Ramiz M. Aliguliyev [1], Gunay Y. Niftaliyeva [2] DOI: 10.25045/jpit.v06.i2.04
Institute of Information Technology of ANAS, Baku, Azerbaijan
[1]r.aliguliyev@gmail.com; [2]gunayniftali@gmail.com

DETECTING TERRORISM-RELATED ARTICLES ON THE E-GOVERNMENT USING TEXT-MINING TECHNIQUES

In this paper, a method based on text-mining techniques for detecting terror-related articles on the e-government is proposed. The proposed method consists of several stages: 1) creation of terror-related vocabulary; 2) creation of a semantic network of words; 3) morphological analysis of words; 4) initial filtration of documents; 5) calculation of the semantic similarity between words by using a semantic network of words; 6) determination of semantic similarity between sentences; 7) determination of semantic similarity between documents; 8) classification of documents. Hybrid similarity measures are introduced to calculate the similarity among words, sentences and documents. A hybrid classification method combining the kNN, Bayes and new proposed Ramiz-Gunay methods for identification of terror-related articles is proposed.

Keywords: *e-government; e-government security; terrorism; text mining; hybrid similarity measure; kNN method; modified Bayes method; Ramiz-Gunay method; hybrid classification method.*

Introduction

Modern criminal groups can commit malicious activities against the state and society not only in the real world, but also in a virtual environment (Internet, e-government). These activities usually have various purposes, including propaganda against the state and dissemination of information, which promote terrorism and shake the foundations of national and spiritual values. [1–7].

The timely detection of this information in e-government is very important in ensuring the security of the state and society, which is one of the most pressing scientific-theoretical and practical problems of our time [6, 7]. It is no coincidence that the problem of e-government security is specified as one of the most actual 13 research directions to be studied in the eGovRTD2020 project adopted by the European Commission [8].

One of the main functions of e-government is to protect the citizens from possible impairment and violence. Linders [9] examined the evolution of citizen-state relations and concluded that the Internet, in particular e-government, is the most effective and affordable tool to inform about possible crimes, as well as to improve the relations between community members and law enforcement authorities. However, studies show that criminal groups "benefit" from this favorable environment as well, and this has become a source of great danger for the state and society. 9/11 is the best example. Further analysis has shown that the organized criminal group behind 9/11 planned and coordinated the entire act via the Internet. One can naturally conclude the virtual world is a very favorable environment for criminal groups to commit their evil deeds.

Understandably, one of the important tasks of the modern state is to identify and analyze secret criminal networks operating via the Internet and e-state. This virtual space has a wide range of opportunities to communicate and coordinate operatively. Criminal group members can communicate via websites, e-mail, blogs, online chats, etc. In most cases, transmitted information consists of text. Therefore, analysis of the text transmitted through the virtual environment is essential to preventing possible terrorist acts and ensuring the security of e-government [10]. Currently, text mining, the intelligent analysis of text gathered from various sources, is considered one of the most advanced and effective technologies in knowledge control [11].

Another reason for the popularity of text mining and its wide range of applications is the dominance of text in the information, regardless of the information's sources. According to International Data Corporation analysts, about 80% of produced information contains text [12].

Thus, intelligent analysis of text is a very important and topical area of study in ensuring e-government security.

Hence, based on the urgency of the security problem, we propose a text-mining-based method to detect terrorism-related text on the e-government. The method is similar to the method proposed in [3]. However, the proposed method has several distinct advantages:

- unlike the method proposed in [3], calculating the similarity of the words, the method takes into account not only semantic similarity between the words, but also syntactic structure of the sentences, i.e. the sequence of words in the sentences;
- to identify potentially suspicious documents more accurately, the similarity between the documents is calculated by a new iterative method: first, the similarity of the words is defined; afterward, the similarity between the sentences is calculated using the similarity of the words; finally, the similarity between the documents is calculated using the similarity of the sentences.
- a hybrid similarity measure is introduced to calculate the similarity between the sentences;
- a new method is proposed for classification.

The article is structured as follows. The second section provides a brief overview of the studies in the data-mining field. The third section describes the stages of the proposed method. A conclusion and acknowledgement of further studies are provided in the fourth section.

