MINING FOR RELEVANT TERMS FROM LOG FILES

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

[1] *Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM)*
Univ. Montpellier 2 - CNRS
161 rue Ada, 34392 Montpellier Cedex 5, France

[2] *Satin IP Technologies*
Cap Omega, RP Benjamin Franklin, 34960 Montpellier Cedex 2, France

Keywords: Natural language processing, Information retrieval, Terminology extraction, Terminology ranking, Log files.

Abstract: The Information extracted from log files of computing systems can be considered one of the important resources of information systems. In the case of Integrated Circuit design, log files generated by design tools are not exhaustively exploited. The logs of this domain are multi-source, multi-format, and have a heterogeneous and evolving structure. Moreover, they usually do not respect the grammar and the structures of natural language though they are written in English. According to features of such textual data, applying the classical methods of information extraction is not an easy task, more particularly for terminology extraction. We have previously introduced EXTERLOG approach to extract the terminology from such log files. In this paper, we introduce a new developed version of EXTERLOG guided by Web. We score the extracted terms by a *Web* and *context* based measure. We favor the more relevant terms of domain and emphasize the precision by filtering terms based on their scores. The experiments show that EXTERLOG is well-adapted terminology extraction approach from log files.

1 INTRODUCTION

In many applications, computing systems generate reports automatically. These digital reports, also known as system logs, represent the major source of information on the status of systems, products, or even the causes of problems that can occur. Although log files are generated in every field of computing, the characteristics of these logs, particularly the language, structure and context, differ from system to system. 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 condition of production and the final products. In this context, a key challenge is to provide approaches which consider *the multi-source*, *heterogeneous* and *scalable structures* of log files as well as their *special vocabulary*. 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, In order to extract information from the logs, we need to adapt Natural Language Processing (NLP) and Information Extraction (IE) techniques to the specific characteristics of such textual data. Another key challenge is evaluation of results. In fact, according to the particularity of such data, and then due to the high noise ratio in results, the classic evaluation methods are not necessarily relevant. To emphasize the precision of results as a must according to the accuracy of context, we have to define the noise filtering method which comply with the particularity of such data.

The creation of a domain ontology is a primordial need for our future work on information extraction from log files. Defining the vocabulary of domain is one of the first steps of building an ontology. To analyze vocabulary and lexical structure of a corpus, extraction of domain terminology is one of the most important phases. We thus aim at extracting the terminology of log files. The extracted terms will be used in the creation of domain ontology in our future works. Also, we will use extracted terms to study the

Saneifar H., Bonniol S., Laurent A., Poncelet P. and Roche M. (2009).
MINING FOR RELEVANT TERMS FROM LOG FILES.
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, pages 77-84
DOI: 10.5220/0002307200770084

different lexical structures of different logs in order to enrich our information extraction methods. In this paper, we introduce a new version of our approach EXTERLOG (EXtraction of TERminology from LOGs), previously presented in (Saneifar et al., 2009), that is developed to extract the terminology from these log files. In this approach, we study how to adapt the existing terminology extraction methods to the particular and heterogeneous features of log files. We also present in this paper a filtering method of extracted terms based on a ranking score in order to emphasize the precision of extracted relevant terms.

In Sect. 2, we detail the utility of building domain ontology and thus the 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 the precision of EXTERLOG.

2 CONTEXT

Today, digital systems generate many types of log files, which give essential information on the 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). These kinds of log files are based on the management of events. That is, the computing system, which generates the 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, condition and states of systems. The aim of the exploitation of these log files is not to analyze the events but to extract information about system configuration and especially about the final product's condition. Hence, log files are considered an important source of information for systems designed to query and manage the production. Information extraction in log files generated by IC design tools has an attractive interest for automatic management and monitoring of IC production. However, several aspects of these log files have been less emphasized in existing methods of text mining and NLP. These specific characteristics raise several challenges that require more research.

