Ensemble of classifiers for ontology enrichment

A V Semenova, V M Kureichik

Southern Federal University, Taganrog, Russia, 44, Nekrasovskiy Street, Taganrog, 347900, Russia

E-mail: alexaforum@rambler.ru

Abstract. A classifier is a basis of ontology learning systems. Classification of text documents is used in many applications, such as information retrieval, information extraction, definition of spam. A new ensemble of classifiers based on SVM (a method of support vectors), LSTM (neural network) and word embedding are suggested. An experiment was conducted on open data, which allows us to conclude that the proposed classification method is promising. The implementation of the proposed classifier is performed in the Matlab using the functions of the Text Analytics Toolbox. The principal difference between the proposed ensembles of classifiers is the high quality of classification of data at acceptable time costs.

1. Introduction

Development and implementation of intelligent expert systems are an important direction to increase the reliability and efficiency of technical operation of industrial facilities. One of the state-of-the-art approaches to improve expert systems (ES) is the use of ontologies. Ontology learning and population based on text collection improve efficiency of the ontology development process.

The quality of domain ontologies depends mainly on the completeness of the most significant concepts in the ontological model based on text corpus. So it is obvious to solve the problem of concepts extraction for an ontology. The results of words and phrases searching as well as theirs extraction are not always optimal. Summarizing the results obtained in [1], there are some problems emerged at the stage of ontological models integration: it is not always possible to correctly find relations between concepts; it is not always possible to extract concepts that are related to the largest number of other concepts; extracted relations between the concepts of the ontology are not always relevant for a particular domain, there is an increase in the space of attributes. For such reasons the authors have considered the possibility of eliminating the above-mentioned difficulties by reducing the dimensionality of the feature space by classifying textual information.
Recently, the problem of model precision improvement based on machine learning algorithms has been of great interest: researches combine capabilities of several classifiers and create ensembles of classifiers. Such approach eventually improves the quality of the solution for the problem.

The purpose of the paper is to develop an ensemble of algorithms for the task of ontology populating based on text corpus from the given domain.

2. Backgrounds

Nowadays the problem of automatic text classification is considered quite actively both in Russian and foreign papers [2].

Most commonly researches of text classification problem represent documents as vectors in the form of a bag-of-words model.

So, Andrews and Fox in [3] describe ways of presenting a set of documents in the form of a vector model, including various ways of texts pre-processing, as well as algorithms for clustering them, such as k-means EM-algorithm and spectral clustering. Since one of the main drawbacks of bag-of-words model is high dimension and sparseness of the resulting vectors, the authors also present methods for reducing the dimensionality of a vector space.

In information retrieval, TF-IDF, short for term frequency-inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus [4]. It is often used as a weighting factor in searches of information retrieval, text mining, and user modelling. The TF-IDF value increases proportionally to the number of times a word appears in the document, but is often offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. Nowadays, tf-idf is one of the most popular term-weighting schemes; 83% of text-based recommender systems in the domain of digital libraries use tf-idf.

Considering the partitional document clustering algorithms, in particular k-means, A.Huang presents a description and comparison of similarity measures between bow vectors [6]. The article describes six different similarity measures, between which an experimental comparison is made on the k-means algorithm. The best results for the metrics of purity and entropy were shown by clustering, using the Jaccard coefficient and the Pearson correlation coefficient as a similarity measure. K.Sathiyakumari et al. [7] also consider the clustering of documents only in relation to their representation in the form of a bag-of-words. They distinguish four clustering methods groups: partial clustering, hierarchical clustering, k-means and EM-algorithm. Although in many other works k-means is included in the group of separation algorithms. As can be seen in the above works, the classification of documents is usually reduced to the classification of their vector representations in the form of a bag-of-words model.

Many research papers are considered clustering of vectors in the general case without regard to text documents. A wider range of possible vector representations of the document is dealt with domain of Mining Text Data [8]. In particular, it describes methods that are based on word sets as document features, as well as topic modelling methods. In addition, approaches to online text clustering, the use
of graph clustering methods and the available information for clustering based on semi-supervised algorithms are reviewed.

