Comparison of Vector Space Representations of Documents for the Task of Matching Contents of Educational Course Programmes

Julius Klenin and Dmitry Botov

Chelyabinsk State University Information Technologies Institute, Chelyabinsk
Chelyabinskaya oblast' 454001, Russian Federation,
jklen@yandex.ru

Abstract. This article presents the study of the topical problem of semantic analysis and comparison of educational courses. This is required in order to update educational programmes in the reality of continuous growth of the amount of available educational content on the Internet, recurring changes in the requirements of standards and the labor market demands. At the moment there are no effective tools for intellectual analysis and processing of educational content.

We review various approaches to the semantic analysis of the educational courses programmes via their vector representations. We present the first to our knowledge experimental quality evaluation of vector space models for text representations of educational course programme documents. More specifically we compare the quality of various popular algorithms: TF-IDF, LSA, LDA, averaged word2vec, paragraph2vec. The evaluation is carried out using various algorithms of clustering and classification on our experimental corpus of educational course programme, used by Russian universities.

Keywords: vector space model, educational content, educational course programme, word embedding, document modelling, paragraph vectors

1 Introduction

One of the major trends emerging in modern higher education is fast and continuous change to existing standards of education, professional standards for graduating specialists and overall recommendations and requirements applied to produced educational content and documentation. Another trend is fast growth of amounts of content available from various sources. With the recent boom of distance education and MOOC (Massive Open On-line Course) systems the competition on the market of educational services is reaching new heights.

With these issues combined, the need for fast analysis and synthesis of educational content arises. Educators have to consider the requirements of various standards and guidelines, requirements of the labor market, relevance of their content and its overall quality, which is only possible to do by comparing it to other existing content. Such data is easy enough to find, in fact, the Internet is almost overflowing with it, making sifting through it an almost impossible task.

There is no real system that would allow for fast comparison, search and ranking of programmes in order to form specific recommendations for educators, as to how they can improve them.

In this paper we focus on evaluation of variety of vector space models as means of producing easy to manage and compare basic feature vectors of educational course programmes. We aim to see which algorithms are better fit to model this specific type of content and thus, would work as a foundation for more complex algorithms.

2 Related Work

Educational data mining (EDM) is a discipline concerned with applying data mining techniques to the educational content. Some of the research in this field is focused on evaluating student-generated content, designed to make the process of grading simpler. For instance, a group researchers [1] is working on a system capable of ranking the readability of text using multilevel (word, semantics, syntax and cohesion levels) linguistic features. Their experiments on Chinese textbooks use discriminant analysis and support vector machines for classification. Another system, called Writing Pal [2] is trained to grade an essay, depending on its linguistic, rhetorical, and contextual features using stepwise regression.

Ontology construction is a popular topic of research, since it makes document comparison more uniform, less dependent on text. For example, curriculum and syllabus ontologies, suggested in [3] are used in a general algorithm for mapping syllabus to the specific knowledge units, which allows for easier classification of it. [4] presents algorithm for classifying examination questions into the concept hierarchy of knowledge domain to determine what exactly the question evaluates. Ontologies are also used by Uzhva in [5] as means for performing precedent-based educational content searches. An approach to course programme comparison via ontologies is suggested in [9].

In-depth overview of various text clustering and classification approaches is presented in [12].

The main issue of these approaches is the dependence on a team of experts, manual assessments and ontology building for every knowledge domain.

3 Representations of Educational Course Programmes

While there are different types of educational documents, it is worth noting that the format of this content varies greatly not only between different types, but also within one, based on organization, departments or authors.

In this paper we are focusing specifically on educational course programmes. In order to understand the average structure and contents of such documents, we analyzed various formats used by universities and MOOC organizations all around the world.

Overall, our research showed the distribution of certain elements in reviewed documents, presented in figure 1.

