IT Curricula Versus Labour Market Requirements in the Area of Cloud Computing in Austria

Peter Garscha[1(✉)] and Alexander Wöhrer[2]

[1] University of Applied Sciences Burgenland, Eisenstadt, Austria
peter.garscha@gmail.com
[2] University of Vienna, Vienna, Austria
alexander.woehrer@univie.ac.at

Abstract. Creating curricula for educational institutions that meet the demands of a fast-moving labour market is a complex process that can take up to several years. Especially in a field like information technology (IT), new technologies require ongoing adaptation of the corresponding curricula. A particular challenge is to put theoretical concepts, such as those taught by universities, in curricula in such a way that they correspond to the technologies that are currently required on the labour market. While this is a general problem, we elaborated it in the context of cloud computing by addressing the following questions: Is it adequately dealt with in the IT curricula of Austrian universities according to the requirements of the IT labour market? And further, how can curriculum alignments be (semi-)automated to help to better meet current IT job market needs? To answer them, the texts of job descriptions and IT study plans of Austrian universities are first analysed and later compared with similarity metrics. After a quantitative analysis, genetic algorithms are applied to improve the coverage of the curricula.

Keywords: Curriculum development · Cloud computing · Austria · Labour market

1 Introduction

A deeper look at the mass of different (IT-)curricula reveals that current curricula are not always up to date - at least in respect to the year of publication. Developing a curriculum for universities is not an easy process. [1] described the development of an IT-curriculum, that lasted ten months from the beginning of the development process till the proposal of the final version. A development process over several months or even longer is not only necessary for the development of a new curriculum, but also for any adjustments. An example is given in Table 1 in which the development process of an Austrian computer science master programme is shown. As it can be seen, the curriculum was originally

Table 1. Example of a computer science curriculum development in Austria

Year	Modification	Year	Modification
04/2007	Original	05/2012	Revision
10/2008	Correction	05/2019	Revision
06/2010	Revision	06/2019	Revision
09/2010	Correction		

developed in 2007 and regularly updated afterwards. But there was also a gap of seven years in between. Cloud computing is a so-called megatrend [2] and therefore not only relevant to computer science. This raises the question to what extent cloud computing is dealt within current curricula, e.g. computer science. The degree to which cloud computing is covered by a curriculum is not easy to answer. First the question must be answered, what is meant by cloud computing, whereupon NIST [3] provides a well accepted definition. In addition, cloud computing itself is not a new technology but based on concepts that have been used for many years, but are not explicitly recognised as cloud computing [4].

Based on the factors that, on the one hand, companies are currently heavily aligning their IT strategies with cloud computing, on the other hand, IT curricula in universities may not meet the needs of the labour market and may be slow to implement curriculum changes, the following scientific research questions arise:

– Is cloud computing adequately dealt within the IT curricula of Austrian universities according to the requirements of the IT labour market?
– How can proposals for improving IT curriculum be generated automatically in order to better meet current requirements of the IT job market?

The methodological approach is divided into two main phases. The first is to develop a method and corresponding software to create relations between the theoretical concepts and concepts of curricula on the one hand and the practice-oriented requirements in job descriptions on the other hand. For this we will use the online encyclopedia *Wikipedia* as an external source of knowledge. Within the second phase, appropriate suggestions for optimisation for the respective curriculum should be generated by the developed prototype. The mentioned research questions are to be evaluated by a statistical analysis.

The rest of this paper is structured as follows: Sect. 2 introduces background and related work while Sect. 3 elaborates on the high-level architecture and implementation details of our prototype. A kernel part of our paper can be found in Sect. 4 presenting the results together with a critical discussion. We finish with our conclusions and outlook for future work in Sect. 5.

2   Background and Related Work

Curricula in computer science have been already analysed in the past. By [5] two curricula of the *Massachusetts Institute of Technology* (Cambridge, USA)

and the *Open University* (Milton Keynes, United Kingdom) have been analysed using *LDA*[1] and *Isomap*[2]. While [5] developed a method for analysing curricula in detail, our work focuses on establishing relations between curricula and job descriptions.

