# Dictionary Search Algorithm for Job Ads Profiling

1st Alex Wittner
*Department of Telecommunication*
*Brno University of Technology*
Brno, Czech Republic
230914@vut.cz

2nd Marek Sikora
*Department of Telecommunication*
*Brno University of Technology*
Brno, Czech Republic
marek.sikora@vut.cz

3rd Sara Ricci
*Department of Telecommunication*
*Brno University of Technology*
Brno, Czech Republic
ricci@vut.cz

***Abstract*—This article describes a novel way for developing an assisting module for analyzing cybersecurity skills in job advertisements. This module will be included in the Cybersecurity Job Ads Analyzer web application, which will be used to examine cybersecurity skill requirements in job opportunities. Our idea speeds up the development of the test database utilized in the analysis by eliminating the need for manual data entry. To that end, various dictionary search algorithms have been examined, and a comparison of four dictionary construction methods is offered.**

***Index Terms*—Dictionary search algorithm, cybersecurity skills analysis, ENISA framework.**

## I. Introduction

Cybersecurity has become more prominent as a result of the Information Technology (IT) industry's rapid development and expansion. The lack of a uniform taxonomy of cybersecurity skills and abilities, however, brought a lack of acknowledgement of the skills needed in cybersecurity, and consequently, a lack of effective coordination of training activities. The recently published European Union Agency for Cybersecurity (ENISA) framework [1] describing 12 cybersecurity profiles gives a common ground for skills classification. In fact, the profiles are described through key skills and key knowledge which can be mapped to work roles.

Dictionary search algorithms [2] can help in the identification of cybersecurity skills in specific work roles by seeking specific words in the job advertisement description. Each work role described in an advertisement can be mapped to an ENISA profile that can be seen as a collection of skills. With the identification of which skills are suggested in work roles, it is possible to compare the job market needs and what is taught in education.

### A. Contribution

In this article, we compare several dictionary search algorithms, select the one that gives the best results for our use case, and propose an extension that increases its accuracy. We then deploy the selected algorithm to enhance the functionality of a free web-based tool called the Cybersecurity Job Analyzer [3] developed in the Cybersecurity Skills Alliance – A New Vision for Europe (REWIRE) project [4]. The app allows cybersecurity skills need analysis, but currently requires manual insertion of new job advertisements where the user decides on the occurrence of each skill. The deployment of our algorithm allows to simplify this entry by automatic identification of present skills. Therefore, a user will only need to insert the Uniform Resource Locator (URL) of the job advert he would like to upload and fill in the basic information, and the rest of the information will be automatically filled by our module.

The rest of this article is organized as follows. Section II describes in more detail the ENISA framework, REWIRE Skills Groups, and the most appropriate dictionary search algorithms. Section III describes our implementation, i.e. the keyword dictionaries creation and the algorithm that is used within the application. Section IV presents the comparison of dictionary search algorithms and the experimental results with statistical analysis of 110 job advertisements. The final section contains our conclusions.

## II. Preliminary

In this section, we discuss the theoretical background that is crucial for the understanding of our implementation. First, we review the ENISA framework describing the cybersecurity profiles. Second, the mapping of the ENISA framework and REWIRE Skills Groups, i.e., skills, is shown. This map is strictly necessary to understand how a dictionary is created. At last, the compared dictionary search algorithms are sketched.

### A. ENISA Framework

The European Cyber Security Framework (ECSF) [1] is designed to provide individuals, employers and training providers in the European Union (EU) member states information and a common understanding of what is relevant for cybersecurity profiles. This is possible due to their identified titles, missions, tasks, skills, knowledge. 12 cybersecurity profiles are identified: Chief Information Security Officer (CISO), Cyber Incident Responder, Cyber Legal, Policy & Compliance Officer, Cyber Threat Intelligence Specialist, Cybersecurity Architect, Cybersecurity Auditor, Cybersecurity Educator, Cybersecurity Implementer, Cybersecurity Researcher, Cybersecurity Risk Manager, Digital Forensics Investigator, and Penetration Tester.