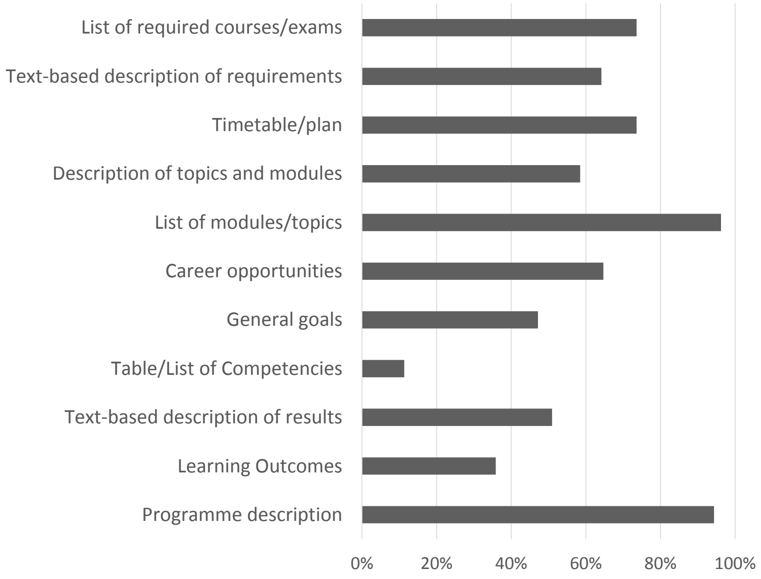

Fig. 1. Occurrence of certain elements in various educational programmes

From this, a 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 more or less specific, practical results;

We suppose that these components are the most important parts of the document and use them as means of comparing courses.

The relation between main elements, mentioned above, and components of the course programme is discussed in more detail below.

3.1 Programme Description

Overwhelming majority of programmes include a brief description. In terms of course programme elements, this description corresponds to programme description (or introduction, or annotation), course goals, and place of the discipline in the educational programme structure. Goals specify broad descriptions of what is expected of a student after completion. The last element specifies relations with neighboring disciplines within the same educational programme - which provide basis required for this course and which depend on this one.

The description is usually a plain text with no inner structure.

3.2 Programme Structure

This element is not confined to a specific naming convention and may include lists, tables or both. It usually specifies the topic and concepts covered by every lecture, practical class, student's own studies and so on. Most often first structure specifies topics and their order, including time, and the class format - lecture, lab work, homework, etc. After that, usually come more details of specific terms, concepts and ideas covered within each topic.

This data within is usually represented as either plain text, or, even more commonly, as a set of concepts, listed one after another.

3.3 Educational results

This element is a list of results, which student should demonstrate after successful completion of the course.

Representations of this element may vary. Most western organizations and some Russian ones, prefer learning outcomes: specific format, consisting of two main parts the action verb, describing the kind of knowledge (being able to recall certain information or to classify a presented sample, for example), and terms describing the knowledge. Action verbs are usually restricted to relatively small taxonomies, while terms, are only limited by the domain of the course.

Russian educators usually use competencies - a broader description of knowledge. In programmes there is usually a section for results. Here competencies are described and matched with specific results. These are similar to the learning outcomes, however the verb, while usually being ”know”, ”can” and ”wield” is not actually governed by any taxonomy.

4 Method

In this paper we present the results of two experiments with vector space models. The first one aims to assess whether or not selected models could produce vectors of high enough quality for educational courses. Results of each model is assessed by using learned vectors to perform the document clustering task and evaluate the resulting clusters. The second experiment is the classification on the same dataset, with the quality of class assignments representing the overall quality of vector space models. The overall structure of the algorithms we implement in our experiments is presented in figure 2.

For our experiment we first reduce full documents to structural elements discussed in 3 and perform basic preparation and processing - deformatting, clean up and lemmatization. The classification task also includes additional stage in which we split the corpus into training and test sets. After this initial stage, the data is ready to be consumed by the vector space models in order to generate their vector representations. Once this is done we can move onto the actual clustering or classification task, by feeding the vectors into respective algorithms.

Finally we perform the evaluation of resulting clusters/classes.