In analysing ten different computer science related curricula, [6] found that some universities focus more on human factors, while others focus more on theoretical concepts. In our work this is not elaborated, although we determined that some curricula are more business-oriented while others are more technically.

Key technologies that should be included in a cloud computing curriculum have been described by [7]. It has been also differentiated between *fundamental technologies* and *enabling technologies*.

The restructuring of their computer science programme has been described by [8]. Further, a self-assessment strategy for evaluating the effectiveness of study programme changes has been introduced, including a job placement tracking and alumni surveys. In contrast, our work aims to automate such a process.

We investigated 84 curricula of Austrian computer science study programmes, and those study programmes that are also associated with computer science. To put into context, curriculum consists of a number of different courses and every course focuses on a special topic of the study programme. At some universities, the courses of a study programme are grouped in modules. Compared to module descriptions we observed that course descriptions are mostly much longer. Additionally we collected 2,014 IT-related job descriptions from an Austrian Internet job platform. In comparison to rather heterogeneous study programmes, job descriptions are significantly more homogeneous, at least in respect of the text length.

An overview on different methods and tools that we used in our work are shown in Fig. 1. We use the *Natural Language API*[3] and the *Custom Search JSON API*[4] of the Google Cloud Platform (GCP) as tools for text analysis. We will use both Google APIs for *indexing*, which means assigning each document a weighted vector of words and which is part of a typical text analysis process [9]. We use the Natural Language API for identifying salient words within curricula and job descriptions. Using Google's Custom Search JSON API allows to search the web programmatically. In our use case we use the Custom Search JSON API to retrieve Wikipedia pages related to the entities that we have found with the Natural Language API. Although Wikipedia is not an acceptable source for citation (as it is not primary nor secondary literature), it became popular in (computer) science as a new research approach [10], e.g. natural language processing. In our work, we will map the descriptions of courses and jobs to a weighted set of different Wikipedia articles.

---

[1] https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation, last accessed on 12th October 2019.

[2] https://en.wikipedia.org/wiki/Isomap, last accessed on 12th October 2019.

[3] https://cloud.google.com/natural-language/, last accessed on 12th October 2019.

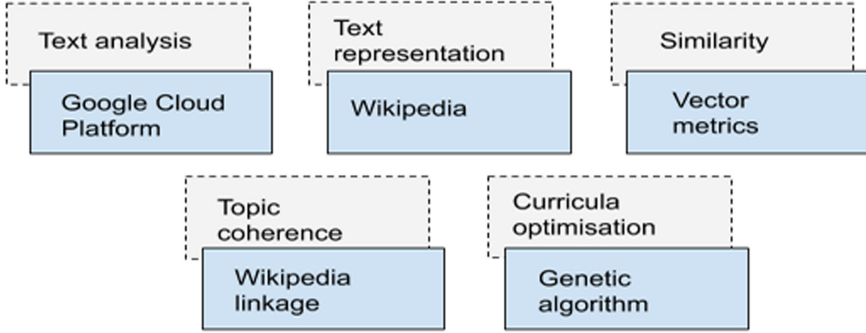[4] https://developers.google.com/custom-search/, last accessed on 12th October 2019.

Fig. 1. Overview on the used toolset

A metric function can be used to express the semantic similarity of two arbitrary texts. Given two vectors x and y in a two-dimensional space, that represent the affinity of two texts to two different topics A and B. That means, a text can be presented as a vector where each vector component represents a different topic and the value of a vector component indicates how much it is related to the topic. According to [11] we use the following cosine-coefficient, where x and y are vectors, n is the total number of topics represented by x and y, and k the index of the respective vector component.

$$\frac{\sum_{k=1}^{n} (weight_{x_k}) \cdot (weight_{y_k})}{\sqrt{\sum_{k=1}^{n}(weight_{x_k})^2} \cdot \sqrt{\sum_{k=1}^{n}(weight_{y_k})^2}}$$