### B. REWIRE Skills Groups

In [3], a total of 31 skills were selected with the help of detailed analyses and comparisons. These skills were created from the ENISA framework [1] by grouping the ENISA key skills and key knowledge into 31 Skills Groups. We refer to REWIRE Deliverable R3.4.1 [5] for more details.

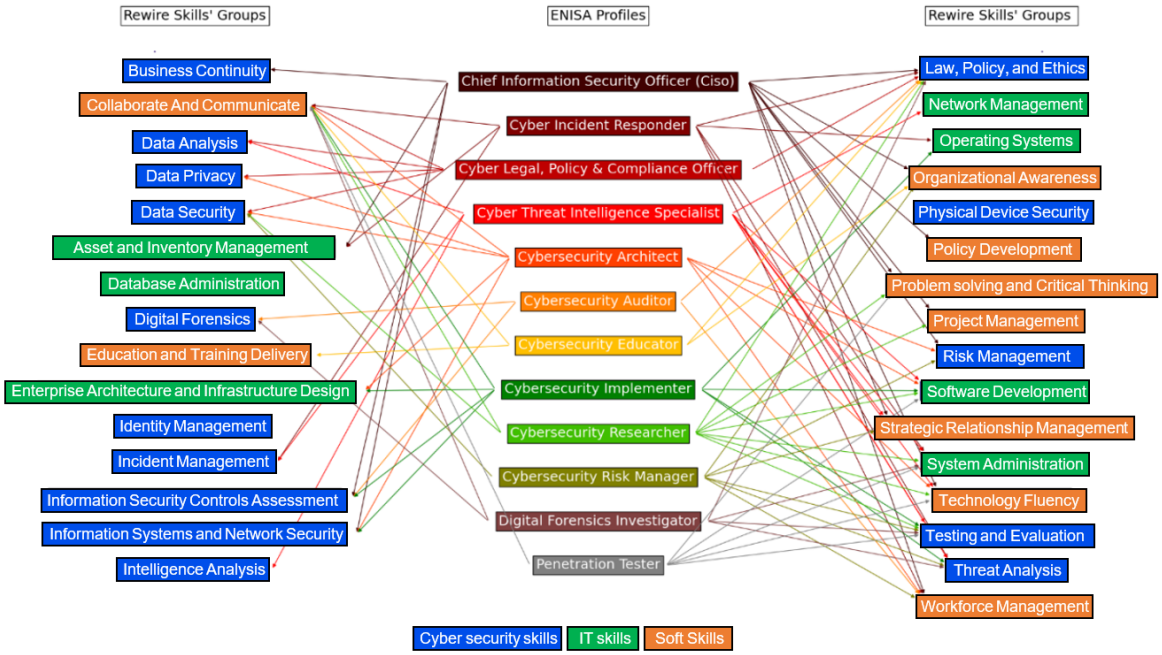


Figure 1: All 31 REWIRE Skills Groups and their connection with ENISA and the categories to which they belong

These skills can be split into 3 categories: 1) Cybersecurity Skills, 2) IT Skills, and 3) Soft Skills. IT Skills are more basic and core information technology expertise, whereas Cybersecurity Skills are those directly related to cybersecurity work roles. Non-technical knowledge is referred to as Soft Skills. Since they are vital for the successful performance of cybersecurity work duties, IT and Soft Skills should also be regarded as essential. It is important to keep in mind that cybersecurity is a multidisciplinary topic and calls for expertise outside of computer science. Figure 1 shows the 31 REWIRE Skills Groups and it is possible to see the interrelationships of the groups and which ENISA profiles.

### C. Aho-Corasick Algorithm

Aho-Corasick (AC) algorithm [2] is a multiple-pattern searching algorithm based on pre-processing of patterns. Specifically, during the looping, all the patterns (finite set) for which the step is searched are worked on at the same time. The time complexity is in the worst case $\theta$(N+M+Z), where 'N' is the length of the text, 'M' is the length of keywords, and 'Z' is the number of matches. Given a dictionary of words, the searching algorithm can be divided into 4 steps:

1) **Create the trie** – a data structure similar to the binary tree with different features.
2) **Create the suffix (failure) links** – links that are used by the matcher when a character that cannot follow a trie edge occurs.
3) **Create output links** – each node contains information whether it is final or not.
4) **Implement a matcher** – use a sequential trie traversal and output all matches while using the output and suffix links.

