Approximation of the meaning for thematic subject headings by simple interpretable representations

Rodion Sulzhenko[1] and Boris Dobrov[2]

[1] Lomonosov Moscow State University, Moscow, Russia rodion13par@gmail.com
[2] Lomonosov Moscow State University, Moscow, Russia dobrov_bv@mail.ru

Abstract. The paper studies methods for approximating a user labeled topics by simple representations in a text classification problem. It is assumed that in real information systems the meaning of thematic categories can be approximated by a fairly simple interpreted expression. An algorithm for constructing formulas is considered, which constructs a representation of a text topic in the form of a Boolean formula – in fact, a request to a full-text information system. The algorithm is based on an optimized selection of various logical predicates with words and terms from the thesaurus. The presented algorithm has been compared with modern machine learning techniques on real collections with noisy expert markup. The described method can be used for text classification, expert evaluation of the content of the heading, assessment of the complexity of the description of the topic, and correcting the markup.

Keywords: Multi-label Text Classification · Interpretable machine learning · Inconsistent markup.

1 Introduction

In real services, the task of classifying (categorizing) texts often arises, namely, assigning the text to one or more specified categories. The example of tasks are assigning news articles to certain categories, assigning texts to certain tags. Modern machine learning methods demonstrate quite good results on similar tasks with a small number of headings. However, on datasets with a large number of categories and with a small number of examples, its quality deteriorates [1].

It is believed that classification errors are caused by the imperfection of the machine learning algorithm itself, namely its inability to describe the training dataset well enough. In this case, the researchers move on to a more capable algorithm with a large number of parameters.

Note that for the text classification the result also depends on the training set. Often the training set is manually marked up by special people – the so-called assessors. With a large number of categories, the task of marking up a set of texts into headings becomes difficult for assessors. Due to these prerequisites, the markup itself can cause errors. Therefore, the creation of a consistent training set

becomes an urgent task. During the process of marking assessors face a number of problems.

Firstly, the categories may be similar or may have some hierarchical structure where one category may include another. Such categories are quite difficult to distinguish without expert knowledge in a specific subject area.

Secondly, depending on the subject area, the number of categories can vary up to several hundred. It is problematic to find an assessor who would understand the details of each of them.

Thirdly, the markup process is complicated by the formulation of the multi--label classification problem, when the text may not necessarily belong to one category, but to many at once. Machine learning algorithms do not cope with this task so well and researchers have to reduce it to a series of binary classification problems. Assessors determine the importance of topics in relation to the content of the text in different ways. There are likely situations when experts agree about the main category of the text, but disagree about auxiliary (side) categories.

In connection with the described prerequisites, the problem of automatic markup verification for adequacy and consistency is relevant.

For such verification of the markup for consistency, it is proposed to develop a method that would represent each topic in the form of interpreted formal description (formula), simple representations. Further, based on the interpretation of this formula, it is possible to draw conclusions about whether the topic corresponds to the documents that were assigned to it according to the markup.

We assume that each heading corresponds a certain section of the subject area, which can be represented by a small verbal description.

2 Related Work

It is logical to assume that many topics can be described by keywords. Based on this assumption, the article [2] suggests using formulas of the form:

$$D_c = \bigcup_{i}^{N} \bigcap_{j}^{M_i} l_{ij}, \tag{1}$$

where

- $M_i = \{1, 2, 3\}$ (each conjunction consists of a maximum of three elements),
- l_{ij} – lemmas from the document in normal form.

The article presents an algorithm for constructing such formulas – abbreviated "FCA". In addition, possible extensions of formulas by adding Boolean negations are considered. This algorithm was tested on the Reuters-21578 [4] dataset and achieved metrics comparable to SVM. The research of one of the authors of this article provides a detailed overview of this algorithm, proves its convergence, introduces quality metrics, and presents the results of numerical experiments with this algorithm on real datasets.

A similar method was proposed in the article [3]. It also used logical expressions based on words from the document. However, simpler formulas were used,

which had less generalizing ability. Formulas were based on the induction of logical rules. Unlike FCA, the method from the article [3] was originally proposed with a slightly different purpose – to present an interpreted method for text classification.

3 Datasets

Although the authors of the FCA algorithm tested it on the Reuters [4] dataset, this dataset was not enough to fully demonstrate the problem investigated in this paper – the difficulty of creating a consistent collection for classification.

It was decided to collect a dataset of texts related to a more extensive and complex subject area – IT. For these purposes, a dataset was collected from the articles of the Internet portal Habr.com. On this site, each of the articles refers to some tags (so-called "hubs"), which in this case we will consider headings (topics). Hubs are also divided into so-called profile hubs (directly related to IT) and non-profile hubs.

