TERMINOLOGY EXTRACTION FROM LOG FILES

Hassan Saneifar[1,2], Stéphane Bonniol[2], Anne Laurent[1], Pascal Poncelet[1], Mathieu Roche[1]

[1] *Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM)*
Université Montpellier 2 - CNRS UMR 5506
161 rue Ada, 34392 Montpellier Cedex 5, France
{*saneifar,laurent,poncelet,mroche*}*@lirmm.fr*

[2] *Satin IP Technologies*
Cap Omega, RP Benjamin Franklin, 34960 Montpellier Cedex 2, France
stephane.bonniol@satin-ip.com

Keywords: Natural Language Processing, Information Retrieval, Terminology Extraction, Log Files

Abstract: In many domains, the log files generated by digital systems contain important information on the conditions and configurations of systems. Information Extraction from these log files is an essential phase in information systems, which manage the production line. In the case of Integrated Circuit designs, log files generated by design tools are not exhaustively exploited. Although these log files are written in English, they usually do not respect the grammar and the structures of natural language. Moreover, such logs have a heterogeneous and evolving structure. According to features of such textual data, applying the classical methods of information extraction is not an easy task, more particularly for terminology extraction. In this paper, we thus introduce our approach EXTERLOG to extract the terminology from such log files. We also aim at knowing if POS tagging of such log files is a relevant approach for terminology extraction.

1 Introduction

In many applications, computing systems generate reports automatically. These digital reports, known as system logs, represent the major source of information on the status of systems, products, or even causes of problems that can occur. Although log files are generated in each field of computing, the characteristics of these logs, particularly the language, structure and context, differ from system to system.

In order to extract information from logs, we need to adapt Natural Language Processing (NLP) and Information Extraction (IE) techniques to the specific characteristics of such textual data. In addition, in some areas, such as Integrated Circuit (IC) design systems, the log files are not systematically exploited in an effective way whereas in this particular field, the log files generated by IC design tools, contain essential information on the conditions of production and the final products. In this context, a key challenge is to provide approaches which consider the variable, heterogeneous and scalable structures and vocabulary of this textual data. Furthermore, although the contents of these logs are similar to texts written in Natural Language (NL), they comply neither with the grammar nor with the NL structure. Therefore, we need to study whether the methods of NLP are relevant to that particular context. Another key challenge is to define an automatic protocol of results evaluation. In fact, according to the particularity of such data, evaluation of results based on classic methods are not necessarily relevant.

In order to create the domain ontology for our future work, we aim at exploring the lexical structure of the log files generated by integrated circuit design tools. Define the vocabulary of domain is the first step of process of building an ontology (). In this article, we seek therefore to explore the lexical structure of the log files generated by different tools of integrated circuit design. This analysis also enables us to study the heterogeneous and variable structure of log files generated by different tools. To analyse lexical structure of a corpus, the first step is to identify the terms of domain (*i.e* extraction of domain terminology). We thus present here our approach EXTERLOG (EXtraction of TERminology from LOGs) that is developed to extract the terminology from these log files. We study the relevance of two main approaches of terminology extraction, both of which extract co-occurrences with and without the use of syntactic pat-

terns. We also present an automatic approach of evaluation of extracted terminology and then compare the obtainned results with those obtainned by expert of domain.

In Sect. 2, we develop the utility of building domain ontology and thus terminology extraction in our context and the special features and difficulties of this domain. Our approach EXTERLOG is developed in Sect. 3. Section 4 describes and compares the various experiments that we performed to extract terms from the logs and specially to evaluate them. Finally, we propose a comparison of EXTERLOG and TERMEXTRACTOR system.

2 CONTEXT

Today, digital systems generate many types of log files, which give essential information on these systems. Some types of log files, like network monitoring logs, web services interactions or web usage logs are widely exploited (Yamanishi and Maruyama, 2005)(Facca and Lanzi, 2005)(Li et al., 2008). These kinds of log files are based on the management of events. That is, the computing system, which generates such log files, records the system events based on their occurring times. The contents of these logs comply with norms according to the nature of events and their global usage (*e.g.* web usage area).

However, in some areas such as integrated circuit design systems, rather than being some recorded events, the generated log files are digital reports on configuration, conditions and states of systems. The aim of the exploitation of these log files is not to analyse the events but to extract information about system configuration and especially about the final product's conditions. Hence, these log files are considered an important source of information for information systems designed to query and manage the production line. Information extraction in log files generated by IC design tools has an attractive interest for automatic management and monitoring of production line. However, several aspects of these log files have been less emphasized in existing methods of data mining and NLP. These specific characteristics pose several challenges that require more research.

