Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 18 No. 14 (2024) | OPEN ACCESS

https://doi.org/10.3991/ijim.v18i14.47901

PAPER

A Textual Content Analysis Model for Aligning Job Market Demands and University Curricula through Data Mining Techniques

Ylber A. Januzaj[1], Driton Sylqa[1](✉), Artan Luma[2], Luan Gashi[2]

[1]University "Haxhi Zeka", Peja, Kosovo

[2]South East European University, Tetovo, North Macedonia

driton.sylqa@unhz.eu

ABSTRACT

Addressing the growing disparity between job market demands and the availability of skilled workers, particularly in the technology sector, is a critical challenge in numerous countries. This study introduces a model that assesses the alignment between job market requirements and university curricula, primarily through textual content analysis. Initially, this research illustrates the operational framework of the model through graphical representation. Subsequently, the proposed techniques vital to the functionality of this model are delineated. Specifically, the integration of data mining techniques is employed for the automated extraction of relevant information from both labor market demands and university curricula. An integral aspect of this study involves outlining the methodology for creating the dataset. This phase is essential as it lays the groundwork for further stages, notably the implementation of code to generate comprehensive results. The findings of this study reveal significant insights into the alignment between job market demands and university curricula. Through textual content analysis and data mining techniques, patterns and discrepancies between the two domains are identified, shedding light on areas for improvement in educational provision. Conclusions drawn from this research underscore the importance of bridging the gap between workforce requirements and educational provisions. By leveraging data-driven approaches, educators and policymakers can make informed decisions to enhance curriculum development and better prepare students for the demands of the job market. The impact of this research extends to both academia and industry, offering actionable insights for curriculum alignment and workforce readiness initiatives. Ultimately, this model contributes to the advancement of educational practices and the enhancement of workforce productivity in response to evolving industry needs.

KEYWORDS

machine learning, web crawling, data mining, dataset, job vacancy, university curricula

Januzaj, Y.A., Sylqa, D., Luma, A., Gashi, L. (2024). A Textual Content Analysis Model for Aligning Job Market Demands and University Curricula through Data Mining Techniques. *International Journal of Interactive Mobile Technologies (iJIM)*, 18(14), pp. 164–176. https://doi.org/10.3991/ijim.v18i14.47901

Article submitted 2024-01-12. Revision uploaded 2024-03-03. Final acceptance 2024-03-04.

© 2024 by the authors of this article. Published under CC-BY.

1 INTRODUCTION

The utilization of machine learning (ML) techniques has significantly revolutionized computational processes, facilitating swifter and more precise outcomes in computer calculations [1]. While manual analyses were prevalent previously, they exhibited substantial shortcomings in terms of both time efficiency and accuracy [2]. Consequently, the imperative integration of ML techniques emerged as a necessity within the framework of our model's application [1, 3].

Extensive efforts have been made to analyze the alignment between labor market demands and university curricula. Nevertheless, there remains an ongoing pursuit to enhance the accuracy and efficiency of these analyses. This study aims to introduce a model that utilizes ML techniques to address this persistent challenge.

The evolving landscape of the job market often presents a significant disparity between the skills demanded by employers and the competencies fostered within university curricula. Rapid advancements in technology and changing industry needs further exacerbate this mismatch, leading to challenges in workforce readiness and employability. Graduates may find themselves ill-equipped to meet the evolving demands of the job market, while employers struggle to find candidates with the requisite skills.

This disparity underscores the necessity for closer alignment between university curricula and industry requirements. Bridging this gap requires a thorough understanding of the specific skills and knowledge areas valued by employers, as well as the ability to integrate these insights into educational programs. Addressing this challenge is essential not only for the success of individual graduates but also for the overall competitiveness and innovation capacity of industries.

This study is motivated by the persistent challenge of aligning university curricula with the dynamic demands of the job market, especially in industries such as technology, where rapid advancements require current skillsets. The disparity between what graduates provide and what employers require often leads to mismatches and inefficiencies in the labor market. Therefore, there is an urgent need for innovative approaches that can effectively bridge this gap and ensure graduates are well-prepared for the workforce.

Furthermore, the significance of this study extends beyond individual career outcomes to broader economic and societal implications. A well-aligned education system not only benefits graduates in terms of employability and career advancement but also enhances overall productivity and competitiveness within industries. By fostering a symbiotic relationship between academia and industry, this research aims to contribute to the overarching goal of fostering sustainable economic growth and development.