The dataset was assembled as follows: according to the list of hubs on Habr.com/ru/hubs/ the first 20 profile hubs with the largest number of subscribers were selected. Further, 500 articles were collected for each hub. Only articles with a positive rating were selected. Thus, ∼7,500 articles were collected, since some articles belonged to several hubs out of 20 selected at once.

Further, the dataset was divided into a training and test part in the proportion of 80:20 – i.e. ∼6000 articles in the training set and ∼1500 articles in the test set. The distribution of classes in the training and test set is approximately the same.

The selected topics are presented in the table below.

Topic	Total docs
Programming	1523
Web development	900
Information security	705
Python	689
Machine learning	670
Javascript	647
Mobile development	572
Interfaces	570
Big data	546
Algorithms	545
Highload	506
Java	498
Debug	492
Android development	485
Complete code	478
PHP	470

Linux	468
Web design	464
IT-standarts	456
Data mining	446

Table 1: Topic distribution in Habr dataset articles

Publications on the Habr.com portal are divided into "articles" and "news". It was noticed that the articles on average are much longer than the news. For some of the methods tested in this paper, the length of the text was a significant limitation. Moreover, there were reasons to believe that shorter texts are more meaningful and representative for this study. In this regard, a dataset of news from the Habr.com portal was collected using the same logic as in the previous paragraph. Collected datasets are published on github [3].

Topic	Total docs
IT-companies	2046
Business laws	722
Information security	633
Finance	616
Social networks	545
Artificial intelligence	544
Soft	502
Smartphones	486
Popular science	481
Machine learning	435
Gadgets	430
Cloud services	427
Open source	407
Mobile development	319
Android development	278
Video	276
Programming	227
Algorithms	159
Web development	132
Big data	128

Table 2: Topic distribution in Habr dataset news

[3] https://github.com/rodion-s/FCA

4 Overview of proposed methods

4.1 Formula Constructing Algorithm

The formula constructing algorithm (FCA) [2] generates formulas of the form (1)

$$D_c = \bigcup_i^N \bigcap_j^{M_i} l_{ij}, \tag{2}$$

where

- $M_i = \{1, 2, 3\}$ (each conjunction consists of a maximum of three elements),
- l_{ij} – lemmas from the document in normal form.

The idea of the algorithm is to iteratively expand the formula with new conjuncts that maximize the F-measure of docs from the topic covered by formula.

A more formal description of the algorithm consists of three main steps.

1. **Compilation of the first conjunction**: for each word found in the documents of the heading (as well as combinations of two and three words), the *F-beta* measure obtained when describing the topic by this word (combination of words) is calculated. *Beta* must be greater than one.

$$F_{beta} = \frac{1}{\frac{\beta}{\text{prec}} + \frac{1}{\text{recl}}} \tag{3}$$

$$\begin{aligned} ConjList = &\{w \mid w \in C\} \cup \{(w_1, w_2) \mid w_1, w_2 \in C\} \cup \\ &\cup \{(w_1, w_2, w_3) \mid w_1, w_2, w_3 \in C\} \end{aligned} \tag{4}$$

$$ConjFirst = argmax_{ConjList} F_{beta} \tag{5}$$

$$Formula(d) := ConjFirst(d) \tag{6}$$

2. **Formula extension**: consider the remaining elementary conjunctions from the list, calculate for them the so-called complementary precision (*addprec*) and complementary recall (*addrecl*). Choose the best conjunction for maximizing $F(conj)$.

$$cntr = |\{doc \mid doc \in C\}| \tag{7}$$

$$cntfr = |\{doc \mid Formula(doc) = True, \quad doc \in C\}| \tag{8}$$

$$addf(conj) = |\{doc \mid Formula(doc) = False,\ conj \in doc\}| \tag{9}$$

$$addfr(conj) = |\{doc \mid Formula(doc) = False,\ conj \in doc,\ doc \in C\}| \tag{10}$$

$$addprec(conj) = \frac{addfr(conj)}{addf(conj)} \times 100 \tag{11}$$

$$addrecl(conj) = \frac{addfr(conj)}{cntr - cntfr} \times 100 \tag{12}$$

$$F(conj) = \frac{1}{\frac{\alpha_1}{recl} + \frac{\alpha_2}{addprec(conj)} + \frac{\alpha_3}{addrecl(conj)}} \tag{13}$$

$$NextConj = argmax_{conj} F(conj) \tag{14}$$

$$Formula := Formula \cup NextConj \tag{15}$$

3. Repeat the previous step until one of the following stop conditions is met:

- the value of *addrecl* for the best conjunct is zero (there is no recall improvement);
- the number of conjuncts in the formula has reached the limit;
- $prec < 10$ and $recl > 90$ (too little precision);
- $recl > 99$ (sufficient result is achieved).