### D. Commentz-Walter Algorithm

Commentz-Walter algorithm [6] consists of two very efficient algorithms, the aforementioned Aho-Corasick algorithm and the Boyer-Moore algorithm [7]. Commentz-Walter also works on the principle of trie and multiple pattern searches, supplemented by indexing. However, unlike Aho-Corasick, the trie is reversed. The matches in this algorithm are made using the principle on which the Boyer-Moore algorithm works, which is end-to-end traversal, hence the reverse trie. Commentz-Walter consists of two phases:

1) **Creating a reverse trie** – The procedure is the same as a trie, but the stamping works from "z" to "a" instead of from "a" to "z".
2) **Matching phase** – matching that works on the same principle as Aho-Corasick with a shifting technique derived from the Boyer-Moore algorithm.

## III. Implementation

This section describes the methodology applied to create the individual dictionaries and the reasons of your choices. Moreover, our proposed extension of the dictionary search algorithms is presented and their usability in the web application is also shown.

### A. Proposed Methods

A dictionary had to be created first, which was then used for the dictionary search algorithm. The accuracy of the search depends on the generated dictionary. 3 different dictionaries, where the words describing each REWIRE Skills Group were created, and then compared their performance. The elements of each dictionary were selected in 4 ways:

- **Expert input** – a preliminary dictionary was proposed by REWIRE partners with a few example words in every group.
- **The ENISA framework** – words are selected from ENISA key skills and knowledge descriptions. For instance, the ENISA Cybersecurity Risk Manager profile contains ”maturity models”, ”mitigate risks”, ”risk management tool”, and ”risk sharing options” that specifically identify the profile and can be added to the dictionary.
- **The National Initiative for Cybersecurity Education (NICE) Competencies framework** [9] – note that REWIRE Skills Groups are mapped to the NICE Competencies framework. As with the first method, the words were selected based on two parameters: uniqueness and expected occurrence in the competencies description. For example, for group Data Security we found the words ”confidentiality”, ”integrity”, and ”availability” in the related competencies definition.
- **Expert knowledge** – some words were chosen according to the author's experience with the given skill, for example in the Database administration the ”PostgreSQL” and ”phpmyadmin” words were added since they are key technologies used in this field. Other words were added based on reading existing job ads. And finally synonyms were used to already existing added words.

The dictionaries consist of:

- **One-word Dictionary** – consists of a single word only. For example, words like Device, Physical and Remote.
- **Two-word Dictionary** – contains objects consisting of two-words. This dictionary allows a more strict description of each Skill group. For example words: cluster analysis, anomaly detection, data mining.
- **Weighted Dictionary** – two sub-dictionaries were created: one with a specific word that unequivocally identifies the group, and one with one-word related to the specific group. For example words: Windows, Linux in group Operating systems with 100% weight and from same category words: system, systems, operating with not 100% weight.

Note that the Weighted Dictionary was generated after comparing the first two dictionaries. In fact, the testing showed that the One-word Dictionary permits recognizing more skills than the Two-word one. However, the One-word Dictionary found many groups that should not have been found. Therefore, the Weighted Dictionary $1^a$ (Table I) was created. The two sub-dictionaries were then used in a word search program with the following logic: if a word from the 100% dictionary appears in the text, then the group to which the word belongs is set to 1. If a word from the NOT 100% dictionary appears in the text, then a value of the $1/$number of words in that category is added to the auxiliary variable of each group. Then the auxiliary variables are examined and if their value is higher than 0.5 (50 %) the group number is set to 1.

This method of using two dictionaries was subsequently improved and that's why Weighted Dictionary $2^b$ (Table I) was created since if a certain group contained a lot of words, the chances of 50 % of the words appearing were very small. Therefore, the following logic was applied: if 3 words from a given group occur in the text, then the group to which the words belong is set to 1 if there are less than 5 words in the group, we work with 50 % accuracy.

