

SUSTAINABLE DEVELOPMENT IN EDUCATION – AUTOMATING CURRICULUM ASSESSMENT

Claudiu VINȚE ^{*}, Ion SMEUREANU,
Marian DÂRDALĂ, Adriana REVEIU

*Department of Economic Informatics and Cybernetics, Bucharest University of Economic Studies,
Bucharest, Romania*

Received 21 October 2020; accepted 21 March 2021; first published online 05 July 2021

Abstract. The perpetual need for developing a sustainable economic environment places the education policies at the foundation of social adaptability. Creating and maintaining curriculum content that meets the demands of a continuously changing society, and the challenges that such a rapid evolution put on the labour market, is one of the top priorities for any education system and institution involved in education at any level. This paper proposes a cognitive computing solution for assessing, in a programmatic manner, large corpora of curriculum content created by teachers from lower secondary education environment for Informatics instruction in Romanian schools. The result of this initiative at the national level is corpora of curricular content that must be evaluated to verify the degree to which the material meets the requirements of the national curriculum. We addressed this crucial yet tedious process by designing and implementing a solution for automating curriculum assessment through cognitive computing. The paper outlines a sustainable framework to evaluate curriculum content in an automated fashion, and for providing critical feedback timely to both content creators, and to policy makers responsible for creating economically viable and future adaptable education strategies.

Keywords: education for sustainable development, curriculum content assessment automation, cognitive computing, cluster analysis, GIS technologies.

JEL Classification: I25, C45, C38.

Introduction

Having to face an increasing demand for IT professionals in the near future, Romania is one the first countries in the European Union to introduce Informatics instruction at lower secondary school level, beginning with 2017. According to one popular estimate, over 65% of children entering primary school today will eventually end up working in job types that have not been invented yet (World Economic Forum, 2016). In order to prepare and cope

*Corresponding author. E-mail: claudiu.vinte@ie.ase.ro

with the challenges that we may face 15–20 years from now, and which are not entirely transparent today (Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education, 2013), Romania introduced in 2017 the Informatics as mandatory discipline in the national curricula for lower secondary school. It has yet to do the same for the higher secondary school, where Informatics instruction is currently not a mandatory discipline (Guerra et al., 2012). The reluctance of introducing the Informatics instruction in the core of the national curricula for the higher secondary education is motivated by the lack of necessary staff required across the country for supporting such an initiative. Otherwise, the strategic need for introducing Informatics instruction as a mandatory discipline in higher secondary education level is generally acknowledged by any governmental body (Micheuz, 2006).

Moreover, our society is becoming more and more digitized, whilst an increasing number of economic activities have been moved to a digital format or at least are embracing ICT technologies to a large extent. A self-evident consequence is the emergence of a growing demand for skilled labour, with improved IT competences, and this request will be stable or even growing in the near future. Therefore, the education system has a paramount role, more than ever before, being pushed to include adequate Informatics courses in curriculum, at all education levels, to prepare digitally skilled young people.

Arguments for the inclusion of Informatics in the curriculum rely not only on the economic rationale that drives countries to create computer experts for supporting a competitive edge in a world pushed by technology, but also on the need for professionals with digital competences to empower innovation and development across all sectors.

There is a deep consensus that supporting teacher professional development in Informatics is critically for launching new Informatics courses and this is currently a major challenge in many countries but, to our current knowledge, there is no research paper published so far that has proposed and validated useful solutions for assessing teaching portfolio in relation with national curriculum requirements. The solution proposed in this paper aims to support teachers in building the best supporting scholar portfolio for their classes.

Although ongoing studies on Informatics education have been published, the research to date has tended to focus on high education (Serban et al., 2019), college or high school rather than teaching the subject effectively at all levels in schools. With a focus on primary and secondary educational level, this paper seeks to provide a methodology and empirical results helping us to address this research gap.

Curriculum assessment automation – context requirements

As many countries are in process of curriculum innovation or launching new Informatics courses, at different education levels, a growing body of research copes with curriculum content assessment and investigates the impact of courses trained to employers on the success of their business activities. There are both country-level and regional-wide initiatives carried by governmental institutions and professions organisations encouraging the wide-spreading Informatics education, the developing of teachers' subject knowledge, alongside with improving teaching resources in order to boost rapid curriculum advance and change. Well-known regional projects are: "Informatics for all", jointly devised by ACM Europe and

Informatics Europe since 2017, to promote Informatics education at all levels in European countries, and “CS for All” in US (Caspersen et al., 2018). Countrywide initiatives emerged in England, where a new Computing subject was introduced for all students since 2013 (British Department for Education, 2013), and also in Australia where the compulsory subject Digital Technologies was launched in 2015 (Webb et al., 2017). Today, Informatics is also a compulsory course for all K-12 students in New Zealand (Barr & Stephenson, 2011), Poland (Sysło & Kwiatkowska, 2015), and France (Dowek, 2016).

As Informatics is a mix of science and technology, three forms of complexity have to be considered when designing Informatics course curriculum and curricular content: equilibrium between technological and scientific activities, the balance among concepts, and equilibrium between the subject core and its interfaces (Dowek, 2016). To date, different methods for curriculum designing were approached by scholars and have reported positive outcomes. Design thinking was used to develop meaningful support materials for teaching Computer Science and IT in late primary to secondary German pupils (Diethelm & Schaumburg, 2016). The analysis of conceptual content of curriculum specifications was used by Barendsen and Steenvoorden (2016) to assess Informatics Dutch curriculum for grades 10–12.

Significant challenges have been acknowledged in defining an appropriate curriculum content and class portfolios for Informatics courses taught at K12 education level. For example, defining a global view of teaching Informatics and of its arrangement over time was noticed by Dowek (2016) as a drawback in France, wherein Scientific Committee of the Société Informatique de France conducted a curriculum appraisal. Insufficient teachers with the appropriate subject matter knowledge, ensuring assessment supports, empowers exciting and challenging learning in Computer Science in a cross-disciplinary approach were pointed up by Webb et al. (2017).

Moreover, very similar challenges were identified when was investigated the impact of courses trained to the staff members, also in industry. Science and technology delivered in a cross-disciplinary manner and the sequence of relevant training courses were considered essentially for business activities success in a research reported by Lee (2013) and conducted in Seoul, Korea.

Overall, these studies highlight the need for future research to produce adequate curricula and proper supporting materials for different school levels: primary, lower secondary and upper secondary (Fethi & Lowther, 2010).

Even if it is an emerging agreement on the importance of starting to learn Computer Science courses early in primary schools, and on the fact that teachers' effort to prepare and select proper educational materials is tremendously and can impact on their involvement in this activity (Ertmer & Ottenbreit-Leftwich, 2010). At the best of our knowledge there are no studies assessing the corpora of countrywide scholar curriculum content, in a programmatic way, in the attempt to reveal if and to which extent they fit with the certified curriculum requirements at K12 educational level (Pereira & Tikhonenko, 2017).

In Romania, at beginning of 2017, building on this nation-wide initiative, and together with group of enthusiastic professors from academia, The Ministry of Education, The Institute of Education Sciences, and lower secondary school teachers, we created www.informat-icainscoli.ro (Informatica în școli, 2020). as a cloud-based curriculum content wiki reference

platform, in order to share and disseminate class support material among the teachers of the newly introduced discipline of Informatics.

The outcome of this initiative is corpora of curricular content created by a community over 2300 teachers at the national level. Mainly, the aforementioned wiki platform hosts over 4000 portfolios created by lower secondary school teachers, from across the country, upon certain content areas within the Informatics & ICT national curriculum. The portfolios reflect the teaching experience of each teacher, and they are to be harmonized in order to ensure a standardized degree of proficiency at the national level (Diana et al., 2017).

Initially, the Romanian Minister of Education estimated the need for training approximately 3675 teachers. There were designated trainers and constituted 147 groups across the country, with training sessions organized and conducted in centres assigned based on the traditional regions the counties belong to: Banat (South-West), Maramures (North-West), Muntenia (South), Bucharest area, Bucovina (North-East), Moldova (East), Dobrogea (South-East), and Transilvania (Central region).

Nationwide, there were involved in training sessions 2304 lower secondary school teachers, and 1917 of them have created portfolios of class lessons, 4047 portfolios overall, for one or multiple content domains that are covered by the 5th grade Informatics & ICT curriculum, namely: Safety & Ergonomics, Systems & Communication, System Architecture, Peripheral Devices, Operating Systems, Internet, Graphic Editors and Algorithms. Figure 1 depicts the percentages of portfolios created for each of those 8 content domains required by the national curriculum.