In some reviews, some authors distinguish methods of semantic classification. G. Marchionini et al. [9] consider the defining difference between semantic clustering from traditional, bag-based words, and use of semantic relations between document words. The authors attribute several groups of algorithms to methods of semantic clustering: algorithms based on ontologies, such as WordNet; algorithms that use a set of related words as document features; as well as algorithms based on graphs of concepts or named entities with semantic relations between them.

Thus, to date, many reviews and experimental comparisons of classification methods have been performed, but most of them do not consider modern methods, for example, word embedding, and also do not take into account peculiarities of scientific articles. One approach to solving the classification problem is to strengthen simple classifiers by combining simple classifiers into one ensemble. The power of classifiers is the efficiency (quality) of the solution of the classification problem [10].

To sum it up, the task of constructing an ensemble of classifiers based on several simple algorithms with different weights and parameters working with unstructured data, is topical.

3. Classification methods

The solution of the classification problem consists of a number of successive steps. The general scheme of the multiclass classification process is shown in Figure 2.

![Figure 2. Stages of the multiclass classification process.](image-url)

Modern methods of machine learning are focused on a feature description. Therefore, text documents are translated into vector representations. In the bag-of-words model, each document is represented as an unordered set of terms [10]:

$$d_i = \{w_{1i}, w_{2i}, \ldots, w_{mi}\},$$

where \(w_{ji}\) is a \(j\)-th term (word) in \(i\)-th documents and \(m\) – total number of different terms in all documents of the collection (dictionary size).

The TF-IDF representation (Term Frequency - Inverse Document Frequency) is the representation of documents in a given space. The parameter IDF (reverse frequency in the case) indicates the general
occurrence of the word in the whole body. The sign value for the n-gramme ti in document dj in this method is calculated by the following formula [10]

$$idf(t_i) \cdot \frac{tf(t_i, d_j) \cdot (k_1 + 1)}{k_1 \cdot \left(1 + b + b \cdot \frac{|d_j|}{d_{avg}}\right) + tf(t_i, d_j)}$$

where $|d_j|$ is the length of the document; $|d_{avg}|$ – the average length of documents in the set; $k_1$ and $b$ are free options.

In Word Embedding methods, the document vector is the average sum of the vector representations of each word. The most popular methods of this class are word2vec [11], GloVe [12] and fastText [13]. The first two methods decompose the matrix of the term-term into two matrices: basic and contextual. Word2Vec uses for this purpose minimization of the log-likelihood taken with a negative sign (minimization of cross-entropy). Word2Vec can also be represented by a three-layer neural network consisting of a linear hidden layer and a softmax output layer, or in the case of a skipgram model, a sigmoid activation function. GloVe is a minimization of the weighted sum of error squares (weighted linear regression).

FastText. Unlike the first two methods, FastText uses additional information about the morphology of a word when constructing a word vector, presenting it (a word) as a sum of alphabetic n-grammes. The advantage of this variant in comparison with the previous one is that the vectors do not depend on the number of documents and have an arbitrary dimension. The disadvantage is the lack of interpretability of the coordinates of the obtained vectors. Interpretability of this option is possible only with respect to the measure of similarity between vectors (most often cosine measure of similarity).

The emergence of word2vec, a family of methods for constructing vector representations of words in word-embedding, has made it possible to reduce the problem of evaluating the semantic similarity of words to calculating the cosine of the angle between vectors of these words. Vectors built on a large unpartitioned text corpus give better results than classical methods based on the distances between words in semantic networks.

Topic models are also referred to as probabilistic topic models, which refer to statistical algorithms for discovering the latent semantic structures of an extensive text body. In the age of information, the amount of the written material one encounters each day is simply beyond one’s processing capacity. Topic models can help to organize and offer insights for us to understand large collections of unstructured text bodies. Originally developed as a text-mining tool, topic models have been used to detect instructive structures in data such as genetic information, images, and networks [14]. The transition from the space of terms to space of topics helps to resolve synonymy and term polysemy, and also to solve problems such as topic search, classification, summarization of documents collection. The advantage of the approach is that the resulting vectors are sparse, which, obviously, will allow one to determine which topics are dominant in the document and, as a consequence, to which class the document belongs.

