for the development of extraction methods and systems to track them in the scientific literature. In the past few years, a great deal of effort has been expended in the creation of systems and publicly available databases to keep up with this new reality.

Acronym extraction can be used to support a large variety of applications: enrich an optical character recognition (OCR) dictionary to boost recognition performance, complete a map for relevant concepts in a digital library, provide the long form of an acronym in a textual Web document, keep a domain dictionary up to date with the latest acronyms published in recent studies, help search engines return documents containing an alternate form of an acronym, boost performance for automated analysis of documents (Woon and Wong, 2009), business information retrieval (Wang *et al.*, 2008), keyword search in relational databases (Park and Lee, 2010) and many others. They are also a challenge for NLP processing and knowledge extraction tools: unknown tokens, non standard morphology, long descriptions that are hard to identify or tie together as a unique concept, different names for a concept, repetition of a concept, etc.

As part of the Binocle project, which is aimed at modeling French business documents, we found that existing systems and algorithms were not appropriate for business corpora. Documents in these corpora, from public and private organizations, can contain mission statements, work procedures, policies, job descriptions, regulations, norms, and other texts, which can be useful to help explain how a particular organization works. Although the main issue is the same

as that for the biomedical literature, significant differences in these documents make related approaches less efficient and reliable. As a result, we leave out the specifics of the biomedical domain to focus on domains that are less well explored, but some notions may still apply in these fields.

Our hypothesis is that statistical data on the compositional similarity of acronyms can help determine whether or not a short form (SF) and a long form (LF) share a definition link by using a supervised machine-learning algorithm. We also speculate that this approach can provide a stable performance level for business corpora of different types and from different domains.

4.2 Background

4.2.1 Definition

The abbreviation and acronym phenomenon has a long history. During the decline of the Roman Empire, Latin scribes were forced to make the most of the writing material they could find or buy, since this resource was scarce. In doing so, they started using sigla (plural of siglum) to abbreviate frequently used words. Some examples, devised a few centuries before medieval times, are still in use today: Latin’s “et” (“and”) became “&”, “exempli gratia” became “e.g.”, “nota bene” was shortened to “N.B.”, and so on. Modern examples shift from common literal expressions to abbreviating commonly used named entities that would often be tedious to repeat in their long form. Earlier cases come from various fields, but mostly from military usage (MASH for Mobile Army Surgical Hospital in 1945) and from biomedical usage (DNA for Deoxyribonucleic Acid in 1953).

There is no global, all-encompassing definition for the modern acronym phenomenon. The 1986 edition of Webster’s Dictionary gives a strict definition of an acronym: “a word formed from the initial letter or letters of each of the successive parts or major parts of a compound term”, for which counter-examples can easily be found, as it focuses on specific cases of the phenomenon. Other variants of the definition include the following: “a word, usually pro-

nounced as such, formed from the initial letters of other words,” in the 1990 edition of The Concise Oxford Dictionary, or “a word composed of the initial letters of the name of something, especially an organization, or of the words in a phrase,” taken from the 1993 edition of Collins COBUILD English Language Dictionary. For example, Webster’s definition excludes cases where letters selected from a word are not the initial ones. There are also cases of acronyms for which no clear definition is available.

In the same way that there is no strict definition of an acronym in the literature, there is no clear boundary between the various names used to represent them: abbreviation, acronym, blend, clipping, contraction, initialism, portmanteau, pseudo-blend, sigloïd, sign, etc. There is also no strict convention for the name of the full-length expression: complete form, definition, description, expansion, long form, meaning, etc. RÚA (2004) shows the many typologies associated with acronyms, although many counter-examples can be found to contradict these definitions.

As our research is focused on knowledge extraction rather than on taxonomy, we look at acronyms in a more generic way, as a referential link between a descriptive expression (the long form) and a contracted expression (the short form). This link defines the two expressions as synonyms of the same concept. In this article, we use LF and SF as generic expressions to denote these two forms of an acronym. As the need for acronym extraction from biomedical texts has already been addressed in many articles, we do not focus on this type of document and the associated rules. Of course, as the acronym phenomenon shares common ground in many domains, some explanations might overlap the two types of documents.

4.2.2 Morphology

Whether they are used simply to communicate or to create a compelling symbol, acronyms can be composed of many features. These features can be used to further abbreviate an SF or to differentiate it from a common SF, but with different features, to improve pronunciation, to

give it a creative twist, and so on. The following table constitutes a (non exhaustive) list of some of the features that can be used in creating an acronym.

Table 4.1 Features used to create acronyms

Component	Usage	Example
Letter	Standard (LF reference, clipping)	Radar, motel, PPP
	Word substitution - phonetics-based	RnB, GnR
	Syllable/phoneme substitution	XGA, XML (X for « ex »)
	Roman numeral	GIV
	Numeric multiplier	Y2K (e.g. 2K for 2000)
	Word substitution – list-based	GIK (K for potassium)
	Pluralization	UAVs
Number	Nested acronym	WSR (weather surveillance radar)
	Standard	MVV12
	Repetition (single or multiple characters can be repeated, before or after the number)	3M, P2PT, AREX2C
Symbol	Word substitution	B2B, P2P, L4D, 2.5G
	Letter or word replacement	W@H, R&D, AT2i+, C#
Punctuation	Separator which may be present in LF	A/V, I/O, Wi-Fi
Case	Separator	C.I.A.
	All uppercase characters	PGP, SSL
	Mixed-case characters	eDoc, iFrame
	All lowercase characters	radar, motel

Of course, some features, like letter or word replacement, can differ for the same symbol in another language, because the word may not exist, or the letters may not exist, in other languages. For example, the @ symbol is mostly used in English as word replacement for “at”, but can also be used in French, to replace “a”, “à” or “à la” (or other accented “a”), which can be either the letter “A”(if it is part of a word) or the “at” equivalent in French.

Other features found in the SF are not represented in the LF. One case is the separator characters that are likely to change or be omitted. For example, the expression “input-output” can be found in documents written as: IO, I.O., I/O, or I-O. Another case is the “s” that denotes the plural state of a concept represented by an acronym, which is normally reserved for widely known and accepted acronyms that are found in dictionaries for which no strict grammatical rules exist. This process may also be different in other languages; the “s” is common in French for this purpose, but “x” is a possible alternative. Finally, another well-known method involves

doubling the first letter to indicate its plural form. For example, pages becomes “pp”, minutes becomes “mm”, etc.

The features illustrated in this list are by no means exhaustive. New and creative usage of symbols could produce new replacement features for which a reader, whether human or software, has no past reference. For example, the “\$” symbol sometimes replaces an “S” in an acronym to give it satirical impact when the concept can be associated with large sums of money, the “>” symbol could be a word replacement for the direction “right” or for “greater than”, “©” or a plain “c” for “copyright”, etc. Even if there is great potential for new features, this list provides a representative overview of the most common features used in acronym creation. The breadth of features also gives an idea of why a global linguistics-based definition is very difficult to come by. Methods for acronym extraction need to deal with these kinds of added features if they are to be successful in providing good performances when working with business documents.

4.2.3 Context of use

We have observed that SF acronyms are almost exclusively used in texts as common or proper names, although, in some rare cases, they show commonly used SFs that are made into adjectives with an “-ized” ending. They can be used to name an employment position (CEO), a department (HR), a title (PhD), a professional affiliation (Dr, Eng), technique (CPR), equipment (LHC), external organization (ISO), and much more.

Sometimes acronyms are used without their full description being specified in the document. The main reason for this is that the acronym is assumed to be known by readers. Some common examples are the use of UN, USA, WHO, and other well-known entities in diplomatic reports and newspapers. But more obscure occurrences can also be found in very specific business or technical documents which target expert readers who must know the specific meaning of those acronyms to be able to understand the document.

In most case, an SF acronym in a specific document is presented with its LF or freely without any introduction. This latter case is usually encountered when the author assumes that the non specified LF is common knowledge for any reader. The two forms are usually introduced when the acronym is first used in a text. This unofficial standard can be omitted when a section is dedicated to an entity represented by an acronym. Some types of publications, like internal news bulletins, assume that the reader is already familiar with the relevant business and terminology. In this context, acronyms will often be used in their SF without introduction. In contrast, presentations may occur several times in the same document, especially those that are very long or are a concatenation of smaller documents from different authors. Although acronyms can be introduced once or more often in a text, this is no guaranty that all further references to the labeled entity will be made with the SF exclusively.

We define two generic introduction methods for acronyms here: explicit and implicit. Explicit presentation clearly introduces both forms to the reader. This is done using a special separator (like parentheses) or a textual expression (“or”, “a.k.a.”, “from now on referred to as”, “also called” or simply “:”), or both. Organizations sometimes regulate their preferred acronym presentation method in their writing standards. Implicit presentations describe cases where both forms appear close enough in a text to allow readers to associate them. No clearly coded method is used to introduce the short and long forms. Even when comparing numerous examples, it is difficult to state with certainty whether or not the author used this loose method consciously, or if the two acronym forms just happened to occur in a close context by coincidence. The two forms can be separated by a wide range of distances; from a few words to a few sentences. Special separator characters may also be found in between, but without a linked presentation goal, which may further complicate the analysis. The following partial sentences provide an example of implicit presentation: “...have to fill out a customer report form (in which case, the CRF must be signed by the manager) and submit it to....”.

The wide varieties of explicit and implicit cases have been generalized in the following patterns:

Table 4.2 Generalized patterns for presentation cases

Type	Pattern	Description/Example
Explicit	.. F _x {F _y }..	Basic presentation with a separator character.
	.. F _x { .. P .. F _y }..	One form textually presented within separator characters.
	.. F _x .. P .. F _y ..	Textual presentation of a form without a separator.
Implicit	.. F _x .. F _y ..	Both forms used in the same sentence without presentation.
	.. F _x { .. F _y .. }..	One form referenced by another enclosed between separators and without presentation.
	.. F _x ..(1 to N sentences).. F _y ..	Both forms appear freely in the text. One form or both can appear in a document structure (e.g. titles).

The F_x and F_y symbols represent either the SF or the LF. This is done to generalize the expression, because both possibilities can be found in business documents. “P” represents a textual presentation like "also known as" or "henceforth called", etc. The characters used in the context description represent all the standard (as in Larkey *et al.* (2000)) as well as the more creative separators: (), { }, [], < >, “ “, << >>, - -, commas, etc. The symbol “...” denotes a flexible number of words (from none to any number) which does not participate in either Fx, Fy, or P.

4.3 Specific properties of business documents

Detailed analysis of recent research on acronym extraction reveals many variations and restrictions which limit their successful application to business documents. Researches in field of standardization Kabak and Dogac (2010) for electronic business documents outline the fact that different organizations have different needs for their textual artifacts which result in the lack of a possible single norm for them. Besides the obvious fact that these methodologies and techniques were designed for different needs, which they usually fill very well, other differences shed light on new challenges. These new elements may explain why the developed methodologies do not always provide the expected performance level when applied to business corpora.

4.3.1 Main differences

The following table summarizes the main differences between business documents and two other types of corpora used in acronym's extraction research.

Table 4.3 Attributes of major corpora's types

Attributes	Business document	Biology-Medical	Technical
Corpora content	Task description, Procedures, Mission description, Norms, rules and regulations	Article's abstracts	Abstracts, First paragraph
Typical source	Document management system	Medline	RFC
Document's length	One to hundreds of pages	250-350 words	A few sentences
Type of quality review	From none to peer review	Peer review enforced	Peer review enforced
Redaction norms	None to strong (rare)	Strong	Strong
Publishing cycle	Can have one to many authors creating and editing the document during a short to long lifespan. Can published only once (rare), multiple times or after each modification.	A few authors (1 to 5) write the document to publish it once, usually as soon as possible.	Document is written by 1 to many authors and can be published from once to many times during his short to medium lifespan.
Use of acronym	Explicit, Implicit, Bilingual (for non-English)	Explicit	Explicit
Structural features	- Incomplete sentences - Multiple level lists- Deep level main structure	Mostly composed of complete sentences	Mostly composed of complete sentences
Negative detection	High	Low	Low

While this can never be a complete survey of the presented corpora's type (as new exceptions can always be found), it still provides an adequate view of the main differences between them. The type of quality review, the redaction norms applied and the publishing cycles are not considered challenges per se, but they help explain why different types of corpora are more or less structured or adapted for analysis. The negative detection rating summarizes the odds that an unspecified extraction method would of an invalid SF-LF association be considered correct.

Three major differences between types of corpora were identified. An important feature is the difference in length between the two types of documents. As seen in the related work section, most research experiment with short abstracts or single paragraphs taken from a scientific paper. It is fair to assume that a 250-word abstract, which must summarize a 20 pages document, is written in a more abbreviated style than a business text. Apart from internal memos or news, very few document types can usefully be as short as the abstract of a published article. Therefore, it is assumed that documents created by an organization may contain from a few pages to hundreds of pages.

Longer documents would not be useful or easy to read if they were composed of a single unsegmented block of text. To make them readable, the author must separate the block of text into paragraphs and use titles and subtitles, include multiple elements in lists when necessary, add tables and images to illustrate some information more clearly, and provide a table of contents, a glossary, appendices, and so on. This semi-structured aspect is both an essential feature for the human reader but a challenge for acronym extraction, which is an issue that is not encountered in short textual extracts.

It is also important to note that biomedical and technical abstracts (and other scientific publications used for acronym extraction corpora) offer a near encyclopedic text quality, which is not always the case in day-to-day business documents. Every organization creating business documents, and even every department within an organization, may rely on different standards (if any), and authors will not necessarily write with the same rigor and quality. This can be observed in many ways in a document: randomly truncated sentence structures from bullet list header, creative emphasis characters for special terms, unpredicted or non standard presentation of acronyms, repeated large chunks of text, in-house neologisms, capitalized titles or words underlined for emphasis, etc. All these characteristics can hinder acronym extraction efforts because they are unpredictable and informal phenomena for which clear rules of interpretation are hard to pinpoint.

4.3.2 Extraction challenges

The above mentioned differences give rise to many challenges. These challenges can be instinctively linked to one or more of the three causes, although the real dependency ratio of each would be subject to debate.

Implicit presentation of acronyms presents a new challenge to acronym extraction methods designed for business documents. Fluctuating with the document's author style, the observed ratio of implicit presentation in the evaluation corpus was approximately 10%, with the majority of these public organizations being considered to be "reliable" authors. This is an issue which can scarcely be ignored for a thorough extraction and one that is accentuated by the fact that longer documents often contain many more references to a specific SF. If no explicit introduction of an acronym is detected in a text, each of these instances becomes a candidate for an implicit presentation and must therefore be analyzed. Depending on the size of the search scope of the chosen method, the analysis of all these instances can result in more matching errors between the SF and the surrounding LF candidates.

The proximity of the acronym's forms also differs widely in business documents. Possibly because of the summary nature of abstracts, current methods typically assume either no distance between forms or a length-based variation. Schwartz and Hearst (2003) exclude the offset between SF and LF, and Park and Byrd (2001) uses the minimum between twice the length of the SF or 5 more than its length in token. Based on the corpus used in this research, which is considered relatively formal in terms of business documents, only 64% of the SF-LF instances are covered by Park and Byrd formula, and 60% for Schwartz and Hearst (2003). These distance parameters may also need to change if they are used on languages other than English, as it would be the case for our corpora. Changing this search distance parameter may generate new problems that were not implemented in the original algorithm, like choosing the better answer among more than one matching expressions.