A brief overview of related works

A variety of methods, algorithms and models based on text-mining techniques have been proposed for detection, identification and tracking of crime and terrorism-related articles in the virtual environment (Internet, e-government). For example, [4, 5] propose new algorithms to define the similarity between the documents for the filtering and authentication of criminal information on the web. [1] uses data extraction and clustering methods of text-mining techniques in identifying criminal documents in Arabic. A rules-based approach is applied for data extraction, and a self-organized neural network (Kohonen network) is applied for document clustering. In [2], a two-staged method is proposed for the identification of criminal articles, which includes documents' detection and their clustering. The first stage eliminates unimportant words in the documents and describes the documents as a vector of important words, after which a measure is introduced to calculate the similarity between the documents. In the second stage, the documents are grouped according to criminal types by applying a clustering algorithm. A new approach based on text analysis to detect terror-related Internet articles is proposed in [3]. This approach creates the list of context words (nouns) from the set of terrorism-related articles using the semantic WordNet [13]. Then it calculates the importance of context words by applying a WUP [14] metric. Last, the approach classifies the documents using b-grams and the Keselj metric [15]. For the detection of multilingual terror-related documents [16] proposes a new approach based on a classification method. This approach uses a combination of a graphic representation model of the web documents and the classification algorithm C45. The method proposed in [17] studies terrorist activities (profiles) by analyzing website articles with the use of data-mining algorithms. In [18], an evolving fuzzy grammar method is proposed to classify criminal-related articles. The method describes selected text fragments in the fuzzy structure.

The proposed method

The proposed method consists of several stages: 1) creation of terror-related vocabulary, depending on the language of the data circulated in the explored space; 2) creation of a semantic network of words for the reviewed language network (the accuracy of the method highly depends on the network); 3) morphological analysis of the words; 4) initial filtration of the documents using the vocabulary; 5) calculation of the semantic similarity between the words;

6) determination of semantic similarity between the sentences; 7) determination of semantic similarity between the documents; 8) classification of the documents. Let's suppose that a terror-related vocabulary database (VBase) has been created in a certain language, and a semantic network of words has been developed (let's denote it WordNet, as in the network developed in English). Note that this knowledge base allows us to determine the semantic relationships between the words. For example, synonyms, hyponyms and others are easily found with the help of this network (Figure 1).

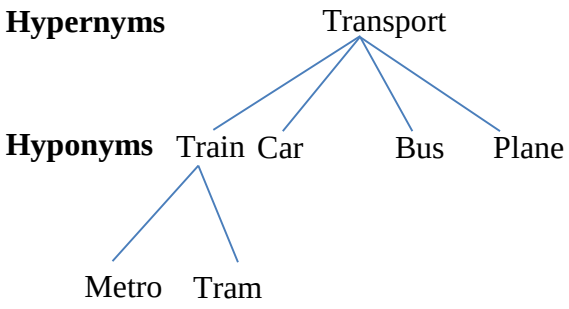

Figure 1. Hypernyms and hyponyms

Each phase of the proposed approach is explained in detail below.

1) Initial filtering of documents

Initial document filtering is performed as follows. First, the terms are extracted from the document and analyzed morphologically (to find the initial form of the word because, depending on adopted suffixes, the same words may convey different meanings) and described as a set of document words (terms), $d = (t_1, t_2, ..., t_m)$. Then, using the Simkevic-Simpson measure [19], the similarity between VBase and the set $d = (t_1, t_2, ..., t_m)$ is calculated:

$$\text{sim}_{S-S}(d, \text{VBase}) = \frac{|d \cap \text{VBase}|}{|d|}, \quad (1)$$

where $|A|$ denotes the number of elements in set A.

If $\text{sim}_{S-S}(d, \text{VBase}) \geq \varepsilon$, then document d is added to the set of suspicious documents and shifted to the next stage for identification. Where, ε is a threshold value defined in an experimental way.

2) Semantic similarity of the words

The semantic similarity between the words is determined as follows:

1. The two words t_1 and t_2 are considered.
2. The roots of the words are found in WordNet.
3. Synonyms of each word are found in WordNet, and their number is assigned;
4. The Least Common Subsume (LCS) of the words t_1 and t_2 is found using WordNet,
5. The semantic similarity between the words is calculated using formulas (2) and (3).

