Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 136 (2018) 293–303

Procedia Computer Science

www.elsevier.com/locate/procedia

7th International Young Scientist Conference on Computational Science

Combined document embedding and hierarchical topic model for social media texts analysis

Amir Uteuov*, Anna Kalyuzhnaya

ITMO University, 197101, 49 Kronverksky pr., St Petersburg, Russia

Abstract

Exploring customer interests from open source information has become a significant issue. On the one hand, consumers deepen their engagement with the brands which values matter to them. On the other hand, annoying marketing calls and polls do not reflect real customers' needs and wants. This article considers topic modeling in application to social media analysis. We have received interpretable topics related to users preferences. Crawled posts texts and texts obtaining from images by an optical character recognition were used as datasets. Focusing on two approaches: probabilistic (LDA, ARTM) and neural network based (doc2vec, word2vec), we suggest the combined model deARTM. Hierarchical ARTM model allows us to obtain relations between texts in several abstraction levels which we used as vector representation. To avoid misspelling sensitivity, our model includes document embedding. In the experimental part, we show that our model can improve results of topic modeling on social media datasets.

Keywords: social media, information retrieval, topic modelling, Additive regularization of topic models, document embedding;

1. Introduction

In this article, we use topic modeling for analyzing public texts of social media communities and users to get interpretable user interests and preferences. The main public resource of user data is social media. This data source is useful for business, government and social researchers. Companies collect data to find a target audience and customize products and services for consumers' demands. We consider the topic modeling problem in application to social media texts. There are many topic models. However, social media texts have noise, misspelling and jargon that make the

* Corresponding author.
*E-mail address:*auteuov@niuitmo.ru

10.1016/j.procs.2018.08.285

analysis difficult. In this article, we suggest a model, which combines probabilistic hierarchical topic modeling with document embedding technique. We provide experiments to demonstrate that document representation in different abstraction levels (hierarchical) in combination with document embedding improve quality of modeling.

There are many works devoted to social media analysis [1]–[3]. Social media has different kinds of data: text, images, audios, links between nodes (posts, users, communities etc.) and time series. Focusing on text mining here, we consider public posts as a collection of documents with latent topics which characterize users and communities. Thus, this task can be interpreted as an unsupervised topic modeling. The goal is an extraction of abstract topics related to the semantic structure of the text. We want to produce clusters in the space of documents, containing similar texts by content and meaning.

Topic modeling is a tool for automatically exploring large document collections, it produces a short description to which each document are related. Moreover, it provides probability estimations of a topic influence on a document and a word influence on a topic. In our study, we concentrate on analyzing real social media texts and build a combined model to increase the quality of topic modeling. We obtained topics such as drivers, cooks, books, goods, religion and services.

In section 3, we provide the basics of topic modeling and neural embedding. In section 4, we discuss suggestions for improvement and describe our developed model. In section 5, we show experiment results on our social media datasets. In section 6 we discuss results and our developed model.

2. Related works

The social media analytic brings benefits to business, entertainment, science, crisis management and politics [4] There are many works applying topic modeling methods [3]–[5]. One of the first topic models was Probabilistic Latent Semantic Analysis (Probabilistic latent semantic analysis)[6]. It finds document clusters and supports probabilistic representation of each document in the space of topics and connections of each topic with weighted top tokens. A more flexible model is Latent Dirichlet Allocation [7](LDA), that suggests Dirichlet distributions over generated terms and topics. A next generalization step is the Additive Regularization of topic models [8] (ARTM), providing regularizers with a technique to adjust the model. Unlike the previous models, it can use different assumptions about word and topic matrices. A significant issue is the evaluation of topic models [9].

In natural language processing (NLP) there are several word embedding techniques: word2vec[10], Global vectors for word representation[11], doc2vec[12], fastText [13], StarSpace [14]. The general idea is representation of word or document in vector space where similar words have similar coordinates. The interesting fact is that relations between words connected with operations on their vectors. For example, vector(“king”) – vector(“man”) + vector(“woman”) = vector(“queen”) and so on. The one can use output vectors as features for any NLP tasks. In addition, there are pre-trained vectors on large text corpuses (Glove, word2vec, fastText). These methods got a wide application in NLP tasks: machine translation [15], speech recognition [16], text similarities [12] and even malware detection [17].