As discussed in the previous section, an acronym can be presented initially in a text using many different techniques. But other presentation devices also exist: referential ones (like a

footnote or a glossary), and graphical hints (italicizing, underlining, or bolding the SF or LF in the text). Graphical hints are not used in any known research, probably because of their low frequency and the complexity of retaining the formatting features during text analysis. This discrepancy among presentation norms makes acronym detection more difficult, because the analysis method must cope with new and unusual cases.

Extraction methods requiring deep syntactical analysis may also find business documents troublesome. Partly because of their semi-structured nature, these documents can contain many sentences that most parsers are unable to process without errors: sentence-like titles, truncated sentences, relevant content in tables, etc. While it varies from one author to another, average sentence length is typically longer in business documents than in biomedical texts, with some spanning an entire paragraph. This often results in the wrong label being applied to a normally well-known token or to badly linked sentence segments. Although most of the targeted elements might not be included in these types of structures, this phenomenon might result in missed entities or false positives.

4.3.3 Special cases

The strength of the link between an SF and an LF can also be a challenge to the extraction process, as an acronym can be based on many features. As seen in 4.1, linking letters in the SF to words or symbols in the LF brings many phenomena into play. But the LF can also contain other unrelated or unlinked tokens, which may interfere with the analysis. Content or function words in the SF may or may not be ignored, the SF's letters may refer to letters in the middle of a word or to a missing word in the LF, a common or well-known ending of an LF might be omitted (like the country or state name at the end of a government department), etc. The following are examples of cases where the unreferenced last words of the LF are nonetheless part of the acronym concept.

(1) PLATON: PLATE-forme Ouverte pour les Nouvelle générations de communications mobiles

(2) ACTIF: Association pour la Coordination des Techniques d'Information et de Formation des personnels sanitaires et sociaux

Also, there are some cases, mostly in non English business documents, where one form (short or long) is in English while the other is in another language (French, in our research). This can occur for many reasons: historical or terminological evolution (well-known Anglicisms, like "management", which was accepted in French instead of "gestion"), easier pronunciation (the English UNESCO is written ONUESC in French), keeping a name better known in another language, and so on.

The following table presents a (non exhaustive) ranking of the link type that may be encountered in corpora used in acronym extraction studies.

Table 4.4 Acronym's type ranking

Estimated Extraction challenge	Type	Description
Low	Trivial	Complete letter-to-initial correspondence All LF words referenced in the SF
	Trivial with function words	Same as for trivial, but with an unreferenced function word
Medium	Non trivial	Non initial letters used Number or symbol possibly present Function word referred in SF
High	Partially matching, multi-lingual	SF and LF not in the same language, but some initials the same in both languages.
	Partial covering	Official LF begins or ends with words not represented in the SF
Very high	Code name	Few relations, or none, between SF and LF, Often clearly introduced on first use
	Bilingual, without matching letters	

The estimated extraction challenge rank considers the strength of the link between the SF and LF, and can be read as the inverse indicator of the link's strength. The code name type is an SF-like expression with an official non acronym description with which the expression can be used interchangeably (see 3 below). Some code names are regulated by strict rules, like chemical

substances or compounds, or a correspondence list in a company project's name database or an airport code repository. Others code names can be unregulated, like arbitrary name given to projects or systems (see 4 below) where the short form does not explicitly refer to the long form. This type of expression may be nearer to the named entity phenomenon than an SF-LF one. Although they fit into the broad definition of an SF-FL link, the SF may not "come" from the LF, which makes it less an acronym-type problem. But they are presented as such because they can be presented and used in the same manner as any acronym in a text. Of course, one of the difficulties lies in the fact that we do not know upfront with which type an acronym is associated.

(3) Infoterra: UN global environmental information exchange network

(4) Rosario: Visual Studio Team System Web Access 2010

Aside from all the challenges posed by the various SF-LF cases, most algorithms being developed need both forms to identify a valid acronym, as they obviously cannot work if both expressions of the concept are not present in the local context of occurrences or in the document. Unpresented acronyms are often used when the LF is common knowledge among readers. For example, internal news bulletins will often use an SF without describing it either in the title or in the body of the text. In other cases, the author may use both forms interchangeably, presuming that the reader is familiar with the synonymous link, with both forms outside the search area of the method. They can also be unpresented, because they are part of a larger concept, as in some company names, e.g. "Metal and steel H.S.", "TSX Inc.", etc. It may be difficult for algorithms to solve this type of scenario using the same approach as for other challenges.

At the same time, some SFs may present more than one meaning, being contextually polysemous. While it is unlikely that an organization (except a really large one) would create this type of expression, it can still be encountered in the extraction process. There may be other cases where the same SF is resolved differently in two different documents. This could happen if one is unpresented and erroneously resolves on a matching expression and the other case

is presented. Alternatively, both meanings of a polysemous acronym could be missed, if one meaning is correctly presented and the second meaning is used later without presentation.

Taking into account the many challenges posed by business documents, it is easy to see that acronym extraction differs from the process applied to date in other studies on various corpora.

4.4 Methodology

4.4.1 Overview

We try to cover acronym detection from the most trivial case to the more difficult bilingual SF-LF case with matching letters, as set out in Table 4.4. We leave out code names and bilingual acronyms without matching letters, as we feel that they could be better processed with other methods. We cover all acronyms with at least two characters.

Almost all acronym extraction studies adopt the same logical structure: first, identify potential SFs, and then try to find an appropriate LF in the neighboring context of each candidate SF. The high-level structure of our approach is thus similar to that of other acronym extraction systems based on machine learning presented in the section on related work. The main difference is the extraction of similarity features to be used during classification to help determine whether or not an SF-LF pair is valid. These similarity features are based on a reference repository built for this purpose. Figure 4.1 illustrates each logical step and the resources required.

4.4.2 Preparation step

This step is designed to enrich a raw document to make it usable by our approach. As part of our larger project, Binocle, this step was already implemented and functional. Raw business documents are inputted to a system based on GATE (Cunningham *et al.*, 2002) which tokenizes, splits sentences, and adds a part-of-speech tag to each token using TreeTagger (Schmid, 1994), a POS-tagging tool. A small script is also executed to correct obvious and unambiguous tagging errors which are left out by this tool in French documents. The relevant information required

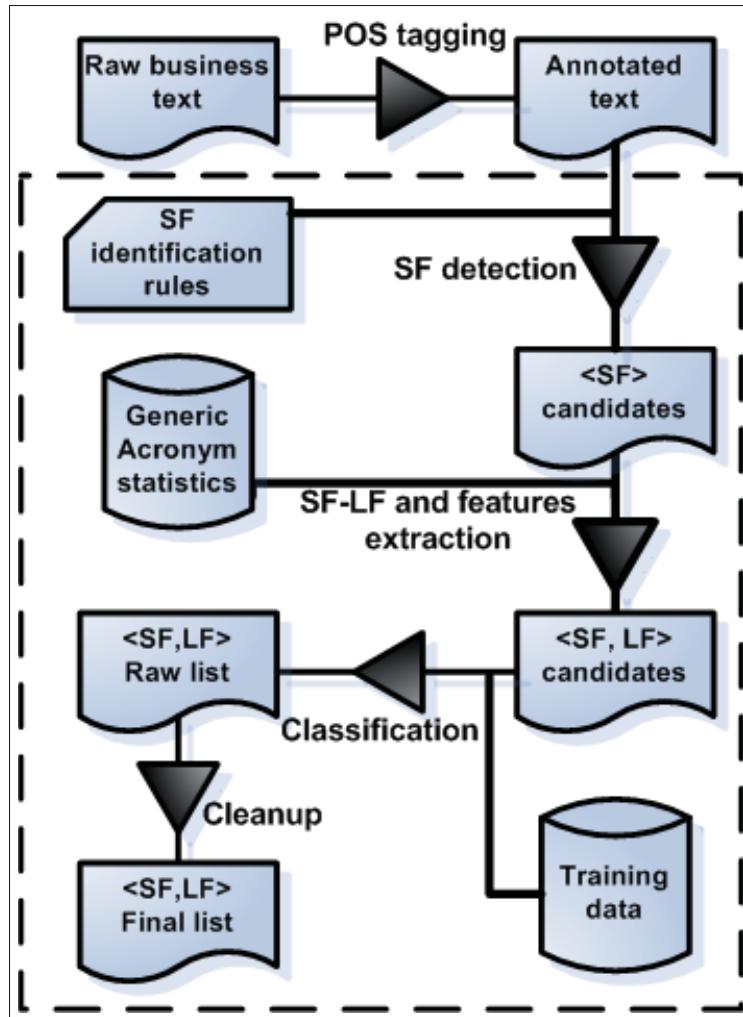


Figure 4.1 Processing steps and resources

for the other steps is as follows: whether or not a token is recognized or unknown, and the starting and ending position of a sentence or text structure (title, chapter name, etc.). The TreeTagger tool, trained on a large French news corpus, tags as “unknown” all the tokens that are not validated with respect to its decision tree.

The preparation step can also include named entity (NE) detection and tagging. The correct detection of those entities can lower the error rate later in the process by removing known and easily recognized patterns from the list of potential SFs. These patterns can target many different types of entities encountered in business documents: locations, addresses, phone numbers, IP addresses, postal codes, Web addresses, typical document file types (“PDF”,

“XLS”, “DOC”, etc.), abbreviated dates (“Dec. 18, 2002”), company stock exchange code names (“SFXC:FD”), coded document names, references to specific laws and their subsections (“L.R.Q. c.4 P-123.2”), processes or directive names (“95/46/CE”, “A/RES/48/13”, etc.), forwarding and copy information at the bottom of official letters (“c.c.”), and so on. They can be detected with a first run on a corpus to identify potential patterns, or they can simply be pointed out by a domain expert, if one is available.

4.4.3 Generic acronym repository

This database is used in the LF extraction step of the process to calculate the “similarity-based” features. It was constructed with publicly available acronyms from various websites and documents. None were the same as those targeted in either the development or the evaluation corpus.

The database contains 9,525 French acronyms (85.3%) and 1603 English ones (14.4%). The remaining few (0.3%) are multilingual acronyms, or SFs and LFs in different languages or in code names. They range from two-letter acronyms to acronyms 12 letters long in English or 11 letters long in French. Acronyms from length 9 to 12 represent only 0.21% of the database. Figure 4.2 show the distribution of acronym length for the whole database.

We focused our manual acquisition process mainly on French acronyms, as the business documents in our corpora were all in French, except for some rare English sentences. As the composition of acronyms was thought to be different in those two languages, we needed a relevant number of both types to produce a useful feature. French expressions, as they get longer, need more function words (conjunction, preposition, etc.) than their English equivalents. The total number of tokens used for the LF in French was supposedly higher than the average number for the English LF for the same length of SF. The two graphs in Figure 4.3 correlate this assumption.

Both languages seem to follow the same curve, although on a different level, for an acronym 2 to 8 characters long in relation to LF length in number of tokens. This makes sense, as longer

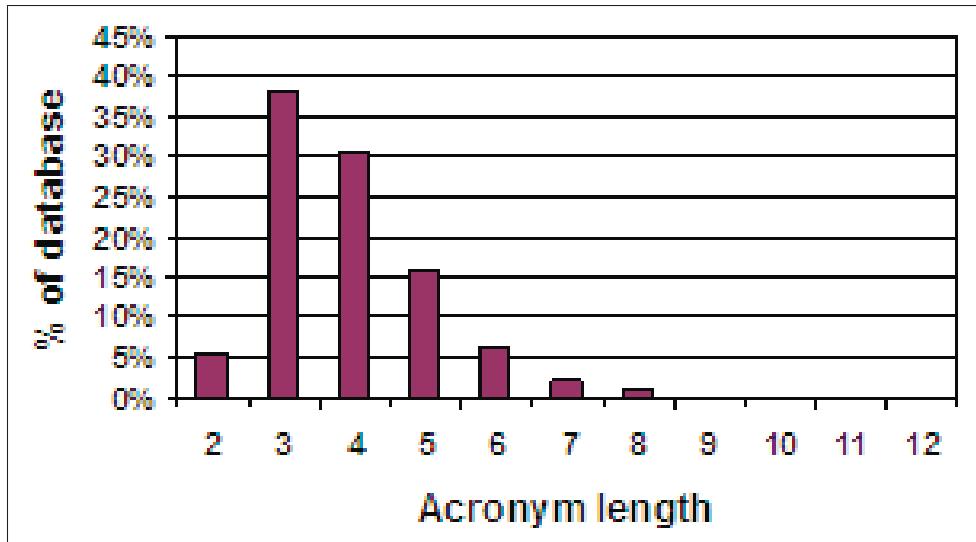


Figure 4.2 Acronym length profile of the generic database

expressions may need more non relevant words (conjunction, preposition, etc.) to coordinate relevant words in the description. Lengths from 9 to 12 characters do not follow the same curve, because they do not include many examples (total of 9 for both languages) in the database. Although they are not as statistically relevant as the shorter ones, we still represent them, in order to provide a complete picture of the generic data being used in the experiment.

As shown in table 4.5, not all acronyms are trivial cases where each word's first letter is taken and then all the letters are strung together.

Table 4.5 Number of non trivial feature acronyms by language

Feature	French	English	Multilingual	Total
Total	9523	1605	24	11169
SF and LF not in the same order	22	2	7	31
First letters of SF and LF are different ¹	441	34	14	489
Unreferenced letters	89	34	10	133
Description includes comma or colon	310	31	2	343
Recursive (see 4.4.3)	0	4	0	4
Containing non initial letters	1465	330	7	1802
Average SF length (in characters)	3.97	3.53	-	-

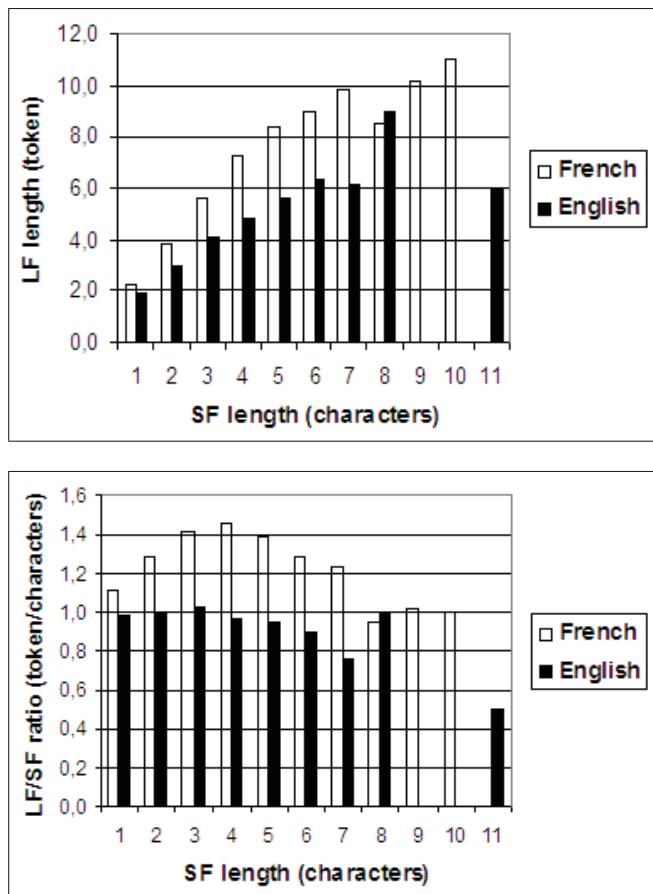


Figure 4.3 Average ratio of LF to SF length

For each entry in the repository, data were extracted from the SF and LF by automated and semi-automated processes to be used in similarity features; length in tokens, number of content words, number of function words, etc. Information about the plausible position of each SF character or symbol referenced in the LF was recorded and verified manually. For example, the National Aeronautics and Space Administration (NASA) would give the position set {1 10 26 32} (see 4.4.3 below) by this standard. Of course, {1 2 26 32} (see 4.4.3) or even {1 15 20 41} (see 4.4.3) would theoretically be acceptable, but it is very unlikely that a human reader would give those interpretations. Multiple position sets may be acceptable when two consecutive words begin with the same letter in the LF for one specific letter in the SF (see 4.4.3). In this case, all the sets are registered in the database.

