

An Adaptive Algorithm for Effective Selection of Training Content for IT Professionals

Volodymyr Sokol^a, Mariia Bilova^a, Andrii Druzhynin^a, Nikita Cherkun^a and Ivan Perepelytsia^b

^a National Technical University "Kharkiv Polytechnic Institute", 2, Kyrpychova str., Kharkiv, 61002, Ukraine

^b Academy Smart Ltd., 67A, 23-ho Serpnya str., Kharkiv, 61000, Ukraine

Abstract

Rapidly evolving IT industry requires software developers to become a part of the lifelong learning process. IT companies benefit from constant increment in skills of their employees and usually support them in terms of corporate training. On the other hand, manual creation of individual learning path for every employee provides additional challenge for human resource management. Therefore, we suggest adaptive learning approach as a solution to that issue, so it becomes an integral part of the corporate knowledge management system.

Adaptive learning is widely used in academic education. However, in corporate environment we encounter different models, needs, data sources and goals. In this work we propose an adaptive learning algorithm and model, which is based on European e-Competence Framework (eCF) approach and therefore is suitable for corporate education.

The algorithm is an example of a knowledge-based recommender system. It performed well on testing data set, since it significantly reduces amount of time, which is needed for an employee training. However, the algorithm has several assumptions, limitations, and therefore we may define future directions for its improvement.

Keywords 1

e-CF, adaptive learning, recommender system

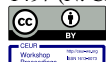
1. Introduction

Informational technologies industry evolves and grows with high velocity. Consequently, it introduced significant changes to the notion of “up-to-date” knowledge or technology, which may become obsolete in a short period of time. A recent survey of the developer ecosystem [1] made by JetBrains lists 34 programming languages which were used by respondents in the last twelve months. Each language in its turn uses dozens of different frameworks and libraries. For instance, the survey lists 14 major JavaScript frameworks. VueJS, the third most popular framework was originally released only in 2014 [2]. Therefore, the need to constantly obtain new knowledge and improve their skills becomes crucial for IT professionals.

In other words, IT professionals have to incorporate lifelong learning process into their career development path. Lifelong learning (LLL) is the continuous building of skills and knowledge throughout the life of an individual. [3]. IT corporations benefit from LLL activities of their employees as well, since trained and skilled personnel generates more value over time. Companies may support education processes providing trainings, learning materials, and dedicated tools such as learning, training, and knowledge management systems (LMS, TMS, KMS appropriately) [4].

Each employee has to possess specific knowledge and skills in order to fulfill particular project role or job profile. Such set of knowledge and skills is specific for each employee. It may take significant

COLINS-2021: 5th International Conference on Computational Linguistics and Intelligent Systems, April 22–23, 2021, Kharkiv, Ukraine
EMAIL: vlad.sokol@gmail.com (V. Sokol); missalchem@gmail.com (M. Bilova); andrii.drz@gmail.com (A. Druzhynin);
cherkunmv@gmail.com (N. Cherkun) ; IvanPerepelytsya@gmail.com (I. Perepelytsia)
ORCID: 0000-0002-4689-3356 (V. Sokol); 0000-0001-7002-4698 (M. Bilova); 0000-0001-8579-701X (A. Druzhynin); 0000-0002-8369-1497 (N. Cherkun); 0000-0001-7683-8780 (I. Perepelytsia)



© 2021 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

resources to create manually individual learning plan for every person in a company. We address that issue by introducing adaptive learning approach, which adapts existing learning materials and courses to the needs of a particular employee [5].

