Detecting semantic duplicates in short news items

Sergei A. Fomin
Operator, Laboratory of Research Center
Civil Defense Academy EMERCOM of Russia
Address: Novogorsk District, Khimki, Moscow Region, 141435, Russian Federation
E-mail: sergio-dna@yandex.ru

Roman L. Belousov
Research Associate, Research Center
Civil Defense Academy EMERCOM of Russia
Address: Novogorsk District, Khimki, Moscow Region, 141435, Russian Federation
E-mail: romabel-87@mail.ru

Abstract

In the paper, we examine a task of detecting text messages that borrow similar meaning or relate to the same event. The noticeable feature of the task at hand is that the considered text messages are short, about 40 words per message on average. To solve this task, we design an algorithm that is based on the vector space model, meaning that every text is mapped to a point in high-dimensional space. Text-to-vector transforming is done using the TF-IDF measure. It should be noted that even for small cases with a volume of about 800 messages the dimension of the vector space can exceed 2,000 components, and on the average the dimension is about 8,500 components. To reduce the dimension of space, the method of principal components is used. The application of this method allows us to rationally reduce the dimensionality of space and leave about 3 percent of the components from their original number.

In this reduced vector space, we use agglomerative hierarchical clustering in accordance with the Lance–Williams algorithm. The actual cluster merge is done using the closest linkage algorithm. We stop merging clusters when the distance between two nearest clusters exceeds some threshold value *r* that is given to the algorithm as a parameter.

We conduct an experiment on the dataset of 135,000 news messages parsed from news aggregator feeds. During the experiment, we build the regression model for the r algorithm parameter value that allows us to predict the value of this parameter that gives good clustering results.

The designed algorithm scores high in quality metrics indicating its sufficient ability to classify a pair of messages as being duplicates or not, as well as the ability to find out whole groups of duplicate messages.

Key words: short text corpora, text clustering, near-duplicates, semantic vector space, neural network.

Citation: Fomin S.A., Belousov R.L. (2017) Detecting semantic duplicates in short news items. *Business Informatics*, no. 2 (40), pp. 47–56. DOI: 10.17323/1998-0663.2017.2.47.56.

Introduction

In June 2014, the Yuri Levada Analytical Center published a report entitled "Russian media landscape: television, press, the Internet" [1]. The report places particular emphasis on the fact that about one third of the population (34%) uses the Internet in order to "watch the latest news" and 20% to "understand what is happening in the country and abroad".

Public commercial companies and state organizations do not disregard these figures and facts, since an active information policy on the Internet should be conducted in order to create a positive image and unblemished business reputation.

To realize these needs, information systems are being developed which allow for automatic collection, processing and analysis of information from various sources. One

of the key requirements placed on such systems is their ability to detect similar publications, as well as publications devoted to one event (http://www.mlg.ru/solutions/pr/analysis/, https://pressindex.ru/#technology).

This article addresses the algorithm for retrieving duplicates in short news items received from RSS feeds of various news portals or otherwise.

Duplicates are news items identical in meaning, which can reveal a partial or complete lexical concurrence. Therefore, duplicates have a semantic similarity.

The problem of retrieving duplicates, including short text documents, is not a new one, and papers [2–5] have been dedicated to solution of the problem.

1. Initial data

Short news items generally consist of a headline and a lead, the first paragraph that answers the questions what, when and where.

For this study, a collection of short news items for 20 days was used. The volume of the collection is about 135,000 news items. If the items cover the same event, then they have the same duplicate label (*dup)*.

Table 1 presents some statistical characteristics of the prepared collection of news items.

Table 1.

Statistical characteristics of a collection on news items

Characteristic	Collection		
	Average	Median	Mode
Number of words in the item	39.72	37	33
Number of unique words in the item	33.77	32	30
Number of items having the same *dup*	14.73	4	1
Number of items per day	6725.95	6487	–

The average length of items is 39.72 words. The items can have grammatical mistakes and typos. The duplicates are retrieved among the items posted on the same day. The structure and types of the initial data are presented in *Table 2.*

Table 2.

Structure and type of initial data

id	*head*	*description*	*time*	*dup*
hash	char	char	smalldatetime	char

Here, *id* is a unique identifier of the item, *head* is an item header, *description* is the main part of the item (lead), *time* is the date and time of the item posting, *dup* is the duplicate label. If two items have the same *dup*, then they are semantic duplicates.

Items which have the same *dup* form groups. *Table 3* provides information on a number of such groups. It is noteworthy that the percentage of unique items is insignificant: they are only 2,385 out of 135,000 such items. The collection has 108 groups consisting of 100 or more items (the largest group consists of 1,039 items).

Table 3.

