Grenoble INP – Institut National Polytechnique de Grenoble
École Nationale Supérieure d'Informatique, Mathématiques Appliqueées et Télécommunications
LIG – Laboratoires d'Informatique de Grenoble
Groupe d'Étude en Traduction Automatique/Traitement Automatisé des Langues et de la Parole

CARLOS EDUARDO RAMISCH

Multi-word terminology extraction for domain-specific documents

Mémoire de Master
Master 2 de Recherche en Infromatique
Option Intelligence Artificielle et Web

Christian BOITET
Advisor

Aline VILLAVICENCIO
Coadvisor

Grenoble, juin 2009

ACKNOWLEDGEMENTS

First of all, for their patience and support, I would like to thank my advisors Aline Villavicencio and Christian Boitet, that were always ready to answer my questions, discuss interesting points of my research and motivate me. Without them I would never be able to arrive this far. Thank you for sharing with me your knowledge and your experience on this passionating and challenging science that is Natural Language Processing.

In my academic trajectory, I counted on the teaching team of two high-quality superior institutions: the Institute of Informatics of UFRGS[1] and ENSIMAG/Télécom of Grenoble INP[2]. Additionally, I would like to thank the CLT team, specially Marc Dymetman and Lucia Specia, for their advice and guidance during my internship, and moreover for the nice time spent with all XRCE colleagues.

Third, I would like to thank all the course and internship colleagues that, at one point or another, shared with me the joys and sacrifices of the academic life. The journey would not be so funny without my colleagues from PLN-UFRGS, GETALP, XRCE, and specially my class mates and friends at UFRGS and ENSIMAG.

Fourth, I would like to thank the institutions and people that provided me with funding for my studies: CNPq, for the scientific initiation scholarship (PIBIC), CAPES for the international exchange scholarship (Brafitec), Floralis for the internship at GETALP and XRCE, for the research internship. Moreover, I would like to thank my parents for their financial support without which I would not be able to accomplish this work.

Fifth, thanks to my friends for the funny moments, for their support and reliability. Thanks to my companion Arnaud for his patience when I was too busy, for his help when I was in need and for his unconditional support. Thanks to my parents Ari and Sônia, to my grand-parents Gisela and Edgar, and to my sister Renata that, since the very start, provided me with the means for studying and having a successful academic career. Finally, to all the people that never stopped believing in the quality of my work and in my capacity, thank you!

A researcher and computer scientist does not work, as in the caricatures, individually, behind his computer in an empty and cold lab. On the contrary, his work is the result of a collective effort. Therefore, I thank all the people that directly or indirectly contributed to the achievement of this work.

[1] Instituto de Informática, Universidade Federal do Rio Grande do Sul

[2] École National Supérieure d'Informatique, Mathématiques Appliquées et Télécommunications, Grenoble Institut National Polytéchnique

CONTENTS

ABSTRACT

One of the greatest challenges of Natural Language Processing is to adapt techniques that work well on general-purpose language to handle domain-specific texts. It is specially important, at the lexical level, to be able to deal with terminology and its implications. The goal of this work is to employ multi-word expression identification techniques on the biomedical Genia corpus in order to evaluate to what extent they can be used for Multi-Word Terms (MWT) extraction in both mono- and multi-lingual systems. We start with a short motivation, relating our research to cross-lingual information access and machine translation and then setting the theoretical bases and definitions of the concepts on which we work: multi-word expression, term and multi-word term. Then, we present a non-exhaustive discussion of some approaches used by different related work for multi-word expressions identification, interpretation, classification and application. Some work on domain-specific terminology is also shown, particularly concerning symbolic methods for automatic ontology discovery and semi-automatic MWT extraction based on statistic and morphological cues. A very brief introduction to statistical machine translation shows some aspects that we investigate in our evaluation. Our methodology can be divided in four steps: pre-processing, in which we homogenise the corpus tokenisation and morphology, candidate extraction through selected morphological patterns, candidate filtering through association measures from the corpus and from the Web and some machine learning, and application-oriented evaluation on a machine translation system. Our results show that good pre-processing heuristics can drastically improve performance. Additionally, part of speech tag pattern selection is domain-dependent and fundamental for higher recall. Among the association measures the *t* test seems to perform better. The corpora we used, Genia and the Web, are essentially different and provide good MWT estimators at different rank intervals, suggesting that they should be combined using more sophisticated tools. While neural networks cannot correctly learn a MWT identification model from our features, support vector machines seem to provide good tradeoff between recall and precision, being even more effective when some frequency threshold is applied. Our approach presents recall around 6.4% with 74% precision. Even though recall seems low, it is considerably higher than a standard MWE extraction technique called Xtract, which obtains 67% precision at 3.8% recall on the Genia corpus. Finally, we point out some future directions for MWT extraction, including better marginal probability estimators, contingency table-based methods for arbitrary MWT lengths, multi-class evaluation and collaborative systems to build multi-lingual specialised lexicons.

An extended abstract in Portuguese is available on appendix B.

An extended abstract in French is available on appendix A.

Keywords: Natural Language Processing, Multiword Expressions, Terminology, Lexical Acquisition, Machine Learning, Statistical Machine Translation, Domain Adaptation.

1 INTRODUCTION

The ultimate goal of Natural Language Processing (NLP) is to make the communication process between humans and computers as natural as asking a friend for a favour. Natural language is perhaps the most natural way for humans to express themselves and corresponds to one of the main components of intelligent behaviour. It is therefore often seen as a sub-area of Artificial Intelligence and inherits from it several theories and techniques. The study of Natural Language Processing has several applications that can help to overcome linguistic barriers not only between humans and computers but also between humans and humans. Examples of the former include question answering, information extraction and natural language generation, while the latter includes applications such as machine translation and cross-lingual information access. The term Natural Language Processing is also sometimes used as a synonym for Computational Linguistics; it is an interdisciplinary area situated at the confluence of Computer Science and Applied Linguistics.

Any intelligent system that would like to interact with human beings should be able to understand and generate natural language. However, language is intrinsically complex and represents a real challenge for Artificial Intelligence. On one hand, human languages are huge in many dimensions: lexicon, syntax rules, morphological rules, semantics, etc. Additionally, linguistic phenomena tend to be arbitrary, involving cultural and historical factors that cannot be taken into account by computational models. On the other hand, language is constantly changing: every day new words are created, old ones are abandoned, new meanings are assigned to old words, etc. Computer systems cannot simply refuse an input as "wrong" because it uses a new word that is not in the lexicon, but they should be flexible and learn fast enough to deal with language evolution. Since language is culturally motivated, the correspondences between two or more human languages are far from being one-to-one relations, and the number of living languages in the world is an obstacle in human-human and human-computer interaction. On top of all that, language is not an isolated phenomenon and its interpretation involves general knowledge of the world, as we can see in the classical example sentence *time flies like an arrow*, which illustrates how much ambiguity natural language contains. In this example, *flies* can be the main verb of a comparison of time with an arrow, or *like* can be the main verb, stating that a specific type of fly is fond of arrows.

Technical and scientific knowledge is transmitted through language, and presents some linguistic properties that differ from general-purpose language. Computer systems should be able to deal with specialised language in all levels, from lexicon and morphology to semantics and style. Domain-specific texts represent a large knowledge source expressed in natural language and are therefore of great interest for NLP. Following a descriptive/cognitive approach that directly contrasts with generative language processing, we believe that computer systems should learn such kind of information from text, following the so-called data-driven NLP methods or corpus linguistics.

1.1 Motivation

Imagine a user with a domain-specific information need, e.g. the gathering of scientific papers about recent advances in example-based machine translation. If the user plans to use a general-purpose Web search engine, this is probably going to be a long and complicated task. On one hand, the results retrieved by a search engine like Google often include a great number of irrelevant user-generated content. To solve this problem, commercial search systems quickly realised that they should create specific indices containing, for instance, selected scientific papers (e.g. Google Scholar[1] or Citeseer[2]). On the other hand, the generic Information Retrieval (IR) techniques used by these tools need to take into account specific domain information, such as terminology. For instance, a Web multi-word keyword search for *segmentation techniques* should return very different sets of documents whether the domain is Natural Language Processing[3] or Image Processing. The same considerations are also true for simple terms, e.g. *leaf* and *tree* are very common terms in Botany and in Informatics, but are related to disjoint concepts in each domain. On top of that, multi-lingual access to information is not possible: most of the technical and scientific literature is only accessible in one language (mostly in English, for scientific documents) and the translation of domain-specific terminology by the user, a tedious task involving the inquiry of several external sources, is generally done manually before entering the keywords of the query into the search engine text box. In short, nowadays a user needs to translate and add context keywords to his/her query before having a substantial amount of relevant documents, and then statically translate the relevant retrieved documents and extract the desired information.

However, what our hypothetical user would have liked would be to have access, through a domain-specific cross-language Information Retrieval (IR) engine, to relevant documents written in or translated to his own language, with precise automatic translation for the terminological multi-word and simplex lexical units of both, query and documents. This example illustrates a common problem that one faces in everyday life when one tries to access the huge amount of information that the World Wide Web made available. In this context, multi-lingual access is obviously a topic of great interest for research and commercial institutions: instead of providing information services to English speakers only, companies would like to reach, for example, the Brazilian public, which could be potentially a very profitable market. Domain-specific multi-lingual access is a particular instance of this problem in which users are experts in a domain and would like to find relevant information (e.g. on the Web) modulo his/her specific knowledge area regardless of the language in which they were originally written.

Situations like these have made Cross-Lingual Information Retrieval (CLIR) a very active research area in the last years (Peters et al. 2008).

Although the motivation of this work is strongly related to CLIR, our goal here is not in itself the creation of a multi-lingual IR environment. Nonetheless, we are broadly interested in the relationship between domain-specific terminology and its applications, specially in what concerns the potential benefits of automatic terminology extraction for Machine Translation (MT). The wider goal of this project is to create a collaborative system integrating these two concepts, terminology and multi-lingualism, to provide truly multi-lingual information access. Collaboration in multi-lingual environments has been the focus of recent research and provides good compromise between automaticity and quality (Huynh et al. 2008). To achieve this goal, in this work, we concentrate on the relationship between terminology and multi-lingualism regardless of specific CLIR techniques or information management and access interfaces.

[1] `http://scholar.google.com`

[2] `http://citeseer.ist.psu.edu/`

[3] Segmentation techniques in NLP are often employed for sentence boundaries detection and tokenisation.

1.2 Goals and hypotheses

In the last years, there has been increasing interest in the investigation of Multi-Word Expressions (MWE). There is no precise and unique definition of MWE: they are actually a general and task-oriented concept created by Computational Linguists to describe a wide range of distinct but related linguistic phenomena such as noun compounds (*traffic light*, *feu rouge*), nominalisations (*rendre visite*, *take a walk*), idiomatic expressions (*cloud number nine*, *sem pé nem cabeça*), institutionalised phrases (*Dear Sir or Madam*, *ad hoc*), determinerless prepositional phrases (*at home*, *à volonté*), light verbs (*to do the dishes*), phrasal verbs (*to take off*, *mitgehen*, *aufstehen*), compound terms (*sistema operacional*, *Datenbanksystem*), etc. In this work, we will adopt the loose definition cited by Sag et al. (2002).

Definition 1.1 (Multi-Word Expression) *A Multi-Word Expression (MWE) is a set of two or more words (i.e. an expression) with non-compositional semantics, i.e. for which the sense of the whole expression cannot be directly nor completely inferred from the sense of its individual words.*

From a textual perspective, the words of a multi-word expression co-occur more often than they would normally do if they were a set of ordinary words co-occurring by chance (Manning and Schütze 1999). Due to their heterogeneous characteristics, MWEs provide a real challenge for current NLP technology.

A related and important definition, this time in the context of domain-specific texts, is that of a *term*. According to Krieger and Finatto (2004), for the expert, terms are the actual representation of knowledge in a specific area, i.e. they are lexical units that stand for abstract concepts of the domain. It is interesting to highlight that the first Terminology researchers believed that there was a 1:1 relationship between terms and concepts, in other words that the semantic relation between them was functional and injective. This assumption has been largely discussed and is nowadays regarded as simplifying and reductive, but it still provides important substrate for understanding what exactly is considered a term in the context of this work.

Definition 1.2 (Term) *A term is a lexical unit that has an unambiguous meaning when used in a text of a specific domain. Terms are the linguistic representation of the concepts of a knowledge area. Terms are often treated as a single lexical unit even though they might be composed of more than one orthographic word. A set of domain-specific terms constitutes a* specialised lexicon *or a* terminology[4] *(Krieger and Finatto 2004).*

Our definition of *term* follows the main principles of the General Theory of Terminology, where terms are lexical representations of abstract concepts (that are language-independent). This theory has been hardly criticised for considering the terms as static lexical units, regardless of their discursive contexts. Although being considered by many terminologists as obsolete, the previous definition is functional enough for our purposes and should fit the processing needs of most current NLP technology. We discuss the advantages and weaknesses of this assumption later in this section.

There are several differences between MWEs and terms, not only epistemologically but also pragmatically. First, terms may be either simplex (single-word) or multi-word units like nominal and verbal locutions, whereas MWEs are inherently composed of two or more words. Second, MWEs include both technical and scientific language and general-purpose everyday language while terms include only the former. Even though the sharp distinction between general and specialised communications has been questioned, the difference is important here because the available computational

[4] We adopted the convention of writing Terminology with a capital *T* when referring to the science, and the homonym terminology with a lowercase *t* to the collection of terms from a specific domain.

methods to deal with MWEs in general-purpose texts are potentially different from methods to handle specialised corpora and terminology. Finally, MWEs find theoretical support in general Linguistics while terms are the objects of study of Terminology, which share some properties but are nevertheless distinct sciences. MWEs and terms have some similar aspects: both have non-conventional semantics, both are a challenge for NLP systems, etc. On the overlap between terms and multi-word expressions lies our object of study. In this work, we are interested in domain-specific terms that are composed by two or more orthographic words, i.e. Multi-Word Terms (MWTs).

Definition 1.3 (Multi-Word Term) *A Multi-Word Term (MWT) is a term that is composed of more than one word. The unambiguous semantics of a multi-word term depends on the knowledge area of the concept it describes and cannot be inferred directly from its parts (SanJuan et al. 2005, Frantzi et al. 2000).*

Note that this definition is essentially different from what terminologists consider to be a *phraseology* like in *to initiate a remote digital loopback test*. From this point of view, phraseologies are much more related to specialised collocations than to our conception of multi-word term. Specialised phraseology deals with more complex constructions that often involve more than one domain-specific concept, and are often seen as intermediary entities between terms and institutionalised sentences (e.g. the action of *initiate* and the term *remote digital loopback test*). We, on the other hand, consider multi-word terms as being a multi-word lexical representation of an abstract term, but sharing with the latter the same properties of monosemy and limited variability, i.e. a multi-word term is, as well as a single-word term, a lexical manifestation of a specialised concept.

Moreover, specialised phraseology allows more flexibility in its syntactic realisations (e.g. nominalisations in *to initiate/the initiation of*) whereas in our definition of MWTs, they are syntactic rigid, even if subject to the same morphological rules of inflection (gender, number) than ordinary lexical units. Although the theoretical bases of our work can be regarded as being part of the General Theory of Terminology, we emphasise that in all steps we are considering the terms in their occurrence contexts, the technical and scientific text itself.

Krieger and Finatto (2004) point out several common points between the theoretical and practical study of Terminology and Translation. For instance, as the need for specialised translation has increased in the last years and Terminology is becoming more important in the education of professional translators, analogously, techniques for automatic MWT identification play a fundamental role for MT systems that employ some semantic processing. However, while translation professionals quickly realised the importance of terminology for their work, this mutual dependency is not reflected by current NLP systems. Our work does not intend to completely cover this huge gap, but we think that a systematic and application-oriented evaluation of current MWT identification methods and techniques on a domain-specific corpus could indeed help to improve the quality of current MT systems. This approach could not only generate better translations but also (and perhaps most importantly) provide tools for quickly porting a system from one domain to another, since domain adaptation is a real challenge for MT systems.

There are three main reasons for which we decided to focus on multi-word terminology instead of general terms: MWTs are frequent, essentially different from single-word terms and a big challenge for NLP. First, Jackendoff (1997) estimates that, in the general lexicon of a native speaker, multi-word units correspond to more than half of the entries. For Krieger and Finatto (2004), when it comes to the specialised lexicon, studies show that more than 70% of the terms are complex lexical units, composed of more than one word. They go further in the characterisation of terms, affirming that terminological units are often nouns and compound nouns (e.g. *virus enhancer*, *T cell receptor*), not rarely presenting Greek or Latin prefixes and suffixes (*antibody*, *hepatitis*), acronyms (*HIV - Human Immunodefficiency Virus*, *EBV - Epstein-Barr Virus*), abbreviations (*Europarl - European Parliament*) and non-verbal symbols like formulae and equations (H_2O, $O(n^2)$). Second, Natural Language Processing methods that deal with MWTs are essentially different from lexical acquisition techniques for single words. Terminology extraction for single-word units is generally based on the context in which they occur, or on their morphosyntactic and semantic properties. On the other hand, MWE techniques often stay on a shallow level, trying to identify the relations between the words that compose a multi-word term regardless of its context. Since there is, at present time, no unified technique that is able to handle elegantly enough both, single-word and multi-word constructions, we decided to focus on the latter. Third, multi-word phenomena concerning not only specialised but also general-purpose lexicon (e.g. collocations and institutionalised phraseology) constitute a big challenge and, in most cases, an obvious weakness in current NLP technology. Problems arise when some degree of semantic interpretation is needed: because of their non-compositional semantics, syntactic constraints and selectional preferences (most of the MWEs are fixed or semi-fixed constructions), these multi-word elements are often at the root of problems like missing lexical entries, incorrect sentence parses and awkward literal translations. In short, we expect that a systematic evaluation of MWE extraction methods focusing on specialised corpora might potentially improve not only our understanding about these phenomena but also the overall quality of NLP systems that should integrate this kind of processing.

In short, correctly dealing with multi-word terminology is one of the problems found in current multi-lingual systems, not only for pure MT on domain-specific texts but also for specialised multi-lingual and cross-lingual information access and retrieval. Therefore, our main goal is to perform a careful and thorough evaluation of current multi-word expressions technology applied on a domain-specific corpus, namely the Genia corpus, and in a second moment to study potential ways for integrating the extracted multi-word terminology with a Statistical Machine Translation (SMT) system.

The remainder of this document is organised as follows: first, we present a bibliographic review to discuss some related work on terminology, multi-word expressions and machine translation (chapter 2). Second, we describe in detail the experimental setup and the steps and resources required to achieve our goals (chapter 3). Third, we present a thorough and systematic evaluation of the proposed multi-word terminology extraction methods, as well as an application-oriented evaluation to a Machine Translation system (chapter 4). Finally, we present a discussion on the global results and sketch future directions for our research (chapter 5).

2 RELATED WORK

One of the biggest challenges in domain-specific multi-lingual access is the automatic detection of terminology. From a computational perspective, terminology is a complex phenomenon in natural language not only because recognising terms in raw text involves the disambiguation of words in their contexts, (e.g. *collision* is a computer networking term in *packet collision rate* while it is not a term in *the cars had a collision*) but also because it involves the automatic identification of domain-specific multi-word lexical units and their further processing. Multi-word terminology units are a specific type of Multi-Word Expression (MWE), i.e. a sequence of words that co-occur frequently or whose semantics are not directly derived from the meaning of each of its parts. In the next sections, we present a discussion of some related work which deals with generic multi-word expressions and also some language- and type-specific MWE extraction and classification methods (§ 2.1). Then, we discuss some selected papers on (multi-word) terminology extraction and processing, including some experiments performed on the Genia corpus (§ 2.2). Finally, we make a brief introduction to Statistical Machine Translation (SMT), on which we intend to validate our results (§ 2.3).

2.1 About multi-word expressions

The term “MWE” covers a large set of distinct but related phenomena, ranging from very idiomatic ones like verb-particle constructions (e.g. *give up* in *She gave up dance classes*), to highly compositional examples like medical terminology (e.g. *blood cell* has compositional semantics but collocational occurrence). MWEs are a challenging problem for NLP because of their flexible and heterogeneous nature, that is, they are difficult to identify and to treat without human intervention. In NLP systems, there are two main techniques to deal with multi-word expressions: the words-with-spaces approach and the compositional approach (Sag et al. 2002). The first consists in considering MWEs as a purely lexical phenomenon and thus inserting these entities in system dictionaries as singleton lexemes. According to the classification proposed by Sag et al. (2002), words-with-spaces can handle only fixed MWEs, but semi-fixed and variable expressions, sometimes highly productive, cannot be correctly dealt with at the lexical level. The compositional approach, on the other hand, treat the component words of a MWE using standard lexical entries, and thus allows the insertion of MWEs through special constraints on selected grammars rules. However, it is sometimes inadequate for expressions with very limited semantic or syntactic productivity as it allows the components of a more rigid MWE to have the full flexibility of simplex words, resulting in unnatural variations such as *ad extremely hoc* or *to kick millions of buckets*. Although considered as a challenging problem, MWE processing is a fundamental requirement of every NLP application performing some degree of semantic interpretation, for instance for a MT system to translate a domain-specific term (e.g *blood cell*), it needs to identify it as an atomic lexical entry and to deal with eventual inflection, intervening material and semantic flexibility.

In addition to their heterogeneous characteristics, it is too costly and time-consuming to manually list multi-word expressions one by one in a (multi-lingual) lexicon as are they are too numerous: Jackendoff (1997) estimates that the size of the multi-word lexicon of a native speaker is comparable to the number of simplex words that he/she knows. Given their importance and permeability in language, techniques for the automatic acquisition of MWEs have been proposed using statistical and symbolic techniques, including association measures, distributional methods, syntactic and semantic features and machine learning. Most of the works on MWEs focus on a specific task[1]:

- **Identification:** This task deals with MWEs *in vivo*, i.e. it is necessary to identify MWE occurrences in real-world corpora. This constitutes a major problem for NLP because simply matching raw text against MWE dictionaries does not suffice to obtain acceptable recall. Given morphological, syntactic and semantic variability, it is necessary to define the bounds of a MWE taking the context into account, through morphosyntactic patterns, shallow or deep syntax and/or statistical properties.

- **Interpretation:** Two levels of interpretation, syntax and semantics, can be investigated. Discovering the syntactic relations between the component words of a MWE is essential e.g. for compound nouns where bracketing is fixed although arbitrary. Here, it is also important to classify MWEs according to their idiomaticity level. Finally, semantic interpretation helps to identify the implicit semantic relations between the words and is essential for systems that perform deep linguistic processing.

- **Classification:** Classification differs from interpretation because it assumes compositional candidates and tries to assign them to pre-defined classes. Machine learning classification and clustering methods are used then to create (hierarchical) semantic associations between MWEs and thus disambiguate them.

- **Applications:** The previous three tasks are important in a large number of NLP applications, including information extraction, machine translation, information retrieval, textual entailment, etc.

Our work investigates MWE *identification* and their impact in *applications*. We will not investigate methods for MWE classification and interpretation. When it comes to MWE identification, most methods to identify MWEs first generate a preliminary list of MWE *candidates* that are, in a second step, filtered automatically and/or manually before further processing. For the first step in the identification of MWEs, for instance, shallow approaches like Part Of Speech (POS) tag patterns are often used but have low performance when the extracted MWEs are highly flexible. Deeper approaches, on the other hand, can be used for more targeted identification. For example Seretan (2008) uses the output of syntactic parsing to extract collocation candidates from corpora, obtaining impressive results on several languages. Analogously, Baldwin (2005) uses several levels of syntactic features to generate verb-particle construction candidates from a text, proving that deeper candidate extraction yields higher recall and better precision. An n-gram error mining technique is used by Villavicencio et al. (2007) to extract generic MWE candidates from the British National Corpus. Manning and Schütze (1999) show that POS patterns can have surprisingly good performance on relatively small domain-specific corpora, as we will discuss in the next section. The identification of compound nouns is addressed by Protaziuk et al. (2007) who propose an approach that combines data mining methods with shallow lexical analysis for discovering compounds.

[1] This classification is freely inspired on the call for papers of MWE09, available at `http://multiword.sourceforge.net/PHITE.php?sitesig=CONF&page=CONF_40_MWE_2009___lb__ACL__rb__`

Multi-word expressions filtering is often a semi-automatic process in which statistical measures help to filter the candidates according to their association strength. The evaluation of this task can be regarded as an IR problem in which true MWEs should be ranked first in a list of candidates. Villavicencio et al. (2007) present a technique based on statistical association measures to classify generic MWE candidates automatically generated by a deep parser. Seretan (2008) equally focuses on association measures to filter MWE candidates in a multi-lingual evaluation environment. Comparative evaluations of association measures have shown that there is no one measure for extracting multi-word units, but that one should rather select a set of techniques that will perform better for a specific type and language (Evert and Krenn 2005, Pearce 2002). Since corpus statistics are often inefficient because of data sparseness, there has been intense research on using the Web as a corpus for collecting the n-gram statistics needed for association measures (Keller et al. 2002, Zhang et al. 2006, Villavicencio 2005, Grefenstette 1999, Kilgarriff and Grefenstette 2003). The combination of these measures using machine learning techniques has also been explored. In some of these experiments Ramisch et al. (2008), using a single machine learning technique, found that statistical association measures are more useful in language- and type-specific MWE classification than in a generic task. Pecina (2008) has looked at feature selection, using more sophisticated machine learning techniques to select good features among a large list of 84 association measures for the identification of multi-word terms. Some research also combines linguistic and statistical information to distinguish true MWEs among all the putative candidates. Pearce (2001), for example, uses WordNet as the basis for collocation extraction from text, employing synonymy relations to distinguish collocational from anti-collocational candidates (e.g. strong tea vs *powerful tea).

Interpretation of MWEs does normally take into consideration the type and language of the construction. The automatic interpretation of English verb-particle constructions is addressed by Ramisch et al. (2008), who use several resources like corpus statistics, the Web, Wordnet and Levin's verb classes to classify the verb-particles according to their idiomaticity. Nakov and Hearst (2005) address the problem of compound noun bracketing by recursively comparing inter-word adjacency and dependency measures for each partition of the compound. In particular, the semantic classification of compound nouns has received considerable attention. Girju et al. (2005) use a distributional approach to build models for doing a fine-grained semantic classification. In order to restrict the range of possible semantic relationships between the head and the modifier, which is potentially non-finite, several works focus on compound nominalisations where the head has a morphologically related verb and the modifier fills a syntactic argument relation with respect to this verb (Lapata 2002).

There has been little emphasis on MWE applications for real-life NLP systems. Our work tries to explore this gap in domain-specific texts. However, we believe that considerable advances need to be made to integrate multi-word constructions with current NLP technology.

2.2 About multi-word terms

Multi-word terminology is not a recent concern of Computational Linguistics. Among earlier efforts to automate terminographic extraction is the work of Justeson and Katz (1995). They use a set of selected POS tag patterns to extract MWT candidates and then filter them using their raw frequencies in the corpus. Although very simple, this methods yields accurate results when high recall is not required, which makes it popular among NLP systems that perform some domain-specific task.

Frantzi et al. (2000) present a mixed approach to extract multi-word terminology from text in which candidates are extracted using shallow POS patterns and then a statistical test, the C-value, is used to guarantee that the extracted pattern is actually a MWT. The C-value measure can handle nested terms in a sophisticated way, extracting arbitrarily long terms. They compare their technique with raw frequency filtering though they do not take into account advances in MWE association measures. Smadja (1993) proposed Xtract for collocation extraction from text, using a combination of n-grams and a mutual information measure. On general-purpose texts, Xtract has precision of around 80% for identifying collocational units. Although, in principle, it could be used to extract multi-word terminology from domain-specific texts, as its aim is not high recall, it may not be very effective in identifying a significant amount of arbitrary domain-specific MWEs.