2.1 Information Extraction from Log Files

To use these logs in an information system, we must implement information extraction methods which are adapted to the characteristics of these logs. Moreover, these features explain why we need a domain ontology to extract information from log files.

There are several levels of design of integrated circuits. At each level, several design tools can be used. Despite the fact that the logs of the same design level report the same information, their structures can differ significantly depending on the design tool used. Specifically, each design tool often uses its own vocabulary to report the same information. In the verification level, for example, we produce two log files (*e.g.* log "A" and log "B") by two different tools. The information about, for example, the "`Statement coverage`" will be expressed as follows in the log "A":

```
      TOTAL     COVERED   PERCENT
      Lines       10        11      12
 statements       20        21      22
```

But this same information in the log "B", will be disclosed from this single line:

```
EC: 2.1%
```

As shown above, the same information in two log files produced by two different tools is represented by different structures and vocabulary. Moreover, design tools evolve over time and this evolution often occurs unexpectedly. Hence, the format of the data in the log files changes, which can make the automatic management of data difficult. We also observed that in these log files, several words are used for the same concept. To present the concept of time, for example, the following words may be found: `Clk`, `CLK`, `Clock`.

Therefore, we need intelligent and generalized methods, which can be applied at the same time on different logs generated by different tools. These methods must take into account the heterogeneity of the structure and vocabulary of these logs. We need the domain ontology to reduce the heterogeneity of terms existing in logs produced by different design tools. This can be useful in the generalization of extraction patterns. For instance, to check "`Absence of Attributes`" as a query on the logs, one must search for the following different sentences in the logs, depending on the version and type of design tool used:

"Do not use **map_to_module attribute**",

"Do not use **one_cold** or **one_hot attributes**",

"Do not use **enum_encoding attribute**",

"The **EVENT attribute** is not supported in subprograms".

Instead of using several patterns, each one adapted to a specific sentence, by associating the words "map_to_module attribute", "one_hot attributes", "enum_ encoding attribute" and "EVENT attribute" to the concept "Absence of Attributes", we use a general pattern that expands automatically according to different logs using the domain ontology. The ontology-driven expansion of query is studied in many works, see (Voorhees, 1994)(Dey et al., 2005).

This ontology allows us to better identify equivalent terms in the logs generated by different tools. Many methods aim to build domain ontology from a given corpus. Several approaches are based on the domain's ontology to better guide the information extraction (Even and Enguehard, 2002). Ontologies categorize domain knowledge and agree on notation norms. Ontologies also define the common vocabulary of a domain (Mollá and Vicedo, 2007). In our context, the domain ontology allows us to categorize the terms associated with a concept sought on the logs. The creation of ontology requires first a lexical analysis of corpus to identify the terms of the domain (). We seek to identify the terms of the logs of each design tool. We will then look at these terms to make the correspondence between them and to create the domain ontology. The terminology of each design tool will be also used to increase the accuracy of information extraction methods. Thus, in this paper, we aim to study the extraction of terminology from log files.

The methods of NLP, including the terminology extraction, developed for texts written in natural language, are not necessarily well suited to the log files. That is due to the specific characteristics of log files, such as the heterogeneity of data, as shown above. The heterogeneity of data exists not only between the log files produced by different tools, but also within a given log file. For example, the symbols used to present a same object, such as the header for a tables, change in a given log. Similarly, there are several formats for punctuation, the separation lines and representation of missing data. In addition, some common characters are used to present different concepts or notions. Furthermore, in these log files, there are many symbols, abbreviations or technical terms which are only understandable considering the documentation of the domain. These terms are often constituted from alphanumeric and special characters.

Also, the language used in these logs is a difficulty that affects the methods of information extraction. Although the language used in these logs is English, the contents of these logs do not usually comply with "*classic*" grammar. In this paper, we thus study these methods and their relevance in this specific context. Finally, we propose our approach EXTERLOG for extracting terminology from these log files.

2.2 Terminology Extraction Background

The extraction of domain terminology from the textual data is an essential task to establish specialized dictionary of a domain (Roche et al., 2004b). The extraction of co-occurring words is an important step in identifying the terms. To identify the co-occurrences, some approaches are based on syntactic techniques which rely initially on the grammatical tagging of words. The terminological candidates are then extracted using syntactic patterns (*e.g.* adjective-noun, noun-noun). We develop the grammatical tagging of log files using our approach EXTERLOG in Sect. 3.2.