Groups of items

Number of items in a group	Number of groups
1	2385
2	1026
3	649
4	522
5	433
6	351
7	278
8	275
9	229
10	195
> 10	2792

2. Problem statement

An article by Yu.G. Zelenkov and I.V. Segalovich is dedicated to the problem of detecting duplicates in text documents. It provides a comparative study of the most popular modern methods of detecting near-duplicates [6]. It is significant that near-duplicates are not always semantically similar, i.e. have the same meaning. Moreover, the methods enabling us to find near-duplicates do not always work correctly for short-length texts.

Therefore, the research task is formulated as follows: to develop an algorithm for detecting semantic duplicates in short news items and grouping them together.

3. Algorithm description

The idea of semantic vector space is taken as a basis for the algorithm implementation, where each text is considered as a point in multidimensional space. The closely spaced points correspond to semantically similar documents [7]. Let us consider the description of each algorithm stage and tools for their implementation.

The first stage is preprocessing. The items posted on one day are aggregated into smaller size corpora. After that, all words in the items are brought to a normal form using the morphological analyzer Mystem (https://tech.yandex.ru/mystem/). Therefore, the original collection of items C can be considered as a combination of aggregated corpora c_i:

$$C = \bigcup_i c_i . \quad (1)$$

The collection of short news items under study is divided into 20 corpora, since it contains items for 20 days.

The second stage is a construction of the vector space model. The items from corpus c_i must be converted to a matrix. To solve this problem the TF-IDF measure [8] is used.

For each word t in a particular item d, a TF-measure is calculated by the formula (2):

$$tf(t,d) = \frac{n_i}{\sum_k n_k}, \quad (2)$$

where n_i is a number of entries of word t to item d;

$\sum_k n_k$ is a total number of words in item d.

For each word t in text corpus c_i the IDF-measure is calculated by the formula (3):

$$idf(t,c_i) = \log \frac{|c_i|}{|(d_i \supset t)|}, \quad (3)$$

where $|c_i|$ is number of items in corpus c_i;

$|(d_i \supset t)|$ is the number of items in which word t is found.

Thus, each item is converted into a vector, which component is a TF-IDF measure of each word in this item. For a specific word, the TF-IDF measure is defined as a product of TF and IDF measures.

The TF measure for the word is defined locally in each item, and the IDF measure is global for the corpus and does not depend on a specific item. Value TF-IDF is interpreted as a "contribution" of a particular word to the meaning of the item.

The result of implementation of the second stage is represented in the form of a matrix (*Table 4*), where each column defines a separate word t from corpus c_i, and each row corresponds to some item d.

Table 4.

TF-IDF matrix

	t_1	t_2	...	t_m
d_1	tf - $idf(d_1, t_1)$	0	...	tf - $idf(d_1, t_m)$
d_2	0	0	...	tf - $idf(d_1, t_2)$
...	...	...	...	...
d_n	tf - $idf(d_n, t_1)$	...	...	tf - $idf(d_n, t_m)$

When constructing a vector space model, words with a high degree of frequency and those which occur once are ignored. For the presented initial data, high-frequency words are those which are encountered in more than 90% of the items.

To build the vector space model, the machine-learning library scikit-learn (http://scikit-learn.org/stable/) was used for the Python programming language.

The third stage is to decrease the number of vector components. The purpose of this stage is to reduce the data dimension, since on average each matrix corresponding to item corpus c_i contains 8,547 columns. To reduce the number of columns in *Table 4* with a minimum loss of information content, principal component analysis is used.

Each column of *Table 4* is variable t_i, and the row is a number of observations. All variables t_i are centered by the formula (4):

$$x_i = t_i - \overline{t_i}, \quad (4)$$

where $\overline{t_i}$ is an average value of variable t_i.

After that, a transition to new variables is accomplished – to the principal component by the formula (5):

$$pc_j = \sum_i^m v_{ij} \cdot x_i, \quad (5)$$

in this case the sum of squares of weight coefficients v_{ij} shall have a unit value.

New variables $pc_1, pc_2, \ldots, pc_m$ are created such that the following conditions [9] are fulfilled:

✦ the first principal component pc_1 has a maximum possible sampling variance $sVar(pc_1)$;

✦ variable pc_2 is uncorrelated with pc_1 and has a maximum possible sampling variance $sVar(pc_2)$;

✦ variable pc_3 is uncorrelated with pc_1, pc_2 and has a maximum possible sampling variance $sVar(pc_3)$;

✦ etc.

To reduce the number of columns in *Table 4*, it is sufficient to omit the variables which have the least weights in the linear combination (5).

The number of columns in the new table is calculated as a product of the number of columns in the old table and parameter *m*, where $m \in (0, 1]$. It is found that if parameter *m* varies from 0.02 to 0.1, the quality of the algorithm's work slightly varies. Values *m* exceeding 0.1 increase the computational cost of the subsequent operations. The most rational value of parameter *m* equals 0.03.

The principal component analysis (PCA) is also implemented in the machine-learning library scikit-learn.

The fourth stage is to measure commonality of two vectors. After implementing the third stage, the number of columns in *Table 4* was reduced, but the number of rows remained unchanged. Each row corresponds to a specific text (document) *d* and is considered as a vector.