The initial phase of this study required identifying the necessary data essential for constructing the model. Since a significant amount of relevant data is publicly available on web pages, an automated data extraction approach was considered practical. Using web crawling techniques was recognized as the most efficient method to automatically retrieve this data from relevant websites [4].

The methodology employed in crafting the model, as expounded in this paper, encompasses several sequential stages. It begins with importing pivotal libraries essential for conducting the analysis. Subsequent steps involve identifying words and determining their frequencies within the dataset. Following this, normalization techniques are applied to prepare the dataset for comparison. The culmination of this process involves using comparison techniques to determine the final results, facilitating a comprehensive comparison between the textual documents representing labor market demands and university curricula.

2 METHODOLOGY

2.1 Introduction

Efforts to address the alignment between labor market demands and university curricula have involved various analytical approaches [5]. However, prevalent manual analyses, while applicable in scenarios with dynamic datasets, suffer from limitations in delivering precise and comprehensive results, primarily due to constraints related to time and cost efficiency.

2.2 Model design and objective

Consequently, the proposed solution to this issue involves implementing an automated model specifically designed to compare labor market requirements with university curricula. This model utilizes mathematical computations carried out through specialized algorithms in the Python programming language, with the goal of providing accurate alignment insights between the two entities.

The primary objective of this model is not only to address the immediate challenge at hand but also to provide a versatile framework applicable across diverse domains. This chapter delineates the design aspects of our model, offering illustrations across various phases. These phases encompass the conversion of counts into numerical values and culminate in the derivation of comprehensive analyses from our model.

2.3 Data collection and corpus formation

An inherent characteristic of our model is its capability to concurrently analyze multiple documents of varying lengths. Each document undergoes an individualized analysis against every other document. The development of a corpus of data is paramount for normalizing values used in the analysis.

This corpus is established through the extraction of specific words obtained from textual content acquired from websites disseminating information about job openings. Additionally, a segment of the analysis pertains to regional universities, utilizing published curriculum information in the technology domain offered by these institutions.

2.4 Model implementation

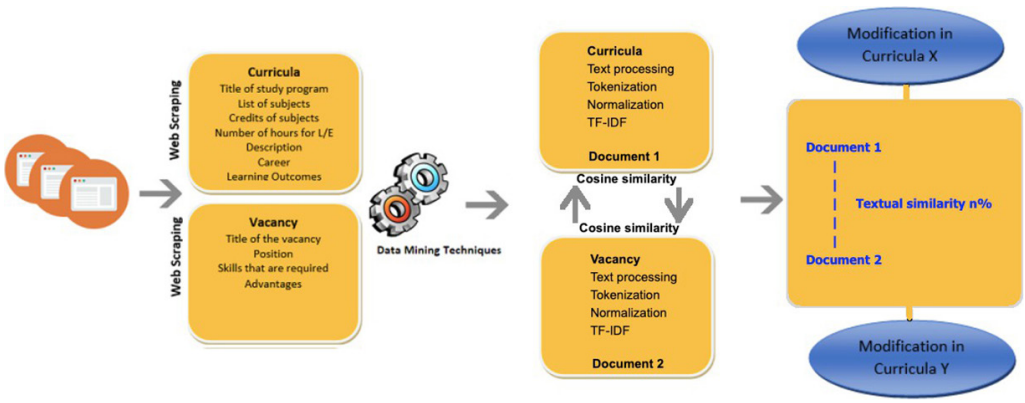

Fig. 1. Automated model sketch

Figure 1 illustrates the schematic representation of our automated model designed for comparing labor market demands with university curricula. The core structure of the original model remains unchanged, with the difference lying in the methodologies used for comparison. The techniques incorporated into our model are derived from extensive research in the field of machine learning.

The initial phase involves identifying websites that disseminate information about technological job opportunities and the curricula of relevant university programs earmarked for research inclusion.

Following the website identification process, a web scraping algorithm is applied to systematically traverse these websites and extract specified information accordingly.

Subsequently, data processing ensues, involving the elimination of extraneous spaces and rows within the data, along with the removal of special characters for data uniformity.