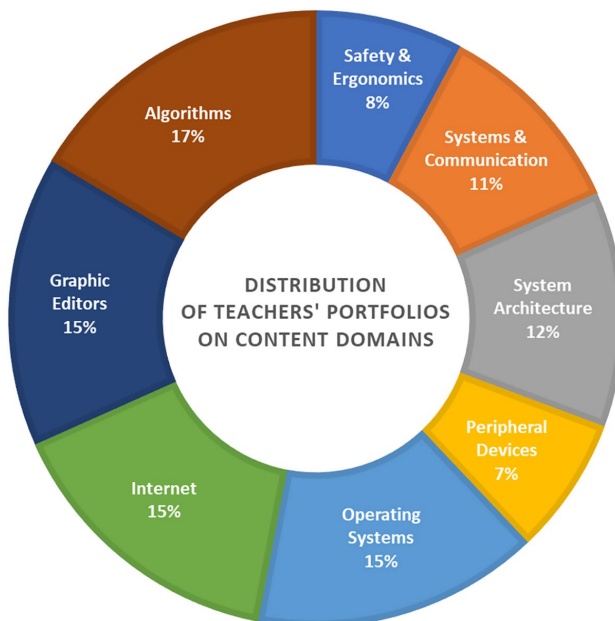


Figure 1. Distribution of teachers' portfolios based on the content domains they covered

Research objective and scope

The objective of our research is to identify and develop a solution to automate the assessment process of thousands of portfolios created by lower secondary school teachers for Informatics instruction, a newly introduced discipline in the compulsory education core of the Romanian curricula for the 5th grade students. The rationales for embarking on this endeavor come chiefly from the following two directions:

- a) to provide automated feedback to thousands of teachers in a consistent and time effective manner;
- b) to offer the means for coordinating agents in education, at the national level, for assessing the quality of teaching support materials used for Informatics instruction in the lower secondary schools.

The scope of the paper is to cover the proposed solution for the content domain of *Algorithms* intended for the 5th grade students, as part of the mandatory curriculum of Informatics instruction in Romanian lower secondary education level. A similar approach has been used for all the content areas of the national curriculum of Informatics instruction intended for the 5th grade students.

The paper is structured in four sections. The first part presents the arguments for approaching the answer to the research question, regarding the possibility of developing an automated solution for curriculum assessment, through the use of cognitive computing. The second section outlines the proposed methodology for developing a cognitive computing model for supporting curriculum assessment automation by making use of IBM Watson cloud-based services. In the third part is presented the cognitive model implementation and the software application developed for achieving the automated curriculum assessment of teachers' portfolios. There is also described the output that the software solution offers as feedback to teachers and we also conduct a cluster analysis on the automated portfolio evaluation results. The final section draws the conclusions upon the outcome of the conducted research, points out the potential limitations of the scientific approach, and offers a perspective regarding further research in the field of cognitive computing for area specific content evaluation.

1. The approach through cognitive computing

The context briefly presented in the introductory paragraphs raised the need to generate a coherent feedback mechanism to teachers, regarding the content of their portfolios, and to evaluate in a unitary manner the degree of compliance with the content requirements of the national curriculum for Informatics instruction. Thereby, if there were to be provided an automated solution for evaluating curriculum content, it was essential to identify a set of criteria to support the assessment process. Among the criteria for curriculum content assessment, the following have been selected:

- *validity* – the curriculum content is valid if it conveys the outcome that it is intended to be conveyed;
- *significance* – if the content is selected and organized for developing of learning activities that will help students in solving practical problems;

- *interest* – refers to a learning-centred curriculum; the students' interest should be considered when selecting the content, since students learn best if the subject matter is meaningful to them;
- *learnability* – the content should be what the students can learn, and it should be within their experience;
- *self-sufficiency* – if it helps learners to attain maximum self-sufficiency in the most economical manner or content selection; this criterion is achieved when the students or learners are given the chance to experiment, observe and carryout field study;
- *utility* – concerns the usefulness of the content in solving real world problems now and in the future.

Having to deal with such large corpora of portfolios, we have to take into account the fact that these portfolios only concern the curriculum intended for the 5th year students. In addition to that, in the coming years there is going to be conceived curriculum content specific for all four years of the lower secondary education level (gymnasium). Therefore, the overall collection of portfolios could easily exceed 20000 portfolios over the next 4 years. In order to provide a continuous and consistent curriculum content assessment process, an automated solution is the only viable option that we may consider.

Our aim is to find a way to automatically assess if teachers' portfolios for the grade 5 students contain the salient concepts required by the national curriculum. More specifically, we intend to find a way to run cognitive queries on the teachers' portfolios in order to check if they meet the official curricular requirements (McCalla, 2000). The output of the cognitive queries should be text passages that contain the ideas that are being searched for, if the corresponding concepts were retrieved in the text portfolios (Wenger, 1987). The data that we employ in our research comes from lower secondary school teachers of Informatics, who were recently trained to teach coding to children. The portfolios are written in Romanian language which is currently unsupported by IBM Watson (Wilson, 2018). This generates additional complexity to the model, since we are to identify entities and run cognitive queries in a language in which Watson artificial neural networks have not been previously trained for.

IBM Watson is a cloud-based AI solution consisting in a collection of services such as: Watson Assistant, for supporting chat-bots development; Watson Knowledge Studio, Natural Language Understanding, and Natural Language Classifier, for cognitive computing modelling, along with Watson Discovery, designed for querying massive amount of presumably unstructured data, on which a previously created cognitive model has been applied (International Business Machines [IBM], 2020).

A cognitive computing cloud-based service is a platform that provides implementations for artificial neural networks (ANNs) which, from user perspective, appear as systems that may learn to perform concrete tasks by being exposed to examples, and without being programmed with any task-intrinsic rules. For instance, an ANN can be trained for medical diagnosis and distinguish a malign melanoma from a benign one, based on images positively illustrating the malign melanomas. In addition to positive examples, there can be introduced negative examples, or example of benign melanomas, which are consequently labelled as non-malign ones. The training of an ANN for such a purpose is conducted without any *a priori* knowledge about malign melanomas, that they have a certain colour, or a certain type

of contour, vascularization etc. Instead, an ANN automatically generates identifying characteristics from the examples that it processes (Thurai, 2017).

To reach our research goals, the curriculum corpora designed in the natural language has to be automatically processed in the endeavour to identify meaning within the text, acting as people do, and using dictionaries, identifying repeated patterns of co-occurring words, and other contextual evidences (Hurwitz et al., 2015).

Cognitive computing offers appropriate technologies for processing and appraisal a large corpora of curriculum content because it is developed around three fundamental principles, namely: learning, modelling and generating hypotheses. A cognitive system assumes that there are multiple correct answers and based on a probabilistic approach, the most appropriate answer is identified or statistically significant patterns along with the linkages in data elements, so that the meaning of text expressed as unstructured data, can be interpreted in the context (Modha et al., 2015). Based on the available data, the system learns to train, to test and to score a hypothesis. Since a cognitive system typically works on knowledge base, or corpus, created by investigating various structured and unstructured data sources, this technology is suitable for gauging knowledge for our corpus of teaching materials. The cognitive computing methods have been meaningfully applied in socio-economic research to assess bank branch front-office employees' activity (Paes de Faria et al., 2020), when a fuzzy cognitive mapping method was integrated with system dynamics, or to identify the determinants of e-commerce in small and medium-sized enterprises (Barroso et al., 2019).

Automation of natural language processing (NLP) has been applied with relevant and valuable results in various fields including education, to establish more personalized and customized teaching and learning environments helpful for enhancing students' performance (Lytras et al., 2019), and to explore online discussion forums of students in business (Hernández-Lara et al., 2019). NLP has been also applied in business activities to uncover semantics from business vocabulary and business rules (Danenas et al., 2020), or for business performance prediction in location-based social commerce (Chang & Li, 2019). A linguistic multi-criteria decision model based on the fuzzy 2-tuple linguistic used to manage data depicting corporate criteria, expressed by distinctive business staff for scheduling the launch prioritization of new products and services was proposed and empirically tested by Cid-López et al. (2018).

For our research purpose NLP techniques are indispensable to capture the meaning of unstructured text from the teaching materials. However, an awkward challenge of natural language processing that still remains is to automatic capture and extract the deep meaning from the complex natural language.

Research papers provide evidence of valuable cognitive computing solutions proposed in health, and marketing, business and education to uncover semantics from business and academic documents and to investigate firms competing in a market (Iandoli et al., 2010).

2. The proposed research methodology

In order to be efficient and reliable when it comes to identify in new texts the entities that quintessentially characterize a specific domain, cognitive models have to be built purposely for a very narrow content area. For example, the Informatics instruction national curriculum

for the 1st year of lower secondary school (5th year in the overall education system), consist of the following 8 content areas or content domains: Safety & Ergonomics, Systems & Communication, System Architecture, Peripheral Devices, Operating Systems, Internet, Graphic Editors and Algorithms.