From the syntactic point of view, terms of areas like automotive industry, biomedical articles or chemistry documents often follow a particular pattern of combination of nouns and adjectives, i.e. compound nouns and related constructions. For instance, in the biomedical abstracts of the Genia corpus, the terms are disease and drug names, chemical elements, body parts and other names (Ohta et al. 2002). Thus, techniques for the automatic identification of compound nouns such as the ones described in the previous section, when applied on domain-specific data, could be of great help in multi-word terminological extraction.

Considering MWT interpretation and classification, SanJuan et al. (2005) presents a technique to compare an automatically extracted with a manually built ontology for the Genia corpus. They combine three linguistic resources and techniques, namely statistical association, WordNet and clustering, to build a hierarchical model over manually-annotated terms of the corpus. They then compare their model to traditional clustering algorithms and to the manually created ontology of the corpus.

Our work differs from the previous in several aspects: first, we do not intend to create a hierarchical structure of the extracted terms. Even if we perform shallow POS pattern extraction, we are going to use MWE identification techniques to filter MWT candidates, thus evaluating whether these methods can be applied on domain-specific terminological MWE extraction. Therefore, we do not rely only on corpus raw corpus frequencies to filter the candidates, but also on more sophisticated resources like association measures, Web frequencies and machine learning.

2.3 About machine translation

Machine translation technology is nowadays incapable of generating fully *automatic* translation with at the same time high *coverage* and linguistic *precision*. Therefore, intermediary solutions have been proposed that choose to optimise two of these aspects in spite of the third. For instance, collaborative translation systems, like the one described by Bey et al. (2006) propose a wide-coverage translation environment where the trade-off between automaticity and quality is customisable. Analogously, Boitet et al. (2008) present a review of software and site localisation techniques, proposing a collaborative gateway for multi-lingual access on the Web.

The field of Statistical Machine Translation (SMT) has received special attention in the last few years for several reasons: very large parallel corpora became available, successful online SMT systems have been launched[2] and with SMT, one can quickly build a translation system between every language pair for which there is enough parallel data, avoiding the costly effort of coding transfer rules by hand. SMT systems learn translation models automatically from parallel data, through probabilistic machine learning algorithms. One of the the most popular translation models used to date is the IBM model described by Brown et al. (1993), where the score of a hypothesis (a partial translation) is the result of the combination of several sub-models that include the word fertility (the probability of a word being translated as more than one word), the word-to-word translation model and the reordering model (probability that a sequence of words $s_1 s_2$ is translated as $t_2 t_1$). According to Lopez (2008), numerous variations of this model have been proposed, and there is in particular a very clear distinction between *generative* models (as IBM-4) that estimate the probability of a source sentence given a possible translation (i.e. $t = argmax\ p(s|t_i)$), and *descriptive* models, where the translation is

[2] For instance, Google claims that they use statistical MT in their translation service. `http://www.google.com/intl/en/help/faq_translation.html`

evaluated with respect to a (log-linear) feature function ($t = argmax\ p(t_i|s)$).

Statistical translation models are based on parallel corpora aligned at the word level. However, since manual word alignment of corpora is nearly impossible, given the dimensions of the data, this parameter estimation problem is usually solved through an Expectation-Maximisation (EM) algorithm. The EM algorithm is an unsupervised learning technique that tries to maximise the model parameters based on the probability of seen data and the model of unseen data (Dempster et al. 1977). This word-alignment approach can be improved with some heuristics, for instance we could generate alignments in both directions (from source to target language and vice-versa) and then only consider their intersection. Afterwards, some rules (which must be tuned according to the task) can be used to expand the alignments and cover all the words.

In terms of MWE identification, one relevant distinction made in SMT systems is that between *word-based* and *phrase-based* SMT. Here, the term *phrase* is employed as a synonym of word sequence, which does not necessarily correspond to a linguistically coherent phrase (e.g. a noun phrase or a prepositional phrase).[3] Phrase-based models are able to capture some syntactic restrictions about word adjacency and limited local reordering. In phrase-based models, for instance, information about fixed or semi-fixed multi-word units is easily integrated into the translation model whereas highly compositional phenomena such as verb-particle constructions are too complex to be handled with this n-gram-based approach. Therefore, the trend in the SMT area seems to be the integration of higher level information (morphology, syntax) into phrase-based SMT, in so-called factored MT systems.

Few works explore the relationship between word alignment for MT and MWE extraction. Among them, we cite Melamed (1997), that looks into the discovery of non-compositional compounds from parallel corpora using statistical methods. Similarly, Caseli et al. (2009) focus on multi-lingual MWE extraction as a by-product of the word alignment discovery process and carefully evaluates the quality of MWE candidates extracted using $n : m$ word alignments. This work provides an in-depth analysis of some of these methods and aims at shedding some light on MWE identification for improving the quality of MT systems.

[3] A fairly comprehensive evaluation of phrase-based SMT can be found in Koehn et al. (2003).

3 BIOMEDICAL TERMINOLOGY IDENTIFICATION

In this chapter, our goal is to describe the process that we employed in order to deal with terminology and multi-lingualism in a domain-specific corpus of the biomedical domain, the Genia corpus. We will start by describing our corpus (§ 3.1) and then we will explain the four steps that we performed in our evaluation: pre-processing (§ 3.2), Part Of Speech (POS) tag pattern extraction (§ 3.3), learning through association measures (§ 3.4), and MT integration (§ 3.5).

3.1 The Genia corpus

The Genia corpus is a collection of texts retrieved from the *Medline* library through the MEdical Subject Headings (MeSH) thesaurus, and contains paper abstracts for the keywords “human”, “blood cell” and “transcription factor” (Ohta et al. 2002). After downloading version 3.0[1], correcting minor tagging errors and eliminating truncated (thus ungrammatical) sentences of the XML file, we analysed its structure. The corpus has several annotations, including Part Of Speech tags and terminology. Table 3.1 summarises the dimensions of the raw corpus, containing 2,000 abstracts with around 18.5K sentences and 490.7K tokens[2], yielding an average sentence length of 26.5 tokens per sentence. Compared to generic corpora like Europarl (40M tokens) or the British National Corpus (100M tokens), the Genia corpus is small. However, it contains domain-specific information that makes it a very interesting resource for terminology identification.

Table 3.1: Raw corpus statistics.

Abstracts	2,000
Sentences	18,519
Tokens	490,752
Average sentence length	26.5 tokens
Annotated terms	97,876
Annotated multi-word terms	55,487
Average terms/sentence	5.29
Average multi-word terms/sentence	3
Average term length	2.06 tokens
Average multi-word term length	2.86 tokens

The corpus annotation for terminology involves terms taken from a domain-specific ontology

[1] The Genia corpus is freely available at `www-tsujii.is.s.u-tokyo.ac.jp/~genia/topics/Corpus/`.

[2] To avoid ambiguity, we will refer to *token* for any occurrence of a word form in the corpus. The term *word*, on the other hand, designs a type, that is a distinct word form, counted once for all its occurrences in the corpus. Morphological inflection distinguishes one word from another.

(Kim et al. 2006). They include single- and multi-word terms such as disease names (e.g., *fibroblastic tumours*) and cell types (e.g., *primary T lymphocyte*). The corpus contains around 97.8K annotated terms (including several instances of the same term), with more than half of it being composed of more than one word. This proportion agrees with theoretical estimations of multi-word terms presented in chapter 1. The annotated terms have around 2 words in average, whereas if we only consider multi-word terms, average length grows to almost 3 tokens. Terms can be arbitrarily large, the maximum length of a term in the corpus is 23 tokens. On average, each sentence contains around 5 terms, from which 3 are multi-words. This demonstrates how much domain-specific information is present in the text, showing again the importance of correctly handling terminology in whatever NLP application dealing with domain-specific texts such as our biomedical abstracts.

The algorithms and experiments described in the remainder of this chapter were implemented as a set of scripts, most of them written in `Python`, `awk` and `bash` dealing with XML and TAB-separated text as intermediary or temporary files. This material is freely available under GNU-GPL at `www.inf.ufrgs.br/~ceramisch/tcc.zip`. A fragment of the output of our program can be consulted in appendix E.

3.2 Pre-processing

The first step of our MWT extraction method is to pre-process the corpus so that our algorithms can have optimal performance with minimum noise. All the steps described in this section are based on heuristics and simple transformation rules. Therefore, they are homogeneously applied on the whole corpus. In further sections, we will separate a training set from a test set, but for the moment there is no point in splitting the corpus.

First, we lowercased[3] the corpus and generated a simplified XML format containing abstracts, sentences and multi-word terms, eliminating all single-word terminology annotation. To better understand the issues involved in pre-processing, let us consider the following example sentence taken from the raw corpus:

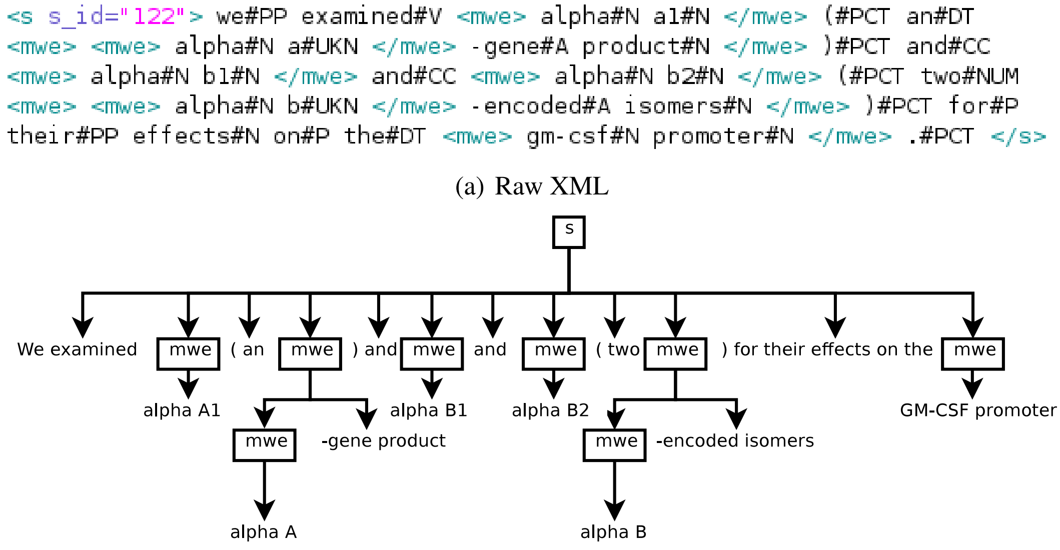
Example 3.1 *We examined alpha A1 (an alpha A-gene product) and alpha B1 and alpha B2 (two alpha B-encoded isomers) for their effects on the GM-CSF promoter.*

Figure 3.1(a) shows the same example sentence annotated for multi-word terms in XML format. It is easier to understand the structure of the sentence if we look at the XML tree it represents, as in figure 3.1(b). Note that most of the multi-word terms in the corpus present a highly nested structure: it is not rare to find three or four nesting levels. Since we did not employ an elegant solution for them, we will consider each nested multi-word term as a separate unit, independently of its internal structure.

At this point, one could wonder why are we trying to extract terms from a corpus that is already annotated for terminology. As stated in section 1.2, our goal is to evaluate how well MWE extraction techniques perform in the task of multi-word terminology extraction on domain-specific documents. The annotation will not be used to extract the terms, but only to evaluate them. Without an annotated corpus as gold standard, we would not be able to perform automatic evaluation of the technique. Therefore, we need to make an important distinction between a candidate set and a gold standard.

Definition 3.1 (MWT candidate) *A MWT candidate is a sequence of words in the corpus that is possibly (and potentially) a MWT. When we extract a MWT candidate from the corpus, we do not*

[3] Case information is important in domain-specific texts (e.g. chemical elements *NaCl* or named entities *Bill Gates*, *March*), but can also introduce noise when capitalised words at the beginning of a sentence differ from their occurrences in other positions. Sophisticated lowercasing schemes exist to deal with such phenomena, and could be used in future work.



```
<s s_id="122"> we#PP examined#V <mwe> alpha#N a1#N </mwe> (#PCT an#DT
<mwe> <mwe> alpha#N a#UKN </mwe> -gene#A product#N </mwe> )#PCT and#CC
<mwe> alpha#N b1#N </mwe> and#CC <mwe> alpha#N b2#N </mwe> (#PCT two#NUM
<mwe> <mwe> alpha#N b#UKN </mwe> -encoded#A isomers#N </mwe> )#PCT for#P
their#PP effects#N on#P the#DT <mwe> gm-csf#N promoter#N </mwe> .#PCT </s>
```

(a) Raw XML

(b) Tree structure

Figure 3.1: Example sentence annotated for multi-word terms.

consider term annotation, but only lexical, syntactic and frequency characteristics of the candidate itself.

Definition 3.2 (MWT Gold Standard) *The MWT Gold Standard (GS) is the set of all annotated terms in the corpus. In order to be attested, a MWT candidate must be in the gold standard. The GS is used to automatically evaluate the quality of the extracted MWT candidates list.*

After pre-processing, the gold standard of the Genia corpus contains 28,340 distinct MWTs. Notice that there are 55,487 occurrences of multi-word terms in the corpus, i.e. terms in the GS might occur more than once in the corpus. On the other hand, after pre-processing and POS pattern extraction, we obtained a set of 63,210 distinct MWT candidates. For any arbitrary set C of MWT candidates, it is possible to calculate precision $P(C)$ and recall $R(C)$ with respect to a given GS (where $\#(\cdot)$ denotes the size of a set):

$$P(C) = \frac{\#(C \cap GS)}{\#(C)} \tag{3.1}$$

$$R(C) = \frac{\#(C \cap GS)}{\#(GS)} \tag{3.2}$$

We will also use the harmonic mean of precision and recall, i.e. the *F-measure*, to evaluate the quality of a candidate set C. Throughout this and the next chapter, f-measure is calculated as follows:

$$F(C) = \frac{1}{\frac{1}{2} \times \left(\frac{1}{P(C)} + \frac{1}{R(C)} \right)} = \frac{2 \times P(C) \times R(C)}{P(C) + R(C)}$$

The next sections describe in detail our pre-processing heuristics concerning POS tagging, tokenisation, lemmatisation and acronym removal. It is important, at this point, to warn the reader that some of these steps are domain- and/or language-dependent, i.e. they cannot be directly applied to an arbitrary MWT extraction environment.

3.2.1 Tokenisation and POS tags

A first problem arises from an incompatibility between some Part Of Speech tags and the term annotation. Consider the following fragment extracted from example 3.1, where we used brackets to delimit terms and the special separator "/" for the POS tags:

Example 3.2 `[ [ alpha/N a/UKN ] -gene/A product/N ]`

In this example, `N` stands for noun, `A` for adjective and `UKN` for unknown. During pre-processing, we replaced detailed POS annotation of the Genia tag set (e.g. *NN*, *NNS*, *NNP*, ...) by simplified morphosyntactic tags (e.g. *N*) that are more adapted to our POS pattern selection rules described in section 3.3. The tag transformation rules are listed in appendix C.

The term annotation of Genia breaks words on hyphens and slashes, but the POS tag annotation does not. Therefore, the word `a-gene/A` of example 3.2 is split into two parts, one of which has an unknown tag. Our MWT extraction method uses POS patterns to extract term candidates, so we cannot allow such inconsistencies in our input. A straightforward solution for this problem would be to ignore all sentences containing such noisy patterns. However, after removing all single-word term annotation, there are still 3,152 sentences containing `UKN` tags, representing 17% of the corpus.

Therefore, we employed a simple heuristic that switches the position of the erroneous word and of the term annotation marker, bringing the POS-tagged word inside the term and concatenating it with the preceding or succeeding word. In example 3.2, we generate the following annotation:

Example 3.3 `[ [ alpha/N a-gene/A ] product/N ]`

With this simple modification, we are able to correct 3,071 tagging problems and only discard 81 sentences at the price of less detailed information in term annotation. Even with this modification, we still encountered leading slashes and dashes in some tokens. Thus, for each token in the original corpus we generated a re-tokenised version eliminating all leading and trailing punctuation signs. We evaluate the impact of re-tokenisation in section 4.1.1.

3.2.2 Lemmas and plurals

The second problem concerns the fact that some candidates have number inflection when they occur in the corpus, i.e. they might be either singular or plural. Ideally, we would like *T cell receptor* and *T cell receptors* to be considered as a single MWT candidate. Therefore, we need to lemmatise the latter so that it becomes equal to the former. If we used an off-the-shelf lemmatiser, however, it is likely that we would obtain tokens like *bind* instead of *binding* for the compound noun *binding site*. Another possibility would be to implement a very simple lemmatiser that eliminates all the "s" from the end of the words, taking at least regular plurals into account. However, in a domain such as biomedicine, we cannot transform words like *virus* into *viru** or *viruses* into *viruse**.

Finally, for greater control of the process and outcome, we implemented some simple yet robust rules to generate the singular form of an arbitrary word. Our singularisation algorithm tries to generate all possible singular forms based on English plural regular and semi-regular terminations and then searches the Web for the singular candidates. The candidate that appears in most pages is then returned, as described by algorithm 2 in appendix D. The algorithm does not cover cases like *children* → *child* or *criteria* → *criterion*, nonetheless it increases the quality of both the candidates and the GS, as discussed in section 4.1.1.

3.2.3 Acronyms and parentheses

A third problem appears because biomedical text (and scientific text in general) tends to contain acronyms inserted in the text to avoid useless repetition. For instance, in our text we created acronyms for recurrent terms such as Multi-Word Term (MWT) and Natural Language Processing (NLP). According to stylistic conventions, acronyms are normally placed inside parentheses right after a term or expression. Since we are dealing with multi-word terminology, some of our term candidates do contain such intervening material, as shown in example 3.4:

Example 3.4 (Intervening acronym) `[ [ granulocyte-macrophage colony-stimulating factor ] (gm-csf) promoter ]`

Such acronyms introduce noise in our results, and are very frequent: there are 1,662 out of 54,629 MWT occurrences following this pattern (around 3% of the GS). In order to remove them, we cannot simply eliminate all the parentheses in the text, since important information could be lost. Instead, we implemented another set of heuristics to detect the acronyms and their corresponding extended form in the text. Algorithm 1, described in detail in appendix D, is able to generate a list of acronyms and their correspondences from the corpus. The set of attested acronyms is then used to eliminate all of them occurring inside parentheses. This list can be considered as a by-product of our pre-processing steps and will be used further in section 3.4.

The algorithm returns a match score that will be higher if the acronym corresponds to a series of initials and numbers in the same order as they are observed in the extended candidate. If the maximum LCS-score is greater than a threshold, the candidate is added to the list of attested acronyms. All the values of the scoring scheme were empirically defined based on several trials. It is interesting to notice that case information could be useful at this point to solve ambiguous cases or even to create a fine-grained scoring scheme. However, since we performed case homogenisation in previous steps, such information is not available anymore. Moreover, similar techniques have already been employed to solve a large set of problems in NLP, including multi-word expressions extraction itself. Longest common subsequence algorithms have been used to identify MWEs in corpora because they can detect co-occurrence with some flexibility (Duan et al. 2006). The solution implemented here is proposed by the author following some example-based machine translation techniques (Biçici and Dymetman 2008). Such an algorithm could be particularly useful in the automatic evaluation of MWEs, to provide fuzzy matches between a candidate and an entry in a dictionary. A didactic and detailed explanation of string matching algorithms for computer science can be found in Gusfield (1997).

Table 3.2 shows some examples of automatically extracted acronyms, including cases in which the acronym is not a trivial concatenation of the initials of the words that define its meaning (here called extended form). A detailed evaluation of these heuristics is presented in section 4.1.2.

Table 3.2: Example of automatically extracted acronyms and their extended forms.

Acronym	Extended form	Acronym	Extended form
AB	antibody	AC	adenylyl cyclase
ACRF	advanced chronic renal failure	ACTH	adrenocorticotropic hormone
AD	alzheimer 's disease	AD	atopic dermatitis
AD	adenovirus	AD2	adenovirus 2
AD3	**a and b***	AD5	adenovirus serotype 5
3OMG	3 o methyl d glucose	9CISRA	9 cis retinoic acid
5'UTR	5' untranslated leader region	5LOX	5 lipoxygenase

3.3 Part Of Speech patterns

In the previous section, we executed a series of pre-processing steps to potentially improve the quality of our MWT candidates. From a raw XML corpus, we re-tokenised noisy words, replaced and filtered out unknown POS tags, lemmatised plural forms and removed parenthesised acronyms. Now that we obtained a clean version of the initial corpus, it is time to extract the multi-word term candidates from the text.

Table 3.3: Some selected POS patterns used to extract MWT candidates, with occurrence examples.

Pattern	**Example**
N-N	*zinc sulfate*
A-N	*malignant tissue*
N-N-N	*zinc finger type*
A-N-N	*lymphoblastoid cell line*
A-A-N	*xenoreactive natural antibody*
A-N-N-N	*cyclic adenosine monophosphate signaling*
N-N-N-N	*dna binding protein complex*
N-A-N	*colostrum inhibitory factor*
A-A-N-N	*fetal nucleated blood cell*
N-A-N-N	*kappa light chain gene*
N-NUM	*nucleotide 46*
A-N-N-N-N	*whole cell dexamethasone binding assay*
A-CC-A-N	*young and aged subject*
N-N-N-N-N	*lymphocyte glucocorticoid receptor binding parameter*
N-N-A-N	*laser scanning confocal microscopy*

Our set of MWT candidates is composed of sequences of words of size $n \geq 2$ that follow recurrent Part Of Speech patterns in the corpus. Since the discovery of POS patterns is based on observation of the corpus, at this point we need to separate part of the data for test and use the rest in training. Previous steps were based on intuitive heuristics and were thus validated and applied on the entire corpus. Here, however, we are observing and learning from the corpus, meaning that evaluation cannot be performed directly on the same data to avoid overfit. Henceforth, we will refer to the *Genia-test* part corresponding to the last 100 abstracts of the corpus, i.e. 895 sentences and almost 24K tokens. The *Genia-train* part corresponds to the remainder of the original pre-processed corpus and will be used throughout the next sections of the current chapter. We will reserve *Genia-test* until we present the evaluation of our method in chapter 4.

To extract MWT candidates from the corpus, we proceeded as follows:

1. Based on term annotation, we extracted the POS patterns of the *Genia-train* gold standard.

2. We selected a set of 118 patterns that occurred more than 10 times.

3. From these top-recall patterns, we selected those with precision in *Genia-train* higher than 30%, in a total of 57 patterns.

4. Our candidate list is composed by all n-grams of the corpus that obey to these 57 POS patterns.

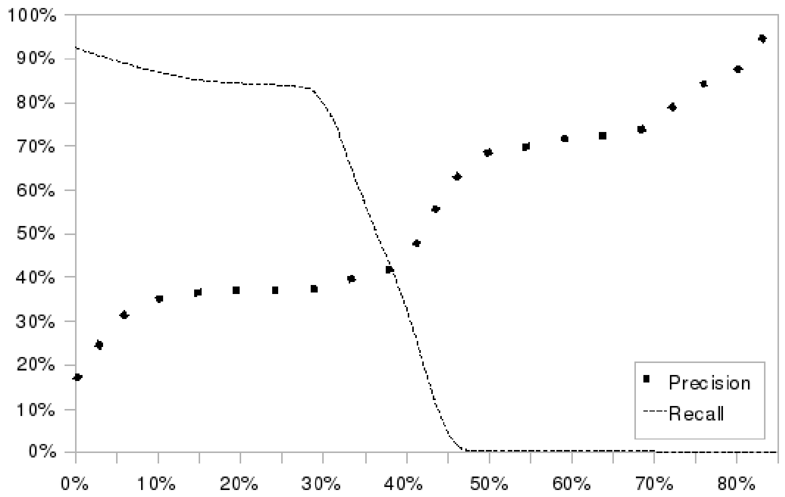


Figure 3.2: Precision and recall of 118 top-recall POS patterns in *Genia-train*.

Some of the patterns selected with this procedure are listed in table 3.3[4]. We can observe the variability of the terms in the corpus from the number of different POS patterns they present. There are 1,177 different sequences of tags, 700 of which are hapax legomena[5]. Therefore, we initially selected only those appearing more than 10 times, keeping the POS patterns reasonably small and avoiding irrelevant noise. At this point, we were extracting 92.57% of the terms with a precision of 16.50%. With those patterns, we then extracted all possible candidates from *Genia-train* and counted the number of true and false positives. The precision and recall values were calculated for different precision thresholds, as depicted in the graphic of figure 3.2. We can see that, while precision is roughly linear with the threshold, recall drops drastically around 30%. Therefore, from the initial 118 patterns, we selected only 57 using as threshold a precision value of 30%.

The output of this step is a list containing 60,585 multi-word candidates that obey the selected POS patterns. Among them, we have 22,507 true instances and 38,078 random word combinations, yielding a baseline precision of 31.52%. Therefore, with pattern selection we almost doubled precision at the price of loosing 3 points in recall, which now is 89.43% of the candidates contained in the GS. For a detailed evaluation, please consult section 4.2.1.

3.4 Association measures

Once we obtained a list of candidate MWTs, we would like to filter them based on their association strength. The notion of "association strength" is related to the probability of co-occurrence of the words composing a candidate. In this work, we will focus on statistical evidences for strong association, according to recent methods for MWE extraction (Evert and Krenn 2005, Ramisch et al. 2008, Pearce 2002, Pecina 2008).

By now, it should be clear that we will deal with the frequencies[6] and probabilities of words in the corpus, and that sparseness will be an obstacle. This is a well-known phenomenon in NLP, related to the zipfian distribution of words in the text and to a the so-called "long tail", referring to a rank plot of the distribution of words in a text (Newman 2005, Baayen 2001). A simple example is the

[4] The complete list of POS patterns can be consulted in appendix C.

[5] The Greek term *hapax legomena* is employed here to denote any element (word, POS pattern, MWT, etc.) occurring only once in the data set.

[6] In this work, we will use the term*frequency* as a synonym of *occurrence count*, according to NLP reference textbooks (Manning and Schütze 1999).

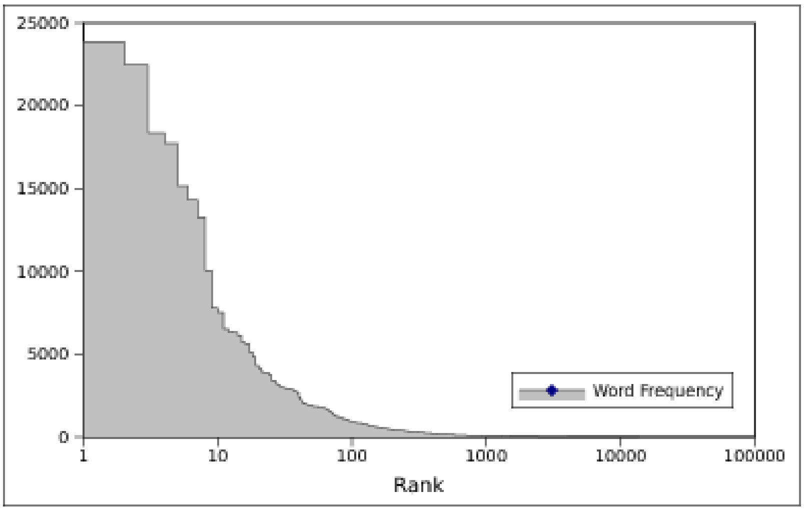


Figure 3.3: Rank plot of the *Genia-train* vocabulary.

observation of the histogram in figure 3.3, where we plotted in logarithmic scale the word frequency distribution of *Genia-train* according to the rank position of words in the corpus vocabulary.