2.5 Text analysis and comparison

Proceeding to the subsequent phase, tokenization and normalization of the text are conducted. This process involves transforming each word into vector values using term frequency-inverse document frequency (TF-IDF).

The culmination of this process involves the application of cosine similarity between nearly prepared documents, resulting in the assessment of textual similarity based on shared words within the documents. These results provide recommendations for program design alterations to better align with labor market demands.

The comprehensive analyses derived from these processes enable informed recommendations for program adjustments, enhancing their compatibility with prevailing labor market requirements.

3 DATASET FORMATION: UTILIZING WEB CRAWLING TECHNIQUES

3.1 Web crawling process

Web crawling, also referred to as web scraping or web indexing, is a programmatic method that facilitates automated navigation through active websites [6, 7, 8]. The functioning of a web crawler involves systematically examining webpages for specified phrases or content criteria predefined by the user. Modern websites often leverage web crawling as an efficient mechanism to maintain the currency of their web pages [9, 10, 11].

One fundamental approach employed by web crawlers involves the automated visitation and subsequent downloading of web pages to a designated local storage destination. These downloaded web pages encompass a spectrum of content ranging from static to dynamic elements. However, the acquisition of dynamic content necessitates tailored scripting to ensure the comprehensive extraction of information. The complexity of scripts corresponds directly to the dynamic nature of the webpage, influencing the depth and breadth of information captured.

In the following section, we provide a visual representation illustrating the operational mechanics of web crawlers, explaining how they systematically retrieve content from web pages.

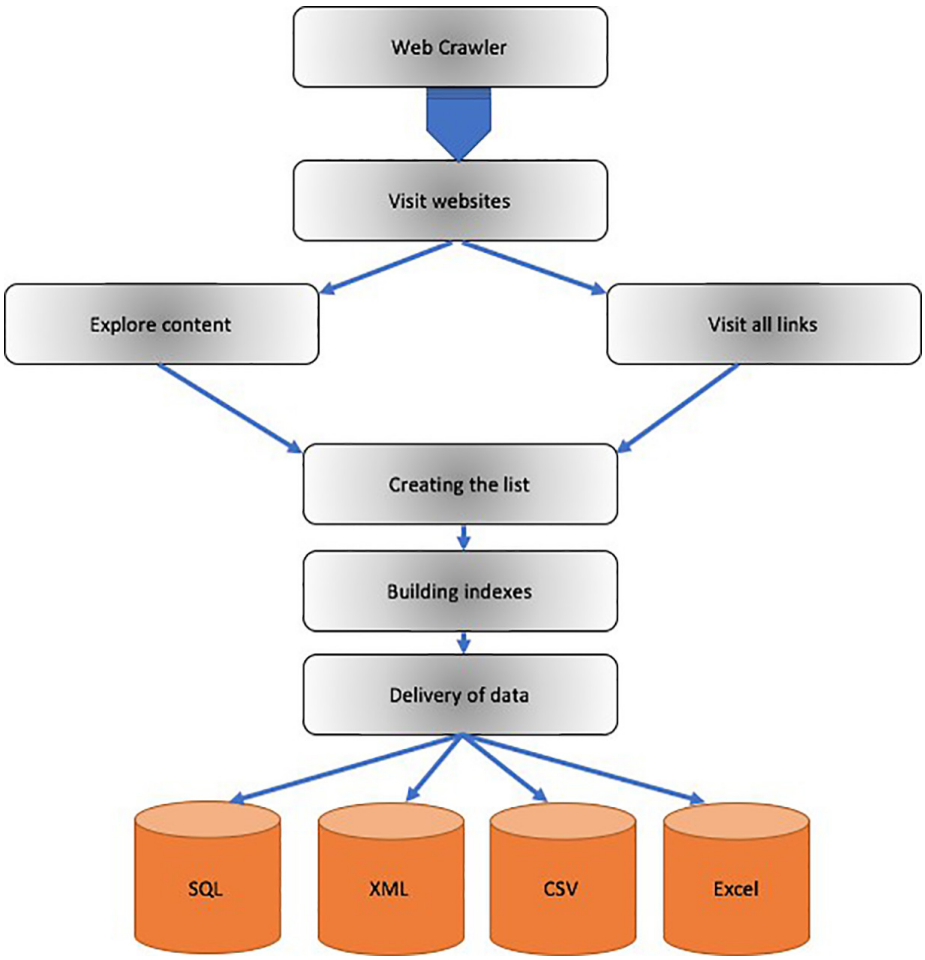

Fig. 2. Web crawling

Figure 2 delineates the sequential stages executed by the web crawler application during its browsing journey. Initiated by predefined website visits, the web crawler proceeds to explore the content based on designated phrases while concurrently traversing embedded links within the webpage.

