DOI: 10.14529/ctcr170301

# APPROACH TO EDUCATIONAL COURSE COMPARISON USING NATURAL LANGUAGE PROCESSING TECHNIQUES

***D.S. Botov**, dmbotov@gmail.com,*
***Yu.D. Klenin**, jklen@yandex.ru*
*Chelyabinsk State University, Chelyabinsk, Russian Federation*

As numbers of educational programmes and courses grow, the need for a method of comparison becomes apparent. In this paper we discuss the overall state of education data mining, the variety of document types and formats used for educational content and propose the combined similarity measure for educational course programmes, Our proposed similarity measure uses three most important in our opinion elements of course programmes – course descriptions, educational results of the course and the structure of the educational course. We describe our approach to calculate similarity for each of this component pairs as well as provide primary experimental results and their evaluation using mean average precision metric.

*Keywords: educational data mining, educational course programme, syllabus, data analysis, similarity measure.*

## Introduction

Recent increase in the amount of available educational content, such as educational programmes, course programmes or course syllabuses, results in the increase of the amount of work required from educational organizations personnel in order to process such content.

There are two main reasons to process those documents. Firstly, educators, who strive to survive in the competitive environment of higher education market are focusing on quality improvement and actualization of their teaching approaches, which requires them to analyze various sources, including content documenting approaches suggested by their colleagues and competitors.

Second reason for educational content mining is knowledge extraction. This reason is of scientific interest, as educational content provides structure for the knowledge itself, allowing potential improvement to the quality of information that we handle every day.

To this extent we attempt to introduce an approach to comparison of course programme documents, which is the legally defined format for course specification in Russian educational organizations.

## 1. Related research

The part of data mining concerned with the field of analyzing educational content is called educational data mining (EDM).

In a more general sense this field includes not only analysis of the educational documents, but also various data mining tasks involving student modeling, behavior analysis, question-answering and evaluation. However, we will focus primarily on educational content analysis.

A group of researchers [1] suggested using Naïve Bayes classifier to assign e-learning materials to the leaf nodes of the given concept hierarchy. Another similar research [2] focuses on classifying examination questions according to the existing hierarchy of concepts in order to evaluate what kind of skill or knowledge the question checks.

One of the popular formats, used in educational materials to describe the results, achieved by students is learning outcomes. Learning outcomes usually consist of the action verb and a terminological part. Action verbs are restricted to the educational taxonomies and a researcher [3] suggested taxonomy-based similarity measure for the task of comparison of two action verbs, named after the researcher – Chernikova's action verb similarity measure. We will discuss learning outcomes in more detail in the section 2.5.

A popular task in educational data mining is knowledge structure extraction, for instance, into ontology, which would describe the structure and relationships between didactical units within educational courses and programmes. For example, a pair of researchers [4] presented a set of ontological models for both the curriculum and syllabus, while also suggesting a general approach to mapping any real syllabus to the elements of ontology of underlying knowledge. In another work [5] an approach for ontology-building for didactical activity model with three phases (teaching-learning-examination) was described. Said algorithm results in a variety of ontologies: Course Basic Subject Ontology, Course Practical Activities Ontology, Basic Examination Ontology and so on.

## 2. Educational and course programme structure

### *2.1. Previous work*

As mentioned above, one of the reasons for exploring the possibility of automatisation of educational content analysis is the overall quality improvement of higher education through simplification of actualization problem. Previously authors have suggested an overall structure for the system, which would be capable of providing informational support for educators in this task [6]. One of the main goals for such a system would be not only simplification of the improvement process itself, but focus on the required result of higher education – professional skills and qualities of a student. This could be achieved through introduction of the requirements specified in professional standards and actual job descriptions given by real employers and analysis of the correlation between these requirements and the educational results declared in the educational content documents themselves.

### *2.2. Programme structure models*

It is important to understand, that educational content varies greatly. Not only representation of knowledge varies between different kinds of educational documents, but also within the same document type depending on what organization, departments or even specific author. To understand the basic average structure and contents of educational document, we analyzed various document formats, focusing on educational and course programmes and syllabuses from universities and MOOC organizations all around the world.

