Process Fact Extraction from the Web

Stephanie Buhagiar, Mike Rosner, Charlie Abela
Deptartment of Artificial Intelligence
University of Malta
sbuh0001,mike.rosner,charlie.abela@um.edu.mt

May 16, 2008

Abstract

The web contains large quantities of information regarding processes running on our computers. In this work, we apply Ontology Based Information Extraction to the task of building profiles of executable files and dynamic link libraries. We assign a concept to each sentence in the process description to obtain a summary of relevant concepts for each process. Such concepts include 'System', 'Backdoor' and 'Mail Propagation' among others. We use Natural Language Processing techniques combined with Machine Learning techniques for Ontologies to classify our description. Similar process descriptions are combined, thus leveraging redundancy to increase reliability rather than information overload. We designed the system to improve with user feedback. Our system obtained satisfactory results overall, when compared to a human doing the same extraction task.

1 Introduction and Background

A user who experiences degraded performance and errors on his computer, will often try to discover the source of the problem by opening the Task Manager and checking out the running processes. He will usually query the web for any unfamiliar process name to identify whether it is harmful or not. Multiple sources on the web provide this information. However, processes running on our computers cannot be uniquely identified by their filename. We take a textual approach to identifying the different types of processes which have the same filename. Viruses are also given different names by different vendors. Thus, grouping together descriptions which in essence refer to the same virus is useful in reducing information overload.

1.1 Web Information Extraction

Our virus desciptions are obtained from the web. Numerous approaches have been applied to extracting information from the web. Redlingshofer [8] proposes

a method for identifying rich text areas on a web page. Ciravegna et al [9] uses combined weak extraction strategies such as shallow parsing based on regular expressions to achieve a higher accuracy rate when extracting information from departmental websites. We keep this philosophy in mind when designing our extraction module.

1.2 Ontologies

In the advent of the Semantic Web, and the importance of defining domain knowledge in ontologies, we base our approach on ontology learning methods for information extraction. Though ontologies about computer programs and malware can be found in Weaver et al [3] and Eden et al [4], none describe the domain in terms of the properties and features we would like to extract. We therefore define our own ontology, using the ontology design guidelines from Noy et al [1] and Gruber [2].

1.3 Sentence Classification

In this work, we make use of a Sentence Classifier to select the relevant sentences from the description and classify them according to our ontology. Khoo et al [7] has also performed experiments regarding feature extraction for sentence classification. Wiemer et al [5] compare sentences to be able to rate students' answers in an Intelligent Tutoring System. As their sentences also tend to be unstructured a bag of words approach is used to compute sentence similarity. We evaluate our results with respect Khoo's work in Section 4.1.

2 Aims and Objectives

The aim of our research is to gather information about processes (executables and dynamic link libraries) from the web and build a profile to describe the properties of each process. When a filename is submitted to our system, we would like to present the user with a list of profiles which are about the filename. Each profile describes a safe or harmful process with a list of concepts or keywords. The original description, annotated with our concepts, will also be made available to the user. The user will only be required to evaluate the facts in our profile. The information will be extracted from the descriptions with the help of an ontology. Our aim is to obtain information from technical descriptions, while developing a system which can be extended to process high level descriptions, short user comments and forum entries. Thus we keep in mind that the information we need is not always available in correct structured English. Process descriptions for the same file name are numerous, thus we will combine very similar descriptions to ease the problem of information overload.

Here we outline our objectives to reach the designated aims:

1. Obtain and tag high quality training data from the web

2. Develop an ontology which adequately describes the process domain
3. Use ontology learning techniques to classify concepts in text
4. Recognise sentences which constitute a concept, as against sentences which don't
5. Assign an overall label to the profile, for example 'Safe' or 'Harmful'
6. Combine similar process descriptions

3 Design

Our design, as hinted in Figure 1, uses a number of extraction strategies in parallel. The final profile is categorised as 'Safe' or 'Harmful' by running a Naive Bayes Classifier over a vocabulary of words extracted from the information gathered by our Extraction Modules. The vocabulary is defined in our ontology which is discussed in Section 3.1 below. In this way, we obtain our final classification based on the most reliable information identified on the page.

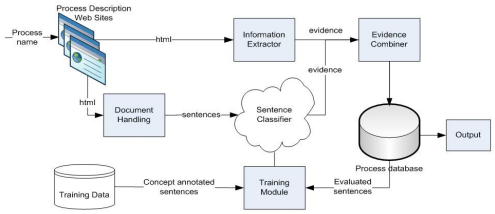

Figure 1: System Architecture

3.1 The Ontology of Processes

Our ontology of processes is composed of two sub-ontologies:

- Process Domain Ontology defines the classes of processes such as 'Dialler', 'Keylogger', 'System' and 'Safe' among others.
- Process Range Ontology defines process characteristics such as 'Modifying the Registry', 'Autostart', 'Keylogging' and 'Email Propogation'.

These two sub-ontologies are connected with the has-a property. As the name indicates, the Process Domain Ontology defines the domain classes of the has-a property, while the Process Range Ontology defines the range classes of the has-a property. Thus if we had to represent our virus profiles in an ontology, each Process Class would have a number of properties, for example one will expect a 'Keylogger' process from the Process Domain Ontology to have a property has-a where the target class is 'Keylogging' from the Process Range Ontology.

3.2 Web Information Extraction Module

Process Descriptions tend to start with a number of labels of the form 'Type: Worm' or 'Filename lsass.exe'. For this reason, we designed a Label Recogniser, which takes a list of typical label names we are interested in, and extracts similar looking labels. We use a modification of the Levenshtein distance algorithm to extract similar labels. If we would like to identify 'Sub-type', we will also extract 'SubType' because the two strings have a small edit distance. We also employ a Sentence Pattern Recogniser to identify common sentence patterns in text.