First, the informative content of the word is determined using $IC(t)$ to calculate the similarity between the words in WordNet [20]:

$$IC(t) = 1 - \frac{\log(\text{synset}(t) + 1)}{\log(t_{max})} \quad (2)$$

Then, semantic similarity between words is calculated using formula (2) [20, 21]:

$$\mathrm{sim}_{\mathrm{IC}}(t_1,t_2)=\begin{cases}\dfrac{2*\mathrm{IC}(\mathrm{LCS}(t_1,t_2))}{\mathrm{IC}(t_1)+\mathrm{IC}(t_2)}, & t_1\neq t_2\\ 1, & t_1=t_2\end{cases} \quad (3)$$

where, $\mathrm{LCS}(t_1,t_2)$ denotes the most similar common word with the words t_1 and t_2 in WordNet (for example, for the case shown in Figure 2 $\mathrm{LCS}(t_1,t_2)=t$), t_{max} denotes the total number of the words in WordNet, and $\mathrm{synset}(t)$ – the number of synonyms of the word t.

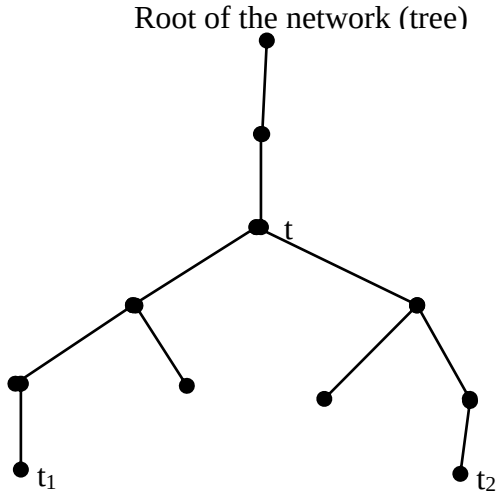

Figure 2. Semantic network of the words

Semantic similarity between the words is also calculated with the use of the WUP metric [14]:

$$\mathrm{sim}_{\mathrm{WUP}}(t_1,t_2)=\frac{2*\mathrm{depth}(t)}{\mathrm{depth}(t_1)+\mathrm{depth}(t_2)+2*\mathrm{depth}(t)}, \quad (4)$$

where, $\mathrm{depth}(t_1)$ denotes the number of nodes from t_1 to t in WordNet (tree), $\mathrm{depth}(t_2)$ is the number of nodes from t_2 to t, $\mathrm{depth}(t)$ the number of nodes from t to the network roots. For example, in the case shown in Figure 2 $\mathrm{depth}(t_1)=\mathrm{depth}(t_2)=3$ and $\mathrm{depth}(t)=2$. Then

$$\mathrm{sim}_{\mathrm{WUP}}(t_1,t_2)=\frac{2*2}{3+3+2*2}=0{,}4.$$

The semantic similarity between the words is defined as a linear combination of the metrics given by the formulas (3) and (4):

$$\mathrm{sim}(t_1,t_2)=\alpha\times\mathrm{sim}_{\mathrm{IC}}(t_1,t_2)+(1-\alpha)\times\mathrm{sim}_{\mathrm{WUP}}(t_1,t_2), \quad (5)$$

$0\le\alpha\le1$ denotes the weight coefficient.

3) Similarity measure of the sentences

Three metrics are used to calculate the similarity between the sentences: semantic, syntactic, and cosine.

A) ***Semantic similarity*****.** Semantic similarity between the sentences is calculated with the use of semantic similarity between the words (5):

$$\mathrm{sim}_{\mathrm{semantic}}(s_1,s_2)=\frac{\sum_{t_1\in s_1,t_2\in s_2}\mathrm{sim}(t_1,t_2)}{m_1+m_2}, \quad (6)$$

where m_1 and m_2 is the number of words in the sentences s_1 and s_2 respectively.