Bigrams[1] are used in (meng Tan et al., 2002) as features to improve the performance of the text classification. The series of three words (*i.e.* trigrams) or more is not always essential (Grobelnik, 1998). The defined rules and grammar are used in (David and Plante, 1990) in order to extract the nominal terms as well as to evaluate them. The machine learning methods based on Hidden Markov Models (HMMs) are used in (Collier et al., 2002) to extract terminology in the field of molecular biology. EXIT, introduced by (Roche et al., 2004b) is an iterative approach that finds the terms in an incremental way. A term found in an iteration is used in the next one to find more complex terms. Some works try to extract the co-occurrences in a fixed size window (*normally five words*). In this case, the extracted words may not be directly related (Lin, 1998). XTRACT avoids this problem by considering the relative positions of co-occurrences. XTRACT is a terminology extraction system, which identifies lexical relations in the large corpus of English texts (Smadja, 1993). SYNTEX, proposed by (Bourigault and Fabre, 2000), performs syntactic analysis of texts to identify the names, verbs, adjectives, adverbs, the noun phrases and verbal phrases. It analyses the text by applying syntactic rules to extract terms. TERMEXTRACTOR, submitted by (Sclano and Velardi, 2007), extracts terminology consensually referred in a specific application domain. A corpus of domain documents is input into the software, which parses the documents, and extracts a list of "syntactically plausible" terms (*e.g.* compounds, adjective-nouns, etc.). To select the terms that are relevant to the area, some measures

[1] N-grams are defined as the series of any "n" words.

based on entropy, like "domain relevant" are used.

The statistical methods used are generally associated with syntactic methods for evaluating the adequacy of terminological candidates (Daille, 2003). These methods are based on statistical measures such as information gain to validate an extracted candidate as a term. Among these measures, the occurrence frequency of candidates is a basic notion. However, these statistical methods are not relevant to be applied on the log files. Indeed, statistical approaches can cope with high frequency terms but tend to miss low frequency ones (Evans and Zhai, 1996). According to the log files described above, the repetition of words is rare. Each part of a log file contains some information independent from other parts. In addition, it is not reasonable to establish a large corpus of logs by gathering log files generated by the same tool at the same level of design. Indeed, it just results the redundancy of words. Validation of terms based on some other resources like as web is studied in many works. Web as a huge corpus is more and more used in NLP methods specially in validation of results. However, in our context, we study the corpus of a very specialized domain. The terms used in this domain are the specialized terms and not frequently seen on the web. Furthermore, to evaluate the extracted terms, we define an approach which use an adapted version of statistical measures on the web to our context besides the reference documents of domain. We develop the evaluation protocol in Sect.refeval.

A lot of works compare the different techniques of terminology extraction and their performance. But most of these studies are experimented on textual data, which are classical texts written in natural language. Most of the corpus that are used are structured in a consistent way. In particular, this textual data complies with the grammar of NL. However, in our context, the characteristics of logs such as not to comply with natural language grammar, their heterogeneous and evolving structures (cf. Sect. 2) impose an adaptation of these methods to ensure that they are relevant in the case of log files.

3 EXTERLOG: EXtraction of TERminologie from LOGs

Our approach, EXTERLOG, is developed to extract the terminology in the log files. The extraction process involves normalisation, preprocessing of log files and grammatical tagging of word in order to extract the POS-candidates. EXTERLOG contains also an evaluation phase which involves the validation by reference documents of domain and a web validation using adapted measures to the context.

3.1 Preprocessing & Normalization

The heterogeneity of the log files is a problem, which can affect the performance of information extraction methods. In order to reduce the heterogeneity of data and prepare them to extract terminology, we apply a series of preprocessing and normalization on the logs. Given the specificity of our data, the normalization method, adapted to the logs, makes the format and structure of logs more consistent. We replace the punctuations, separation lines and the headers of the tables by special characters to limit ambiguity. Then, we tokenize the texts of logs, considering that certain words or structures do not have to be tokenized. For example, the technical word "`Circuit4-LED3`" is a single word which should not be tokenized into two words "Circuit4" and "LED3". Besides, we distinguish automatically the lines representing the header of tables from the lines which separate the parts. After normalization of logs, we have less ambiguity and less common symbols for different concepts. This normalization makes the structure of logs produced by different tools more homogeneous.