For educational organizations, operating under Russian jurisdiction, there exists a set of rules, legally imposed by the government, primarily by educational standards and specifications released by the Russian Ministry of Education and Science. In order to model the overall structure of educational programme (Fig. 1), we have previously suggested the following semantic network model, which consists of the following concepts: educational programme, course (describes specific subject or module), topic (a specific subsection of a course) and learning outcome (represents results).

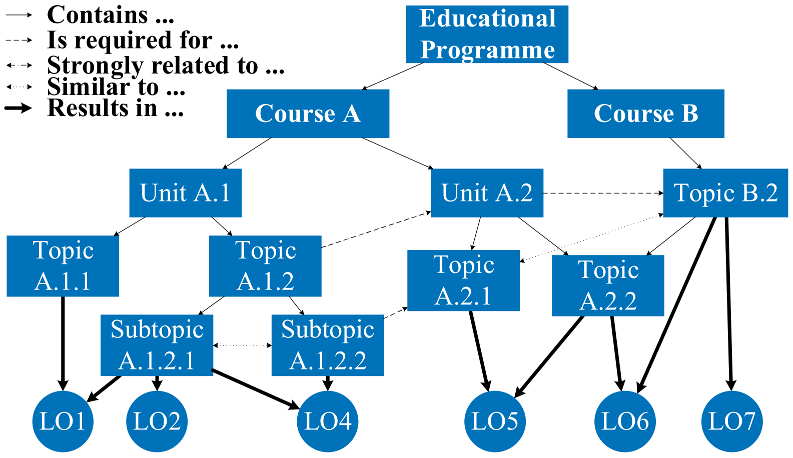

**Fig. 1. Knowledge structure of educational and course programmes**

That scheme represents the structure of the knowledge, covered by the programme. The standardized format of educational and course programme description documents, on the other hand, could be better captured by using the following model (Fig. 2).

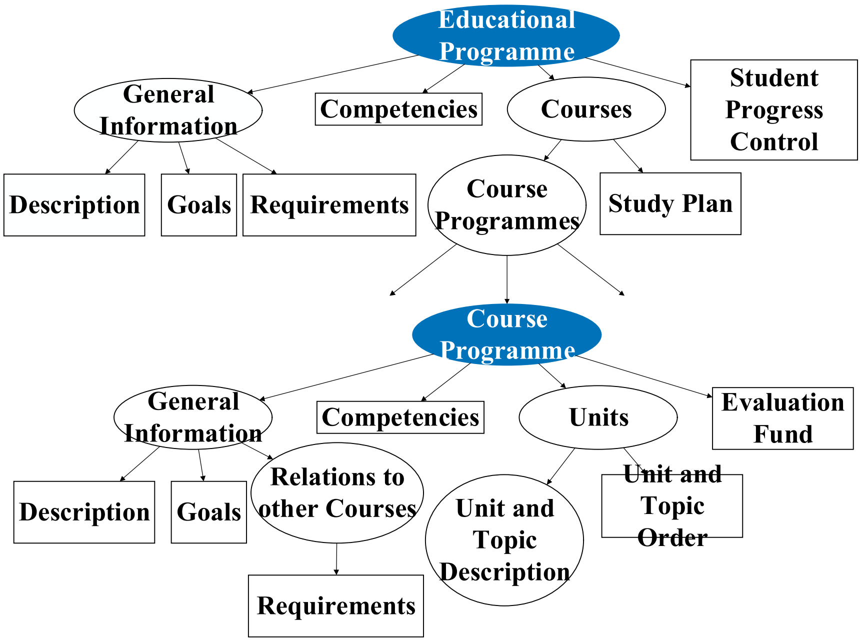

**Fig. 2. Document structure of educational and course programmes**

Our research showed that while Russian universities are legally restricted to a more or less well defined format, most western organizations do not have such strict regulations regarding their documentation. However, even among them certain trend could be noticed, with most formats containing three main components:

- generic description of the course or educational programme;
- overall structure of course or programme and component description;
- list of results, usually in a format similar to learning outcomes.

In our understanding, these components are the most important to the contents of the document and that's why we focus on them as means of distinguishing and comparing different courses. These components are discussed in more detail below.

### *2.3. Programme description*

Overwhelming majority of course or educational programme specifications include a somewhat brief description of a course or an educational programme. The main purpose of it is to quickly explain what the course will be covering and why should the student be interested in it.