Even with content areas closely related to each other, such as the ones enumerated above, the current development in the Artificial Intelligence (AI) field makes the process of covering all of them in a single model at least very difficult if not too complex. The methodology that we propose for tackling such a problem is to conceive and implement specific cognitive model for each content area. Therefore, the methodological approach for each content area will have to follow the same steps, or model development phases:

- A. *define the ideas that are intended to be checked for coverage in teachers' portfolios;*
- B. *create the cognitive model;*
- C. *apply documents for training the ANN, based on the defined model; the training documents are created based on the national curriculum requirements for the specific content areas, and have to be representative scientific and didactic texts for the content area the particular model is intended for;*
- D. *deploy the model to cognitive discovery services;*
- E. *employ cognitive queries in a programmatic manner for retrieving in a new, "first time to see" document, passages that cover the concepts defined in the cognitive model;*
- F. *based on the results returned by the cognitive queries, generate feedback intended for the teacher who created the portfolio document in the first place.*

The experimentally determined rule of thumb is that an entity type has to be mentioned in at least 10 different training contexts in order to obtain a satisfactory degree of confidence in the ANN ability to recognize it in a new text. The ideas that are to be verified for being treated in the teachers' portfolios concerning a certain content area derive from the actual questions we want to answer to by evaluating the portfolios. For instance, if there has to be checked for content area concerning Algorithms that a definition for the concept of algorithm has been provided in the portfolio, then we have to answer the question: "has the concept of algorithm been defined?"

It has to be emphasised that the steps from A to C are part of an iterative process designed to improve the ANN ability to learn about the concepts that are established through the model. Furthermore, the whole process of refining the cognitive model is an incremental one, and the steps from A to E could be repeated for multiple times, depending on the output obtained by applying the cognitive queries to a new document (see Figure 2).

The process of training an artificial neural network is not a deterministic one. It relies on confidence levels that may, or may not be accepted by the research team that conceives the cognitive model. This approach implies that we begin the model with 2–3 concepts and the relations that might be useful to establish among them. Then, there are provided documents for training and based on them the ANN learns the concrete context in which the concepts actually operate through texts created by human experts in the field, and learn how these concepts may be connected to each other, based on previously defined relation types between them.

There no precise suggestion regarding the number of training examples that has to be provided in order for an ANN to learn how to recognize the intended concepts (entity types) in a new text.

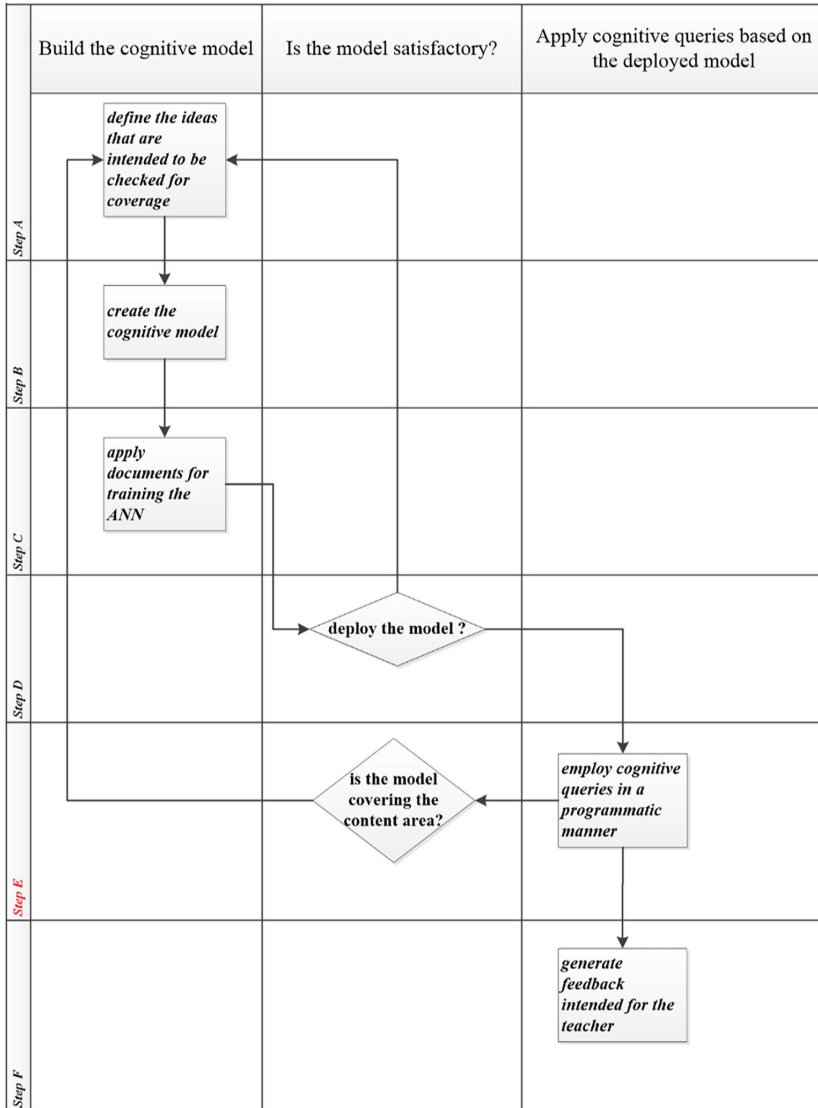


Figure 2. The proposed methodological steps and phases for developing a cognitive model

2.1. Define the ideas that are to be covered for each content area

The whole cognitive computing approach to provide automated curriculum assessment is saliently based on two questions and the answers that we are able to provide to these questions:

- a) Which would be the ideas that we are to look for, and that have to be treated in teacher portfolios, in order to consider that the given text covers properly a certain content area of the national curriculum?
- b) Which would be the concepts that best synthesize the ideas previously established for being checked?

These two questions are to be asked and answered recurrently for each content area we want to develop a cognitive model. Thereby, they constitute the detailed research questions that we addressed throughout our research.

In this paper we focus on the content area concerning *Algorithms* as it required by the national curriculum for the 5th grade. The ideas that are checked for being covered regarding algorithms are the following:

- portfolio contains real life examples of algorithms at work, like crossing the street, brushing the teeth, preparing for going to school, cooking a dish etc.; examples presented in the natural language;
- it is introduced the concept of operation sequences executed with a predefined goal;
- it is provided a formal definition of algorithm;
- it is explained the concept of logical expression;
- there are introduced the programming structures: linear, conditional (decisional), repetitive;
- identify the data types employed by computer algorithms;
- illustrate the step-by-step processing of data having given a predefined purpose;
- there are examples of grouping data in collections or set;
- introductory discussions upon formalizing algorithms, using the natural language;
- portfolio contains examples of simple algorithm implementations in the Scratch programming language.

Once the ideas that have to be checked for coverage in the teachers' portfolios have been established for a specific content area, then the first concrete steps in terms of interacting with Watson Knowledge Studio (WKS) is to identify the relevant entity types that best describe the established ideas to look after (IBM Cloud Docs, WKS, 2018b).

2.2. Identify the relevant entity types

Within WKS environment entity types represent concepts that agglutinate a set of concrete terms, or keywords from the natural language. Comprehensive educational solutions have been developed around IBM Watson AI technologies to improve learning outcomes, to deliver personalized content and recommendations to students, and for assisting in career guidance (Neti, 2018).

Our research concerns portfolios created by lower secondary school teachers from Romania upon Informatics instruction for the 5th grade students. The portfolios are written in Romanian language, with inconsistent usage of the language specific diacritics. This creates the basis for multiple spelling forms for a given word encountering in the teachers' documents. We shall discuss this aspect in more details when presenting the need for creating dictionaries. It is important to point out that IBM Watson does not offer support for the Romanian language as of July 2020. It means that Watson ANN has not been exposed to any training regarding the Romanian language, and the expectations are that it is not able to make any *a priori* recognition of language specific patterns: how phrases are constructed, the particular order of words and their role as prepositions, conjunctions, adjectives, adverbs, nouns, or verbs distinguished according to the kind of idea that is to be conveyed and the function

performed in a sentence. The lack of such support for Romanian language means that our cognitive model has to consist of few entity types that have to be learned by the Watson ANN based on many text samples.

2.3. Recognize the salient relation types between entities

Relation types describe the manner in which the entity types interact with each other. For a given number of entity types n there could be created maximum $n*(n-1)$ relations between any two of them. This figure is given by the number of arrangements of n distinct entities taken 2 at a time. Consequently, the maximum number of possible relation types that could be established between any two entities grows rapidly with the number of entities. For instance, a cognitive model consisting of 6 entity types may employ as many as 30 relation types. The relation types help the machine learning algorithms create connections and rapports between entities and improve the ability to recognize the entity type contextually. For example, if we consider the entity types *ALGORITHM* and *PROPERTY* then the relation type *characterizeBy* could link these two entities in the sense that an algorithm is characterized by certain properties. There could be established a relation type from property to algorithm as well, regarding a property as being *specificTo* an algorithm. In the process of building a cognitive model there may be required relations between two entities in both directions, but it has to be considered the fact that a higher number of relation types raise the complexity of the model and the need for an extensive training process.

Hence, the number of relation types that are created between entities has to be judiciously considered. Moreover, as WKS documentation states: “for a relation mention to exist, text must explicitly define the relation and bind mentions of the two entities together, and must do so within a single sentence” (IBM, 2018b). Figure 3 illustrates the manual annotation of relation types applied on a training document. The cognitive model is based on 6 entity types and 11 relation types considered to be representative for how entity types rapport to one another.