3.2 Grammatical Tagging

Grammatical tagging (also called *part-of-speech tagging*) is a method of NLP used to analyse the text files which aims to annotate words based on their grammatical roles. In the context of log files, there are some difficulties and limitations for applying a grammatical tagging on such textual data.
Indeed, the classic techniques of POS tagging are developed using the standard grammar of natural language. In addition, they are normally trained on texts written in a standard natural language, such as journals. Therefore, they consider that a sentence ends with a fullstop, for example, which is not the case in the log files that we handle. More specifically, in these log files, sentences and paragraphs are not always well structured. Besides, there are several constructions that do not comply with the structure of sentences in natural language. To identify the role of words in the log files, we use BRILL rule-based part-of-speech tagging method (Brill, 1992). As existing taggers like BRILL are trained on general language corpora, they give inconsistent results on the specialized texts. (Amrani et al., 2004) propose a semi-automatic approach for tagging corpora of speciality. They build a new tagger which corrects the base of rules obtainned by BRILL tagger and adapt it to a

corpus of speciality. In the context of log files, we need also to adapt BRILL tagger just as in (Amrani et al., 2004). We thus adapted BRILL to the context of log files by introducing the new *contextual* and *lexical* rules. Indeed, the classic rules of BRILL, which are defined according to the NL grammar, are not relevant to log files. For example, a word beginning with a number is considered a "*cardinal*" by BRILL. However, in the log files, there are many words like `12.1vSo10` that must not be labelled as "*cardinal*". Therefore, we defined the special *lexical* and *contextual* rules in BRILL. The structures of log files can contribute important information for extracting the relevant patterns in future works. Therefore, we preserve the structure of files during grammatical tagging. We introduce the new tags, called "*Document Structure Tags*", which present the different structures in log files. For example, the tag "\TH" represents the header of tables or "\SPL" represents the lines separating the log parts. The special structures in log files are identified during normalization by defined rules. Then, they are identified during tagging by the new specific contextual rules defined in BRILL. We finally get the logs tagged by the grammatical roles of words and also by the labels that determine the structure of logs.

3.3 Extraction of Co-occurrences

We are looking for co-occurrences in the log files with two different approaches:

1. extraction of co-occurrences respecting a defined *part-of-speech* syntactic pattern,
2. extraction of co-occurrences without using the syntactic patterns.

We call the co-occurrences extracted by the first solution "POS-candidates"[2]. This approach consists of filtering words by the syntactic patterns. The syntactic patterns determine the adjacent words with the defined grammatical roles. The syntactic patterns are used in (Daille, 2003) and (Bourigault and Fabre, 2000) to extract terminology. For complex terms identification, (Daille, 2003) defines syntactic structures which are potentially lexicalisable in French. (Bourigault and Fabre, 2000) perform a syntactical analysis of text to identify part-of-speech roles. As argued in (Daille, 2003), the base structures of syntactic patterns are not frozen structures and accept variations. According to the terms found in our context, the syntactic patterns that we use to extract the "POS-candidates" from log files are:

"\JJ - \NN" (Adjective-Noun),

"\NN - \NN" (Noun-Noun).

The co-occurrences extracted by the second approach are called "bigrams". A bigram is extracted as a series of any two adjacent relevant words[3]. Bigrams are used in NLP approaches as representative features of a text (meng Tan et al., 2002),(Grobelnik, 1998). However, the extraction of bigrams does not depend on the grammatical role of words. To extract significant bigrams, we consider the tool words (stopwords) existing in the logs. Therefore, we normalize and tokenize the logs to reduce the rate of noise. The extracted bigrams represent two ordinary adjacent words. In this case, we do not filter the words according to their grammatical roles.

In this paper, we refer to POS-candidates and bigrams as "*terminological candidates*". These terminological candidates must be evaluated to find the relevant terms of the domain.

3.4 Evaluation of Candidates

The extracted terminological candidates should be evaluated to identify the most relevant terms according to the context. All the extracted terminological candidates are not necessarily the relevant terms. Moreover, we are focalised on a specialized domain where just some terms are bidden to domain's context.

3.4.1 Validation by Statistical Measures

The statistical measures are often used in terminology extraction field to evaluate the terms (see (Daille, 1996; Roche et al., 2004a)). The following are the most widely used.