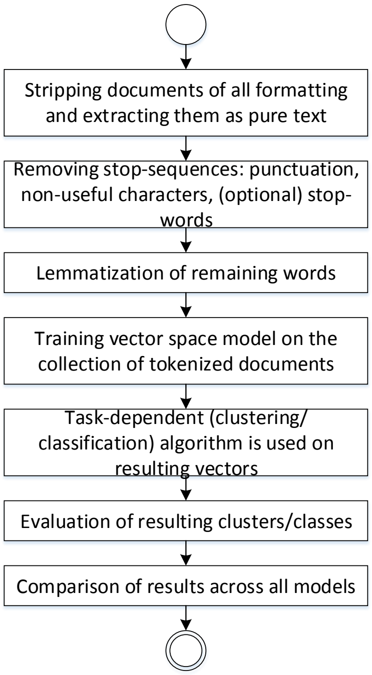
Stripping documents of all formatting and extracting them as pure text
Removing stop-sequences: punctuation, non-useful characters, (optional) stop-words
Lemmatization of remaining words
Training vector space model on the collection of tokenized documents
Task-dependent (clustering/ classification) algorithm is used on resulting vectors
Evaluation of resulting clusters/classes
Comparison of results across all models

Fig. 2. General structure of the text analysis algorithm

4.1 Building the Educational Course Corpus

One of the main problems in our research is the lack of an existing corpus (at least there are none available to the researchers). To perform our experiments we have attempted to create one ourselves.

The availability of documents can sometimes be a major problem depending on a discipline. While universities are required to publish all of their current educational and course programmes on their websites, large part of them choose not to. Second issue we faced is the document type variety. Most organizations provide their programmes in either open office xml (docx, specifically) or PDF formats. But some opt to use low quality scans (analysis of which requires OCR techniques), RTF documents, or something else. Third problem is that the documents contain complex, often poorly made data structures. This also shifts the task of document analysis from NLP towards optical recognition.

Overall it appears that corpus construction in this field is either a task for software of high complexity, or a manual labor task. For current research we decided to collect initial dataset manually.

Our current corpus contains just over a hundred different educational course programmes, made by various Russian universities. For each discipline there are roughly 4 documents. Groups of 3-5 similar disciplines fall under the shared knowledge domain. There are 7 such domains: information technologies, economical studies, mathematics and statistics, linguistics, history, medicine and law.

Overall metrics for our corpus are: 25k sentences and a total of 129k normalized tokens.

4.2 Clustering Task

As stated above, we perform the task of clustering on vectors, generated by each model. Evaluation of clusters is then performed to show how well can an basic clustering algorithm discern documents, based only on vectors and no other knowledge.

Since we dont yet have document evaluation performed by human experts, we are using two levels of classes, that are already available - the discipline of each course, and the general knowledge domain that discipline would fall under.

We have selected three popular clustering algorithms: agglomerative cluster with average linking, k-means and Wards clustering algorithm. While agglomerative clustering works with cosine similarity measure basic k-means and Ward do not. This also allows us to determine if we can replace cosine similarity with Euclidean. To do so, we normalize the vectors, which means the distance between two vectors only depends on the cosine of the angle between them. Two measures were compared in [10], showing similarity of their results.

4.3 Classification Task

Similar to clustering we use two existing sets of classes within data to evaluate classification results the discipline of each course, and the knowledge domain that discipline would fall under. We also divide our corpus into training and test sets, roughly 3 to 1 in number of documents, while trying to keep all classes represented in both.

For this experiment we selected several popular and well established classification algorithms: logistic regression, decision tree classifier, k-nearest neighbors, c-support vectors (svc) and two tree-based ensembles - random forest classifier and extra-random trees. These are generic algorithms used often in various machine learning tasks.

4.4 Vector Space Models

There are a lot of different algorithms, which can be considered vector space models, since the only requirement for them is to be able to represent the textual documents in a form of a numeric vector. The main appeal for these models is the ability to represent complex information in a relatively simplistic form, that allows for vector calculus to be applied to analysis of texts.

In this paper we decided on evaluating models of various complexity based primarily on their popularity within the NLP community. An important quality of these models is that they are based on documents being constructed out of word tokens, which means, that upon finishing the training, each model can be used to infer the vector for a never before seen document, as long as it shares the vocabulary with the training corpus.