2.1 IE & 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 the log files.

In the field of integrated circuits design, several levels need to be considered. At every level, different design tools can be used which make the generated log files the *multi-source* data. 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 the 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. Therefore, the *format of the data* in the log files changes, which make the automatic management of data difficult. 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 an object, such as the header for tables, change in a given log. Similarly, there are several formats for punctuation, the separation lines, and representation of missing data. Therefore, we need intelligent and generalized methods, which can be applied at the same time on different logs (*multi-source textual data*) which have the multi-format and heterogeneous data. These methods must also take into account the variable vocabulary of these logs. To generalize the extraction methods, we thus need to identify the terms used by each tool in order to create the domain ontology. This ontology allows us to better identify equivalent terms in the logs generated by different tools and so to reduce the heterogeneity of data. 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**"

Instead of using several patterns, each one adapted for a specific sentence, by associating the words "map_to_module attribute", "one_hot attributes" and "enum_ encoding 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).

The ontology will allow us to better identify equivalent terms in the logs generated by different tools. Several approaches are based on the domain ontology to better guide the information extraction (Even and Enguehard, 2002). An ontology also defines 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 a lexical analysis of a corpus to identify the terms of the domain. We hence seek to identify the terms of the logs of every design tool. We will then look at these terms in order to make the correspondence between them and to create the domain ontology. Thus, we aim at studying the extraction of terminology from log files.

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. Moreover, there are words that are often constituted from alphanumeric and special characters.

Due to these specific characteristics of 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.

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., 2004). 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., 2004) 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.

As described above, we have previously studied the extraction of terminology based on identifying the co-occurring words *without* using the syntactic patterns from log files (see (Saneifar et al., 2009)). As explained in (Saneifar et al., 2009), the terminology extraction based on syntactic patterns is quite relevant to the context of log files. We shown that the accuracy of terms extracted based on syntactic patterns is indeed higher than the precision of bigrams extracted without such patterns. Despite the fact that normalization and tagging the texts of logs is not an easy task, our previous experiments show that an effort in this direction is useful in order to extract quality terms. But according to the need of high accuracy in this domain and the fact that manual validation of terms by an expert is expensive, we develop here the automatic evaluation phase of EXTERLOG. This evaluation of terms is detailed in Section 3.4.

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

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

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. Since, it just results the redundancy of words. Evaluation of terms based on some other resources like as web is studied by many works. The 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. Then, we could not use the classic statistical measures based on simple frequencies of terms in corpus in order to give a score to every extracted term. Furthermore, our approach aims at reducing the noise ratio in results, thus emphasizing the precision, by filtering the extracted terms using a web based statistical measures which considers in the same time the context of log files. We detail this aspect in Sect. 3.4.

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 for the case of log files.

3 EXTERLOG: EXTRACTION OF TERMINOLOGY FROM LOGS

Our approach, EXTERLOG, is developed to extract the terminology in the log files. The extraction process involves normalization, preprocessing of log files and grammatical tagging of word in order to extract the terms. EXTERLOG contains also a filtering phase of extracted terms based on a scoring measure.

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 the 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). Since 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 obtained 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. Since, 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 extract the co-occurrences in the log files respecting a defined *part-of-speech* syntactic pattern. We call the co-occurrences extracted using syntactic pattern "POS-candidates"[2]. 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. 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).

These extracted terms at this phase must be scored to favor the most relevant terms of the domain.

3.4 Filtering of Candidates

All the extracted terms are not necessarily the relevant terms of the domain. Because of some huge log files and the large vocabulary of the logs, there exists so many extracted terms. Also, according to the particular features of such data, in spite of adapted normalization and tagging methods that we used, there exists some noise (no relevant terms) in the extracted terms. Moreover, we are focused on a specialized domain where just some terms are really bidden to the domain's context. Thus, we score, rank and then filter the extracted terms in order to favor the most relevant terms according to the context. The statistical measures are often used in terminology extraction field to evaluate the terms (see (Daille, 1996)). The following ones 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:

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

Cubic Mutual Information. The Cubic Mutual Information is an empirical measure based on MI, that enhances the impact of frequent co-occurrences, something which is absent in the original MI (Daille, 1994).

$$MI3(x,y) = \log_2 \frac{nb(x,y)^3}{nb(x)nb(y)}$$

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 occurrence.