As for the standard, used by Russian educational organizations, there are no specific partition called "description" but it function is carried out by the sections named "Course Goals" and "Place of the discipline in the educational programme structure". The first one describes the general results, expected of a student after completion of the course, such as broad descriptions of skills and qualities acquired, experience gained and so on. The second section focuses more on the relations between described course and its neighboring disciplines, specifying which courses should precede this one and where the acquired skills will be applied in the future.

From the data mining perspective, the description is usually represented by a plain text with no inner structure.

### *2.4. Programme structure*

Representations of this component vary greatly, from simple lists of broad topics, touch upon during the course, to exact time tables specifying the topic and concepts covered by every lesson.

For Russian universities, which we are currently focusing on, the format prescribes several tables, specifying what specific units within the course (or courses within educational programme) would be covered and in what order, how much time each will take, what lectures will be given, what specific didactical would be covered within each, what, if any, topics are delegated to the students to study on their own, what works will be carried out in laboratory and in practical classes. Multilevel lists of covered concepts are quite popular as well.

The text data within the structure of the document is usually represented as either plain text, or, even more commonly, as a set of concepts in a form of keywords (or key phrases), listed one after another.

### *2.5. Educational results*

Most course specifications and educational programmes include the specific list of expected results, which student should be able to exhibit and be tested for after successful completion of the course.

While there exists a broad range of possible representations of this data, most western organizations stick to using learning outcomes for this purpose. Learning outcome is a specific format, that, as mentioned above, consists of two main components – the action verb, describing what kind of knowledge or skill is required of a student, and terms, describing specific knowledge concepts, that action verb is applied to.

As we mentioned above, action verbs are coming from a relatively small set of words, listed in various educational taxonomies and can be easily compared, using Chernikova's taxonomical similarity measure. Terms, on the other hand, are only limited to the knowledge domain, covered by the course and are represented by plain text, describing one or more keywords (or key phrases) and their relations to the verb and between themselves.

Russian designated format for educational results are competencies – a more broad and general description of one's knowledge, similar, however, to learning outcome.

## 3. Suggested approach for course analysis

In order to analyze and compare two course specification documents, one has to perform a pairwise analysis of the main components of these documents.

### *3.1. Keyword extraction*

While there exists many algorithms for keyword extraction, we have decided to use a modified RAKE algorithm for our purposes, because it is simple, easy to understand, implement and use and doesn't require a training set.

RAKE [7] stands for Rapid Automatic Keyword Extraction and, as the name implies, the algorithm itself presents a fast way to extract key phrases from plain text: the algorithm separates text into a list of candidates – sequences of contiguous words divided by various phrase delimiters (mainly punctuation, or any other symbol that in the current context can serve as such) and by any of the specified stopwords. The algorithm is based on the observation that keywords and key phrases rarely contain punctuation or stopwords. The actual key phrases are then weighted based on the scores of the words inside of them. These scores in turn based on word frequency and on word's degree – the total sum of lengths of all the candidate phrases containing the word:

$$Score(Keyword) = \sum_{\substack{word_i \\ \epsilon\, Keyword}} \frac{\deg(word_i)}{freq(word_i)}.$$

Our modifications to the algorithm itself solve the problem of original algorithm not dealing with declensions of words. While it has little impact in languages such as English, where nouns and adjectives rarely display differences of genders or cases, it is much more important for Slavic languages, such as Russian. Our solution simply makes sure to transform the word to its lemma before starting frequency analysis.

As we extract the keywords from the text we can set a level of minimal score required to count as a keyword, based, for instance, on a maximum score.

### *3.2. Programme description analysis*

As programme description is usually given in a form of plain text, the task of analyzing it can be considered a broad text similarity problem. While there exists large amount of various approaches to this task, at this point of the experiment, we are suggesting a simplistic one: extracting key phrases from the description and performing comparison only on subsets of key terms used. Not only this approach is simple, but we also consider terminology, coming from the specific knowledge domain, to be one of the most impactful factors for the contents of a course description, proposing, that comparison of such terminology, should suffice in this task.

Comparison of two sets of keywords can also be done in a variety of ways, but currently we are going to use a word2vec model [8], trained on Russian Wikipedia corpora to calculate the word vectors for extracted keywords, combine them into a single vector, holding entire description of the document and comparing those, using cosine similarity. This approach was suggested by the authors of genism framework [9].