B) ***Cosine metric***. The cosine metric is based on a vector model. To calculate the similarity between the sentences based on a vector model, each of them is described as a vector first, and then the distance (similarity) between the two vectors is calculated. For example, sentences of s_1 and s_2 are given. In traditional approaches, the sentences are described as vectors, and the length of the vector is equal to the number of words in a document (or document sets). In this case, the length of the vector is many times greater than the length of the sentence (the number of words in the sentence) , and therefore, most of the elements of the vector equal zero. So, it is not effective in terms of calculation. Therefore, to calculate the similarity between the two sentences, the sets of words are formed by different words found only in these sentences. A set of words is denoted as, $WS = \{t_1, t_2, ..., t_m\}$ where, m denotes the total number of different words. The set of words in two sentences are established in the following sequence [20, 22]:

1. Two sentences s_1 and s_2 are selected.
2. Each word t extracted from the sentence s_1 is handled as follows:
   2.1. Its root word (RW) is determined with the use of the WordNet lexical database.
   2.2. If RW is involved in WS, then return to Step 2 and continue the process with the next word taken from s_1. Otherwise, proceed to Step 2.3;
   2.3. If RW is not involved in a set of WS, we must add RW to WS, then go to Step 2 and continue the process with the next word taken from s_1. The process is continued until the words of the sentence end.

The above mentioned process is repeated for a sentence s_2.

The semantic vector model is used to determine the similarity between the sentences [23, 24]. For this reason, the following steps are performed:

1. *Setting the vector*. Each element of the vector corresponds to the word in WS. Hence, the length of the vector equals the number of words in WS.
2. *Defining the vector elements.* Each element of the semantic vector (weight of the word) is determined as follows:
   2.1. If the word t of WS sentence is included to the sentence s_1, then the weight of the word in the vector equals 1, otherwise proceed to the next step;
   2.2. If the word t is not included to the sentence s_1, then the similarity between the words in the sentence s_1 and word t is calculated by formula (5).
   2.3. If the similarity between the words differs from zero, then the weight of word t in the vector is defined as its largest value. Otherwise, proceed to the next step.
   2.4. If the similarity between words equals zero, then the weight of the word t in the vector is defined as zero.

Thus, using the cosine metric, the similarity between the two vectors is calculated as follows:

$$\text{sim}_{\cos}(s_1, s_2) = \frac{\sum_{j=1}^{m} (w_{1j} \times w_{2j})}{\sqrt{\sum_{j=1}^{m} w_{1j}^2} \times \sqrt{\sum_{j=1}^{m} w_{2j}^2}} \tag{7}$$

where $s_1 = (w_{11}, w_{12}, ..., w_{1m})$ and $s_2 = (w_{21}, w_{22}, ..., w_{2m})$ are the vectors corresponding to the sentences s_1 and s_2; w_{pj} denotes the weight of the word t_j in vector s_p; m is the total number of words.

C) ***Syntactic similarity***. The semantic weight of sentence depends not only on the semantic weight of the words, but also on the words' sequence, that is position of the words in the sentence. For example, according to the abovementioned similarity metric (semantic similarity), the sentences "Ali called Hassan" and "Hassan called Ali" are estimated as similar sentences, as they are built with the same words. Therefore, when calculating the semantic similarity of the sentences, the words' sequence (their position in the sentence) must also be taken into account. Thus, a syntactic-vector approach is used to calculate the similarity of sentences based on the position of words in a sentence [25]. For this reason, the following steps are performed first [20]:

1. *Setting the vector*. The words in WS and in the sentences are used to set a syntax-vector. The length of the syntax-vector equals the number of words in WS .
2. *Defining the vector elements*. Each element of the syntactic vector indicates the weight of the word, and it equals the position of the word in the sentence. This weight is defined as follows:
   2.1. If the word t is included to the sentence s_1, then its weight in the vector is equal to its position in the sentence, otherwise proceed to the next step;
   2.2. If the word t is not included to the sentence s_1, then the similarity between the word t in the sentence s_1 and the word t is calculated by formula (5).
   2.3. If the similarity between the words differs from zero, then the value of the proper element in the vector (the weight of the word) is defined as the position of the word with the largest value in the sentence s_1.
   2.4. If the similarity between the words equals zero, then the value of the proper element in the vector is defined as zero.