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

These measures are based on the occurrence frequencies of terms in corpus. Scoring the terms based on frequencies of terms in corpus of logs is not a relevant approach in our context. As we have already explained, the techniques based on frequency of terms in a corpus (e.g. pruning terms having low frequency) are not relevant to this context as a *representative term* does *not* necessarily have a *high frequency* in log files. That is why we score the terms according to their frequencies on the Web as a large corpus where frequency of a term can be representative. Working on a specialized domain, we have bias scores based on the simple count of occurrences of a term on Web. Indeed, on Web, we capture occurrences of terms regardless of the context in which they are seen. Thus, we should consider only the occurrences of terms

[2]POS: Part-Of-Speech

on web which are situated in the IC design context. We use therefore 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.

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

$$AcroDef_{MI3}(a^j) = \frac{nb(\bigcap_{i=1}^{n} a_i^j + C)^3}{\prod_{i=1}^{n} nb(a_i^j + C | a_i^j \notin M_{stop-words})} \quad \text{where } n \geq 2$$

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 to given query. Then $nb(a_i^j + C)$ returns the number of pages applying query $a_i^j + C$ which means all words of the term a^j in addition to 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 a set of words representing the context. The $AcroDef_{Dice}$ formula based Dice's formula is written as follows:

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

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 cases, authors use "AND" search engine operator between the words of context. That is, they request the pages containing all words in "*C*". However, working on a very specialized domain which contains some more specific sub domains, we do not get the best results by using just an "AND" operator for the words of context.

To specify the words which represent the context of log files, we build 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 (see (Salton and Buckley, 1987)). Tf-idf gives higher score to the frequent words of a domain which are not frequent in other ones. Then, we choose the first five words (ranked in tf-idf order) of IC design documents as representing word of the context. As argued above, we look for web pages containing a given term and *two* or *more* words of context (*using the operators OR and AND*). Finally, the extracted terms are ranked by means of their *AcroDef* scores. We favor the most ranked terms by filtering those having most low *AcroDef* scores.

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. All the experiments are done using the Google search engine.

4.1 Evaluation of Terms by *AcroDef*

The extracted terms by EXTERLOG from the log files are so numerous which make difficult the final validation by experts of domain. Thus, we experiment by taking a sample of extracted terms. We select the 200 more frequent terms extracted from logs of every IC design level. Note that in few levels, there exists less than 200 terms. The taken sample consists of 700 terms at all.

To filter the extracted terms from log files, we rank them by *AcroDef* (cf. 3.4). To apply *AcroDef*, we determine the context words as described in Sect. 3.4. Then, we use the Google search engine to capture the number of pages containing a given term and *two* or *more* words of context. Suppose a given term like "CPU time" where C_i $i \in \{1-5\}$ are the context words, the query used in Google search engine is "CPU time" AND C_1 AND C_2 OR C_3 OR C_4 OR C_5.

Once *AcroDef* scores are calculated, we rank the terms based on their *AcroDef*. The more *AcroDef* has a higher value, the more the term is representative (*seen*) in our context. Then, we select the most rated terms in the goal of emphasizing the precision by reducing the noise ratio (no relevant terms) in results. Once the terms filtered, we asked two domain experts to evaluate remain terms in order to determine the precision of our terminology extraction approach from log files. First extracted terms are tagged by a domain expert as *relevant* or *not relevant* according to the context and their usefulness in the logs. Then, another expert reviewed the tagged terms by the first

expert. Then, the precision is calculated as percentage of remain terms (*after filtering by AcroDef scores*) which are tagged as "relevant" by experts.

$$Precision = \frac{|Terms_{relevant} \cap Terms_{remained}|}{|Terms_{remained}|}$$