Upon successful extraction of textual content, the crawler generates lists of these web pages, creating indexes for subsequent downloading and storage.

The final step involves archiving or delivering the extracted data, which can be stored in various formats such as SQL, XML, CSV, Excel, and others, customized to meet specific research requirements.

3.2 Objective of web crawling in research

The primary objective of web crawling in our research is to extract information published by websites related to job openings. This extraction focuses on specific keywords to enable subsequent clustering. Web pages specific to regions that contain data on university programs, including subject descriptions and program details, play the basis for clustering these educational offerings. This clustering serves a crucial role in aligning educational programs with market demands.

Additionally, web crawling extends to websites disseminating details about competitions offered by various companies. This focused extraction, relying on specific

keywords, reveals additional information regarding position requirements. Each position entails multifaceted knowledge prerequisites, encompassing programming, databases, and networking, among other domains. These diverse job descriptions contribute to assembling a comprehensive array of job requirements.

4 MODEL DEVELOPMENT ALGORITHM: METHODOLOGY AND COMPONENTS

The algorithm employed in crafting the model is segmented into distinct components, each tailored to address the unique requisites of both the labor market and university curriculum alignment. Commencing with the initial phase, the algorithm embarks on the integration of essential libraries indispensable for its execution. These libraries form the foundational framework upon which the algorithm is constructed, facilitating its subsequent operations.

Following this preliminary stage, a detailed exposition of each constituent part of the algorithm ensues, accompanied by comprehensive commentary elucidating the nuances and functionalities embedded within each segment. This systematic breakdown offers a comprehensive insight into the intricate workings of the algorithm, delineating its role in addressing the convergence between labor market demands and university curriculum requirements.

The discussion of model design is elaborated further to provide a clearer understanding of the research methodology employed. Instead of delving into technical discussions such as detailed source code explanations, the focus shifts towards explaining how the research design was carried out to obtain research results.

The research design process involved carefully considering the alignment of labor market demands and university curriculum requirements. The methodology employed included an initial phase of requirement analysis, followed by iterative design and refinement stages. This iterative approach allowed for the incorporation of feedback from stakeholders and experts in the field.

Visual aids, such as diagrams and flowcharts, were utilized to illustrate the research design process and model components. These aids helped make complex concepts more understandable to readers, aligning with the goal of providing a comprehensive explanation of the model design without delving into overly technical details.

The design decisions were informed by relevant literature and methodologies in the field. By contextualizing the research design within the broader field of study, the rationale behind specific design choices became clearer, and the alignment with research objectives was reinforced.

```
1  #Import of all libraries
2  from __future__ import division
3  from sklearn.feature_extraction.text import TfidfVectorizer
4  import math
5  import string
```

Fig. 3. Import of libraries of automated model

Figure 3 showcases the code segment responsible for importing crucial libraries essential for executing functions within our algorithm. At the onset, the algorithm integrates mathematical libraries alongside the TFIDF library from 'sklearn.' This TFIDF library remains pivotal throughout our algorithm, facilitating mathematical computations involving division operations and empowering text analysis functions through string manipulations via the 'string' module.

During the implementation phase of our model, the utilization of a corpus becomes imperative to normalize word values, especially for frequently used terms. This normalization process aims to adjust the weighting of commonly used words relative to less frequent terms, ensuring equitable representation within our algorithm.

The corpus itself comprises 'unigram' words, representing individual words extracted from textual content obtained from websites disseminating technology-related job offers. This dataset includes comprehensive information detailing word occurrences, their respective frequencies, and the percentage of their occurrence concerning the total words extracted from job offer web pages in the technology domain.