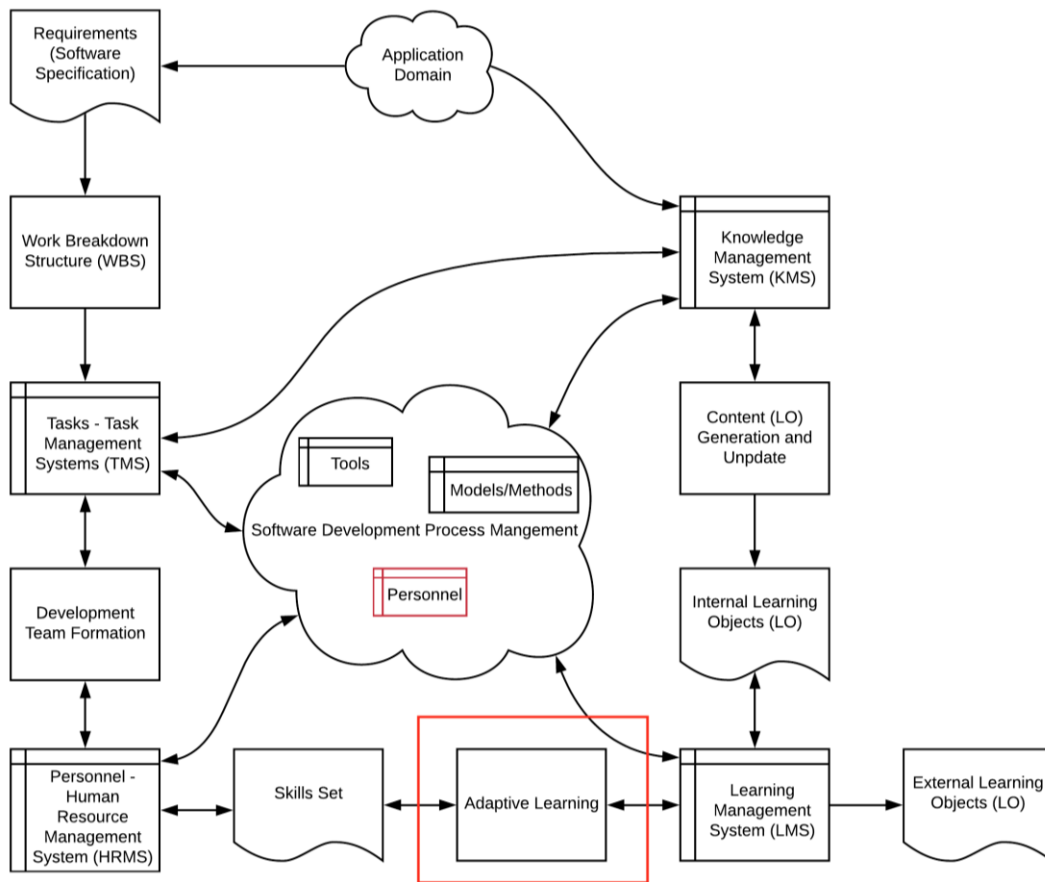


Figure 1: Framework schema [6]

This paper proposes a solution for a part of a corporate software development process management framework (see fig. 1). An adaptive learning algorithm takes employee skills set from human resource management systems (HRMS) and learning objects from corporate LMS/KMS or external sources as input. It produces a learning path for a particular employee to fit needs for a particular project role or profile. As a learning path we understand set of learning materials and courses, which are adapted for specific employee. Adaption includes omitting those parts of learning materials and courses, which are already known for employee.

This paper is structured as following: section two gives a brief background on related research and section three describes a proposed solution.

2. Related research and publications

2.1. Recommendation systems in adaptive learning

Recommendation systems is a well-known approach for suggesting items in different subject areas such as movies [7], products [8] and academic collaborations [9].

We may distinguish several major recommendation techniques (see fig. 2).

Collaborative filtering is one of the most widely used techniques, which is based on customer's ranking data and purchase or activity history and demographic profile. It suggests relevant items based on other customers with similar interests.

Collaborative filtering systems can be classified into two main categories: memory-based and model-based. Model-based recommender systems use a group of ratings as an input to different

statistical and machine learning algorithms. Memory-based algorithms make rating predictions using a whole dataset of ranked items. Both of them utilize the concept of similarity between users and/or items. Generally speaking, they construct a neighbourhood of similar users or items using the concept of similarity between users and/or items based on search history, user ratings and implicit feedback etc. Those systems offer flexibility and real-time customization with several drawbacks like item coverage, cold-start problem, new item or user problem and data sparsity [10].