2.4. The need of creating dictionaries associated to entity types

We mentioned earlier in the section that the entities we define for the cognitive model need to have concrete representation in terms of actual words that may play contextually the role envision for a given entity type. For example, the entity type *ACTIVITY* may have associated a dictionary containing Romanian words that designate various activities such as:

- *adunare*,
- *afişare*,
- *alege*,
- *analiza*,
- *apare*,
- *aplica*,
- *atribui*,
- *calcula*,
- *conţine*.

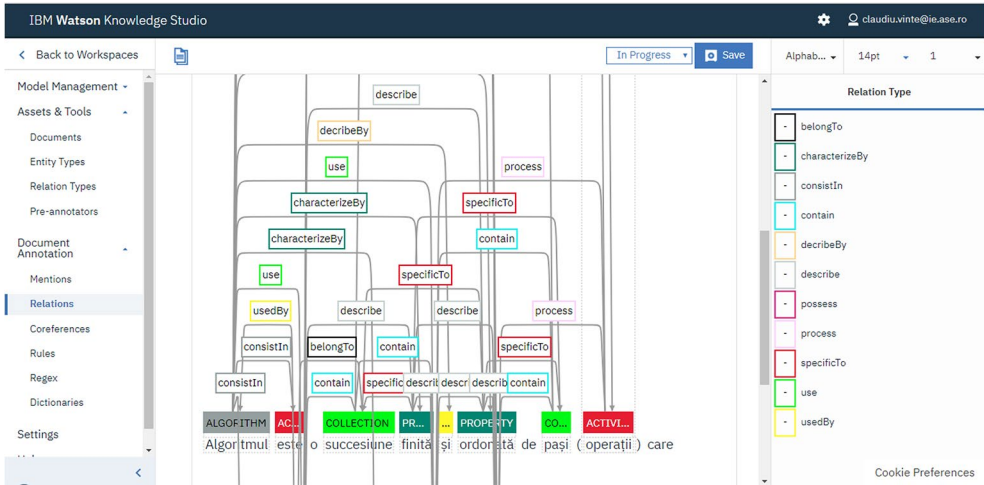


Figure 3. The manual annotation of relation types on a training document

It has to be observed that the words associated to an entity type may represent various parts of speech. The following parts of speech are the ones WKS interface offers to opt from when creating a new entry in a dictionary:

- Adjective,
- Adverb,
- Conjunction,
- Determiner,
- Interjection,
- Noun,
- Preposition,
- Pronoun,
- Quantifier,
- Unknown,
- Verb.

Watson Knowledge Studio provides a service for pre-annotating the documents used for training purpose, based on the previously created dictionaries. This means that when one of the words from the dictionary associated to an entity type is encountered in the text used for training, that word is annotated (marked with a specific colour in the web user interface) as being a mention of the entity type it belongs to. The pre-annotation process is very useful for two reasons:

- a) reduces the manual annotation load that would be required for categorizing the key-words in the text as belonging to a certain entity type (see Figure 4 below);
- b) once created, dictionaries can be downloaded from the cloud environment, saved locally and have them available for other potential cognitive models, new entity types or iterations of the current model.



Figure 4. The results of pre-annotation process applied on document used for training purpose

3. Model implementation and results

Developing a cognitive model for the content area of *Algorithms* implies defining the specific ideas that are intended to be checked for being covered in teachers' portfolios, such as those presented in section 2.1. Using these ideas as point of departure, we identified the entity types necessary for covering the concepts that the ideas refer to, and the relevant relation types that may be established between the entities.

3.1. Covering the content area of Algorithms – entity types, entity relations and dictionaries

For the content area of Algorithms, we defined 6 entity types, namely:

- *ACTIVITY*
- *ALGORITHM*
- *COLLECTION*
- *DATA*
- *EXPRESSION*
- *PROPERTY*

In terms of the relations types necessary to anchor the usage of the entity types in texts and to describe the logical connection between any two them, we proposed 11 relations types (Table 1).

To each entity type has been associated a dictionary containing the Romanian words that could be encountered in teachers' portfolios as being a contextual instance of an entity type. In order to identify these representative words for each entity type, we had to go through a set of representative portfolios and select the words manually.

Table 1. List of relation types along with the entity types that are put in in relation

Relation Type	First Entity Type	Second Entity Type
<i>belongTo</i>	DATA	COLLECTION
<i>characterizeBy</i>	ALGORITHM	PROPERTY
<i>consistIn</i>	ALGORITHM	ACTIVITY
<i>contain</i>	COLLECTION	DATA
<i>decribeBy</i>	ALGORITHM	EXPRESSION
<i>describe</i>	PROPERTY	DATA
<i>possess</i>	DATA	PROPERTY
<i>process</i>	ACTIVITY	DATA
<i>specificTo</i>	PROPERTY	COLLECTION
<i>use</i>	ALGORITHM	DATA
<i>usedBy</i>	DATA	ALGORITHM

There could be easily hundreds of words associated to an entity type. Romanian language in particular generate a higher complexity at this level because the verbs could be encountered in various forms at different tenses, nouns could be at singular or plural, they may also have genitive and dative cases, and so on.

3.2. Training the model through training documents

Training documents have to contain short paragraphs that describe as accurate as possible the concepts that have been selected to represent the ideas to be checked for being covered in teachers' portfolios. For example, if we are to define the notion of algorithm and point out the connection to the concept of activity implied for implementing an algorithm, then the texts included below represent examples of well formulated sentences for the training process: simple and straight to the point.

"Un algoritm este un proces sau un set de reguli care trebuie urmate în calcule sau alte operațiuni de rezolvare a problemelor, în special de către un computer."

"Algoritmul este o succesiune finită și ordonată de pași (operații) care transformă datele de intrare în date de ieșire."

If the idea was to make the connection between the entity types *ALGORITHM*, *DATA* and *ACTIVITY*: as the operations which are to be applied to data following the steps described by the algorithm, then the following text in Romanian would be a good candidate for being included in a training document.

"Datele cu care lucrează algoritmi sunt:

- date de intrare - preluate prin citire,
- date intermediare - aflate în urma unor operații,
- date de ieșire - transmise prin scriere (afișare)."

Training documents can be managed more efficiently if they are short and repeat a certain idea multiple times, in terms of word orders.

3.3. Deploying the model for querying actual teachers' portfolios

Once the cognitive model has been created in Watson Knowledge Studio environment, and the training process has taken place, if the pattern recognition ability of the model was greater than 80% then the model can be deployed to Watson Discovery services (WD). In the WD environment can be created special queries which are then applied to new portfolios that are added to the content specific collection. These queries are applied to non-structured data, such as the teachers' portfolio previously uploaded into collection as .PDF files. For example, a query looking for the idea of *using expressions* (logical, arithmetical etc.) *for implementing algorithms* may look like the following one:

```
enriched_text.entities:(text:(logic*|arithmetic*,type::expression),enriched_text.entities:(text:(algo*|program*),type::algorithm)
```

These queries are organized in batteries for each content domain, 3 questions for each idea whose existence in the text we want to verify. Then the queries are run successively, in a programmatic manner against the target document. The return of such query executions is in the form of passages from the target text that contain the concepts which have been queried for, if any such passages are retrieved. For example, the following set of ideas are to be pursued for the content domain *Algorithms*:

1. *An algorithm has the following properties: clarity, generality, finiteness and efficiency*
2. *An algorithm is a sequence of steps that describes how to solve a problem*
3. *An algorithm uses input, output and/or intermediate data*
4. *An algorithm works with several types of data, namely: numerical, logical, strings, real, image, sound, etc.*
5. *The operators can be: arithmetic, relational and logical*
6. *Any logical scheme can be decomposed into three basic structures, namely: LINEAR, ALTERNATIVE AND REPETITIVE*
7. *We can use the following categories of operators: mathematical operators (+, -, *, /, %); relational operators (<, <=, >, >=, ==, !=); logical operators (! - no, && - and, || - or)*
8. *Among the operations that compose an algorithm we find: the reading operation, the assignment operation, the writing operation*
9. *In graphical environments, blocks of several types are used for algorithms: operator type, data type, event type, etc.*

For each idea, whose existence in the text we want to verify, we designed 3 queries sorted from the strictest one, to the most relaxed one. For the idea:

An algorithm is a sequence of steps that describes how to solve a problem

the corresponding set of queries in Watson Discovery Language (WDL), based on the cognitive model previously created, and which has defined and trained entity types and relation types specific to the content domain, are the following, from the strictest to the most relaxed one:

- a) `nested(enriched_text.relations).filter(enriched_text.relations.type:"Defined_By").term(enriched_text.relations.sentence,count:5)`

- b) `nested(enriched_text.entities).filter(enriched_text.entities.type::"Algorithm").term(enriched_text.entities.text,count:10),nested(enriched_text.entities).filter(enriched_text.entities.type::"Definition").term(enriched_text.entities.text,count:5)`
- c) `nested(enriched_text.entities).filter(enriched_text.entities.type::"Definition").term(enriched_text.entities.text,count:5)`