¹All “codename” cases can be included. The database does not contain many examples, but some organizations’ documents might contain a large number of them.

- (5) SPARQL: SPARQL Protocol And RDF Query Language
- (6) National Aeronautics and Space Administration: {1 10 26 32}
- (7) National Aeronautics and Space Administration: {1 2 26 32}
- (8) National Aeronautics and Space Administration: {1 15 20 41}
- (9) LIEF: “Ligue des Infirmiers et des Infirmières de l’Espace Francophone”: {1 11 46 53}
and {1 29 46 53}

4.4.4 Short-form identification

In the first step, annotated texts are processed to target potential candidates for the short portion of an acronym. We try to tag all uses of acronyms in a document, as we cannot predict at this point which, if any, will present a link (either explicit or implicit) to the LF.

The search for an SF is based mainly on the part-of-speech information tagged (similar to Pustejovsky *et al.* (2001)) into each document by TreeTagger. Having been trained on a large news corpus, it is hypothesized that most non public acronyms found in business documents will be tagged as unknown tokens by this tool. We use this TreeTagger behavior to target relevant candidates for the next step.

A contextual grammar based on the classification shown in Table 4.2 is used to identify tokens with the following patterns:

Unknown tokens can be all uppercase, all lowercase, or mixed-case. The grammar is also built to detect older-style SFs with dots as character separators. The POS tagger tends to consider them as separate tokens and therefore tags them separately. Candidate tokens must be longer than 1 letter, excluding inner separators (dash, dot, slash, etc.).

The level 2 presentation expressions are defined in a customizable list of lemmatized expressions to augment the detection power relative to plain string recognition (e.g. [adverb] [0 to

Table 4.6 Identification rules for an SF

Level	Description
1	An unknown token enclosed between separators (parentheses, commas, dashes, double quotes, etc.) without any other token inside the separator e.g. “... for each named entity (NE) which ...”
2	An unknown token either enclosed in separators with other tokens or introduced with a presentation expression in close proximity. e.g. “... for each named entity (also commonly called NE) which...” “... for each named entity (which will be referred to as NE in this document) which...”
3	All remaining unknown tokens

N tokens] [call/know/refer] [0 to N tokens] [SF]). The lemmatized versions of relevant expressions are used to avoid adding all tense variations for expressions containing verbs.

4.4.5 SF-LF candidate extraction

Using the SF candidate list from the last step, a local search is executed to extract neighboring LF candidate and their attributes. The scope of this search is delimited by a user-defined number of sentences called the search space value. This value indicates the number of sentences before and after the sentence where an SF candidate is found. For a value of 2, a total of 5 sentences could be searched, if available. The search can be limited, if the sentence containing the candidate is near the beginning or the end of the document.

Before the search takes place, an acronym expansion function generates all possibilities that are considered plausible. This function covers most of the examples presented in 4.1 for replacement-type features like symbol to word, number to word, and numeric character multiplier. Constraints are in place to reduce the expansion scope to realistic cases. For example, numbers in acronyms are used to generate a word replacement (in French for our case) if possible (2: to, two, second, 4: for, four, fourth) or repetition letters or set of letters (TE2X will generates TEEEX, TEXX, and TETEX), but high numbers are not used for repetition in expressions like SQL92 or G9. The proposed limit for a numeric character multiplier is 7, although no case of repetition higher than 4 has been encountered so far.

With each candidate SF (and for each expansion of it, if any), a context-restricted search takes place. The sentence segments found before and after the SF are searched first. In the case where the SF should begin or end a sentence, only the corresponding segment is searched. A variable (user-defined) number of sentences placed before and after the first one are then processed. A search in one direction (to the right or to the left of the SF) is stopped if the same SF is encountered. This stopping condition might exclude recursive acronyms, but, as these are extremely rare, the advantages outweigh the risks. The search may continue in the other direction, if it is still unprocessed.

The search process starts with the application of a string-matching algorithm (an adapted version of the KMP Knuth *et al.* (1977) algorithm in this case) which returns, for each letter or symbol in the SF, a set of positions of the matching characters in the LF. A null entry is then added to each position set to simulate the possibility of an unreferenced word. Using the entries of each position set sequentially, the complete list of all the possible combinations (ordered or unordered) is generated independently for each segment or complete sentence. The only matches rejected at this point are the ones composed entirely of null entries. A short example is demonstrated in Figure 4.4 using a two-character acronym. All the possible matching letters are underlined in the LF and the resulting position sets are shown with the underscore as a place holder for the null value. All the ordered and unordered combinations, excluding the all-null instance, are listed in the following figure.

Two exclusion rules are then checked against each match. Candidate LF expressions, as defined by the generated position set, ending with a function word (appearing in a user-defined list) are rejected as incomplete (see 4.4.5). Expressions beginning with a function word are rejected if their first letter is different from the first letter of the LF (see 4.4.5). Features are then extracted, for each match that is assessed by these rules, to be used in the next step.

(10) CE: comité exécutif du (executive committee of)

(11) CE: le comité exécutif (the executive committee)

AJ : <u>Auto_Journal</u>		
A : { 1, 11, _ }		
J : { 6, _ }		
Pattern	Matches	
A x J	1	6
A : { 1, 11, _ }	1	<u>6</u>
J : { 6, _ }	11	<u>6</u>
	11	<u>6</u>
	<u>6</u>	<u>6</u>
J x A	<u>6</u>	1
A : { 1, 11, _ }	6	11
J : { 6, _ }	6	<u>1</u>
	<u>6</u>	<u>1</u>
	<u>1</u>	11

Figure 4.4 Match-generation example

4.5 Candidate SF-LF evaluation

This step consists of a machine-learning supervised classification algorithm, which takes feature-enriched SF-LF candidates as input and decides whether or not they are describing a valid or invalid link between the two expressions. Most features used are comparative in nature, either related to the generic acronym database, to an SF length-related best-case scenario, or to linguistics statistics. A total of seven features are used.

4.5.1 Structural features

The first feature is a ratio between a composite score based on the SF-LF relation and the maximum score it could get in a best-case scenario. The composite score is obtained by the

addition of points associated with the acronym's characteristics, commonly used in, or inspired by, other classifier-based works. They are either linked to the placement of an SF letter in the LF or the arrangement of the letters in the expression, namely:

Table 4.7 Subcomponent of the potential score feature

Type	Points	Characteristic
Single letter	A	Capitalized first letter in a token
	B	Capitalized non first letter in a token
	C	Non capitalized first letter in a token
	D	Non capitalized non first letter in a token
	E	A content word
	F	Proximity (next to another referenced letter)
Whole expression	G	All SF letters ordered in the LF
	H	All SF letters found in the LF

The points given for each matched characteristic can be modified to adapt to a specific language or acronym type. A list of token types to identify content words, based on the POS tags, can also be parameterized. The optimized divider score is obtained with the following formula:

$$\text{Best case} = (\text{len(SF)} * (\max(A, B, C, D) + E)) + G + H$$

The “len(SF)” expression is the number of letters in the SF, and B is the single letter characteristic that gives the highest number of points. The first part of the formula suggests that each SF's letter would designate a content word, and would be a highly rewarding capitalization and placement type. The points for having found all the SF's letters in order in the LF would then be added to the sum. The ratio for this feature can range from 0 to 100%. The example in Table 4.8 shows an LF for which each non capitalized reference initially receives the C and E point value, as well as the G and H points, because all letters are referenced in the same order as in the SF. The best case shown in the last row assumes that value A is greater than value B, C, or D. The expression's total value would then be divided by the total value of the best case, resulting in the final value for the potential score feature of this specific SF-LF candidate.

“... with the federal bureau of investigation (FBI).”

Table 4.8 Potential score ratio example

	Points given
Expression:	$3 * (C + E) + G + H$
Best case:	$3 * (A + E) + G + H$

The second feature, “level”, is a rank associated with the introduction patterns in Table 4.2. It can be viewed as the level of confidence that an SF is a formally introduced acronym. The ranking goes from 1 to 3, with the first level corresponding with the first explicit patterns in Table 4.2. The second level includes the two other explicit patterns that include a textual presentation. The third level includes the 3 implicit patterns from the same table, as they are the least likely to introduce an SF or an LF into the local search space.

The third feature, “triviality”, is designed to indicate when an SF is likely to be mismatched with an LF. It was inspired by the observation that an SF composed of frequently occurring letters (for a specific language) can be easily associated with a non acronym expression which completely fits the initial letters in the SF. The value of this feature is the product of each letter’s probability of occurrence in a generic text. The formula 4.1 represent the calculation where L is the number of letters in a candidate SF and $f(i)$ is the language-related frequency of the letter at position i . The symbols and numbers are omitted, and it must be calculated independently for each language. The four most common letters in English are, in order: “e”, “t”, “a”, and “o”, but these positions are occupied in French by: “e”, “s”, “a”, and “i”, all with different probabilities. By using the product of each letter’s frequency, the resulting number also indicates the likelihood that the specific set of letters will appear in an expression. Thus, a high value indicates that the corresponding SF could be more easily mismatched than one with a lower triviality values.

$$\prod_i^L f(i) \quad (4.1)$$

4.5.2 Similarity features

Three other features are calculated by comparing a value extracted from an LF candidate and building a ratio with the average value of an SF, all in the same language and with the same length in the generic acronym repository. These are designated as “similarity” features. The first two are the number of “function words” and the number of “content words” found in the LF, which add up to the total number of tokens in the expression. These numbers are divided by the corresponding average calculated from the repository to obtain the final ratio which differs from the non-comparative number used in several other researches. The third similarity feature is a ratio called “coverage”, and defines how regularly an LF is segmented by each letter of the corresponding SF. It was introduced as a way to potentially eliminate cases where the matching letters in the LF would be outbalanced but still generate a high value for the potential score feature. As shown in figure 4.5, the acronym ILO could have the same potential score for "Iceland representative for International Labour Organisation" and "International Labour Organisation" but the coverage value would be different. It summarizes the information on the segment in a single value by calculating the average ratio between an LF segment's length ($\|seg_i\|$) and the average segment length ($\|gen_i\|$) of generic acronyms of the same length (and in the same language) found in the database. To prevent ratios from canceling one another out, they are kept in a range of 0 to 1 by using the highest value (between the LF length and the generic length) as the denominator. The calculations can be expressed in the following formula:

$$C = \frac{\sum_i^j \frac{\min(\|seg_i\|, \|gen_L\|)}{\max(\|seg_i\|, \|gen_L\|)}}{\|SF\|} \quad (4.2)$$

where k is the segment number, L is the corresponding SF length, $\|seg_i\|$ is the length of the segment i , and $\|gen_L\|$ is the average length of the segment for an acronym of length L . If the value of L is unavailable for a specific acronym's length, the standard segment's length is used, which is the current LF length divided by the SF length. The minimum and maximum functions enforce the range of limitation of the ratio for each segment. Figure 4.5 shows two

examples with the acronym of the International Labour Organization (ILO), matching on the initial letter of each word. The real length (seg_i) of each segment is shown in the upper section, and the average length (gen_L) of 10, calculated from the long instances of 3 characters in the generic database, is aligned below the LF. The effect of the minimum and maximum functions can be seen in S1 or S3, where the real length is the denominator, compared to S2, where the average length is chosen instead. The three ratios are then divided by 3, which results in a coverage ratio of 0.7492 for the first expression's specific match and a lower 0.5924 value for the outbalanced second example.

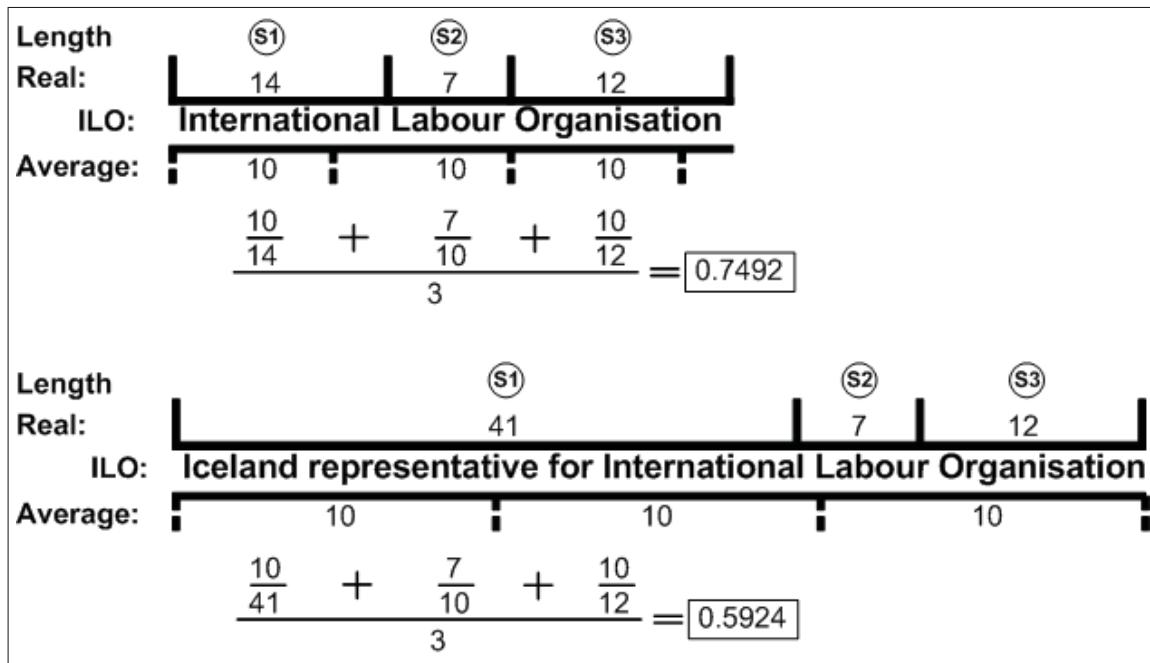


Figure 4.5 Coverage calculation examples

The last feature, “distance”, is the number of tokens between the SF and the candidate LF. It is calculated between the SF and the closest token in the LF (e.g. “... for each curriculum vitae which will be called CV for...” would yield a distance of 4 tokens). This number is always positive, regardless of the order of appearance of the two forms. This is the only feature that relates directly to the text instance and is included in most machine-learning experiment for acronym extraction.