The most popular method of measuring the commonality of two vectors is to find the cosine of the angle between them [7]. The higher the cosine value, the more similar are the vectors.

For the convenience of further use of clustering algorithms, the cosine value is subtracted from the unit. The result is a matrix of cosine distances A:

$$a_{ij} = 1 - \cos(d_i, d_j). \quad (6)$$

Finding the cosine of the angle between two vectors is implemented in the data analysis library scikit-learn for Python programming language.

It is important to note that the value of the cosine does not yet allow us to judge whether the two items are semantic duplicates.

The fifth stage is to cluster the vectors. The items that fall into the same groups are semantically similar, and the appropriate vectors form clusters.

To cluster vectors **d**, agglomerative hierarchical clustering is used. The purpose of this clustering is as follows. First, each vector corresponding to a text item is treated as a separate cluster. The distances between these clusters are contained in matrix **A** obtained in the fourth stage of the algorithm.

Then the merge process is started. In each iteration, a new cluster $W = U \cup V$ instead of a pair of the closest clusters U and V is formed. The distance from new cluster W to any other cluster S is calculated by the Lance–Williams algorithm [10]:

$$\begin{aligned} R(W,S) &= \alpha_U R(U,S) + \alpha_V R(V,S) + \\ &+ \beta R(U,V) + \gamma \left| R(U,S) - R(V,S) \right|, \end{aligned} \quad (7)$$

where distances $R(U,S)$, $R(V,S)$, $R(U,V)$ and numeric parameters α_U, α_V, β, γ are calculated by the nearest neighbors algorithm [10]:

$$\begin{aligned} R(W,S) &= \min_{w \in W, s \in S} \rho(w,s), \\ \alpha_U &= \frac{1}{2},\ \alpha_V = \frac{1}{2},\ \beta = 0,\ \gamma = -\frac{1}{2}. \end{aligned} \quad (8)$$

The process of merging clusters is terminated when the distance between two clusters exceeds a certain value of parameter *r*.

This agglomerative clustering algorithm is implemented in the machine learning library scikit-learn.

The agglomerative clustering algorithm presented enables us to find pairs of semantic duplicates and combine them into groups.

4. Numerical experiment

Within the numerical experiment, the algorithm quality was evaluated taking into account, on the one hand, the ability to classify text item pairs as semantic duplicates, and on the other hand, the ability to cluster the duplicates found.

To evaluate the algorithm abilities to combine items into sense-groups, i.e. assessment of the clustering quality, the adjusted Rand Index (ARI) [11] and the Adjusted Mutual Information Index (AMI) are applied [12].

The ARI and AMI indexes present a measure of agreement and a measure of similarity between two partitions of a set of objects, respectively.

The classification of items was evaluated according to the following metrics: accuracy (*P*), completeness (*R*) and *F*-measure – a harmonic mean between the accuracy and completeness [13]. Using these metrics, the algorithm's ability to classify pairs of text items as semantic duplicates is determined. For convenience of perception of the classification results, let us designate a class of duplicates by digit 1, and a class of non-duplicates by digit 0.

Let us give an example. If two items that fall into the same group have the same *dup* in *Table 2* (class 1), then the classification has been done correctly and semantic duplicates (class 1) are found. This classification is called true-positive (*TP*).

Another example. Two items are classified as semantic duplicates (class 1), i.e. they fell into one group. In this case, these items have different *dups* in *Table 2* (class 0). This goes to prove that the algorithm incorrectly classified this pair of items. Such a solution is called a false-positive (*FP*).

A true-negative *TN* and false-negative *FN* classification are distinguished.

The above classification types and selected metrics are related by formula (9):

$$P = \frac{TP}{TP + FP},\ R = \frac{TP}{TP + FN},\ F = 2\frac{P \cdot R}{P + R}. \qquad (9)$$

The metrics considered and quality indexes are implemented in the machine-learning library scikit-learn.

The main part in evaluating algorithm quality is played by clustering parameter *r*, which is determined by the agglomerative clustering. It is obvious that the optimal value of parameter *r* depends on the individual features of the corpus of short news items. A value that maximizes the *F*-measure in class 1 is considered to be an optimal value *r* for a separate corpus. It is worth noting that the term "optimality" is used in the narrow sense. This means that resulting values *r* are optimal only for certain algorithm setting parameters. For other setting parameter values, the optimum values may vary.

Therefore, an answer to the question "How much does the value of parameter *r* depend on these features, and can it be predicted?" is of practical interest.

The first stage of the experiment consisted in empirical selection of such a value of clustering parameter *r*, at which the *F*-measure in class 1 reaches its greatest value. It is class 1 that determines the algorithm quality, inasmuch as by virtue of the specifics of the data under study, the completeness and accuracy in class 0 are always close to unit.