Note that $\alpha_1, \alpha_2, \alpha_3, \beta$ are parameters of the algorithm.

Thus, FCA is looking for a tradeoff between precision and recall regarding the topic. The first conjunct sets the initial precision of the formula, and subsequent conjuncts increase the recall. To find the final formula that combines the optimal ratio of precision and recall, it is necessary to truncate the resulting formula to the point where the maximum of the F-measure is reached.

4.2 Formula Constructing Algorithm extension using thesaurus

The formula constructing algorithm does not take into account the word order, context, and frequency of words. This paper proposes the modernization of the algorithm using a thesaurus.

Note that the basic elements of the formula l_{ij} in (1) may not necessarily be lemmas (words) from the text. For instance, terms from some thesaurus can be used as basic elements. The thesaurus RuThes [7, 8] was used for experiments with Habr dataset. An example of the RuThes hierarchy is shown in Fig. 1 in the appendices. The following types of thesaurus entities were parsed from the texts:

- *LEM* – lemma,
- *TERM* – term from thesaurus.

Each entity in the document has been assigned a weight, reflecting its importance, frequency. It is proposed to fix a numerical threshold for each type of entities from the thesaurus (*LEM/TERM*). If the weight of the occurrence of a particular entity in a particular document is less than the threshold set for this type of entity, this occurrence is not taken into account when constructing formulas.

5 Experimental results

5.1 Basic

To measure the quality of the proposed methods, several basic experiments with well-known architectures were conducted on Habr datasets.

SVM. Only the text was left in the articles, the words were lemmatized, the stop-words were removed. TF-IDF was used as a feature space over words, words with a document frequency < 5 were truncated. The SVM classifier [5] was trained to distinguish each category from all the others according to the one-versus-all technique.

BERT. The BERT [6] neural network model was used. The source text of the article included in the first 512 tokens (about $250 - 300$ words) was used as a feature space, since the length of the input sequence for most BERT-like architectures is limited to 512 tokens. The pre-trained LaBSE [10] model was used for the experiments. This model was trained for multi-label text classification on Habr datasets.

In addition, a similar experiment was conducted on a slightly different feature space – 384 tokens were taken from the beginning of the article and 128 tokens from the end of the article. The motivation of this modernization is based the fact that usually the main meaning of the article, which is important for determining the topic, is concentrated precisely at the beginning and at the end of the article. This approach allowed us to give the model more information about the subject of the text compared to the experiment, which uses 512 tokens from the beginning of the article.

FCA. For this experiment, as for the SVM, only the text was left in the articles, the words were lemmatized, the stop words were removed. Words with a document frequency < 5 were truncated.

The PFA algorithm was implemented in Python. The formulas were constructed on the training part of the dataset of Habr articles, the quality was measured on the test part of the dataset. The following algorithm parameters were used: $\beta = 5$, $\alpha_1 = 1$, $\alpha_2 = 10$, $\alpha_3 = 5$.

FCA (RuThes). The key experiments were conducted with an upgraded version of the FCA algorithm based on the RuThes thesaurus according to the description in Section 4.2. Several experiments were conducted with different combinations of the parameters of the weights *LEM, TERM*. In order to speed up the selection of parameters, several thresholds for entity weights were passed in these experiments (20, 40, 60, 80). The threshold of 20 corresponds to almost any word, 40 to a local topic, 60 to the "center" of a local or main topic, 80 to the "center" of the main topic [9]. The best results were achieved with the

parameters: entity *LEM* with a threshold of 20, entity *TERM* with a threshold of 60.

Below are results on the test part of the dataset:

Model	F1-macro
SVM	0.6783
BERT (LaBSE)	0.6823
BERT (LaBSE**)	0.6943
FCA	0.5447
FCA (RuThes)	0.5711

Table 3. Results on Habr article dataset

Model	F1-macro
SVM	0.67
BERT (LaBSE)	0.6563
BERT (LaBSE**)	0.6722
FCA	0.51
FCA (RuThes)	0.5484

Table 4. Results on Habr news dataset

BERT (LaBSE**) – the LaBSE model, 384 tokens were taken from the beginning of the document and 128 tokens were taken from the end of the document.

5.2 Formula analysis

Below are examples of formulas constructed by the FCA algorithm using the RuThes ontology on the Habr article dataset (enities of RuThes thesaurus were translated to English).