However, coordinates of embedding vectors do not have probabilistic meaning. In each corpus, we get different vector representations. Probabilistic models have limitations, they strictly depend on tokens spelling, initial dictionary, and do not allow a clustering of completely new documents for this collection. Improving these approaches researchers have developed coupled models. LDA2vec [18] proposes to combine word embedding and LDA model using both representations. Topical word embedding (TWE) [19] combines skip-gram embedding model with LDA. Probabilistic word embedding [20] implements the ARTM approach on a word-word matrix and produces a vector representation in probability terms.

3. Topic modelling

Topic modelling is a method for documents representation by a finite number of latent topics. Probabilistic topic models are unsupervised generative models [8]. A token/term is a word or ngram (n words joined), which we treat as a minimal meaning unit. Document d is a set of words/tokens W. A topic t is a probability distribution over words/terms in a vocabulary.

The main assumption is that documents are a matrix product of word-topic and topic-document matrices (1).

$$\frac{n_{dw}}{n_d} \approx \Phi\Theta = (\Phi S)(S^{-1}\Theta) = \Phi'\Theta' \quad (1)$$

In addition, we suggest:

- Each topic is a probability distribution over words, topic depends on own terms vocabulary
- The document is a probability distribution over topics, document relates to finite topics number

Additional restrictions to get a probability interpretation of topics:

$$\sum_{w \epsilon W} \phi_{wt} = 1; \sum_{t \epsilon T} \theta_{td} = 1; \phi_{wt} > 0; \Theta_{td} > 0$$

- φ_{wt} - probability of the term *w* in each topic *t*
- θ_{td} - probability of topic *t* in each document *d*

Therefore, a generative word distribution is:

$$p(w|d) = p(w|t)p(t|d) = \sum_{t \epsilon T} \varphi_{wt}\theta_{td}$$

We can find it through optimization problem (2)

$$\sum_{d \epsilon D} \sum_{w \epsilon d} n_{dw} \, ln \sum_t \phi_{wt}\theta_{td} + R(\Phi, \Theta) = L(\Phi, \Theta) + R(\Phi, \Theta) \rightarrow max \quad (2)$$

The method to solve it is EM algorithm:

$$\begin{cases} p_{tdw} = norm_{t \epsilon T}(\phi_{wt}\theta_{td}) \\ \phi_{wt} = norm_{w \epsilon W}\left(n_{wt} + \phi_{wt}\frac{\partial R}{\partial \phi_{wt}}\right), n_{wt} = \sum_{d \epsilon D} n_{dw} p_{tdw}, \\ \theta_{td} = norm_{t \epsilon T}\left(n_{td} + \theta_{td}\frac{\partial R}{\partial \theta_{td}}\right), n_{td} = \sum_{w \epsilon d} n_{dw} p_{tdw}, \end{cases}$$

In this notation, $R(\Phi, \Theta)$ defines a type of probabilistic model.

Probabilistic latent semantic analysis [6] is a first probabilistic topic model. $R(\Phi, \Theta) = 0$. In this case, results matrixes cannot contain zero rows, this model cannot smooth or sparse distributions.

Latent Dirichlet Allocation is a generative model, using an assumption about the Dirichlet distribution genesis of Φ, Θ. In terms of regularizers, it has a cross-entropy regularizer

$$R(\Phi, \Theta) = \sum_{t,w} \beta_w ln \, \varphi_{wt} + \sum_{d,t} \alpha_t ln \, \theta_{td}$$

Where α, β are parameters of Dirichlet distributions, which can adjust word and topic distributions.

Hierarchical Dirichlet process (HDP) is an extension of LDA to produce hierarchic levels of topics. In this case, one can obtain several sets of topics in different hierarchic levels.