The following formula is used to calculate the similarity of a sentence based on the position of words in a sentence [20, 25]:

$$\mathrm{sim}_{\mathrm{wordorder}}(s_1, s_2) = 1 - \frac{\|\mathrm{o}_1 - \mathrm{o}_2\|}{\|\mathrm{o}_1 + \mathrm{o}_2\|} \tag{8}$$

where, $\mathrm{o}_1 = (w_{11}, w_{12}, ..., w_{1m})$ and $\mathrm{o}_2 = (w_{21}, w_{22}, ..., w_{2m})$ are syntactic vectors of sentences s_1 and s_2; denotes w_{pj} the weight of the word t_j in vector o_p . $\|\cdot\|$ is the Euclidean norm.

D) Liner combination. Linear combination of semantic cosine and syntactic measures is used to calculate the similarity between the sentences:

$$\mathrm{sim}_{\mathrm{sentences}}(s_1, s_2) = \beta_1 \cdot \mathrm{sim}_{\mathrm{semantic}}(s_1, s_2) + \beta_2 \cdot \mathrm{sim}_{\mathrm{wordorder}}(s_1, s_2) + \beta_3 \cdot \mathrm{sim}_{\cos}(s_1, s_2) \tag{9}$$

where, β_i $(0 \le \beta_i \le 1, i = 1,2,3)$ are the weight parameters and provide the following condition:

$$\beta_1 + \beta_2 + \beta_3 = 1 \tag{10}$$

4) Similarity measure of the documents.

The similarity between the sentences (9) is used to define the similarity between the documents:

$$\mathrm{sim}_{\mathrm{documents}}(d_1, d_2) = \frac{\sum_{s_1 \in d_1, s_2 \in d_2} \mathrm{sim}_{\mathrm{sentences}}(s_1, s_2)}{n_1 + n_2}, \tag{11}$$

where, n_1 and n_2 are the numbers of the sentences in the documents d_1 and d_2 respectively. For simplicity, $\mathrm{sim}_{\mathrm{documents}}(d_1, d_2)$ is used instead of $\mathrm{sim}(d_1, d_2)$ below.

5) Classification of documents.

Suppose that a set of classes $\mathbf{C}=(C_1,...,C_k)$ is known as the classification of documents belonging to the given document sets $\mathbf{D}=(d_1,...,d_N)$ or to any (or some) of the classes $\mathbf{C}=(C_1,...,C_k)$. In this case, the document belongs to the closest class(es). Many classification methods have been proposed so far. KNN (k- Nearest Neighborhood) [26], Bayes [27] and the proposed RG (Ramiz-Gunay) method are used to determine the extent to which document d_i belongs to class C_q.

A) KNN method. According to this method, the extent to which the document d_i belongs to class C_q is defined by the value found by the following formula:

$$\text{score}_{k\text{NN}}(d_i \mid C_q)=\sum_{d\in k\text{NN}_q(d_i)}\text{sim}(d_i,d)\,,\ i=1,2,...,N\,;\ q=1,2,...,k\,, \tag{12}$$

where, $k\text{NN}_q(d_i)$ is k number of documents, which is closest to the document d_i to class C_q.

The document d_i belongs to the class with the highest value $\text{score}_{k\text{NN}}(d_i \mid C_q)$, in other words, $d_i \in C_{k^*}$ if $k^*=\arg\max_q \text{score}_{k\text{NN}}(d_i \mid C_q)$.

B) Modified Bayes method. According to the Bayes method, the degree to which document d_i belongs to class C_q is defined by the value of the following conditional probability:

$$\text{P}(C_q|d_i)=\frac{\text{P}(C_q)\text{P}(d_i|C_q)}{\text{P}(d_i)}, \tag{13}$$

Since $\text{P}(d_i)$ remains stable in the classification process, we do not need to take into account the denominator of the fraction in (13).

where $\text{P}(C_q)$ is the probability of the documents being in class C_q. This apriori probability is defined as follows:

$$\text{P}(C_q)=\frac{\sum_{i=1}^{N}\text{P}(C_q|d_i)}{N}. \tag{14}$$

It is assumed that the use of the words in the documents does not depend on other words for the calculation of the quantity of $\text{P}(d_i|C_q)$. Then, the probability of $\text{P}(d_i|C_q)$ is calculated using the formula below:

$$\text{P}(C_q|d_i)=\text{P}(C_q)\prod_{j=1}^{m}\left(\text{P}(\text{t}_j|C_q)\right)^{w_{ij}}. \tag{15}$$

where $\text{P}(\text{t}_j|C_q)$ denotes the probability of the word t_j in the class C_q, and m is the number of words in document set $\mathbf{D}$:

The probability $\text{P}(\text{t}_j|C_q)$ is calculated as follows:

$$\text{P}(\text{t}_j|C_q)=\frac{\sum_{d_i\in C_q}w_{ij}}{\sum_{j=1}^{m}\sum_{d_i\in C_q}w_{ij}}\,,\ j=1,...,m\,. \tag{16}$$

where, w_{ij} denotes the weight of the word t_j in the document d_i.