Topic	Formula	F-1 train (a) – FCA (RuThes) (b) – LaBSE**	F-1 test (a) – FCA (RuThes) (b) – LaBSE**
Android development	(/LEM="ANDROID")	(a) 0.7076 (b) 0.9946	(a) 0.7086 (b) 0.8055
Data mining	(/LEM="R") or (/LEM="ANALYSIS" and /TERM="DATA (INFORMATION)") or (/TERM="DATA ANALYSIS") or (/TERM="DATA (INFORMATION)" and /TERM="MACHINE LEARNING" and /LEM="LEARNING")	(a) 0.4059 (b) 0.9932	(a) 0.4229 (b) 0.5399

Debug	(/LEM="DEBUG" and /TERM="DEBUGGING THE PROGRAM") or (/LEM="ERROR" and /TERM="BUG") or (/TERM="DEBUGGER PROGRAM")	(a) 0.5296 (b) 1.0	(a) 0.5333 (b) 0.6458
Information security	(/LEM="ATTACK") or (/LEM="SECURITY") or (/LEM="VULNERABILITY") or (/TERM="ATTACK, MILITARY STRIKE") or (/TERM="COMPUTER ATTACK") or (/TERM="INTRUDER") or (/TERM="VULNERABILITY TO HACKER ATTACK")	(a) 0.6234 (b) 0.9964	(a) 0.6585 (b) 0.7619
PHP	(/LEM="LARAVEL") or (/LEM="PHP") or (/TERM="PHP (SCRIPT LANGUAGE)")	(a) 0.8000 (b) 0.9911	(a) 0.8070 (b) 0.8525
Python	(/LEM="PYTHON") or (/TERM="PYTHON")	(a) 0.6085 (b) 0.9978	(a) 0.6080 (b) 0.6932
Java	(/LEM="JAVA") or (/LEM="SPRING") or (/TERM="JAVA")	(a) 0.7060 (b) 0.1	(a) 0.6667 (b) 0.7933
Javascript	(/LEM="J") or (/LEM="JAVASCRIPT") or (/LEM="REACT")	(a) 0.5998 (b) 0.9916	(a) 0.6029 (b) 0.7414
Machine learning	(/LEM="MODEL" and /TERM="TRAINING, EDUCATIONAL ACTIVITIES") or (/LEM="TRAINING") or (/TERM="DATA (INFORMATION)" and /LEM="MODEL") or (/TERM="MACHINE LEARNING") or (/TERM="NEURAL NETWORKS")	(a) 0.6765 (b) 0.9986	(a) 0.6736 (b) 0.7655

Table 5: Examples of formulas for Habr article dataset

Below are formulas constructed by the FCA algorithm using the ontology of RuThes on the dataset *news* Habr.

Topic	Formula	f1 train (a) – FCA (RuThes) (b) – LaBSE**	f1 test (a) – FCA (RuThes) (b) – LaBSE**
Big data	(/LEM="DATA") or (/LEM="DATA" and /TERM="ANALYSIS (REVIEW)") or (/TERM="BIG DATA") or (/TERM="DATA SCIENCE" and /LEM="DATUM") or (/TERM="ANALYSIS (REVIEW)" and /LEM="DATA" and /LEM="PROBLEM")	(a) 0.6049 (b) 1.0	(a) 0.5000 (b) 0.6133
IT-companies	(/LEM="2022") or (/LEM="COMPANY" and /TERM="COMPANY (ORGANIZATION)") or (/LEM="RUSSIAN") or (/LEM="SERVICE") or (/TERM="COMPANY (ORGANIZATION)" and /TERM="RUSSIAN FEDERATION")	(a) 0.6389 (b) 1.0	(a) 0.6428 (b) 0.7478

Information security	(/LEM="ATTACK") or (/LEM="SECURITY" and /TERM="INFORMATION SECURITY") or (/LEM="INTRUDER") or (/LEM="VULNERABILITY") or (/LEM="HACKER") or (/TERM="COMPUTER ATTACK") or (/TERM="COMPUTER HACKER") or (/TERM="ATTACK, COMMIT AN ATTACK") or (/TERM="CRIMINAL") or (/TERM="INFORMATION SECURITY SPECIALIST") or (/TERM="INFORMATION LEAK") or (/TERM="VULNERABILITY TO HACKER ATTACK")	(a) 0.6733 (b) 0.9977	(a) 0.6601 (b) 0.7601
Artificial intelligence	(/LEM="AI" and /TERM="ARTIFICIAL INTELLIGENCE") or (/LEM="INTELLIGENCE") or (/LEM="ARTIFICIAL") or (/TERM="NEURAL NETWORKS")	(a) 0.7392 (b) 1.0	(a) 0.7131 (b) 0.7705
Machine learning	(/LEM="MACHINE") or (/LEM="MODEL" and /LEM="TRAINING") or (/TERM="ARTIFICIAL INTELLIGENCE" and /LEM="TRAINING") or (/TERM="MACHINE LEARNING" and /LEM="MODEL") or (/TERM="MACHINE LEARNING" and /LEM="TRAINING") or (/TERM="MACHINE LEARNING" and /TERM="NEURAL NETWORKS") or (/TERM="MACHINE LEARNING" and /TERM="HUMAN")	(a) 0.6675 (b) 0.9983	(a) 0.7034 (b) 0.7157
Smarthones	(/LEM="SMARTPHONE" and /LEM="DEVICE") or (/LEM="SMARTPHONE" and /TERM="SMARTPHONE" and /TERM="TECHNICAL DEVICE") or (/TERM="COMPANY (ORGANIZATION)" and /LEM="SMARTPHONE" and /TERM="SMARTPHONE") or (/TERM="PROSPECTUS" and /LEM="SMARTPHONE" and /TERM="SMARTPHONE")	(a) 0.7000 (b) 1.0	(a) 0.6969 (b) 0.7697
Popular science	(/LEM="COSMIC") or (/LEM="SCIENTIST") or (/TERM="AMERICAN SPACE AGENCY") or (/TERM="SPACE ORBIT") or (/TERM="EARTH")	(a) 0.7240 (b) 1.0	(a) 0.7440 (b) 0.8099