Additive regularization of topic models (ARTM) [8]– an extension of PLSA, providing regularizers technique to adjust model parameters, where $R(\Phi, \Theta)$ adds restrictions on topic model to provide fine-tuning. In this case, we can add several regularizers to get into account our wishes on topics.

Smoothing and sparsing regularization

$$R(\Phi, \Theta) = \beta_0 \sum_{t \epsilon T} \sum_{w \epsilon W} \beta_w ln \, \varphi_{wt} + \alpha_0 \sum_{d,t} \alpha_t ln \, \theta_{td}$$

Topic decorrelation regularization, row sparser for θ

$$R(\Theta) = \tau \sum_{t \epsilon T} ln \sum_{d \epsilon D} p(d) \, \theta_{td}$$

Topic covariance regularization, reducing the overlapping between the topic-word distributions

$$R(\Phi) = -\tau \sum_{t \in T} \sum_{s \in T \setminus s} \sum_{d \in D} \varphi_{wt}\, \varphi_{ws}$$

Coherence regularization

$$R(\Phi) = \tau \sum_{t \in T} ln \sum_{u,v \in W} C_{uv} \varphi_{ut}\, \varphi_{vt}$$

ARTM has hierarchic version Hierarchical ARTM, which we describe in section 4.

The state-of-the-art quality metrics of topic models are perplexity, coherence, purity and contrast [8][9]. Perplexity (3) is a measure of model train quality, smaller perplexity is better. It represents a model uncertainty to input data. The perplexity shows a model convergence per iterations. The perplexity is an estimation of log-likelihood from expression (2).

$$P = \exp(-\tfrac{1}{n} \textstyle\sum_{d \in D} \sum_{w \in d} n_{dw} \ln p(w|d)) \quad (3)$$

The coherence *c* (4) is a measure of tokens co-occurrence in texts, it relates to human estimations of topics [6][21]. It calculates times when tokens occur in one sliding window (usually 2-10 words around) and estimate to a total token occurring in the initial text. Therefore, it is an expensive computation for a large collection.

$$PMI(w_i, w_j) = log \frac{p(w_i, w_j)}{p(w_i) p(w_j)}$$

$$c(t) = PMI_t = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i}^{k} PMI(w_i, w_j) \quad (4)$$

$p(w_i, w_j)$ – probability that w_i, w_j presents in this document

$p(w)$ – probability that *w* presents in a collection

The topic purity and contrast are empirical metrics, they based on our suggestion that topic should have different top tokens with high weights.

$$Purity_t = \sum_{w \in W(t)} p(w|t); Contrast_t = \frac{1}{|W_t|} \sum_{w \in W(t)} p(t|w);$$

The bag of words hypothesis [22] assumes that a word order in documents does not influence on a content. Therefore, it is possible to represent text as a set of words with some weights, in simple case it is a term frequency. Term frequency inversed document frequency (tf-idf [22]) is a bag of words measure for each word per each document, related to words occurring in a whole text collection and in current document (1). It can be useful to adjust of an importance for each token (3).

$$tf\text{-}idf = tf(t,d) \cdot idf(t,D) = \frac{n_t}{\sum_k n_k} \cdot log \frac{|D|}{|\{d_i \in D | t \in d_i\}|} \quad (5)$$

Tf-idf representation has applications for classification and text similarity tasks as a baseline model.

3.1. Word embedding

Word embedding is a way of word/texts representation to vector space, where similar words by meaning have similar coordinates [10]. The interesting property is that words by meaning connected with linear operations on their vectors in this space. For example, vector(“king”) – vector(“man”) + vector(“woman”) = vector(“queen”). The embedding vectors are dense instead of sparse ones in probabilistic models (LDA, ARTM). There are many word embeddings approaches.