In order to deal with data sparseness, several works suggest to use the World Wide Web as a linguistic resource that could provide word and n-gram frequency values, claiming that it is possible to approximate the occurrence of a word in general language by looking up how many Web pages contain that word. This word frequency approximation by a document frequency index surely introduces some noise into the word counters, but this effect should be minimised by the size of the Web (Grefenstette 1999, Kilgarriff and Grefenstette 2003, Villavicencio 2005, Keller and Lapata 2003). An interesting problem arises when we try to transform Web frequencies in Web probabilities, because we need to answer to the question "How many documents are there in the Web?". Since this is an active research subject and that we do not require precision because this number will become a scale factor with the same weight for each candidate, we will be happy to use a reasonably high estimation. In the following equations, we assume that the Web contains around 50 billion documents, which is probably overestimated but "good enough" in this context[7] (de Kunder 2007).

Despite of sparseness, we believe that there is valuable information in the word frequencies of the candidates, so we extracted not only candidate frequencies but also marginal word frequencies from the corpus. Here, we used the list of acronyms generated in section 3.2.3 to expand the frequencies of the words that compose an acronym, e.g. for every instance of *hiv*, we also increment the counters for the words *human*, *immunodeficiency* and *virus*. This simple substitution is not only semantically coherent (the meaning of the acronym is expressed by the words that compose it) but also diminishes the number of hapax legomena in the vocabulary, from 4,903 to 4,835.

Therefore, we use two sources: the *Genia* corpus itself, and the Web through the *Yahoo!* Web Search API[8]. We implemented a simple cache mechanism that stores older searches for reuse while they are valid and that distributes processing over several machines[9]. Even though we use *Yahoo!*'s search index as an abstraction for the whole Web, we will call it the *Web* corpus. At this point, then, we have a list of candidates containing:

- The re-tokenised candidate (with words in the singular form);

[7] Up-to-date estimation obtained on May 30, 2009 at `http://www.worldwidewebsize.com/`.

[8] Available at:`http://developer.yahoo.com/download/download.html`

[9] Since *Yahoo!* limits the service to 5,000 queries per day per IP address, we distributed processing to work around this constraint.

- The Part-of-Speech pattern of the candidate;
- The frequency of the candidate and of its words in the *Genia-train* corpus;
- The frequency of the candidate and of its words in the *Web* corpus.

The validation of the candidate MWTs is done using a set of statistical Association Measures (AM): raw probability estimated through the maximum likelihood estimator (prob), Pointwise Mutual Information (PMI), Student t-test (t) and Dice coefficient (Dice). These typical measures of association are summarised in table 3.4 and are calculated based on frequencies obtained in the both the *Genia-train* corpus and the *Web* corpus. Formally, we consider the candidate as a generic n-gram with n word tokens w_1 through w_n. The frequency of a contiguous sequence $w_i \dots w_j$ in a corpus of size N is denoted by $f_C(w_i \dots w_j)$. To measure the strength of the association between the words of the candidate, we compare its observed probability with the marginal probabilities of its component words, expecting that the probability of co-occurrence of the words in a MWT is higher than randomly combining any sequence of n words. The t measure is actually a rank interpretation of a classical hypotheses test. Therefore, null hypothesis supposes statistical independence, i.e. in general human language[10] the frequency of a sequence corresponds to the product of the probabilities of its members scaled by the size of the corpus:

$$H_0 : f_\emptyset(w_1 \dots w_n) = \frac{f(w_1) \dots f(w_n)}{N^{n-1}}$$

Table 3.4: Statistical measures used to validate MWT candidates.

Measure	**Formula**
prob	$\frac{f(w_1 \dots w_n)}{N}$
PMI	$log_2 \frac{f(w_1 \dots w_n)}{N^{n-1} f_\emptyset(w_1 \dots w_n)}$
t	$\frac{f(w_1 \dots w_n) - N^{n-1} f_\emptyset(w_1 \dots w_n)}{\sqrt{f(w_1 \dots w_n)}}$
Dice	$\frac{n * f(w_1 \dots w_n)}{\sum_{i=1}^{n} f(w_i)}$

Most of the recent advances in statistical MWE extraction also use measures based on contingency tables, i.e. instead of looking only at the marginal frequencies of the words, they also consider their probability of “non-occurrence”, building a table with possible combinations of w_i and $\neg w_i$, for all i. Even though they are very robust for rare events and do not rely on unjustified assumptions (e.g. the t-test assumes that words are normally distributed whereas they actually follow a power-law distribution), these methods are only suitable for n-grams where $n = 2$ or $n = 3$, and cannot be intuitively extended for candidates with arbitrary length.

In our case, the *Web* corpus often generates contingency table incoherences, for instance, a bigram $w_1 w_2$ has a frequency of f and the word w_1 appears $f_1 < f$ times in the corpus. Web search engines

[10] Here we use the statistical notation $f_\emptyset(\cdot)$ for the null hypothesis, while the subscript notation $f_C(\cdot)$ is used for the frequency in a corpus C. We will use subscript notation wherever we feel that it is important to indicate from which corpus this information comes from. For frequencies, if no corpus is specified, we will write simply $f(\cdot)$.

tend to generate rough approximations of the number of retrieved documents instead of counting precisely how many results are obtained, mostly because users will never go through the whole results list. Moreover, previous experiments showed that a contingency hypercube based on *Web* frequencies will be so noisy that it would become as useful as random scores. Therefore, we focused only on pointwise and heuristic association measures. Future work could explore the impact of adapting contingency-table methods to arbitrary n-grams, using some divide and conquer technique. The choice of one or a set of statistical association measures remains an open problem in MWE extraction. Even though several works do focus on evaluating association measures for multi-word units identification, the results are hard to generalise in different contexts. An interesting discussion about this topic can be found in Seretan (2008).

Once we have obtained different measures of association for the MWT candidates, we combine them using Machine Learning (ML) algorithms, as in Pecina (2008). Therefore, we use the annotation of the *Genia-train* part of the corpus as classes, i.e. if the MWT candidate is contained in the gold standard, it has the class `MWT`, otherwise it pertains to the class `NonMWT`. Since we are only using the annotation of the training part of the corpus, we can evaluate our method on the test part without overfitting the models generated by the learning algorithm.

A wide number of ML algorithms are available to test over our data set. We tried some of them on our data set: decision trees, Bayesian algorithms, Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP). Decision trees provide intuitive models that are easy to interpret and, during our tests, seem to offer a good cost/benefit ratio in terms of the time taken to build the model and its quality. The algorithm builds up a tree by successively subdividing the set of instances according to an entropy gain measure, trying to minimise the variability in the leaves of the tree.

Bayesian learning is specially fit for numeric data such as our AMs, and thus we expect to obtain good results using this technique. Naive bayesian learning treats each feature as a Gaussian distribution and then tries to estimate the parameters of the distributions, namely mean and variance. We are going to employ a more sophisticated algorithm that builds the entire Bayes net of feature/class dependencies and then uses a heuristic search method to optimise its parameters (transition probabilities).

Finally, SMV and MLP are universal function approximators and are particularly robust on difficult problems (although they are very costly in terms of computational resources). Neural networks are inspired on simple units, called artificial neurons, that store the model of the data on their synaptic weights. Support vector machines optimise the class separation border through numeric methods, being very flexible about the underlying formalism of the border function. In SVM, the "kernel" corresponds to the type of function that is used as class separator. Common kernels are linear, polynomial and Gaussian functions.

A systematic comparison of these learning algorithms techniques, along with the parameters and options used in MWT classification, is presented in section 4.2.4. The model generated in this section is a generic machine for MWT identification in text. The evaluation of of this model is presented in section 4.2.4. Moreover, we performed an application-oriented evaluation integrating the detected MWTs in a statistical MT engine, as described in the next section.

3.5 MT Integration

The Europarl corpus[11] contains more than 1 million sentences from the speeches held in the European Parliament, with each utterance translated into 10 languages. In our experiments, we used only the English-Portuguese (`en_pt`) portion of the corpus because our goal is to create a statistical MT system that translates from an English source to a Portuguese target.

[11] The Europarl corpus v.3 is freely available at `http://www.statmt.org/europarl/`

Even though it would be much more interesting to evaluate our technique on a truly multi-lingual environment including languages like German, in which multi-word constructions tend to be lexically indicated by the concatenation of their members, there are two reasons, theoretical and pragmatic, for the choice of the Portuguese-English language pair. First, a study including different kinds of languages like German, English and French, would be most interesting if we had a parallel biomedical corpus from which we could extract terminology using multi-lingual techniques based on word-level alignment, similar to the methods proposed by Melamed (1997), Caseli et al. (2009). Since at present we do not know of the free availability or even existence of such resources, we could not investigate any further in this direction. Second, Portuguese is the mother tongue of the author and a language for which we could easily find native speakers to perform the evaluation, turning `en_pt` into the natural and only choice for language pair in the evaluation.

Moses is a statistical machine translation package that is widely used in the NLP community (Koehn et al. 2007). It includes both, a training module and a decoder, allowing to automate the whole translation pipeline. In order to create a standard baseline MT system using the Moses package trained on Europarl, we executed the steps described below for state-of-the art performance on general-purpose corpora[12]:

- Pre-processing:
    1. Sentence-align the raw Europarl corpus using the standard Church and Gale algorithm[13].
    2. Filter out bad alignments, XML entities and very long sentences.
    3. Tokenise and lowercase.
    4. Separate two sets of 2K sentences for development (*dev*) and for test (*test*), leaving the reminder for training (*train*).
- Training:
    1. Generate a trigram language model for the target language (`pt`);
    2. Calculate the word alignments using the EM algorithm implemented in the GIZA++ tool;
    3. Generate the phrase table, i.e. the translation model itself (Brown et al. 1993).
- Tuning:
    1. Using a Minimum Error Rate Training algorithm, optimise the feature weights on *dev* set.

With a trained general-purpose MT system on Europarl, we would like to include external MWT information without re-training the models. In our case, external MWT information is obtained automatically from a mono-lingual corpus (Genia) using the methodology described in previous sections. Several techniques have been proposed, for instance, to handle German or Swedish compounds, by first splitting the words of the compound, then translating it or, when German is the target language, by merging the words after translation (Nießen and Ney 2004, Nießen et al. 2000). Most of the approaches (including our own) remove all ambiguity, supposing that the output of the MWT identification step is always correct. However, since we know that this assumption is not true, future work could explore association strength of a MWT candidate as a way to integrate some fuzzy degree of “MWTness” directly into a phrase-based MT system, blurring the sharp border between attested MWTs and false MWT candidates.

[12] These steps are described in WMT08 shared task description, available at `http://www.statmt.org/wmt08/baseline.html`

[13] Europarl is only aligned at the paragraph level.

In our evaluation, two types of integration were tested: *words-with-spaces* and *head replacement*. In the former, the idea is to use a word-with-spaces approach that forces the MT decoder to treat the MWE as a single lexical unit. This would avoid classical problems like word-by-word translation, wrong reordering and other collateral effects of missing lexical entries.

Before inputting the sentence in the MT system, we identify the MWTs using the model generated in section 3.4. Then, for each attested MWT, we join its members using a special marker "#". Thus, the unit will simulate a single unknown word and will be transliterated to the output. Henceforth, this merging technique will be called *words-with-spaces* (`wws`). When Moses decodes the input sentence and generates a Portuguese translation, and in the presence of a domain-specific dictionary or manual translation for the extracted terms, we could replace the multi-word unit by its translation. Such a system could be built using a collaborative MT and corpora management system like SECTra_w (Huynh et al. 2008). In our tests, we simply replaced the "#" marker by a space, keeping the term unchanged in the translation.

The second technique that we employed is the replacement of the term on the source side by its head, for instance *simian epstein barr virus oncogene* becomes *oncogene* and *simian epstein barr virus* becomes simply *virus*. Head discovery is a complicated task which is out of the scope of our work, but an interesting approach using the Web is presented by Nakov and Hearst (2005). We do not apply a bracketing dependency or adjacency model to our candidates, but only use the simple information that English nominal contiguous compounds tend to be right-bracketed with the head being almost always the last noun of the sequence, if any. We will call this the `head` merging technique.

This last technique for merging the candidate has a serious drawback: since the terms are simplified, lots of important information is lost. The head word here acts like a placeholder with some semantics, that is the reader can discover "about what" the original sentence was talking but not "exactly what it says". Instead of solving the problem, we would actually display to the user in the final evaluation interface that this was a simplification and what the original term in English was. Thus, the user gets a better translation for the sentence and can still retrieve the original information by consulting a specialised dictionary. The automatic creation of a specialised multi-lingual dictionary is unfortunately a complicated problem that is not in the context of this work.

For this specific corpus, we also have a gold standard terminology. Therefore, a third artificial merging technique was implemented in which we detect expressions annotated as MWTs in the original data and then concatenated their terms (`GS wws`) or replaced them by their heads (`GS head`). This configuration is used as a reference against which we compare our output. Table 3.5 shows a examples of input (source language) sentences using different merge configurations.

Table 3.5: Example of source sentence modifications using four merging techniques.

	Example inputs
Input	*ras-related gtp-binding proteins and leukocyte signal transduction*
`wws`	*ras-related#gtp-binding#proteins and leukocyte signal#transduction*
`head`	*proteins and leukocyte transduction*
`GS wws`	*ras-related#gtp-binding#proteins and leukocyte#signal#transduction*
`GS head`	*proteins and transduction*

Both techniques follow a recent trend in MT called *Textual Entailment (TE)*. This area studies algorithms to determine whether the meaning of a text or sentence t_1 is contained in the meaning of another text or sentence t_2, i.e. that t_2 entails t_1. This is particularly useful for MT because a system could try to generate a simplified version of a sentence before translating it, probably with some information loss but surely improving the quality of the translation. A good introduction to TE can be found in Quiñonero-Candela et al. (2005). Our technique falls in this category because we try to simplify the input by detecting and removing complicated nested terminology before translation. The choice of the merging technique depends on the target application. For example, `wws` could be best when the application users have some basic knowledge about the MWTs in English but cannot accept word-by-word translations for them. On the other hand, for non-expert users, less precise but completely translated information might be more useful, so that `head` is preferred.

4 EVALUATION

In this chapter, our goal is to systematically evaluate the technique described in chapter 3. Therefore, we are going to analyse each of the steps (pre-processing, candidate extraction, candidate filtering and machine translation integration) using standard tools from information retrieval and machine learning (precision, recall).

To evaluate pre-processing, we will analyse not only the extracted MWT candidates but also the gold standard. We are interested in obtaining a GS with less entries, unifying similar terms through lemmatisation, unified tokenisation and acronym detection and removal. With a cleaner GS, we expect to improve the evaluation parameters of the candidates.

Several dimensions of terminology extraction are evaluated. First, we measure the accuracy of the candidate selection procedure that uses POS tags based on their observation on *Genia-train*. Second, we evaluate the average precision of each of the Association Measures (AMs) taken individually as an association strength indicator. Third, we analyse the nature of the corpora from which we draw n-gram frequencies, namely the Genia corpus and the *Web*. Fourth, we describe the performance of several machine learning algorithms on the selected candidates using association measures as feature vectors. Fifth, we show how to improve precision by discarding all candidates that occur less than an established frequency threshold. Sixth, we compare the output of our method, i.e. the list of attested MWTs, with the list of collocations obtained by a standard method for terminology extraction, Xtract.

Finally, we suggest a methodology to evaluate the integration of automatically extracted MWTs with a statistical MT system, comparing different merging and replacement techniques and evaluating the potential benefits of MWT detection for machine translation.

4.1 Evaluating pre-processing

As pre-processing heuristics are not learnt from the corpus, we can evaluate them on the whole corpus, ignoring the split into *Genia-train* and *Genia-test*. The evaluation of pre-processing is based on three types of measures:

1. the size of the GS and of the MWT candidate list;

2. the initial precision and recall of the extracted patterns, summarised by the corresponding F-measure;

3. the average frequency of hapax legomena and of the top-100 most frequent candidates.

Two different evaluations are performed, first for uniform tokenisation and lemmatisation of plural/singular (§ 4.1.1) and second for acronym detection and removal (§ 4.1.2).

4.1.1 Re-tokenisation and lemmatisation

Re-tokenisation and lemmatisation are simple heuristics to unify inflected variants of MWT candidates and put them in a normalised form. We recall that we are only performing simple number lemmatisation by transforming regular-plural words into their corresponding singular form. In this section, we will refer to these two pre-processing steps as a single *normalisation* phase. Normalisation can be beneficial to the resulting list of MWT candidates both in terms of recall and precision. On one hand, we avoid sparseness, because two candidates c_1 and c_2 with frequencies f_1 and f_2 are considered as different instances when no pre-processing is applied. If during pre-processing we discover that, for example, c_1 is the plural of c_2, they will be unified in a canonical form with frequency $f_1 + f_2$. Hence, we expect to improve recall even though normalisation yields a smaller GS. On the other hand, when we identify that c_1 is an annotated MWT, then c_2 will also be a term (remember that our definition of MWT does not take the context into account). This improves the probability of an unannotated candidate to be considered as a MWT and thus improves precision.

Given our simplified noun lemmatisation algorithm of section 3.2.2, in a first moment we performed a manual annotation of our plural/singular transformation heuristics, evaluating according to the correctness of the singular form the list of 1,361 words in the corpus ending with “s”[1]. Results show that more than 99% of the chosen tokens were correctly transformed, with problems only for some short words (e.g. *doses* → **dos*, *mars* → **mar*) and irregular cases not taken into account by the algorithm (e.g. *matrices* → **matrice*).

Table 4.1: Evaluation of re-tokenisation and lemmatisation (norm.) on MWT candidates.

	Without norm.	**With norm.**
MWTs in GS	29,513	28,340
Candidate MWTs	66,817	63,210
Precision	15.56%	37.24%
Recall	35.22%	83.06%
F-measure	21.58%	51.42%
Top-100 mean frequency	189.98	205.38
Hapax legomena	51,169 (77%)	47,770 (76%)

Table 4.1 shows the set of evaluation measures for two different configurations: a first run where we extracted the 57 selected POS patterns without re-tokenisation and lemmatisation of the candidates (*Without norm.*) and another run, this time re-tokenising and merging singular and plural forms of the candidates (*With norm.*). We observe that the size of the gold standard decreased by 3.89%, and the list of candidates is 5.35% smaller. This shows that normalisation generates a more compact candidate list and GS, as we expected, because variants of a candidate are joined together into a single normalised form. From a linguistic perspective, this makes sense since two realisations of the same term that have different lexical forms are actually representing the same concept, and should thus be considered as a single unit. However, it is important to note that this merging is done independently of the contexts and only at lexical level (but not syntactic, semantic or pragmatic), and that we focus on MWT type identification but not on token disambiguation. That is, if a multi-word construction is a term only in some specific context, this will not be captured by our method and all occurrences of the ambiguous construction will be considered as a multi-word term even though some of them are not directly related to the concept represented by the homonym term. Nonetheless, this problem

[1] Evaluated manually by a non-native speaker, with the help of on-line resources.

should be minimised by the fact that our corpus contains only biomedical documents and that the nomenclature of a knowledge area is constructed in such a manner that term ambiguity is avoided and that syntactically they do not allow much variation. If we were dealing at the same time with two or more domains in a same corpus, this problem should become more explicit (e.g. *virus* is a term in Biomedicine and in Informatics, but both represent different concepts and should be dealt with separately).

When we look at recall and precision, results are surprisingly good. We go from an initial recall of 35.07% to a recall of more than 83%. Here, we can see the importance of a proper tokenisation on the candidates: the first list, without normalisation, contains a very high variability in terms of POS patterns for MWTs. When we apply a homogeneous tokenisation to each candidate, the number of POS patterns containing internal punctuation, dashes and slashes decreases. For example, the candidates *t-cell* and *t cell* will have the same POS pattern after normalisation and will be correctly captured by the selected rules. Without normalisation, we would probably consider *t-cell* as a single word, ignoring it as a MWT candidate. This results are unexpected not only because of the high improvement in recall but also because in the normalised version, the GS is actually smaller than the non-normalised GS, showing that we improve both the quality of the candidates and the quality of the annotation. If we consider precision, we double the number of true positives in the list, even if the list is smaller, i.e. we keep out much more false positives than in the non-normalised version. In short, normalisation makes the F-measure of the MWT candidates (before any filter) jump from 21.43% to 51.32%. The last two lines of the table show that normalisation is also effective against data sparseness: the average frequency of the top-100 most frequent candidates increased by 14, while the number of hapax legomena was reduced by almost 5%.

Here we can perceive that, unfortunately, more than two thirds of our candidates appear only once in the corpus, suggesting that statistical AMs for these candidates using only the Genia corpus would not only be unreliable but hardly of any help, in other words, useless. Nonetheless, the results depicted in table 4.1 demonstrate the importance and impact of systematic pre-processing on the data before applying any state-of-the-art MWE extraction technique directly on the raw corpus data.

These results bring significant improvement, but should also be evaluated carefully. Re-tokenisation and lemmatisation of the candidates is a heuristic process that involves language-dependent rules. Not only the rules for singular and plural vary from language to language, but also tokenisation may depend on the conventions of the domain. Removing slashes and dashes might also remove some useful information about words that already share some semantics in the text. Therefore, to avoid the potential loss of important information, in a real-world system these deleted symbols should be kept along with the candidate through the whole processing pipeline and then re-integrated to the final results after MWT detection ends. For some languages, plurals might not be formed using a simple set of rules on the word termination like in English. For instance, in German most plurals are formed by adding a special suffix (*-er*, *-en*, *-n*, *-e*, *-s*) to the end of the word, but plural may also be invariant and the preceding vowel may or may not be *umlaut*ed. This difficulty might be even more evident for languages that have more than two plural forms, including trial and paucal (few objects) plural. Therefore, it is important to compare the cost of implementing such domain- and language-dependent heuristics with the potential improvement in the quality of the extracted MWT candidates. For resource-rich languages, this would not constitute a problem due to the availability of off-the-shelve lemmatisers and tokenisers, as well as relatively good POS taggers. Other kinds of normalisation, such as unifying the POS tags of gerunds and nouns or consider adjectives and nouns number and gender inflection are also good candidates for pre-processing heuristics in English and in other languages.

4.1.2 Acronym detection evaluation

Before extracting MWT candidates, we remove all expressions inside parentheses that look like

Table 4.2: Evaluation of Acronym Removal (AR) on MWT candidates.

	Without AR	**With AR**
MWTs in GS	28,601	28,340
Candidate MWTs	61,401	63,210
Precision	37.53%	37.24%
Recall	80.58%	83.06%
F-measure	51.20%	51.42%
Top-100 mean frequency	205.05	205.38
Hapax legomena	46,330 (75%)	47,770 (76%)

an acronym. Using this simple procedure, we were able to detect and remove more than 1,300 parenthesised acronyms in the corpus, and they are potentially good indicators of terms. To evaluate the impact of Acronym Removal (AR) step, we proceed like in the previous section, i.e. we analyse the GS and the candidates list in two configurations: without AR and with AR. Table 4.2 shows that, on one hand, the GS size decreased with AR, because around 300 terms that were considered as different instances before (because of the occurrence of internal parentheses) are now joined in a single entry. On the other hand, as some acronyms occur with parentheses usually after (and rarely in the middle of) the terms, around 1,000 acronyms detected in the text are not useful to diminish the number of entries in the GS. When we look at the MWT candidates, the list increases by almost 3% because our POS patterns do not contain punctuation signs like parentheses, and therefore are more effective on the acronym-free version of the corpus. Concerning the quality of the extracted patterns, we notice that even if we gained more than 2.5% in recall, precision dropped slightly, resulting in an insignificant increase in the F-measure of the candidate list with AR. We could argue that in candidate extraction, unlike in in candidate filtering, recall is more important than precision, since the candidates are going to be filtered in further processing. However, we must be careful about such analysis since the small difference in recall does not justify the cost of tuning acronym removal for specific languages or domains.

Another aspect worth evaluating concerns the frequency boost achieved with AR for the marginal probabilities of the candidates in the Genia corpus. Since we have a list of attested acronyms as a by-product of the acronym removal step every time an acronym occurred, the frequencies of the words in the expanded version of the acronym were incremented. Analysing the vocabulary of the corpus, we have an average word frequency of 37 occurrences, while the top-100 most frequent words have an average of 2,788 occurrences per word. With frequency extension, these values increase to respectively 47 and 3,566 occurrences in average. In the last two lines of table 4.2 we can see that the frequencies of the top-100 most frequent candidates remain essentially unchanged, but that there is a significant difference for rare candidates occurring only once in the corpus. As a consequence of the AR step, we increase the number of hapax legomena by 1% (4,440). This shows that creating acronym correspondence lists and using them to boost frequencies might be a good option in domain-specific texts where sparseness is a serious problem, but will also generate more candidates and therefore, more rare events.

4.2 Terminology extraction evaluation

The goal of this section is to evaluate several factors involved in MWT extraction using the technique proposed here. Until now, we evaluated pre-processing on the whole corpus, but now we are going to evaluate models generated *from* the corpus. Hence, all the evaluation results presented in this section were performed on a held-out portion of the corpus, *Genia-test*, which was not used in the creation of the models. Recall that *Genia-test* was separated from the remainder of the corpus after pre-processing. It contains the last 100 abstracts of the corpus, i.e. 895 sentences with 23,939 tokens. During the experiments, the annotation contained in *Genia-test* was not used, to avoid any bias in our evaluation.

4.2.1 POS pattern selection

The patterns of MWT candidates used in this work were based on those suggested by Justeson and Katz (1995) (J&K). However, even though the J&K patterns are generic enough to cover a large number of terminological constructions, they do not take into account the fact that biomedical terms are considerably longer than in other areas and that they often contain non-standard tokens such as numbers. Therefore, the use of their patterns would yield a maximum recall of 66.71%, i.e. we would be ignoring one third of true positive candidates with a precision of 29.58% on *Genia-train*, as we can see in the comparative table 4.3. Please note that these values for recall and precision do not correspond to the output of our method, but to the initial proportion of true positives in the candidates list. That is, recall corresponds to the number of MWTs in the candidates list over the size of the GS, while precision represents the proportion of true MWTs with respect to the size of the candidates list, according to equations 3.1 and 3.2. In this table, we can also verify that selecting all the patterns that occur more than 10 times in the training corpus ($f > 10$) drastically improves recall at the price of cutting precision by half. Then, in order to, improve precision, we decided to select only patterns that appear more than 10 times in the training corpus *and* for which the proportion of true positives is greater than 30% ($f > 10, p > .3$), in a total of 57 patterns, eliminating recurrent noisy patterns that retrieve much more random word combinations than MWT candidates.

Table 4.3: Precision and recall of three POS pattern sets on *Genia-train*. These values were used in POS pattern selection.