Table 1. Frequency and percentage of used words

Word	Frequency	Percentage
technology	3599	9.127%
analyst	1832	4.646%
information	1605	4.070%
security	1497	3.796%
support	1146	2.906%
senior	988	2.506%
end	662	1.679%
user	633	1.605%
software	628	1.593%
head	620	1.572%
development	614	1.557%
technician	570	1.445%

Table 1 illustrates a comprehensive list of words extracted from the compiled textual content obtained from web pages. Notably, the term 'technology' emerges as the most frequently used word, constituting over 9% of the entire corpus, indicating its prominence relative to other terms. Several other words exhibit substantial usage, with a frequency exceeding 500 occurrences within the document.

This 'unigram' corpus, composed of individual words, is slated for integration into our algorithm to facilitate normalization procedures. The aim is to harmonize the weighting of words used in our document by comparing them to this precompiled corpus. This integration is a crucial step towards achieving fair word representation within our algorithm.

In the subsequent sections, we will explain the methodologies used to seamlessly integrate this word list into our document, ensuring its optimal utilization for normalization purposes.

```
ngram_freq = {
    'technology': 3599,
    'analyst': 1832,
    'information': 1605,
    'security': 1497,
    'support': 1146,
}
```

Fig. 4. Ngram words

Figure 4 displays the filament corpus used to normalize word frequencies. This corpus, referred to as 'gram_freq,' consists of a list of words with their corresponding

frequencies. Within our code, these words play a crucial role in the division operation, normalizing each world's frequency relative to the total word count. The choice to develop a tailored corpus arises from the insufficiency of publicly accessible resources, such as those from Google, which contain numerous irrelevant words for our specific research context.

Following the importation of libraries and the creation of our tailored corpus, our algorithm proceeds to establish an information source crucial for comparison purposes. Leveraging the groundwork laid by our tailored corpus, we acquire job offers and university curriculum documents. These documents undergo text processing procedures, including the removal of special characters and irrelevant elements, ensuring their alignment with our analytical framework.

The meticulous preparation of this information source forms the bedrock of our comparative analysis, aligning job offers and university curriculum documents for subsequent assessments within our model.

```
#Tokenization of data, and transformin in lowercase
tokenize = lambda document: document.lower().split(" ")

#Source of our data. Both of documents, vacancy and curricula
vacancy_document = open("/Users/Ylber/Desktop/eurotech/vacancy_document.json").read()
curricula_document = open("/Users/Ylber/Desktop/eurotech/curricula_document.json").read()

#The documents which are used in our anaysis
used_documents = [
    vacancy_document,
    curricula_document,
]
```

Fig. 5. Documents that will be calculated

The procedures outlined in Figure 5 involve revealing information sources and applying the tokenization process to documents, which includes transforming all words into a uniform format. The documentation source disclosure specifies destinations for stored documents after processing procedures. Furthermore, the 'used_documents' variable is employed, representing a comprehensive list of documents designated for analysis within our model. It is important to highlight the scalability of the model, as it can accommodate an unlimited number of documents for comparative analysis.

Text tokenization comprises a tripartite process, beginning with the 'lambda' function, which takes the document as an argument. Subsequently, using this argument, the next step involves converting all words to lowercase. This conversion helps standardize word weights, ensuring consistency between capitalized and lowercase words. After this transformation, words are split using the 'split' function.

After these intricate procedures, our documents undergo preparation for TF-IDF computation, converting them into numerical or vector values. Subsequent sections explain the definitions of TF (term frequency), IDF (inverse document frequency) functions, and other functions that assist in normalizing word weights within the corpus.

```
#Definition of TF based on library
def TF_of_documents(word, tf_document):
    return tf_document.count(word)
```

Fig. 6. Definition of TF

Figure 6 encapsulates the section of the algorithm dedicated to defining the "term_frequency" mechanism, a pivotal process in the tokenization of documents. This procedure involves splitting each document into individual words and enumerating the frequency of each word's occurrence within the document.

Through the implementation of term frequency, the algorithm meticulously tokenizes the documents by parsing each word into discrete units and quantifying the number of times each word recurs within the document. This rigorous process serves as a foundational step in the comprehensive analysis of textual content, enabling a granular examination of word occurrences essential for subsequent computations.

```
#The process of normalization
def first_normalization_technique(word, tf_document):
    count = tf_document.count(word)
    if count == 0:
        return 0
    return 1 + math.log(1 + count / (ngram_freq[word] if word in ngram_freq else 1) * 1000 )
```

Fig. 7. Process of normalization

Normalization is crucial due to the frequent occurrence of words, known as 'stop words,' in the extracted files from different websites. Equating these words with others requires the use of two normalization techniques.