We also need to determine how much a Wikipedia article A is related to a given topic. Assumed that we have already identified that topic by another Wikipedia article B, we can calculate the relatedness to any Wikipedia article A by the link distance of this two articles. More precisely, as probably every Wikipedia article has incoming and outgoing links from and to other articles, we define the topic coherence C between two articles a_1 and a_2 by

$$C(a_1, a_2) := \min(\{ld(a_1, a_2)\})$$

where $ld(a_1, a_2)$ represents an oriented linkage from article a_1 to article a_2, defined by the number of edges. Note, that $C(a_1, a_2)$ is not necessarily equal to $C(a_2, a_1)$.

We will further use genetic algorithms to optimise current IT-curricula to better meet the requirements of the job market and to increase cloud computing aspects in the curricula. Genetic algorithms are part of algorithms for solving multi-objective optimisation problems or at least to find suitable solutions. [12] provide an overview on different methods for multi-objective optimisation as well as a mathematical introduction and also a brief history on genetic algorithms. Although genetic algorithms are quite powerful, it is not guaranteed that the global optimum will be found [13].

3   Implementation

The basic concept of our work is the mapping from a text, that can be a job description or a course of a curriculum, to a weighted vector of Wikipedia articles that is being used as a meaningful representation. An example in Table 2 shows such a mapping of a course description about virtualisation technologies of a master programme on Cloud Computing Engineering at the University of Applied Sciences Burgenland in Austria. Although the text has been mapped to a total of 183 articles, only the most valuable with the highest weight are shown.

Table 2. Mapping example of a course description

Course description	Wikipedia article	Weight
Virtualization as emulation of various resources (hardware, storage, network, application). Levels, architectures and protocols of virtualized systems. Overview of the most important products for hardware, operating system, application, storage, desktop virtualization, their use, advantages/disadvantages, limitations. Use, security assessment, testing and validation of various virtualization technologies and products, both open source and commercial	List of Virtualizationproducts	0.1772
	x86-Virtualization	0.1410
	Virtualization (Informatics)	0.1410
	Applicationvirtualization	0.1377
	Virtual Desktop Infrastructure	0.1377
	Open-Source-Hardware	0.1289
	Virtualization	0.1016
	Storagevirtualization	0.1016
	Containervirtualization	0.1016
	Xen	0.1016
	Docker (Software)	0.1016
	Hercules (Emulator)	0.0722

As one of our goals is to establish a relationship between curriculum and a job description, we can map both to two different vectors and compare them with a similarity measure. The first step is the mapping of a text to a weighted vector of words using the Natural Language API of the Google Cloud Platform.

As job descriptions and curricula use different wordings for the same concepts, we will further map these words to Wikipedia articles using the Google Custom Search API. Finally, after we derived two weighted vectors of Wikipedia articles for two texts t_1 and t_2, the similarity $S(t_1, t_2)$ of the texts can be calculated using a metric. The result is a real number greater than or equal to zero.

Curriculum can be split up to a number of courses that are either mandatory or optional. But both, a course and a job are represented in the same manner by a text. That means we are now able to compare different job descriptions with each other, to compare different courses with each other and to compare a course with a job. Therefore, we define: let C be a set of curricula, $c \in C$, J a set of job-descriptions, $j \in J$, $M(c)$ the set of courses of the curricula c, $m(c)$ a course of curricula c and $|X|$ the number of elements in an arbitrary set X. The

similarity between a curriculum c and a job-description j can be expressed by the mean similarity of all courses of the curriculum and the job-description:

$$S(c,j) \; := \; \frac{\sum_{i=1}^{|M(c)|} S(m_i(c), j)}{|M(c)|}$$

Quite similar definitions can be used for comparing two courses or two curricula. According to [14], who investigated different similarity measures, there is no clear answer for which is the best method of combining similarity values.