Table 6: Examples of formulas for Habr article dataset

Note that the FCA creates logical and interpretable formulas that reflect the topic. The quality on the training and testing set is similar, there is no effect of overfitting. For many headings, the formulas turned out to be short and succinct. Formal metrics for PFA are inferior to LaBSE, the difference is $\approx 10\%$. Note that the metrics for LaBSE are also low, which indicates contradictions in the markup.

5.3 Label correction

The publications on the Habr portal are usually categorized by the users themselves. Moderators can also change the topics. During the process of analyzing the errors of the FCA algorithm, the hypothesis about the inconsistency of the original markup of the Habr dataset was confirmed. It was decided to conduct an experiment with the re-marking of contradictory documents. The document was considered controversial if the predicted topics on it differed from the true markup. Only type I and type II errors were subject to re-marking. The re-marking was carried out by one of the authors of the article using a visualization tool that displays errors of the type I and type II errors of the methods (see Fig. 2 in the appendices). The re-marking was made for three topics of the Habr news dataset – "Information security", "Machine learning" and "Artificial intelligence". The number of changed labels is shown in the table below.

Topic	Total labels	Corrected labels
Information security	633	205
Machine learning	435	211
Artificial intelligence	544	176

Table 7. Results of re-marking Habr news dataset

The results for each category are presented in the tables below. The numbers in the tables mean the F1-score on the test part of the dataset for the corresponding topic.

Markup / Method	SVM	LaBSE**	FCA (RuThes)
Source markup	0.7529	0.7692	0.6601
Corrected markup	0.8065	0.8225	0.8088
Corrected markup* (training on source markup)	0.7936	0.7743	0.8164

Table 8: Results of the re-marking of the topic "Information security"

Markup / Method	SVM	LaBSE**	FCA (RuThes)
Source markup	0.6975	0.687	0.7034
Corrected markup	0.8427	0.8541	0.878
Corrected markup* (training on source markup)	0.7937	0.7871	0.9234

Table 9: Results of the re-marking of the topic "Machine learning"

Markup / Method	SVM	LaBSE**	FCA (RuThes)
Source markup	0.7352	0.7429	0.7131
Corrected markup	0.8543	0.843	0.8125
Corrected markup* (training on source markup))	0.7936	0.772	0.8847

Table 10: Results of the re-marking of the topic "Artificial intelligence"

Topic	Formula on source markup	Formula on corrected markup
Information security	(/LEM="ATTACK") or (/LEM="SECURITY" and /TERM="INFORMATION SECURITY") or (/LEM="INTRUDER") or (/LEM="VULNERABILITY") or (/LEM="HACKER") or (/TERM="COMPUTER ATTACK") or (/TERM="COMPUTER HACKER") or (/TERM="ATTACK, COMMIT AN ATTACK") or (/TERM="CRIMINAL") or (/TERM="INFORMATION SECURITY SPECIALIST") or (/TERM="INFORMATION LEAK") or (/TERM="VULNERABILITY TO HACKER ATTACK")	(/TERM="CRIMINAL") or (/LEM="ATTACK") or (/LEM="VULNERABILITY") or (/TERM="VULNERABILITY TO HACKER ATTACK) or (/TERM="INFORMATION SECURITY SPECIALIST) or (/TERM="INFORMATION LEAK") or (/TERM="COMPUTER HACKER) or (/LEM="INTRUDER")
Machine learning	(/LEM="MACHINE") or (/LEM="MODEL" and /LEM="TRAINING") or (/TERM="ARTIFICIAL INTELLIGENCE" and /LEM="TRAINING") or (/TERM="MACHINE LEARNING" and /LEM="MODEL") or (/TERM="MACHINE LEARNING" and /LEM="TRAINING") or (/TERM="MACHINE LEARNING" and /TERM="NEURAL NETWORKS") or (/TERM="MACHINE LEARNING" and /TERM="HUMAN")	(/TERM="NEURAL NETWORKS") or (/TERM="MACHINE LEARNING" and /LEM="TRAINING") or (/TERM="MACHINE LEARNING" and /LEM="ARTIFICIAL INTELLIGENCE ") or (/TERM="MACHINE LEARNING" and /LEM="HUMAN")
Artificial intelligence	(/LEM="AI" and /TERM="ARTIFICIAL INTELLIGENCE") or (/LEM="INTELLIGENCE") or (/LEM="ARTIFICIAL") or (/TERM="NEURAL NETWORKS")	(/TERM="ARTIFICIAL INTELLIGENCE")