The primary normalization technique involves dividing the frequency of each word by the total number of times that word appears across the document corpus. Subsequently, this calculated value is multiplied by 1000 and incremented by 1. This normalization process ensures a standardized representation of word frequencies, mitigating the influence of commonly used words and facilitating equitable comparisons across the document corpus.

Upon completion of data normalization, the next step involves computing IDF, which quantifies the significance of each term within the document corpus.

```
#Definition of IDF based on library
def inverse_document_frequencies(tf_documents):
    idf_values = {}
    all_tokens_set = set([item for sublist in tf_documents for item in sublist])
    for tkn in all_tokens_set:
        contains_token = map(lambda doc: tkn in doc, tf_documents)
        idf_values[tkn] = 1 + math.log(len(tf_documents)/(sum(contains_token)))
    return idf_values

def tfidf(documents):
    tf_documents = [tokenize(d) for d in documents]
    idf = inverse_document_frequencies(tf_documents)
    tfidf_documents = []
    for document in tf_documents:
        doc_tfidf = []
        for word in idf.keys():
            tf = first_normalization_technique(word, document)
            doc_tfidf.append(tf * idf[word])
        tfidf_documents.append(doc_tfidf)
    return tfidf_documents
```

Fig. 8. Definition of IDF

Figure 8 elucidates the IDF calculation, highlighting the utilization of normalized words obtained through the previous normalization technique. Within our algorithm, the 'lambda' function conserves values computed during the initial normalization process. These preserved values serve as arguments for subsequent IDF mathematical computations.

The IDF computation involves the division algorithm between word frequency and the total number of words in the document. Notably, to prevent the generation of negative values, 1 is added to the resulting function output.

Upon successfully computing IDF values, the algorithm proceeds to calculate the TF-IDF by multiplying the TF score with the IDF score. This cumulative process signifies the initial phase of computing document similarities.

Subsequently, the declaration of functions responsible for computing cosine similarity, a technique that facilitates comparisons between documents of varying lengths, will be presented. Cosine similarity computations offer a robust method for assessing document similarities despite differing document sizes.

```
#Application of tfidfvectorizer
tfidf_from_sklearn = TfidfVectorizer(norm='l2',min_df=0, use_idf=True, smooth_idf=False, sublinear_tf=True, tokenizer=tokenize)
sklearn_score = tfidf_from_sklearn.fit_transform(used_documents)
```

Fig. 9. Declaration of tfidf vectorizer

Figure 9 illustrates the secondary technique used to convert initially declared documents into vector values. In this process, the information source labeled as 'used_document' is referenced, containing a subset of documents intended for conversion into numeric representations.

This technique is instrumental in transforming documents into vectorized numeric values, enabling a structured format for computational analysis. The 'used_document' subset represents a portion of the comprehensive document corpus prepared for conversion into numerical vectors, facilitating subsequent computational operations and comparisons.

```
def cosine_similarity(vector1, vector2):
    product = sum(p*q for p,q in zip(vector1, vector2))
    extent = math.sqrt(sum([val**2 for val in vector1])) * math.sqrt(sum([val**2 for val in vector2]))
    if not extent:
        return 0
    return product/extent
```

Fig. 10. Definition of cosine similarity

Now that our data has been successfully transformed into numeric or vector values, we are ready to execute the algorithm responsible for computing cosine similarity. As explained in the preceding illustrations, the cosine similarity calculation involves dividing the dot product of vectors by the product of their respective magnitudes. This resulting value serves as a comparative metric between both documents, indicating their similarity or dissimilarity.

Following the computation of cosine similarity, effectively capturing the outcomes from both techniques, the subsequent segment of the algorithm integrates these results for comprehensive comparison. In this final phase, an algorithm is presented to compare the outcomes derived from the TF-IDF and cosine similarity calculations.