To calculate the topic coherence of a job or a curriculum to any given topic, we define the coherence $C(a_1, a_2)$ of a Wikipedia article a_1 to a topic represented by another Wikipedia article a_2 by the shortest path of links from article a_1 to article a_2. For a job j, represented as a set of weighted Wikipedia articles $A(j) \; := \; \{w_1 \cdot a_1(j), ..., w_n \cdot a_n(j)\}$, the coherence to a topic a can be defined as:

$$C(j,a) \; := \; \frac{\sum_{i=1}^{|A(j)|} (1 - w_i) \cdot (C(a_i(j), a) + 1)}{|A(j)|}$$

Consequently, let M be a set of modules or courses of a curriculum c, and $m \in M$ represented as a set of weighted Wikipedia articles $A(m) \; := \; \{w_1 \cdot a_1(m), ..., w_n \cdot a_n(m)\}$, the coherence of an entire curriculum c to a topic a can be defined as:

$$C(c,a) \; := \; \frac{\sum_{k=1}^{|M|} \sum_{i=1}^{|A(m_k)|} (1 - w_i) \cdot (C(a_i(m_k), a) + 1)}{\sum_{k=1}^{|M|} |A(m)|}$$

4   Results and Discussion

Using the Google Natural Language API we retrieved 75,017 salient words which correspond to an average of 29.33 words per course. Considering all curricula as a whole, these 75,017 words decreased to 16,228 unique words. On the other side, from 2,048 job descriptions we got a total of 183,047 salient words on an average of 90.89 words per job description. Considering all job descriptions as a whole, they reduce to 29,179 unique words. In total, of all jobs and courses we got 41,615 unique salient words. 25,387 (61.00%) of them have been found only in job descriptions, 12,436 (29.88%) have been found only in curricula and just 3,792 (9.12%) have been found both in jobs and curricula.

In order to prove that only parts of cloud computing can be assigned to topics and concepts in current IT curricula, we have to define these subtopics. Unfortunately, our proposed model to map curricula to sets of Wikipedia articles and to calculate their distances to a specified Wikipedia article representing a cloud subtopic is not suitable, because some cloud-related topics like e.g. *Infrastructure as a Service (IaaS)* and none of the common known deployment models of cloud services (as described by [15]) have corresponding Wikipedia articles in the German version of Wikipedia at this time. Hence, we selected all Wikipedia articles that link to the Wikipedia article of cloud computing and vice versa all articles that have a link from cloud computing. We found 464 articles and

filtered the list, so that it didn't contain names of companies or related products. For the remaining 58 subjects, of which only a few can be clearly assigned to the topic of cloud computing, we calculated the coherence to all curricula. Ordering the results revealed that the articles that rank best are very common topics like Internet, Software or Computer. That is not very surprisingly as every curriculum is related to IT topics.

According to [16] the IT job market can be divided in five categories: programming languages, web development, database, operating systems and networking. We added cloud computing as a sixth category, as it is of special interest for us. We set a corresponding Wikipedia article for each category and calculated the link distances for all articles, that have been found by the Google Search API for the collected jobs and curricula, to each of these six articles. We calculated the link distance to each job category and determined the top 100 jobs that are closest to each category based on their link distance. Finally, we evaluated the similarity of all courses to the jobs of each category.

The results shown in Fig. 2 indicate that the jobs that are related to cloud computing are not the least targeted jobs by current IT-related curricula at universities in Austria. A reason can be that cloud computing comprises a wide variety of common technologies and subjects (Internet, IoT, security, virtualisation, etc.) and that it cannot be defined clearly as other categories. While jobs that are related closely to the category *Webapplication* are targeted best by the curricula and jobs in the categories *Programming Languages* and *Operating System* are poorly targeted. It seems to be meaningful that jobs close to web applications are targeted more than the other categories because there is an ongoing common trend in transforming existing software systems in enterprises to web-based and service-oriented solutions.