Figure 1 depicts a graph of variance of selected metrics (accuracy, completeness, *F*-measure for class 1) depending on the values of clustering parameter *r* for a random text corpus from the collection under study. The graph has a pronounced discreteness. This is due to the peculiarities of the agglomerative clustering: since each value of parameter *r* determines the distance between the clusters, and a number of clusters is limited, then the metrics quality values can vary only on a certain limited set of values of clustering parameter *r*.

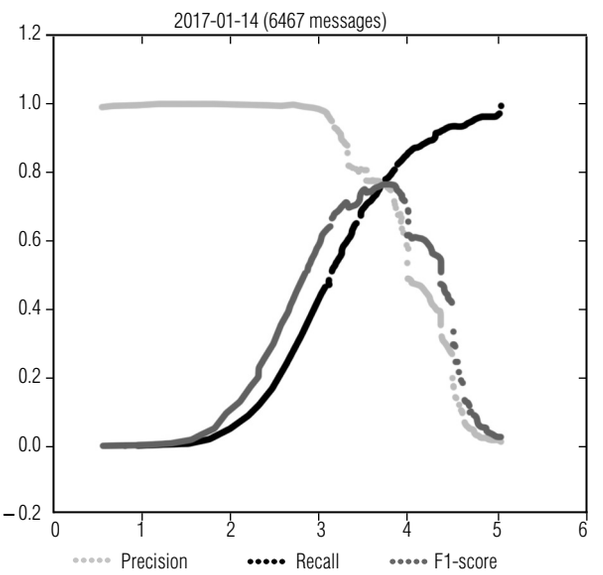

Fig. 1. Quality of classification (class 1) of pairs of corpus items

Table 5 depicts values of the classification metrics, values of the clustering metrics for each corpus c_i and appropriate values of parameter *r*.

Table 5.

Matrix values

$\|c_i\|$	Algorithm quality					*r*
	Classification			Clustering		
	P	*R*	*F*	*ARI*	*AMI*	
855	0.93	0.75	0.83	0.83	0.83	2.05
2117	0.78	0.91	0.84	0.84	0.90	2.79
6056	0.66	0.67	0.67	0.67	0.76	3.59
7553	0.87	0.65	0.74	0.74	0.75	3.68
4142	0.69	0.83	0.75	0.75	0.769	3.88
2934	0.80	0.76	0.78	0.77	0.84	3.02
5093	0.75	0.77	0.76	0.76	0.82	3.44
6478	0.79	0.69	0.74	0.74	0.73	3.35
6869	0.72	0.76	0.74	0.74	0.83	3.89
6350	0.77	0.63	0.69	0.69	0.79	3.61
7097	0.73	0.72	0.73	0.73	0.77	3.78
6496	0.73	0.62	0.67	0.67	0.68	3.31
8366	0.71	0.73	0.72	0.72	0.81	4.04
11745	0.76	0.67	0.71	0.71	0.72	4.01
11106	0.79	0.75	0.77	0.77	0.79	4.20
7568	0.77	0.64	0.70	0.70	0.70	3.47
12221	0.76	0.74	0.75	0.75	0.80	4.46
10472	0.71	0.76	0.73	0.73	0.78	4.15
6467	0.76	0.78	0.77	0.77	0.83	3.72
4534	0.86	0.76	0.81	0.81	0.82	3.29

The data analysis in *Table 5* makes it possible to draw the conclusion that if the value of clustering parameter *r* is chosen correctly, then the algorithm quality can be estimated as good. However, the value of parameter *r* can vary greatly for each corpus.

The second stage of the experiment consisted in finding the dependence of the value of clustering parameter *r* on the individual features of the corpora of short news items.