3.4. Running queries programmatically through the usage of Java API

The interaction with IBM Watson Discovery service can be done either via the web-based graphical user interface or by using the application programming interface (API) provided for one of the following programming languages: Java, cURL, NodeJS and Python (IBM, 2018a). We developed a Java software solution, using Java SDK version 11, that takes as input data a specified content area from the national curriculum for Informatics instruction intended for the 5th grade students, and a portfolio created for the specified content domain (IBM, 2018c). The output generated by the server side of the software solution provides a list of ideas that have been checked for being covered in the text, along with the most relevant 5 passages that contain the looked after concepts, if there is any. The following pseudocode details the automated process of evaluating teachers' portfolios, file by file, by creating collections of documents that incorporate the cognitive model developed previously for each of those 8 content domains that have to be covered according to the 5th grade national curriculum for Informatics instruction. The algorithm assigns grades based on the results returned by running each query, as follows:

- a) if the strictest query for a pursued idea returns passages from the document, then a grade of 10 is assigned; if the strictest query does not return any passage, then the next query, less strict, is run;
- b) if the next query, less strict, returns passages, then grade 8 is assigned. Otherwise, it goes to the third query, the most relaxed one;
- c) if the third query, the most relaxed for the idea pursued, returns passages, then grade 6 is awarded;
- d) if not even the most relaxed question returns passages from the document, then a grade of 4 is given for finding the idea in question in the teacher's document.

For each document and content domain, the algorithm computes an arithmetic mean of the grades attributed based on running the queries for each idea to be pursued.

```
//Upload Watson Discovery environment configuration "config.properties" file
//Initialization: create a new Discovery instance with environment
username = discovery.username
password = discovery.password
url = discovery.url
instance = discovery.instance

discovery = new Discovery()
discovery.setEndPoint(url)
discovery.setUsernameAndPassword(username, password)
```



```

environmentId = environment.id
configurationId = configuration.id
collectionId = collection.id

//Initialize the list of ideas from ideas.txt file
Ideas = read(ideas.txt)
//Initialize the list of queries from queries.txt file, 3 queries
per idea
Queries = read(queries.txt)

//Create a new collection of documents
getCollectionOptions = new getCollectionOptions.Builder(environment-
tId, collectionId).build()
getCollection = discovery.getCollection(getCollectionOptions).exe-
cute()

//Upload a new document (filename) to collection
createDocumentBuilder.filename(fileName)
discovery.addDocument(createDocumentBuilder.build()).execute()
documentId = createDocumentResponse.getDocumentId()

//Make sure the document is ready to be queried according to the
cognitive //model associated with the collectionId the document was
uploaded to
while (documentId.Status is not READY) do
    discovery.getDocumentStatus(getDocumentStatusOptions).execute()
    if (documentId.Status is PROCESSING) then
        wait (for 500 milliseconds)
    end if
end while

foreach (i in Ideas) do
    foreach (q in Queries) do
        queryResponse = discovery.query(queryOptions.build()).execute()
        queryPassagesList = queryResponse.getPassages()
        //If passages are retrieved for the applied query
        if ((queryPassagesList.size() > 0) then
            // Save the number of passages
            foreach (passage in queryPassagesList) do
                // Save the passages in passages.txt file
                passage.getPassageText()
                // Save the passage's score
                passage.getPassageScore()
            end foreach
        end if
    end foreach
end foreach

```

It has to be noted that Discovery Java API does allow to query individual files uploaded into collection, but the query is applied to an entire collection of documents. Consequently, in order to attempt retrieving passages at the document level, the algorithm has to process one document uploaded into collection at a time.

3.5. Results of the automated curriculum assessment and the analysis conducted at the national level

In order to provide the very looked after feedback to the parties with decisional role involved in the training process of the teachers nationwide, we put together the results of the automated curriculum assessment of all teachers' portfolios in a tabular format. The overall pool of teachers that participated in the training program, and developed portfolios for various content domains at their choice, are considered the individual observations in a multivariate analysis model. The recorded variables of the model are constituted by the county name the school that teacher works for belongs to (as categorical variable, used for further grouping), and the content domains. Specifically, the values of observed variables represent the grades obtained by each teacher for the portfolios they developed, according to their choices of content domains. Consequently, the multivariate model consists of 1917 observations and 8 quantitative variables. Table 2 below is an excerpt from the actual data, for providing the perspective of how it is structured. In order to comply with GDPR requirements, we anonymized the data for the column that contains the name of the teacher who participated in the training program.

Table 2. Teachers evaluated portfolios based on the domains they created content for

Teacher	County	Safety & Ergonomics	Systems & Communication	System Architecture	Peripheral Devices	Operating Systems	Internet	Graphic Editors	Algorithms
1	Alba	0	0	0	0	7.82	0	0	6.4
2	Alba	0	0	0	0	0	0	0	7
3	Alba	0	0	0	0	0	0	0	7.5
4	Alba	0	8	4	0	0	0	0	7.5
5	Alba	0	0	0	0	0	0	0	8.8
6	Alba	0	0	5.85	0	8.35	7.33	6	9
7	Alba	0	0	0	0	0	0	0	9.05
8	Alba	7.2	10	6.15	8.3	7.89	8.33	4	9.2
9	Alba	0	0	4	0	0	0	5	9.6
10	Alba	6.6	8	5.75	8	0	7.33	7	9.6
11	Alba	0	0	0	0	0	0	0	10
.
.
.
1911	Vrancea	0	0	4	0	0	0	0	6
1912	Vrancea	0	0	0	0	0	0	4	6
1913	Vrancea	0	0	0	0	0	0	4	6
1914	Vrancea	0	0	6.4	8.34	7.87	7.67	4	6
1915	Vrancea	0	0	4.7	0	6.2	7.67	5	6
1916	Vrancea	0	4	0	0	6.84	7.33	6.67	6
1917	Vrancea	7.8	8	4.3	8.33	7.47	6	6.33	7.6

Based on the individual observations, represented by the coverage quality of the specific content domains each teacher’s portfolios refers to, we conducted a cluster analysis in order to discover how teachers relate to each other in connection to the number of content domains they tackled in their portfolios, and how well they covered those content areas, based on the assessments supplied by the cognitive computing process. We implemented the cluster analysis in Python programming language, version 3.7. Figure 5 emphasizes the grouping of those 1917 observations (teachers) in the field of content domains (number of content area covered in their portfolios – X axis) and the grades obtained by their portfolios as a result of the cognitive computing assessment (grade levels – Y axis). It has to be noticed that good majority of teachers developed portfolios that covered more than 3 content domains, and they generally obtained grades over 4. The clustering also shows that there are teachers who did not covered certain content domains and consequently received zero as grade. The cluster analysis in Python made use of *numpy* (1.19.1), *pandas* (1.1.0), *sklearn* (0.0) and *matplotlib* (3.3.0) packages. The clustering model used K-Means algorithm parametrized for retrieving maximum 10 unique clusters.

There emerges the need to obtain a perspective regarding the manner in which the counties, and hence the regions of the country relate to the nationally conducted training process and the quality of portfolios that have been developed regionally.

We aggregate the data contained in Table 1, by grouping the teachers (the observations) based on county the school they work for belongs to. The values of the observed variables (8 content domains) associated to the teachers belonging to the same county have been aggregated using the following formula (1) for computing the average grades on columns only for the cells that have positive values at the individual observation level.

$$\bar{X}_{ik} = \frac{\sum_{j=1}^{N_k} \{x_{ij}, x_{ij} > 0\}}{\sum_{j=1}^{N_k} \{1, x_{ij} > 0\}}, \tag{1}$$

where $i = \overline{1,8}$, and N_k is the number of observations for each county, $k = \overline{1,42}$.