It introduces the following definitions:

$$\overrightarrow{L2_w} = \frac{\overrightarrow{V_w}}{|\overrightarrow{V_w}|}$$

is a l2-normalized word2vec vector of the word $w$.

$$\overrightarrow{m_{ws}} = \frac{\sum_{w_i \in ws} \overrightarrow{L2_{w_i}}}{|ws|}$$

is a mean of l2-normalized word2vec vectors of all the words in the wordset $ws$.

The similarity between two wordsets $ws_1$ and $ws_2$ is a cosine between the means of the words in those wordsets:

$$sim_{wordsets}(ws_1, ws_2) = cos\ (\overrightarrow{m_{ws_1}}, \overrightarrow{m_{ws_2}}).$$

Thus the similarity between two documents $d_i$ and $d_j$ can be calculated as wordset similarity for the sets of keywords in these documents:

$$sim(d_i, d_j) = sim_{wordsets}(Keywords(d_i), Keywords(d_j)).$$

### *3.3. Course structure analysis*

While course structure generally presented as a structured text, we consider appropriate at this stage of our research to not involve the structure into analysis and just utilize the textual contents of it.

To this end we view the course structure description as the plain text, similar to the course description. Thus we employ the same approach to extract key phrases using modified RAKE algorithm and, after combining the resulting words into a flat set of words, compare those between two different courses.

### *3.4. Educational Results Analysis*

As the main format of educational results we are currently dealing with is a competency, we focus on transforming them into more manageable representations, reminiscent of learning outcomes (Fig. 3).

| | Remember 1 | Understand 2 | Apply 3 | Analyze 4 | Evaluate 5 | Create 6 |
|---|---|---|---|---|---|---|
| 1 | Recognize | | | | | |
| 2 | Recall | | | | | |
| 3 | | Interpret | Execute | | | |
| 4 | | Exemplify | | Differentiate | Check | |
| 5 | | Classify | | Organize | | |
| 6 | | Summarize | | Attribute | | |
| 7 | | Infer | | | | |
| 8 | | Compare | | | | |
| 9 | | Explain | Implement | | | |

**Fig. 3. Chernikova's action verb taxonomical similarity measure**

Since verbs in competencies are not actually restricted to a taxonomy, we use an approximate approach when comparing them. To do this we are going to map the verbs, used in competencies onto the Chernikova's taxonomical similarity measure. This can be done by using trained word2vec model to find for each of the compared verbs the set of the most similar action verbs in the taxonomy and average the resulting similarity. In our case we use a simple mean average of pairwise Chernikova's similarities:

$$sim_V(V_i, V_j) = sim_V\left(\left[AV_q\right]_i, \left[AV_p\right]_j\right) = \frac{\sum sim_{CAVe}(AV_q, AV_p)}{\left|\left[AV_q\right]_i\right| * \left|\left[AV_p\right]_j\right|}.$$

Where $sim_{CAVe}(AV_q, AV_p)$ is Chernikova's similarity and is calculated in a following manner:

$$sim_{CAVe}(AV_i, AV_j) = \frac{max\left(d_{EucKrth}(AV_i, AV_j)\right) - cur\left(d_{EucKrth}(AV_i, AV_j)\right)}{max\left(d_{EucKrth}(AV_i, AV_j)\right)}.$$

Here "$cur\left(d_{EucKrth}(AV_i, AV_j)\right)$ represents the distance between the subcategories, to which the action verbs $AV_i$ and $AV_j$ refer. The $max\left(d_{EucKrth}(AV_i, AV_j)\right)$ stands for the longest possible distance in the model".

The terms, used in competencies, at this stage can be considered plain text and compared as such, using wordset similarity measures, described above.

In the end similarity between two results can be calculated as a simple linear combination (or weighted voting) of the similarity scores of its components and overall similarity between two sets of results can be taken as mean average of pairwise comparisons between results:

$$sim_{res}(d_i, d_j) = sim_{res}([res_i], [res_j]) = \frac{\sum_{\substack{(v_p,t_p)\epsilon[res_i], \\ (v_q,t_q)\epsilon[res_j]}} (w_{verb} * sim_V(v_p, v_q) + w_{term} * sim_T(t_p, t_q))}{|[res_i]| * |[res_j]|}.$$

Here, $[res_i]$ and $[res_j]$ are sets of results for documents $d_i$.and $d_j$ respectively.