The skip-gram negative sampling (SGNS) word embedding model learns word representation by predicting a local context for each word in a collection. It analyses co-occurrence of word pair (u, v). This is skip-gram negative sampling optimization problem [20]:

$$p(u|v) = \frac{exp \sum_t \varphi_{ut} \theta_{tv}}{\sum_{w \in W} exp \sum_t \varphi_{wt} \theta_{tv}}$$

$$\sum_{v \in W} \sum_{u \in W} n_{uv} \log \sigma(\textstyle\sum_t \varphi_{ut} \theta_{tv}) + k \mathbb{E}_v \log \sigma\,(-\sum_t \varphi_{ut} \theta_{tv}) \rightarrow max \quad (6)$$

where σ - sigmoid function, u, v – pair of words, k – positive and negative samples parameter.

The alternative way is a continuous bag of word (CBOW) predicting word by its context [12].
There are neural word embedding implementations:

- Word2vec [16] – approach based on SGNS, CBOW
- Glove [11] – neural network based approach and pre-trained vectors set in different languages
- doc2vec/paragraph2ve [12] – an extension of word2vec for documents (DBOW, DM)
- StarSpace [14] – extended approach for vector representation of complex entities (“page embedding”)

4. Combined model deARTM

There are empirical suggestions: model has to be more stable to dictionary redesign [5] and model should not be sensitive strictly to tokens spelling [23], each word can have different embedding under different topics [19], topics relations can influence on representation. Also in our particular task, we want to train model one time and then infer representation for new documents. These are reasons why we develop a new model. Hierarchic ARTM model shows topics in different abstraction levels, each of them we treat as a vector in common space. These ideas are the basis of our combined model.

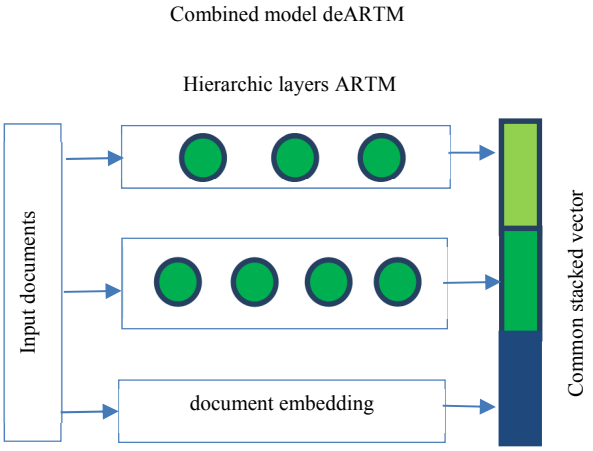

Fig. 1. Combined model: Hierarchic ARTM layers and document embedding; rectangles are model layers; green circles are topics on each layer; each layer defines part in common representation vector

The proposed model deARTM (document embedding and ARTM) consists from frequency filtering method, hARTM model with several layers and document embedding part as an addition layer (Fig. 1). Thus, our model is a combination of optimization problems for each hierarchic levels l ARTM topic model, and document embedding model:

$$\sum_l L(\Phi,\Theta) + R_l(\Phi,\Theta) \to max \quad (7)$$

$$\sum_{v\in W}\sum_{u\in W} n_{uv} \log \sigma(\sum_t \varphi_{ut}\theta_{tv}) + k\mathbb{E}_v \log \sigma\,(-\sum_t \varphi_{ut}\theta_{tv}) \to max \quad (8)$$

Hierarchic ARTM is an extension of ARTM where in one optimization problem included several ARTM models with its own topic counts (7). This model takes into count relations between topics on each layer and provides parameter for its adjustment. For a model estimation, we use perplexity, coherence metric (4) and assessors estimation for topics interpretability. We have compared results for the combined model with hARTM and doc2vec separately. We tried different stacking methods for embedded vectors: use several hARTM layers, use only last layer, simply concatenate doc2vec and hARTM layer, weighted concatenation, normalization doc2vec. In the final case, we stack in one space hARTM vectors and document embedding ones and apply a normalization procedure. Thus, we obtain for each document a new vector representation based on hARTM vectors and document embedding. Through this, the average coherence score by layer increases in 0.02, and it improves the classification task score (example of

20newsgroups in next section). In addition, we see more interpretable topics for vkposts and vkimages datasets. Building topics only on document embedding needs more research.