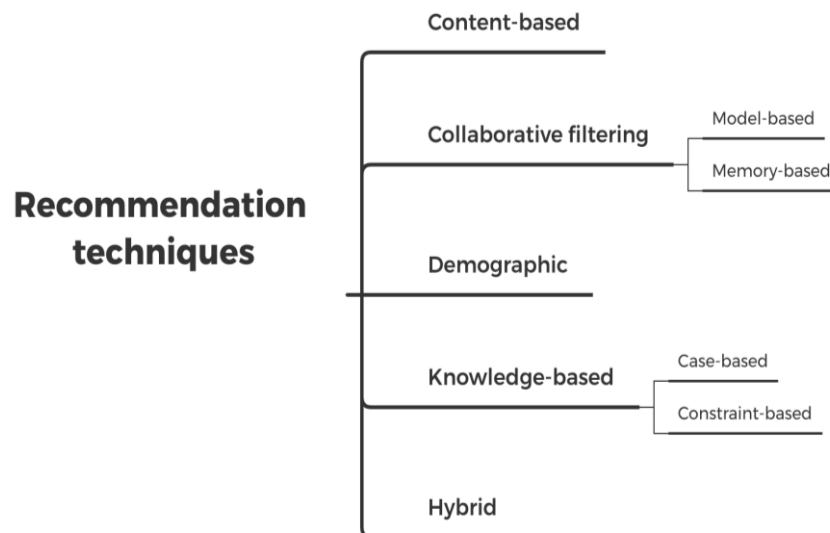


Figure 2: Common recommendation technique

Content-based systems use extensive modeling of user interests and content features. For example, such a recommender system can use customer’s rating and product database to offer similar items. Content-based filtering is mostly applied to text-based documents that are difficult for computers to interpret [11]. However, it has several drawbacks. It does not evaluate product quality, style and other subjective features based on collaborative feedback. Also it is not able to generate accident finds - users are limited to items, which are similar to previously rated ones. Despite benefits like minimization of cold-start and data sparsity problems it suffers from overspecialization, new user problems and inability to evaluate item quality [12].

Knowledge-based systems use customer’s queries and domain knowledge to generate recommendations. There are two main types of such systems. Case-based systems tries to find previously solved cases or scenarios which are similar to the customer's one. Similarity assessment usually relies on domain knowledge. Constraint-based systems use explicitly defined constraints like filters, query etc. When no similar item is found, both knowledge-based approaches exploit mechanisms that enable determination of minimal set of changes to the given set of customer requirements such that a solution can be found [13]. The major drawback is the need for well-defined business rules and domain model [14].

Demographic recommenders use demographic characteristics of users to suggest items [14].

Hybrid approaches combine two or more approaches to solve the drawbacks of different recommenders. For example, combining content and collaborative filtering may solve the cold-start and data sparsity problems. Obviously, such systems have increased implementation and analysis complexity [14].

Besides the usage in different forms of ecommerce applications, recommender systems may be used to enhance academic or corporate learning by adapting study materials to the needs of a particular user. It is called adaptive learning, which we cover later in this paper. Recommendation systems may offer desired levels of customization for adaptive learning systems. Usually, adaptive learning utilizes such recommendation techniques as knowledge-based [15] or hybrid [16], since e-learning needs advanced domain and knowledge modeling.

2.2. Adaptive learning

The term “adaptive” is associated with a range of diverse system characteristics and capabilities in the e-learning industry. There are four types of adaptation in the e-learning systems:

- adaptive interaction;
- adaptive course delivery;
- content discovery and assembly;
- adaptive collaboration support.

Adaptive interaction takes place in the user interface in order to enhance the user experience without modifications in learning content. For example, modification of color schemes, font sizes and various accessibility features [17].

Adaptive course delivery is one of the most popular categories of adaptive learning environments. It optimizes the fit between user and courses. Namely, it includes dynamic course restructuring, adaptive navigation support and enhanced selection of alternative course material [18].

Content discovery and assembly aggregates learning content from different sources according to user needs [7].

Adaptive collaboration support is intended to capture adaptive support in learning processes that involve communication between multiple persons (and, therefore, social interaction), and, potentially, collaboration towards common objectives [7].