### B. Identifying Skills in Cybersecurity Job Ads

The previous steps of creating a dictionary and a comparison algorithm will then be used in the implementation of the existing web application. The web application provides the possibility of inserting new job ads and manual skills analysis is needed. After the final implementation of dictionaries and algorithms, the insertion of new job ads will be much easier. The authorized user will only need to insert the Hypertext Markup Language (HTML) link and confirm either the presence or not of a REWIRE Skills Groups. Figure 2 depicts a sketch of the analysis helper of the web application. As can be seen, the user will be given a preset value for the given group, according to the criteria that have been set. The user is able to view the words that have been found in the given group and at the same time they are highlighted, using the arrows it is possible to navigate to the next part of the text where others found words can be.

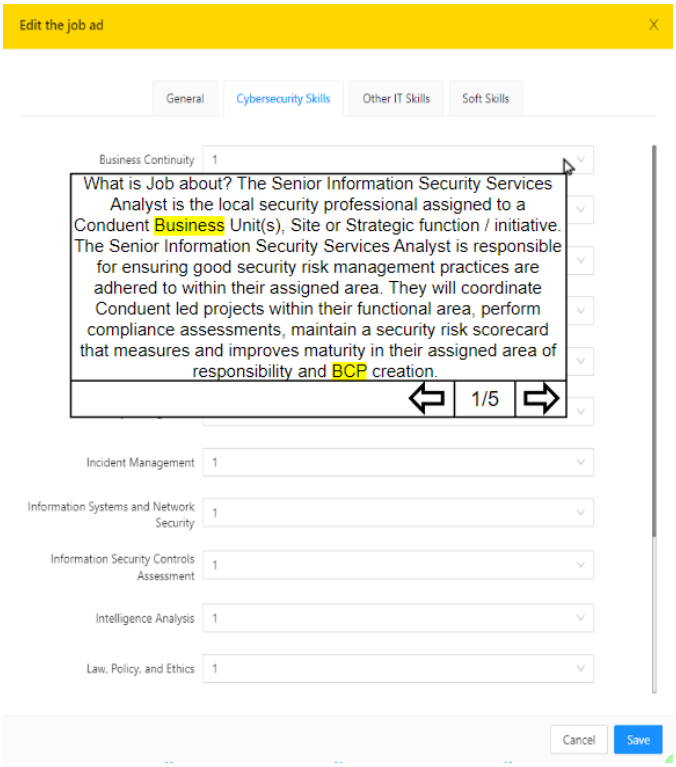


Figure 2: Sample of web application

## IV. Experimental Results

This section describes the results of our benchmarking. It compares which dictionary algorithm was the most suitable for this work. It also shows which of the dictionaries used has

the best results in terms of accuracy and why accuracy cannot be evaluated only by positive matches.

### A. Comparison of Dictionary Search Algorithms

Since it is necessary to search and find matches as quickly as possible (in order to work with the web application), algorithms were chosen for their speed and reliability. These algorithms belong to the group of multipattern searching algorithms. As the name suggests, these algorithms can search for several keywords (patterns) in a text at the same time.

Based on the study [8], Commentz-Walter is faster in cases where the length of the patterns is longer. The span of a long pattern cannot be determined directly, but patterns that contain more than 20 characters can be considered a kind of breakpoint. The range of 20-120 characters per pattern is given as the span of a long pattern. There is no guarantee that once the length of 20 characters is reached, Commentz-Walter is automatically better. Another important point is the number of patterns, as the Commentz-Walter algorithm becomes less efficient when working with more than 13 patterns. In our search, there are patterns with an average length of 8.2 characters and the current number of patterns is 515. According to the theory, the Aho-Corasick algorithm should be faster.

### B. Comparison of Proposed Methods

In this section, we compare 4 generated dictionaries on their accuracy. At the test database, we ran the analysis on 110 job ads that were also analyzed manually. The results are shown in Table I. In the table, two kinds of accuracy have been considered: 1) the accuracy for binary classification:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN},$$

where $ACC$ stays for accuracy, $TP$ for true positives, $TN$ for true negatives, $FP$ for false positives, and $FN$ for false negatives, and 2) since the application involves‘ to the user to confirm the group, the $TP$ has more relevance than the other category, therefore, we consider this value alone.