### *3.5. Combining results of partial comparison*

As we describe above, we can calculate three separate similarity scores for the parts of compared documents. While there are different approaches to combining similarity scores of separate parts into one, we consider that at this point in our research, the simplistic linear combination approach will suffice.

This means, that total similarity score between two compared documents will be calculated as follows:

$$sim(d_i, d_j) = w_{descr} * sim_{descr}(d_i, d_j) + w_{struct} * sim_{struct}(d_i, d_j) + w_{res} * sim_{res}(d_i, d_j),$$

where $w_x$ and $sim_x$ define the weight and the calculated similarity respectively for each individual part of two documents being compared.

Since each of the similarity score is a real number between 0 and 1, the weights should be picked in such a way, that the overall similarity of two documents still lied within those boundaries. Thus the weights need to be picked with this limitation in mind:

$$w_{descr} + w_{struct} + w_{res} = 1.$$

### *3.6. General structure of the algorithm*

The approach we are using can be described as follows:

1. Model preparation stage:

a) Language model preparation: We prepare a cleaned up version of Russian Wikipedia dump – removing metadata, transforming words to their lemmas etc. Using this as a corpora, we can train word2vec model;

b) Action verb similarity pre-calculation: Since we are using Chernikova's similarity measure to compare verbs in learning outcomes, we pre-calculate all of the possible pairwise similarities to speed up the comparison process later on.

2. Document preparation stage:

a) Input data preparation: Due to a variety of file formats and document formats used for course programmes, at this stage we have to manually extract relevant elements and prepare them in a unified, easily processable format;

b) Initial processing: As we load the documents into the experimental setup, we still need to run some preprocessing on them – transforming them to a uniform type format, remove non-standard symbols (for example, popular in Russian guillemets, also known as Latin quotation marks – "«»"), build a list of competencies and so on.

3. Description similarity analysis:

a) Keyword extraction: For each description, we perform additional cleanup on the fly – transforming words to their lemmas, feeding the result to the word2vec for even more discriminative cleanup – removal of all terms missing from the word2vec vocabulary. What is left can be fed to the RAKE algorithm, resulting in a ranked list of key phrases and their weights. As a last step we remove all the keywords below our selected minimal level of importance;

b) Comparing the sets of keywords: For each description pair, we now have a list of most important keywords, which all exist within word2vec vocabulary. These sets of keywords are then sent in for wordset similarity calculation.

4. Structure similarity analysis:

a) Keyword extraction: Similar to the descriptions, we use the same keyword extraction pipeline on structure of each course;

b) Comparing the sets of keywords: Again, just as with descriptions, we use wordset similarity measure to calculate the similarity between structures.

5. Educational results analysis:

a) Verb to action verb approximation: Unless the verb, used in the result is known to the system already, we employ word2vec with the task of calculating the subset of most similar action verbs from Chernikova's taxonomy, which we then cut based on selected minimal similarity level. This provides us with a list of action verbs, most similar to the verb, used in the result description;

b) Verb similarity based on action verb similarity measure: We compare two mappings onto the Chernikova's taxonomy and compile a list from pre-calculated similarities for each pair of action verbs. The mean of this list is taken as a similarity between two original verbs;

c) Term similarity: Similar to description and structure, albeit with much shorter text fragments, we calculate wordset similarity between two terminology components;

d) Competency similarity calculation: By using linear combination of the similarities between verb and term components we calculate the overall similarity between two results;

e) Comparing two sets of competencies: After calculating all the possible pairwise similarities between results from two sets, we take the mean of them as an overall similarity between two sets of results.

6. Combining all similarities and evaluating results:

a) Combining similarities: To combine three component's similarities together, we use linear combination with adjustable weights. The result is an overall value of similarity between the documents;

b) Evaluation: After calculating all the pairwise similarities between documents, we can use the results for each document to evaluate the quality of the algorithm. To do this we sort them by similarity and apply selected consideration barriers – minimal levels of similarity worth considering. After this we can use our knowledge of what discipline each document is supposed to be labeled and calculate Average Precision of our list. By doing so for each document and taking the mean average of Average Precisions for each, we get a single value which describes quality of our algorithm – Mean Average Precision.