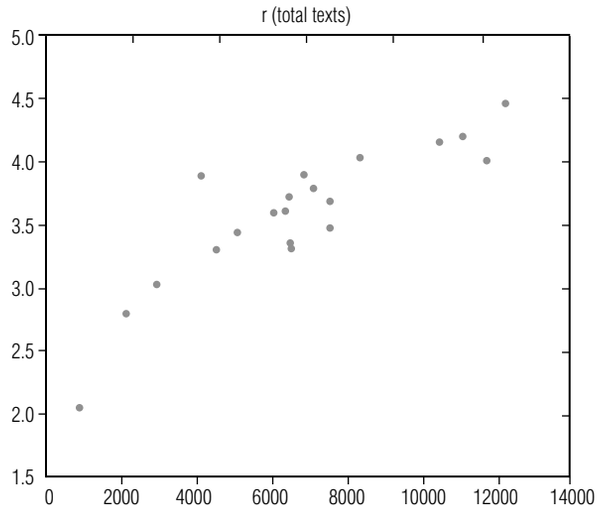

Fig. 2. Dependence of the optimal value of parameter *r* from a number of items

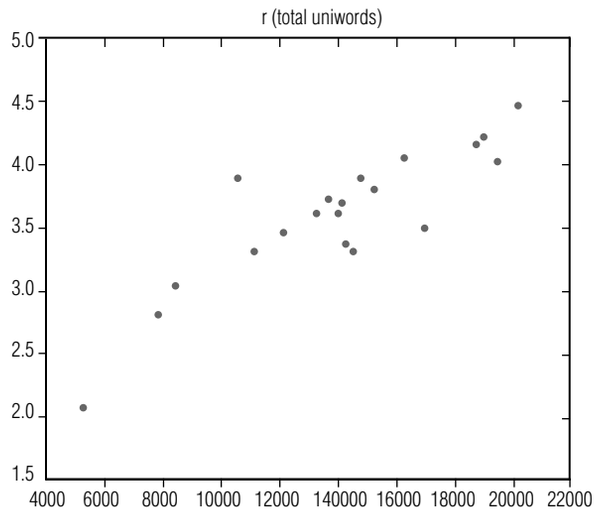

Fig. 3. Dependence of the optimal value of parameter *r* from the number of unique words in the corpus

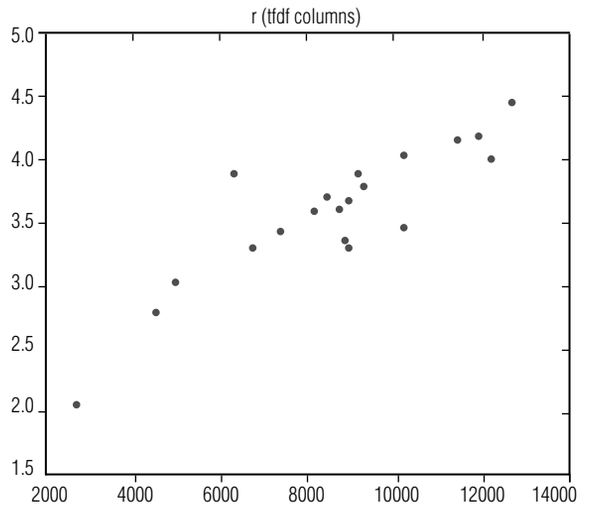

Fig. 4. Dependence of the optimal value of parameter *r* rom the number of columns on the TF-IDF matrix

Individual features of the corpus of short news items can be determined by values of the following parameters:

– p_1 is the number of items in the corpus;

– p_2 is the average length of items;

– p_3 is the most widely used (mode) length of items;

– p_4 is the average number of unique words in the items;

– p_5 is the mode of the number of unique words in the items;

– p_6 is the total number of unique words in the corpus;

– p_7 is the number of columns of TF-IDF matrix (*Table 4*);

– p_8 is the clustering parameter *r*.

Among the values of the parameters obtained during the experiment, some regularities were detected.

For example, with an increase of the number of items in corpus p_1, the clustering parameter *r* as a whole also increases (*Figure 2*).

Figure 3 depicts a growth trend for the optimal value of parameter *r* with the increase of a number of unique words in corpus p_6.

Figure 4 depicts a dependence diagram of the optimal value of parameter *r* from p_7 – the number of columns in the TF-IDF matrix.

Two models will be used to predict the values of clustering parameter *r* based on the values of input variables $p_1, ..., p_7$: the multilinear regression model and the neural network model (MLP *i-h-o*, *hidden*, *output*), where *i* is the dimension of the input value vector, *h* is a number of neurons in the hidden layer, *o* is a dimension of the output vector, *hidden* is an activation function of the neurons of the hidden layer, *output* is a function of activating neurons of the output layer.

The multilinear regression model appears as follows:

$$p_8 = \beta_0 + \beta_1 p_1 + \beta_2 p_2 + \cdots \beta_7 p_7 + \varepsilon, \qquad (10)$$

where ε is a random component, error;

β_i – unknown parameters.

The unknown parameters $\widetilde{\beta}_i$ of model (10) are estimated by the least-square method. For testing the model quality, the sliding control procedure for individual objects (leave-one-out CV) [14, 15] was used: 19 observations, where each observation contains the values of parameters $p_1, p_2, ..., p_8$, were used as a training sample for constructing the multilinear regression model, 1 observation was used for monitoring, i.e. predicting parameter *r*.

Taking into consideration the fact that the collection has 20 corpora with a total volume of about 135,000

items, 20 iterations of cross-validation were carried out. *Table 6* depicts the cross-validation results for multilinear regression model (10).

Table 6.

Cross-validation results for multilinear regression model

$\|c_i\|$	R_2	Prediction evaluation	
		r_i	$\tilde{r}_i$
855	0.85	2.051	2.783
2117	0.89	2.793	2.480
6056	0.90	3.599	3.326
7553	0.91	3.689	4.160
4142	0.92	3.889	3.217
2934	0.89	3.029	2.986
5093	0.90	3.441	3.631
6478	0.90	3.354	3.65
6869	0.90	3.895	3.588
6350	0.90	3.611	3.466
7097	0.91	3.787	4.159
6496	0.91	3.314	3.675
8366	0.90	4.043	3.787
11745	0.91	4.014	4.432
11106	0.89	4.203	4.057
7568	0.90	3.476	3.617
12221	0.88	4.465	4.319
10472	0.90	4.159	3.884
6467	0.90	3.721	3.616
4534	0.89	3.297	3.296

In the above *Table*, $|c_i|$ is the number of items in corpus c_i; R^2 is a coefficient of determination of the model built based on 19 observations; r_i is a real optimal value of the clustering parameter in corpus; c_i; $\tilde{r}_i$ is a predicted value.