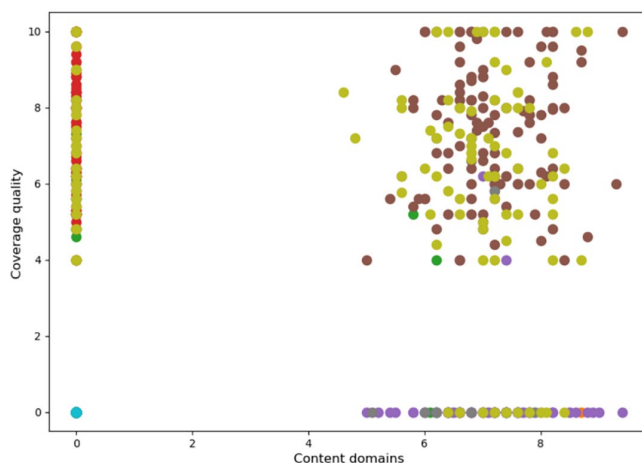


Figure 5. 10-means clustering of 1917 observations in the field of content domains and quality coverage

Table 3. Aggregated average content domain grades per county

County	Safety & Ergonomics	Systems & Communication	System Architecture	Peripheral Devices	Operating Systems	Internet	Graphic Editors	Algorithms
Alba	7.41	6.64	5.83	7.7	7.57	7.32	6.16	7.7
Arad	7.2	7.06	5.04	7.53	7.51	7.13	5.13	7.35
Arges	7.7	6.8	5.97	8.78	7.6	7.88	6.22	7.88
Bacau	7.08	7.64	5.97	7.43	7.38	7.63	4.1	0
Bistrita Nasaud	6.88	7.18	5.3	7.58	7.23	6.95	4	7.2
Botosani	6.15	7	4.77	7.33	0	0	0	5.8
Braila	7.13	7.71	5.61	7.58	7.57	7.27	4.1	0
Brasov	7.68	7.18	6.03	6.94	7.55	7.41	5.64	8.05
Bucuresti	0	8.1	4.17	6.6	7.53	8.13	4.97	7.33
Calarasi	7.05	7.56	6.1	6.13	7.77	7.67	7.42	8.14
Caras-Severin	7.5	6.77	5.49	8.33	7.56	7.24	4.13	0
Cluj	7.03	8.1	4.64	6.82	7.51	7.36	5.1	7.94
Constanta	6.43	8.2	4.3	7.2	7.47	7.45	4.46	7.6
Covasna	8.04	7.04	5.49	6.25	7.68	7	5.86	7.64
Dambovita	7.21	7.05	5.63	7.48	7.53	7.8	6.33	7.87
Dolj	7.2	6.64	6.46	7.8	7.66	7.6	5.98	8.35
Galati	8.85	5.73	5.52	7.49	7.82	7.6	5.98	8.32
Giurgiu	0	0	0	0	7.33	7.11	4.25	8.45
Gorj	7	7.52	5.4	8.11	7.64	7.73	4	8.24
Harghita	7.17	7.8	4.82	8.5	7.49	7.67	4.58	7.2
Hunedoara	6.98	7.87	5.95	7.64	7.57	7.13	4.62	8.25
Ialomita	7.08	7.6	5.8	7.17	7.44	7.64	5	8.72
Iasi	9	8	7.5	7.83	7.73	7.13	4.11	9.35
Ilfov	6.4	9.27	5.6	0	7.48	7.44	6.45	8.62
Maramures	0	6.6	6.29	7.03	7.72	7.33	7.31	8.23
Mehedinti	7.4	6.6	5.56	5.94	7.34	8.25	5.33	8.5
Mures	7	4.8	5.38	0	0	0	0	7.6
Neamt	6.6	4	4	0	0	0	0	0
Olt	7.77	6.53	5.26	7.62	7.64	7.45	5.4	7.36
Prahova	7.13	7.53	4.79	6.96	7.73	7.59	4.34	7.8
Salaj	6.8	7.19	5.33	7.33	7.97	7.29	4.13	7.2
Satu Mare	6.85	7.32	5.1	6.48	7.62	7.46	4.08	6.64
Sibiu	6.91	6.9	6.21	7.81	7.73	7.42	4.15	7.79
Suceava	7.13	7.21	6.29	7.46	7.6	7.37	5.8	7.84
Timis	6.86	6.98	4.56	7.79	7.48	7.59	4.98	7.43
Tulcea	0	0	4.88	0	7.28	6.84	5.34	5.6
Valcea	7.28	7.26	5.57	7	7.67	7.47	6.92	8.09
Vaslui	6.61	7.45	4.88	7.07	7.71	7.52	4.46	7.96
Vrancea	6.6	8.4	6.74	5.67	7.61	7.4	5.47	7.6

As result of the county aggregation process, we obtain the below Table 3. There are 3 counties from which there were no teachers to developed relevant portfolios, hence Table 3 contains 39 out of 42 Romania’s counties.

We conducted cluster analysis on the aggregated data, in order to identify how counties are get grouped based on the quality of portfolios they deliver through their teachers. We employed K-Means algorithm for the clustering model, and parametrized it accordingly in order to retrieve maximum 10 unique clusters. Figure 6 shows that the grouping replicates, to a certain extent, the clusters obtained departing from the individual portfolios’ coverage quality. It is even more evident that, in average, most of the portfolios developed at the county level covered more than 6 content domains, and obtained grades above 5.5. These results express synthetically that the training process has been conducted efficiently and homogeneously across the country and that the outcome is a fairly well-prepared cohort of teachers ready to deliver acceptable Informatics instruction in the Romanian secondary schools. We have to point out that the highest level of quality portfolios, with average grades above 8.5, were developed in just a handful of counties by relatively few teachers.

Furthermore, we computed the average grade per content domains at the county level. The values associated to the observed variables (8 content domains) have been aggregated for each county using the following formula (2) for computing the average grade on lines only for the cells that have positive values at the content domain level.

$$\bar{X}_j = \frac{\sum_{i=1}^8 \{x_{ij}, x_{ij} > 0\}}{\sum_{i=1}^8 \{1, x_{ij} > 0\}}, \tag{2}$$

where $j = \overline{1, 42}$.

Based on the average grades that agglutinate all the content domains’ scores at the county level and employing GIS technologies we were able to provide a national perspective regarding the quality of portfolios created by teachers as outcome of the training sessions. This aggregated perspective is depicted in Figure 7.

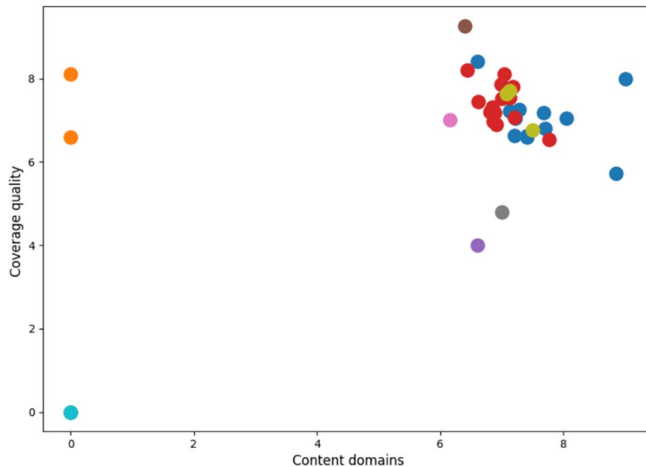


Figure 6. 10-means clustering of Romania’s 42 counties in the field of content domains and coverage quality

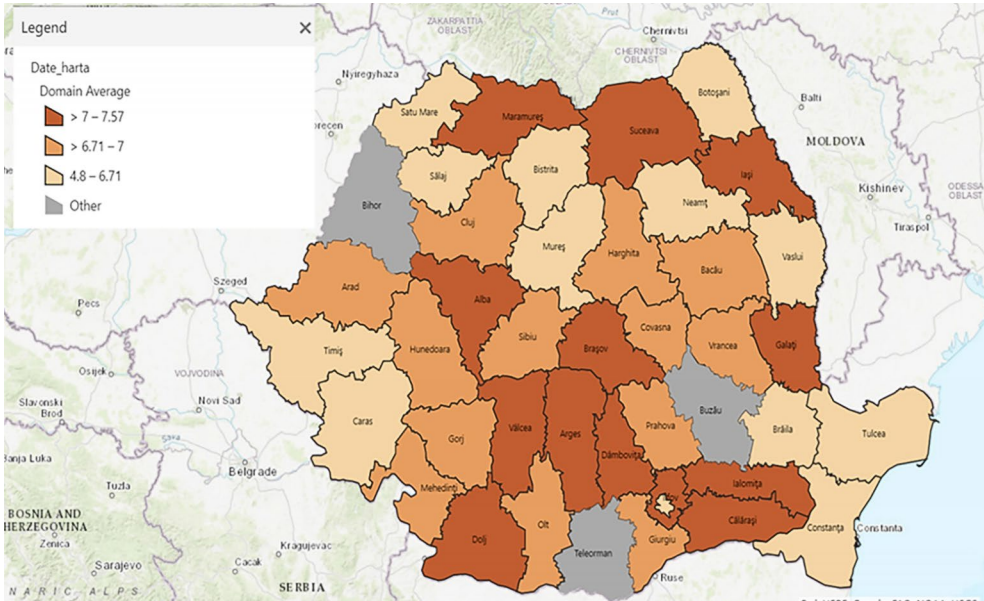


Figure 7. Aggregated average marks of content domain portfolios at the county level

The ArcGIS Maps for Office component is built by ESRI as part of the ArcGIS platform. ESRI GIS technologies and resources provide support for Excel as well as PowerPoint (Flanagan, 2017). The component attaches to Office applications through add-in functional modules and allows viewing data organized in tabular form, such as spreadsheets on maps (Dârdală et al., 2019). The attributes that represent geographic data are automatically identified and the user can represent them on the map using different symbology based on non-spatial attributes for representation.

In order to identify how the counties relate to each other, in terms of portfolios' quality developed by their teachers, we additionally conducted hierarchical cluster analysis (HCA) on the data aggregated based on formula (1). In addition to the Python packages mentioned earlier, we employed for HCA the *scipy* (1.5.2) package.