## 4. Experiment

### *4.1. Setup and Results*

To try out suggested approach, we have selected a set of twenty course programmes for four different disciplines, produced in sixteen universities all over Russian Federation. For this dataset we performed calculations of similarity for all possible combinations of programmes.

At this stage we decided to take the following parameters:

- Our algorithm use 0.5 weights for both verb and term components of the learning outcome similarity;
- Our keyword extracting algorithm considers only top 75 % of the extracted keywords to be significant;
- Verbs in competencies are mapped to the top 10 % of the list of action verbs, ranked by similarity.

After calculating the similarity for each document pair, we ordered them by it, which resulted in twenty ranked lists.

| *Rank* | *Course Title* | *Specialization* | *Similarity* |
|---|---|---|---|
| 0 | Marketing | Economics | 0.96219 |
| 1 | Marketing | Veterinary-Sanitary Expertise | 0.92643 |
| 2 | Marketing | Management | 0.91959 |
| 3 | Marketing | Economics | 0.91834 |
| 4 | Marketing | National Economy Management | 0.90368 |
| 5 | Production Management | Management in Construction | 0.90211 |
| High-Consideration Barrier | | | |
| 6 | Production Management | Applied Informatics | 0.87522 |
| Consideration Barrier | | | |
| 7 | Production Management | Management | 0.85975 |
| 8 | Production Management | Management | 0.82717 |

**Fig. 4. Example output for one of the marketing course programmes**

As the general nature of all the documents is the same, there exists some similarity even between programmes for completely unrelated subjects. To combat that, we decide to select two barriers of document importance in each list. The lower barrier cuts off the lower 50 % of each list, based on the maximum and minimum similarities present, and represents our estimations of the value, at which subject similarity overtakes the general similarity between course programme documents. The second, more discriminate barrier – the high consideration barrier – cuts of the lower 75 % of each list, meaning only extremely close documents are left in the list.

As the result we are left with 20 ranked lists, each with two levels of importance in it. The example output is given in Fig. 4.

### *4.2. Evaluation*

Since our results are akin to the task of information retrieval, in order to evaluate the achieved results we decided to use the Mean Average Precision metric. This metric is primarily used to evaluate the results of search systems and is the mean average of Average Precision score for a set of queries:

$$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q}.$$

Here $Q$ is a set of queries and $AveP$ is the Average Precision score. This metric is calculated as:

$$AveP = \sum_{k=1}^{n} P(k) * \Delta r(k).$$

Here, $k$ is the rank in the sequence of retrieved documents, $n$ is the number of retrieved documents $P(k)$ is precision at cut-off $k$ in the list and $\Delta r(k)$ is the change in recall from items $k-1$ to $k$.

In our case, we treat list of ranked documents as results, returned for the query of the document they are compared to, which plays the role of the query, and their relevance is decided by whether or not their discipline tag matches the discipline tag of the “query” document. The value of the Mean Average Precision score is then calculated by averaging results across 20 of our document lists.

After performing basic evaluation and tuning the weights of the components in our combined similarity measure, we have achieved the following results (Fig. 5).

| *Weights of components* | *High Barrier* | *Low Barrier* |
|---|---|---|
| $w_{descr} = 0.125$<br>$w_{struct} = 0.75$<br>$w_{res} = 0.125$ | 0.8075 | 0.8268 |
| $w_{descr} = 0.175$<br>$w_{struct} = 0.65$<br>$w_{res} = 0.175$ | 0.8315 | 0.8027 |

**Fig. 5. Evaluation Results**

Results listed are the best performance of our method for lower consideration barrier and higher consideration barrier respectively.

## Conclusion

In this paper we have presented an overview of general structure of educational course programme and suggested an approach to comparison and ranking of such documents based on keyword extraction and word vector analysis. We also employed Chernikova's similarity measure for comparison of educational results. To evaluate this approach we have applied it to the task of ranking various documents against a query document and scored results using mean average precision metric. Scores show rather high quality of our approach, with MAP being above 0.8.