The neural network model is represented by a three-layer architecture of the neural network, including an input layer, a hidden layer and an output layer. The neural network was trained using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm, which allows us to minimize the error sum of squares (*sos*).

To use the neural network model of the neural network model, a preliminary processing of the input data is conducted, which consists in scaling each input and output variable according to formula (11), so that all values of the variable belong to interval [0, 1]:

$$\delta = \frac{1}{p_i^{max} - p_i^{min}}, \qquad (11)$$
$$p_i' = 0 - \delta \cdot p_i^{min} + \delta \cdot p_i,$$

where p_i^{max}, p_i^{min} are minimum and maximum values of variable p_i;

p_i' is a scaling variable.

Table 7 provides the cross-validation results for the neural network model with architecture (MLP *i-h-o*, *hidden*, *output*).

Table 7.

Cross-validation results for the neural network model

$\|c_i\|$	MLP network configuration	R^2	Prediction estimation	
			r_i	$\tilde{r}_i$
855	7-12-1, log, log	0.97	2.051	2.818
2117	7-7-1, exp, exp	0.97	2.793	2.778
6056	7-6-1, tanh, tanh	0.97	3.599	3.597
7553	7-8-1, tanh, ident	0.85	3.689	3.732
4142	7-7-1, tanh, log	0.86	3.889	3.662
2934	7-4-1, ident, ident	0.88	3.029	2.992
5093	7-10-1, log, tanh	0.80	3.441	3.585
6478	7-6-1, exp, tanh	0.96	3.354	3.444
6869	7-8-1, ident, log	0.75	3.895	3.855
6350	7-12-1, tanh, log	0.90	3.611	3.617
7097	7-12-1, exp, ident	0.98	3.787	3.785
6496	7-9-1, exp, exp	0.97	3.314	3.494
8366	7-11-1, tanh, exp	0.92	4.043	3.916
11745	7-11-1, log, ident	0.85	4.014	4.113
11106	7-5-1, exp, log	0.93	4.203	4.203
7568	7-10-1, ident, tanh	0.90	3.476	3.539
12221	7-10-1, exp, log	0.95	4.465	4.202
10472	7-5-1, tanh, log	0.91	4.159	4.004
6467	7-4-1, exp, exp	0.95	3.721	3.720
4534	7-4-1, log, tanh	0.92	3.297	3.286

Table 8 presents a comparison of the algorithm quality factors obtained for the optimal values of clustering parameter *r* and the values predicted by the two models, respectively.

Table 8.

Comparison of the algorithm quality factors

Optimal value					
	P	*R*	*F*	*ARI*	*AMI*
Min	0.66	0.62	0.67	0.67	0.69
Max	0.93	0.91	0.84	0.84	0.90
Average	0.77	0.73	0.75	0.74	0.79
Multilinear regression					
	P	*R*	*F*	*ARI*	*AMI*
Min	0.29	0.39	0.44	0.43	0.68
Max	0.88	0.95	0.82	0.81	0.84
Average	0.68	0.71	0.67	0.67	0.77
Neural network model					
	P	*R*	*F*	*ARI*	*AMI*
Min	0.55	0.63	0.64	0.64	0.69
Max	0.86	0.95	0.84	0.84	0.90
Average	0.72	0.74	0.73	0.72	0.79

Analysis of the results obtained makes it possible to draw the following conclusions.

The values of the clustering parameter *r* obtained using the neural network model allow us to increase the algorithm quality and approach the optimal parameters taking into account the preset settings of algorithm parameters.

The results of the numerical experiment confirmed the fact that the proposed algorithm, on the one hand, is able to classify items as semantic duplicates, and, on the other hand, combine the duplicates found into groups based only on the frequency characteristics of corpora and texts.

Conclusion

This article presents an algorithm of retrieving semantic duplicates in short news items based on the idea of semantic vector space [7]. With this approach, each news item is considered as a point in the multidimensional space.

For quality assessment, metrics are introduced which evaluate the algorithm's ability to classify the items as semantic duplicates and combine the duplicates found into groups.

It has been established that the algorithm quality depends heavily on clustering parameter *r*. The paper proposes models which make it possible to predict parameter *r* based on the characteristics of the text corpus under study.

The algorithm developed showed a quite acceptable work quality.

It is assumed that the algorithm work quality can be improved by using methods that take into account the context, for example, word2vec and doc2vec [16].