5. Experiments

This section describes experiments in discovering latent topics in our datasets. We use a standard experiment pipeline: preparation data, modeling and statistics estimation. We use data crawled from Russian social media from wall of communities (vkposts_words), data from wall users (vkusers_posts), raw text after optical character recognition of 4000 users images posts (img_users). These datasets were collected in period for two years for active users and communities. Table 1 describes datasets after deleting stop words. Fig. 2 shows words frequency variance for top 600 tokens in each datasets. We use it to calculate boundaries for tokens frequency filtering.

Table 1. Datasets description

Title	docs	words count	uniq_words	max term count	language
vkposts_words	1000	32922470	212361	139582	Russian
vkusers_posts_raw	4000	13120728	1798863	162200	Russian
vkmusic_texts	21500	2515322	107600	26841	Russian, English
vkusers_img	3656	2732643	739170	36438	Russian, English
facebook_nyc	148	452031	44977	3947	English
log_texts	1047218	3991780	580034	26641	Russian
20newsgroups	11314	1658417	71791	20359	English
random_4K	4048	2174745	885843	24108	-
wikinews_2noun	12112	199263	65766	3140	Russian
postnauka	1728	595760	28484	3475	Russian

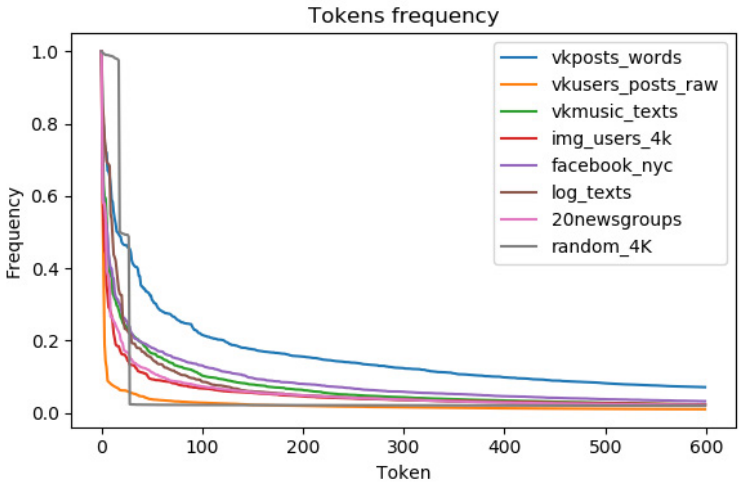

Fig. 2. Datasets tokens normalized frequency

This section details data preparation. For the datasets 20newsgroups and wikinews we perform simple text cleaning, converting to lowercase, excluding stop words. For social media data, we apply more sophisticated cleaning, excluding links, additional symbols, html tags and emoji. For the recognition of text from images we use the neural network based framework Tesseract (“Tesseract optical character recognition,” n.d.). Then, we exclude repeated symbols (more 4 times going in row) and delete non-letters characters. For all datasets we convert text documents to 1, 2, 3-gram BOW representations in Vowpal Wabbit format (“Vowpal Wabbit,” n.d.). It allows us to reduce an initial

text size up to 10 times and use real numbers for word weight, not just integer count of words but also real value (tf-idf [22]) for each token in each document. In this way, a token can have another weight from document to document, that can be useful for selecting documents significant tokens.

We conducted the following experiments: compare text representations, hARTM and doc2vec classification, topic modeling of communities, users, users images text, sensitivity to infer new texts to model, typical topic modeling scores for randomly generated text, 20newsgroups classification task (Fig. 3), hARTM+doc2vec clustering and classification on datasets.