After the markup was corrected, the quality of all methods increased, the FCA method quality is not inferior to LaBSE. The PFA formulas became shorter and more logical, which indicates that the markup contradictions were partially eliminated in the dataset.

Methods trained on the old markup, but tested on the corrected one, behave differently. The quality of the methods on the corrected markup is higher than on the original one, despite the fact that they were trained on the original one. In other words, the predictions of the methods correspond more to the corrected markup than to the original one. This indicates that the methods tend to correct random and non-system markup errors. Moreover, in this case, the FCA method is superior to other methods, that is, it is more resistant to errors in the training set. Thus, the FCA method can be used as a tool for finding errors in markup.

6 Conclusion

The paper explores the possibilities and limitations of the interpreted text classification method – the method for constructing logical formulas (FCA) over selected text objects by a sufficiently large subject headings in a complex and wide domain (collections of texts on information technology topics from the habr.com portal).

The approach is based on the assumption that the meaning of a category can be expressed by a fairly simple formula of simple meanings (words or ontology concepts – as set of synonymical word expressions).

The presence of explicitly interpreted rules can be useful for analyzing the quality of the training set, the meaning of the category content, and simplifying the maintenance of real systems for classifying text streams. In addition, such methods are much simpler and computationally more efficient than methods using large neural network language models.

To assess the achievable quality level of the interpreted FCA method, it was compared with methods like BERT and SVM. Initially, it was assumed that, due to the limited power of the functionality used, methods of the FCA type can lose to methods based on large neural network language models of the BERT type, but the exact parameters had to be clarified.

In the course of the work, it turned out that although the resulting interpretable formulas modeling the meaning of headings looked very logical, their formal metrics on the original markup lagged behind such methods as BERT and SVM, while all machine learning methods did not approximate the original labeling well.

According to the results of the analysis of the quality of expert markup, a significant incompleteness of labeling was revealed. For broad and complex subject areas, it is fundamentally not easy to obtain a representative and consistent training set. The task becomes more complicated if the collection is marked up by the authors of the publications themselves, which increases the subjectivity of the markup.

The collection was partially re-labeled. On the “corrected” training collection, the quality of all machine learning methods significantly increased, and the results of the FCA method were not inferior to methods like BERT and SVM.

Based on the experiments carried out, the following conclusions can be drawn:

– a large deviation of the results of classification by machine learning methods from expert markup may indicate problems with manual markup;
– the increase in the quality of text classification on the “corrected” set suggests that all machine learning methods are “internally” resistant to non-systemic deviations in markup. Thus, all considered machine learning methods (BERT, SVM, FCA) can be used to help experts;
– on the “corrected” labeled set, the results of the considered simple method for constructing Boolean formulas FCA are not inferior in quality to methods like BERT and SVM.

References

1. Dumais S.T., Lewis D.D., Sebastiani F., Report on the Workshop on Operational Text Classification Systems (OTC-02), 2002. http://www.sigir.org/forum/F2002/sebastiani.pdf.
2. Агеев М.С., Добров Б.В., Макаров-Землянский Н.В. Метод машинного обучения, основанный на моделировании логики рубрикатора // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды 5-й Всерос. науч. кон. RCDL’2003 - СПб.: НИИ Химии СПбУ. 2003. С. 150-158. http://rcdl.ru/doc/2003/B2.pdf.
3. Chidanand Apte, Fred Damerau, Automated Learning of Decision Text Categorization, ACM Transactions on Information Systems 12, 3, 233–251, 1994.
4. Reuters-21578 Text Categorization Test Collection. http://www.daviddlewis.com/resources/testcollections/reuters21578/
5. Vapnik V. The Nature of Statistical Learning Theory. – Springer-Verlag – New York, 1995.
6. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. – https://arxiv.org/pdf/1810.04805.pdf
7. Loukachevitch, N., Dobrov, B. (2020). RuThes Thesaurus for Natural Language Processing. The Palgrave Handbook of Digital Russia Studies, 319. https://link.springer.com/content/pdf/10.1007/978-3-030-42855-6_18.pdf
8. Лукашевич Н.В., Добров Б.В., Павлов А.М., Штернов С.В. (2018) Онтологические ресурсы и информационно-аналитическая система в предметной области "Безопасность"/ Онтология проектирования - том 1, № 8, с. 74-95. http://ontology-of-designing.ru/article/2018_1(27)/6_Loukachevitch.pdf
9. Лукашевич Н. В. Тезаурусы в задачах информационного поиска. – 2010. https://istina.msu.ru/download/8944241/1q8q4M:1022DY7tc6zxbUNUs6LlYFujd2c/
10. Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT Sentence Embedding. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Appendices