POS patterns	**Precision**	**Recall**
J&K	**29.58%**	66.71%
$f > 10$	16.50%	**92.57%**
$f > 10, p > .3$	31.52%	89.43%

A small sample of the part of speech patterns selected for candidate extraction is presented in table 4.4. For more details, a complete version of this table is included in appendix C. The table shows that the top-5 patterns correspond to J&K. Then, we list combinations of nouns and adjectives of different lengths. Special cases are the patterns `N-NUM`, containing a number, and `A-CC-A-N`, that contains a conjunction. Since we do not provide sophisticated treatment for conjunctions, such patterns must be extracted in their rough form. For instance, the candidate *young and aged subject* would ideally be considered as two candidates: *young subject* and *aged subject*. Therefore, integrating also a syntax-based approach such as the one proposed by Seretan (2008) would lead to better performance on this type of candidate. It is interesting that the pattern `N-P-N` suggested by Justeson and Katz, i.e. two nouns assembled by a preposition (e.g. *degrees of freedom*), is absent from the table. Even if, from a linguistic point of view, they potentially capture good MWT candidates, as they correspond to the passive form of productive noun compounds and nominalisations, this corpus does not seem to contain enough of such candidates for producing relevant results.

For comparison purposes, we show the precision and recall values from *Genia-train* in first three columns of table 4.4. However, as this corpus was used during POS pattern selection these values do

Table 4.4: Precision and recall of the most frequent patterns in *Genia-train* and *Genia-test*.

Pattern p	$P_{train}(p)$	$R_{train}(p)$	$\sum R_{train}(p)$	$P_{test}(p)$
N-N	42%	22%	22%	65%
A-N	31%	19%	41%	77%
N-N-N	43%	10%	51%	74%
A-N-N	40%	9%	60%	78%
A-A-N	32%	4%	64%	88%
A-N-N-N	38%	3%	67%	83%
N-N-N-N	39%	3%	70%	85%
N-A-N	36%	2%	72%	87%
A-A-N-N	39%	2%	74%	89%
N-A-N-N	38%	1%	74%	95%
N-NUM	32%	1%	75%	81%
A-N-N-N-N	41%	1%	76%	95%
A-CC-A-N	30%	1%	77%	93%
N-N-N-N-N	36%	1%	77%	90%
N-N-A-N	35%	1%	78%	93%

not correspond to a realistic evaluation of the selected POS patterns. In the last column the precision of the patterns for *Genia-test* is very high, between 65% and 95%. Since some of the candidates are not in the GS of the test set but belong to the GS of the training part (and are therefore true candidates), the recall obtained is of 200%. This pathological result is a consequence of our fully automatic evaluation process, so that the number of positives candidates with respect to the GS is much greater than the number of manually-annotated terms in *Genia-test*. Manual annotation would probably show that the selected patterns identify most of the MWTs in the corpus while a large portion of the negative candidates are actually context-dependent terms, i.e. they are in the GS because they might be terms in some contexts but thy might also remain unannotated when they do not constitute a term.

4.2.2 Association measures evaluation

Each Association Measure (AM) provides some information about the association strength between the words of the candidate. Our goal is not to choose "the" best AM, since a machine learning algorithm is going to combine them all to generate the final list of extracted MWTs. It is nonetheless important to understand what each measure tells us about the nature of our MWT candidates. This allows us both to evaluate (a) how informative/useful is each of the four measures and (b) how good are the frequencies taken from each corpora for use with AMs

Figures 4.1, 4.2 and table 4.5 are used to perform this evaluation. They are actually two forms of visualising the same information, since average precision in table 4.5 corresponds to the area below the curves of figures 4.1 and 4.2, because at each recall level (or each true positive candidate) we calculate the precision and then we average all the precisions, exactly as we would do if we calculated the area under a curve. The dotted line in both graphics corresponds to the "baseline" precision, i.e. the proportion of true positives in the candidates list. Note that we are using the term "baseline" to refer to the proportion of true positives in the test data (of 24.16%) and not to the results of another system. This definition of baseline will be used throughout the next sections. All the measures will eventually converge to this line since, when we achieve 100% recall, we will have ran through the entire list and will have a precision corresponding to the initial proportion of true positives.

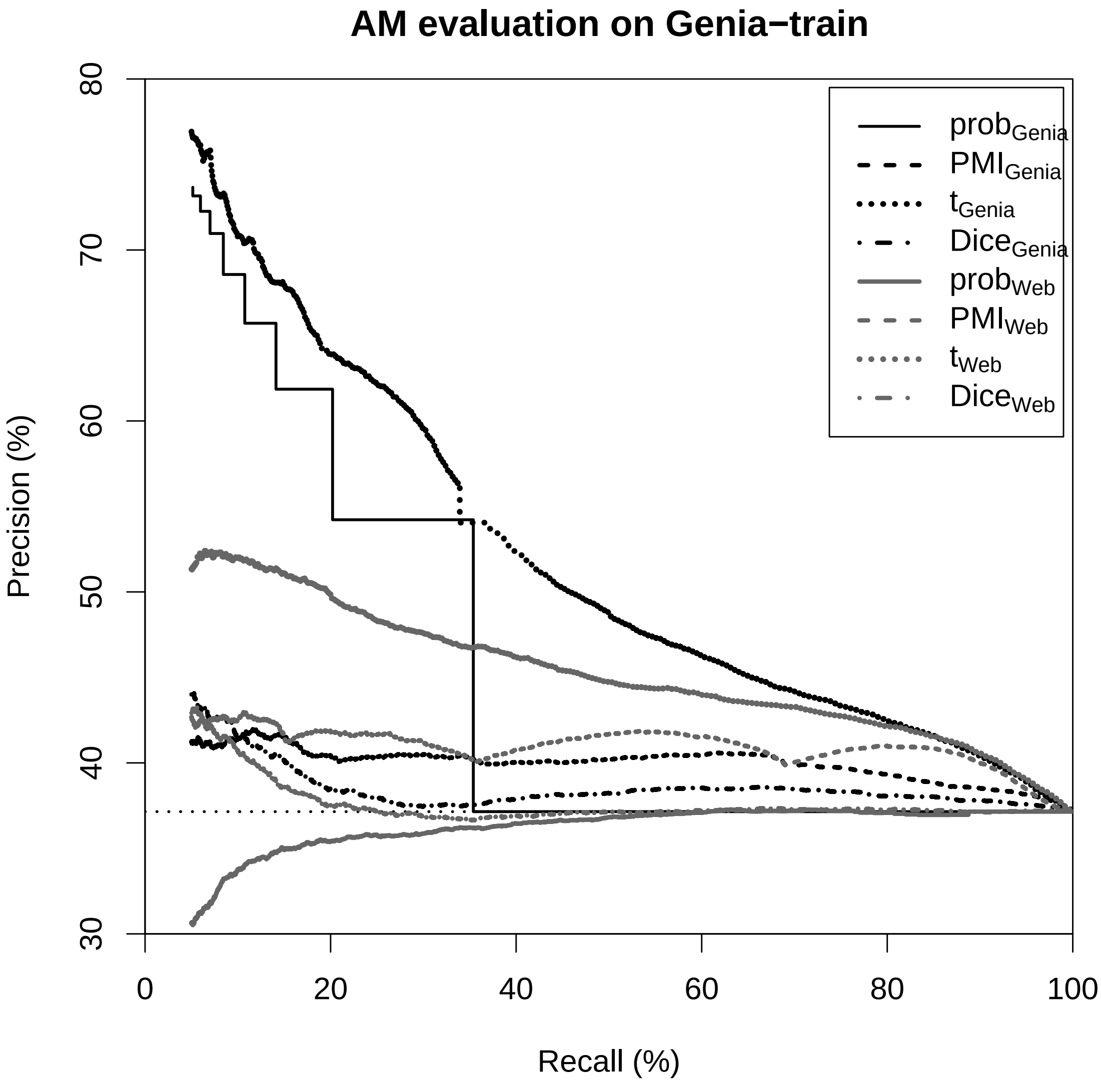


Figure 4.1: Precision vs. recall in *Genia-train*. The horizontal dotted line represents baseline precision.

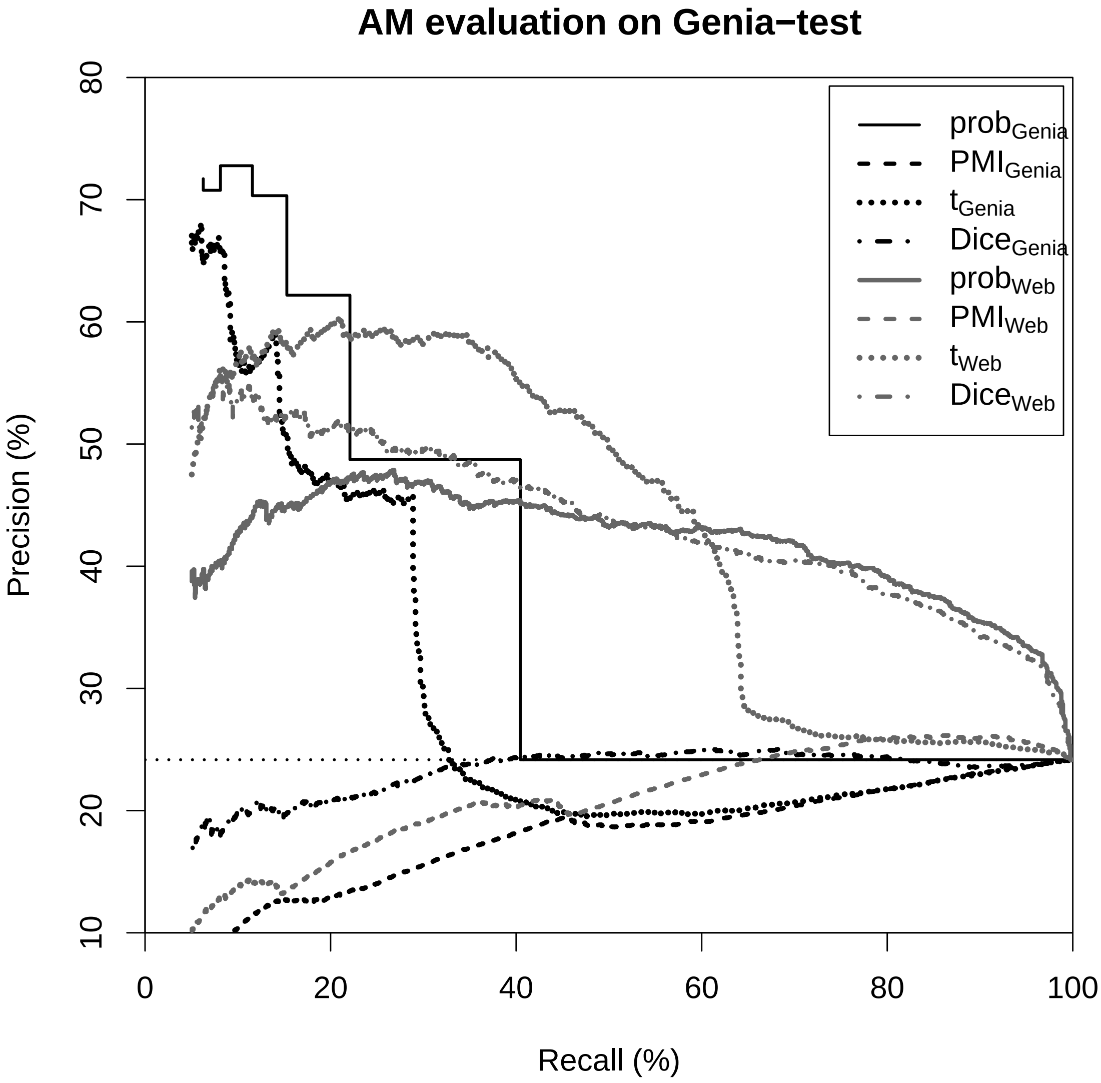


Figure 4.2: Precision vs. recall in *Genia-test*. The horizontal dotted line represents baseline precision.

The graphics show that the association measures calculated from *Genia-test* are much more noisy than those calculated on the training corpus, because it is considerably smaller. Therefore, we will focus our analysis on figure 4.1 and on the first column of table 4.5[2]. First of all, some of the measures have very similar performances. For instance, the first two curves correspond to the t measure both in the Genia corpus and in the *Web*, confirming that the t-test is very effective for ranking collocational units[3], as discussed in related work in MWE extraction (Evert and Krenn 2005, Manning and Schütze 1999). In the test split, however, for recall levels greater than 30%, the precision of the t_{Genia} measure drops drastically, perhaps because at this level frequencies become too sparse for the test to be conclusive. This phenomenon is not observed in other AMs, though.

The second conclusion that we can draw from these graphics concerns the rectangular-shaped curve corresponding to the raw probability of the candidate in the Genia corpus. In both slices, this measure achieves precision significantly above the baseline, telling us that probably we should filter out low-frequency candidates, establishing a threshold for the `prob` measure (please refer to section 4.2.5 for more details).

Pointwise Mutual Information performs relatively bad in both slices and corpora, lying below the baseline for *Genia-test* and slightly above for the training data. This is a surprising result, as MWTs are related to collocations and `PMI` is believed to be a good indicator of collocational behaviour (Smadja 1993). However, these results are compatible with recent AM evaluations. Evert and Krenn (2005), for example, reported that the t-test and log-likelihood measures are the best for their data sets, and that `PMI` is not telling us much about association strength of multi-word candidates. Since log-likelihood is based on contingency tables, and we cannot compute these for the *Web*, we are not able to say whether it would be appropriate for biomedical terminology.

Finally, in terms of *Web*-based AMs, two of the measures have inconsistent behaviour: $prob_{Web}$ and $Dice_{Web}$ perform well on the test data but not on the training data. Moreover, the *Web*-based measures tend to have better results on the test split: except for `PMI`, all the AMs are largely above the baseline (41%, 43% and 44% against 24% baseline). In section 4.2.3, we evaluate the differences between both corpora, but this does not completely explain the differences encountered here and more investigation is needed to determine possible causes for the results. Except for `PMI`, the evaluation of *Web*-based AMs is, at this point, inconclusive.

A pertinent question one could ask would be: "But if the raw probability gives such good precision, why then calculate the other measures?". To try to answer this question, we performed simple experiments using several different ML algorithms only on two attributes –Genia frequency and *Web* frequency. However, none of the resulting models are able to distinguish positive and negative candidates. In this case, the models tend to guess always the same class for a candidate, giving a very large recall for one class usually the one with the majority of the candidates but null recall to the other. Moreover, a model that learnt information from all AMs will perform better than a single AM, confirming that although `prob` is a quite good estimator, the other AMs add complementary information to the MWT detection model.

As a last experiment on AMs, we executed a simple and automatic best-first attribute selection algorithm on the training set. This data analysis tool is part of the WEKA package, and was ran over the *Genia-train* portion of the corpus. The output tells us that the features that have greater contribution in the model are $prob_{Genia}$, t_{Genia} and t_{Web}. These results confirm our discussion and validate our evaluation of association measures.

[2] We note that as there is no learning involved in the collecting of frequencies.

[3] This is the case in spite of the false assumption that words are distributed normally.

Table 4.5: Average precision of the AMs on the MWT candidates for *Genia-train* and *Genia-test*.

AM	*Genia-train*	*Genia-test*
$prob_{Genia}$	46.16%	38.83%
PMI_{Genia}	40.08%	17.51%
t_{Genia}	52.90%	31.36%
$Dice_{Genia}$	38.84%	22.78%
$prob_{Web}$	35.77%	41.49%
PMI_{Web}	41.02%	20.52%
t_{Web}	45.13%	43.21%
$Dice_{Web}$	37.85%	43.74%
Baseline	37.15%	24.16%

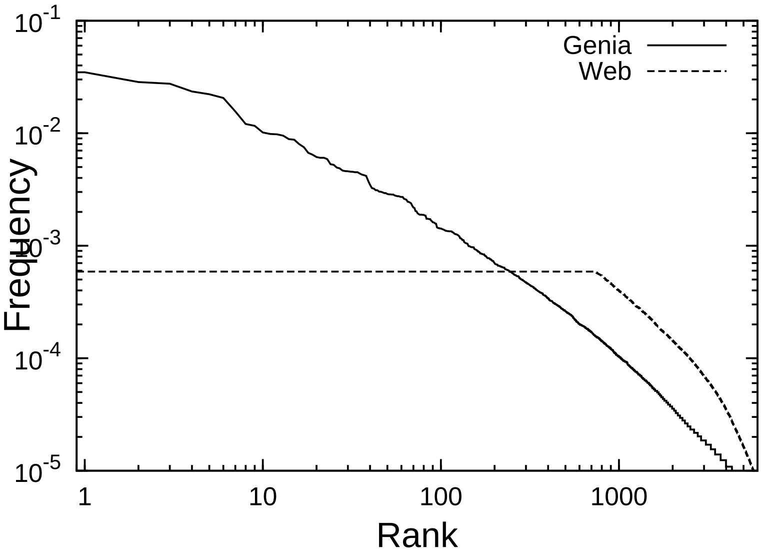


Figure 4.3: Rank plot of the Genia vocabulary for corpus frequencies in *Genia-train* and in the *Web*.

4.2.3 Corpus nature evaluation

When we evaluated association measures, we observed that they differ from one corpus to another. The goal in this section is to identify why they differ and what the potential strengths and weaknesses of each corpus are, looking at the Genia corpus and the Web through *Yahoo!*'s Web search API. Figure 4.3 shows a rank plot of the vocabulary of the corpora, in which the horizontal axis corresponds to the rank of each word and the vertical axis to its frequency using a log-log scale to observe the whole data at once.

There is a flagrant difference between the curves depicted in figure 4.3: while the Genia corpus follows a well-defined power law, the *Web* corpus seems somehow truncated for the ranks between 1 and 1,000 (most frequent candidates). For the former, this Zipfian-like distribution is also found in several language-related works, where the shape of distributions follow a power law, which can be approximated in more or less detail by some function (Manning and Schütze 1999, Newman 2005, Baayen 2001). For the *Web* corpus, on closer inspection of the data, we discover that actually, *Yahoo!*'s API returns a precise value of 2,147,483,647 occurrences for all 735 high-frequency words in the vocabulary, corresponding to 5.4% of the corpus (the most visible part of the curve). Thus, it becomes impossible to tell whether a word as frequent as *the*, that occurs 22,446 times in the Genia corpus, is different from the words *department* or *green*, that appear only once in Genia, because all of them occur precisely 2,147,483,647 times in the *Web* according to *Yahoo!*. This magic number seems to be a simplification done by the search engine to return approximately the number of indexed pages for all queries that contain high-frequency words. But even in this case, except if all the websites in the world decided simultaneously to save the planet, would all the pages in the *Web* contain the word *green* at least once? Therefore, the widely employed assumption that the number of occurrences of a word can be approximated to the number of *Web* pages that contain it might not be a precise heuristic for high-frequency words.

We also calculated the correlation between the vocabulary ranks of the two corpora. Pearson's correlation for the two corpora is 0.12 while Kendall's τ test (more accurate for rank correlation) is 0.21 and these still reflect the impact of the flat curve for *Web*-based data. Both confirm that there is a huge difference between the corpora. These results disagree with what was shown by Zhang et al. (2006), that traditional and *Web* corpora may be used interchangeably. While their evaluation was based on n-gram frequencies, our evaluation concentrates on the 1-gram vocabulary of the language, from which we calculate marginal probabilities. Underestimated marginal probabilities will yield imprecise AM values and could be one of the reasons for the performance of *Web*-based methods (§ 4.2.2). There are several possible solutions for this problem, and there are probably very adequate statistical models to approximate the frequency of top-ranked candidates based on their frequencies in two different corpora. One could, for example, calculate whether the frequencies fall in the same rank interval and, if not, approximate the less reliable frequency (*Web*) by a proportion calculated from the rank of the word in a more reliable corpus (Genia). Such kind of adjustment combines the best of both approaches: traditional corpora are good for high frequencies but very sparse whereas the *Web* can contribute with more reliable frequencies for rare words, as it does not suffer as much from sparseness, but is not a good source for high-frequency words.

4.2.4 Learning algorithms evaluation

Given that different AMs capture slightly distinct aspects of the data, instead of selecting or averaging the value of AMs for each candidate, which would mean that we need to decide which AMs are better and which are worse in general, we decided to combine them by learning a model from the training data. This model was evaluated on a *Genia-test*, which contains 895 sentences (24K tokens). We compared four families of ML algorithms; the results are shown in table 4.6 and alternatively in figure 4.4. We used the precision and recall on the `MWT` class as a performance measure because, for all learning algorithms, the F-measure of the class `NonMWT` lies between 80% and 100%. In other words, in terms of MWTs, these data sets are unbalanced, as they contain a much larger proportion of `NonMWT`s than of `MWT`s, and the learning algorithms obtain a better precision guessing that a candidate is not a term. For more details on the particularities of each ML algorithm, please refer to Mitchell (1997).

First, we used a classical decision tree algorithm (J48). The most serious drawback of this algorithm (and others, as we will see) is that it tends to give equal importance to both classes, `MWT` and `NonMWT`. Assuming that each is equally important, it classifies correctly 75% of the test instances, with an F-measure of 85.2% for the `NonMWT` class. However, the algorithm is not able to prioritise performance on the `MWT` class, which is the most relevant for this work, obtaining a very low recall for it. That is, it suffers from the low-recall problem that we would like to avoid. Moreover, this is an intrinsically numeric task, probably with complex decision borders and therefore not fully suitable for decision trees, as these are designed to deal with finite-value attributes and linearly separable problems. On the other hand, decision trees are easily interpretable and give an indication of the complexity of the problem and contribution of the features: the generated tree contains 484 nodes with most of the leaves filtering on the attribute `POSPattern`. Closer to the root of the tree are the attributes t_{Genia} and t_{Yahoo}, indicating that they might be good MWT indicators, as suggested in the previous section.

Bayesian net learning is specially fit for numeric data. Because Bayesian methods are based on conditional probability, they could provide us with a good model for MWT candidate filtering. We tested the Bayesian net algorithm[4] using two different heuristics to search the net for the optimal configuration: spanning tree (TAN) and Simulated Annealing (SA). Even though TAN presents slightly better F-measure, both present comparable results in terms of accuracy. The precision of SA is quite high if compared with other algorithms, while the recall of both algorithms still presents the same problem as with decision trees: Bayesian net with SA classifies 76% of the test instances correctly because it has an F-measure of 86% on the `NonMWT` class. However, for the purposes of this work, the main interest is in classifying positive instances correctly.

Neural networks build implicit data models based on artificial neurons (in our case, perceptrons). We used a Multi-Layer Perceptron (MLP) with 33 neurons in the hidden layer (chosen automatically by WEKA) and a voted perceptron. Results here with two models (MLP and voted perceptron) are very contrasting: while the performance of the MLP is poor, the voted perceptron performs surprisingly well, especially on recall. The problem of class weights becomes more obvious for MLP: while class `NonMWT` has F-measure of 86%, positive instances have F-measure of 1% due to very low recall. On the other hand, the voted perceptron makes several mistakes on the `NonMWT` class (73% F-measure), and has an overall performance of 64% of correctly classified test instances. However, the overall recall of 63.8% is the best of the ML results. There are several hypotheses for the low performance of the MLP. First, the automatically chosen parameters might make the learning algorithm to oscillate so that it never converges to an optimal solution. Second, it is possible that the number of neurons in the hidden layer cannot model the problem correctly. Indeed, additional experiments showed that a MLP with as much as 100 hidden neurons has an F-measure of 13.2%. However, even larger values could be necessary, implying very high computational needs. Finally, it is also possible that the global error-minimisation procedure is not capable of handling highly unbalanced data sets. Further investigation is needed to determine why is the voted perceptron algorithm so efficient on this particular problem whereas a more sophisticated learning algorithm such as MLP performs bad.

The last class of algorithms that we used is Support Vector Machines (SVM), which have received increasing attention in the ML community in the last years. Two configurations are tested: polynomial and radial kernels. These parameters change the form of the class border surfaces and will therefore determine how close the model is to the data. Results are interesting both in terms of precision and recall, even though each of the configurations tends to give more importance to either precision or recall. Polynomial kernels have a very good recall (comparable to voted perceptron) whereas radial kernels offer the highest precision, 52.6%. There are two important considerations about SVM: first, they tend to overfit on very difficult problems, creating models with almost as much support vectors as the number of instances in the training set. In the next session, we investigate this hypothesis. Second, even if polynomial kernel seems to perform better than radial kernel, we are going to use radial kernels in the next evaluation steps because, when compared to other MWT extraction techniques, for the purposes o this work it is more important to prioritise precision.

The evaluation of ML algorithms also allows us to estimate the importance of Web-based AMs. To do this, we performed an additional test by re-building an re-evaluating the models for the algorithms in table 4.6, but this time using only 5 features: the POS pattern and the 4 AMs calculated on Genia, i.e. this new models ignore all information provided by the *Web*. The overall results show that the F-measure drops for all algorithms. For example, J48 has a precision of 47.9% and a recall of 19.7% (F-measure = 27.9%), but without Web statistics these values drop to 46.9% precision, 16.4% recall

[4] Bayesian net differs from naive Bayes because the former trains a whole net while the latter models each variable independently.

Table 4.6: Precision, recall and F-measure of machine learning algorithms on MWT class (accuracy) for *Genia-test* set.

Learning Algorithm	**Precision**	**Recall**	**F-measure**
J48	47.9%	19.7%	27.9%
Bayes (TAN Search)	48.6%	32.1%	38.6%
Bayes (SA Search)	52.4%	28.9%	37.2%
MLP	27.3%	**0.5%**	1.0%
Voted Perceptron	36.3%	**63.8%**	46.3%
SVM (polynomial kernel)	34.9%	60.1%	44.1%
SVM (radial kernel)	**52.6%**	**35.4%**	42.3%

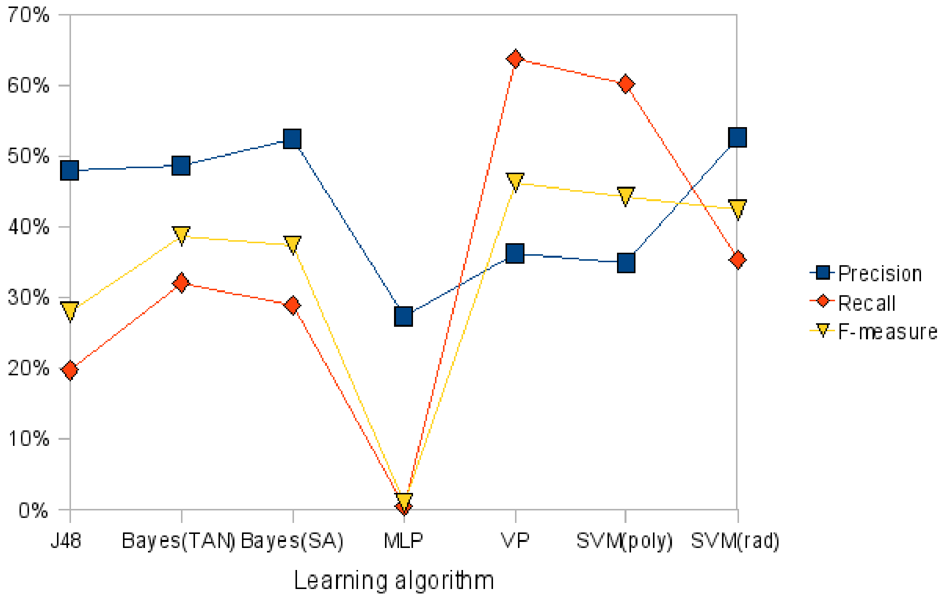


Figure 4.4: Graphical representation of the accuracy of machine learning algorithms on the MWT class for the *Genia-test* set.

and 24.3% F-measure. The same is true for Bayesian net algorithms, where the F-measure drops from 38.6% and 37.2% to 29% and 28.2% respectively for TAN and SA search strategies. Finally, if we evaluate the importance of Web-based AMs with respect to a radial SVM, we have a drop from 52.6% to 42.7% in precision and from 35.4% to 14.6% in recall. This shows that the use of the Web as a general-purpose corpus is not only a theoretical tool against sparseness but also an effective way to improve the quality of acquired MWTs.