Mutual Information. One of the most commonly used measures to compute a sort of relationship between the words composing what is called a **co-occurrence** is Church's Mutual Information (MI) (Church and Hanks, 1990). The simplified formula is the following where *nb* designates the number of occurrences of words and couples of words:

$$IM(x,y) \quad = \quad \log_2 \frac{nb(x,y)}{nb(x)nb(y)}$$

Cubic Mutual Information. The Cubic Mutual Information is an empirical measure based on MI,

[2]POS: Part-Of-Speech

[3]The relevant words, in our context, are all words of the vocabulary of this domain excluding the stop words like "be", "have" or "the".

that enhances the impact of frequent co-occurrences, something which is absent in the original MI (Daille, 1994).

$$IM3(x,y) = \log_2 \frac{nb(x,y)^3}{nb(x)nb(y)}$$

This measure is used in several works related to noun or verb terms extraction in texts (Roche and Prince, 2007).

Dice's Coefficient. An interesting quality measure is Dice's coefficient (Smadja et al., 1996). It is defined by the following formula based on the frequency of occurence.

$$Dice(x,y) = \frac{2 \times nb(x,y)}{nb(x)+nb(y)}$$

These measures are based on the occurence frequency of terms in corpus. However, as described below, we could not rely on the occurence frequencies of terms in log corpus. That is why we evaluate the extracted terms by means of their frequencies on Web as a huge resource of textual data. Although, by using Web, the specialized context of data is ignored.

In fact, we are interested to terms which are bidden to domain's context. But on web we capture occurrences of terms regardless of the context in which they are seen. To solve this problem, we use an extension of described measures called $AcroDef$. $AcroDef$ is a quality measure where context and web resources are essential characteristics to be taken into account (see (Roche and Prince, 2007)). The below formulas define the $AcroDef$ measures, respectively based on MI and Cubic MI.

$$AcroDef_{IM}(a^j) = \frac{nb(\bigcap_{i=1}^{n} a_i^j + C)}{\prod_{i=1}^{n} nb(a_i^j + C | a_i^j \notin M_{tools})} \text{ where } n \geq 2$$

$$AcroDef_{IM3}(a^j) = \frac{nb(\bigcap_{i=1}^{n} a_i^j + C)^3}{\prod_{i=1}^{n} nb(a_i^j + C | a_i^j \notin M_{tools})} \text{ where } n \geq 2$$

In $AcroDef$, the **context** "C" is represented as a set of significant words. The nb function used in the preceding measures represents the number of pages provided by the search engine. More precisely, The nb function is the number of pages returned with the n words x_i^j ($i \in [1,n]$) of the term x^j. Then $nb(a_i^j + C)$ returns the number of pages applying query $a_i^j + C$ using the AND operator of search engine with the words of the term a^j and those of context C. In our case, for example, for a term x^j like "atpg patterns" consisting of two words (so $i = 2$), $nb(atpg \bigcap patterns + C)$ is the number page returned by applying query "atpg pattern" AND C on a search engine, where C is the words representing the context. The $AcroDef_{Dice}$ formula based Dice's formula is written as follows:

$$\frac{\left|\{a_i^j + C | a_i^j \notin M_{tools}\}_{i \in [1,n]}\right| \times nb(\bigcap_{i=1}^{n} a_i^j + C)}{\sum_{i=1}^{n} nb(a_i^j + C | a_i^j \notin M_{tools})} \text{ where } n \geq 2$$

In (Roche and Prince, 2007), "C" is represented as a set of significant words (*e.g.* encryption, information and code to represent the Cryptography context). The authors made some experiments with different number of words represented as context. In all case, the search operator used between the words of context is "AND" operator. That is, they request the pages containning all words that represent the context. However, working on a very specialized domain which contains some more specific sub domains, we do not get the best results by using just "AND" operator for the words of context.

To specify the words which represent the context of log files in our case, we make a corpus of documents including the reference documents of Integrated Circuit design tools and tree other domains documents. We rank the words of corpus by using tf-idf measure. Tf-idf favours the frequent terms of a domain which are not frequent in other domains. We choose the five most ranked words of IC design documents as representing word of the context. As argued above, we look for web pages containning the terms and *two* or *more* words of context. Then, the extracted terms are ranked by $AcroDef$ measures.