These new similarity features target the challenges offered by acronym's extraction on business document's corpora as a whole. As explained in previous section, longer unstructured texts can present more instances of loosely coupled implicit acronym pairs. The search for these implicit pairs warrants a thorough generation of cases which can trick traditional approach into erroneous assertions. As these candidate cases can be found in many different forms, it became apparent that the evaluation process needed a way to relate to a more generic definition of an acronym. This "common knowledge" of the way SF and LF are linked together may help classifiers rule out instances that fit traditional characteristics but fail to match the generic essence of this link. More specifically, some challenges can be traced back to the presented similarity features. For example, the potential score feature can help the detection of bilingual acronyms by loosening the link between traditional same-language SF and LF feature like having the same first letter. The coverage and word's type related similarity features support the potential score ratio by eliminating invalid candidates who do not associate adequately with the common acronym definitions. Finally, the triviality feature can help rule out instances in which small SF composed of frequent occurring letters are linked to non-relevant LF with high potential score value, which is typically extracted from series of incomplete sentences. For example, multiple single word titles in table of content or item's enumeration in lists, which are natural occurring features in the targeted corpora's nature, can generate these invalid candidates.

4.5.3 Cleanup step

In the final step, a heuristic is used to eliminate false positives in the raw SF-LF list by choosing a single entry for each distinct SF in a document. We hypothesize that each distinct SF in a document has either one definition or none, which excludes the case of a polysemous acronym with more than one meaning in a given context. This step is necessary because the classifier used is not designed to restrict the output to a single best answer for each distinct SF.

Proceeding by document, each distinct case of an SF with more than one selected LF is identified. Then, for each LF, the one with the higher "level" number is kept. In the case of a tie, the

LF with the highest potential score ratio is kept. These ratios can be compared because they belong to the same SF. Ties of both highest score and level would be outputted as two possible interpretations and would be marked for revision. The result of this step is the final list of SFs with their corresponding LF for each remaining acronym.

4.6 Experiment

We tested our approach using the standard data mining method. We used a development corpus to develop our model, then trained the machine learning module, and finally tested on a final corpus which had remained hidden from view during the entire development phase. The results were compared with two other methods, Biotext Schwartz and Hearst (2003) and the Alice system Ao and Takagi (2005). The former is designed to extract acronyms from biomedical abstracts and the later is a rule-based system which, in addition to process Medline corpora, can accept unstructured documents. The biomedical field has been one of the most prolific for acronym extraction in the past few years. It is also among the few where some algorithms and systems were publicly available to test new corpora. As opposed to many other researches which can compare their results using the same corpora, we could only use the top performing publicly available systems and algorithms to apply on our specific corpora. Prior to the performance tests, some parameters and values needed to be defined.

4.6.1 Parameter and score template definition

A score template was defined before the potential score feature could be extracted. Using the position set for the SF-LF reference for each entry in the generic acronym database, a randomized iterative series of tests was developed to optimize the attribution of points for each subcomponent of the potential score feature found in Tables 11 and 14. In each test, every subcomponent was given a random number of points ranging from 0 to 36. Then, for every acronym in the database, all the possible position sets for the SF-LF reference were generated. Each resulting set was then given points according to the evaluated score template. If the highest scoring set was a registered position set for that acronym, this score template recorded

a success. After all the acronyms had been evaluated, the total number of successes was stored with the template definition. A total of 1,482 tests were run, with the best scoring defining 97.16% of the acronyms in the database. The types of content words used were names, proper names, adjectives, and infinitives. The following table describes the complete score template used in the experiment.

Table 4.9 Score template definition

<i>Code</i>	<i>Characteristic</i>	<i>Score</i>
A	Capitalized first letter in a token	29
B	Capitalized non first letter in a token	4
C	Non capitalized first letter in a token	28
D	Non capitalized non first letter in a token	2
E	A content word	15
F	Proximity (next to another referenced letter)	11
G	All SF letters ordered in the LF	25
H	All SF letters found in the LF	4

We did not use exclusion rules for named entities, as suggested in the section on the preparation step. As the evaluation documents came from different domains, we could not predict which rule would be relevant. We left all the NE unidentified in both the training and the evaluation corpora. The search context for LF candidates was set to 2 sentences, resulting in a possibility of 5 sentences being searched for each SF candidate.

4.6.2 Corpora

Two French business document-based corpora were used during this research. The training corpus (A) was taken from 7 French Canadian government department websites, including adoption, justice, information access, public works, etc. The 69 documents composing this training set included 167 distinct cases of explicitly or implicitly presented acronyms. Documents could contain presented acronyms, unpresented acronyms, or both, or neither. These publicly available documents were manually chosen as they contained at least one acronym, either explicit or implicit. SF candidates identified automatically from the first step added up to 1,360 cases spread across all documents.

Once all the tests, parameterization, and optimization had been completed with the training corpus, the evaluation corpus (B) was provided by a third party, who had compiled another 69 publicly available documents from 8 different organizations: space agency, health department, university, humanitarian agency, research funding agency, etc. We manually identified 143 cases of valid acronym links against 1,399 automatically picked candidates. These valid cases were counted on a per-document basis, so the same SF could be counted once (but only once) for each document, if it were presented either implicitly or explicitly. Document types were also much more diverse than in the training corpus.

4.6.3 Classifier training

We used six tree-based and two rule-based classifiers to test the features with the dataset. All tests were run with Weka 3.6.2 (Hall *et al.*, 2009) Experimenter tool. These classifiers were chosen to test the features with a variety of machine learning approaches. Also, both categories of classifiers offer an interpretable output, either with a tree or a rule set, which can be analysed to better understand the impact of each feature.

The J48 (Quinlan, 1994) is an open source implementation of the C4.5 decision tree algorithm which splits each node of the tree by using information entropy. For each node, the information gain for each feature is evaluated and the value yielding the better gain is added as a new decision branch. As the name implies, the J48graft (Webb, 1999) is based on the exact same step, but with grafting. The REPTree (Witten and Frank, 2005) algorithm is another decision tree implementation based on information gain. It then applies a reduced-error pruning step to simplify the decision. The ADTree (Freund and Mason, 1999), which stands for alternative decision tree, works by creating prediction nodes (single values) at the root and leaves of the tree and decision nodes (conditions) in between. Each instance is then run through the tree, summing the appropriate prediction nodes for each decision. The sign of the total value is then evaluated to classify the instance. The BFtree algorithm (Friedman *et al.*, 2000) generates a best-first decision tree classifier by using binary split for both nominal and numeric attributes and fractional instances for missing values. Splits aim to minimize the statistical dispersion

calculated with the Gini coefficient. The SimpleCart (Breiman *et al.*, 1984) is a recursive partitioning method that builds classification and regression trees for predicting continuous dependent variables (regression) and categorical predictor variables (classification).

The first rule-based classifier is JRip (Cohen, 1995) which is a propositional rule learner which creates new rules by adding antecedents (or conditions) to the rule until the rule is perfect (100% accurate) and then pruning it to optimize the complete rule set. On the other hand, the Ridor algorithm (Gaines and Compton, 1995) creates a default rule first and then generates exceptions for the default rule with the least (weighted) error rate. Then it generates the best exceptions for each exception and iterates until pure.

We used the first corpus to generate positive and negative examples to train the supervised classifier as described in the "SF-LF candidate extraction" section. The 69 documents were processed using the methodology described in the last section, stopping at the classification step. For example, the 1,360 SF candidates produced 57,112 SF-LF raw candidates for a search space's value of 2. These instances were semi-manually (automatically tagged then verified and corrected manually) cross referenced with the correct definitions, which was 167 for the 2-valued search space. A class feature was added to the training and evaluation datasets which was set to "true" for candidate SF-LF pairs that textually matched a SF-LF pair from the official reference list, other non-matching instances being set to "false". The following table contains the number of instances and related valid pairs for each value of the search space from 1 to 6.

Table 4.10 Size and valid entries for each dataset

Search Space	Training corpus A		Evaluation corpus B	
	All instances	Valid pairs	All instances	Valid pairs
1	35152	199	45408	252
2	57112	239	70406	287
3	73855	271	96853	313
4	94794	310	123377	342
5	114850	336	149778	363
6	132410	368	174882	388

The difference between the number of cases in the definition list and that in the training set comes mainly from common/trivial uses of an SF within the parameterized search context of a corresponding LF, which is considered an implicit presentation case. For example (see 4.6.3), if we had an explicit presentation of the “SF (LF)” type with the same SF used one or two sentences later in the text, the two SF instances would correctly point to the single LF, producing two valid entries in the training set. As such, changing the search range parameter could increase or decrease this difference.

(12) “...vise à obtenir une APP (approbation préliminaire de projet) et l’autorisation d’aller de l’avant avec la totalité ou une partie de la phase de définition du projet. À l’appui d’une demande d’APP, les ministères...”

4.7 Results

We compared our approach with the Biotext acronym extraction algorithm (which is publically available from their project website¹) and the online Alice system. These algorithms were initially designed for biological and medical abstracts. Biotext’s performance is equivalent to that of a more complex system, but it also works unmodified on French business documents because it does not use tools like part-of-speech tagging, as long as the authors respect the strict format for which it was designed, mainly "LF (SF)". The Alice system uses a more complex structure and is designed to process formatted Medline articles as well as free text. As the source code was not publicly available, it was difficult to evaluate the language impact on performances, but many valid extracted SF-LF pairs showed that it could process the more linguistically complex cases as well as the simpler ones.

To further reduce the linguistic differences, we modified the 69 evaluation documents by replacing all accentuated characters by their non-accentuated equivalent. The capitalization of each character was preserved (‘É’ switched for ‘E’, ‘à’ for ‘a’, so on), as it is most significant in this type of extraction. This was done to avoid the case where a baseline algorithm could not

¹ <http://biotext.berkeley.edu/>

correctly match together the two characters. We also removed all the French elisions (“l’”, “d’”, “c’”, etc.) which could prevent correct detection of the first letter of a word.

These baseline algorithms were compared with all eight classifiers using multiple parameter variations. Each of these variations was tested on each search space related dataset, resulting in more than 8,000 tests. The recall and precision values were calculated at the output of the entire process for each test presented. The F-measure is calculated with the harmonic mean formula method: $2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$. Inputs for tested classifiers come from the bi-directional sentence search in contrast to Biotext’s backward-only approach.

The following table presents the average recall and precision performance calculated for each search space value from 1 to 6 for the two baseline approach and the eight tested classifiers. The f-measure values are calculated on these averages. For clarity, the custom parameters column shows only the parameters that differ from the standard options in the Weka tool’s default settings. The tested algorithms are ordered by decreasing f-measure values.

Table 4.11 Average algorithm’s performance on the evaluation corpus

	Custom parameters	Recall(%)	Precision(%)	F-measure(%)
Biotext	-	48.64	84.83	61.83
Alice	-	59.32	66.96	62.91
J48	C 0.45, M 6	78.74	89.38	83.72
JRip	F 2, O 4	80.01	84.90	82.39
ADTree	B 12	75.33	90.38	82.17
SimpleCart	M 6	74.81	87.75	80.76
Ridor	F 5, S 2, N 5	75.75	86.39	80.72
REPTree	M 4	73.99	88.53	80.61
BFTree	M 3	74.01	87.22	80.08
J48Graft	U, M 4	69.79	90.73	78.89

Alice lower precision is mainly due to errors on document section with incomplete sentences (like the table of content) which are not ended by a dot, which is quite common in business documents. Both baseline approaches still provide the same order of f-measure over the evaluation corpus.

The following table presents the top performance rating (by f-measure) for each classifier for search space values from 1 to 6. The first line of information in the cells is formatted as following: F-Measure (recall/precision). The second indicates, when necessary, the non-default parameters used to obtain the corresponding performance.

Table 4.12 Algorithm performance per search space

	Search space size					
	1	2	3	4	5	6
J48	89.92 (89.9/89.9) U, M 4	89.90 (90.9/89.1) C 0.25, M 2	87.51 (85.2/90.0) C 0.45, M 5	88.6 (87.2/90.1) C 0.45, M 6	80.74 (74.7/87.9) U, M 6	83.94 (80.5/87.7) C 0.45, M 6
JRip	86.55 (86.9/86.2) F 2, N 1, O 6	86.17 (85.6/86.7) F2, N 1, O 6	86.15 (86.5/85.8) F 4	85.55 (86.4/84.8) F 2, O 4	84.96 (86.4/83.6) O 6	84.1 (84.4/83.7) F 2, N 1
ADTree	90.60 (88.7/92.5) B 9, E 0	84.17 (81.3/87.3) B 7	85.39 (84.9/85.9) B 11, E 1	86.48 (85.4/87.6) B 7, E 0	86.17 (85.4/86.9) B 7, E 0	87.01 (85.1/89.1) B 9, E 0
Simple Cart	87.00 (83.2/91.2)	86.91 (83.4/90.7) A	80.32 (74.0/87.8) M 5	84.47 (81.3/87.9) M 3	83.26 (82.4/84.1) M 6	81.09 (76.7/86.1) M 5, A
Ridor	86.14 (84.5/87.5)	85.38 (83.5/87.4) F 2	84.88 (85.4/84.4) F 5, N 3	85.14 (83.2/87.2) N 5	85.29 (86.1/84.5) F 5, N 3	83.85 (81.7/86.1) N 4
REPTree	85.88 (83.1/88.9)	86.90 (89.7/84.3)	85.19 (82.7/87.9)	86.18 (86.3/86.1) P	83.97 (79.8/88.6) M 4	82.81 (78.8/87.2) M 3, N 2
BFTree	86.13 (83.1/89.4)	84.8 (81.1/88.9) M 3, N 2	82.43 (79.8/85.3) M 3	84.11 (80.2/88.5) M 3, N 6	79.26 (72.3/87.8) M 3	82.21 (79.2/85.5) M 3, N 6
J48graft	86.29 (82.0/91.1) U, M 4	86.49 (82.7/90.7) U, M 4	80.96 (73.4/90.3) U	79.50 (71.4/89.7) U	80.96 (73.0/90.8) U, M 5	82.67 (75.5/91.4) U, M 4

Tables 18 and 19 show the features used for each tested algorithm for the search space 1 and 6 respectively. The number indicates the relevance rank of the feature for the corresponding algorithm. This rank is based on the number of instances classified with the help of a specific feature, with the first rank occupied by the feature which helps classify the more instances. Features not used in an algorithm are indicated with a dash. The average relevance rank of each feature for all the algorithms using it is shown in the last line of the table. The feature

columns in each table are sorted in ascendant order of average relevance rank. The lower the number is, the higher his discriminative power.