When we check the $ACC$ value, the Weighted Dictionary $2^b$ produces the higher matches, whereas, if we consider only the $TP$ matches, One-word Dictionary performs considerably better than the others. Therefore, in case the dictionary search is the final output of the analysis, the Weighted Dictionary $2^b$ method needs to be deployed. In our case, where a user checks will confirm the selection, we are interested in the higher $TP$ matches. In fact, a user can easily discard the $FP$ values.

| Name of Dictionary | TP (%) | ACC (%) |
|---|---|---|
| Two-word Dictionary | 40.9 | 58.6 |
| One-Word Dictionary | 71.7 | 59.6 |
| Weighted Dictionary $1^a$ | 53.87 | 63.17 |
| Weighted Dictionary $2^b$ | 58.69 | 63.37 |

Table I: Dictionaries results
[a] match with half of the words in the group
[b] match with 3 words in the group

## V. Conclusion

In this article, we presented a methodology for creating a helping module for the analysis of skills in job advertisements. In order to select the most suitable algorithm to identify skills, we ran a comparison of existing dictionary search algorithms. Accordingly to the presented comparison, the best pattern-searching algorithm for our purposes is the Aho-Corasick multipattern searching algorithm since this work is completely dominated by a larger number of short keywords, which are more advantageous for this algorithm than for the Commentz-Walter algorithm.

After selecting Aho-Corasick multipattern searching algorithm, 4 dictionaries were created where also weights were assigned to the word in each group. Note that our helping module allows the pre-filling of the skills that are found in the job advertisement, and it is up to the user to check the validity of the finding. The experimental results pointed out that the most convenient dictionary is the One-Word Dictionary. Possible future work is to improve the accuracy of the protocol by working on the weight and the dictionary and exploring new techniques.

## Acknowledgment

The following funding source is gratefully acknowledged: the ERASMUS+ Sector Skills Alliance (SSA) programme of the European Union (grant 621701-EPP-1-2020-1-LT-EPPKA2-SSA-B 'REWIRE').

## References

[1] The European Union Agency for Cybersecurity. (2022, Sep. 19). *European Cybersecurity Skills Framework Role Profiles* [online]. Available at: https://www.enisa.europa.eu/publications/european-cybersecurity-skills-framework-role-profiles.

[2] GeeksforGeeks. (2022, Jun. 14). *Aho-Corasick Algorithm for Pattern Searching* [online]. Available at: https://www.geeksforgeeks.org/aho-corasick-algorithm-pattern-searching/.

[3] Ricci S, Sikora M, Parker S, Lendak I, Danidou Y, Chatzopoulou A, Badonnel R, and Alksnys D. "Job Adverts Analyzer for Cybersecurity Skills Needs Evaluation," in *Proceedings of the 17th International Conference on Availability, Reliability and Security*, 2022 Aug 23, pp. 1-10.

[4] REWIRE - Cybersecurity Skills Alliance. (2022). *REWIRE Project Results* [online]. Available at: https://REWIREproject.eu/results/.

[5] REWIRE - Cybersecurity Skills Alliance: "R3.4.1 Mapping the framework to existing courses and schemes (Chapter 5.4)" [online], 2022, Available at: https://REWIREproject.eu/wp-content/uploads/2022/11/REWIRE_R3.4.1_Deliverable-v7-Final.pdf.

[6] Wikipedia contributors. (2023, Jan. 30). *Commentz-Walter algorithm* [online]. Available at: https://en.wikipedia.org/wiki/Commentz-Walter_algorithm.

[7] GeeksforGeeks. (2022, Nov. 9). *Boyer Moore Algorithm for Pattern Searching* [online]. Available at: https://www.geeksforgeeks.org/boyer-moore-algorithm-for-pattern-searching/.

[8] Dewasurendra S, Vidanagamachchi S. (2018). "Average time complexity analysis of Commentz-Walter algorithm". *Journal of the National Science Foundation of Sri Lanka* [online]. Sri Lanka, Available at: http://doi.org/10.4038/jnsfsr.v46i4.8630.

[9] National Initiative for Cybersecurity Education (NICE) — NIST. (2020). *The Workforce Framework for Cybersecurity (NICE Framework)* [online]. Available at: https://www.nist.gov/itl/applied-cybersecurity/nice/nice-framework-resource-center/workforce-framework-cybersecurity-nice#currentversion.