Usually, adaptive learning system architecture consists of several well-known models:

- domain model represents information about offered courses like common workflows, relations between course elements, etc. [18];
- learner model refers to the information about system users [17];
- group model describes characteristics and similarities between user groups;
- adaptation model defines what and how has to be adapted [17].

Recommender systems described previously may provide robust solution for several adaptation aspects, such as content discovery and assembly, adaptive course delivery [19].

However, adaptive learning in a corporate environment differs from academic application. Those specifics will be covered.

2.3. Corporate e-Learning for IT professionals

Corporate e-learning approaches has several major differences from academic ones, such as:

- goals;
- learner and domain models;
- data sources.

Corporate learning aims to train professionals to fit particular role in a project. Each project requires specific skills and knowledge, which and may rapidly change. That leads to the second major difference – they have different domain and learner models. Speaking of IT, domain and learning model should:

- define IT-specific competences, roles, skills and knowledges;
- express competence level of IT professionals.

Fortunately, that issues are already solved by competence frameworks. One of them is European e-Competence framework. The European e-Competence Framework is based on competences. Its purpose is to provide general and comprehensive e-Competences specified at five proficiency levels that can then be adapted and customized into different contexts from IT business and stakeholder application perspectives. It consists of four dimensions [20].

1. 5 e-Competence areas, derived from the ICT business processes (plan, build, run, enable, manage);
2. Set of reference e-Competences for each area, with a generic description for each competence. 41 competencies identified in total to provide the European generic reference definitions of the framework.
3. Proficiency levels of each e-Competence provide European reference level specifications on e-Competence levels from e-1 to e-5.

4. Samples of knowledge and skills relate to e-Competences in dimension (i.e., Java Programming).

According to case studies, eCF was successfully used for following cases:

- internal ICT staff development in large ICT demand organizations [21] (e.g., EDF, Euro Disney and Pôle Emploi [22]);
- training development in a corporate/ICT supplier environment [21] (KPN [22]);
- competence based e-curriculum [21].

This competence framework can be easily used as a base for an adaptive learning algorithm which will be covered later.

Third difference is in the source of data. For instance, learner data may be obtained from specialized HRMS. Learning materials may be extracted from internal KMS or LMS and from massive open online courses like Coursera, Udemy or PluralSight as well.

Next section combines information on adaptive learning, recommender systems to propose a solution of an adaptive learning problem in the corporate environment.

3. Proposed solution

3.1. Glossary and model

As we discussed before, adaptive learning for IT professionals needs IT-specific domain and learner models. Below you can find a model based on eCF [20].

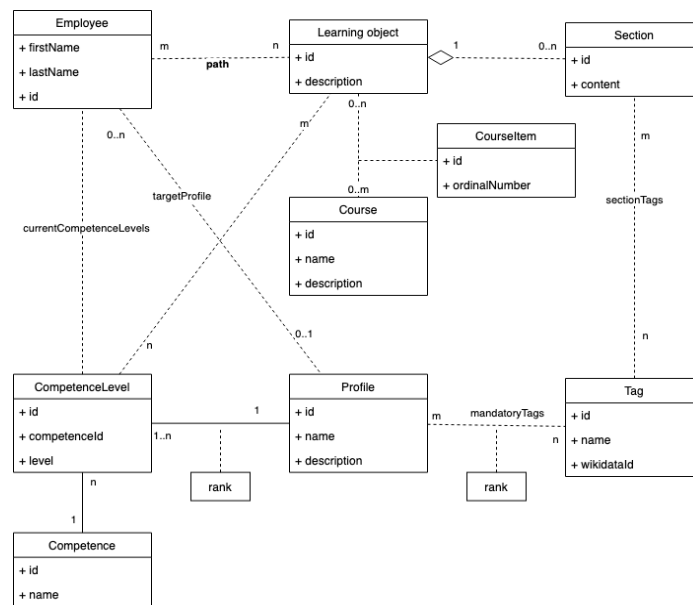


Figure 3: Data model

One of the central entities in the model is a “Competence” and its level (“CompetenceLevel”). We need them to define requirements for a specific Profile. For example, “Middle Java developer” profile is derived from the “Developer” role in e-CF. It requires a third level of “Application development” e-CF competence.