Table 4.13 Feature's relevance rank for search space 1

Algorithm	Potential Score	Relevant Words	Coverage	Distance	Function Words	Triviality	Level
J48	1	2	3	3	4	5	-
JRip	1	2	-	3	4	5	-
AD Tree	1	3	4	2	-	-	-
Simple Cart	1	2	3	-	-	-	-
Ridor	1	4	2	3	-	4	-
REP Tree	1	2	3	4	-	-	-
BF Tree	1	2	3	4	-	4	-
J48graft	1	2	3	-	4	-	-
<hr/>							
Average	1	2,38	3	3,17	4	4,5	-

Table 4.14 Feature's relevance rank for search space 6

Algorithm	Potential Score	Relevant Words	Coverage	Distance	Function Words	Triviality	Level
J48	1	2	4	3	5	6	-
JRip	1	2	3	4	-	5	-
AD Tree	1	4	3	2	-	-	-
Simple Cart	1	3	2	4	-	5	-
Ridor	1	2	3	-	-	-	-
REP Tree	1	2	3	-	4	5	-
BF Tree	1	3	2	4	-	5	6
J48graft	1	2	4	3	5	6	7
<hr/>							
Average rank	1	2,5	3	3,33	4,67	5,33	6,5

The classifiers were configured with the same parameters as the top performing settings shown in Table 4.11. We presented ranks based only on the lower and higher search space values because the average slowly shift from 1 to 6 in a even manner. Mostly, as the search space increases, the coverage rank decreases mostly at the same rate that the relevant word ratio increases. The introduction level feature is not used by any classifier for search space 1, but is used by the bottom ranking classifiers (last column of Table 4.14) for search space 6.

4.8 Future work

With the introduction of similarity-based features, composite features, and the untapped extraction strategies linked to business document processing, effort can now be invested in diversified exploration branches. Capitalizing on the adaptability of the method presented here, organization- or author-oriented training to optimize the score template could be integrated to adapt to specific writing norms and styles. User feedback could also be implemented to provide stronger guidelines for the classifier or cleanup step. Existing comparison-based features could be investigated to verify whether or not more relevant links could be discovered between a generic acronym database and the analysis technique.

Because it is part of a modular structure, the classifier module could be fitted with a different acronym-matching module, as some types of business documents may benefit from candidate extraction not based on linguistic information. Other algorithms and techniques could be combined to address a specific class of problem in acronym extraction from business documents. For example, a pattern-alignment algorithm could take care of the straightforward case of SF-LF (level 1), a rule-based module could address the semantic-related level 2 cases, and a data mining approach could deal with the remaining undefined level 3 acronyms. This could potentially boost the extraction performance of the overall approach.

4.9 Conclusion

In this article, we presented a classifier-based method for acronym extraction from business documents using a small features list which included new similarity based features. Many discrete examples were given to stress that documents which contains implicitly defined acronyms cannot be processed with the same approach developed for short samples from technical or biomedical texts. Strong evidence was given for comparison-based features, which perform well in the semi-structured nature of typical business documents. As specific machine-learning algorithms could provide extraction performance of at least 86% for each search space size, it

is thought that they provide a good starting point for the exploration of flexible and efficient acronym extraction strategies with this type of corpus.

Most resources (corpora, list of expressions, generic database) used in this research can be found on the LiNCS website at <http://lincs.etsmtl.ca/binocle.html>.

CHAPTER 5

GENERAL DISCUSSION

While each of our contributions meet the specific objectives of our research, they do not fulfill them completely on every aspect. This is to be expected in this domain as many of the research efforts are dependent of works done outside the scope of this thesis, like existing tools or methods. On the other hand, we acknowledge some limitations of the completed work regarding some aspect of the current contributions or against possible application. This section details those limitations as well as the quality of the contribution against the state of the art. Each subsection is linked to the specific chapter with the corresponding name.

5.1 Concept extraction from business documents for software engineering projects

5.1.1 Performance and usage

The challenge of putting together a pipeline of multiple natural language processing tools is the cross contamination of error from each tool to its neighbor. The process chain have a multiplicative effective on the first error which is transmitted and then reintegrated in the next output, and so on. While the precision is less than ideal for a typical knowledge extraction method, it is still encouraging to reach a stable improvement in comparison to the baseline on such a complex task.

Even if some of the implemented modules are not language independent, they either already exist for other languages (dictionaries, stop word or expression list, etc) or can be easily created or adapted for language other than French. In addition, other filtering modules can be added to improve the performance level of this type of task. The inherent challenge is to raise the precision level while limiting its effect on the recall measure.

The set W , used to define the weights in the propagation phase of the density-based reordering algorithm described in section 2.4.10, was defined on the basis of multiple tests with different

distribution of values. We tried various strictly decreasing and increasing sets as well as skewed value curves to measure their effect on the presented scoring method. As it turned out, the other value sets did not change the order in a significant way in larger documents because relevant concepts were too far down the reordered list to be included within the boundary of the gold standard length (which was used to define the number of concepts to score in a system's evaluation). As a result, the MDR score didn't change on large documents. Nonetheless, the general trend pointed towards a link between the length of document, the ratio of logical structure and the potential relevance score.

The rule that forced annotators to only reuse terms and expressions from the source document limits the degree of expressibility and flexibility of the output in relation to certain possible usage of the concepts. For example, domain models or class diagrams often make use of generalizations or abstractions of specialized classes or concepts. These high level concepts are usually created or named by an expert and are seldom found directly in documents as part of the domain knowledge. In consequence, they cannot always be extracted from documents and will therefore not be found in the extracted candidate list.

5.1.2 Extraction issues

The use of an exclusion list of contextualized stop expressions to remove irrelevant word occurrences may filter out some relevant concepts in specific cases. As the patterns are made to be adaptive in order to capture most variant of flexible or semi-flexible commonplace expressions, they may also adapt to relevant expressions which wording is close to detected commonplace expressions. For example, by applying the same flexibility-enabling placeholders in the stop expression “taking the fast lane” (rushing or doing something very quickly, accelerating something) in the same way as the expression “to gain ground” presented in Table 2.3, a document taken from a company using production lines may contain an expression like “The product is then *taken from the fast lane* and switched to a slower one”. In this scenario, the middle part of the sentence, “taken from the fast lane”, would be matched by the resulting exclusion pattern

but would erroneously remove the occurrence of “lane” (which would probably be considered relevant in this context) from the candidate concepts list.

One solution would be to restrict the flexibility of the exclusion patterns in such way as to only capture known or approved variations, which would require an analysis on a large corpus in order to effectively detect a wide array of valid variations. Another simpler solution would be to remove patterns for which a large number of occurrences is present in the text without falling in an exclusion pattern. This would of course not work for relevant terms or expressions with a single or few occurrences which all falls within an exclusion pattern.

5.1.3 Evaluation method

One limitation related to the adoption of the Pyramid inspired evaluation method is the need for systems to rank their results. This may be problematic for classifier-based methods using binary output class as they do not provide this type of output by default. One workaround for these systems might be to use the confidence level of the model for each entry as a basic ranking order. While it can be argued that this does not represent the relative relevance of one concept compared to the next of rank, it does provide a relative indicator of which entry is closer to the source model based on the features used during the training step of the classifier.

5.1.4 Gold standard

The current gold standard definition as a list of concepts presents an issue when this list is compared to the ordered list of concepts produced by an extraction process (either our process or any other similar extraction process) because it doesn’t take into account the multiple meaning of an expression or its possible context. For example, the term “bank” could mean “a financial institution” in some cases or “a group of components of the same type designed to act simultaneously” in others, both within the same analyzed corpus. The same type of problem can be found for expressions that are properties of two different concepts, like “power” which could mean “magnification power” when associated with viewing lens or a “legislative power” when

attributed to a figure of authority. While such a case did not occur in the current experiment, it could create incoherence in the validation and evaluation process when one of the two meaning is relevant while the other is not. The same problem may arise during gold standard definition in the likely occurrence that two meanings are found to be relevant by different experts.

5.2 Hybrid extraction method for French complex nominal multiword expressions

5.2.1 Performance

The drop in the F1 performance for the revised Laporte corpus might be explained by the size of the corpus which is much smaller than the French TreeBank (FTB). The new reference terms might have a low frequency score, so they would be excluded by raising the threshold by 25%. The FTB corpus being larger in size, the extracted expressions are much less influenced by the higher threshold. Furthermore, at this level of performance, our method doesn't produce noise from irrelevant CMWE which is a desirable property when robust extraction is needed.

Using root lemma from ngrams seems to offer a sufficiently flexible medium for the detection of complex multiword expressions, leaving room to refine the statistical methods for the analysis of the validity of these expressions. Although, because of the rarity of these terms, they do not always have a significant frequency allowing an easy identification. It is therefore more efficient to evaluate their frequency with the growth ratio between a local context (the root of the semilattice) and the global context of the corpus.

Many entries in the original gold standard only appeared verbatim once in their respective corpus. While other types of extractor (like deep-parser methodology) might successfully identify them, statistical approaches, either hybrid or full-fledged, are at a loss because they cannot differentiate them from non-relevant expressions. They were left in both the original and revised gold standard to enable comparison with tools and methods using different techniques.

The algorithm also helps solve another problem associated with basic frequency approach. Considering the normalized nature of business documents, repetition of segments (phrases or

paragraphs) in an almost identical fashion through the entire corpus is common. It can include introduction paragraphs shared by many work procedures, the definition of gender usage in a text, a confidentiality declaration at the end of a document, etc. These sections artificially inflate the corpus frequency for some ngrams and the corresponding CMWEs are then included in the extraction list of straightforward frequency analysis. Calculating the growth ratio of the ngrams as is the current approach therefore diminishes the importance of irrelevant CMWE and removes them from the extracted list.

5.2.2 Sources of errors

Some errors are generated by TreeTagger, even if this part-of-speech labeling tool offers very good performance in French. This creates, at the source, false ngrams that permeate the whole process. Some ngrams that would be eliminated by the acquisition phase rules are kept and analyzed as valid ngrams. A manual study of the selected expressions at the acquisition stage (see Section 3.3.1) shows that this type of error creates erroneous expressions at a rate of 2.7%. About half of these erroneous ngrams are removed from the execution chain by subsequent stages.

Special care must be taken in the choice of words belonging to the stop list. Not enough “boundary” words in this list results in an over production of ngrams and, possibly, irrelevant extracted CMWE. Too many restrictive words can directly eliminate CMWE during the preparation phase by removing them from the generated ngrams list.

Finally, the use of root lemma for long nominal multiword expression might group together non-equivalent occurrences, although this behaviour is counterbalanced by the fact that it offers better results in sparse corpora.

5.3 Classifier-based acronym extraction for business documents

5.3.1 Performance and features

As hypothesized, our approach provides significantly better results than the base algorithm, especially for the recall. This is probably because of the variability of the phenomena used to introduce the acronym in the evaluation corpus. Although the Biotext algorithm cannot represent all the published and unpublished extraction methods, the significant performance level it achieved in its respective field was deemed to make it a good choice as a basis for comparison. Considering both the large variety of business document types used and the different organizations from which they were selected, the overall performance of our method indicates that the small feature list was relevant in helping to classify the targeted cases.

Further analysis reveals some corpus-specific and method-related factors that may explain why higher scores were not attained. Some of the documents, written in French, contained a few English sentences. As the POS-tagger library was for French only, these sentences were not correctly parsed, resulting in erroneous score attribution, in turn leading to false positives. The classification step was also hindered by the fact that the training corpus did not contain many level 2 presentation cases (see Table 4.2), which results in a drop in extraction performance with the same type of scenario in the evaluation corpus. As a result, the second explicit pattern from Table 4.2 was considered as a case of the second implicit pattern, thus classifying a level 2 instance as a level 3 instance which did not use the same decision branches. To accentuate this effect, the small list containing presentation expressions from the training corpus did not overlap many of those used in the evaluated one.

Some ambiguous cases made it to the cleaning step. These were SF-related, with two or more LF expressions for which the potential score ratio (and other features) was high enough to be considered a valid instance. Although the features helped to classify more than 99 % of the dataset's SF-LF raw candidates correctly, they may not be accurate enough for specific ambiguous cases. This slight lack of performance could also come from the score template, which

was optimized on out-of-context instances. The scores could have been adjusted differently if the instances used for training had been found “in the wild”. But the template is still a good discriminator for the majority of acronyms found in the processed corpora.

The choice of the global score feature over its subcomponents was made as a way to soften the link made by the classifier during training. As such, it cannot integrate fact-like classification choices, like “all successes in training data have the same first letter”. The classifier with this global feature is therefore considered more flexible than one trained with the subcomponents. These latter items would also be less intuitive, and therefore less suited for a generalized feature, as they are almost all boolean in nature.

As shown in table 4.13 and 4.14, the potential score is used in priority by both tree-based and rule-based algorithms. The rank of this feature indicates its discriminating strength in comparison to the other features used in regard to the number of instances classified. The two other similarity features introduced in this research, coverage ratio and relevant word ratio, are used by almost all classifiers as second or third rank features, showing their usefulness over other traditional features as distance and introduction type. The function words ratio is much less used by the classifiers. One reason may be that French expressions contain many more function words than in other languages like English. In this case, the classifiers considered that the relevant words outbalanced the function words in most of the expressions. The triviality feature, although ranked lower than the previous ones, is adopted by more classifiers than the function words ratio as the search space grows. It is also used by the top average performing algorithm like J48 and JRip.

As speculated, these features present a solid and stable, but yet flexible, choice for acronym extraction from business documents from many domain. They offer performance ratings of the same order as the order domain’s approach on their relevant corpora. F-measure results in table 4.11 show that these features can be used by many types of algorithms to produce exploitable extraction mechanism. The differences of performance level between tables 4.11

and 4.12 show that restricting the search space to a specific size for a targeted corpus can be advantageous.

We can observe in table 4.12 a diminution of performance as the search space value increase from 1 to 6, or from 3 to 13 in sentences searched, in almost all algorithms. This can be explained by the increase in both dataset size (more than triple the size) and valid pairs to find. But as the submitted and valid instances grow by 54% from search space size 1 to 6, the results of the top overall performing algorithms (like J48 and Jrip) are slowly decreasing but at a much lower rate.

5.3.2 Limitations

Some business documents differ greatly from those in the development and evaluation corpora, which are typically highly structured and contain few syntactically correct sentences. A good example is the forms that are widely used in organizations at key stages of internal and external processes. While they do not convert gracefully into raw text because of their visual and structural dimensions, they often still contain relevant acronyms (fully used in their SF and LF, or used partially in either one) that may be interesting to detect. The linguistic-based candidate detection step may fall short on this type of document because the syntactically incorrect sentences might mislead the POS-tagging tool. This could result in known words being tagged as unknown, which would elevate the chances of them being erroneously recognized as SF candidates containing high-frequency letters (like “the”, “is”, or “this”). Also, as in (13), if the first word is incorrectly tagged as a noun instead of a determinant, it could be falsely associated with the real SF candidate acronyms.

(13) Le Mouvement de Liaison des Associations de Sécurité Routière... (LASER)

The loose SF recognition rules may allow some false SF candidates to be considered for LF searches. An example from our evaluation corpus is the token “DISCUSSIONS”, a subtitle in a text which was not recognized by the POS-tagger tool. This can cause problems during

the SF-LF feature extraction process, because unrecognized tokens may be too long for the similarity data contained in the generic database. If few or no cases are accessible for the same SF length, the features might not be representative. This may lead to longer tokens being wrongly classified.

The final cleanup step excludes the possibility of polysemous acronyms in a single document. An acronym with more than one meaning could still be extracted from a corpus if its definitions are extracted in different documents. If there is reasonable doubt, a list of such acronyms could be produced to be validated by a domain expert.