We employed the agglomerative clustering algorithm, using the complete linkage method along with the Euclidian metric. In the agglomerative algorithm we employ the complete link method (max proximity) for grouping. The complete link or the full bond defines the cluster proximity as the farthest distance between two points, x and y , which are in different clusters, A and B :

$$d(A, B) = \max \{d(x - y)\} \text{ for } x \in A, y \in B. \quad (3)$$

The selection of clusters is considered based on the minimum distance between them. We used the full because bond is less susceptible to noise and outliers. The resulted dendrogram is showed in Figure 8. The optimal partition generated only 3 clusters, result which strengthens the previous observation that the content domain coverage quality is fairly homogeneous across the country.

Leveraging even further the cluster analysis, we conducted HCA for the observed variables of the model taking into account the individual values of all the observations (Table 2).

The observed variables consist in those 8 content domains of the national curriculum. We employed the same agglomerative clustering algorithm, using the complete linkage method along with the correlation between the variables as metric. The resulted dendrogram is showed in Figure 9.

It stands out that the coverage of content domain Algorithms has been conducted distinctively from the other content areas. The content domains Graphic Editors, Internet and Operating Systems received greater, and were treated in a higher qualitative manner in teachers' portfolios.

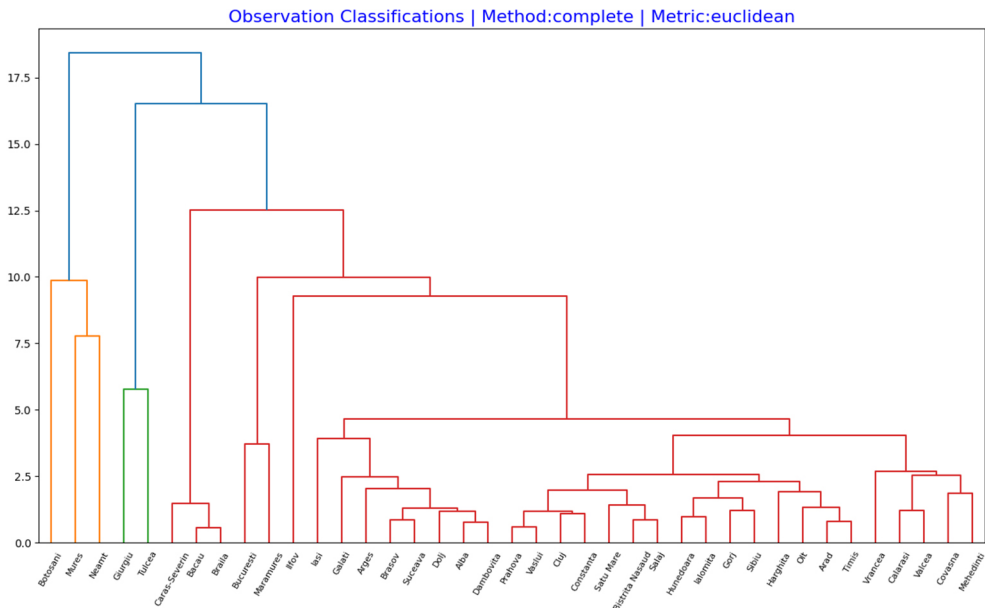


Figure 8. Hierarchical clustering of Romanian counties based on content domain coverage quality

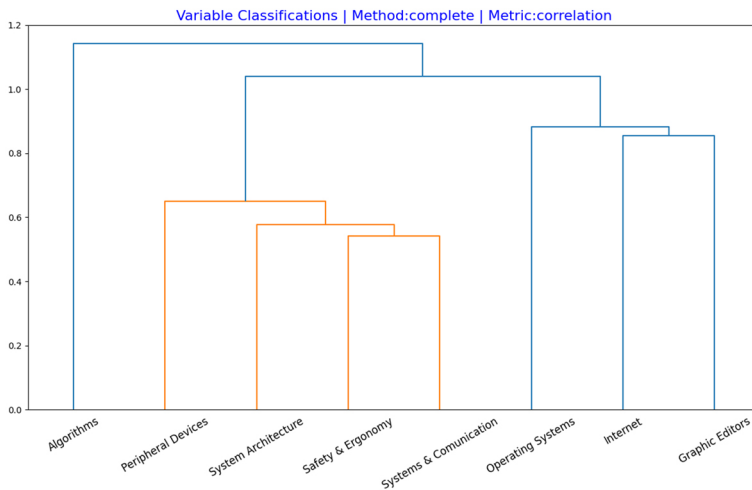


Figure 9. Hierarchical clustering of content domains variables based on individual observations

Conclusions and further research

This paper presents the outcome of our research conducted to identify and develop a cognitive computing solution for an assessing, in a programmatic manner, large corpora of curriculum content created by lower secondary school teachers of Informatics in Romania. In an effort to provide access to class supporting content for the discipline, we created a wiki-based platform through which a community over 2300 teachers at the national level shares their expertise and class experience regarding Informatics instruction and dissemination of their computer programming and communication technology expertise. The result of this initiative at the national level is corpora of curricular content that must be evaluated to verify the degree to which the material meets the requirements of the national curriculum for the discipline. We addressed this critical yet tedious process by designing and implementing a solution for providing automated feedback to teachers through cognitive computing. The outcome of our research and the implemented solution proves that both data representation and querying in IBM Watson have significant advantages over a simple text search approach in that they are built to reflect human knowledge representation. Most importantly, a cognitive query will return its results with their associated context, allowing a human user to better process them and better understand their usefulness.

Cognitive queries also address the issue of high degree of variation in the representation of concepts by human creators in free form text. Searching in a traditional fashion would quickly lead the problem to intractability as it would be necessary to represent all possible combinations. Since Watson ANNs learn from the examples which are provided during the training phase, the cognitive solution is capable of finding related representations of the entity types and relation types which are the subject of the search without the need of specifying them directly. The feedback that we are able to provide through the automatic curriculum assessment, supported by the cluster analysis conducted on the evaluation results concern the following findings:

- a) the results express synthetically that the training process has been conducted efficiently and homogeneously across the country, most of the teachers' portfolios received grades in the range between 6 and 8; hence the outcome of the nationwide training program is a fairly well-prepared cohort of teachers ready to deliver acceptable Informatics instruction in the Romanian secondary schools;
- b) average grades above 8.5 were attained in just a handful of counties by relatively few teachers; this support the hypothesis that many of the teachers come from other domains than Informatics, and that the training initiative of the Ministry of Education was in fact well received and opportune;
- c) the best portfolios have not been created necessarily in counties or regions with representative academic institutions; Bucharest, Timisoara, Cluj and Constanta for example, which are important university centers in Romania, scored relatively modest grades; there may have been motivational factors at play for teachers coming from smaller cities and less populated areas who perceived the new discipline of Informatics as challenging and tackled the training process more seriously.

In terms of potential limitations, the cognitive computing solutions that we proposed along with the developed software application, provide an assessment mechanism which has primordially the role of verifying the degree of coverage that teachers' portfolios provide for

a given content area of the curriculum. In addition to that, the automated process of portfolio assessments cannot be applied right away to a new content area or field of knowledge. It demands, as prerequisites, the construction of a domain specific cognitive model, a dedicated training process which makes use of content specific documents, particular ideas translated into entity and relations types to search for in new documents. In other words, the whole approach has to be tailored to domain specific knowledges that have to be well understood by a human specialist in the first place. These are fundamental limitations which are related to the current developments in artificial intelligence (AI), which are beyond the scope of this paper. On the other hand, this approach to curriculum assessment, through which passages that tackle targeted concepts in new documents are automatically retrieved, has undeniable benefits in terms of consistency and duration of the assessment process, particularly when there are thousands of documents to be evaluated.

Our ongoing research aims to conceive technologically sustainable frameworks that respond in a timely fashion to future developments in the area of curriculum content assessment automation, in order to empower the responsible policy makers with advance instruments for creating economically viable and adaptive education strategies.

Author contributions

Conceptualization, Data curation, Formal analysis, C.V.; Supervision, I.S., M.D., Writing–original draft, C.V. and A.R., Writing–review & editing, C.V., A.D., I.S. and M.D.

Disclosure statement

Authors do not have any competing financial, professional, or personal interests from other parties. The authors declare no conflict of interest.