If the weight of the word t_j in the class C_q equals zero, then, according to the formula (16), the probability $\text{P}(\text{t}_j|C_q)$ equals zero as well. Thus, according to the formula (15), the probability $\text{P}(C_q|d_i)$ equals zero, too. Therefore, in practice, the following formula is used:

$$\mathrm{P}(\mathrm{t}_j \left| C_q \right.) = \frac{\frac{1}{N}\sum_{i=1}^{N} w_i + \sum_{d_i \in C_q} w_{ij}}{\sum_{i=1}^{N} w_i + \sum_{j=1}^{m}\sum_{d_i \in C_q} w_{ij}},\ j = 1,...,m \quad . \tag{17}$$

If we shift to a logarithmic scale in the formula (15) and divide the result by the total weight (w_i) of the words in the document d_i, we obtain the following equation:

$$\mathrm{score}_{\mathrm{M\,Bayes}}(C_q \left| d_i \right.) = \mathrm{P}(C_q \left| d_i \right.) = \frac{\log \mathrm{P}(C_q)}{w_i} + \sum_{j=1}^{m} \mathrm{P}(\mathrm{t}_j, d_i) \log \mathrm{P}(t_j \left| C_q \right.), \tag{18}$$

where $P(t_j, d_i) = w_{ij} / w_i$ is the probability of the word t_j to be used in the document d_i $w_i = \sum_{j=1}^{m} w_{ij}$, $i = 1,...,n; q = 1,...,k$.

Similar to the *k*NN method, $\mathrm{score}_{\mathrm{Bayes}}(C_q \left| d_i \right.) = \mathrm{P}(C_q \left| d_i \right.)$ is adopted in the formula (18). According to the model, d_i belongs to a class for which the probability $\mathrm{P}(C_q \left| d_i \right.)$ has the highest value, $d_i \in C_{k^*}$, where $k^* = \arg\max_{1 \le q \le k} \mathrm{score}_{\mathrm{M\,Bayes}}(C_q \left| d_i \right.)$.

C) Ramiz-Gunay (RG) method. With the help of this method, the degree to which document d_i belongs to class C_q is defined by the following formula:

$$\mathrm{score}_{\mathrm{RG}}(d_i \mid C_q) = \lambda \times \frac{\mathrm{sim}(\mathrm{O}_{d_i}, \mathrm{O}_{C_q})}{\sum_{p=1}^{k} \mathrm{sim}(\mathrm{O}_{d_i}, \mathrm{O}_{C_p})} + (1-\lambda) \times \frac{\sum_{v \in C_q} \mathrm{sim}(\mathrm{O}_{d_i}, \mathrm{O}_v)}{\sum_{p=1}^{k} \sum_{d \in C_p} \mathrm{sim}(\mathrm{O}_{d_i}, \mathrm{O}_d)} \tag{19}$$

where, $\mathrm{sim}(\mathrm{O}_{d_i}, \mathrm{O}_{C_q})$ is a similarity measure between the image O_{d_i} of the document d_i and the image O_{C_q} of the class C_q; $\mathrm{sim}(\mathrm{O}_d, \mathrm{O}_v)$ is a similarity measure between the images O_d and O_v of the documents d and v; λ with the weight coefficient, $0 \le \lambda \le 1$.

Image O_{C_q} is defined as the centre of the class, C_q $\mathrm{O}_{C_q} = (w_1^q, w_2^q, ..., w_m^q)$:

$$w_j^q = \frac{1}{\left| C_q \right|} \sum_{d \in C_q} w_j^{q,d} \ ,\ q = 1,...,k \ ,\ j = 1,...,m, \tag{20}$$

where, $\left| C_q \right|$ denotes the number of documents in the class C_q, $w_j^{q,d}$ and the weight of j–th words in the document d included in the class C_q .