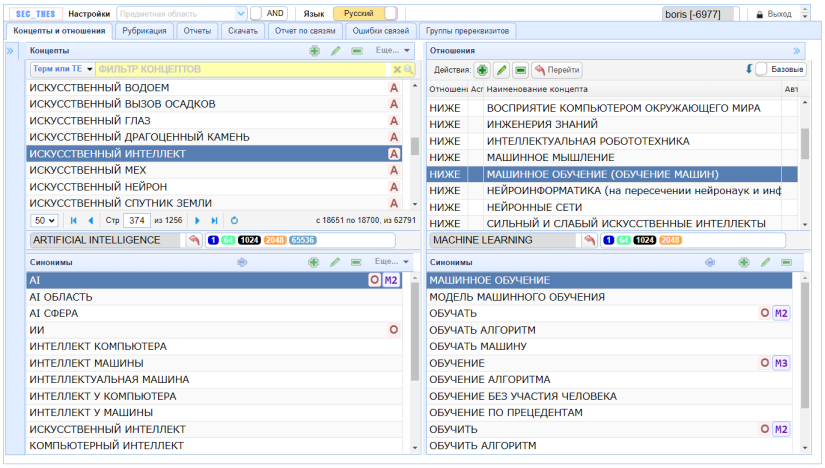

Fig. 1. Example of RuThes thesaurus hierarchy

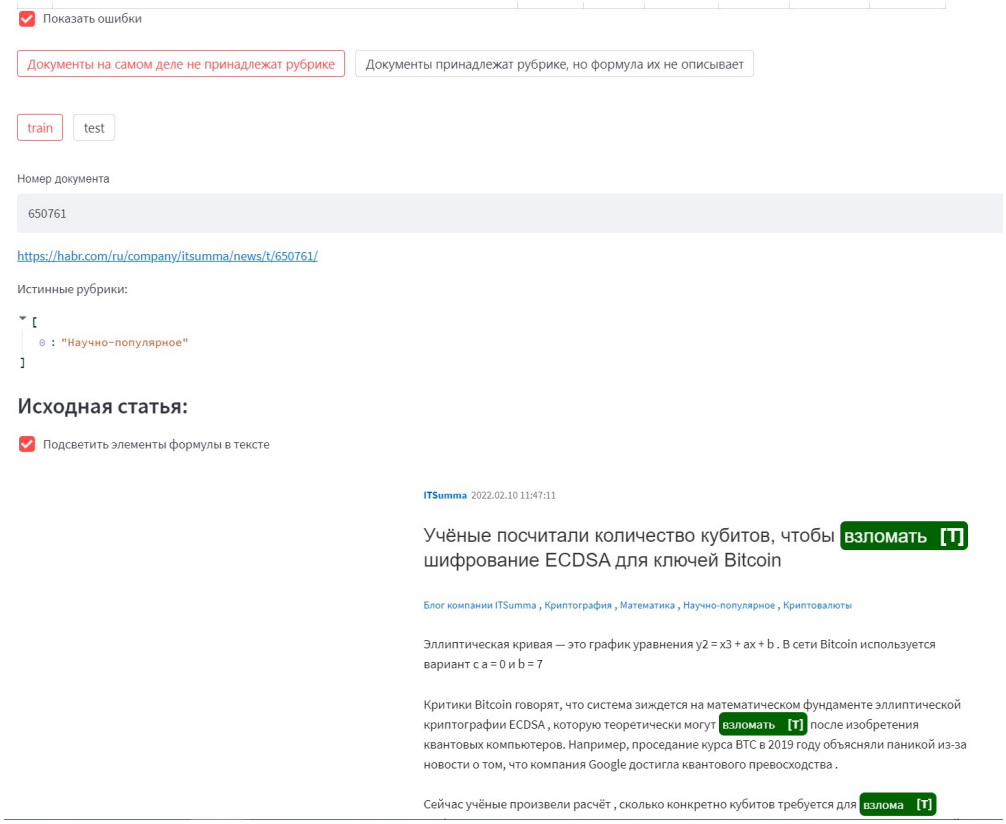

Fig. 2. Visualization tool for displaying misclassified documents

Review answers

——————— REVIEW 1 ———————
SUBMISSION: 60
TITLE: Approximation of the meaning for thematic subject headings by simple interpretable representations
AUTHORS: Rodion Sulzhenko and Boris Dobrov

———— Overall evaluation ————
SCORE: 2 (accept)
—— TEXT:
The paper describes an interesting approach for generation text-representational formulas.