Below we give a brief overview of each used model.

TF-IDF Tf-idf is a popular approach when it comes to transforming from text-based information into numeric operations due to its simplicity. Tf-idf is a combination of two basic scores of statistical importance of a word in a corpus - the frequency of word's occurrences, suggesting that more important words appear more often, and the inverse frequency of its occurrences in collection overall, proposing that term is more important if it occurs only in some documents, making them stand out.

Tf-idf weights for all terms in the corpus make up a sparse term-document matrix, where columns are the numeric vectors for each document in the collection [13]. These vectors are be denoted as follows

$$d_i = [w_1, w_2, \ldots, w_N] \tag{1}$$

where

$$w_i = frequency(j, i) * \log_2 \frac{K}{documentfrequency(j)} \tag{2}$$

Here, the corpus consists of K documents and holds N unique words. $frequency(j, i)$ is the number of occurrences of the word w_j in the document d_i, and $documentfrequency(j)$ is number of documents in the collection, in which w_j occurs.

The main issue of tf-idf is resulting dimensionality, equal to the number of unique terms used in the collection, which can be all the words in the language (1M+ for English). This leads to sparseness, due to each document only using so many words.

Word Embeddings As mentioned above, since models, such as tf-idf, assign each word a singular dimension in their vector space, their vectors are extremely sparse. The models we review below try to solve this issue in different ways and can be referred to as word embedding algorithms.

The term word embedding describes NLP techniques that generate vectors with less dimensionality. Term refers to the mathematical embedding - mapping of one space into another - in this case - mapping high dimensional space into continuous vector space with fewer dimensions.

While distributional semantic models, topic models and neural language models all fall under this term, there exists an opinion in machine learning community, that only a certain subset of algorithms qualifies to be called word embeddings usually referring to neural language models. In this paper we are using this term for neural network-based approaches as well as topic modelling and DSM algorithms.

LSI Latent Semantic Indexing [14], also known as Latent Semantic Analysis attempts to build a low-rank approximation of the original term-document matrix by applying a Singular Value Decomposition to it.

In the process, some rows will merge, while preserving the correlation between columns. The approach merges the most similar rows together, which, means that the merged terms have similar weights, which, according to the distributional hypothesis, means that they have similar meanings.

Formally SVD approximates term-document matrix likes so

$$X = U\Sigma V^T \tag{3}$$

the resulting document vectors can be represented as rows of V^T, or

$$d_i = \Sigma_{i,i} * V_i^T \tag{4}$$

LSI provides dense vector representation of the document collection with each dimension being a general concept - a vague combination of similar terms. This approach is more efficient, but has a different issue - the dimensions are less interpretable than separate words in tf-idf model.

In our experiments the model is trained to recognize 10 dimensions, which we found to be the most number of distinct dimensions of this kind in our dataset.

LDA Similar approach at document representation involves topic models. These algorithms are used to extract topics from documents and determine what topics does document cover and what topic does each word come from. The documents can then be represented with a vector of probabilities of each topic appearing in the document, with the dimensionality of number of such topics.

One of the popular topic models is Latent Dirichlet Allocation. This model treats each document as a mixture of topics and uses Dirichlet prior to generate the initial proportions for each topic. After this initial distribution, model attempts to enhance it through Gibbs sampling, for example. This iterates over words in the document, updating the probabilities of word and document belonging to a topic. The result is two low-rank matrices. First contains vectors of terms over topics and second is the same for documents. Their dimensionality is the same, so it can be said, that LDA decomposes original term-document matrix into two thinner, denser ones.

Similar to LSI we have found that settling for 10 topics in collection provides the best results.

Word Vectors One of the more recent trends in vector space models involves using various neural networks to learn word and document vectors in low dimensional space. Overwhelming popularity has been achieved by one of such models - Word2Vec, created by a team of researchers from Google, led by Tomas Mikolov [6].