Finally, there may be a speed issue, if the targeted application needs to quickly access the process output. The source of this issue is the completeness of the contextual search for each SF. Evaluating all the possibilities can obviously generate an enormous number of matches from a medium-length acronym. For example, a 6 letters long acronym with some common letters could match $20 \times 3 \times 6 \times 35 \times 12 \times 7 = 1,058,400$ times in a limited context window. To exploit this technique in a production environment, some improvements should be considered. The issue could be minimized using many strategies: parallelism, generating and testing the ordered possibilities first, continuing with the unordered ones only if none is successful, etc.

GENERAL CONCLUSION

In this thesis, we have addressed four major objectives related to the automated extraction of concepts for software engineering projects. Of these four themes, developing a gold standard and a method for software-oriented extraction processes were the basis for the evolution of the presented system but also a stepping stone for future extraction systems. Hopefully, other similar resources will be developed to enable progress in other languages as well.

Building on this resource, the global processing pipeline was the second step. It was shown, as the basic hypothesis had suggested, that capitalizing on structural hints left by the document's authors enabled a better relevance-based ordering of concepts to be used in software engineering project. It was also demonstrated that a filter-based system produces a less noisy output compared to the main statistical methods available in text mining.

The two specialized methods for multiword expression and acronym extraction were developed to provide a significant increase in performance over similar state-of-the-art methods. It also enabled to shed some light on some missing areas of currently researched linguistic fields. While not trivial in nature, our methods can be reproduced in other languages and, hopefully, provides the same order of performances.

While it was designed for French, the performance upgrade of the developed pipeline is thought to be useful even in systems of the same nature for other languages which may offer better performing tools. It could be, as an example, used on the output of any system that produce candidate concepts by using more complex or specialized tools, in order to reorder or filter the concept list.

Still, many questions remains opened for future works.

- a. Can robust negative modules be developed to increase the precision of the pipeline while leaving untouched the recall factor? Such tools would be a valuable asset to this type of extraction pipeline.

- b. Could this pipeline be extended to suit other purposes in automated software modeling like enhancing the relationship extraction step to define relevant interactions between domain concepts? As concept tuples can appear multiple times in a business documentation, the same noise problem might be found as overgenerated candidates should be eliminated to choose the few relevant relationships.
- c. Can the output of this system be used to define the stereotype of expressions (concept or attribute) using structural hints annotated from this pipeline? This could provide a final step in producing a complete UML domain model for a software project.

Articles in peer review journals

- Pierre André Ménard and Sylvie Ratté, “Concept extraction from business documents for software engineering projects”, Submitted to Automated Software Engineering on April 1st 2014.
- Pierre André Ménard and Sylvie Ratté, “Hybrid extraction method for French complex nominal multiword expressions”, Submitted to Computational Intelligence journal on April 3rd 2014.
- Pierre André Ménard and Sylvie Ratté, “Classifier-based acronym extraction for business documents”, Published in Knowledge and Information Systems, Springer, 2011, Volume 29, Issue 2, pp 305-334. DOI: 10.1007/s10115-010-0341-9

Conference presentation and publication

- “Fondements d’un prototype pour la création semi-automatisée de productions visuelles pour le génie logiciel à partir de documents corporatifs”, 75th ACFAS conference, 2007
- Sylvie Ratté, Wilfred Njomgue and Pierre André Ménard, “Highlighting document’s structure”, International Conference on Computer, Electrical, and Systems Science, and Engineering (CESSE), 2007

- “Incremental n-grams lattice cleanup for complex multiword expressions”, International Conference on Advanced Computer Theory and Engineering, IEEE Computer Society, 2008.

Internship

- Four months internship with Nexa inc. to develop a prototype of concept extraction based on social network analysis using text mining approaches. Financed by the Mitacs-Accelerate internship program.

Awards

- a. FQRNT, doctoral research scholarship.
- b. École de Technologie Supérieure (ÉTS), Internal Scholarship.

Paper reviewing

- Knowledge and Information Systems journal (1 paper)
- International Conference on Advanced Computer Theory and Engineering, 2008 (2 papers)
- International Conference on Information Management and Engineering, 2009 (2 papers)

This work was supported by NSERC grant RGPIN283191-04 to Sylvie Ratté.

APPENDIX I

REVISED MULTIWORD EXPRESSION GOLD STANDARD FOR THE FRENCH TREEBANK AND LAPORTE CORPORA

This section presents all the multiword expressions added as part the revised corpora for the French TreeBank and the Laporte corpora used in Section 3.4.1. For the FTB corpus, the starting list was extracted using all the expressions in the xml file which were tagged with the “compound” keyword. For the Laporte corpus, all expressions contains within “<N> </N>” tags were extracted and filtered on their length to keep only expressions composed of at least five words.

The two following list represent the additional multiword expressions which we thought deserve to be part of the tagged expressions. These expressions added to the original one from each corpus were defined in Chapter 3 as *Laporte_{Revised}* and *FTB_{Revised}*. They are presented separately by length of the expressions.

1. French TreeBank

1.1 Length 5

ambassade du liban à paris
aménagement du temps de travail
baisse des taux d'intérêt
bilans de fin d'année
caisse nationale d'assurance maladie
caisse nationale de l'énergie
calendrier de mise en oeuvre
carnet de commandes d'avions
chaîne d'informations en continu
chute du mur de berlin

comité d'établissement de billancourt
comité d'établissement du cib
conseil des bourses de valeurs
cour d'appel de paris
coûts de main d'oeuvre
département américain de la défense
départements français d'outre mer
direction de l'industrie touristique
économiste de la banque mondiale
entreprise de biens d'équipement
fédération nationale des transporteurs routiers
industrie française de la chaussure
institut d'émission de francfort
livret de caisse d'épargne
ministère soviétique de l'intérieur
président du conseil d'administration
président du tribunal de commerce
projet de loi de finances
sociétés en commandite par actions
système européen de banques centrales
tribunal de commerce de chambéry
tribunal de commerce de luxembourg
tribunal de commerce de paris
tribunal de commerce de rennes

1.2 Length 6

bilan de la banque de france
conseil d'administration de l'entreprise

contrats de retour à l'emploi
 décision de la cour d'appel
 délégation à l'aménagement du territoire
 dépôt de bilan de la chaîne
 gestionnaires du régime d'assurance chômage
 gouverneur de la banque d'espagne
 gouverneur de la banque de france
 indice des prix à la consommation
 loi d'orientation pour la ville
 loi sur les modalités de sécession
 magistrats de la cour des comptes
 pays d'europe de l'est
 président de la chambre de commerce
 président de la commission des finances
 présidents de compagnies d'assurances nationalisées
 référendum sur le traité de maastricht
 réforme de la politique agricole commune
 régime général de la sécurité sociale
 secrétaire d'etat à la mer

1.3 Length 7

abattement sur l'impôt sur la fortune
 conseil d'administration d'aéroports de paris
 cour d'appel d'aix en provence
 discours sur l'état de l'union
 ministres de l'économie et des finances
 obligations à bons de souscriptions d'actions
 pays de l'europe de l'est

président de la commission de déontologie boursière

président du conseil national de l'habitat

projet de loi sur la sous traitance

secteur du bâtiment et des travaux publics

1.4 Length 8

comité d'organisation des jeux olympiques d'albertville

directeur de cabinet du président de la république

haut comité pour le logement des personnes défavorisées

loi sur la maîtrise des dépenses de santé

ministre de l'agriculture et de la forêt

ministre de l'agriculture et de la pêche

1.5 Length 9

président de la chambre de commerce et d'industrie

1.6 Length 10

ministre de l'industrie et de l'aménagement du territoire

projet de loi sur la maîtrise des dépenses de santé

1.7 Length 11

comité d'etat ukrainien pour la sûreté nucléaire et la radioprotection

2. Laporte

2.1 Length 5

conseils d'administration des entreprises
défis de la guerre énergétique
rupture des contrats de travail
union pour un mouvement populaire

2.2 Length 6

ouverture des marchés de l'énergie

2.3 Length 7

émissions de gaz à effet de serre
loi de financement de la sécurité sociale
projet de loi sur l'épargne salariale
projet de privatisation de gaz de france

2.4 Length 9

projet de loi de financement de la sécurité sociale

2.5 Length 10

députés du groupe de l'union pour un mouvement populaire
rapporteur de la commission des affaires culturelles familiales et sociales

2.6 Length 13

rapporteur de la commission des affaires économiques de l'environnement et du territoire

2.7 Length 15

rapporteur pour avis de la commission des affaires économiques de l'environnement et du territoire

rapporteur pour avis de la commission des finances de l'économie générale et du plan

REFERENCES

Berry, MW and J Kogan, 2010. *Text mining: applications and theory*.

Brachman, RJ and HJ Levesque, 2004. *Knowledge representation and reasoning*.

CAWLEY, MC, 1981. *Everything that Linguists have Always Wanted to Know about Logic*.

Cimiano, P, 2006. *Ontology learning and population from text: algorithms, evaluation and applications*.

Cios, KJ, 2007. *Data mining: a knowledge discovery approach*.

Dubois, J, M Giacomo, and L Guespin, 2002. *Dictionnaire de linguistique*.

Fomichov, Vladimir A., 2010. *Semantics-Oriented Natural Language Processing*.

Gašević, D, D Djurić, and V Devedžić, 2006. *Model driven architecture and ontology development*.

Glass, Robert L., 2003. *Fact and fallacies of software engineering*.

Hitzler, P and H Scharfe, 2010. *Conceptual Structures in Practice*.

Indurkhy, N and T Zhang, 2005. *Text mining: predictive methods for analyzing unstructured information*.

Kao, A and SR Poteet, 2007. *Natural language processing and text mining*.

Kendal, SL and M Creen, 2007. *An introduction to knowledge engineering*.

Morley, C, J Hugues, B Leblanc, and O Hugues. 2005. "Processus métiers et SI". *Evaluation, modélisation, mise en oeuvre*.

Sowa, JF, 2000. *Knowledge representation: logical, philosophical, and computational foundations*.

Swart, H De, 1998. *Introduction to natural language semantics*.

BIBLIOGRAPHY

Abbott, R.J. nov 1983. "Program design by informal English descriptions". *Communications of the ACM*, vol. 26, n° 11, p. 882–894.

Abeillé, A., L. Clément, and F. Toussenel. 2003. "Building a treebank for French". *Treebanks. Kluwer, Dordrecht*, p. 165–188.

Abran, Alain, J. Moore, Pierre Bourque, RL Dupuis, and L. Tripp, 2004. *Guide to the Software Engineering Body of Knowledge*.

Adrian, Benjamin, Heiko Maus, and Andreas Dengel. 2009. *iDocument : Using Ontologies for Extracting Structured Information from Unstructured Text*. Technical report.

Aires, José, Gabriel Lopes, and J.F. Silva. 2008. "Efficient multi-word expressions extractor using suffix arrays and related structures". In *Proceeding of the 2nd ACM workshop on Improving non english web searching*. (New York, New York, USA 2008), p. 1–8. ACM.

Altenbek, Gulila and Ruina Sun. dec 2010. "Kazakh Noun Phrase Extraction Based on N-gram and Rules". *2010 International Conference on Asian Language Processing*, p. 305–308.

a.M. Ibrahim. mar 1989. "Acronyms observed". *IEEE Transactions on Professional Communication*, vol. 32, n° 1, p. 27–28.

Anderson, Theresa Dirndorfer. 2006. "Studying human judgments of relevance: interactions in context". In *Proceedings of the 1st international conference on Information interaction in context*. p. 6–14. ACM.

Ao, Hiroko and T. Takagi. 2003. "An algorithm to identify abbreviations from medline". *GENOME INFORMATICS SERIES*, vol. 698, p. 697–698.

Ao, Hiroko and Toshihisa Takagi. 2005. "ALICE: An Algorithm to Extract Abbreviations from MEDLINE". *Journal of the American Medical Informatics Association*, vol. 12, n° 5, p. 576–586.

Baeza-Yates, R. and B Ribeiro-Neto, 1999. *Modern information retrieval*. 167 p.

Bajwa, Imran Sarwar and M Abbas Choudhary. 2006. "Natural Language Processing for Scenario based UML Diagrams Generation". In *18th National Conference on Computer Application*. p. 171–176.

Baldwin, Timothy, Colin Bannard, Takaaki Tanaka, and Dominic Widdows. 2003. "An empirical model of multiword expression decomposability". *Proceedings of the ACL 2003 workshop on Multiword expressions analysis, acquisition and treatment -*, p. 89–96.

Batini, Carlo, Stefano Ceri, and Sham Navathe, 1992. *Conceptual database design: an entity-relationship approach*. 470 p.

Bermingham, Adam and A.F. Smeaton. 2009. "A study of inter-annotator agreement for opinion retrieval". In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. p. 784–785. ACM.

Biskri, I, J G Meunier, and S Joyal. 2004. "L'extraction des termes complexes: une approche modulaire semiautomatique". In *Le Poids des mots (Actes des 7èmes Journées Internationales d'Analyse Statistique des Données Textuelles, Louvain-La-Neuve, Belgique)*, Gérard Purnelle, Cédrick Fairon & Anne Dister (eds). Presses Universitaires de Louvain, vol. 1, p. 192–201.

Blumberg, Robert and Shaku Aire. 2003. "The problem with unstructured data". *DM Review*.

Bonin, Francesca, Felice Dell'Orletta, Giulia Venturi, Simonetta Montemagni, Linguistica Computazionale, Antonio Zampolli, and Dipartimento Informatica. 2010a. "Contrastive Filtering of Domain-Specific Multi-Word Terms from Different Types of Corpora". In *Proceedings of the Workshop on Multiword Expressions: from Theory to Applications*. p. 76–79.

Bonin, Francesca, Giulia Venturi, and Simonetta Montemagni. 2010b. "A Contrastive Approach to Multi-word Term Extraction from Domain Corpora". In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. p. 19–21.

Booch, G, I Jacobson, and J Rumbaugh. 1999. "The unified software development process". *Reading: Addison Wesley*.

Borgida, Alex. sep 2007. "How knowledge representation meets software engineering (and often databases)". *Automated Software Engineering*, vol. 14, n° 4, p. 443–464.

Bourigault, Didier. 1992. "Surface grammatical analysis for the extraction of terminological noun phrases". In *Proceedings of the 14th conference on Computational linguistics-Volume 3*. p. 977–981. Association for Computational Linguistics.

Breiman, Leo, Jerome H Friedman, Richard A Olshen, and Charles J Stone, 1984. *Classification and Regression Trees*.

Butler Group. 2005. *Document and Records Management*. Technical Report February.

Calzolari, Nicoletta, C. Fillmore, Ralph Grishman, Nancy Ide, Alessandro Lenci, C. MacLeod, and Antonio Zampolli. 2002. "Towards best practice for multiword expressions in computational lexicons". In *Proc. of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*. p. 1934–40.

Castro, Lucia, F. Baião, and G. Guizzardi. 2009. *A Survey on Conceptual Modeling From a Linguistic Point of View*. Technical report.