The following information can be useful.

Please provide formula generation algorithm step by step example (probably, for a text 10-20 words long), or a reference to algorithm code on github (in the latter case, readers do not need an example, because they can look in code). If the code cannot be published, then an example can be provided in the paper.

Please provide data set size in bytes and execution time of the algorithm.

In the results, the following terms are shown: ≪DATA (INFORMATION)≫ How the terms with braces are processed? Should the braces be ignored?

Some terms contain stop words, like "TO HACKER ATTACK", "COMMIT AN ATTACK", are this stop words ignored? Probably, there are no stop words in the original Russian terms. Please give a comment.

Question related to Section 5 Experimental results:

Please provide additional comments related to how FCA algorithms are used for classification.

Is the following true?

1) FCA algorithm generates one formula for each topic. In this process, all documents of the specific topic from the training set united in one document, then the formula is generated. Therefore, we have one to one relationship between a topic and a formula.

2) An full-text search is performed for each formula. The search results of the search are considered as connected with the topic that corresponds the formula.

Can a document be associated with several topics?

Answers:

- **Provide or a reference to algorithm code on github.**
  The datasets and the code is now available at https://github.com/rodion-s/FCA. We also added a link in the paper.

- **Please provide data set size in bytes and execution time of the algorithm.**
  Dataset size: ≈5gb raw html, ≈3gb RuThes ontology parsing result, ≈100mb cleaned text without ontology. Execution time: ≈3min per topic.

– **Some terms contain stop words, like "TO HACKER ATTACK", "COMMIT AN ATTACK", are this stop words ignored? Probably, there are no stop words in the original Russian terms. Please give a comment.**
Terms such as ≪DATA (INFORMATION)≫, "TO HACKER ATTACK", "COMMIT AN ATTACK" were extracted from the text using the ontology of RuThes. See 4.2 for details.

– **Please provide additional comments related to how FCA algorithms are used for classification.**
To classify a document, you can apply the resulting formula and check whether the document satisfies it.

– **FCA algorithm generates one formula for each topic. In this process, all documents of the specific topic from the training set united in one document, then the formula is generated. Therefore, we have one to one relationship between a topic and a formula.**
Yes, we have a one to one relationship between a topic and a formula, FCA algorithm generates one formula for each topic.
But documents from the same topic are not combined into one large text. Simplistically, the algorithm searches for words that occur in as many documents belonging to this category as possible. That is, for the words with the highest documentary frequency relative to this category.

– **Can a document be associated with several topics?**
Yes, one document can relate to several topics in our dataset.

——————— REVIEW 2 ———————
SUBMISSION: 60
TITLE: Approximation of the meaning for thematic subject headings by simple interpretable representations
AUTHORS: Rodion Sulzhenko and Boris Dobrov

———— Overall evaluation ————
SCORE: 1 (weak accept)
—— TEXT:
The article is devoted to the method of interpretable thematic classification of texts based on the FSA method. The research topic is relevant.

There are a number of comments. 1. Disclosure of the concept of FCA is not given. "The article presents an algorithm for constructing such formulas – abbreviated "FCA."" Is it Formal Concept Analysis?

2. Literature review (section 2) is too short. There are a number of works that it makes sense to briefly review in the context of the objectives of the study. For example, step forward for Topic Detection on Twitter: An FCA-based approach

Juan Cigarrán , Ángel Castellanos, Ana García-Serrano https://daneshyari.com/article/preview/381960.pdf It is recommended to expand the "related work" section.

3. It would be advisable to open the datasets to other researchers for reproducibility of the results.

4. The study does not answer the question of what the results would have been had if the SVM and the BERT model been fine-tuned more precisely.

Answers:

- **1. Disclosure of the concept of FCA is not given. "The article presents an algorithm for constructing such formulas – abbreviated "FCA."" Is it Formal Concept Analysis? 2. Literature review (section 2) is too short.**
  No, it is not a Formal Concept Analysis. "FCA" is a Formula Constructing Algorithm.
- **3. It would be advisable to open the datasets to other researchers for reproducibility of the results.**
  The datasets and the code is now available at https://github.com/rodions/FCA. We also added a link in the paper.

- **4. The study does not answer the question of what the results would have been had if the SVM and the BERT model been fine-tuned more precisely.**
  It is assumed that optimal hyperparameters have already been selected for all algorithms. Metrics are given for the best results.