References

- Barendsen, E., & Steenvoorden, T. (2016). Analyzing conceptual content of international informatics curricula for secondary education. In A. Brodnik, & F. Tort (Eds.), *Lecture notes in computer science (LNCS): Vol. 9973. Informatics in schools: Improvement of informatics knowledge and perception* (pp. 14–27). Springer. https://doi.org/10.1007/978-3-319-46747-4_2
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48–54. <https://doi.org/10.1145/1929887.1929905>
- Barroso, R. M. R., Ferreira, F. A. F., Meidute-Kavaliauskiene, I., Banaitiene, N., Falcao, P. F., & Rosa, A. A. (2019). Analyzing the determinants of e-commerce in small and medium-sized enterprises: A cognition-driven framework. *Technological and Economic Development of Economy*, 25(3), 496–518. <https://doi.org/10.3846/tede.2019.9386>
- British Department for Education. (2013). *Computing programmes of study: Key stages 1 and 2. National curriculum in England*. <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study>
- Caspersen, M. E., Gal-Ezer, J., McGettrick, A., & Nardelli, E. (2018). *Informatics for All: The strategy*. ACM Europe & Informatics Europe. <https://doi.org/10.1145/3185594>
- Chang, X., & Li, J. (2019). Business performance prediction in location-based social commerce. *Expert Systems with Applications*, 126, 112–123. <https://doi.org/10.1016/j.eswa.2019.01.086>

- Cid-López, A., Hornos, M. J., Carrasco-González, R. A., & Herrera-Viedma, E. (2018). Prioritization of the launch of ICT products and services through linguistic multi-criteria decision-making. *Technological and Economic Development of Economy*, 24(3), 1231–1257. <https://doi.org/10.3846/tede.2018.1423>
- Danenas, P., Skersys, T., & Butleris, R. (2020). Natural language processing-enhanced extraction of SBVR business vocabularies and business rules from UML use case diagrams. *Data & Knowledge Engineering*, 128, 101822. <https://doi.org/10.1016/j.datak.2020.101822>
- Dárdalá, M., Furtună, T. F., & Ioniță, C. (2019). Design and implementation of a software component for geospatial data visualization in Excel. In *Proceedings of the IE 2019 International Conference* (pp. 293–298). <https://doi.org/10.12948/ie2019.04.22>
- Diana, N., Eagle, M., Stamper, J., & Koedinger, K. R. (2017). Teaching informal logical fallacy identification with a cognitive tutor. In *International Conference on Artificial Intelligence in Education*. Springer, Cham.
- Diethelm, I., & Schaumburg, M. (2016). IT2 school development of teaching materials for CS through design thinking. In A. Brodник, & F. Tort (Eds.), *Lecture notes in computer science (LNCS): Vol. 9973. Informatics in schools: Improvement of informatics knowledge and perception* (pp. 193–198). Springer. https://doi.org/10.1007/978-3-319-46747-4_16
- Dowek, G. (2016, October 13–15). Elements to define a coherent curriculum for the K12 education: The example of France. In *Informatics in schools improvement of informatics knowledge and perception. Proceedings of the 9th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, ISSEP 2016*. Münster, Germany. Springer.
- Ertmer, P. A., & Ottenbreit-Leftwich, A. T. (2010). Teacher technology change: How knowledge, confidence, beliefs, and culture intersect. *Journal of Research on Technology in Education*, 42(3), 255–284. <https://doi.org/10.1080/15391523.2010.10782551>
- Fethi, I. A., & Lowther, D. L. (2010). Factors affecting technology integration in K-12 classrooms: A path model. *Educational Technology Research and Development*, 58(2), 137–154. <https://doi.org/10.1007/s11423-009-9132-y>
- Flanagan, M. (2017). *ArcGIS maps for office: Your maps in Excel and PowerPoint*. http://proceedings.esri.com/library/userconf/fed17/papers/fed_08.pdf
- Guerra, V., Kuhnt, B., & Blöchliger, I. (2012). *Informatics at school – Worldwide. An international exploratory study about informatics as a subject at different school levels*. University of Zurich, Switzerland. https://www.researchgate.net/publication/275031370_Informatics_at_school_-_worldwide_An_international_exploratory_study_about_informatics_as_a_subject_at_different_school_levels
- Hernández-Lara, A. B., Perera-Lluna, A., & Serradell-López, E. (2019). Applying learning analytics to students' interaction in business simulation games. The usefulness of learning analytics to know what students really learn. *Computers in Human Behavior*, 92, 600–612. <https://doi.org/10.1016/j.chb.2018.03.001>
- Hurwitz, J., Kaufman, M., & Bowles, A. (2015). *Cognitive computing and big data analytics*. John Wiley & Sons. <https://www.wiley.com/en-us/Cognitive+Computing+and+Big+Data+Analytics-p-9781118896624>
- Iandoli, L., Ponsiglione, C., & Zollo, G. (2010). Modelling networked cognition: A socio-computational approach. *Economic Computation and Economic Cybernetics Studies and Research*, 4(44), 121–141. https://www.researchgate.net/publication/292491734_Modelling_networked_cognition_A_socio-computational_approach
- Informatica în școli. (2020). *Ghid pentru profesorii de clasa a V-a*. www.informaticainscoli.ro
- Informatics education: Europe cannot afford to miss the boat*. (2013). Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education. April 2013.
- International Business Machines. (2018a). *IBM Cloud API Docs. Java Discovery*. <https://cloud.ibm.com/apidocs/discovery?code=java>

- International Business Machines. (2018b). *IBM Cloud Docs. Watson Knowledge Studio*. <https://console.bluemix.net/docs/services/watson-knowledge-studio/typesystem.html#typesystem>
- International Business Machines. (2018c). *IBM Watson Discovery API. Java Class Discovery*. <https://watson-developer-cloud.github.io/java-sdk/docs/master/com/ibm/watson/discovery/v1/Discovery.html>
- International Business Machines. (2020). *IBM Watson Products and Services*. <https://www.ibm.com/watson/products-services/>
- Lee, J. (2013). Cross-disciplinary knowledge: desperate call from business enterprises in coming smart working era. *Technological and Economic Development of Economy*, 19(Supp1), S285–S303. <https://doi.org/10.3846/20294913.2013.880080>
- Lytras, M., Visvizi, A., Damiani, E., & Mthkour, H. (2019). The cognitive computing turn in education: Prospects and application. *Computers in Human Behavior*, 92, 446–449. <https://doi.org/10.1016/j.chb.2018.11.011>
- McCalla, G. (2000). The fragmentation of culture, learning, teaching and technology: Implications for the artificial intelligence in education research agenda in 2010. *International Journal of Artificial Intelligence in Education*, 11(2), 177–196.
- Micheuz, P. (2006). *Is it Computer Literacy, IT, ICT or Informatics? What is going on in Austria's compulsory schools in the context of educational standards?* http://sedici.unlp.edu.ar/bitstream/handle/10915/24357/Documento_completo.pdf%3Fsequence%3D1
- Modha, D. S., Ananthanarayanan, R., Esser, S. K., Ndirango, A., Sherbondy, A. J., & Singh, R. (2015). *Cognitive Computing*, *Communications of the ACM*, 54(8), 62–71. <https://dl.acm.org/doi/10.1145/1978542.1978559>
- Neti, C. (2018, June 26). *How Watson Education, Scholastic and Edmodo are using AI to close the learning gap*. <https://www.ibm.com/blogs/watson/2018/06/using-ai-to-close-learning-gap/>
- Paes de Faria, A. C. C., Ferreira, F. A. F., Dias, P. J. V. L., & Cipi, A. (2020). A constructivist model of bank branch front-office employee evaluation: An FCM-SD-based approach. *Technological and Economic Development of Economy*, 26(1), 213–239. <https://doi.org/10.3846/tede.2020.11883>
- Pereira, C., & Tikhonenko, S. (2017). *Informatics education in Europe: Institutions, degrees, students, positions, salaries. Key data 2011–2016* (An Informatics Europe Report). Informatics Europe.
- Serban, D., Pelau, C., & Dinca, V. M. (2019). Panel data analysis for measuring the impact of e-skills on the ecological behavior of individuals. *Economic Computation and Economic Cybernetics Studies and Research*, 53(1), 57–74. <https://doi.org/10.24818/18423264/53.1.19.04>
- Sysło, M. M., & Kwiatkowska, A. B. (2015). Introducing a new computer science curriculum for all school levels in Poland. In A. Brodnik & J. Vahrenhold (Eds.), *Lecture notes in computer science: Vol. 9378. ISSEP 2015: Informatics in Schools. Curricula, Competences, and Competitions* (pp. 141–154). Springer. https://doi.org/10.1007/978-3-319-25396-1_13
- Thurai, A. (2017). *What can cognitive computing do for you?* <https://www.ibm.com/blogs/cloud-computing/2017/08/cognitive-computing-watson/>
- Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2017). Computer science in K-12 school curricula of the 21st century: Why, what and when? *Education and Information Technologies*, 22, 445–468. <https://doi.org/10.1007/s10639-016-9493-x>
- Wenger, E. (1987). *Artificial intelligence and tutoring systems: Computational and cognitive approaches to the communication of knowledge*. Morgan Kaufmann Publishers. <https://dl.acm.org/doi/10.5555/42185>
- Wilson, M. (2018). *Why cloud AI is a solid bet*. <https://www.ibm.com/blogs/cloud-computing/2018/03/cloud-ai-solid-bet/>
- World Economic Forum. (2016). *The future of jobs employment, skills and workforce strategy for the fourth industrial revolution*. Executive Summary. <http://reports.weforum.org/future-of-jobs-2016/chapter-1-the-future-of-jobs-and-skills/>