We plan to continue our work and in the future focus on improving the overall quality of the system:

- Experimenting with other methods of calculating learning outcome similarity:
  - Experimenting with different keyword extraction algorithms;
  - Improving cleanup algorithm to reduce the impact of the general non-subject-specific terminology;
  - Improve quality of the word2vec model;
  - Utilize the text structure of the course structure specification;
- Experimenting with a larger dataset;
- Replicating this experiment for other kinds of educational documents, primarily – education programmes.

## References

1. Saini P.S., Sona D., Veeramachaneni S., Ronchetti M. Making E-learning better through Machine Learning. *International Conference on Methods and Technologies for Learning*, 2005, pp. 1–6.

2. Foley J., Allan J. Retrieving Hierarchical Syllabus Items for Exam Question Analysis. *Advances in Information Retrieval*, March 2016, pp. 575–586.

3. Chernikova E., Nikolaev P. The Similarity Measure and Algorithm for Comparison of the Learning Outcomes. *Proceedings of the Fourth International Conference on Internet Technologies and Applications*, 2011, pp. 465–473.

4. Chung H., Kim J. An Ontological Approach for Ssemantic Modelling of Curriculum and Syllabus in Higher Education. *International Journal of Information and Education Technology*, 2016, vol. 6, no. 5, pp. 365–369. DOI: 10.7763/IJIET.2016.V6.715

5. Oprea M. On the Use of Educational Ontologies as Support Tools for Didactical Activities. *Proceedings of the International Conference on Virtual Learning(ICVL2012)*, Nov. 2012, pp. 67–73.

6. Botov D., Klenin J. Educational Content Semantic Modelling for Mining of Training Courses according to the Requirements of the Labor Market. *Proceedings of the 1st International Workshop on Technologies of Digital Signal Processing and Storing*, Russia, Ufa, UGATU, 2015, pp. 214–218.

7. Berry M.W., Kogan J. Text Mining: Theory and Applications. John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom, 2010, pp. 3–19.

8. Mikolov T., Sutskever I., Chen K., Corrado G.S., Dean J. Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.

9. Rehurek R., Sojka P. Software Framework for Topic Modelling with Large Corpora. *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, Malta, Valletta, 2010, pp. 45–50.

***Received 16 April 2017***

**УДК 004.855** **DOI: 10.14529/ctcr170301**

# ПОДХОД К СРАВНЕНИЮ ОБРАЗОВАТЕЛЬНЫХ КУРСОВ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА

***Д.С. Ботов, Ю.Д. Кленин***

*Челябинский государственный университет, г. Челябинск*

По мере роста числа образовательных программ и курсов потребность в методах их сравнения становится очевидной. В этой статье рассматривается общее состояние сферы интеллектуального анализа образовательного контента, разнообразие существующих форматов и типов документов, представляющих образовательный контент, и предлагается комбинированная мера близости для программ образовательных курсов. Предложенная мера близости использует три наиболее важных составляющих программы курса: описание курса, образовательные результаты курса и структура курса. В статье описывается предложенный подход к вычислению меры близости для пар каждого из этих компонентов, а также представлены первичные результаты экспериментов и их оценка с использованием метрики mean average precision.

*Ключевые слова: интеллектуальный анализ образовательного контента, программа образовательного курса, учебный план, анализ данных, мера близости.*

**Ботов Дмитрий Сергеевич**, старший преподаватель кафедры информационных технологий и экономической информатики, Челябинский государственный университет, г. Челябинск; dmbotov@gmail.com.

**Кленин Юлий Дмитриевич**, магистрант кафедры информационных технологий и экономической информатики, Челябинский государственный университет, г. Челябинск; jklen@yandex.ru.

***Поступила в редакцию 16 апреля 2017 г.***

**ОБРАЗЕЦ ЦИТИРОВАНИЯ**

Botov, D.S. Approach to Educational Course Comparison Using Natural Language Processing Techniques / D.S. Botov, Yu.D. Klenin // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». – 2017. – Т. 17, № 3. – С. 5–14. DOI: 10.14529/ctcr170301

**FOR CITATION**

Botov D.S., Klenin Yu.D. Approach to Educational Course Comparison Using Natural Language Processing Techniques. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics*, 2017, vol. 17, no. 3, pp. 5–14. DOI: 10.14529/ctcr170301