We found that the curricula that matched the job market best are mostly from computer science whereas the curricula that match the job market poorly are subject specific like medical informatics or media informatics. It is much harder to evaluate how far a single course matches the job market as the results are dependent on the text length of the considered course description. The longer
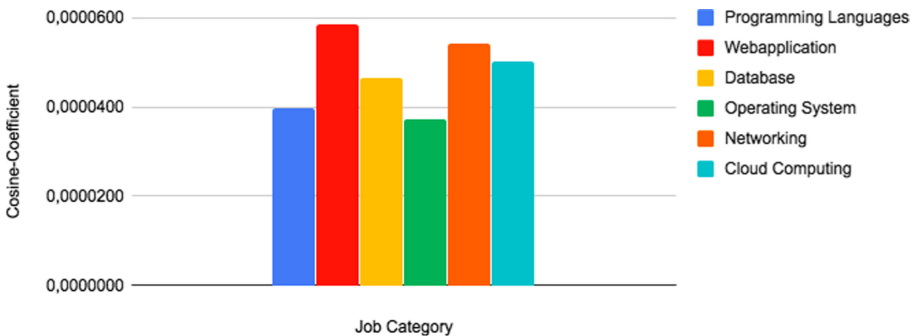

Fig. 2. Average similarity between IT curricula and the closest jobs of each category

a text is, the more salient words can be found that are not related to IT and only cause noise.

As it is our vision to upload curricula under development to a website to get instant recommendations for improvements towards coverage of the job market and the relatedness to certain topics, we tried to optimise curricula with the use of genetic algorithms. A genetic algorithm typically is based on three genetic operators: a selection-, a recombination- and a mutation-operator. Thus, a mutation can simply be defined as exchanging a random course of a curriculum by a random course of any other curriculum. A crossover-operator can be defined as follows: Select all courses that are contained in both curricula and select the remaining number of missing courses from one of both curricula randomly. For selecting the fittest individuals, we calculated how well every curriculum matches the set of jobs and choose the best ones.
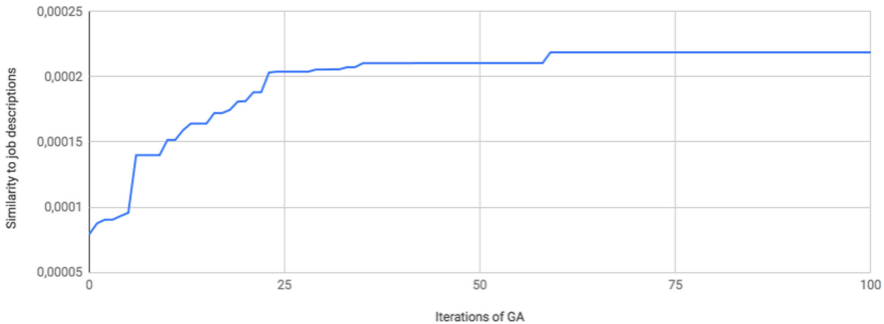

Fig. 3. Average optimisation of curricula by a genetic algorithm

We found that the genetic algorithm improves the similarity to the jobs and the linkage distance to the topic cloud computing. On average, the similarity of 56 selected curricula (that consist of at least 20 courses) to the set of job descriptions could be increased by 148% as shown in Fig. 3, whilst the distance to the topic cloud computing could be only increased by 1.01%. At the same time, the text lengths of the curricula drop by 5.86%. In order for the changes not to impact the curricula too much, we set a constraint of a maximum of five courses that may be exchanged. The algorithm achieves most of the optimisation after 25 iterations, hence it converges quite fast.

5   Conclusions and Future Work

Our main contribution to the research area of computer science education is the development of a new approach for comparing curricula with job descriptions. We wanted to show to which extent the IT job market is targeted by current IT curricula at universities in Austria and applied our developed approach. Additionally, the topic cloud computing was of special interest to us, because we

supposed that it is not covered sufficiently by the curricula, accordingly to the demand on the IT job market.