Word2Vec is a two-layer neural network, which uses distributional semantics to learn the correlation between words and their contexts. Two architectures are presented: continuous bag-of-words and skip-gram. First is trained to predict words based on context words. Second takes a single word and tries to predict probabilities of other words being its context. Word2Vec is trained in such a way that vectors of distributionally similar words start getting closer and the same goes for contexts. It has been proven in [7], that word2vec is doing an approximation of a matrix factorization over term-context matrix, although in a modern and computationally efficient way, however, unlike previous models the dimensions cannot be interpreted and are completely arbitrary.

In our experiments we train skip-gram model, using negative sampling as optimization technique to learn vectors with 50 dimensions, matching the dimensionality of paragraph vectors. We then calculate document vectors, as suggested in [11]. One way to do this is averaging of word vectors for each document

$$d_i = \frac{\sum\limits_{j \in N} w_{i,j}}{\mid d_i \mid} \tag{5}$$

with $w_{i,j}$ being j-th word in document i, and $\mid d_i \mid$ standing for the number of words in i-th document.

Since not all words share importance, we apply the tf-idf weights to word vectors and create the set of weighted averaged word2vec vectors

$$d_i = \frac{\sum\limits_{j \in N} TFIDF(w_{i,j}) * w_{i,j}}{\mid d_i \mid} \tag{6}$$

Paragraph Vectors Paragraph2Vec [8] is an approach, similar to Word2Vec, that was applied to entire documents. The only important difference from Word2Vec is the use of a secondary matrix, which consists of vectors for documents encountered in training.

The model also includes two NN structures. Distributed Memory model (PV-DM) is similar to CBOW, using both context vectors and the paragraph vectors to predict a word from the sampled context. Distributed Bag-of-Words (PV-DBOW) is similar to the skip-gram model, only using paragraph vector to predict words, sampled from these paragraphs.

In our experiment we train both models of paragraph2vec network and task them with learning document vectors with 50 dimensions.

5 Evaluation

Below we present and discuss the results of both experiments performed for this article: the clustering and classification of educational courses based on their vector representations.

5.1 Educational Course Clustering

In order to evaluate the quality of clusters we use a variety of metrics for quality of clustering evaluation: adjusted Rand score, adjusted mutual information measure, homogeneity, completeness, harmonic mean of the last two - v-measure, Fowlkes-Mallows score and Silhouette score. All of these, save the last one, estimate how well the clusters discern real classes, present in dataset, while silhouette score determines how well defined the clusters themselves are. For the first six metrics, the value lies in range from 0 to 1, with 1 being perfect match of clusters and real classes. Silhouette score, on the other hand, is the value in range of -1 to 1, with -1 being erroneous clustering, 0 signalling of overlap in clusters and 1 being well formed clusters.

Results for the first case of clustering - course-specific clusters are presented in table 1. The results for more general knowledge domain level of clusters are shown in table 2.