For a more detailed analysis of the behaviour of the radial-kernel SVM classifier table 4.7 shows the resulting confusion matrix. In terms of lexicography, these results indicate that from a list of 4,685 candidates, the classifier will return 763 candidates from which a human lexicographer or grammar engineer would need to filter out 362 negative instances. In comparison, if no model was applied to the candidates, the work would be considerably more costly since it would involve the manual analysis of the whole list in which precision is not 53% but only 24%. On the other hand, with this model 731 true positives would be incorrectly classified as instances of the `NonMWT` class and consequently discarded. While this may seem too high a cost, when compared to other MWT extraction techniques the performance is actually very good (see section 4.2.6). It is also important that these results are taken in context. Although they are lower than the performance obtained in other NLP tasks such as parsing and POS tagging, given the intrinsic complexity of the problem and the current state-of-the-art, the performance of our model is above a traditional terminology extraction technique, as we will see in the next sections.

Table 4.7: Confusion matrix for SVM (radial kernel) classifier. On a test set with 4,685 MWT candidates, 3,592 are correctly classified (main diagonal).

	Classified as:	
	NonMWT	MWT
NonMWT	**3191**	362
MWT	731	**401**

4.2.5 Threshold evaluation

A common technique that helps to improve the precision of MWE extraction techniques concerns the definition of a straight frequency threshold under which the candidates are discarded for precision improvement at the expense of a drastic recall drop. Remember, for example, that at least 75% of our candidates appear only once, implying that a frequency threshold as low as 1 would already exclude more than three fourths of our candidates. In this experiment, we tried two different values for the frequency threshold: 1 and 5. Table 4.8 compares three configurations: the first corresponds to the previous experiments in which no threshold is applied to the candidates. The second configuration is the one in which we exclude all hapax legomena from the candidates list. The last configuration only keeps candidates that appear more than 5 times in the Genia corpus.

Our evaluation is based on the comparison between the "baseline" and the final precision and recall. While precision is measured as usual, considering the size of the candidate list, recall is calculated considering that the whole GS of *Genia-test* contains 2,009 MWTs. Therefore, the recall of the method does not consider the proportion of true positives with respect to true candidates in the truncated list but we also take into account the discarded candidates, simulating a classifier that says that all candidates occurring less than a threshold are of the class NonMWE. This is not only more realistic than simply considering as true positives the candidates in the list, but also eases comparison with other MWT extraction methods.

We can see in 4.8 that there is no optimal high-precision and high-recall solution: precision improves when recall decreases and vice-versa. When we apply no frequency threshold, the proportion of true positives found by the classifier in comparison with the baseline grows from 24% to 53% but recall falls below 20%. As we increase the threshold to 1, precision slightly improves, and the recall drop caused by the classifier is also minimised: an initial proportion of 23% true positives becomes a final proportion of 21%, while we improved precision by more than 8%. One could think that an even greater threshold would yield even better results, but the last column shows that this is not true. With an initial proportion of 73% true positives, the SVM classifier performs slightly better providing 74% of precision, but at the price of extremely low recall, around 6%. In this last configuration, our training set contains around 3.5K instances and our whole test set is as small as 180 instances, from which 131 are true positives. This shows the huge contrast between not using thresholds at all (60.5K training instances, 4.6K test instances) and setting a high threshold for MWT candidates occurrence.

It is impossible for the moment to draw precise conclusions about the best threshold, since the recall is highly dependent of the final application. While on one hand, for building a dictionary, it is important to have high precision and to minimise manual effort, on the other hand a MT system might want to cover a wider range of MWTs even if some of them are noisy. Application-oriented evaluation of the proposed technique will be discussed in next section and is based on the results of threshold evaluation. In any case, discarding only hapax legomena will already eliminate considerable noise and improve both precision and recall.

Table 4.8: Comparative evaluation of radial-kernel SVM on *Genia-test* with (a) no frequency threshold, (b) candidates excluding hapax legomena and (c) candidates occurring more than five times.

	No threshold	$f_{Genia} > 1$	$f_{Genia} > 5$
# instances (train)	60585	14688	3528
# instances (test)	4685	940	180
# MWT (test)	1132	458	131
Baseline precision	24.16%	48.72%	72.78%
Baseline recall	56.35%	22.80%	6.52%
SVM Precision	52.56%	56.83%	74.14%
SVM Recall	19.96%	20.91%	6.42%

4.2.6 Compare with related work

In order to contextualise our results, we also compare our technique with standard terminology extraction techniques. Since most related work deals with MWEs in general and do not focus on specialised terminology, it is difficult to compare them with our method. On the other hand, commercial tools do not perform truly automatic terminology discovery, generally using some pattern-matching algorithm to compare a manually compiled taxonomy, ontology or terminology, with the provided input text. Moreover, most of them are not available under any form and must be manually re-implemented, and this is outside the scope of this work. Therefore, we compared our technique with the available terminology extraction algorithm Xtract (Smadja 1993). Xtract uses bigram mutual information to discover collocational patterns and then extends the MWE arbitrary n-grams. We used the implementation of Xtract integrated with the Dragon Toolkit for NLP and IR (Zhou et al. 2007). This version of Xtract uses POS tags to filter the extracted terms and works on the lemmatised lexical units. We ran Xtract on the pre-processed unannotated version of *Genia-train* and *Genia-test* and compare with the former, for a fair comparison, since the latter is much smaller than the former and Xtract cannot use the available information on the training part as our method does.

Table 4.9 shows precision and recall for four different configurations: Xtract on training and test parts and our technique with two different frequency thresholds. From the whole training corpus, Xtract could only identify 1,558 MWTs, from which 1,041 are true positives. This yields a recall of less than 4% while precision is of almost 67%. When we run Xtract on the test data, results are even worse on recall: only 2.44% of the MWTs were retrieved, even if they present 70% precision. With our technique, we are able to obtain recall that is ten times larger than Xtract with only around 10% performance drop. In addition, if we used a strict threshold we could obtain a low recall that is still twice as high as the recall of Xtract on the training corpus whereas our precision is superior to Xtract on any slice of the corpus. While it would be possible to fine tune Xtract and improve candidate annotation to potentially obtain a better performance (the one described in the literature is of around 80% precision), this is also true of our method. However, unlike for the latter, it may be difficult to improve recall since Xtract does not use external sources like the *Web* to overcome sparseness and has internal thresholds that discard noisy candidates if they do not occur enough times in the input corpus.

Table 4.9: Comparative performances of (a) Xtract on *Genia-train*, (b) Xtract on *Genia-test*, (c) our technique with threshold = 1 and (d) with threshold = 5.

	Xtract (train)	**Xtract (test)**	$f_{Genia} > 1$	$f_{Genia} > 5$
Extracted candidates	1,558	70	739	174
True Positives	1,041	49	420	129
Recall	3.84%	2.44%	20.91%	6.42%
Precision	66.81%	70%	56.83%	74.14%
F-measure	7.26%	3.62%	30.57%	11.82%

4.3 Application-oriented evaluation

The goal of application-oriented evaluation is to discover whether MWT information might improve Machine Translation (MT) quality. Therefore, we presented several heuristics to include terms in the system input through pre-processing and then treat them after translation, as an additional post-processing step. The system used for translation is Moses, a Statistical MT decoder. Current SMT technology works on flat structures and we believe that our MWTs are best fit to such approach. However, even if training Moses on the Europarl corpus is considered state-of-the-art for statistical translation, this is a very costly task, both in terms of processing time and disk space ,which, added to time constraints, prevented it from being pursued here.[5] The alternative of manual evaluation of the results was also prevented due to the unavailability of native speaker annotators.

Despite of these difficulties, we describe here some general ideas for MT integration evaluation. First, it is important to determine which integration method (`wws` or `head`) is most efficient to increase the general readability of the translation. We expect `wws` have higher adequacy because with the `head` replacement we are loosing information and linguistic accuracy. On the other hand, it is necessary to determine with which method Moses can generate more natural or grammatical sentences. To evaluate the quality of the translation, one possibility is to ask human annotators to rank the translations obtained with the four configurations described in chapter 3 plus a configuration in which no pre- or post-processing are performed. Each evaluator would annotate a set of 50 sentences of the *Genia-test* corpus with 250 translations into Portuguese. More languages could be added to the evaluation when more language pairs become available for Moses (but, as we explained before, this is conditioned to available hardware and processing time).

Second, our hypothesis is that the MWT information could be helpful during post-edition. In order to evaluate that one could implement or extend an existing translation interface (e.g. SECTra_w) to show to the user the pertinent information concerning the detected terminological unit. Evaluation at this step is related to human-machine interfaces evaluation methodology, and should therefore consider dependent variables like translation time, user comfort and translation quality. Not only could this improve the quality of post-edition (through faster and more accurate translations) but also, the implicit feedback could be used to build multi-lingual terminology resources with the help of professional and volunteer translators.

Since the work reported here is part of a larger research project, we intent to execute the experiments described here as future work, and be able to provide further results to answer our questions concerning the interaction of automatically identified multi-word terminology and automatic translation of specialised text.

[5] One training step, for example, might generate as much as 12Gb data at once.

5 CONCLUSIONS AND FUTURE WORK

In the work presented here, we performed a systematic evaluation of multi-word terminology extraction on a biomedical corpus. First, we discussed some basic concepts such as the notions of multi-word expression, term and multi-word term. Multi-word expressions are very frequent in human language and correspond to combinations of two or more words that share some syntactic or semantic properties that cannot be totally inferred from the properties of the component words. Terms, on the other hand, are the objects of study of Terminology and correspond to lexical units with unambiguous meanings that relate them to concepts of a specific knowledge area. Terms might be composed of one or more words, and in the latter case we defined to call them by convention Multi-Word Terms (MWT). We also showed some common points between the theoretical Terminology and a Computational Linguistic approaches, showing which premises and hypotheses apply to our work.

Related work, be it on terminology or multi-word expressions extraction, tends to focus on a specific type of construction and language. Hence, we analysed some related research that deals with general-purpose collocations, compound nouns and phrasal verbs, as well as some studies that compare and evaluate extraction techniques. There are several tasks related to multi-word units treatment: identification in text, selection and ranking according to their quality, discovery of syntactic and semantic properties (idiomaticity, selectional preferences, translations), and use in NLP applications. Most of the related work focuses on a single type of construction and a single task, and n-gram approaches rarely work with arbitrary *n*, usually working on bigrams. There is little work on the applications of such techniques on terminology discovery and domain-specific corpora. From this perspective, we believe that the work presented here is quite new and can help in filling the gap between general-purpose and domain-specific NLP methods for multi-word units, thus becoming a tool in domain adaptation of language systems and resources.

The Genia corpus is a collection of biomedical texts annotated for terms. We used the corpus to evaluate MWT extraction techniques and parameters. Terminology is very present in the corpus, and terms tend to be very long, nested, and to contain internal acronyms and wrong POS tags and tokenisation. To ameliorate the quality of the input, we performed three simple pre-processing steps: uniform tokenisation, plural removal (simplified lemmatisation) and acronym removal. The evaluation showed that the first two pre-processing steps can significantly improve the quality of the candidates, being a powerful tool against sparseness, increasing initial POS patterns coverage and quality. The last step, acronym removal, helps to improve recall but does not change the quality of the candidates. Even though it produces slightly better F-measures, it does not seem to be essential in the processing pipeline.

From the raw corpus, we extracted a list of MWT candidates based on flat POS patterns. We believe that better results could be obtained with a more sophisticated candidate-extraction technique, for instance a syntax-based method such as the technique introduced by Seretan (2008) for general collocations or the technique of Baldwin (2005) for verb-particle constructions. However, since our evaluation is automatic and the Genia corpus annotation only captures syntactically fixed terms (i.e.

all the terms are contiguous word sequences), we decided to use an n-gram extraction technique. Moreover, our definition of MWT is essentially different from domain-specific phraseology. Thus, our candidates do not present much flexibility in terms of syntax, only allowing morphological inflection. With our POS pattern selection procedure, we obtain a list with more than 60.5K MWT candidates from which only 24% are true positive instances corresponding to almost 90% of the multi-word terms in the corpus.

To try to improve the quality of our MWT candidates, we decided to apply a machine learning algorithm using statistical association strength measures as features. We showed that the Genia corpus is sparse. Most of the candidates occur only once in the whole training corpus. In order to overcome sparseness, we used the Web to estimate word frequencies based on the number of pages in which they appear. However, during the evaluation we discovered that while the Web is a powerful tool for low-frequency words, it truncates high frequencies and becomes of little help. In future work, we would like to explore heuristic or statistical techniques for combining traditional corpora and Web frequencies to obtain better estimators in the association measures. Our evaluation showed that the *t* test and raw probabilities are good MWT indicators, whereas PMI does not help do distinguish true MWTs from random word combinations. We could not draw any further conclusion for the Dice coefficient and for most Web-based measures and we would like to investigate this phenomenon in depth in future experiments.

Several learning techniques were tested and evaluated for terminology extraction. Decision trees are easy to understand but their model is too simple for our data set. Bayesian learning, as well as support vector machines, seem to offer the best results in terms of precision and recall, but whether to choose higher recall or higher precision remains an application-dependent decision. Multi-layer perceptron are not adequate for our MWT candidates, but a voted perceptron can achieve quite high recall and acceptable precision. Based on our analysis, we decided to use radial-kernel SVMs in the rest of the evaluation. Although a good global quality, this learning algorithm is computationally costly and could possibly overfit the model, but this hypothesis was not investigated here and could be explored in future research. Finally, we evaluated the impact of frequency thresholds and compared our results with Xtract, a standard technique for collocation extraction. Frequency thresholds may have catastrophic impacts on recall, but considerably improve precision. Even the most restrictive threshold employed in our experiments yielded better precision and recall than Xtract, namely we obtained 74.14% precision and 6.42% recall on *Genia-test* while Xtract obtained 66.81% precision and 3.84% recall on *Genia-train*. This shows that a machine learning approach using several association measures and more than one corpus can obtain considerably better results than a method that uses a single association measure and sparse corpus frequencies.

If we take a look at a fragment of the final MWT list generated by our SVM classifier on the $n_{Genia} > 5$ list, we will get 5.1. We can observe that most MWTs are compound nouns or contain nouns in their structure, except for the two foreign (Latin) expressions *in vitro* and *in vivo*. The most common length of the extracted MWTs is two words, but some of the terms are 3- or 4-grams (e.g. *nuclear factor kappa b*). The false positives seem to be part of or to contain a term, because in some cases they are very similar to other candidates in the list. Manual annotation and 3 or 4 class evaluation (`MWT`, `NonMWT`, `Contains` and `PartOf`) could confirm this hypothesis, and this could be verified in future work. In this sense, the MWTs extracted by our method are very similar to the ones discovered by Xtract, where the most common configuration is a couple of nouns but some longer terms might also occur.

For the future, we would like to conclude and explore experiments on MT integration in what concerns the manual evaluation of the translations that take into account extracted MWTs. This would help draw more pragmatic conclusions about the feasibility of our approach.

At this point, the reader could be asking himself “But what is the point in automatically building

Table 5.1: Example of MWTs extracted from *Genia-test*. Errors based on automatic evaluation given our gold standard are marked with an asterisk "*".

Extracted MWT	POS	Extracted MWT	POS
t cell	N-N	***thromboxane receptor**	N-N
nf kappa b	N-N	peripheral blood	A-N
kappa b	N-N	nf kappa b	N-N-N
cell line	N-N	human t	A-N
transcription factor	N-N	receptor alpha	N-N
i kappa b	N-N-N	monocytic cell	A-N
***i kappa**	N-N	binding activity	N-N
***kappa b alpha**	N-N-N	stat protein	N-N
***b alpha**	N-N	***receptor gene**	N-N
i kappa b alpha	N-N-N-N	***rap1 protein**	N-N
gene expression	N-N	protein kinase	N-N
in vitro	FW-FW	in vivo	FW-FW
b cell	N-N	type 1	N-NUM
tyrosine phosphorylation	N-N	promoter activity	N-N
nk cell	N-N	beta 1	N-NUM
t lymphocyte	N-N	***b activation**	N-N
nuclear factor	A-N	***thromboxane receptor gene**	N-N-N
***nf kappa**	N-N	signal transduction	N-N
binding site	N-N	nuclear extract	A-N

terminology for a corpus that already contains annotated terminology". There are two answers for this question and both are very important for a complete comprehension of this work.

First, since the beginning our goal was not to create a terminology lexicon but to evaluate the techniques that could be used for doing that. When one evaluates a technique, one can either annotate the output by hand or one can obtain a gold standard against which one compares the results. Since the former solution is costly in terms of human resources, we adopted an automatic approach to evaluation, hence the need for an annotated corpus.

Of course we did not use annotation only for evaluation but also for patterns selection and learning, and this brings us to our second point: generalisation. Pattern selection does not necessarily need annotation, since most domains share some properties on the characteristics of the terminology. Noun-noun and adjective-noun combinations will always be very present and other variations might also occur. For this step, annotation is completely non-essential since this part of the technique can be performed on a trial and error basis, using heuristic and expert knowledge about the terminology of the domain and even using the POS patterns of similar domains. For instance, one could use our 57 POS patterns on a pediatrics or cardiology corpus and would probably obtain comparable performances. For the learning step, we assume that our features are totally domain-independent, i.e. we only look at association strength to decide whether a candidate is a good or a bad MWT. Therefore, we believe that the classifiers built here could be widely applied to other domains with comparable results, so that no annotation would be needed to apply this method, for example, on a chemistry or ecology corpus. In other words, we believe that not only our *methodology*, but also the *models* learned on the biomedical annotated data are general enough to be applied to other languages and domains, only requiring minor modifications. For the moment, this is only a (perhaps optimistic) hypothesis, as, given the scope of this work and the need for annotated data, all of the experiments are based on a single corpus. Further work should definitely investigate how these models behave when applied to other domains. Nonetheless, our evaluation is systematic enough to separate the data into training and test sets, and in the latter part the MWT annotation was completely ignored (the *Genia-test* part). Therefore, we are are optimistic about the fact that this technique could be adapted to other domains in a straightforward way, contributing for efficient domain adaptation of language tools and resources like machine translation and information retrieval. When these tools are able to cope with specialised lexicon, we will be able to make a big step toward wide knowledge diffusion and access, which is an important goal of Natural Language Processing itself.

REFERENCES

R. Harald Baayen. *Word Frequency Distributions*, volume 18 of *Text, Speech and Language Technology*. Springer, 2001. ISBN 978-0-7923-7017-8.

Timothy Baldwin. Deep lexical acquisition of verb-particle constructions. *Computer Speech & Language Special Issue on Multiword Expressions*, 19(4):398–414, 2005.

Timothy Baldwin, Emily M. Bender, Dan Flickinger, Ara Kim, and Stephan Oepen. Road-testing the English Resource Grammar over the British National Corpus. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal, May 2004. European Language Resources Association.

Youcef Bey, Christian Boitet, and Kyo Kageura. The TRANSBey prototype: an on- line collaborative wiki-based CAT environment for volunteer translators. In *Third LREC International Workshop on Language Resources for Translation Work, Research & Training (LR4Trans-III)*, pages 49–54, Genoa, Italy, May 2006.

Ergun Biçici and Marc Dymetman. Dynamic translation memory: Using statistical machine translation to improve translation memory fuzzy matches. In *Proceedings of the 9th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2008)*, Haifa, Israel, February 2008. Springer.

Christian Boitet, Valérie Bellynck, Mathieu Mangeot, and Carlos Ramisch. Towards higher quality internal and outside multilingualization of web sites. In Pushpak Bhattacharyya, editor, *Summer Workshop on Ontology, NLP, Personalization and IE/IR – ONII-08*, Bombay, India, 2008.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2), 1993. ISSN 0891-2017.

Helena Medeiros Caseli, Carlos Ramisch, Maria das Graças Volpe Nunes, and Aline Villavicencio. Alignment-based extraction of multiword expressions. *Language Resources & Evaluation Special Issue on Multiword Expressions (to appear)*, 2009.

Maurice de Kunder. Geschatte grootte van het geïndexeerde world wide web. Master's thesis, Universiteit van Tilburg, 2007.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1): 1–38, 1977.

Jianyong Duan, Ruzhan Lu, Weilin Wu, Yi Hu, and Yan Tian. A bio-inspired approach for multi-word expression extraction. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL 2006)*, pages 176–182, Sidney, Australia, July 2006. Association for Computational Linguistics.

Stefan Evert and Brigitte Krenn. Using small random samples for the manual evaluation of statistical association measures. *Computer Speech & Language Special Issue on Multiword Expressions*, 19(4): 450–466, 2005.

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

Roxana Girju, Dan Moldovan, Marta Tatu, and Daniel Antohe. On the semantics of noun compounds. *Computer Speech & Language Special Issue on Multiword Expressions*, 19(4):479–496, 2005.

Gregory Grefenstette. The World Wide Web as a resource for example-based machine translation tasks. In *Proceedings of the Twenty-First International Conference on Translating and the Computer*, London, UK, November 1999. ASLIB.

Dan Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, Cambridge, UK, 1997. ISBN 0-521-58519-8.

Cong-Phap Huynh, Christian Boitet, and Hervé Blanchon. SECTra_w.1 : an online collaborative system for evaluating, post-editing and presenting MT translation corpora. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco, May 2008. European Language Resources Association.

Ray Jackendoff. Twistin' the night away. *Language*, 73:534–59, 1997.

John S. Justeson and Slava M. Katz. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1):9–27, 1995.

Frank Keller and Mirella Lapata. Using the web to obtain frequencies for unseen bigrams. *Computational Linguistics*, 29(3):459–484, 2003. ISSN 0891-2017.

Frank Keller, Maria Lapata, and Olga Ourioupina. Using the Web to overcome data sparseness. In Jan Hajič and Yuji Matsumoto, editors, *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 230–237, Philadelphia, USA, July 2002. Association for Computational Linguistics.

Adam Kilgarriff and Gregory Grefenstette. Introduction to the special issue on web as corpus. *Computational Linguistics*, 29(3), 2003. ISSN 0891-2017.

Jin-Dong Kim, Tomoko Ohta, Yuka Teteisi, and Jun'ichi Tsujii. GENIA ontology. Technical report, Tsujii Laboratory, University of Tokyo, 2006.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL 2003)*, pages 48–54, Edmonton, Canada, 2003. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, pages 177–180, Prague, Czech Republic, July 2007. Association for Computational Linguistics.

Maria da Graça Krieger and Maria José Bocorny Finatto. *Introdução à Terminologia: teoria & prática*. Editora Contexto, 2004. ISBN 85-7244-258-8.

Mirella Lapata. The disambiguation of nominalizations. *Computational Linguistics*, 28(3):357–388, 2002. ISSN 0891-2017.

Adam Lopez. Statistical machine translation. *ACM Computing Surveys*, 40(3):1–49, 2008. ISSN 0360-0300.

Christopher D. Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT Press, Cambridge, USA, 1999. ISBN 0-262-13360-1.

I. Dan Melamed. Automatic discovery of non-compositional compounds in parallel data. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing (EMNLP-2)*, Brown University, USA, August 1997. Association for Computational Linguistics.

Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, USA, 1997.

Preslav Nakov and Marti Hearst. Search engine statistics beyond the n-gram: Application to noun compound bracketing. In Ido Dagan and Dan Gildea, editors, *Proceedings of the Ninth Conference on Natural Language Learning (CoNLL-2005)*, University of Michigan, USA, June 2005. Association for Computational Linguistics.

Mark E. J. Newman. Power laws, pareto distributions and zipf's law. *Contemporary Physics*, 46: 323–351, 2005. ISSN 0010-7514.

Sonja Nießen and Hermann Ney. Statistical machine translation with scarce resources using morpho-syntactic information. *Computational Linguistics*, 30(2):181–204, 2004. ISSN 0891-2017.

Sonja Nießen, Franz Joseph Och, Gregor Leusch, and Hermann Ney. An evaluation tool for machine translation: Fast evaluation for machine translation research. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC 2000)*, pages 39–45, Athens, Greece, May 2000. European Language Resources Association.

Tomoko Ohta, Yuka Tateisi, and Jin-Dong Kim. The GENIA corpus: an annotated research abstract corpus in molecular biology domain. In *Proceedings of the Second Human Language Technology Conference (HLT 2002)*, pages 82–86, San Diego, USA, March 2002. Morgan Kaufmann Publishers.

Darren Pearce. Synonymy in collocation extraction. In *WordNet and Other Lexical Resources: Applications, Extensions and Customizations (NAACL 2001 Workshop)*, pages 41–46, June 2001.

Darren Pearce. A comparative evaluation of collocation extraction techniques. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, Las Palmas, Canary Islands, Spain, May 2002. European Language Resources Association.

Pavel Pecina. A machine learning approach to multiword expression extraction. In *Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008)*, Marrakech, Morocco, June 2008.

Carol Peters, Valentin Jijkoun, Thomas Mandl, Henning Müller, Douglas W. Oard, Anselmo Peñas, Vivien Petras, and Diana Santos, editors. *Advances in Multilingual and Multimodal Information Retrieval, 8th Workshop of the Cross-Language Evaluation Forum, Revised Selected Papers*, volume 5152 of *Lecture Notes in Computer Science*, Budapest, Hungary, 2008. Springer. ISBN 978-3-540-85759-4.

Grzegorz Protaziuk, Marzena Kryszkiewicz, Henryk Rybinski, and Alexandre Delteil. Discovering compound and proper nouns. In *Proceedings of the RSEISP'07 International Conference on Rough Sets and Emerging Intelligent Systems Paradigms*, pages 505–515, 2007.

Joaquin Quiñonero-Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché Buc, editors. *The PASCAL Recognising Textual Entailment Challenge*, volume 3944, 2005. Springer. ISBN 978-3-540-33427-9.

Carlos Ramisch, Paulo Schreiner, Marco Idiart, and Aline Villavicencio. An evaluation of methods for the extraction of multiword expressions. In *Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 50–53, Marrakech, Morocco, June 2008.

Ivan Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for NLP. In *Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2002)*, pages 1–15, Mexico City, Mexico, February 2002. Springer.

Eric SanJuan, James Dowdall, Fidelia Ibekwe-SanJuan, and Fabio Rinaldi. A symbolic approach to automatic multiword term structuring. *Computer Speech & Language Special Issue on Multiword Expressions*, 19(4):524–542, 2005.

Violeta Seretan. *Collocation extraction based on syntactic parsing*. PhD thesis, University of Geneva, 2008.

Frank A. Smadja. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143–177, 1993. ISSN 0891-2017.

Aline Villavicencio. The availability of verb-particle constructions in lexical resources: How much is enough? *Computer Speech & Language Special Issue on Multiword Expressions*, 19(4):415–432, 2005.

Aline Villavicencio, Valia Kordoni, Yi Zhang, Marco Idiart, and Carlos Ramisch. Validation and evaluation of automatically acquired multiword expressions for grammar engineering. In Jason Eisner, editor, *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pages 1034–1043, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Yi Zhang, Valia Kordoni, Aline Villavicencio, and Marco Idiart. Automated multiword expression prediction for grammar engineering. In *Proceedings of the COLING/ACL Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, pages 36–44, Sidney, Australia, July 2006. Association for Computational Linguistics.

Xiaohua Zhou, Xiaodan Zhang, and Xiaohua Hu. Dragon toolkit: Incorporating auto-learned semantic knowledge into large-scale text retrieval and mining. In *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence - ICTAI 2007*, volume 2, pages 197–201, Washington, USA, 2007. IEEE Computer Society.