3.4.2 Validation by Reference Documents

Beside the validation of terms by statistical measures, we compare the extracted terms with the terms extracted from reference documents of domain. For each level of integrated circuit design, there exist some documents which explain the principles of design and more specifically the details of design tools. We extract terminology of these reference documents. To evaluate the extracted terms from log files, we compare these terms with the terminology of reference documents to find the common terms. Note that to create domain ontology we have to extract terms from log files in spite of existing reference documents. The extracted terminology from reference documents are not complete because many terms and lexical structures exit only in log files. The non common terms can be validated by $AcroDef_{Dice}$ measure.

4 Experiments

In all experiments the log corpus is composed of logs of five levels of IC design. For each level, we considered two logs generated in different conditions of design systems. The size of the log corpus is about 950 KB. The corpus of reference documents consists of three documents per design level. These documents are of considerable size. Each document consists of approximately 600 pages.

4.1 POS-candidates vs. Bigrams

We experimented two different approaches for the extraction of terminology from these logs: (1) using syntactic patterns (*POS-candidates*) and (2) without the use of syntactic patterns (*bigrams*). Here, we analyse the terminological candidates obtainned by each one. To evaluate the terms extracted by each method, we compare them with terminology of reference documents. At this stage, precision of extracted candidates is calculated as the percentage of POS-candidates (bigrams) existing among the reference terms. Table 1 shows the accuracy of POS-candidates and bigrams based on the *validation by reference documents*. The comparison of terminological candidates with the reference terms shows that the terminology extraction based on syntactic patterns is quite relevant to the context of log files. The accuracy of POS-candidates is indeed higher than the precision of bigrams. Despite the fact that normalization and tagging the texts of logs is not an easy task, our experiments show that an effort in this direction is quite useful in order to extract quality terms.

4.2 Evaluation of Terms by $AcroDef$

To evaluate the extracted terms from log files by EXTERLOG, we rank them by $AcroDef$. To apply $AcroDef$, we determine the context words as described in Sect.3.4.1. The selected words to represent the context are : "*IP block*", "*Synopsys*", "*design compiler*", "*leda*" and "*Semiconductor*". We use Google search engine to capture the number of pages containning a given term and two or more words of context. For a given term like "CPU time", the query used in Google search engine is like "CPU time" AND "IP block" AND Synopsys OR "design compiler" OR leda OR Semiconductor. Then, we asked two domain experts to evaluate the extracted terms manually. They tagged extracted terms from logs as *relevant* or *no relevant* according to the context and their utility in the logs. We compare the ranked terms by $AcroDef$ with terms tagged by experts to calculate the precision and recall of our terminology extraction approach. More precisely, we consider different thresholds according to the values of $AcroDef$. For each case, the precision is calculated as percentage of terms of which $AcroDef$ value is above threshold and are also tagged as "relevant" by experts. We calculate recall as percentage of relevant terms (tagged by experts) which are classed above threshold. Table **??** and **??** show the obtained results by $AcroDef_{MI3}$ and $AcroDef_{Dice}$ on the terms extracted by EXTERLOG.

The results demonstrate that classification of extracted terms on two categories of "relevant" and "no relevant" using the ranking scores based on $AcroDef_{Dice}$ is more accurate in our context. By configuring the threshold of $AcroDef_{Dice}$ equal to t_1 in EXTERLOG, we extract the relevant terms with the precision of 75% and we cover 79% of log files relevant terms.

	Precision	Recall	F-score
t_1	78%	62%	69
t_2	82%	54%	65
t_3	87%	49%	62
t_4	87%	48%	61

Table 2: Evaluation of terms extracted by EXTERLOG based on $AcroDef_{MI3}$

	Precision	Recall	F-score
t_1	75%	79%	76
t_2	74%	74%	74
t_3	77%	60%	67
t_4	81%	52%	63

Table 3: Evaluation of terms extracted by EXTERLOG based on $AcroDef_{Dice}$

4.3 EXTERLOG vs. TermExtractor

Here, we compare the results of our approach EXTERLOG with those obtained by TERMEXTRACTOR on the same corpus of logs used in the previous experiments. To adapt TERMEXTRACTOR to this context, we configured it according to characteristics of log files and especially the type of terms found in this context. The configuration of TERMEXTRACTOR is described below.

- the terms consist of two words.
- the minimum frequency of a term in the corpus is equal to 1.

	Level 1		Level 2		Level 3		Level 4		Level 5	
Candidate terms	POS	Bigrams	POS	Bigrams	POS	Bigrams	POS	Bigrams	POS	Bigrams
precision	67.7	11.3	20.7	6.5	37.8	9.9	40.1	6.5	19.6	5.1

Table 1: Precision of terminological candidates extracted from logs based on the reference terms.