Table 1. Evaluation of cluster quality for course clusters

Model	Clusters	Adj.	Adj.	Homo-	Comple-	V-score	Fowlkes-	Silho-
		Rand	MI	geneity	teness		Mallows	uette
TF-IDF	Agglom.	0.2788	0.3896	0.8419	0.7439	0.7899	0.3384	**0.7606**
	Ward	0.2758	0.3385	0.8035	0.7514	0.7765	0.3116	0.2563
	K-means	0.2832	0.3373	0.8053	0.7454	0.7742	0.3223	0.2873
LSI	Agglom.	0.4035	0.4812	0.8623	0.8035	0.8319	0.4414	-0.2237
	Ward	0.3933	0.4241	0.8207	0.8089	0.8148	0.4111	-0.1266
	K-means	0.3619	0.4117	0.8214	0.7964	0.8087	0.3842	-0.1538
LDA	Agglom.	0.2600	0.3458	0.8186	0.7265	0.7698	0.3107	-0.4902
	Ward	0.3021	0.3834	0.8173	0.7749	0.7955	0.3318	-0.4045
	K-means	0.2694	0.3429	0.8036	0.7552	0.7787	0.3017	-0.4609
Avr. W2V	Agglom.	0.2073	0.2761	0.7808	0.7182	0.74820	0.2473	-0.051
	Ward	0.1243	0.1640	0.7345	0.6680	0.6997	0.1676	0.0852
	K-means	0.1271	0.1728	0.7387	0.6666	0.7008	0.1726	0.0899
Weighted Avr. W2V	Agglom.	0.0944	0.2229	0.7880	0.6357	0.7037	0.1842	-0.0578
	Ward	0.0817	0.1599	0.7445	0.5814	0.6529	0.1617	0.0668
	K-means	0.0788	0.1529	0.7385	0.5853	0.6530	0.1542	0.0941
PV-DM	Agglom.	0.7795	0.8301	**0.9529**	0.9281	**0.9403**	**0.7904**	0.3632
	Ward	**0.7829**	**0.8305**	0.9367	**0.9312**	0.9339	0.7897	0.3968
	K-means	0.7273	0.7854	0.9275	0.9107	0.9190	0.7376	0.3600
PV-DBOW	Agglom.	0.6202	0.6923	0.8936	0.8698	0.8815	0.6363	0.2077
	Ward	0.6587	0.7123	0.8875	0.8835	0.8855	0.6690	0.2266
	K-means	0.6540	0.6985	0.8846	0.8769	0.8807	0.6649	0.2064

In both cases the paragraph vectors have shown the best quality of clusters, even reaching perfect matching with real classes in case of general knowledge domain clustering.

Applying tf-idf weighting improved the score of word2vec models, which shows that models need to have a way of dealing with word importance.

Other models have performed fairly mediocre, which can be expected of them especially since for most models silhouette score shows overlapping clusters, which reflects the overlapping of the real classes, existing in the dataset.

5.2 Educational Courses Classification

The results for both groups of classes are presented in table 3.

Table 2. Evaluation of cluster quality for general knowledge domain clusters

Model	Clusters	Adj. Rand	Adj. MI	Homogeneity	Completeness	V-score	Fowlkes-Mallows	Silhouette
TF-IDF	Agglom.	0.2210	0.3280	0.4667	0.4234	0.4440	0.3494	**0.6446**
	Ward	0.2130	0.3720	0.5777	0.4628	0.5139	0.3880	0.2054
	K-means	0.2645	0.4087	0.5528	0.4952	0.5224	0.3951	0.3171
LSI	Agglom.	0.3959	0.5492	0.7329	0.6130	0.6676	0.5262	0.3582
	Ward	0.4661	0.5617	0.6695	0.6262	0.6471	0.5525	0.2535
	K-means	0.3320	0.4973	0.6006	0.5723	0.5861	0.4350	0.1294
LDA	Agglom.	0.2510	0.3733	0.4747	0.4689	0.4718	0.3593	-0.1104
	Ward	0.2418	0.3631	0.4982	0.4553	0.4758	0.3685	-0.109‘1
	K-means	0.2155	0.3316	0.4773	0.4286	0.4516	0.3525	-0.1239
Avr. W2V	Agglom.	0.2484	0.3038	0.4530	0.3984	0.4240	0.3797	0.1575
	Ward	0.0973	0.1547	0.3704	0.2523	0.3002	0.3068	0.2045
	K-means	0.0896	0.1281	0.3371	0.2362	0.2778	0.2972	0.1982
Weighted Avr. W2V	Agglom.	0.1630	0.2857	0.5147	0.3849	0.4404	0.3597	0.0096
	Ward	0.0428	0.1564	0.4532	0.2689	0.3376	0.3031	0.2040
	K-means	0.0466	0.1347	0.4712	0.2449	0.3223	0.3266	0.4871
PV-DM	Agglom.	**1**	**1**	**1**	**1**	**1**	**1**	0.253261
	Ward	0.9105	0.9205	0.9438	0.9291	0.9364	0.9237	0.2346
	K-means	0.8502	0.9078	0.9453	0.9179	0.9314	0.8735	0.2310
PV-DBOW	Agglom.	0.7915	0.8353	0.8564	0.8534	0.8549	0.8213	0.1125
	Ward	0.7088	0.7724	0.8141	0.7974	0.8057	0.7522	0.1067
	K-means	0.4067	0.5230	0.5890	0.5754	0.5821	0.4951	0.0645