ANNEXE A RÉSUMÉ ÉTENDU

EXTRACTION DE TERMES MULTIMOT POUR DES DOCUMENTS SPÉCIALISÉS

A.1 Introduction

L'un des objectifs majeurs du Traitement Automatique des Langues (TAL) est de rendre la communication entre les êtres humains et les ordinateurs aussi simple que demander un service à un ami. Le langage naturel est une composante essentielle au comportement intelligent, ce qui conduit à la classification du TAL en tant que sous-domaine de l'intelligence artificielle, dont le TAL hérite des nombreux principes et méthodes. Le langage humain est, pour plusieurs raisons, complexe pour la modélisation informatique. Une fraction significative de la connaissance humaine est exprimée par le langage naturel : il est donc un moyen d'échange de connaissances et se situe ainsi à la base de toute communication dans le plan scientifique. Par conséquent, le TAL porte un grand intérêt à ce que l'on appelle le langage spécialisé ou le langage technico-scientifique. Une des caractéristiques de cette langue de spécialité est la richesse de son lexique en structures terminologiques, ces dernières étant l'objet de ce travail. Dans ce contexte, nous croyons que la terminologie peut être acquise automatiquement à partir de corpus spécifiques à des domaines et de techniques d'apprentissage dirigé par les données.

Supposons un utilisateur fictif qui doit trouver des informations sur la traduction automatique basée sur les exemples. Avec un moteur de recherche sur le Web, cette tâche peut être fastidieuse, car non seulement la majorité des résultats seront en anglais, mais il sera également difficile de filtrer la grande quantité de contenu (non pertinent) généré par les utilisateurs. Les moteurs de recherche spécialisés doivent tenir compte de la terminologie du domaine, par exemple les mots *arbre* et *feuille* sont des termes, mais les concepts qu'ils représentent ne sont pas les mêmes en botanique et en informatique. En somme, aujourd'hui, l'utilisateur ayant un besoin d'information spécifique à un domaine exprimé dans sa langue maternelle doit d'abord consulter plusieurs sources d'information auxiliaires pour traduire les mots-clés, et ensuite les documents renvoyés pas la recherche. Non seulement sur le Web mais de façon générale, l'accès multilingue aux textes spécialisés est d'un grand intérêt parce qu'il a des applications dans tous les domaines de la science et de la diffusion des produits. L'étude de méthodes pour la recherche d'informations multilingues n'entre pas dans le contexte de ce travail, mais l'exemple ci-dessus sert à motiver l'intégration des termes du domaine avec la traduction automatique.

Tout d'abord, nous devons définir trois concepts-clés avec lesquels nous travaillerons : *expression multimot*, *terme* scientifique et technique, et *terme multimot*. La définition d'Expression Multimot (EMM) est vaste, puisque ce terme englobe de nombreux phénomènes linguistiques tels que les noms composés (*traffic light*, *feu rouge*), les expressions idiomatiques (*clound number nine*, *sem pé nem cabeça*), les termes composés (*système d'exploitation*, *Datenbanksystem*), etc. Les termes sont, contrairement aux EMM de la langue courante, des entités linguistiques liées au texte technique et scientifique. Un terme peut être une unité mono-lexicale tandis qu'une EMM est nécessairement composée de plusieurs mots. Dans la Théorie Générale de la Terminologie, il est estimé qu'il existe une relation fonctionnelle et injective (1 :1) entre les termes et les concepts. Cette idée a toutefois été sévèrement critiquée pour être très réductrice et pour ne pas tenir compte de l'habitat naturel des termes : le texte. Les Termes Multimot (TMM) sont définis comme des locutions qui ont le statut de terme. Nous soulignons que les termes multimot ne correspondent pas à la notion de phraséologie du domaine. Un TMM accepte peu de variabilité (morphologique, mais rarement syntaxique), tandis qu'une phraséologie est hautement variable. Alors que le premier représente un concept particulier, il n'est pas rare que la phraséologie soit une structure complexe comprenant plus d'un concept.

La traduction et la terminologie ont de nombreux points communs en ce qui concerne la traduction spécialisée. De même, les TMMs et la traduction automatique (TA) devraient être des domaines proches, malgré la réalité actuelle dans le TAL qui reflète peu cette interdépendance. Ainsi, l'un des objectifs de ce travail est la détection automatique de termes pour l'adaptation des ressources linguistiques informatisées à plusieurs domaines. Nous avons choisi de travailler uniquement avec les termes multimot parce que (a) ils représentent environ 70% de la terminologie d'un domaine, (b) les méthodes en informatique pour la détection des termes multimot et « mono-mots » sont très différentes et (c) les TMMs représentent un défi majeur en linguistique computationnelle.

Definition A.1 *Une Expression Multimot (EMM) est une suite de deux mots ou plus dont la sémantique est non-compositionnelle, c'est-à-dire que le sens du syntagme ne peut pas être totalement compris à travers le sens des mots qui le composent (Sag et al. 2002).*

Definition A.2 *Un terme est une unité lexicale ou multilexicale avec un sens non ambigu quand elle est employée dans un texte spécialisé, de façon à ce que la terminologie est la manifestation linguistique des concepts d'un domaine (Krieger and Finatto 2004).*

Definition A.3 *Un Terme Multimot (TMM) est un terme composé par plus d'un mot (SanJuan et al. 2005, Frantzi et al. 2000).*

A.2 Révision bibliographique

Le travail présenté ici traite de l'intégration de trois sous-domaines du TAL : les expressions multimot, la terminologie et la traduction statistique. Les EMM constituent un défi pour le TAL en raison de leur grande variabilité morphosyntaxique et de leur nature hétérogène. Les techniques pour le traitement des EMM sont classifiées comme étant soit du type « mots-avec-espaces » soit compositionnelles. Quatre tâches sont liées au traitement des EMMs : l'identification de leur occurence dans les textes, l'interprétation de leur structure (syntaxe), leur classification ou construction de groupements (sémantique) et leur application à d'autres systèmes de TAL. Il est possible d'identifier les candidats à EMM à travers des motifs morphologiques de surface (Villavicencio et al. 2007, Manning and Schütze 1999) ou par des métodes plus ou moins profondes qui tiennent compte des phénomènes syntaxiques (Seretan 2008, Baldwin 2005). Le filtrage des candidats peut se faire par des mesures statistiques (Evert and Krenn 2005, Villavicencio et al. 2007) souvent complétées par des fréquences obtenues dans le Web (Keller et al. 2002, Zhang et al. 2006). Les approches symboliques emploient généralement des thésaurus et des dictionnaires de synonymes (Pearce 2001). Aussi l'apprentissage artificiel suppervisé peut se montrer effectif pour la construction de modèles d'identification de EMMs (Pecina 2008). L'interprétation de la structure interne des EMMs dépend de leurs types spécifique, par exemple, Ramisch et al. (2008) classent les verbes composés en anglais par rapport à leur idiomaticité, et Nakov and Hearst (2005) travaillent sur la structure d'imbrication des noms composés. Par rapport à la classification des EMMs, nous citons le travail de Lapata (2002), qui traite en particulier des nominalisations.

Une technique couramment utilisée pour l'extraction de termes est l'utilisation de motifs morphologiques fréquents (Justeson and Katz 1995). Frantzi et al. (2000) proposent une technique plus fine pour le classement des candidats, pendant que Smadja (1993) utilise l'information mutuelle des mots afin de détecter des collocations. Par rapport à la terminologie biomédicale du corpus Genia, SanJuan et al. (2005) effectuent une classification sémantique automatique qui est ensuite comparée à l'ontologie du domaine.

La traduction automatique (TA) a évolué au fil des dernières années vers des méthodes collaboratives au niveau opérationnel ou statistiques. Les premières sont fondées sur le triplet automaticité, couverture et précision, permettant à l'utilisateur de choisir quels aspects sont plus importants pour leur application (Bey et al. 2006, Boitet et al. 2008). D'autre part, les systèmes de traduction statistique apprennent les probabilités de traduction à partir d'un corpus parallèle (Brown et al. 1993). L'alignement au niveau des mots est obtenu à travers l'algorithme d'apprentissage non supervisé EM. Selon Koehn et al. (2003), les systèmes de TA statistiques peuvent être de deux types : basés sur les mots ou basés sur les syntagmes (séquences de mots). Dans des domaines spécialisés, des problèmes peuvent se produire lorsque le corpus d'entraînement est différent du texte à traduire, comme le montre Baldwin et al. (2004) sur un sustème d'analyse syntaxique. L'identification simultannée des TMMs et des alignements au niveau des mots permet d'extraire à la fois les unités lexicales multimot et leurs traductions (Melamed 1997, Caseli et al. 2009).

A.3 Extraction de la terminologie biomédicale

Le corpus Genia est un ensemble de 2.000 résumés dans le domaine des sciences biomédicales sur les mots-clés « humain », « cellules sanguines » et « facteur de transcription » (Ohta et al. 2002). Il contient 18,5K phrases et 490,7K tokens, annotés avec la classe morphosyntaxique et avec la terminologie, comme les noms de maladies (*tumeur fibroblastique*) et de cellules (*lymphocytes t primaires*). En moyenne, les phrases contiennent 3 TMMs, chacun avec une moyenne de 3 tokens. Cela montre que la terminologie est omniprésente dans ce corpus. Les termes du corpus Genia ont tendance à être profondément imbriqués. Les expériences décrites ci-dessous ont été implémentées à travers une série de scripts en Python, awk et bash disponibles à l'adresse `www.inf.ufrgs.br/~ceramisch/tcc.zip`.

Tout d'abord, les données du corpus ont été prétraitées (SanJuan et al. 2005) : toutes les phrases ont été converties en minuscules et une version simplifiée du corpus en XML a été générée. Dans cette version, l'annotation des termes est gardée, non pas pour être utilisée pendant l'extraction, mais afin de servir comme base de comparaison lors de l'évaluation. D'une part, un *candidat à TMM* est extrait du corpus s'il obéit à certaines normes ou règles pré-établies, indépendamment de l'annotation. D'autre part, la *terminologie standard* est l'ensemble des termes annotés qui serviront comme base de comparaison pour l'évaluation automatique des candidats. La terminologie standard de Genia possède 28.340 TMM distincts. Prenons l'exemple ci-dessous (la classe du mot est indiquée par un slash, les termes sont entre crochets) :

Example A.1 `[ [ alpha/N a/UKN ] -gene/A product/N ]`

Remarquez que l'annotation terminologique casse les mots dans les tirets et dans les slashs. Cela est nuisible puisque à l'une des moitiés du mot est affectée une classe inconnue (UKN). Comme ce phénomène arrive dans 17% des phrases du corpus, nous avons décidé d'inclure le token extérieur dans l'expression, de façon à améliorer la qualité de l'annotation. Le résultat peut illustré par l'exemple ci-dessous. Avec cette modification, seuelement 81 phrases sont éliminées pour avoir des mots avec une classe inconnue :

Example A.2 `[ [ alpha/N a-gene/A ] product/N ]`

Afin de prendre en compte les déclinaisons de nombre, il a fallu réduire chaque réalisation d'un mot à sa forme au singulier. Un lemmatiseur généraliste pourrait ramener le nom *binding* au verbe *bind*, tandis qu'une simple règle d'élimination de la lettre « s » à la fin des mots aboutirait à des lemmes comme *virus → viru** ou *viruses → viruse**. Par conséquent, nous avons développé un lemmatiseur simple basé sur des règles et sur des fréquences du Web, dont le fonctionnement est décrit en détail dans l'annexe D.

La dernière étape de prétraitement concerne une caractéristique propre aux textes biomédicaux, qui ont tendance à insérer des acronymes entre parenthèses après les termes récurrents. Lorsque ces parenthèses se présentent à l'intérieur d'un TMM, ils génèrent du bruit pour les filtres, comme dans l'exemple *[[granulocyte-macrophage colony-stimulating factor] (gm-csf) promoter]*. Environ 3% des termes contiennent ces parenthèses, et pour supprimer uniquement celles contenant des acronymes, nous avons utilisé un algorithme qui combine des règles et une procédure de correspondance par sous-chaîne de longueur maximale, décrite dans l'annexe D. Ainsi, nous sommes capable d'identifier les acronymes dans le corpus et, ensuite, éliminer ceux qui apparaissent entre parenthèses. Cette liste d'acronymes (par exemple *AD – alzheimer 's disease*, *5LOX – 5 lipoxygenase*, *AD – adenovirus*) est un produit supplémentaire à l'extraction des TMMs.

Une fois que le corpus a été prétraité de manière uniforme, il est nécessaire d'extraire les candidats à TMM. Ainsi, avant tout il faut le diviser en deux parties : *Genia-test* (100 résumés, 895 phrases) et *Genia-train* (le reste). La première partie est prévue pour une utilisation future au cours de la phase d'évaluation. La dernière partie est utilisée pour effectuer non seulement l'apprentissage des motifs morphologiques mais aussi des poids donnés à chaque mesure d'association durant l'étape de filtrage des candidats, postérieure à l'extraction.

L'extraction de motifs morphologiques est effectuée de la façon suivante : (1) en nous basant sur l'annotation des termes du corpus, nous avons extrait la terminologie standard de *Genia-train*, (2) nous avons sélectionné les 118 motifs qui apparaissent plus de 10 fois, (3) parmi ces derniers, nous avons gardé seulement les 57 motifs dont la précision (proportion de candidats positifs) est supérieure à 30% et (4) nous avons extrait du corpus l'enseble des n-grammes qui corespondent à ces motifs. Les valeurs des seuils de fréquence et de précision ont été obtenues à partir de l'observation empirique des données. Les motifs sélectionnés comprennent des séquences de noms et d'adjectifs (*N-N*, *A-N*, *A-N-N*), mais aussi de mots étrangers (*FW-FW*), et des verbes (*N-V-N*). Le résultat de cette étape est une liste avec 60.585 candidats à être des TMM, dont 22.507 sont positifs (31,52%).

Après avoir extrait les candidats, nous procédons alors à leur filtrage. Les filtres sont basés sur des Mesures d'Association (MA) statistiques (Evert and Krenn 2005, Ramisch et al. 2008, Pecina 2008). La distribution de probabilités du vocabulaire de la langue est zipfienne et comporte un phénomène connu sous le nom de « longue queue ». Par conséquent, les fréquences deviennent très faibles, ce qui implique une basse fiabilité pour les MA. Ainsi, nous avons utilisé également les fréquences obtenues à partir du World Wide Web (via une API de recherche de *Yahoo!*), que nous considérons ici comme un corpus de langue générale dorénavant dénommé *Web*. Pour estimer la taille N du corpus, nous avons utilisé une estimation grossière[1], qui est simplement un facteur d'échelle linéaire identique pour tous les candidats. La liste d'acronymes est utilisée ici pour augmenter la fréquence des mots qui y participent à chacune de leurs occurences. Pour chaque candidat $w_1 \dots w_n$ et pour chaque corpus de taille N, la fréquence simple $f(w_1 \dots w_n)$ et les fréquences marginales $f(w_1) \dots f(w_n)$ sont utilisées pour le calcul des MA ci-dessous :

[1] 50 milliards de pages, selon `http://www.worldwidewebsize.com/`

$$\texttt{prob} = \frac{f(w_1 \ldots w_n)}{N} \qquad \texttt{PMI} = log_2 \frac{f(w_1 \ldots w_n)}{N^{n-1} f_\emptyset(w_1 \ldots w_n)}$$

$$\texttt{t} = \frac{f(w_1 \ldots w_n) - N^{n-1} f_\emptyset(w_1 \ldots w_n)}{\sqrt{f(w_1 \ldots w_n)}} \qquad \texttt{Dice} = \frac{n * f(w_1 \ldots w_n)}{\sum_{i=1}^{n} f(w_i)}$$

Ici, l'hypothèse nulle suppose l'indépendance entre les mots, c'est-à-dire que leur co-occurence se produit par hasard et que leur fréquence correspond à $f_\emptyset(w_1 \ldots w_n) = (f(w_1) \ldots f(w_n))/N^{n-1}$. Les mesures qui utilisent le tableau de contingence, même si elles sont plus robustes, ne sont pas utilisées ici, parce que, si n est arbitraire, il est non seulement difficile de produire un hypercube de fréquences, mais aussi le bruit et les incohérences (surtout dans le corpus *Web*) empêchent en pratique son utilisation. Plusieurs travaux discutent le choix de la meilleure MA, mais notre méthode consiste à utiliser l'apprentissage artificiel pour cette tâche. Nous séparons donc les candidats à filtrer en deux catégories : `MWT` et `NonMWT`. Les algorithmes utilisés incluent les arbres de décision (J48), les classificateurs bayésians par méthodes statistiques, les réseaux de neurones artificiels (MLP et *Voted Perceptron*) et les classificateurs numériques qui agissent sur la frontière de décision, c'est-à-dire les machines à vecteur de support (SVM).

Une fois que les termes multimot ont été extraits du corpus, nous proposons des moyens pour les intégrer la boîte à outils Moses de TA statistique. Le système a été entraîné sur le corpus Europarl avec la paire de langues anglais → portugais, pour laquelle nous disposons de locuteurs natifs aptes à évaluer les résultats. Le corpus Europarl a été aligné au niveau des phrases, filtré, la séparation des mots et la casse ont été homogénéisées, puis nous avons séparé deux sous-ensembles pour le développement et pour les tests. Ensuite, nous avons généré un modèle de langue, les alignements mot à mot et la table de traduction, pour finalement apprendre, par minimisation du taux d'erreur, les coefficients de pondération de chaque attribut du modèle.

Il existe plusieurs techniques proposées dans la littérature pour séparer et recombiner des mots composés en allemand ou en suédois (Nießen and Ney 2004, Nießen et al. 2000). Nous utilisons une méthode similaire pour les termes en anglais : considérons, à titre d'exemple, la phrase d'entrée *ras-related gtp-binding proteins and leukocyte signal transduction*. La première approche (wws) consiste à concaténer les mots qui intègrent un terme, par exemple, *ras-related#gtp-binding#proteins and leukocyte signal#transduction*. La deuxième méthode (head) est de remplacer le terme par son noyau, supposé être le dernier nom du n-gramme, par exemple, *proteins and leukocyte transduction*. La terminologie standard a également été utilisée pour évaluer la séparation et la recombination de termes dans Moses. La technique de remplacement du noyau simplifie le texte original pour faciliter le fonctionnement du système de TA, de manière que le texte original entraîne le texte simplifié, qui est alors traduit. Sémantiquement, l'information véhiculée par la traduction contient le sens de la phrase originale, mais ne lui est pas équivalent.

A.4 Évaluation

La première évaluation concerne le prétraitement, et est donc effectuée sur la totalité du corpus. Nous appellerons ici « normalisation » le processus de lemmatisation et homogéneisation de la séparation de mots. Cette étape est suivie par l'Élimintation d'Acronymes (EA). L'évaluation manuelle sur un ensemble de mots de l'algorithme de transformation pluriel-singulier montre qu'il donne un résultat correct sur 99% des mots. Dans le tableau A.1, nous pouvons vérifier que, vis-à-vis de la configuration sans normalisation, la terminologie standard est 3,89% fois plus petite, alors que la liste des candidats est réduite de 5,35%. Cela montre que les candidats et termes similaires (par exemple, où l'un est le pluriel de l'autre) sont désormais traités en tant qu'entité unique représentant un même

TAB. A.1 – Évaluation de la normalisation et de l'Élimination d'Acronymes (EA).

	Sans norm.	**Sans EA**	**Avec norm. et EA**
TMMs de la term. standard	29.513	28.601	28.340
Candidats à TMM	66.817	61.401	63.210
Précision	15,56%	37,53%	37,24%
Rappel	35,22%	80,58%	83,06%
F-measure	21,58%	51,20%	51,42%
Fréquence moyenne des top-100	189,98	205,05	205,38
Hapax legomena	51.169 (77%)	46.330 (75%)	47.770 (76%)

concept. Malgré son aspect réductionniste (le contexte est ignoré), cette approche donne des résultats pratiques intéressants : le rappel augmente significativement avec la normalisation, de 35% à 83% ; la précision est doublée, passant de 15% à 37%. De plus, les fréquences des mots sont moins faibles puisque les candidats similaires sont regroupés et leurs nombre d'occurrences additionnées. Le nombre de hapax legomena (qui apparaissent une seule fois) diminue, mais correspond encore aux deux tiers des candidats. D'autres heuristiques de normalisation pourraient être utilisées, comme par exemple l'unification des déclinaisons de genre ou des conjugaisons verbales. Il est néanmoins important de souligner que tout traitement de ce type est fortement dépendant de la langue et du domaine en question.

Tojours dans le même tableau, nous présentons la configuration sans EA. Ici, la terminologie standard est aussi plus petite, mais la réduction n'est pas aussi importante que nous l'espérions (la liste d'acronymes a 1.300 entrées), probablement parce que la plupart des acronymes apparaissent après les TMMs, et non pas au milieu d'eux. Plus de candidats sont extraits, ce qui augmente le rappel de 2,5% mais diminue légèrement la précision. Il semble aussi que l'EA aide à générer des données moins rares. Les acronymes sont également utilisés pour augmenter la fréquence des mots qu'ils remplacent. Par conséquent, le nombre d'occurrences marginal moyen augmente de 37 à 47. Cependant, l'importance de l'étape d'élimination des acronymes est remise en question, car les améliorations provoquées sont peu visibles sur le résultat final.

TAB. A.2 – Précision et rappel des motifs de TMM.

Motifs morphologiques	**Précision**	**Rappel**
J&K	**29,58%**	66,71%
$f > 10$	16,50%	**92,57%**
$f > 10, p > .3$	31,52%	89,43%

Une fois que le prétraitement est terminé, nous voulons évaluer l'étape de sélection des motifs morphologiques des candidats. Quand nous comparons les motifs utilisés par Justeson and Katz (1995) (J&K), ceux qui ont plus de 10 occurrences et ceux qui ont été sélectionnés ($f > 10, p > .3$), nous pouvons voir que, d'une part, J&K ont un bon rappel mais une précision très faible sur ce corpus (où les TMMs ont tendence à être longs et imbriqués), et que, d'autre part, les motifs les plus fréquents ont un rappel élevé tandis que la précision est légèrement supérieure à la moitié de J&K.

Cela nous mène aux motifs sélectionnés, qui offrent un bon équilibre, avec la plus grande précision parmi les ensembles de motifs étudiés et un rappel légèrement inférieur à celui de l'ensemble $f > 10$. Parmi les motifs sélectionnés, nous soulignons (a) des problèmes par rapport aux conjonctions comme dans *young and aged subject*, qui est en fait composé de deux TMMs indépendants, TAB. A.3 – Précision moyenne (%) des MA statistiques.

	Genia				*Web*				
	prob	*PMI*	*t*	*Dice*	*prob*	*PMI*	*t*	*Dice*	**Base**
train	46,16	40,08	52,90	38,84	35,77	41,02	45,13	37,85	37,15
test	38,83	17,51	31,36	22,78	41,49	20,52	43,21	43,74	24,16

(b) la présence de motifs contenant des chiffres (*interleukin 2*) et (c) l'absence du motif N-P-N, qui est considéré par J&K comme un bon indicateur de TMM. Il est intéressant de vérifier que, quand la terminologie standard de l'ensemble d'entraînement est utilisée pour évaluer l'ensemble de test, les motifs sélectionnent environ deux fois plus de termes que l'annotation existante. Ainsi, le rappel atteint 200% alors que, pour chaque motif, la précision oscille entre 65% et 95%. Ces résultats montrent qu'une annotation plus soigneuse pourrait améliorer les valeurs absolues de l'évaluation.

Le tableau A.3 montre la précision moyenne de chacune des mesures d'association ainsi que la précision de la ligne de base (proportion initiale de candidats positifs) pour les fragments d'entraînement et de test. D'après ces valeurs, malgré la supposition fausse que les mots ont une distribution normale, le test *t* a une bonne performance sur les deux parties du corpus, ce qui est en accord avec les résultats d'Evert and Krenn (2005), Manning and Schütze (1999). La probabilité simple (prob) semble également un bon indicateur de TMM puisque sa précision moyenne est élevée. Parmi les mesures évaluées, PMI est celle qui présente la plus faible performance, étant même inférieure à la ligne de base sur *Genia-test*, contrairement à ce qui est indiqué dans Smadja (1993). Cependant, Evert and Krenn (2005) trouvent le même résultat et affirment que le test *t* et le rapport de vraisemblance logarithmique (mesure qui utilise le tableau de contingence) ont souvent les meilleures performances. Un algorithme simple de sélection d'attributs confirme que les mesures les plus pertinentes sont $prob_{Genia}$, t_{Genia} et t_{Web}. Quelques incohérences ont été observées dans le comportement des MA sur le corpus *Web*, puisque les mesures semblent mauvaises sur la partie *Genia-train* tandis qu'elles sont bonnes sur *Genia-test*.

L'analyse du vocabulaire du corpus *Web* (figure A.1) montre qu'il existe une explication possible pour ce comportement apparemment arbitraire des mesures sur le *Web*. L'API de *Yahoo!* retourne, pour les 735 mots les plus fréquents, un nombre constant d'occurrences (2.147.483.647), de sorte qu'il est impossible de faire la différence entre des mots comme *the* et *green* (qui apparaissent respectivement 22.446 et 1 fois dans le corpus Genia). Cette situation pathologique, résultat d'une ap-

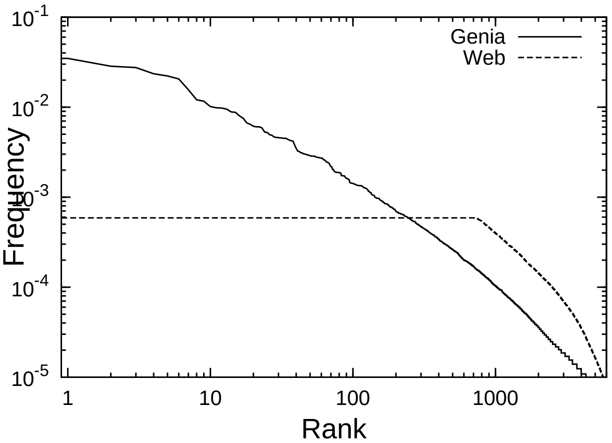


FIG. A.1 – Histogramme du vocabulaire de *Genia-train* e du *Web*.

TAB. A.4 – Performance des algorithmes d'apprentissage.

	J48	**Bayes(TAN)**	**Bayes(SA)**	**MLP**	**VP**	**SVM(pol.)**	**SVM(rad.)**
Précision	47.9%	48.6%	52.4%	27.3%	36.3%	34.9%	**52.6%**
Rappel	19.7%	32.1%	28.9%	**0.5%**	**63.8%**	60.1%	**35.4%**

TAB. A.5 – Évaluation comparative de la méthode SVM avec plusieurs seuils et l'algorithme Xtract.

	Genia-test				*Genia-train*
	Aucun seuil	$f_{Genia} > 1$	$f_{Genia} > 5$	**Xtract**	**Xtract**
nb. TMMs extraits	763	739	174	70	1.558
Précision	52,56%	56,83%	74,14%	70%	66,81%
Rappel	19,96%	20,91%	6,42%	2,44%	3,84%

proximation grossière de la part du moteur de recherche, génère la courbe « cassée » visible dans le graphique, certainement nuisible aux mesures d'association. Ainsi, contrairement à ce que suggèrent Zhang et al. (2006), le coefficient de corrélation de Pearson entre les corpus est de seulement 0,12 alors que le coéfficient τ de Kendall est de 0,21. Dans l'avenir, nous souhaitons étudier une façon d'intégrer les points forts des deux corpus, puisque Genia sous-estime la probabilité des mots peu fréquents, tandis que *Web* le fait pour les mots les plus fréquents.