This algorithm combines the results obtained from both techniques, enabling a thorough comparison between the documents. These results provide valuable insights into the levels of similarity or dissimilarity between the documents, offering a comprehensive assessment based on the TF-IDF and cosine similarity methodologies.

```
#Comparison between cosine similarity and tfidf
results_tfidf = tfidf(used_documents)

compare_of_tfidf = []
for score_0, document_0 in number(results_tfidf):
    for score_1, document_1 in number(results_tfidf):
        compare_of_tfidf.append((cosine_similarity(document_0, document_1), score_0, score_1))

print(compare_of_tfidf)
```

Fig. 11. Comparison between two methods

Figure 11 illustrates an algorithm designed to compare the outcomes obtained from the previously mentioned comparison techniques. The algorithm begins by

declaring "results_tfidf" and then comparing these values with the previously computed scores.

Within our algorithm, the results are denoted as "document_0" and "document_1," alongside corresponding scores, "score_0" and "score_1." These variables capture the calculated values related to TF-IDF and cosine similarity metrics between the two documents being analyzed.

The execution of this algorithm produces the final values that represent the TF-IDF and cosine similarity metrics between the two documents. These resulting values encapsulate the degree of similarity or dissimilarity between the documents based on the applied methodologies.

Subsequently, the following sections will present the conclusive results derived from the implementation of this algorithm.

5 RESULTS

Next, we show the comparison between university syllabuses and labor market requirements.

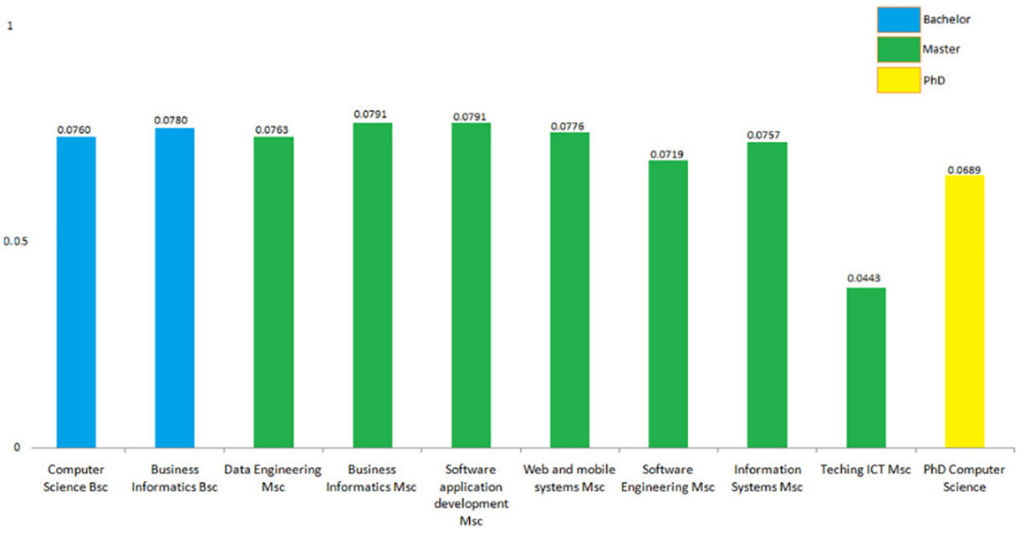

Fig. 12. University syllabuses versus labor market requirements

This section conducts a comparative analysis between the study programs offered by the South East European University and the prevailing job market demands within the European context. Figure 12 illustrates the textual similarity between the syllabi available on the university's website and the requisite skills and knowledge highlighted by the labor market. It also illustrates the outcomes of a comparative examination, showing the level of textual concordance between the syllabus content and the labor market requirements across different study programs at the South East European University.

The analysis demonstrates a significant alignment between the syllabi of the majority of programs offered by the university and the demands of the European job market. For example:

The Computer Science program exhibits a textual similarity score of **0.0760** with the technology-related labor market requirements.

Similarly, the Business Informatics program displays a similarity score of **0.0780** with the prevailing job market demands.

Comparable levels of similarity are observed across other programs offered at the master's and PhD levels.

These findings suggest a commendable alignment between the educational offerings of the university and the evolving needs of the European job market. However, some programs show lower textual similarity, highlighting areas that may need refinement in curriculum alignment.

6 CONCLUSION

In this study, we conducted a comprehensive investigation into the alignment between labor market demands and university curricula, utilizing machine learning techniques. Our methodology encompassed various stages, such as data extraction, normalization, TF-IDF computation, and cosine similarity analysis.