Competence levels are ranked by their importance for a particular profile starting from 1. For example, “Testing” competence is ranked second in the “Middle Java Developer” profile because it is less important than “Application development”. Moreover, it should be stated that competence level is expressed as a number from 1 to 5. All possible values are listed in the e-CF document [20].

“Tag” entity describes a specific knowledge. For example, “Middle Java developer” should have specific knowledge of “SQL” or “Gradle”. “Tags” may also have their importance in profile if needed.

Learning object (LO) represents any learning resource which may come from sources like KMS/LMS or external sources like MOOCs or open web pages. Learning objects are described by competence levels which they cover and optional tags. For example, objects like “Building Spring Boot

application with Kotlin” may have “Application development” level 2 and tagged with “Spring Boot” and “Kotlin” tags. Learning objects may belong to a course - a special sequence of LOs, which should be processed only one after another.

Employees in this model are described with personal data, current competence levels and knowledge (tags) and target profile.

Next section proposes an algorithm which constructs effective “learning path” using existing learning objects in order to reduce employee training time.

3.2. Algorithm description

The algorithm aims to reduce the learning time of employee training by selecting effective path and existing learning objects. Algorithm needs following data as input:

- static data (competences, their levels, tags);
- dynamic data (target profile, current employee competences and tags, learning objects).

An algorithm consists of several steps. Namely:

1. Selection of target profile;
2. Minimal difference calculation;
3. Learning path construction.

Algorithm constructs a path of learning objects to achieve needed levels of competence.

Minimal difference calculation is one of the main concepts in the algorithm. The idea is to compute the difference between target position competence levels and current employee competence.

Let competence level be denoted as:

$$CL = \langle Competence, level \mid level \in \mathbb{N}, level \in [1,5] \rangle, \quad (1)$$

where CL_{all} - universal set of competence levels (see CompetenceLevel in Figure 3),

CL_{target} - set of competence levels from e-CF profile (see relation between Profile and CompetenceLevel in Figure 3).

Required competence levels set(CLs) contain all CLs which belong to target competences, but their level is less or equal comparing to target one:

$$CL_{total} = \{CL \in CL_{all} \mid level \leq CL_{target}\}, \quad (2)$$

$$L_{diff} = \{CL \in CL_{all} \mid level \in (level_{employee}, level_{target})\}. \quad (3)$$

For example, if an employee has level 1 of “Application development”, but needs level 3, the difference will contain “Application development” levels 2 and 3.

We need to calculate match between employee e and profile p :

$$match_{e,p} = 1 - \frac{|CL_{diff}|}{|CL_{total}|} \quad (4)$$

Minimal difference is constructed with heuristic match threshold θ . E.g., a point, when an employee is “Good enough” for a particular position. We select a minimal number of competences (CL_{min}) by the following algorithm:

1. Sort CL_{diff} by profile rank (descending) and competence level (ascending). In other words, the most needed and low-level competences will be put in the first place.
2. Calculate competences level amount:

$$Cn = \text{ceil}(\theta - match_{e,p}) * |CL_{diff}| \quad (5)$$

3. Take n items from CL_{diff}

For example, if employee needs following competences:

- “Application development” level 1;
- “Application development” level 2;
- “Application development” level 3;
- “Testing” level 2/

And already have “Application development” level 1, then CL_{total} will contain 4 items, but CL_{diff} will contain only 3 items and employee matches profile with following coefficient:

$$match_{e,p} = 1 - \frac{3}{4} = 0.25. \quad (6)$$

If $\theta = 0.75$ then we can take only

$$n = \text{ceil}(0.75 - 0.25) * 3 = 3(\text{competence levels}). \quad (7)$$

For the above-mentioned example, we take only Application development levels 1-3.

Learning path construction is a cyclic process which consists of several stages depicted below. Accessibility, sort criterions, achievements and obtained knowledge concepts are described below.