La performance des algorithmes d'apprentissage artificiel sur le corpus *Genia-test* est comparée dans le tableau A.4. Seul la classe `MWT` est considérée parce que, comme les données ne sont pas équilibrées, la classe `NonMWT` a une F-mesure supérieure à 80% pour tous les algorithmes. L'algorithme J48 génère un modèle de données facile à interpréter (un arbre avec 484 nœuds) qui classifie correctement 75% des instances. Les attributs les plus proches de la racine (les plus pertinents) sont les valeurs *t* dans les deux corpus, tandis que l'attribut qui semble être le moins important est le motif morphologique, généralement dans le nœud qui précède les feuilles.

Le problème de cet algorithme (et bien d'autres) est qu'il considère que les deux classes ont le même poids. Dans ce sens, l'apprentissage bayésien est plus efficace, surtout quand la recherche par arbre de dispersion minimal (TAN) est utilisée au lieu de la recherche par recuit simulé (SA).

La couverture de ces deux méthodes reste décevante parce qu'elles ont le même problème que l'algorithme J48, i.e. elles donnent un poids égal aux deux classes. Le réseau de neurones multicouche (MLP) avec 33 neurones dans la couche caché a une performance très basse : seulement 5% des candidats positifs sont capturés.

L'algorithme *Voted Perceptron (VP)* a une performance surprenante, avec un très bon rappel sur les candidats positifs. Les causes de ce résultat restent à étudier.

Les meilleurs résultats ont été obtenus avec les machine à vecteur de support (SVM) de noyau polynomial et radial. Chacun tend à donner plus d'importance à l'un des aspects : le premier a un bon rappel alors que le dernier a une précision de 52,6%. Le choix entre un noyau polynomial ou radial dépend de l'application. Par exemple, la construction d'un dictionnaire est très onéreuse, de sorte qu'une haute précision diminue la quantité d'effort manuel à fournir. Cependant, une application de TAL peut éventuellement vivre avec des données bruitées à condition que le rappel soit suffisamment grand. Dans lesétapes suivantes, nous avons choisi un SVM à noyau radial, qui renvoie sur l'ensemble de test une liste de 763 instances positives dont 362 sont fausses. Au cas où nous n'aurions pas filtré les données avec un SVM, la proportion de candidats positifs serait de seulement 24%.

Malgré l'impression que l'on a, que les valeurs absolues de précision et de rappel sont basses.

Quand nous comparons ces résultats avec le système conventionnel d'extraction d'unités multimot Xtract, nous pouvons voir que la méthode suggérée ici est supérieure. Dans le tableau A.5, nous comparons la performance du classificateur SVM avec plusieurs valeurs de seuil de fréquence sur la liste de candidats. La configuration où nous éliminons les hapax legomena ($f_{Genia} > 1$) fait monter la précision et le rappel, ce qui montre que ces candidats doivent effectivement être enlevés de la liste. Cependant, un seuil haut ($f_{Genia} > 5$) fait rapidement chuter le rappel. Comme notre méthode utilise les informations présentes dans *Genia-train* pour extraire les TMMs de *Genia-test*, nous la comparons avec Xtract sur l'ensemble de test, et aussi avec Xtract sur l'ensemble d'entraînement, plus grand donc avec des fréquences plus fiables et moins faibles. Néanmoins, dans les deux cas, Xtract est très conservateur et génère une liste de TMMs avec une précision assez haute mais un rappel extrêmement bas. Nous observons que, même dans la configuration la plus restrictive, le SVM obtient un résultat supérieur à celui obtenu par Xtract, prouvant ainsi que la méthode d'extraction de TMMs proposée dans ce travail a un gain réel sur un algorithme standard.

Finalement, pour des raisons pragmatiques de temps de traitement et d'espace de stockage, il n'a pas été possible d'achever l'évaluation de l'intégration TMM-TA. L'idée de cette étape est de fournir aux évaluateurs (des locuteurs natifs) une liste de traductions produites par les différentes méthodes d'intégration (wws et head avec TMMs extraits et terminologie standard). Les traductions seront alors évaluées par leur fluidité, car leur adéquation sera certainement plus basse, pourvu que la méthode effectue une simplification sur l'entrée. Une interface permettant la visualisation des termes dans l'original et la collaboration dans leur traduction sera également créée et soumise à des techniques standard d'évaluation d'interfaces.

A.5 Conclusions

Le travail présenté ici offre une évaluation systématique des techniques d'extraction terminologique sur un corpus biomédical. D'abord, nous avons discuté les hypothèses et les notions de base d'expression et terme multimot, ainsi que leur caractérisation à travers des théories linquistiques. Plusieurs travaux traitent les expressions multimot avec un type et dans une langue déterminés. La plupart des tâches et des méthodes discutées ont comme base le bigramme, et sur ce point nous croyons que la méthode proposée ici est supérieure, puisque nous sommes capable de traiter les TMMs avec une longueur n quelconque.

En somme, notre méthode est composée de quatre étapes : prétraitement, extraction de candidats à travers des motifs morphologiques sélectionnés, filtrage des candidats à travers une combination de mesures d'association et d'algorithmes d'apprentissage artificiel, et en dernier lieu, intégration de la terminologie avec les systèmes de TA statistiques. Le tableau A.6 liste quelque expressions extraites de *Genia-test* avec le SVM $n_{Genia} > 5$. La plus grande partie des termes sont des noms composés avec prédominance des bigrammes. Les exemples négatifs semblent soit appartenir à un terme plus long, soit contenir un terme plus court ; probablement une évaluation fine avec des classes comme `Contains` et `PartOf` pourrait vérifier cette hypothèse.

Il est important de souligner le fait que la terminologie extraite ne veut pas, en aucun moment, remplacer ou améliorer l'annotation existant a priori dans le corpus. L'objectif de ce travail n'était pas la génération d'une nouvelle terminologie pour le corpus Genia, parce que celui-ci comporte déjà une telle structure. Au contraire, l'annotation des termes a été utilisée pour atteindre notre objectif initial qui était d'évaluer (de façon automatisée) les méthodes d'extraction d'unités multimot dans un domaine spécifique.

Dans l'avenir, nous voulons évidemment conclure l'évaluation suggérée ici par l'intégration entre terminologie et traduction automatique, afin d'établir la validité de la méthodologie que nous proposons. De plus, nous voulons étudier de nouvelles façons de combiner les fréquences obtenues dans des

TAB. A.6 – Exemples de TMMs extraits de *Genia-test* (« * » représente une erreur).

TMM extrait	Motif	TMM extrait	Motif
t cell	N-N	**thromboxane receptor***	N-N
nf kappa b	N-N	peripheral blood	A-N
kappa b	N-N	nf kappa b	N-N-N
cell line	N-N	human t	A-N
transcription factor	N-N	receptor alpha	N-N
i kappa b	N-N-N	monocytic cell	A-N
i kappa*	N-N	binding activity	N-N
kappa b alpha*	N-N-N	stat protein	N-N
b alpha*	N-N	**receptor gene***	N-N
i kappa b alpha	N-N-N-N	**rap1 protein***	

corpus de nature différente. Un des points faibles de l'évaluation que nous avons réalisé est qu'elle est fondée sur un seul corpus d'un domaine unique, donc l'objectif des travaux à venir est aussi étendre ces résultats à d'autres langues et domaines.

Le modèle et les attributs de filtrage que nous avons discutés sont totalement indépendants de la langue et du domaine. Ainsi, seules les étapes de prétraitement et de sélection de motifs doivent être adaptées à un nouveau contexte d'application. Il sera donc possible d'adapter facilement les outils de TAL pour traiter le lexique spécialisé des différents domaines, de manière à aider à la diffusion des connaissances.

APÊNDICE B RESUMO ESTENDIDO

EXTRAÇÃO DE TERMOS MULTI-PALAVRAS PARA DOCUMENTOS ESPECIALIZADOS

B.1 Introdução

O objetivo maior do Processamento de Linguagem Natural (PLN) é tornar a comunicação entre humanos e computadores tão simples quanto pedir um favor a um amigo. A língua natural é um componente essencial do comportamento inteligente, levando à classificação do PLN como sub-área da inteligência artificial, área da qual o PLN herda muitos dos métodos e princípios. A língua humana, por uma série de fatores, é complexa para a modelagem computacional. Uma fração importante do conhecimento humano é expresso através da linguagem natural: a língua é meio de intercâmbio de conhecimentos e serve para moldar as bases da ciência. Conseqüentemente, sistemas de PLN têm grande interesse na chamada linguagem especializada ou técnico-científica. Uma das características desse tipo de linguagem é seu léxico, rico em estruturas terminológicas, que são o foco desse trabalho. Nesse contexto, acredita-se que a terminologia possa ser adquirida automaticamente através de corpora de domínio específico e técnicas de aprendizado dirigido pelos dados.

Suponhamos um usuário que necessita encontrar informações sobre tradução automática baseada em exemplos. Usando um mecanismo de busca na Web, essa tarefa pode ser tediosa pois, não somente a maioria dos resultados estará em inglês, mas provavelmente será também difícil filtrar a grande quantidade de conteúdo (irrelevante) gerada por usuários. Mecanismos de busca especializados precisam levar em conta a terminologia do domínio, por exemplo, os termos *árvore* e *folha* possuem estatuto terminológico, mas os conceitos que representam não são os mesmos em botânica e em informática. Em resumo, com a tecnologia atualmente disponível, o usuário que exprime uma necessidade de informação específica a um domínio usando sua língua mãe precisa interrogar diversas fontes externas para traduzir as palavras-chave e os documentos retornados na busca.

Não só na Web, mas de maneira geral, o acesso multi-língüe a textos especializados é de grande interesse, pois possui aplicações em todas as áreas da difusão da ciência e também dos produtos. O estudo de métodos de recuperação de informações multi-língüe não cabe nesse trabalho, porém o exemplo serve de motivação para a problemática da integração dos termos do domínio com a tradução de máquina.

Primeiramente, é necessário definir três conceitos fundamentais ao trabalho: a expressão multi-palavras, o termo técnico-científico e o termo multi-palavras. A definição de Expressão Multi-Palavras (EMP) é ampla, pois o termo engloba diversos fenômenos distintos como compostos nominais (*traffic light*, *feu rouge*), expressões idiomáticas (*sem pé nem cabeça*, *cloud number nine*), termos compostos (*sistema operacional*, *Datenbanksystem*). Os termos são, ao contrário das EMPs de propósito geral, fenômenos lingüísticos ligados ao texto técnico-científico. Termos podem ser unidades lexicais únicas, enquanto EMPs são necessariamente compostas de mais de uma palavra. Seguindo a linha da Teoria Geral da Terminologia, acredita-se que existe uma relação 1:1 entre termos e conceitos. Essa concepção, no entanto, foi bastante criticada por ser reducionista e não levar em conta o termo no seu habitat natural: o texto. Nesse trabalho, assume-se que termos são unidades monossêmicas e independentes de contexto. Os Termos Multi-Palavras (TMP) são definidos como locuções que possuem estatuto terminológico. Destaca-se que os mesmos não correspondem ao conceito de fraseologia do domínio. O primeiro aceita pouca variabilidade (morfológica, raramente sintática) enquanto o segundo é altamente flexível. Enquanto o primeiro representa um único conceito, não é raro que o segundo seja uma estrutura complexa que associe mais de um conceito.

A Terminologia e a Tradução possuem diversos pontos em comum no que diz respeito à tradução especializada. Da mesma forma, TMPs e Tradução Automática (TA) deveriam ser áreas bastante ligadas, apesar de que a realidade atual em PLN reflete pouco essa interdependência. Por isso, um dos objetivos desse trabalho é a detecção de termos para a adaptação de recursos lingüístico-computacionais a diferentes domínios. Escolheu-se trabalhar apenas com termos multi-palavras por (a) os termos multi-palavras serem representativos de cerca de 70% da terminologia, (b) os métodos computacionais de detecção de termos multi-palavras e mono-palavras serem bastante diferentes e (c) os TMPs representarem um grande desafio em lingüística computacional.

Definição B.1 *Uma Expressões Multi-Palavras (EMP) é um conjunto de duas ou mais palavras com semântica não-composicional, ou seja, o sentido do sintagma não pode ser compreendido totalmente através do sentido de suas componentes (Sag et al. 2002).*

Definição B.2 *Um termo é uma unidade lexical ou multi-lexical com significado não ambíguo quando empregada em textos especializados, ou seja, a terminologia de um domínio é a representação lingüística dos seus conceitos (Krieger and Finatto 2004).*

Definição B.3 *Um Termo Multi-Palavras (TMP) é um termo composto por mais de uma palavra. (SanJuan et al. 2005, Frantzi et al. 2000).*

B.2 Trabalhos relacionados

O trabalho aqui apresentado aborda a integração de três sub-áreas de PLN: expressões multi-palavras, terminologia e tradução estatística. EMPs constituem um desafio para PLN por apresentarem alta variabilidade morfossintática e natureza heterogênea. Técnicas para tratar EMPs são classificadas em "palavras-com-espaços" ou composicionais (Sag et al. 2002). Quatro tarefas estão ligadas ao tratamento dessas estruturas: identificação de ocorrências em textos, interpretação da estrutura (sintática), classificação/agrupamento (semânticos) e aplicação em outros sistemas de PLN. É possível identificar candidatos a EMP através de padrões morfológicos rasos (Villavicencio et al. 2007, Manning and Schütze 1999) ou de sintaxe (profunda) (Seretan 2008, Baldwin 2005). Já a filtragem dos candidatos pode ser feita através de medidas estatísticas (Evert and Krenn 2005, Villavicencio et al. 2007), possivelmente complementadas por freqüências advindas da Web (Keller et al. 2002, Zhang et al. 2006). Abordagens simbólicas empregam geralmente tesauros e dicionários de sinônimos (Pearce 2001). Adicionalmente, aprendizado de máquina supervisionado é por vezes usado para criar modelos de extração de EMPs (Pecina 2008). A interpretação da estrutura interna das EMPs depende de seu tipo específico, por exemplo, Ramisch et al. (2008) classificam verbos frasais com relação à idiomaticidade ao passo que Nakov and Hearst (2005) tratam da estrutura de aninhamento de nomes compostos. Com relação à classificação de EMPs, cita-se o trabalho de Lapata (2002), que investiga em particular nominalizações.

Uma técnica comumente usada para a extração de termos é a utilização de filtros morfológicos frequentes (Justeson and Katz 1995). Frantzi et al. (2000) propõem uma técnica mais apurada para a classificação de candidatos, enquanto Smadja (1993) usa a informação mútua das palavras para detectar comportamento colocacional. Considerando apenas termos da área biomédica no corpus Genia, SanJuan et al. (2005) realizam uma classificação semântica automática e a comparam com a ontologia do domínio.

A Tradução Automática (TA) tem evoluído nos últimos anos para métodos colaborativos ou estatísticos. O primeiro é baseado na tripla automaticidade, cobertura e precisão, deixando o usuário escolher qual dos aspectos será favorecido (Bey et al. 2006, Boitet et al. 2008). Já algoritmos estatísticos aprendem probabilidades de tradução a partir de um corpus paralelo (Brown et al. 1993). Os alinhamentos de palavras são obtidos através do algoritmo EM de aprendizado não-supervisionado. Sistemas podem ser baseados em palavras mas também em sintagmas (seqüências de palavras), como descrito em Koehn et al. (2003). Em domínios especializados, sistemas de PLN manifestam problemas quando o texto de treinamento difere do texto de entrada, por exemplo a falta de entradas no léxico de um sistema de análise sintática descrito por Baldwin et al. (2004). A extração de EMPs através de alinhamentos palavra-a-palavra fornece meios de extrair ao mesmo tempo unidades léxicas multi-palavras e suas respectivas traduções (Melamed 1997, Caseli et al. 2009).

B.3 Extração terminológica na biomedicina

O corpus Genia é um conjunto de 2.000 resumos da área biomédica sobre as palavras-chave “humano”, “célula sangüínea” e “fator de transcrição” (Ohta et al. 2002). Ele contém 18,5K frases e 490,7K tokens, anotados com classe morfossintática e também terminologia como nomes de doenças (*tumor fibroblástico*) e células (*linfócito t primário*). Em média, as frases contém 3 TMPs, cada um com em média 3 tokens, mostrando o quão onipresente é a terminologia do domínio no corpus. Os termos do corpus Genia tendem a ser profundamente aninhados. Os experimentos descritos a seguir são uma série de scripts desenvolvidos em Python, awk e bash e disponíveis em `www.inf.ufrgs.br/~ceramisch/tcc.zip`.

Primeiramente, os dados do corpus foram pré-processados (SanJuan et al. 2005). Todas as sentenças foram convertidas para minúsculas e uma versão XML simplificada do corpus foi gerada. Nessa versão, a anotação de termos foi mantida, não para ser usada na extração, mas para servir de base de comparação durante a avaliação. Um *candidato* a TMP é extraído do corpus se obedecer a certos padrões e regras pré-estabelecidos, independentemente de anotação. Já a *terminologia padrão* é, ao contrário, o conjunto de termos anotados que servirão como base de comparação e avaliação automática. A terminologia padrão do Genia possui 28.340 TMP distintos. Considere o exemplo abaixo (classe da palavra indicada por “/”, termos entre colchetes):

Exemplo B.1 `[ [ alpha/N a/UKN ] -gene/A product/N ]`

Veja que a anotação de terminologia quebra a palavra em hífens e barras. Isso é prejudicial pois a meia palavra acaba ficando com classe morfossintática desconhecida (UKN). Como tal problema ocorre em 17% das frases, decidiu-se incluir o token externo na expressão para melhorar a anotação. O resultado é como o exemplo abaixo. Com essa modificação, apenas 81 frases foram descartadas:

Exemplo B.2 `[ [ alpha/N a-gene/A ] product/N ]`

Para tratar inflexão de número, é necessário converter cada realização de palavra à sua forma singular. Um lematizador de propósito geral poderia converter para *bind* o substantivo *binding*, enquanto uma regra simples de eliminação de “s” no fim das palavras levaria a lemas do tipo *virus* → *viru** ou *viruses* → *viruse**. Por isso, foi criado um lematizador simples baseado em regras e em freqüências da Web, cujo funcionamento é descrito no anexo D.

A última etapa de pré-processamento diz respeito à característica dos textos biomédicos, que tendem a inserir acrônimos entre parênteses após um termo recorrente. Quando esses parênteses ocorrem no meio de um TMP, introduzem ruído para os filtros, como no exemplo *[[granulocyte-macrophage colony-stimulating factor] (gm-csf) promoter]*. Aproximadamente 3% dos termos possuem parênteses. Para remover apenas aqueles que contém acrônimos, usou-se um algoritmo que mistura regras e correspondência difusa por sub-cadeia máxima, descrito no anexo D. Dessa forma, é possível detectar acrônimos no corpus e então remover somente aqueles que se encontram entre parênteses. Essa lista de acrônimos (e.g. *AD – alzheimer 's disease*, *5LOX – 5 lipoxygenase*, *AD – adenovirus*) é um produto colateral da extração de TMPs.

Uma vez que o corpus foi uniformemente pré-processado, é necessário proceder à extração de candidatos a TMP. Para isso, antes de mais nada, separou-se duas partes: *Genia-test* (100 resumos, 895 frases) e *Genia-train* (restante do corpus). A primeira foi posta de lado para utilização futura durante a etapa de avaliação. A última parte foi usada para realizar o aprendizado, não só dos padrões morfológicos, mas também dos pesos dados a cada medida de associação durante a etapa de filtragem, posterior à extração, de candidatos.

A seleção dos padrões morfológicos foi feita da seguinte forma: (1) com base na anotação de termos, extraiu-se a o conjunto de padrões morfológicos da terminologia padrão de *Genia-train*, (2) selecionou-se os 118 padrões com ocorrência maior do que dez, (3) dentre os últimos, manteve-se apenas os 57 com precisão (proporção de candidatos positivos) superior a 30% e (4) extraiu-se do corpus todos os n-gramas que obedeciam a esses padrões. Os valores de freqüência de corte e de precisão mínima foram obtidos com base em observações empíricas dos dados. Os padrões selecionados incluem seqüências de nomes e adjetivos (*N-N*, *A-N*, *A-N-N*), mas também palavras estrangeiras (*FW-FW*) e gerúndios (*N-V-N*). O resultado dessa etapa é uma lista com 60.585 candidatos a TMP que obedecem aos padrões selecionados, dos quais 22.507 são positivos (31,52%).

Tendo extraído os candidatos, procede-se então à sua filtragem. Os filtros são baseados em Medidas de Associação (MA) estatísticas (Evert and Krenn 2005, Ramisch et al. 2008, Pecina 2008). A distribuição de probabilidades do vocabulário da língua é zipfiana, com um fenômeno conhecido como "grande cauda". Conseqüentemente, os dados de freqüência tornam-se esparsos e as MA apresentam pouca confiabilidade. Para amenizar esse efeito, usou-se também as freqüências obtidas na World Wide Web (através da API de busca de *Yahoo!*), que consideramos como um corpus de língua geral doravante denominado *Web*. Para estimar o tamanho N do corpus usou-se uma estimativa grosseira[1], que é simplesmente um fator linear de escala igual para todos os candidatos. A lista de acrônimos é usada aqui para aumentar as freqüências das palavras que os compõem a cada ocorrência dos mesmos. Para cada candidato $w_1 \ldots w_n$ e para cada corpus de tamanho N, a freqüência simples $f(w_1 \ldots w_n)$ e as freqüências marginais $f(w_1) \ldots f(w_n)$ são usadas para calcular as MA abaixo:

$$\texttt{prob} = \frac{f(w_1 \ldots w_n)}{N} \qquad \texttt{PMI} = log_2 \frac{f(w_1 \ldots w_n)}{N^{n-1} f_\emptyset(w_1 \ldots w_n)}$$

$$\texttt{t} = \frac{f(w_1 \ldots w_n) - N^{n-1} f_\emptyset(w_1 \ldots w_n)}{\sqrt{f(w_1 \ldots w_n)}} \qquad \texttt{Dice} = \frac{n * f(w_1 \ldots w_n)}{\sum_{i=1}^{n} f(w_i)}$$

Aqui, a hipótese nula a rejeitar H_0 supõe independência entre as palavras, ou seja, elas co-ocorrem por acaso e sua freqüência é $f_\emptyset(w_1 \ldots w_n) = (f(w_1) \ldots f(w_n))/N^{n-1}$. Medidas baseadas em tabela de contingência, apesar de mais robustas, não se aplicam porque, no caso de n arbitrário, não só é difícil de intuir um hipercubo para as medidas, como também o ruído e as incoerências (no corpus *Web*) impedem, na prática, sua aplicação. Diversos trabalhos discutem a escolha da melhor medida,

[1] 50 bilhões de páginas, segundo `http://www.worldwidewebsize.com/`

Tabela B.1: Avaliação da normalização e Eliminação de Acrônimos (EA).

	Sem norm.	**Sem EA**	**Com norm. e EA**
TMPs na terminologia padrão	29.513	28.601	28.340
Candidatos a TMP	66.817	61.401	63.210
Precisão	15,56%	37,53%	37,24%
Cobertura	35,22%	80,58%	83,06%
F-measure	21,58%	51,20%	51,42%
Freqüência média no top-100	189,98	205,05	205,38
Hapax legomena	51.169 (77%)	46.330 (75%)	47.770 (76%)

porém o método aqui proposto consiste em usar aprendizado de máquina para essa tarefa, separando os candidatos filtrados em duas classes: `MWT` e `NonMWT`. Os algoritmos usados incluem regras de árvore de decisão (J48), aprendizado estatístico de classificadores bayesianos, treinamento de redes neurais artificiais (MLP e *Voted Perceptron*) e classificadores numéricos que atuam sobre a fronteira de decisão, i.e. máquinas de vetor de suporte (SVM).

Uma vez extraídos os termos multi-palavras do corpus, sugere-se maneiras de integrá-los a um mecanismo de TA estatístico. O sistema escolhido, chamado Moses, foi treinado sobre o corpus Europarl com o par de línguas inglês → português, para o qual se tem falantes nativos dispostos a avaliar os resultados. O corpus Europarl foi alinhado no nível das frases, filtrado, homogeneizado na separação de palavras e na caixa, e então separado em três sub-conjuntos para desenvolvimento, teste e treinamento. Sobre o último, gerou-se um modelo de linguagem, os alinhamentos palavra-a-palavra e a tabela de tradução para, em seguida, treinar por minimização da taxa de erro os pesos de cada atributo do modelo.

Existem diversos métodos propostos para unir e separar componentes em alemão ou sueco (Nießen and Ney 2004, Nießen et al. 2000). Usou-se uma técnica semelhante para os termos em inglês: considera-se como exemplo a frase de entrada *ras-related gtp-binding proteins and leukocyte signal transduction*. A primeira abordagem (wws) consiste em concatenar as palavras formando um termo, por exemplo *ras-related#gtp-binding#proteins and leukocyte signal#transduction*. A segunda abordagem (head) consiste em substituir o termo pelo seu núcleo, inferido como sendo o último substantivo do n-grama, por exemplo *proteins and leukocyte transduction*. A terminologia padrão também foi usada para avaliar a técnica de união e separação de termos no Moses. A substituição de núcleo é uma forma de simplificação do texto original que visa facilitar o trabalho do sistema de TA, de forma que o texto original implica o texto simplificado, que é, por sua vez, traduzido. Semanticamente, a informação transmitida pela tradução contém o significado da frase original (mas não é equivalente).

B.4 Avaliação

A primeira avaliação concerne o pré-processamento e é, portanto, realizada sobre a totalidade do corpus. Referir-se-á ao processo de lematização e homogeneização da separação de palavras como *normalização*, etapa que é seguida da Eliminação de Acrônimos (EA). A avaliação manual de um conjunto de palavras pelo autor mostra que o algoritmo de extração de plurais tem precisão de 99%; na tabela B.1 verifica-se que, com relação à configuração sem normalização, a terminologia padrão é 3,89% melhor, enquanto a lista de candidatos é 5,35% menor. Isso mostra que candidatos e termos similares (por exemplo, onde um é o plural do outro) passam a ser tratados como uma só entidade, representantes do mesmo conceito. Apesar de reducionista (contexto não é levado em consideração) essa abordagem tem resultados práticos interessantes: a cobertura aumenta consideravelmente com o uso da normalização, de 35% para 83%; a precisão dobra, ou seja passa de 15% a 37%. Além disso, as freqüências das palavras são menos esparsas, pois candidatos semelhantes são agregados, e suas freqüências, somadas. O número de hapax legomena (ocorrência única) diminui, porém ainda corresponde a dois terços da lista de candidatos. Outras heurísticas de normalização podem ser usadas, como inflexão de gênero ou de verbo, porém é muito importante lembrar que qualquer processamento desse tipo é altamente dependente da linguagem e do corpus em questão. Ainda na mesma tabela, vê-se a configuração sem EA. Aqui a terminologia padrão também diminui de tamanho (300), porém menos do que esperado (a lista de acrônimos possui 1.300 entradas), provavelmente porque a maioria dos últimos ocorre após e não no meio dos TMPs. Extrai-se mais candidatos, aumentando a cobertura em 2,5%, porém a precisão é ligeiramente menor. Tem-se a impressão de que a EA ajuda a gerar freqüências ligeiramente menos esparsas. Também usando os acrônimos para inflar as freqüências das palavras que os compõem, a freqüência marginal média aumenta de 37 para 47 unidades. No entanto, é questionável se esse procedimento é essencial, dado que as melhorias são pequenas sobre o resultado final.