Chang, Jeffrey T, Hinrich Schütze, and Russ B Altman. 2002. "Creating an online dictionary of abbreviations from MEDLINE". *Journal of the American Medical Informatics Association*, vol. 9, n° 6, p. 612–620.

Chen, Jing and Jianfeng Wu. nov 2009. "Improved algorithm for keywords extraction from documents without corpus". *2009 IEEE 10th International Conference on Computer-Aided Industrial Design & Conceptual Design*, p. 2339–2341.

Chen, P. jun 1983. "English sentence structure and entity-relationship diagrams". *Information Sciences*, vol. 29, n° 2-3, p. 127–149.

Church, K.W. and W.A. Gale. 1991. "Concordances for parallel text". In *Proceedings of the Seventh Annual Conference of the UW Centre for the New OED and Text Research*. p. 40–62.

Church, K.W. and Patrick Hanks. 1990. "Word association norms, mutual information, and lexicography". *Computational linguistics*, vol. 16, n° 1, p. 22–29.

Cohen, William W. 1995. "Fast Effective Rule Induction". In *Proceedings of the Twelfth International Conference on Machine Learning*. p. 115–123.

Constant, Matthieu. 2011. "MWU-aware Part-of-Speech Tagging with a CRF model and lexical resources". *ACL HLT 2011*, , p. 49–56.

Crowder, Richard and Yee-Wai Sim. 2004. "An Approach to Extracting Knowledge from Legacy Documents".

Cunningham, D.H., D.D. Maynard, D.K. Bontcheva, and M.V. Tablan. 2002. "GATE: A framework and graphical development environment for robust NLP tools and applications". *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*.

Cunningham, Hamish, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, Adam Funk, Angus Roberts, Danica Damljanovic, Thomas Heitz, Mark A. Greenwood, Horacio Saggion Peters, Johann Petrak, and Yaoyong Wim Li. 2011. "Text Processing with GATE (Version 6)".

da Silva, J., Gaël Dias, Sylvie Guilloré, and J. Pereira Lopes. 1999a. "Using localmaxs algorithm for the extraction of contiguous and non-contiguous multiword lexical units". *Progress in Artificial Intelligence*, p. 849–849.

da Silva, J.F. and G.P. Lopes. 1999. "A local maxima method and a fair dispersion normalization for extracting multi-word units from corpora". In *Sixth Meeting on Mathematics of Language*. p. 369–381.

da Silva, J.F., JGP Lopes, MF Xavier, and G Vicente. 1999b. "Relevant Expressions in Large Corpora". In *Actes de l'atelier "Corpus et Traitement Automatique des Langues: Pour*

une réflexion méthodologique" (TALN'99), Institut d'Etudes Scientifiques, Cargèse, Corse (France), July. p. 12–17.

Daille, Béatrice. 1996. "Study and implementation of combined techniques for automatic extraction of terminology". *The Balancing Act: Combining Symbolic and*.

Daille, Béatrice, Éric Gaussier, and Jean-Marc Langé. 1994. "Towards automatic extraction of monolingual and bilingual terminology". In *Proceedings of the 15th conference on Computational linguistics* -. (Morristown, NJ, USA 1994), p. 515. Association for Computational Linguistics.

Danielsson, P. 2007. "What Constitutes a Unit of Analysis in Language?". *Linguistik online*, vol. 31, n° 2, p. 2007.

Dannélls, Dana. 2006. "Automatic acronym recognition". *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Posters & Demonstrations on - EACL '06*, p. 167.

de Cruys, T Van. 2007. "Semantics-based multiword expression extraction". *Perspective on Multiword Expressions*, , p. 25–32.

de Marneffe, M.C., S. Padó, and C.D. Manning. 2009. "Multi-word expressions in textual inference: Much ado about nothing?". In *Proceedings of the 2009 Workshop on Applied Textual Inference*. p. 1–9. Association for Computational Linguistics.

Deeptimahanti, D.K. and R. Sanyal. 2011. "Semi-automatic generation of UML models from natural language requirements". In *Proceedings of the 4th India Software Engineering Conference*. p. 165–174. ACM.

Dias, Gaël. 2003. "Multiword unit hybrid extraction". In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment-Volume 18*. p. 41–48. Association for Computational Linguistics.

Dias, Gaël, Sylvie Guilloré, J.C. Bassano, José Gabriel, and J.G.P. Lopes. 2000. "Combining Linguistics with Statistics for Multiword Term Extraction: A Fruitful Association?".

Doucet, Antoine. 2006. "Fast extraction of discontiguous sequences in text: a new approach based on maximal frequent sequences". *Proceedings of IS-LTC*, vol. 68.

Duan, J., M. Zhang, L. Tong, and F. Guo. 2009a. "A Hybrid Approach to Improve Bilin-gual Multiword Expression Extraction". *Advances in Knowledge Discovery and Data Mining*, p. 541–547.

Duan, Jianyong, Ru Li, and Yi Hu. apr 2009b. "A bio-inspired application of natural language processing: A case study in extracting multiword expression". *Expert Systems with Applications*, vol. 36, n° 3, p. 4876–4883.

Dunning, Ted. 1993. "Accurate methods for the statistics of surprise and coincidence". *Computational linguistics*, vol. 19, n° 1, p. 61–74.

Enguehard, Chantal and Laurent Pantera. 1994. "Automatic natural acquisition of a terminology". *Journal of quantitative linguistics*, p. 27–32.

Evans, D.A. and Chengxiang Zhai. 1996. "Noun-phrase analysis in unrestricted text for information retrieval". In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*. p. 17–24. Association for Computational Linguistics.

Fazly, Afsaneh. 2007. "Distinguishing subtypes of multiword expressions using linguistically-motivated statistical measures". *A Broader Perspective on Multiword Expressions*, , p. 9–16.

Fellbaum, C. 1998. "WordNet: an electronic lexical database (language, speech, and communication)". *The MIT Press*.

Feng, Fangfang and W. Bruce Croft. 2001. "Probabilistic techniques for phrase extraction". *Information processing & management*, vol. 37, n° 2, p. 199–220.

Fothergill, Richard and Timothy Baldwin. 2011. "Fleshing it out: A Supervised Approach to MWE-token and MWE-type Classification". *aclweb.org*, p. 911–919.

Fox Nik, Jared; Brown. 2007. "Automatically Extracting Acronyms from Biomedical Text".

Frantzi, Katerina, Sophia Ananiadou, and Hideki Mima. 2000. "Automatic recognition of multi-word terms: the C-value/NC-value method". *International Journal on Digital Libraries*, vol. 3, n° 2, p. 115–130.

Frantzi, Katerina T. and Sophia Ananiadou. 1996. "Extracting nested collocations". *Proceedings of the 16th conference on Computational linguistics* -, p. 41.

Freund, Yoav and Llew Mason. 1999. "The alternating decision tree learning algorithm". In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE*. (Bled, Slovenia 1999), p. 124–133. Citeseer.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2000. "Additive logistic regression : A statistical view of boosting". *Annals of statistics*, vol. 28, n° 2, p. 337–407.

Gaines, Brian R and Paul Compton. 1995. "Induction of Ripple-Down Rules Applied to Modeling Large Databases". *Journal of Intelligent Information Systems*, vol. 5, n° 3, p. 211–228.

Gerdemann, Dale and Gaston Burek. 2010. "Challenges for Discontiguous Phrase Extraction". In *Proceedings of the first Workshop on Supporting eLearning with Language Resources and Semantic Data*.

Godby, C. Jean. dec 2001. "Two Techniques for the Identification of Phrases in Full Text". *Journal of Library Administration*, vol. 34, n° 1-2, p. 57–65.

Goldman, JP, L Nerima, and E Wehrli. 2001. "Collocation extraction using a syntactic parser". *Proceedings of the ACL Workshop on Collocations*.

Grali, Filip, Monika Czerepowicka, and Filip Makowiecki. 2010. "Computational Lexicography of Multi-Word Units : How Efficient Can It Be ?". In *Proceedings of the Workshop on Multiword Expressions: from Theory to Applications*. p. 1–9.

Gralinski, Filip, Agata Savary, Monika Czerepowicka, and Filip Makowiecki. 2010. "Computational Lexicography of Multi-Word Units: How Efficient Can It Be?". *Proceedings of the Multiword Expressions: From Theory to Applications*, , p. 2–10.

Green, Spence, M.C. de Marneffe, John Bauer, and C.D. Manning. 2010. "Multiword Expression Identification with Tree Substitution Grammars: A Parsing tour de force with French". *aclweb.org*.

Grimes, Seth. 2008. "Unstructured Data and the 80 Percent Rule". <<http://breakthroughanalysis.com/2008/08/01/unstructured-data-and-the-80-percent-rule/>>.

Grishman, R. 1996. "Design of the MUC-6 evaluation". *Proceedings of a workshop on held at*, p. 1–11.

Gross, Maurice. 1986. "Lexicon-grammar: the representation of compound words". In *Proceedings of the 11th conference on Computational linguistics*. p. 1–6. Association for Computational Linguistics.

Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. "The WEKA Data Mining Software: An Update". *SIGKDD Explorations*, vol. 11, n° 1.

Hartmann, Sven and Sebastian Link. 2007. "English sentence structures and EER modeling". In *Proceedings of the fourth Asia-Pacific conference on Comceptual modelling-Volume 67*. p. 27–35. Australian Computer Society, Inc.

Hashimoto, Chikara and Daisuke Kawahara. oct 2009. "Compilation of an idiom example database for supervised idiom identification". *Language Resources and Evaluation*, vol. 43, n° 4, p. 355–384.

Hearst, M A. 1992. "Automatic acquisition of hyponyms from large text corpora". *Proceedings of the 14th conference on Computational linguistics-Volume 2*, p. 539–545.

Hoffman, Robert R., Paul J. Feltovich, Kenneth M. Ford, David D. Woods, Gary Klein, and Anne Feltovich. 2002. "A Rose by Any Other Name... Would Probably Be Given an Acronym". *IEEE INTELLIGENT SYSTEMS*.

Ittoo, Ashwin, Laura Maruster, Hans Wortmann, and Gosse Bouma. 2010. "Textractor : A Framework for Extracting Relevant Datasets". *Language*, p. 71–82.

Jackendoff, Ray. 1997. "The architecture of the language faculty". *Computational Linguistic*, p. 652–655.

Jain, Alpa, Silviu Cucerzan, and Saliha Azzam. aug 2007. "Acronym-Expansion Recognition and Ranking on the Web". *2007 IEEE International Conference on Information Reuse and Integration*, p. 209–214.

Justeson, JS and SM Katz. 1995. "Technical terminology: some linguistic properties and an algorithm for identification in text.". *Natural language engineering*, p. 9–27.

Kabak, Y and A Dogac. 2010. "A Survey and Analysis of Electronic Business Document Standards". *Acm Computing Surveys*, vol. 42, n° 3, p. –.

Katz, Graham. 2006. "Automatic identification of non-compositional multi-word expressions using latent semantic analysis". *of the Workshop on Multiword Expressions*:, , p. 12.

Kempe, André. 2006. "Acronym-Meaning Extraction from Corpora Using Multi-Tape Weighted Finite-State Machines". *Arxiv preprint cs/0612033*.

Knuth, Donald, James H Morris, and Vaughan Pratt. 1977. "Fast pattern matching in strings". *SIAM Journal on Computing*, vol. 6, n° 2, p. 323–350.

Kof, Leonid. 2010. "Requirements analysis: concept extraction and translation of textual specifications to executable models". *Natural Language Processing and Information Systems*.

Kotonya, Gerald and Ian Sommerville, 1998. *Requirements Engineering : Processes and Techniques*.

Krenn, Brigitte, Pavel Pecina, and Frank Richter. 2008. "Towards a Shared Task for Multiword Expressions". In *LREC Workshop*.

Kruchten, P. 1999. "The Rational Unified Process".

Laporte, E., Takuya Nakamura, and Stavroula Voyatzsi. 2008. "A French Corpus Annotated for Multiword Nouns". *Towards a Shared Task for Multiword Expressions (MWE 2008)*, p. 27.

Larkey, L.S., Paul Ogilvie, M.A. Price, and Brenden Tamilio. 2000. "Acrophile: An Automated Acronym Extractor and Server". In *Proceedings of the fifth ACM conference on Digital libraries*. (Dallas TX 2000), p. 205–214. ACM.

Leffingwell, Dean and Don Widrig, 2000. *Managing Software Requirements: A Unified Approach*.

Leroy, G, H Chen, and J Martinez. 2003. "A shallow parser based on closed-class words to capture relations in biomedical text". *Journal of Biomedical Informatics*, vol. 36, n° 3, p. 145–158.

Li, Quanzhi and Yi-Fang Brook Wu. dec 2006. "Identifying important concepts from medical documents.". *Journal of biomedical informatics*, vol. 39, n° 6, p. 668–79.

Li, Ru, Lijun Zhong, and Jianyong Duan. 2008. "Multiword Expression Recognition Using Multiple Sequence Alignment". *2008 International Conference on Advanced Language Processing and Web Information Technology*, p. 133–138.

Lillehagen, Frank and John Krogstie, 2008. *Active Knowledge Modeling of Enterprises*.

Lin, Dekang, Kenneth Church, Heng Ji, Satoshi Sekine, David Yarowsky, Shane Bergsma, Kailash Patil, Emily Pitler, Rachel Lathbury, Vikram Rao, and Others. 2010. "New tools for web-scale N-grams". Citeseer.

Loucopoulos, P and R Champion. jun 1988. "Knowledge-based approach to requirements engineering using method and domain knowledge". *Knowledge-Based Systems*, vol. 1, n° 3, p. 179–187.

Loucopoulos, P. and R.E.M. Champion. 1990. "Concept acquisition and analysis for requirements specification". *Software Engineering Journal*, vol. 5, n° 2, p. 116–124.

Mala, G. and G. Uma. 2006. "Elicitation of Non-functional Requirement Preference for Actors of Usecase from Domain Model". *Advances in Knowledge Acquisition and Management*, p. 238–243.

Manning, Christopher D. and Hinrich Schütze, 1999. *Foundations of Statistical Natural Language Processing*.

Martens, Scott and Vincent Vandeghinste. 2010. "An Efficient , Generic Approach to Extracting Multi-Word Expressions from Dependency Trees". In *Proceedings of the Workshop on Multiword Expressions: from Theory to Applications*. p. 84–87.

Maynard, Diana, Horacio Saggion, and Milena Yankova. 2007. "Natural language technology for information integration in business intelligence". *Business Information*.

Meadow, Charles T., Bert R. Boyce, Donald H. Kraft, and Carol L. Barry, 2007. *Text Information Retrieval Systems, Third Edition*. 277–286 p.

Ménard, Pierre André and Sylvie Ratté. sep 2010. "Classifier-based acronym extraction for business documents". *Knowledge and Information Systems*.

Merkel, Magnus and M. Andersson. 2000. "Knowledge-lite extraction of multi-word units with language filters and entropy thresholds". In *Proceedings of 2000 Conference User-Oriented Content-Based Text and Image Handling (RIA'00)*. p. 737–746. Citeseer.

Meziane, Farid and Nikos Athanasakis. 2008. "Generating Natural Language specifications from UML class diagrams". *Requirements Engineering*, p. 1–31.