Support vector machines (SVMs, also support vector networks [1]) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

Long short-term memory (LSTM) units (or blocks) are a building unit for layers of a recurrent neural network (RNN). A RNN composed of LSTM units is often called as an LSTM network. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell is
responsible for «remembering» values over arbitrary time intervals; hence the word «memory» in LSTM. Each of the three gates can be thought of as a «conventional» artificial neuron, as in a multi-layer (or feed forward) neural network: that is, they compute an activation (using an activation function) of a weighted sum. Intuitively, they can be thought as regulators of the flow of values that goes through the connections of the LSTM; hence the denotation «gate».

Feature-extracting methods based on ontologies. All methods select the depth of the ontology partitioning into categories. The choice of the depth of the partition is performed on the basis that there are a lot of data in each category; all categories were approximately balanced in all languages used in the classification. Considering feature extraction from ontologies probabilistic topic models are used. Probabilistic Topic models support relation with ontologies, and support transfer learning. It is possible to obtain hierarchical information from ontologies and to learn them from data collections from various sources (for example, Wikipedia). But probabilistic topic models have a drawback: to effectively use them, before doing stochastic matrix decomposition it is necessary to perform word processing in order to increase the compatibility of various word-document combinations. This disadvantage is easily overcome by Word Embedding.

4. Results and discussion
In this paper the authors have implemented an ensemble of classifiers that is based on the SVM classifier, Fasttext classifier, word embedding and LSTM. The choice of these models is based on the above analysis and proceeds from the considerations that it is impossible to uniquely identify a model that exceeds all other known classification methods. The ensemble will take advantage of each of the models described in the document.

The use of classifiers makes it possible to improve the accuracy of classification. Construction classifiers is carried out independently from each other on training sets obtained from the initial random replacement of documents.

The idea of the approach is to consistently construct k classifiers and combine their classification results. The classifier is built on the initial learning set, the documents of which are involved in learning with some weight coefficients. After training, the classifier is checked on the initial learning sample and the coefficients are recalculated. The coefficient decreases if the document is classified correctly, and increases otherwise.

Plan for the development of the classifier. First, a framework of the ensemble of classifiers is developed. Then each part of it is iteratively filled. The increase in the functional of each part occurs after a complete iteration over all parts of the ensemble.

Framework of the proposed ensemble of classifiers is shown in Figure 3.

![Figure 3. The framework of ensemble classifiers.](image)

Implementation of the proposed classifier is performed in the Matlab environment (Text Analytics Toolbox). The functions used are: extractFileText (read text data), erasePunctuation (clearing the text of the punctuation marks), normalizeWords, stopWords, tfidftokenizedDocument bagOfWords, trainWordEmbedding (learning a vector model), building a word2vec model, etc.
Text corpus: news articles Reuters (Reuters-21578). Reuters-21578 [15] is used in many studies to verify the quality of the developed clustering algorithm. Here are the characteristics of the samples:
- general characteristics - 91 categories for classification; Both samples are unbalanced by the number of elements in the category;
- training sample - 11413 articles for training;
- test sample - 4024 articles for classification.

Baseline. As a baseline for comparison of the proposed method, it is suggested to take methods at the previous iteration of the classifier update. The quality of the classification was measured using the precision metric (classification accuracy) based on micro-averaging [16].

The results of the experiments are given in Table 1.

| Method         | Features                                                                 | Precision | Training speed | User complexity                         |
|----------------|---------------------------------------------------------------------------|-----------|----------------|------------------------------------------|
| SVM            | Tf-idf                                                                   | 73.5 %    | High           | Easy enough. One needs to prepare the data in tf-idf format |
| Fasttext       | Frequencies of digrams and trigrams (standard classifier settings)        | 71.3 %    | High           | Easy enough. "Works out of the box," one needs to vary the number of iterations in the learning process |
| Word embedding | Learned simultaneously with convolution, vector representations of words (word embedding). | 72.3 %    | Low            | More difficult than the first two methods. One needs to build the architecture and configure the parameters. |
| LSTM           | Learned simultaneously with recurrent neurons vector representations of words (word embedding). | 70 %      | Extremely low  | More difficult than other methods. Just as in the case of cnn, one needs to build an architecture and configure hyperparameters. |