- words constituted of fewer than three letters are allowed.
- words can contain numbers.

Table 4 shows the results obtained by TERMEXTRACTOR compared with those obtained by EXTERLOG (*using syntactic patterns*).

By analysing the terms extracted by TERMEXTRACTOR, we find that the structure of logs has influenced the extraction of terms. For example, let us consider the following line in a log file:

```
Protocol        optimization        warning
```

By applying the classic methods of grammatical tagging, we obtain:

`Protocol`/NNP[4] `optimization`/NN[5] `warning`/NN

As shown, the classic methods of normalization and especially grammatical tagging (notably those used in TERMEXTRACTOR) do not consider the number of blank spaces between words (i.e. *structure*).
Therefore, using TERMEXTRACTOR, we would qualify "`Protocol optimization`", which respects the syntactic pattern "Noun-Noun", as a terminological candidate. However, EXTERLOG which considers the structure of texts (here, the blank spaces between the words "`Protocol`" and "`value`"), avoids extracting terminological candidates like "`Protocol optimization`", which is not a relevant term.
Furthermore, the technical terms of domain are rarely found by TERMEXTRACTOR. These terms are normally made up of special characters. For example, the following technical terms are not found by TERMEXTRACTOR, whereas they are found by EXTERLOG:

`ks_comp engine`,

`rule b9`,

`policy ieee_rtl_synth_subset`,

`policy ver_starc_dsg`,

`scirocco_cycle ruleset`.

As these terms are not normally made up of traditional words (i.e. *those found in classical texts*), standard methods of normalization and grammatical tagging are not able to label them relevantly. However, EXTERLOG based on BRILL tagger that we have adapted to our context (with new contextual and lexical rules), identifies these technical terms. Validation of technical terms using reference documents remains a sensitive issue because such terms rarely appear in the references. The evaluation of these terms by an expert is therefore essential for our future work.

Table 4 shows that the recall of words obtained by TERMEXTRACTOR is very low. In fact, TERMEXTRACTOR is based on statistical measures for filtering terms. Although we have reduced the minimum thresholds of measures used by TERMEXTRACTOR, it filters a considerable number of terms. As we have seen in our experiments concerning pruning of candidates according to their frequency, we may lose valid terms with low frequency in logs.

5 Conclusion & Future Work

In this paper, we described a particular type of textual data: log files generated by tools for integrated circuit design. The text of these log files does not comply with the grammar of natural language, despite the fact that it is similar to texts written in natural language. In addition, log files have highly heterogeneous and evolving structures. To extract domain terminology, we extracted the co-occurrences with two different approaches: (1) extraction of co-occurrences using the syntactic patterns and (2) extraction without syntactic patterns. Although these texts (*log*) do not usually comply with the grammar of natural language and in spite of their specific structures, results of experiments show that terms obtained using the syntactic patterns are more relevant than those obtained without using syntactic patterns. In addition, we have applied the specific preprocessing and normalization methods to improve the precision of extracted terms. Our experiments show that our approach extracts more relevant terms of domain than other terminology extraction methods like TERMEXTRACTOR.
To improve the performance of terminology extraction, we will develop our normalization method. Given the importance of accurate grammatical tagging, we will improve the grammatical tagger. Man-

[4]Proper Noun
[5]Noun

	Level 1		Level 2		Level 3		Level 4		Level 5	
	EXT	TER	EXT	TER	EXT	TER	EXT	TER	EXT	TER
Precision	67.7	56.1	20.7	14.0	37.8	38.1	40.1	35.2	19.6	26.3
Recall	0.7	0.3	7.6	0.3	1.3	0.4	9.5	2.5	0.3	0.1

Table 4: Precision and recall of terms extracted by EXTERLOG (EXT) and by TERMEXTRACTOR (TER)

ual expertise of terms extracted using our system should be conducted to confirm the results presented in this article. Finally, we plan to take into account the terminology extracted using our system to enrich the patterns of information extraction currently used by our current methods.

REFERENCES

Amrani, A., Kodratoff, Y., and Matte-Tailliez, O. (2004). A semi-automatic system for tagging specialized corpora. In *PAKDD*, pages 670–681.

Bourigault, D. and Fabre, C. (2000). Approche linguistique pour l'analyse syntaxique de corpus. *Cahiers de Grammaire - Université Toulouse le Mirail*, (25):131–151.