Tabela B.2: Precisão e cobertura dos padrões de TMP.

Padrões morfológicos	**Precisão**	**Cobertura**
J&K	**29,58%**	66,71%
$f > 10$	16,50%	**92,57%**
$f > 10, p > .3$	31,52%	89,43%

Concluído o pré-processamento, quer-se avaliar a etapa de seleção de padrões morfológicos dos candidatos. Quando se compara os padrões usados por Justeson and Katz (1995) (J&K), os com freqüência maior que 10 e aqueles que foram selecionados ($f > 10, p > .3$), percebe-se que J&K possui boa precisão, porém cobertura muito baixa sobre esse corpus (onde os TMPs tendem a ser longos e aninhados). Já os padrões mais freqüentes possuem melhor cobertura com precisão pouco superior à metade da obtida com J&K. Por isso mesmo é que a lista de padrões selecionados oferece bom equilíbrio entre ambos, com a maior precisão dentre os conjuntos comparados e uma cobertura ligeiramente inferior à de $f > 10$. Dentre os padrões selecionados, destaca-se (a) problemas com relação à presença de conjunções, como em *young and aged subject* quando, na verdade, esse TMP corresponde a dois termos independentes, (b) a presença de padrões contendo numerais (*interleukin 2*) e (c) a ausência do padrão N-P-N, que é considerado por J&K como bom indicador de TMP. É interessante verificar que, quando a terminologia padrão do conjunto de treinamento é usada para avaliar o conjunto de teste, os padrões reconhecem cerca de duas vezes mais termos do que a anotação existente. Dessa forma, a cobertura chega a 200%, enquanto os valores de precisão oscilam entre 65% e 95%, mostrando que uma anotação mais extensa poderia melhorar os resultados.

A tabela B.3 mostra a precisão média de cada uma das medidas de associação bem como a precisão de base (proporção inicial de candidatos positivos) para os fragmentos de treinamento e de teste. De acordo com os valores apresentados, apesar de assumir palavras com distribuição normal, o teste

Tabela B.3: Precisão média (%) das MA estatísticas.

	Genia				*Web*				
	prob	*PMI*	*t*	*Dice*	*prob*	*PMI*	*t*	*Dice*	**Base**
train	46,16	40,08	52,90	38,84	35,77	41,02	45,13	37,85	37,15
test	38,83	17,51	31,36	22,78	41,49	20,52	43,21	43,74	24,16

Tabela B.4: Desempenho dos algoritmos de aprendizado.

	J48	Bayes(TAN)	Bayes(SA)	MLP	VP	SVM(pol.)	SVM(rad.)
Precisão	47.9%	48.6%	52.4%	27.3%	36.3%	34.9%	**52.6%**
Cobertura	19.7%	32.1%	28.9%	**0.5%**	**63.8%**	60.1%	**35.4%**

t possui boa acurácia em ambos os corpora, como é também mostrado por Evert and Krenn (2005), Manning and Schütze (1999). A probabilidade simples (prob) também parece ser um bom indicador de TMPs. A pior das medidas parece ser PMI, cuja precisão média é inferior à linha de base para *Genia-test*, ao contrário do que acredita Smadja (1993). No entanto, Evert and Krenn (2005) chegam à mesma conclusão, afirmando ainda que *t* e a verossimilhança logarítmica (baseada em tabela de contingência) costumam dar bons resultados. Um algoritmo simples de seleção de atributos confirma esses resultados, dizendo que as medidas mais relevantes são $prob_{Genia}$, t_{Genia} e t_{Web}. Algumas incoerências foram observadas no comportamento das MAs sobre o corpus *Web*, pois as medidas parecem ser ruins em *Genia-train*, mas boas em *Genia-test*.

A análise do vocabulário de cada um dos corpora (figura B.1) mostra que existe uma possível razão para um comportamento arbitrário das medidas da *Web*. A API de *Yahoo!* retorna para as 735 palavras mais freqüentes um número constante de ocorrências (2.147.483.647), de forma que torna-se impossível diferenciar palavras como *the* e *green* (que aparecem respectivamente 22.446 e 1 vezes no corpus Genia). Essa situação patológica, fruto de uma aproximação grosseira do mecanismo de busca, gera a forma “truncada” do gráfico e certamente prejudica as medidas de associação. Dessa forma, a correlação de Pearson entre os corpora é de 0,12, enquanto o coeficiente τ de Kendall é de 0,21, contrariando o que dizem Zhang et al. (2006). Trabalhos futuros devem incluir maneiras de aproximar as freqüências com base nos pontos fortes dos dois corpora, visto que Genia tem dificuldades com eventos esparsos, enquanto *Web* não consegue diferenciar palavras muito freqüentes.

O desempenho dos algoritmos de aprendizado de máquina sobre *Genia-test* pode ser visto na tabela B.4. Apenas a acurácia da classe `MWT` é mostrada, pois, como os dados não são balanceados, a classe `NonMWT` possui acurácia superior a 80% para todos os algoritmos. A árvore de decisão J48 possui um modelo de dados facilmente interpretável (árvore com 484 nodos) que classifica corretamente 75% das instâncias. Os atributos mais próximos da raiz (mais importantes) são os testes *t* nos dois corpora, enquanto o menos importante parece ser o padrão morfológico, geralmente nas folhas da árvore. O problema desse e de outros algoritmos é considerar que ambas as classes possuem o

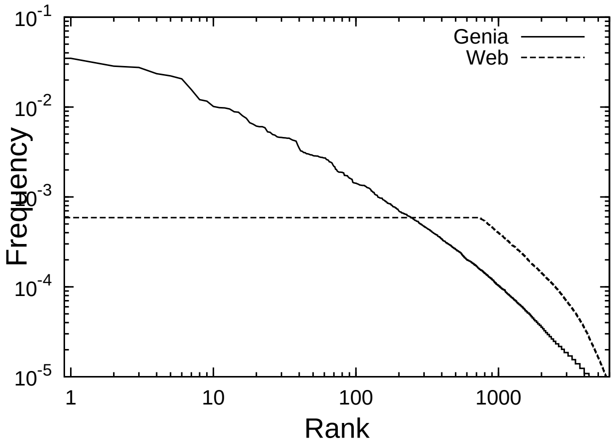


Figura B.1: Histograma do vocabulário de *Genia-train* e da *Web*.

Tabela B.5: Avaliação comparativa de SVM com diferentes limiares e algoritmo Xtract.

	Genia-test				*Genia-train*
	Sem limiar	$f_{Genia} > 1$	$f_{Genia} > 5$	**Xtract**	**Xtract**
n.º TMP extraídos	763	739	174	70	1.558
Precisão	52,56%	56,83%	74,14%	70%	66,81%
Cobertura	19,96%	20,91%	6,42%	2,44%	3,84%

mesmo peso, levando a um baixo desempenho sobre a classe `MWT`. Nesse sentido, o aprendizado bayesiano é mais eficiente, sobretudo com a busca por árvore de dispersão mínima (TAN) ao invés de arrefecimento simulado (SA). A cobertura de ambos ainda é decepcionante e os métodos sofrem do mesmo problema que J48, isto é, dão igual importância a ambas as classes. A rede neural multicamadas (MLP) com 33 neurônios ocultos tem um desempenho muito ruim: captura apenas 5% dos candidatos positivos. O algoritmo *Voted Perceptron (VP)* tem desempenho surpreendente, com larga cobertura dos candidatos positivos. As razões para tal resultado serão investigadas em trabalhos futuros. Os melhores resultados foram obtidos com algoritmos de máquinas de vetor de suporte (SVM) com núcleo polinomial e radial. Cada um deles privilegia um dos aspectos: o primeiro tem boa cobertura, enquanto o último tem uma precisão de 52,6%. A escolha de um deles depende da aplicação, por exemplo a construção de um dicionário sendo custosa, pode-se privilegiar a precisão para diminuir o esforço manual. Uma aplicação de PLN, no entanto, pode conviver com ruído desde que a cobertura seja suficientemente ampla. Para as próximas comparações, usou-se o SVM com núcleo radial, que retorna sobre o conjunto de teste 763 instâncias positivas das quais 362 são falsas. Se o SVM não fosse aplicado, apenas 24% das instâncias seriam verdadeiras positivas.

Não obstante os valores absolutos de precisão e principalmente de cobertura parecerem baixos, é comparando-os com o método de extração de unidades multi-palavras Xtract que se observa o quanto o método sugerido é superior. Na tabela B.5, compara-se o desempenho do SVM usando diversos limiares de freqüência sobre os candidatos, de forma a aumentar a precisão e diminuir a cobertura. A configuração que elimina os hapax legomena ($f_{Genia} > 1$) gera cobertura e precisão superiores, mostrando que esses candidatos devem com efeito ser removidos da lista. No entanto, com um limiar alto ($f_{Genia} > 5$) a cobertura cai drasticamente. O método por SVMs utiliza as informações de *Genia-train* para extrair TMPs de *Genia-test*. Por isso, compara-se não somente os SVMs com Xtract sobre o conjunto de teste mas também com Xtract sobre o conjunto de treinamento, maior e portanto com freqüências mais confiáveis e menos esparsas. No entanto, em ambos os casos Xtract parece ser muito conservador, gerando uma lista de candidatos com alta precisão e cobertura extremamente baixa. Observa-se que, mesmo na configuração mais restritiva, o SVM obtém um desempenho global superior ao método Xtract, provando que se obteve com esse trabalho real ganho sobre um algoritmo padrão de extração de TMPs.

Finalmente, por razões pragmáticas de tempo de processamento e espaço de armazenamento disponíveis, não foi possível concluir a avaliação da integração TMP-TA. A idéia, para essa etapa, é que seja fornecida aos avaliadores (falantes nativos) uma lista com as traduções geradas por diferentes métodos (wws e head com TMPs extraídos e terminologia padrão). As traduções devem, então, ser ordenadas pela sua fluidez, visto que sua adequação será certamente inferior, já que o método realiza a simplificação da entrada. Uma interface que possibilite a fácil visualização dos termos originais e a colaboração na sua tradução também será criada e submetida a técnicas padronizadas de avaliação de interfaces.

Tabela B.6: Exemplos de TMPs extraídos de *Genia-test*(asteriscos indicam erros).

TMP extraído	Padrão	TMP extraído	Padrão
t cell	N-N	***thromboxane receptor**	N-N
nf kappa b	N-N	peripheral blood	A-N
kappa b	N-N	nf kappa b	N-N-N
cell line	N-N	human t	A-N
transcription factor	N-N	receptor alpha	N-N
i kappa b	N-N-N	monocytic cell	A-N
***i kappa**	N-N	binding activity	N-N
***kappa b alpha**	N-N-N	stat protein	N-N
***b alpha**	N-N	***receptor gene**	N-N
i kappa b alpha	N-N-N-N	***rap1 protein**	

B.5 Conclusões

O trabalho aqui apresentado mostra uma avaliação sistemática de técnicas de extração terminológica sobre um corpus biomédico. Primeiro, discutiu-se hipóteses e noções básicas de expressões e termos multi-palavras, bem como sua caracterização por teorias lingüísticas. Diversos trabalhos relacionados abordam línguas e tipos específicos de expressões multi-palavras. Algumas tarefas e métodos foram discutidos, sendo que a maioria deles é baseada em bigramas, no que se acredita que o trabalho aqui apresentado é superior pois é capaz de extrair TMPs com tamanho *n* qualquer. Resumidamente, o método que é introduzido nesse trabalho consiste em quatro etapas: pré-processamento, extração de candidatos através de padrões morfológicos selecionados, filtragem de candidatos por medidas de associação juntamente com aprendizado de máquina e, por último, integração com sistemas de TA estatísticos. A tabela B.6 mostra algumas das expressões extraídas com o SVM e $n_{Genia} > 5$. Elas são em sua maioria nomes compostos, com predominância de bigramas. Os exemplos negativos parecem fazer parte de ou conter uma expressão: provavelmente uma avaliação mais fina com classes como `Contains` e `PartOf` poderia verificar essa hipótese.

É importante sublinhar o fato de que a lista de termos extraída não vem, de maneira alguma, a substituir ou melhorar a anotação previamente existente no corpus. O objetivo desse trabalho não é gerar uma nova terminologia para o corpus Genia, pois este já possui tal informação. Ao contrário, essa anotação de termos foi usada para alcançar o objetivo inicial de avaliar (automaticamente) métodos de extração de multi-palavra sobre um domínio específico.

Idéias de trabalhos futuros incluem, evidentemente, a conclusão da avaliação sugerida para a integração entre terminologia e tradução automática, a fim de estabelecer a validade da metodologia aqui proposta. Além disso, quer-se investigar novas formas de combinar freqüências de diversos corpora. Um dos pontos fracos da avaliação é que ela é totalmente baseada em um só corpus, portanto pretende-se também expandi-la para outros domínios e/ou línguas.

Considera-se que o modelo e os atributos de filtragem aqui discutidos são totalmente independentes de língua e de domínio. Conseqüentemente, apenas as etapas de pré-processamento e seleção de padrões devam ser adaptadas a um novo contexto de aplicação. Dessa forma, poder-se-á adaptar facilmente ferramentas de PLN para processar o léxico especializado dos diferentes domínios, auxiliando na difusão do conhecimento.

APPENDIX C PART-OF-SPEECH PATTERNS

Table C.1: Complete list of the 57 POS-patterns used to extract MWT candidates from the Genia corpus, with occurrence examples.

Pattern	P(p)	R(p)	$\sum R(p)$	Example
N-N	42%	22%	22%	*zinc sulfate*
A-N	31%	19%	41%	*malignant tissue*
N-N-N	43%	10%	51%	*zinc finger type*
A-N-N	40%	9%	60%	*lymphoblastoid cell line*
A-A-N	32%	4%	64%	*xenoreactive natural antibody*
A-N-N-N	38%	3%	67%	*cyclic adenosine monophosphate signaling*
N-N-N-N	39%	3%	70%	*dna binding protein complex*
N-A-N	36%	2%	72%	*colostrum inhibitory factor*
A-A-N-N	39%	2%	74%	*fetal nucleated blood cell*
N-A-N-N	38%	1%	74%	*kappa light chain gene*
N-NUM	32%	1%	75%	*nucleotide 46*
A-N-N-N-N	41%	1%	76%	*whole cell dexamethasone binding assay*
A-CC-A-N	30%	1%	77%	*young and aged subject*
N-N-N-N-N	36%	1%	77%	*lymphocyte glucocorticoid receptor binding parameter*
N-N-A-N	35%	1%	78%	*laser scanning confocal microscopy*
A-A-A-N	35%	0%	78%	*human pulmonary alveolar macrophage*
A-A-N-N-N	40%	0%	78%	*human peripheral blood mononuclear cell*
A-N-A-N	30%	0%	79%	*hematopoietic cell specific molecule*
N-A-N-N-N	42%	0%	79%	*immunoglobulin heavy chain gene cluster*
N-A-A-N	36%	0%	79%	*vzv immune human donor*
N-N-NUM-N	33%	0%	80%	*t beta 5 element*
A-CC-A-N-N	38%	0%	80%	*thymic and peripheral t lymphocyte*
N-NUM-N-N	31%	0%	80%	*zeta 2 globin promoter*
A-V-N	31%	0%	80%	*specific binding complex*
FW-FW	73%	0%	80%	*treponema pallidum*
N-N-A-N-N	36%	0%	81%	*glucocorticoid receptor positive cell population*
A-N-A-N-N	36%	0%	81%	*germinal center follicular b cell*
A-N-N-N-N-N	39%	0%	81%	*human hepatitis b virus x protein*
A-A-A-N-N	33%	0%	81%	*human myeloid nuclear differentiation antigen*
N-A-A-N-N	40%	0%	81%	*ig variable heavy chain sequence*
N-N-N-A-N	37%	0%	81%	*jurkat t cell complementary dna*
A-N-N-A-N	41%	0%	81%	*nuclear factor kappab inducing kinase*
FW-FW-N-N	44%	0%	82%	*in vivo footprint analysis*
N-N-N-N-N-N	38%	0%	82%	*reverse transcription polymerase chain reaction assay*
A-A-N-A-N	44%	0%	82%	*unfractionated peripheral blood mononuclear cell*
A-CC-N-N	33%	0%	82%	*biochemical and mutagenesis analysis*

Table C.1: Complete list of the 57 POS-patterns used to extract MWT candidates from the Genia corpus, with occurrence examples.

Pattern	P(p)	R(p)	$\sum R(p)$	Example
A-N-N-NUM-N	45%	0%	82%	*early growth response 1 gene*
A-N-NUM-N	31%	0%	82%	*human chr 20 centromere*
N-NUM-P-NUM	38%	0%	82%	*nucleotide +13 to +15*
N-N-NUM-N-N	33%	0%	82%	*rprl footprint ii repressor site*
A-A-N-N-N-N	33%	0%	82%	*acute lymphoblastic leukemia cell line reh*
A-N-N-N-NUM-N	66%	0%	82%	*human immunodeficiency virus type 1 enhancer*
N-NUM-N-N-N	46%	0%	82%	*interleukin 2 receptor alpha subunit*
A-N-N-N-N-N-N	73%	0%	82%	*nuclear factor kappa b transcription factor family*
A-CC-N-N-N	56%	0%	83%	*human and mouse intraspecy hybrid*
A-N-A-A-N	34%	0%	83%	*human t acute lymphoblastic leukemia*
N-N-N-N-NUM	36%	0%	83%	*igg fc receptor type i*
A-NUM-N-N	42%	0%	83%	*last 13 amino acid*
A-V-N-N	37%	0%	83%	*normal activated b cell*
N-N-A-N-N-N	42%	0%	83%	*cell cycle regulated histone h4 gene*
N-A-N-N-N-N	38%	0%	83%	*adipocyte fatty acid binding protein gene*
A-N-NUM-N-N	48%	0%	83%	*human class ii transplantation gene*
A-N-N-N-NUM	48%	0%	83%	*vascular cell adhesion molecule 1*
N-A-N-NUM	39%	0%	83%	*p130 phosphorylated form 3*
A-A-A-N-N-N	40%	0%	83%	*human acute lymphoblastic t cell line*
A-CC-A-N-N-N	44%	0%	83%	*murine and human tumor cell line*
A-N-N-N-N-NUM	85%	0%	83%	*monocytic cell line mono mac 6*

Table C.2: Simplication rules used to convert the Genia POS tag set to a set of customised generic tags.

Simple tag	Genia tags	Morphosyntactic category
N	NNPS, NNP, NNS, NN, NPS, NP	Nouns
A	JJR, JJS, JJ	Adjectives
V	VBD, VBG, VBN, VBP, VBZ, VVD, VVG, VVN, VVP, VVZ, VHD, VHG, VHN, VHP, VHZ, VV, VB, VH, MD	Verbs
R	RBR, RBS, WRB, RB	Adverbs
P	IN, TO, RP, EX	Prepositions and particles
DT	PDT, WDT, DT, CT, XT	Determiners
PP	PRP, PRP$, PP, PP$, WP, WP$, POS	Pronouns
CC	CC, CCS	Conjunctions
PCT	No annotation	General punctuation
FW	FW	Foreign word
UKN	*	Unknown tag
NUM	CD	Numbers

APPENDIX D PRE-PROCESSING ALGORITHMS

Algorithm 1 Heuristics to detect and acronyms between parentheses.

```
procedure ACRONYM($a, C$)
Require: $a$ acronym candidate, $C$ sequence of context words preceding $(a)$
Require: $POS(a) \neq$ NUM, $2 \leq len(a) \leq 8$
    NORMALISE(a)                                  ▷ Eliminate hyphens, dashes, spaces, ...
    $E \leftarrow \emptyset$
    for $c \in C$ do
        if $c$ starts with initial of $a$ then
            $E \leftarrow E \cup c$                       ▷ Possible extended versions of $a$
        end if
    end for
    return $\mathrm{argmax}_{e \in E}$ WEIGHTED-LCS($e, a$)
end procedure
```

The WEIGHTED-LCS procedure calculates the weighted Longest Common Subsequence between the acronym and the extended candidate. Different positive weights are given to M to complete matches (+15) and initial matches (+13), negative weight to partial matches (−11) and larger negative weight for non-matches (−13). Symmetric negative weights are given to I and D, for insertions and deletions (−11). If all matching scores are below a threshold (−9), we ignore the acronym because no extended version was considered strong enough. The dynamic programming matrix *LCS* is filled according to the following recurrence equation:

$$\begin{aligned} LCS(i,j) = \max(\ & LCS(i-1,j-1)+M, \\ & LCS(i-1,j)+D, \\ & LCS(i,j-1)+I) \end{aligned}$$

Algorithm 2 Heuristics to find the singular form of a noun.

```
procedure SINGULAR($w$)
Require: $w$ ends with “s”, $len(w) > 2$
    $W_{sing} = \emptyset$
    if $w$ ends with “ss” or “is” or “us” then
        return $w$                                        ▷ access, analysis, virus
    else if $w$ ends with “ies” then
        $W_{sing} \leftarrow W_{sing} \cup w -$ “ies” $+$ “y”            ▷ bodies → body
    else if $w$ ends with “oes” then
        $W_{sing} \leftarrow W_{sing} \cup w -$ “oes” $+$ “o”            ▷ heroes → hero
    else if $w$ ends with “es” then
        $W_{sing} \leftarrow W_{sing} \cup w -$ “s”                    ▷ diseases → disease
        if $w$ ends with “xes” or “ches” or “shes” or “ses” then
            $W_{sing} \leftarrow W_{sing} \cup w -$ “es”               ▷ boxes, witches, dishes
        end if
        if $w$ ends with “ses” then
            $W_{sing} \leftarrow W_{sing} \cup w -$ “es” $+$ “is”      ▷ analyses → analysis
        end if
    else
        $W_{sing} \leftarrow W_{sing} \cup w -$ “s”                    ▷ receptors ← receptor
    end if
    return $\mathrm{argmax}_{w_{sing} \in W_{sing}} f_{Web}(w_{sing})$    ▷ Search the Web for $W_{sing}$
end procedure
```

APPENDIX E IMPLEMENTATION

Below, we present the output of running the `runAll.sh` script of our system. It takes the Genia corpus, transforms it to a generic format and then works on simplified XML and TAB-separated files to extract MWT candidate tables and evaluate several aspects discussed in our work. This file calls auxiliary Python and awk scripts that actually perform MWT extraction and evaluation.

```
0) Obtaining corpus...
  * File genia.xml already present!
  * File gpml.merged.dtd already present!
1) PRE-PROCESSING
  1.1) Cleaning errors...
    Raw corpus statistics
    * 2000 abstracts
    * 18519 sentences
    * 490752 tokens
    * 97876 terms
    * 55487 multiword terms
    * 5.28516658567 terms per sentence
    * 2.99622009828 multiword terms per sentence
    * 2.05631615514 words per term
    * 2.86330131382 words per multiword term
  1.2) XML format conversion...
  1.3) Simplification...
    * 3152 sentences have tagging problems (UKN tag)!
    * Simplifying tags...
    * 3071 tagging problems corrected.
    * 81 sentences discarded (tagging problems not corrected).
    * Removing redundant MWE tags: iteration 0
    * Removing redundant MWE tags: iteration 1
    * Removing redundant MWE tags: iteration 2
    * Removing redundant MWE tags: FIXED POINT
  1.4) Acronym processing...
    * Extracting acronyms from text...
    * Generating acronym-free version of corpus...
        * Removing redundant MWE tags: iteration 0
        * Removing redundant MWE tags: iteration 1
        * Removing redundant MWE tags: FIXED POINT
    * Finalizing pre-processing: generated file "genia-pp.xml"
2) POS PATTERN MATCHING
    * Generating test and training splits of pre-processed corpus
  2.1) Extract GS multiword terminology from train set...
    * Extract GS multiword terminology from test set...
  2.2) Tentative patterns precision and recall...
    * Extracting tentative candidates based on most frequent patterns
    * Selecting max-recall patterns with precision above 0.3...
  2.3) Extract selected patterns...
```

```
    * Train candidates
    * Test candidates
    * Finalizing pattern extraction: generated files:
    "mwes/cand-select-uniq.txt" and  "mwes/cand-select-test-uniq.txt"
3) FREQUENCIES
  3.1) From Genia corpus
    * Obtaining Genia marginal frequencies (train)...
    * Obtaining Genia marginal frequencies (test)...
  3.2) From the Web
    * Obtaining Web marginal and global frequencies (train)...
    * Obtaining Web marginal and global frequencies (test)...
    * Correcting frequency incoherencies (train)...
    * Correcting frequency incoherencies (test)...
4) MEASURES
  * Calculating association measures (train)...
  * Calculating association measures (test)...
  * Converting files format ".sts" into ".arff"
  * Ready for WEKA: "cand.arff" and "cand-test.arff"
5) EVALUATIONS
  5.1) Evaluate retokenising and lemmatising...
    * Get new GS (WITHOUT retokenisation and lemmatising)...
      - 29513 MWTs
    * Get new GS (WITH retokenisation and lemmatising)...
      - 28340 MWTs
    * Extracting candidates (WITHOUT retokenisation and lemmatising)...
      - 66817 MWT candidates
      - Precision .1556 Recall .3522 F-measure .2158
      - Hapax legomena 51169
      - Average frequency of top-100 candidates 189.98
    * Extracting candidates (WITH retokenisation and lemmatising)...
      - 63210 MWT candidates
      - Precision .3724 Recall .8306 F-measure .5142
      - Hapax legomena 47770
      - Average frequency of top-100 candidates 205.38
  5.2) Evaluate acronym detection and removal...
    * Get new GS (WITHOUT acronym detection)...
      - 28601 MWTs
    * Get new GS (WITH acronym detection)...
      - 28340 MWTs
    * Extracting candidates (WITHOUT acronym detection)...
      - 61401 MWT candidates
      - Precision .3753 Recall .8058 F-measure .5120
      - Hapax legomena 46330
      - Average frequency of top-100 candidates 205.05
    * Extracting candidates (WITH acronym detection)...
      - 63210 MWT candidates
      - Precision .3724 Recall .8306 F-measure .5142
      - Hapax legomena 47770
      - Average frequency of top-100 candidates 205.38
  5.3) Evaluate Association Measures...
    * Evaluation on test set:
Average precision (%):
prob[Genia]  PMI[Genia]    t[Genia] Dice[Genia]   prob[Web]    PMI[Web]
      38.83       17.51       31.36       22.78       41.49       20.52
     t[Web]   Dice[Web]
      43.21       43.74
(baseline: 24.16%)
    * Evaluation on train set:
```

```
Average precision (%):
prob[Genia]  PMI[Genia]    t[Genia] Dice[Genia]   prob[Web]    PMI[Web]
      46.16       40.08       52.90       38.84       35.77       41.02
     t[Web]   Dice[Web]
      45.13       37.85
(baseline: 37.15%)
    * Resulting graphics in "eval/am"
  5.4) Evaluate pattern selection...
  5.5) Compare with Xtract...
    * Calculating Xtract MWEs (train)...
      - Xtract Precision .6681 Xtract Recall .0384
    * Calculating Xtract MWEs (test)...
      - Xtract Precision .7000 Xtract Recall .0243
  5.6) Compare corpora...
Corpus1 x Corpus2
pearson correlation results  0.121811 0.122419 0.000000
spearman score results  294136053760.000000 -33.836575
kendall score results 0.208740 36.579624 0.000000
  5.7) Evaluate thresholding...
  * PLEASE CORRECT MANUALLY THE .ARFF FILES AND RUN WEKA!
```