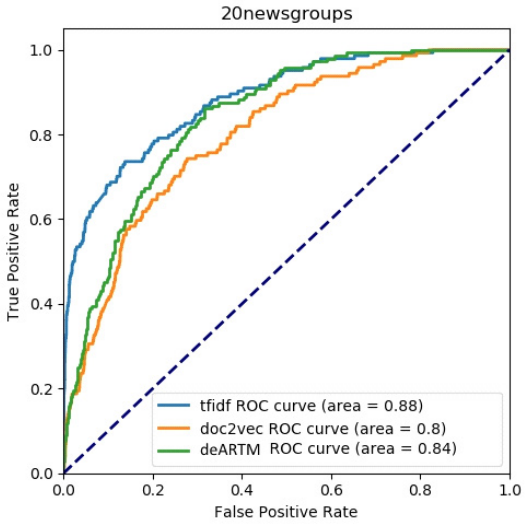

Fig. 3. 20newsgroups ROC curves

There is a natural idea that each documents has different abstraction levels (contexts) at the same moment. A given text can be devoted to IT in general, software development in a lower level, then web application development, programming high performance applications in Scala and so on. On each levels, we have top tokens related the specific domain. The theta matrix shows hierarchic topics representations for each documents (Fig. 4), where by horizontal axe are topics and by vertical line are documents. As can be seen, the first columns are most significant topics while the majority topics has fewer weights. Each document consists from several topics, and there are documents almost related to only one topic.

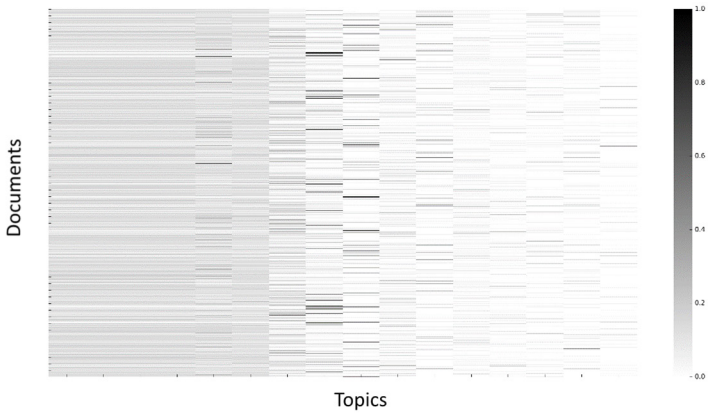

Fig. 4. Theta matrix with stacking hierarchic layers of hARTM

Tables 2, 3 show scores for wikinews datasets for cases 1gram and 2gram vocabularies. First column shows count

of topics used in hierarchical layer ARTM. As can be seen, the perplexity decreases with topic numbers growth, in opposite to the coherence. The model reached the lower perplexity in unigram case. The proper coherence estimation for bigram tokens is an open question because it requires converting unigram text tokens to bigram with saving a word order. In these tables, level is a number of topics in each level, and other columns are scores per level. There number of topics on level is an empirical value; there is no way to strictly calculate it. We used the number of topics in the range from 2 to 1024 with a step of 2^n.

Table 2. Wikinews unigram hARTM scores

Topics	Coherence	Contrast	Perplexity	Purity	Sparsity phi	Sparsity theta
2	0,217	0,799	649,821	0,924	0,094	0,009
4	0,263	0,665	998,178	0,768	0,196	0,018
8	0,351	0,611	998,930	0,622	0,363	0,032
16	0,386	0,543	845,732	0,490	0,476	0,058
32	0,362	0,493	666,563	0,373	0,594	0,103

Table 3. Wikinews bigram hARTM scores

Topics	Coherence	Contrast	Perplexity	Purity	Sparsity phi	Sparsity theta
2	0,406	0,940	193160,750	0,988	0,462	0,028
8	0,462	0,703	120043,844	0,837	0,652	0,133
16	0,466	0,540	65701,281	0,742	0,818	0,259
32	0,491	0,320	50326,613	0,357	0,908	0,427

Table 4 presents scores for randomly generated text. This result shows the mean coherence scores for wikinews dataset and random dataset differ in 0.025 (Table 3, 4). Therefore, we concluded these scores could not help to divide fake texts from valid ones. It could be useful for text obtaining after image recognition. Another test experiment is a generating topic by choosing random tokens from a vocabulary. In this case, frequent tokens are most significant tokens in each topic. Thus, these scores do not always reflect a topics interpretability in a human view. That is a reason to add assessors' estimation of topic interpretability.