Brill, E. (1992). A simple rule-based part of speech tagger. In *In Proceedings of the Third Conference on Applied Natural Language Processing*, pages 152–155.

Church, K. W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. In *Computational Linguistics*, volume 16, pages 22–29.

Collier, N., Nobata, C., and Tsujii, J. (2002). Automatic acquisition and classification of terminology using a tagged corpus in the molecular biology domain. *Journal of Terminology, John Benjamins*, 7(2):239–257.

Daille, B. (1994). *Approche mixte pour l'extraction automatique de terminologie : statistiques lexicales et filtres linguistiques*. PhD thesis, Universit Paris 7.

Daille, B. (1996). Study and Implementation of Combined Techniques for Automatic Extraction of Terminology. In *The Balancing Act: Combining Symbolic and Statistical Approaches to Language, MIT Press*, pages 49–66.

Daille, B. (2003). Conceptual structuring through term variations. In *Proceedings of the ACL 2003 workshop on Multiword expressions*, pages 9–16, Morristown, NJ, USA. Association for Computational Linguistics.

David, S. and Plante, P. (1990). De la nécessité d'une approche morpho-syntaxique en analyse de textes. *Intelligence Artificielle et Sciences Cognitives au Québec*, 2(3):140–155.

Dey, L., Singh, S., Rai, R., and Gupta, S. (2005). Ontology aided query expansion for retrieving relevant texts. In *AWIC*, pages 126–132.

Evans, D. A. and Zhai, C. (1996). Noun-phrase analysis in unrestricted text for information retrieval. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 17–24, Morristown, NJ, USA. Association for Computational Linguistics.

Even, F. and Enguehard, C. (2002). Extraction d'informations à partir de corpus dégradés. In *Proceedings of 9ème conference sur le Traitement Automatique des Langues Naturelles (TALN'02)*, pages 105–115.

Facca, F. M. and Lanzi, P. L. (2005). Mining interesting knowledge from weblogs: a survey. *Data Knowl. Eng.*, 53(3):225–241.

Grobelnik, M. (1998). Word sequences as features in text-learning. In *In Proceedings of the 17th Electrotechnical and Computer Science Conference (ERK98*, pages 145–148.

Li, D. H., Laurent, A., and Poncelet, P. (2008). Mining unexpected web usage behaviors. In *ICDM*, pages 283–297.

Lin, D. (1998). Extracting collocations from text corpora. In *In First Workshop on Computational Terminology*, pages 57–63.

meng Tan, C., fang Wang, Y., and do Lee, C. (2002). The use of bigrams to enhance text categorization. In *Inf. Process. Manage*, pages 529–546.

Mollá, D. and Vicedo, J. L. (2007). Question answering in restricted domains: An overview. *Computational Linguistics*, 33(1):41–61.

Roche, M., Azé, J., Kodratoff, Y., and Sebag, M. (2004a). Learning interestingness measures in terminology extraction. a roc-based approach. In *Proceedings of "ROC Analysis in AI" Workshop (ECAI 2004)*, pages 81–88.

Roche, M., Heitz, T., Matte-Tailliez, O., and Kodratoff, Y. (2004b). EXIT: Un système itératif pour l'extraction de la terminologie du domaine à partir de corpus spécialisés. In *Proceedings of JADT'04 (International Conference on Statistical Analysis of Textual Data)*, volume 2, pages 946–956.

Roche, M. and Prince, V. (2007). AcroDef : A quality measure for discriminating expansions of ambiguous acronyms. In *CONTEXT*, pages 411–424.

Sclano, F. and Velardi, P. (2007). Termextractor: a web application to learn the shared terminology of emergent web communities. In *Proceedings of the 3rd International Conference on Interoperability for Enterprise Software and Applications (I-ESA 2007)*, Funchal, Portugal.

Smadja, F. (1993). Retrieving collocations from text: Xtract. *Comput. Linguist.*, 19(1):143–177.

Smadja, F., McKeown, K. R., and Hatzivassiloglou, V. (1996). Translating collocations for bilingual lexicons: A statistical approach. *Computational Linguistics*, 22(1):1–38.

Voorhees, E. M. (1994). Query expansion using lexical-semantic relations. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 61–69, New York, NY, USA. Springer-Verlag New York, Inc.

Yamanishi, K. and Maruyama, Y. (2005). Dynamic syslog mining for network failure monitoring. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 499–508, New York, NY, USA. ACM.