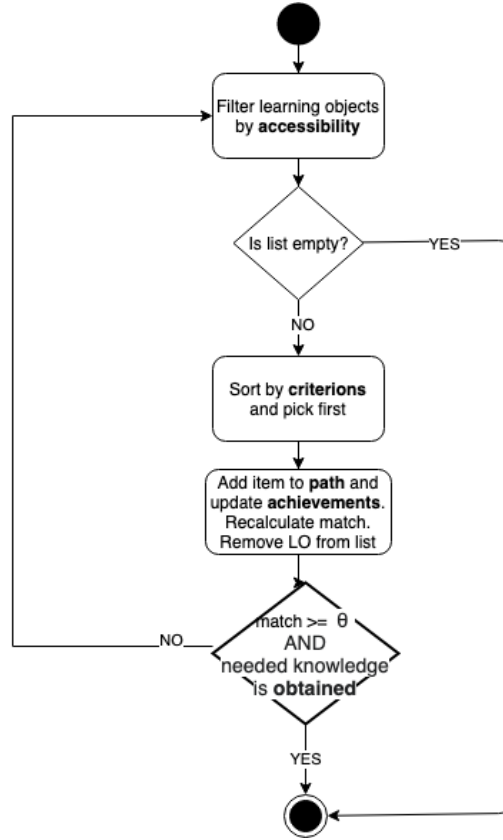


Figure 4: Learning path construction algorithm

Let CL_{LO} be a set of competence levels of a particular learning object.

And $CL_{current}$ be a set of current competence levels of a particular employee.

Then learning object is accessible if and only if:

$$\forall CL \in CL_{LO}: \text{level}_{LO} \leq \text{level}_{current} + 1. \quad (8)$$

After accessibility filtering, learning objects are being sorted by several **criterions** with following priority:

1. Amount of $CL \in CL_{diff}$, descending;
2. Sum of needed competence profile ranks, descending;
3. Number of needed tags for profile, descending;
4. Time to learn, ascending.

Achievements are a set of tags known by employees and achieved competence levels.

Algorithm has following stop conditions:

- there are no accessible objects (exception case);
- employee is good enough for target profile ($match \geq \theta$) and employee achievements contain all needed tags.

As a result, we should get a sequence of learning objects which can be used as a learning plan for an employee to fit to a particular profile.

3.3. Testing results

Algorithm testing was performed using developed software which has following features:

- create, read, update and delete operations for all entities listed in Section 4.1

- obtain learning path recommendation for a particular employee

Introduction to Software Testing

This course is primarily aimed at those learners interested in any of the following roles: Software Engineer, Software Engineer in Test, Test Automation Engineer, DevOps Engineer, Software Developer, Programmer, Computer Enthusiast.

Competences

- User Experience level 2
- Testing level 1
- User Support level 1

Content

- Testing Foundations, time: 300 mins, Software Testing Programming
- Testing in the Software Development Lifecycle, time: 480 mins, Software Testing Programming
- Writing Good Unit Tests, time: 600 mins, Software Testing Programming
- Introduction, time: 420 mins, Software Testing Programming

Figure 5: Learning object UI

Testing was performed using an artificial dataset gathered manually from different sources like Coursera [23] and Stepik [24] which consists of 109 learning objects. Algorithm input data is presented below.

Table 1

Job profile

Characteristic	Value
Name	Middle Java developer
Mandatory skills and knowledges (tags)	Java, Kotlin, Spring Framework
Competence levels, by priority	1. Application Development level 3 2. Testing level 2 3. Component Integration level 2 4. Documentation Production level 3 5. Problem Management level 3
“Good enough” threshold (θ), %	80%

Target profile competence levels correspond to e-CF profile of Developer. Mandatory tags like Java or Kotlin are used to add better precision for recommendation. For testing purposes, an employee has zero knowledge or skills.

Employee will match profile with 82% accuracy

Target competences

- Testing 2
- Component Integration 2
- Application Development 3
- Documentation Production 3
- Problem Management 3

Achieved competences

- Application Development 3
- Application Design 1
- User Experience 2
- Testing 2
- User Support 2
- Component Integration 2
- Documentation Production 3

Learning will take

88 hours

Figure 6: Recommendation characteristics UI

The algorithm recommends 11 learning objects with competences shown on Figure 6, at the “Achieved competences” section. Database contains 39 matching learning objects with target competences. It means that without suggestions, an employee has to study all 39 learning objects instead of 11 recommended. We can see that the algorithm reduced learning objects by 72%. It is necessary to note that employee meets target profile requirements only by 80% (see Table 1, “Good enough threshold”).