Critically summarised, we only focused on IT related curricula and jobs in Austria and only investigated the IT labour market. Nevertheless, our approach can be also applied to other domains and corresponding results would be very interesting. There are possibly many other approaches instead of using similarity metrics for comparing curricula and job descriptions. Although the use of Wikipedia is very convincing for us and is used by many scientists by now, we have to rely on external data and the trustworthiness has to be kept in mind. We decided to use cloud services and Google as an external provider to identify salient words in curricula and job descriptions. Here, too, we have to rely on external data and we actually do not know how Google identifies salient words in texts.

For future research as well as for application we describe two different issues. The first is the missing framework of curricula, the second is the automated improvement of curricula. Some curricula contains only course descriptions whilst others contain only module descriptions, that put related courses together. Few curricula also contain both, descriptions of courses and modules. There are certainly more differences that can be found in comparing curricula. Due to these variations, it is very hard to compare different courses or modules. We recommend the development and implementation of a common framework for curricula descriptions that are being designed at universities. In the best case, future developers of curricula also have an online tool that they can use to analyse their curricula whilst they are still work in progress.

Remarks

References

1. Herbert, N., de Salas, K., Lewis, I., Dermoudy, J., Ellis, L.: ICT curriculum and course structure: the great balancing act. In: Proceedings of the Sixteenth Australasian Computing Education Conference, pp. 21–30. ACM, New York (2014)
2. Peciak, R.: Megatrends and their implications in the globalised world. Horiz. Polit. **7**(21), 167–184 (2016)
3. Mell, P., Grance, T.: The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology. NIST Special Publication 800-145 (2011)
4. Carlin, S., Curran, K.: Cloud computing technologies. Int. J. Cloud Comput. Serv. Sci. **1**(2), 59 (2012)
5. Sekiya, T., Matsuda, Y., Yamaguchi, K.: Analysis of computer science related curriculum on LDA and Isomap. In: Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education, pp. 48–52. ACM, New York (2010)

6. Sekiya, T., Matsuda, Y., Yamaguchi, K.: Curriculum analysis of CS departments based on CS2013 by simplified, supervised LDA. In: Proceedings of the Fifth International Conference on Learning Analytics and Knowledge, pp. 330–339. ACM, New York (2015)

7. Border, C.B.: Cloud computing in the curriculum: fundamental and enabling technologies. In: Proceeding of the 44th ACM Technical Symposium on Computer Science Education, pp. 147–152. ACM, New York (2013)

8. Nordstrom, S., Randrianasolo, A., Borera, E., Mhlanga, F.: Winds of change: toward systemic improvement of a computer science program. In: Proceedings of the 14th Annual ACM SIGITE Conference on Information Technology Education, pp. 201–206. ACM, New York (2013)

9. Korde, V., Mahender, C.N.: Text classification and classifiers: a survey. Int. J. Artif. Intell. Appl. **3**(2), 85 (2012)

10. Bar-Ilan, J., Aharony, N.: Twelve years of Wikipedia research. In: Proceedings of the 2014 ACM Conference on Web Science, pp. 243–244. ACM, New York (2014)

11. Haenelt, K.: Ähnlichkeitsmaße für Vektoren (2012). http://kontext.fraunhofer.de/ haenelt/kurs/folien/Haenelt_VektorAehnlichkeit.pdf. Accessed 2 Aug 2019

12. Marler, R.T., Arora, J.S.: Survey of multi-objective optimization methods for engineering. Struct. Multidisc. Optim. **26**(6), 369–395 (2004). https://doi.org/10.1007/ s00158-003-0368-6

13. Thengade, A., Dondal, R.: Genetic algorithm-survey paper. In: MPGI National Multi Conference, pp. 7–8. Citeseer (2012)

14. Newman, D., Lau, J.H., Grieser, K., Baldwin, T.: Automatic evaluation of topic coherence. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 100–108 (2010)

15. Yang, H., Tate, M.: A descriptive literature review and classification of cloud computing research. Commun. Assoc. Inf. Syst. **31**, 2 (2012)

16. Koong, K.S., Liu, L.C., Liu, X.: A study of the demand for information technology professionals in selected internet job portals. J. Inf. Syst. Educ. **13**(1), 21–28 (2002)