Table 4. Random text scores

Topics	Coherence	Contrast	Perplexity	Purity	Sparsity phi	Sparsity theta
2	0,461	0,955	3584,824	0,987	0,470	0,059
8	0,444	0,644	2559,050	0,602	0,690	0,111
16	0,444	0,524	1754,746	0,508	0,854	0,136
32	0,373	0,319	1381,045	0,254	0,914	0,193

We compared LDA, ARTM with doc2vec for predicting topic of new documents. We add biology related texts (from postnauka) to models trained on social networks data. In this case, we calculate the percentage of intruded documents joined to one group. Probabilistic topic models need to use known words in new documents to infer topics, in opposite them doc2vec, fastText models. The existing embedding represents new documents in space of old documents vectors. Topic model classifies new document using only known tokens, but if these words are just a stop words the model was confused. The experiment shows that document embedding has better results for novelty

detection.

The results of applying our hARTM combined model for topic modeling show increasing average coherence metric in 0.25 (Table 5). hARTM and doc2vec models had similar scores. However, doc2vec had noisier top tokens, and hARTM model had less score in classification task.

Table 5. Social media vkposts scores

Topics	Coherence	Contrast	Perplexity	Purity	Sparsity phi
8	0,65	0,57	2305,42	0,59	0,25
32	0,66	0,49	2802,27	0,26	0,28
64	0,64	0,41	2713,10	0,16	0,47
70	0,65	0,41	2625,76	0,16	0,57

For images text datasets, we got topics with OCR artifacts (not meaningful words and characters), which helped to filter these errors documents from valid ones. Interestingly, that we got clear recognized texts from published mobile screenshots (vkposts from twitter usually). It seems in social media, there is a trend to publish screenshots. By using hierarchic topic models hARTM, we estimated relations between parent and child topics. Thus, hierarchic models improve topic interpretability than single ARTM. Experiments with 20newsgroups show that topic models representation can be used for solving classification problem.

To estimate model results we use perplexity, coherence and assessors score for topic interpretability. As can be seen developed model deARTM has better results on more noisy data (Table 6). The perplexity is a measure of training probabilistic topic model, therefore we do not calculate it for doc2vec. The improvement of topic models embedding is reached by using doc2vec and vectors from hierarchic layers, because it makes the model less sensitive to words spelling. We used the same deARTM representation for classification problem.

Table 6. Models scores

Dataset	Perplexity	Coherence	Topics interpretability
Vkposts			
hARTM	2918	0,62	4
doc2vec	-	0,71	2
deARTM	36	0,62	4
vkusers img			
ARTM	16334	0,52	3
doc2vec	-	0,74	2
deARTM	7109	0,61	4
Vkmusic			
ARTM	910	0,43	5
doc2vec	-	0,73	2
deARTM	222	0,67	5

6. Discussions

In this way, our results show that deARTM can be used for clustering of social media texts. hARTM improves topics interpretably and document embedding helps in classification tasks. We test our approach on classic 20newsgroups datasets for solving classification problem. Experiment shows that deARTM in comparison with hARTM has the same scores in classification task on 20newsgroups datasets (Fig. 3) and better scores for topic modeling on vkposts datasets (Table 6). A weak point of our model is a using distinct optimization problems (7), (8). We will consider a fully consistent single model in the future.

7. Conclusions

In this paper, we proposed an approach for topic modeling applied to social media. The text preparation with 2gram, td-idf weights is more efficient. The lemmatization does not give sufficient improvement, since there are many misspellings in social media. The coherence is a tricky measure of topic model, it should be used in couple with the perplexity and an assessors estimation. Document embedding model is more reliable for classification of documents and novelty detection, but it needs special clustering procedure to obtain more interpretable topics, although it has higher coherence score. The developed model deARTM (document embedding ARTM) based on a hierarchical ARTM model and document embedding. This model follows the trend of combined topic models. It improves results for noisy texts (vkposts, vkimages). The considered approach allowed decreasing perplexity and increase topic tokens coherence, and expand topic tokens via document embedding.