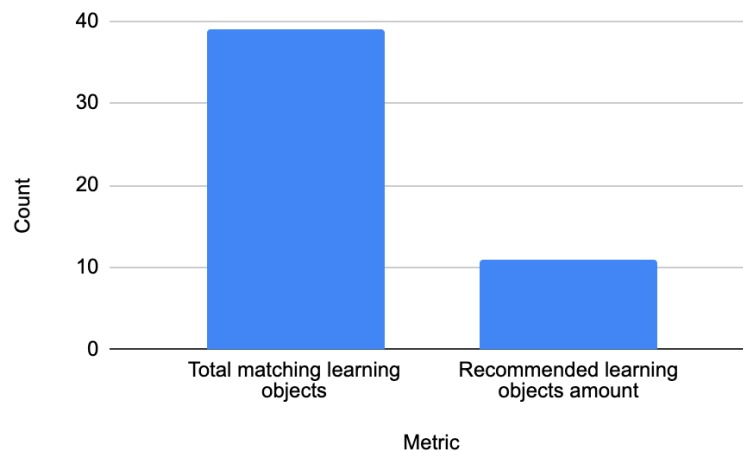


Figure 7: Total matching learning objects vs recommended learning objects amount (lower is better)

Speaking of time, recommended path duration is ~89.7% less than total matching learning objects study duration.

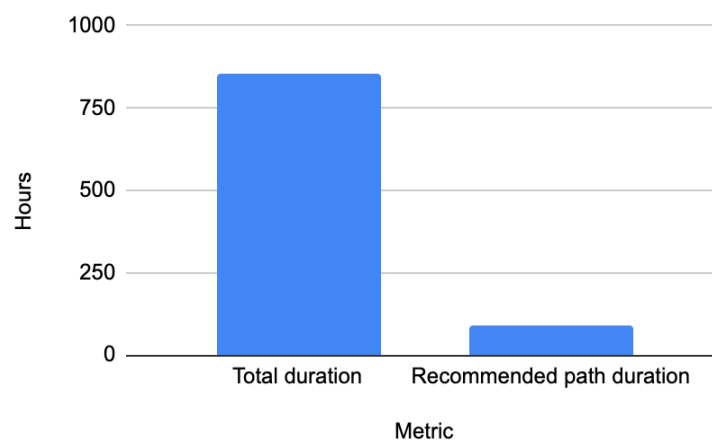


Figure 8: Total matching object duration vs recommended path duration (lower is better)

We can clearly see that the algorithm significantly reduces the amount of training time for employees. Unfortunately, current iteration of algorithm development has several assumptions and limitations such as:

- it is assumed that one competence object may cover one or more competence levels;
- need of extensive data preparation (e.g., marking learning objects with tags and competence levels);
- inability to automatically assess learning object quality;
- inability to adapt to employee’s learning style;
- heuristic “good enough level” assumption.

Nevertheless, it solves the problem of adaptive course delivery and content discovery in the field of corporate e-learning by presenting an individual study plan for a particular employee.

4. Conclusion and future work

This work proposed a way to reduce employee training time to fit a particular job profile by using the adaptive learning algorithm based on European e-Competence framework and knowledge-based recommender system. It may be used as an Adaptive learning unit in the existing corporate software development process management framework [6].

However, there are several points for improvement. First, we need to improve the dataset and perform testing on real data. Depending on the results testing, we can choose either to automate metadata extraction to reduce the amount of manual work or to improve adaptation quality. Adaptation quality may be increased using ontological similarity between learning objects or assessing quality of learning objects with other types of recommender systems (i.e. to introduce collaborative filtering).