For practical application of the proposed algorithm, the optimization method is also to be developed. That will enable us to reduce the algorithm running time and reduce the memory requirements. In the current algorithm implementation, the time and memory requirements increase as a square of a number of items in the text corpus. ■

References

1. Volkov D., Goncharov S. (2014) *Rossiyskiy media-landshaft: televidenie, pressa, Internet* [The Russian media landscape: TV, press, Internet]. Available at: http://www.levada.ru/17-06-2014/rossiiskii-media-landshaft-televidenie-pressa-internet (accessed 14 October 2015) (in Russian).
2. Rangrej A., Kulkarni S., Tendulkar A.V. (2011) Comparative study of clustering techniques for short text documents. Proceedings of the *20th International World Wide Web Conference (WWW 2011). Hyderabad, India, 28 March – 01 April 2011*, pp. 111–112.
3. Errecalde M.L., Ingaramo D.A., Rosso P. (2010) A new AntTree-based algorithm for clustering short-text corpora. *Journal of Computer Science and Technology*, vol. 10, no. 1, pp. 1–7.
4. Petersen H., Poon J. (2011) Enhancing short text clustering with small external repositories. Proceedings of the *9th Australasian Data Mining Conference (AusDM'11). Ballarat, Australia, 01–02 December 2011*, pp. 79–89.
5. Kirichenko K.M, Gerasimov M.B. (2001) *Obzor metodov klasterizatsii tekstovoy informatsii* [Review of text information clustering methods]. Available at: http://www.dialog-21.ru/en/digest/2001/articles/kirichenko/ (accessed 17 January 2017) (in Russian).
6. Zelenkov Yu.G., Segalovich I.V. (2007) *Sravnitel'nyy analiz metodov opredeleniya nechetkikh dublikatov dlya Web-dokumentov* [Comparative analysis of methods for near-duplicate detection for Web-documents]. Available at: download.yandex.ru/company/paper_65_v1.rtf (accessed 28 September 2016) (in Russian).
7. Turney P.D., Pantel P. (2010) From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, no. 37, pp. 141–188.
8. Wu H.C., Luk R.W.P., Wong K.F., Kwok K.L. (2008) Interpreting TF-IDF term weights as making relevance decisions. *ACM Transactions on Information Systems*, vol. 26, no. 3, pp. 13.1–13.37.
9. Aivazyan S.A., Buhstaber V.M., Enyukov I.S., Meshalkin L.D. (1989) *Prikladnaya statistika. Klassifikatsiya i snizhenie razmernosti* [Applied statistics. Classification and reduction of dimensionality]. Moscow: Finance and Statistics (in Russian).

10. Vorontsov K.K. *Lektsii po algoritmam klasterizatsii i mnogomernogo shkalirovaniya* [Lectures on algorithms of clustering and multidimensional scaling]. Available at: http://www.ccas.ru/voron/download/Clustering.pdf (accessed 28 September 2016) (in Russian).
11. Hubert L., Arabie P. (1985) Comparing partitions. *Journal of Classification*, no. 2, pp. 193–218.
12. Vinh N.X., Epps J., Bailey J. (2010) Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, no. 11, pp. 2837–2854.
13. Sokolov E. (2015) *Seminary po vyboru modeley* [Workshops on models selection]. Available at: http://www.machinelearning.ru/wiki/images/1/1c/Sem06_metrics.pdf (accessed 17 January 2017) (in Russian).
14. Kohavi R. (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the *Fourteenth International Joint Conference on Artificial Intelligence (IJCAI 95). Montreal, Quebec, Canada, 20–25 August 1995*, vol. 2, pp. 1137–1145.
15. Refaeilzadeh P., Tang L., Liu H. (2009) Cross-validation. *Encyclopedia of Database Systems*. Springer, pp. 532–538.
16. Shuriga L. *Sovremennye metody analiza tonal'nosti teksta* [Modern methods of sentiment analysis in text]. Available at: http://datareview.info/article/sovremennyie-metodyi-analiza-tonalnosti-teksta/ (accessed 01 February 2017) (in Russian).

Поиск семантических дубликатов в коротких новостных сообщениях

С.А. Фомин
бакалавр технических наук
оператор лаборатории научно-исследовательского центра
Академия гражданской защиты МЧС России
Адрес: 141435, Московская область, г. Химки, мкр. Новогорск
E-mail: sergio-dna@yandex.ru

Р.Л. Белоусов
кандидат технических наук, научный сотрудник научно-исследовательского центра
Академия гражданской защиты МЧС России
Адрес: 141435, Московская область, г. Химки, мкр. Новогорск
E-mail: romabel-87@mail.ru

Аннотация

В статье рассмотрена задача, связанная с обнаружением публикаций, схожих по смыслу, а также публикаций, посвященных одному событию. Особенность решаемой задачи заключается в том, что в качестве публикаций рассматриваются короткие новостные сообщения, средняя длина которых составляет 40 слов. Для решения указанной задачи разработан алгоритм, в основу которого положена векторная модель семантики, где каждый текст рассматривается как точка в многомерном пространстве. Преобразование корпуса текстов в матрицу производится с помощью меры TF-IDF. Необходимо отметить, что даже для небольших корпусов (объемом порядка 800 сообщений) размерность векторного пространства может превосходить 2000 компонент, а в среднем размерность составляет около 8500 компонент. Для сокращения размерности пространства используется метод главных компонент. Его применение позволяет рационально сократить размерность пространства и оставить около трех процентов компонент от их исходного количества.