Momtazi, Saeedeh, S. Khudanpur, and Dietrich Klakow. 2010. "A comparative study of word co-occurrence for term clustering in language model-based sentence retrieval". In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. p. 325–328. Association for Computational Linguistics.

Mota, Cristina, Paula Carvalho, and Elisabete Ranchhod. 2004. "Multiword lexical acquisition and dictionary formalization". In *Proceedings of the Workshop on Enhancing and Using Electronic Dictionaries - ElectricDict '04*. (Morristown, NJ, USA 2004), p. 73. Association for Computational Linguistics.

Nadeau, David and P.D. Turney. 2010. "A supervised learning approach to acronym identification". In *The Eighteenth Canadian Conference on Artificial Intelligence (AI'2005)*. p. 319–329. Springer.

Nenkova, Ani and Rebecca Passonneau. 2004. "Evaluating content selection in summarization: The pyramid method". *Proceedings of HLT-NAACL*.

Ni, Weijian and Yalou Huang. 2008. "Extracting and organizing acronyms based on ranking". In *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*. (Chongqing, China 2008), p. 4542–4547. IEEE.

Nicholson, Jeremy and Timothy Baldwin. 2005. "Statistical Interpretation of Compound Nominalisations". *Technology*, , p. 152–159.

O'Donnell, A.M., D.F. Dansereau, and R.H. Hall. 2002. "Knowledge maps as scaffolds for cognitive processing". *Educational Psychology Review*, vol. 14, n° 1, p. 71–86.

Okazaki, N. and S. Ananiadou. 2006a. "Building an abbreviation dictionary using a term recognition approach". *Bioinformatics*, vol. 22, n° 24, p. 3089.

Okazaki, Naoaki and Sophia Ananiadou. 2006b. "A term recognition approach to acronym recognition". In *Proceedings of the COLING/ACL on Main conference poster sessions*. p. 643–650. Association for Computational Linguistics.

Oliveira, Ana, F.C. Pereira, and A. Cardoso. 2001. "Automatic reading and learning from text". In *Future Trends in Artificial Intelligence, Proceedings of ISAI*. p. 1–12. Citeseer.

Osada, Akira, Daigo Ozawa, and Haruhiko Kaiya. 2007. "The role of domain knowledge representation in requirements elicitation". *Proceedings of the 25th conference on IASTED International Multi-Conference: Software Engineering*.

Pal, Santanu, Tanmoy Chakraborty, and Sivaji Bandyopadhyay. 2010a. "Handling Multiword Expressions in Phrase-Based Statistical Machine Translation". *mt-archive.info*, , p. 215–224.

Pal, Santanu, Sudip Kumar Naskar, Pavel Pecina, Sivaji Bandyopadhyay, and Andy Way. 2010b. "Handling Named Entities and Compound Verbs in Phrase-Based Statistical Machine Translation". In *Proceedings of the Workshop on Multiword Expressions: from Theory to Applications*. p. 45–53.

Pandey, Gaurav and R Daga. 2007. "On Extracting Structured Knowledge from Unstructured Business Documents". *Citeseer*, p. 155–162.

Park, Jaehui and Sang-goo Lee. 2010. "Keyword search in relational databases". *Knowledge and Information Systems*, p. 19.

Park, Y. and R.J. Byrd. 2001. "Hybrid text mining for finding abbreviations and their definitions". In *Proceedings of the 2001 conference on empirical methods in natural language processing*. p. 126–133. Citeseer.

Pfleeger, S. L. and J. M. Atlee, 2009. *Software engineering - theory and practice*. ed. 4th.

Piao, Scott S. L., Paul Rayson, Dawn Archer, Andrew Wilson, and Tony McEnery. 2003. "Extracting multiword expressions with a semantic tagger". *Proceedings of the ACL 2003 workshop on Multiword expressions analysis, acquisition and treatment* -, p. 49–56.

Piao, Scott Songlin, Paul Rayson, Dawn Archer, and Tony McEnery. oct 2005. "Comparing and combining a semantic tagger and a statistical tool for MWE extraction". *Computer Speech & Language*, vol. 19, n° 4, p. 378–397.

Popescu, Daniel, S. Rugaber, N. Medvidovic, and D. Berry. 2008. "Reducing Ambiguities in Requirements Specifications Via Automatically Created Object-Oriented Models". *Lecture Notes in Computer Science*, vol. 1, p. 103–124.

Powers, David. 2012. "The Problem with Kappa". *EACL 2012*, p. 345–355.

Pressman, Roger, 2001. *Software engineering: a practitioner's approach*.

Prieto-Diaz, R. 1991. "Domain Analysis concepts and research directions". *Domain Analysis and Software Systems Modeling*,, p. 9–32.

Pustejovsky, James, José Castano, B. Cochran, Maciej Kotecki, and Michael Morrell. 2001. "Automatic extraction of acronym-meaning pairs from MEDLINE databases". *Studies in health technology and informatics*, p. 371–375.

Quinlan, Ross. 1994. "C4.5: Programs for Machine Learning". *Machine Learning*, vol. 16, n° 3, p. 235–240.

Rafeeqe, P C and K A Abdul Nazeer. 2007. "Text Mining for Finding Acronym-Definition Pairs from Biomedical Text Using Pattern Matching Method with Space Reduction Heuristics". *Advanced Computing and Communications, International Conference on*, p. 295–300.

Ramisch, Carlos, Aline Villavicencio, and Christian Boitet. 2010. "Multiword expressions in the wild?: the mwetoolkit comes in handy". In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*. p. 57–60. Association for Computational Linguistics.

Ranchhod, E. 2005. "Using Corpora to Increase Portuguese MWE Dictionaries. Tagging MWE in a Portuguese Corpus". *Proceedings from The Corpus Linguistics*.

Rose, Stuart, Dave Engel, and Nick Cramer. 2010. Automatic keyword extraction from individual documents. Wiley, editor, *Text Mining*.

RÚA, P.L. 2004. "Acronyms & Co.: A typology of typologies". *Estudios Ingleses de la Universidad Complutense*, vol. 12, p. 109–129.

Rumpler, RAB and JM Pinon. 2003. "An Analysis of Tools for an Automatic Extraction of Concept in Documents for a Better Knowledge Management". *Information Technology and Organizations*, p. 201–205.

Sag, I, Timothy Baldwin, Francis Bond, and Ann Copestake. 2002. "Multiword expressions: A pain in the neck for NLP". *Linguistics and Intelligent*.

Savary, Agata. 2000. "Recensement et description des mots composés: méthodes et applications". PhD thesis, Université de Marne-la-Vallée.

Savary, Agata. 2009. "Multiflex: A Multilingual Finite-State Tool for Multi-Word Units". *Implementation and Application of Automata*, p. 237–240.

Sawyer, Pete, Paul Rayson, and Ken Cosh. nov 2005. "Shallow knowledge as an aid to deep understanding in early phase requirements engineering". *IEEE Transactions on Software Engineering*, vol. 31, n° 11, p. 969–981.

Schamber, Linda. 1994. "Relevance and Information Behavior". *Annual review of information science and technology (ARIST)*, vol. 29, p. 3–48.

Schmid, Helmut. 1994. "Probabilistic part-of-speech tagging using decision trees". *International Conference on new methods in language processing*, p. 44–49.

Schone, P and D Jurafsky. 2001. "Is knowledge-free induction of multiword unit dictionary headwords a solved problem". *Proc. of the 6th Conference on Empirical Methods* . . .

Schumann, Eduardo Torres and Klaus U. Schulz. may 2005. "Stable methods for recognizing acronym-expansion pairs: from rule sets to hidden Markov models". *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 8, n° 1, p. 1–14.

Schwartz, A.S. and M.A. Hearst. 2003. "A simple algorithm for identifying abbreviation definitions in biomedical text". In *Pacific Symposium on Biocomputing*. p. 451–462. Citeseer.

Shen, Hong. 2005. "Voting between multiple data representations for text chunking". *Advances in Artificial Intelligence*.

Sheremetyeva, Svetlana. 2009. "On extracting multiword NP terminology for MT". *c 2009 European Association for Machine Translation*, , p. 205.

Silva, JF, Zornitsa Kozareva, Veska Noncheva, and GP Lopes. 2004. "Extracting named entities. a statistical approach". In *Proceedings of the XIme Confrence sur le Traitement des Langues Naturelles—TALN, 19–22 Avril, Fez, Marroco*. p. 347–351.

Silva, Joaquim and G. Lopes. 2010. "Towards automatic building of document keywords". In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. p. 1149–1157. Association for Computational Linguistics.

Sohn, Sunghwan, Donald C Comeau, Won Kim, and W John Wilbur. 2008. "Abbreviation definition identification based on automatic precision estimates". *Bmc Bioinformatics*, vol. 9, n° 1, p. 402.

Song, M. and I. Yoo. 2007a. "A Hybrid Abbreviation Extraction Technique for Biomedical Literature". In *Bioinformatics and Biomedicine, 2007. BIBM 2007. IEEE International Conference on*. p. 42–47. IEEE.

Song, M. and I. Yoo. 2007b. "A Hybrid Abbreviation Extraction Technique for Biomedical Literature". In *Bioinformatics and Biomedicine, 2007. BIBM 2007. IEEE International Conference on*. p. 42–47. IEEE.

Stock, Oliviero and Carlo Strapparava. 2005. "Hahacronym: A computational humor system". In *Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*. p. 113–116. Association for Computational Linguistics.

Subhashini, R and V.J.S. Kumar. 2010. "Shallow NLP techniques for noun phrase extraction". In *Trendz in Information Sciences & Computing (TISC), 2010*. p. 73–77. IEEE.

Subirats, Carlos and Hiroaki Sato. 2004. "Spanish framenet and framesql". In *4th International Conference on Language Resources and Evaluation. Workshop on Building Lexical Resources from Semantically Annotated Corpora. Lisbon (Portugal)*. Citeseer.

Taghva, Kazem and Je Gilbreth. 1999. "Recognizing acronyms and their definitions". *International Journal on Document Analysis and Recognition*, p. 191–198.

Tanguy, Ludovic and Nikola Tulechki. 2009. "Sentence Complexity in French: a Corpus-Based Approach". *Language*, vol. 15, p. 06.

Torii, Manabu, Hongfang Liu, Zhangzhi Hu, and Cathy Wu. 2006. "A comparison study of biomedical short form definition detection algorithms". In *Proceedings of the 1st international workshop on Text mining in bioinformatics*. p. 52–59. ACM.

Tseng, Y.H. 1998. "Multilingual keyword extraction for term suggestion". In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. p. 377–378. ACM.

Venkatsubramanyan, Shailaja. 2004. "Multiword expression filtering for building knowledge maps". *on Multiword Expressions*: , , p. 40–47.

Vincze, Veronika, T. István Nagy, and G. Berend. 2011. "Multiword expressions and named entities in the Wiki50 corpus". In *Proceedings of RANLP*. p. 289–295.

Vintar. 2004. "Comparative Evaluation of C-value in the Treatment of Nested Terms". *Workshop Description*.

Vivaldi, Jorge and Horacio Rodríguez. 2007. "Evaluation of terms and term extraction systems A practical approach". *Terminology*, vol. 13, n° 2, p. 225–248.

Vossen, Piek, 1998. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*.

Wang, Zheng, Qing Wang, and Ding-Wei Wang. 2008. "Bayesian network based business information retrieval model". *Knowledge and Information Systems*, vol. 20, n° 1, p. 63–79.

Webb, Geoffrey I. 1999. "Decision Tree Grafting From the All-Tests-But-One Partition". In *IJCAI '99 Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*. p. 702–707. Morgan Kaufmann Publishers Inc.

Wehrli, Eric, Violeta Seretan, and Luka Nerima. 2010. "Sentence Analysis and Collocation Identification". In *Proceedings of the Workshop on Multiword Expressions: from Theory to Applications*. p. 27–35.

Witte, Rene, Qiangqiang Li, Yonggang Zhang, and Juergen Rilling. ".

Witten, I.H. and E Frank, 2005. *Data Mining - Practical Machine Learning Tools and Techniques*.

Woon, Wei Lee and Kuok-Shoong Daniel Wong. 2009. "String alignment for automated document versioning". *Knowledge and Information Systems*, vol. 18, n° 3, p. 293–309.

Xu, Jian, Jingsong Yu, and Huilin Wang. dec 2010. "Automatic Extraction of Multiword Expressions Combining Statistical and Similarity Approaches". *2010 Fourth International Conference on Genetic and Evolutionary Computing*, p. 256–259.

Xu, Jun and Yalou Huang. apr 2006. "Using SVM to Extract Acronyms from Text". *Soft Computing*, vol. 11, n° 4, p. 369–373.

Xu, Yun, Zhihao Wang, Yiming Lei, Yuzhong Zhao, and Yu Xue. 2009. "MBA: a literature mining system for extracting biomedical abbreviations ". *Bmc Bioinformatics*, vol. 10, n° 1, p. 14.

Yeates, S., D. Bainbridge, and I.H. Witten. 2000. "Using compression to identify acronyms in text". In *Proceedings DCC 2000. Data Compression Conference*. p. 582. IEEE Comput. Soc.

Yeates, Stuart. 1999. "Automatic extraction of acronyms from text". In *New Zealand Computer Science Research Students' Conference*. p. 117–124. Citeseer.

Yi, Jeonghee. 1999. "Mining the web for acronyms using the duality of patterns and relations". In *Proceedings of the 2nd international workshop on Web information and data management*. p. 48–52.

Yoshida, Minoru and Hiroshi Nakagawa. 2005. "Automatic term extraction based on perplexity of compound words". p. 269–279.

Yu, Hong, Won Kim, Vasileios Hatzivassiloglou, and John Wilbur. 2006. "A large scale, corpus-based approach for automatically disambiguating biomedical abbreviations". *ACM Transactions on Information Systems (TOIS)*, vol. 24, n° 3, p. 380–404.

Yu H Friedman C., Hripcsak G. 2002. "Mapping abbreviations to full forms in biomedical articles". *American Medical Informatics Association*, , p. 262–272.

Zahariev, Manuel. 2003. "Efficient Acronym-Expansion Matching for Automatic Acronym Acquisition.". *INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE ENGINEERING*, p. 32–37.

Zahariev, Manuel. 2004a. "A Linguistic Approach to Extracting Acronym Expansions from Text". *Knowledge and Information Systems*, vol. 6, n° 3, p. 366–373.

Zahariev, Manuel. 2004b. "Automatic sense disambiguation for acronyms". *Proceedings of the 27th annual international conference on Research and development in information retrieval - SIGIR '04*, p. 586.

Zhang, Wen, Taketoshi Yoshida, and Xijin Tang. 2009a. "Distribution of Multi-Words in Chinese and English Documents". *International Journal of Information Technology & Decision Making (IJITDM)*, vol. 08, n° 02, p. 249–265.

Zhang, Wen, Taketoshi Yoshida, Xijin Tang, and Tu-Bao Ho. 2009b. "Improving effectiveness of mutual information for substantival multiword expression extraction". *Expert System with Application*, vol. 36, n° 8, p. 10919–10930.

Zielinski, Daniel and Y.R. Safar. 2005. "Research meets practice: t-survey 2005: An on-line survey on terminology extraction and terminology management". *Proceedings of Translating and the Computer (ASLIB 27)*.