5. References

- [1] The State of Developer Ecosystem 2020, 2020. URL: <https://www.jetbrains.com/lp/devecosystem-2020/>
- [2] The State of Developer Ecosystem 2020: Javascript, 2020 URL: <https://www.jetbrains.com/lp/devecosystem-2020/javascript/>
- [3] M. Laal, P. Salamati, Lifelong learning; why do we need it?, *Procedia-Social and Behavioral Sciences* 31 (2012) 399–403.
- [4] М. В. Ткачук, В. Є. Сокол, М. О. Білова, О. С. Космачов, Класифікація, типова функціональність та особливості застосування систем електронного навчання та тренінгу персоналу в ІТ-компаніях, *Сучасні інформаційні системи*, 2 (2018) 87–95.
- [5] V. E. Sokol, M. V. Tkachuk, Y. M. Vasetska, Adaptive training system for IT-companies personnel: design principals, architectural models and implementation technology, *Bulletin of the National Technical University “KhPI”: a collection of scientific papers. Thematic issue: System analysis, management and information technology*, 51 (2017) 38–43.
- [6] V. Sokol, M. Bilova, A. Kharin, An Approach for Creating Learning Content from Knowledge Management System, *Proceedings of Integrated Computer Technologies in Mechanical Engineering 2020 Conference, Kharkiv, Ukraine, 2020*, pp. 692–703.
- [7] D. Wiley, Learning objects: Difficulties and opportunities, *Academic ADL Co-Lab News Report* 152-030406, 2003.
- [8] T. Zhou, L. Chen, J. Shen, Movie recommendation system employing the user-based cf in cloud computing, in: *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, 2017.
- [9] H. Hwangbo, Y. S. Kim, K. J. Cha, Recommendation system development for fashion retail e-commerce, *Electronic Commerce Research and Applications* 28 (2018) 94–101.
- [10] Y. Liu, Ch. Yang, J. Ma, W. Xu, Zh. Hua, A social recommendation system for academic collaboration in undergraduate research, *Expert Systems* 36 (2019).
- [11] F. Alyar, N. J. Navimipour, Recommender systems: A systematic review of the state of the art literature and suggestions for future research, *Kybernetes* 47 (2018).
- [12] M. Komkhao, J. Lu, Z. Li, W. A. Halang, Incremental collaborative filtering based on mahalanobis distance and fuzzy membership for recommender systems, *International Journal of General Systems*, 42 (2013) 41–66.
- [13] M. Montaner, B. Lopez, J.L. De La Rosa, A taxonomy of recommender agents on the internet, *Artificial Intelligence Review*, 19 (2003) 285–330.
- [14] A. Felfernig, R. Burke, Constraint-based recommender systems: technologies and research issues, in: *Proceedings of the 10th international conference on Electronic commerce*, 2008.
- [15] J. K. Tarus, N. Zhendong, M. Ghulam, Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning, *Artificial intelligence review* 50 (2018) 21–48.
- [16] J. K. Tarus, N. Zhendong, Y. Abdallah, A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining, *Future Generation Computer Systems* 72 (2017) 37–48.

- [17] A. Paramythis, S. Loidl-Reisinger, Adaptive learning environments and e-learning standards, in: Proceedings of Second European conference on e-learning, 2003.
- [18] P. Brusilovsky, Adaptive hypermedia, *User Modeling and User Adapted Interaction* 11 (2001) 87–110.
- [19] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, R. Koper Recommender systems in technology enhanced learning. In: P. B. Kantor, F. Ricci, L. Rokach & B. Shapira (Eds.), *Recommender systems handbook*. Springer, Boston, MA, 2011. pp. 387–415.
- [20] European competence framework overview, version 3.0, 2021 URL: <https://www.ecompetences.eu/e-cf-overview/>
- [21] Case studies for the application of the European e-Competence Framework 3.0. CWA 16234:2014, 2021 http://www.ecompetences.eu/wp-content/uploads/2014/02/Case_studies_e-CF_3.0_CEN_CWA_16234-4_2014.pdf
- [22] Current e-CF users, 2021 URL: <https://www.ecompetences.eu/it/current-e-cf-users/>
- [23] Coursera home page, 2021 URL: <https://www.coursera.org/>
- [24] Stepik catalog page, 2021, URL: <https://stepik.org/>