As expected, similarly to the previous experiment, the best quality is achieved by paragraph vectors, even reaching perfect class assignments, which confirms that vectors produced by paragraph2vec are good enough for semantic analysis of course programmes. Another expected result - the quality is higher for the task with more generic classes - knowledge domains. Another notable fact is that, aside for the decision tree classifier and random forest classifier, all classification algorithms showed similarly high results, which means the achieved quality does not depend as much on the classifier itself, as it does on the actual vector space model.

Table 3. Evaluation of course classification tasks

		Courses			Knowledge Domains		
		Precision	Recall	F-measure	Precision	Recall	F-measure
TF-IDF	Extra-trees	0.47	0.58	0.50	0.70	0.73	0.70
LSI	Extra-trees	0.38	0.50	0.41	0.93	0.92	0.92
LDA	Logistic Regression	0.31	0.50	0.37	0.76	0.81	0.76
Avr. W2V	Decision Tree	0.24	0.31	0.26	0.48	0.42	0.40
W. A. W2V	K-nearest neighbors	0.37	0.50	0.41	0.76	0.69	0.69
PV-DM	K-nearest neighbors	**0.88**	**0.92**	**0.90**	**1**	**1**	**1**
PV-DBOW	Random Forest	0.88	0.92	0.90	0.97	0.96	0.96

Similarly to the clustering task, averaged word2vec approaches were not able to produce good enough vectors to successfully classify the documents, while other models again showed rather average scores.

6 Conclusion

We have presented the results of the evaluation of various vector space models and their applicability to the analysis of educational courses. Paragraph2Vec approach gives the best results for both clustering and classification tasks. In future research we are considering growing the corpora, adding more interpretable features to existing vectors, using structural information in analysis and studying other types of educational content.

References

1. Sung et al. *Constructing and validating readability models: the method of integrating multilevel linguistic features with ma-chine learning.* Behavior Research Methods, 47 (2) (2015), pp. 340-354.
2. McNamara D.S., Crossley S.A., Roscoe R.D. *Natural language processing in an intelligent writing strategy tutoring system.*. Behavior Research Methods, 45, 2013, pp. 499-515.
3. Chung H., Kim J. *An Ontological Approach for Semantic Modeling of Curriculum and Syllabus in Higher Education.* International Journal of Information and Education Technology vol. 6, no. 5, pp. 365-369, 2016.
4. Foley J., Allan J. *Retrieving Hierarchical Syllabus Items for Exam Question Analysis.* Advances in Information Retrieval, March 2016, pp. 575-586.
5. Uzhva A.Yu. *Automatic development of ontology model for case-based reasoning in search of eductional resources using analyzys of education programms.* Modern problems of science and education. no. 1, 2013.
6. 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.
7. O Levy, Y Goldberg *Neural word embedding as implicit matrix factorization.* Advances in neural information processing systems, 2177-2185, 2014.
8. Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J. *Distributed Represenations of Sentences and Documents.* In Proceedings of ICML 2014, pp. 11881196, 2014.
9. Chernikova E. *A Novel Process Model-driven Approach to Comparing Educational Courses using Ontology Alignment*, 2014.
10. Qian, G. et al. *Similarity between Euclidean and cosine angle distance for nearest neighbor queries* In: SAC04: proceedings of the 2004 ACM symposium on applied computing New York, NY, USA: ACM 2004. P. 12321237
11. Lilleberg J., Zhu Y. and Zhang Y., *Support vector machines and Word2vec for text classification with semantic features.* IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC), 2015.
12. Aggarwal C., Zhai C., *Mining Text Data* Springer Publishing Company, Incorporated, 2012.
13. Manning C. D., Raghavan P. and Schtze H. *Introduction to Information Retrieval.* Cambridge University Press. 2008.
14. Deerwester S. et al. *Indexing by Latent Semantic Analysis.* Journal of the American Society for Information Science, 1990, 41 (6): 391407.