3.3 Ontology Based Information Extraction Module

Our sentence classifier is based on the Adapted Hieron Algorithm. This is a large margin learning algorithm which classifies instances with respect to an ontology. This Algorithm is discussed in detail in Li et al [6]. The philosophy behind it is that if it cannot classify an instance correctly, it will classify it as close as possible, to a parent or sibling node. Our Process Range Ontology was modified to include an 'Irrelevant' subtree which enables us to use Ontology-based evaluation measures as described in Li et al.

3.4 Combining Evidence

The last stage of the extraction process is to decide whether the description is about a safe or harmful process. Features from our ontologies which are present in extracted information are given to the Naive Bayes Classifier which gives its predictions as either 'Safe' or 'Harmful'. The decision is also supported by a number of rules.

Finally we store our descriptions in a database. We design our system to be easily adapted to the user. The Process Range Ontology can be changed and the classifier retrained. We also provide a user interface so that the user can correct the profiles and evaluate the concepts assigned to sentences. Once the sentences are evaluated, they are collected to be used as training data.

4 Results

We started the evaluation of our system by looking at the results of the individual modules as well as the accuracy of the system as a whole. The results are presented in the following sub-sections.

4.1 Sentence Classifier

We evaluated our Sentence Classifier with the standard precision and recall measures and we used the Ontology-based precision and recall methods defined in Li et al [6]. We also propose our own modification of the Ontology-based evaluation measures to assess whether the Adapted Hieron succeeds in classifying

Evaluation Method	Precision	Recall
Standard	59%	49%
Ontology Based	94%	79%
Modified Ontology Based	76%	66%

Figure 2: Comparision of Results of various Evaluation Measures

as close as possible to the correct node. Our proposed measure achieves reasonably higher values when compared to the standard precision and recall, thus indicating that the sentences which were not classified correctly were classified to some nearby node. We attribute our low scores for standard precision and recall to the lack of sufficient training data. Our labelled sentence sets can be classified as small when compared to better performing sentence classification systems in Khoo et al [7]. Our results are presented in Figure 2.

4.2 Naive Bayes Classifier

The Naive Bayes Classifier achieves 94% overall precision and recall when predicting whether a description is safe or harmful. The use of rules increased the accuracy of the process category predictions to 100%.

Filename	Original Number of Desc	No of Profiles	Virus names	Files	OS Affected	Type	Concepts Identified	Overall Rating
svchost.exe	18	5	100%	100%	76%	85%	85%	81%
lsass.exe	5	2	100%	100%	80%	100%	80%	73%
csrss.exe	4	1	100%	100%	57%	100%	37%	65%
mstask.exe	4	2	100%	100%	92%	100%	68%	65%

Figure 3: Overall System Evaluation for Harmful Processes

The overall evaluation for the system was carried out on 31 harmful descriptions and 4 safe descriptions belonging to 4 different filenames. Profiles were extracted manually, then compared to the system-generated profiles. The evaluation for harmful process profiles is shown in Figure 3. The Original Number of Descriptions to No of Profiles ratio shows the effect of grouping by virus aliases. All groups were created correctly when compared to the groups created by our evaluator.

From our evaluation we found out that our system always missed a single concept from the list identified by the user. When the number of concepts is small, this had a great effect on precision as is the case with 'csrss.exe' in Figure 3. However, a similar concept according to our ontology, was assigned instead of the missing one, which credits our learning approach.

When evaluating safe process profiles, 'Company' and 'Filename' were always identified correctly while an incorrect value for 'Product' was given to one process from the 4 processes evaluated.

5 Conclusions

We deem our approach successful as by leveraging intra and inter description redundancy, we manage to achieve reasonably better accuracy for extracted information in our profiles, than if we had considered the results of the Sentence Classifier alone. We significantly reduce the time taken to obtain process information by reading profiles instead of original descriptions. For future work, we plan to increase our training dataset and assign multiple concepts per sentence. We also plan to include spyware and adware among our classifications.

References

[1] Noy, Natalya F. and McGuinness, Deborah L. : Ontology Development 101: A Guide to Creating Your First Ontology. (2001)

[2] Gruber, T. R.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. Formal Ontology in Conceptual Analysis and Knowledge Representation. Kluwer Academic Publishers, Deventer, The Netherlands (1993)

[3] Weaver, Nicholas and Paxson, Vern and Staniford, Stuart and Cunningham, Robert: A taxonomy of computer worms. WORM '03: Proceedings of the 2003 ACM workshop on Rapid malcode. ACM Press, New York, NY, USA (2003) 11–18

[4] Amnon H Eden, Raymond Turner: Problems in the Ontology of Computer Programs. Applied Ontology Volume 2, IOS Press, Amsterdam (2006)

[5] Peter Wiemer-Hastings and Iraide Zipitria: Rules for Syntax, Vectors for Semantics. University of Edinburgh, Edinburgh, Scotland (2001)

[6] Li, Yaoyong and Bontcheva, Kalina and Cunningham, Hamish: Hierarchical, Perceptron-Like Learning Ontology Based Information Extraction. WWW 2007 Semantic Web Track. Sheffield, UK (2007)

[7] Khoo, Anthony and Marom, Yuval and Albrecht, David: Experiments with Sentence Classification. Proceedings of the 2006 Australasian Language Technology Workshop Australia (2006) 18–25

[8] Redlingshofer, Leopold: Data Extraction and Label Assignment For Web Databases. 12th International WWW Conference. (2005)

[9] Fabio Ciravegna and Alexiei Dingli and David Guthrie and Yorick Wilks: Integrating Information to Bootstrap Information Extraction from Web Sites. In IJCAI 2003. (2003)