$Terms_{relevant}$ = terms validated by expert in **sample scale**
$Terms_{remained}$ = terms remained after filtering

We calculate the recall as the percentage of all relevant terms (*tagged by experts in **sample scale***) which remain after filtering.

$$Recall = \frac{|Terms_{relevant} \cap Terms_{remained}|}{|Terms_{relevant}|}$$

$Terms_{validated}$ = terms validated by expert in **sample scale**
$Terms_{remained}$ = terms remained after filtering

We also calculate F-score as the harmonic mean of precision and recall to measure our approach accuracy.

$$F-score = \frac{2*(Precision*Recall)}{Precision+Recall}$$

We experiment with different numbers of the most ranked terms as the ones which remain after filtering. That is, suppose n terms in sample, we filter the terms by selecting the m most ranked terms by *AcroDef* score. Table 1 shows the results of filtering with different m as threshold of filtering. In $m = 500$, for example, we take the 500 most ranked terms. In $m = 700$, we do not actually filter any terms. Thus, the recall is equal to 100%. The results show that by means of our filtering approach, we favor more relevant terms and emphasize the precision.

Table 1: Precision, Recall, and F-score of terms in each level of filtering based on $AcroDef_{MI}$ score.

m	Precision	Recall	F-score
200	80 %	38 %	52 %
300	78 %	56 %	65 %
400	74 %	71 %	72 %
500	72 %	87 %	79 %
600	66 %	95 %	78 %
700	59 %	100 %	74 %

Table 2 demonstrates the same filtering results but based on $AcroDef_{MI3}$ scores. Table 3 shows the same experiments using $AcroDef_{Dice}$ as scoring measure. According to results, $AcroDef_{MI3}$ is more relevant to score the extracted terms in our context. By using $AcroDef_{MI3}$ we reach better precision. That is, $AcroDef_{MI3}$ score better the relevant terms of domain.

Table 2: Precision, Recall, and F-score of terms in each level of filtering m based on $AcroDef_{MI3}$ score

m	Precision	Recall	F-score
200	86 %	41 %	56 %
300	79 %	57 %	67 %
400	76 %	74 %	75 %
500	72 %	87 %	79 %
600	66 %	95 %	78 %
700	59 %	100 %	74 %

Table 3: Precision, Recall, and F-score of terms in each level of filtering m based on $AcroDef_{Dice}$ score

m	Precision	Recall	F-score
200	85 %	41 %	55 %
300	79 %	57 %	67 %
400	74 %	72 %	73 %
500	72 %	87 %	79 %
600	66 %	95 %	78 %
700	59 %	100 %	74 %

5 CONCLUSIONS & FUTURE WORK

In this paper, we describe a particular type of textual data: log files generated by tools for integrated circuit design. Since these log files are multi-source, multi-format, heterogeneous, and evolving textual data, the NLP and IE methods are not necessarily well suited to extract information.

To extract domain terminology, we extracted the co-occurrences. For that, we apply the specific pre-processing, normalization and tagging methods. To reduce the noise ratio in extracted terms and favor more relevant terms of this domain, we score terms using a Web and context based measure. Then, we select the most ranked terms by filtering based on score of terms. The experiments show that our approach of terminology extraction from log files, EXTERLOG, can achieve an F-score equal to 0.79 after filtering of terms.

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. Finally, we plan to take into account the terminology extracted using our system to enrich the patterns of information extraction from log files.

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.

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., Heitz, T., Matte-Tailliez, O., and Kodratoff, Y. (2004). 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.

Salton, G. and Buckley, C. (1987). Term weighting approaches in automatic text retrieval. Technical report, Ithaca, NY, USA.

Saneifar, H., Bonniol, S., Laurent, A., Poncelet, P., and Roche, M. (2009). Terminology extraction from log files. In *DEXA '09: Proceedings of the 20th international conference on Database and Expert Systems Applications*. Springer-Verlag.

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.