В сокращенном пространстве для объединения векторов в кластеры применяется агломеративная иерархическая кластеризация по алгоритму Ланса–Уильямса, который запускает процесс слияния кластеров. Слияние кластеров производится с помощью вычисления расстояния между ближайшими элементами этих кластеров. Процесс слияния кластеров прекращается в том случае, если расстояние между двумя кластерами превышает некоторое значение r.

При проведении численного эксперимента построена регрессионная модель, позволяющая найти наиболее подходящее значение параметра r для каждого корпуса сообщений. В качестве исходных данных для проведения численного эксперимента использовалась коллекция коротких новостей, общий объем которых составляет около 135 тысяч сообщений.

Разработанный алгоритм имеет достаточно высокие показатели качества, которые учитывают, с одной стороны, способность классифицировать пары текстовых сообщений как семантические дубликаты, а с другой – способность объединять найденные дубликаты в группы.

Ключевые слова: коллекция коротких текстовых сообщений, кластеризация текстов, нечеткие дубликаты, векторная модель семантики, нейронная сеть.

Цитирование: Fomin S.A., Belousov R.L. Detecting semantic duplicates in short news items // Business Informatics. 2017. No. 2 (40). P. 47–56. DOI: 10.17323/1998-0663.2017.2.47.56.

Литература

1. Волков Д., Гончаров С. Российский медиа-ландшафт: телевидение, пресса, Интернет. [Электронный ресурс]: http://www.levada.ru/2014/06/17/rossijskij-media-landshaft-televidenie-pressa-internet/ (дата обращения 14.10.2015).
2. Rangrej A., Kulkarni S., Tendulkar A.V. Comparative study of clustering techniques for short text documents // Proceedings of the 20th International World Wide Web Conference (WWW 2011). Hyderabad, India, 28 March – 01 April 2011. P. 111–112.
3. Errecalde M.L., Ingaramo D.A., Rosso P. A new AntTree-based algorithm for clustering short-text corpora // Journal of Computer Science and Technology. 2010. Vol. 10. No. 1. P. 1–7.
4. Petersen H., Poon J. Enhancing short text clustering with small external repositories // Proceedings of the 9th Australasian Data Mining Conference (AusDM'11). Ballarat, Australia, 01–02 December 2011. P. 79–89.
5. Кириченко К.М., Герасимов М. Б. Обзор методов кластеризации текстовой информации. [Электронный ресурс]: http://www.dialog-21.ru/en/digest/2001/articles/kirichenko/ (дата обращения 17.01.2017).
6. Зеленков Ю.Г., Сегалович И.В. Сравнительный анализ методов определения нечетких дубликатов для Web-документов // Труды 9-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (RCDL'2007). Переславль-Залесский, Россия. 15–18 октября 2007 г. С. 166–174.
7. Turney P.D., Pantel P. From frequency to meaning: Vector space models of semantics // Journal of Artificial Intelligence Research. 2010. No. 37. P. 141–188.
8. Interpreting TF-IDF term weights as making relevance decisions / H.C. Wu [et al.] // ACM Transactions on Information Systems. 2008. Vol. 26. No. 3. P. 13.1–13.37.
9. Прикладная статистика. Классификация и снижение размерности / С.А. Айвазян и [др.]. М.: Финансы и статистика, 1989.
10. Воронцов К.К. Лекции по алгоритмам кластеризации и многомерного шкалирования. [Электронный ресурс]: http://www.ccas.ru/voron/download/Clustering.pdf (дата обращения 28.09.2016).
11. Hubert L., Arabie P. Comparing partitions // Journal of Classification. 1985. No. 2. P. 193–218.
12. Vinh N.X., Epps J., Bailey J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance // Journal of Machine Learning Research. 2010. No. 11. P. 2837–2854.
13. Соколов Е. Семинары по выбору моделей. [Электронный ресурс]: http://www.machinelearning.ru/wiki/images/1/1c/Sem06_metrics.pdf (дата обращения 17.01.2017).
14. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection // Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI 95). Montreal, Quebec, Canada, 20–25 August 1995. Vol. 2. P. 1137–1145.
15. Refaeilzadeh P., Tang L., Liu H. Cross-validation // Encyclopedia of Database Systems. Springer, 2009. P. 532–538.
16. Шурига Л. Современные методы анализа тональности текста. [Электронный ресурс]: http://datareview.info/article/sovremennyie-metodyi-analiza-tonalnosti-teksta/ (дата обращения 01.02.2017).