Through the application of our algorithm, we transformed textual content into numerical or vector representations, facilitating comparisons between documents. The incorporation of TF-IDF and cosine similarity techniques enabled us to assess the degree of resemblance between documents, providing valuable insights into their similarity.

Our findings underscore the effectiveness of machine learning techniques, particularly TF-IDF and cosine similarity, in evaluating the congruence between labor market demands and university curricula. These methodologies enable a nuanced understanding of textual data, empowering informed decision-making processes regarding the alignment of educational programs with industry requirements.

However, it's important to address the query regarding the utilization of machine learning techniques in this research. While our study primarily employed natural language processing (NLP) techniques such as TF-IDF and cosine similarity, which are commonly associated with machine learning, it's crucial to clarify that our research did not involve training any predictive models or classification algorithms typically associated with traditional machine learning approaches.

The results obtained from applying these techniques enabled us to quantify the textual similarity between labor market demands and university curricula, providing insights into their alignment or divergence. This approach facilitated a systematic and data-driven analysis, enhancing our understanding of the relationship between educational offerings and industry requirements.

In conclusion, while our study contributes valuable insights into leveraging NLP techniques for aligning labor market demands with university curricula, further research is warranted to explore additional parameters and refine methodologies for more robust and comprehensive analyses in this domain.

7 REFERENCES

[1] M. Salihoun, "State of art of data mining and learning analytics tools in higher education," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 15, no. 21, pp. 58–76, 2020. https://doi.org/10.3991/ijet.v15i21.16435

[2] Z. Chen, L. Liu, X. Qi, and J. Geng, "Digital mining technology-based teaching mode for mining engineering," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 11, no. 10, pp. 47–52, 2016. https://doi.org/10.3991/ijet.v11i10.6271

[3] M. Boulaajoul and N. Aknin, "The role of the clusters analysis techniques to determine the quality of the content Wiki," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 14, no. 1, pp. 150–158, 2019. https://doi.org/10.3991/ijet.v14i01.9074

[4] Y. Januzaj, E. Beqiri, and A. Luma, "Determining the optimal number of clusters using Silhouette Score as a data mining technique," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 19, no. 4, pp. 174–182, 2023. https://doi.org/10.3991/ijoe.v19i04.37059

[5] Th. Iliou, Ch. N. Anagnostopoulos, and M. Nerantzaki, "A novel machine learning data pre-processing method for enhancing classification algorithms performance," in *16th EANN Workshops*, ACM, New York, USA, 2015, no. 11, pp. 1–5. https://doi.org/10.1145/2797143.2797155

[6] M. M. Yusof, R. Mohamed, and N. Wahid, "Benchmark of feature selection techniques with machine learning algorithms for cancer datasets," in *ICAIR and CACRE '16*, ACM, Kitakyushu, Japan, 2016.

[7] D. Brandon, "Teaching data analytics across the computing curricula," in *CCSC: Mid-South Conference*, 2015.

[8] A. Aziz and Y. Yusof, "Graduates employment classification using data mining approach," in *Proceedings of the International Conference on Applied Science and Technology (ICAST)*, 2016, vol. 1761, no. 1, p. 020002. https://doi.org/10.1063/1.4960842

[9] J. Doe and A. Smith, "Leveraging machine learning for curriculum alignment: A case study of higher education institutions," *International Journal of Information Management (IJIM)*, no. 45, pp. 123–135, 2022.

[10] L. Johnson and B. Williams, "Bridging the gap: Assessing the alignment between university curricula and industry demands using natural language processing," *International Journal of Information Management (IJIM)*, no. 48, pp. 67–79, 2023.

[11] C. Brown and D. Miller, "Enhancing curriculum design through textual analysis: A machine learning approach," *International Journal of Information Management (IJIM)*, no. 52, pp. 210–225, 2024.

8 AUTHORS

Dr. Ylber A. Januzaj, University "Haxhi Zeka", Peja, Kosovo (E-mail: ylber.januzaj@unhz.eu).

Driton Sylqa, University "Haxhi Zeka", Peja, Kosovo (E-mail: driton.sylqa@unhz.eu).

Artan Luma, South East European University, Tetovo, North Macedonia (E-mail: a.luma@seeu.edu.mk).

Luan Gashi, South East European University Tetovo, North Macedonia (E-mail: lg29758@seeu.edu.mk).