As future work, we plan to further study combining probabilistic topic representation and document embedding. It is interesting to try prebuilded topic embedding to improve modeling on inconsistent data. In addition, we are planning to test our model with additional temporal feature for topic detection and tracking problem and implement auto topic naming. Another challenge is how to filter significant words not by only term frequency or co-occurrence in sentences but by meaning in a human view.

Acknowledgements

This research is financially supported by The Russian Science Foundation, Agreement #14-21-00137.

References

[1] I. Guy, U. Avraham, D. Carmel, S. Ur, M. Jacovi, and I. Ronen, “Mining expertise and interests from social media,” in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 515–526.

[2] J. Leskovec, L. Backstrom, and J. Kleinberg, “Meme-tracking and the dynamics of the news cycle,” in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 497–506.

[3] S. Liu and P. Jansson, “Topic Modelling Analysis of Instagram Data for the Greater Helsinki Region,” 2017.

[4] S. Stieglitz, M. Mirbabaie, B. Ross, and C. Neuberger, “Social media analytics--Challenges in topic discovery, data collection, and data preparation,” *Int. J. Inf. Manage.*, vol. 39, pp. 156–168, 2018.

[5] M. Schinas, S. Papadopoulos, Y. Kompatsiaris, and P. A. Mitkas, “Visual event summarization on social media using topic modelling and graph-based ranking algorithms,” in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, 2015, pp. 203–210.

[6] T. Hofmann, “Probabilistic latent semantic analysis,” in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, 1999, pp. 289–296.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *J. Mach. Learn. Res.*, vol. 3, no. Jan, pp. 993–1022, 2003.

[8] K. Vorontsov and A. Potapenko, “Additive regularization of topic models,” *Mach. Learn.*, vol. 101, no. 1–3, pp. 303–323, 2015.

[9] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin, “Automatic evaluation of topic coherence,” in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010, pp. 100–108.

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.
[11] "Global vectors for word representations." [Online]. Available: https://nlp.stanford.edu/projects/glove/.
[12] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International Conference on Machine Learning*, 2014, pp. 1188–1196.
[13] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv Prepr. arXiv1607.04606*, 2016.
[14] L. Wu, A. Fisch, S. Chopra, K. Adams, A. Bordes, and J. Weston, "StarSpace: Embed All The Things!," *arXiv Prepr. arXiv1709.03856*, 2017.
[15] W. Y. Zou, R. Socher, D. Cer, and C. D. Manning, "Bilingual word embeddings for phrase-based machine translation," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1393–1398.
[16] S. Bengio and G. Heigold, "Word embeddings for speech recognition," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.
[17] J. Saxe and K. Berlin, "eXpose: A Character-Level Convolutional Neural Network with Embeddings For Detecting Malicious URLs, File Paths and Registry Keys," *arXiv Prepr. arXiv1702.08568*, 2017.
[18] C. E. Moody, "Mixing dirichlet topic models and word embeddings to make lda2vec," *arXiv Prepr. arXiv1605.02019*, 2016.
[19] Y. Liu, Z. Liu, T.-S. Chua, and M. Sun, "Topical Word Embeddings.," in *AAAI*, 2015, pp. 2418–2424.
[20] A. Potapenko, A. Popov, and K. Vorontsov, "Interpretable probabilistic embeddings: bridging the gap between topic models and neural networks," in *Conference on Artificial Intelligence and Natural Language*, 2017, pp. 167–180.
[21] M. Röder, A. Both, and A. Hinneburg, "Exploring the space of topic coherence measures," in *Proceedings of the eighth ACM international conference on Web search and data mining*, 2015, pp. 399–408.
[22] I. C. Mogotsi, "Christopher d. manning, prabhakar raghavan, and hinrich schütze: Introduction to information retrieval." Springer, 2010.
[23] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv Prepr. arXiv1607.01759*, 2016.