In the work the analysis of existing classification methods for the task of text classification for ontology population is carried out. Despite the fact that all methods showed themselves in terms of accuracy, approximately the same, SVM has the highest performance, which is explained by a small amount of data. The more data, the higher the quality of neural network methods and lower for the support vector method. Consequently:

1. If the data is small - it is recommended to use the SVM method.
2. If there is a lot of data per class, the CNN or LSTM method is recommended.
3. When changing from a small number of data to a larger one, it is worth trying fastText.

5. Conclusion
The proposed new version of classifiers ensemble (SVM - method of reference vectors, LSTM - neural network, word embedding - distributive semantics), intended for ontology learning systems on a text corpus from the given domain showed high results. The proposed approach is simple for implementation. In the future, it is planned to build a high-level indicative description of the text. To do this, it is necessary to syntactically analyze the sentences of the text, on the basis of which to perform the semantic marking of the roles and in the future to extract the facts of the triplets: object, predicate, subject.
Acknowledgments
The authors express their gratitude to the members of the Editorial Board who made comments and gave recommendations on the improvement of the materials of this article. The work was carried out due to the partial financing of GZ No. 2,5537.2017 / 6.7 in SFedU. Grant of the RFBR №. 18-07-50.

References
[1] Semenova A V Kureichik V M 2017 Combined Method for Integration of Heterogeneous Ontology Models for Big Data Processing and Analysis Proceedings of the 6th Computer Science On-line Conference 1 302-311
[2] Parhomenko P A Grigorev A A Astrahantsev N A 2017 Obzor i eksperimentalnoe sravnenie metodov kusterizatsii tekstov Trudyi ISP RAN 29 161-200
[3] Andrews N O Fox E A 2007 Recent developments in document clustering Technical report Computer Science (Virginia Tech)
[4] Aggarwal C C Zhai C X 2012 Mining text data (Springer Science & Business Media)
[5] Whissell J S Clarke C L 2011 Improving document clustering using Okapi BM25 feature weighting Information retrieval 5 (14) 513-523
[6] Huang A 2008 Similarity measures for text document clustering Proceedings of the sixth new Zealand computer science research student conference (Christchurch, New Zealand)
[7] Sathiyakumari K Manimekalai G 2011 A survey on various approaches in document clustering International Journal of Computer Technology and Application 5 (2) 1534-1539
[8] Aggarwal C C Zhai C X 2012 Mining text data (Springer Science & Business Media)
[9] Marchionini G 2006 Exploratory search from finding to understanding Communications of the ACM 4 (49) 41-46
[10] Vyugin V V 2013 Matematicheskie osnovyi mashinogo obucheniya i prognozirovaniya (M: MTsNMO)
[11] Mikolov T Chen K Corrado G Dean J 2013 Efficient Estimation of Word Representations in Vector Space. Retrieved from: https://arxiv.org/abs/1301.3781
[12] Pennington J Socher R Manning C D 2014 GloVe: Global Vectors for Word Representation. Retrieved from: http://www.aclweb.org/anthology/D14-1162
[13] Bojanowski P Grave E Joulin A Mikolov T 2017 Enriching Word Vectors with Subword Information. Retrieved from: https://arxiv.org/abs/1607.04606
[14] Choi F Wiemer-Hasting P Moore J 2001 Latent semantic analysis for text segmentation. Proceedings of NAACL 1 (Pittsburgh PA) pp 109-117
[15] Levis D D 2018 Reuters-21578. Retrieved from: http://www.daviddlewis.com/resources/testcollections/reuters21578
[16] Ageev M Kuralenok I Nekrestyanov I 2006 Ofitsialnyie metriki ROMIP. Retrieved from: http://romip.ru/romip2006/appendix_a_metrics.pdf