Analogically, Image O_d is defined as the center of the document d, $\mathrm{O}_d = (w_1^d, w_2^d, ..., w_m^d)$:

$$w_j^d = \frac{1}{\left| d \right|} \sum_{s \in d} w_j^{d,s} \ ,\ j = 1,...,m \ , \tag{21}$$

where, $\left| d \right|$ denotes the number of sentences in the document d, $w_j^{d,s}$, as the weight of j-th word in the sentence s included in the document d .

D) Hybrid method. As a final classification method, a linear combination of the results obtained by means of formulas (12), (18) and (19) is used:

$$\mathrm{score}(d^{\mathrm{new}} \mid C_q) = \gamma_1 \cdot \mathrm{score}_{k\mathrm{NN}}(d^{\mathrm{new}} \mid C_q) + \gamma_2 \cdot \mathrm{score}_{\mathrm{Bayes}}(d^{\mathrm{new}} \mid C_q) + \gamma_3 \cdot \mathrm{score}_{\mathrm{RG}}(d^{\mathrm{new}} \mid C_q), \tag{22}$$

where, $0 \le \gamma_i \le 1$, $(i = 1,2,3)$ denote weight coefficients and provide the following condition:

$$\gamma_1 + \gamma_2 + \gamma_3 = 1 \tag{23}$$

Thus, the document d^{new} belongs to a class C_{k^*} such that it has the highest $\text{score}(d^{\text{new}} \mid C_q)$ and may have the new value $d^{\text{new}} \in C_{k^*}$ for the class, where $k^* = \arg\max_q \text{score}(d^{\text{new}} \mid C_q)$.

6) Evaluation

Accuracy, precision, recall, and the F-measure are used to evaluate the classification:

$$\text{Accuracy} = \frac{T_p + T_n}{T_p + T_n + F_p + F_n}, \tag{24}$$

$$\text{Precision} = \frac{T_p}{T_p + F_p}, \tag{25}$$

$$\text{Precision} = \frac{T_p}{T_p + F_p}, \tag{26}$$

$$\text{F} - \text{measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{27}$$

where, T_p denotes the number of precisely classified terror-related documents; F_p is the number of incorrectly classified terror related documents; T_n is the number of precisely classified non-terror-related documents; F_p is the number of incorrectly classified non-terror-related documents.

Conclusion and future work

It is known that the text-mining technique provides great opportunities for text analysis and identification. Studies show that the technique is widely applied and currently implemented successfully. The article examined the application of the technique to ensure the security of e-government and proposed a new text mining-based integrated approach to detect terror-related documents in e-government. There are still a few problems to be resolved:

- ✓ establishing a semantic network for examined language;
- ✓ slang words used by the criminal groups for communication;
- ✓ grammatically correct spelling of words;
- ✓ texts in different languages;
- ✓ extracting terms from the text;
- ✓ selecting classification methods and guaranteeing accuracy.

All these factors have a direct impact on the accuracy of the proposed method. Another important issue is that the classes (topics) should be known in advance, although this is not always the case because the classes are dynamic. Over time, new issues may arise, so the most correct approach is to group the documents automatically and then identify them. All of these are avenues for further studies. Most of the issues (for example, developing the rules for the automatic morphological analysis of words and establishing a semantic network of words) are multidisciplinary.

References

1. Alruily M., Ayesh A., Al-Marghilani A. Using self organizing map to cluster arabic crime documents, Proceedings of the International Multiconference on Computer Science and Information Technology, Wisla, Poland, 18–20 October, 2010, pp.357–363.
2. Bsoul Q., Salim J., Zakaria L.Q. An intelligent document clustering approach to detect crime patterns, Procedia Technology, 2013, vol.11, pp.1181–1187.
3. Choi D., Ko B., Kim H., Kim H. Text analysis for detecting terrorism-related articles on the web, Journal of Network and Computer Applications, 2014, vol.38, pp.16-21.
4. Ku C.-H., Leroy G. A crime reports analysis system to identify related crimes, Journal of the American Society for Information Science and Technology, 2011, vol.62, no.8, pp.1533–1547.
5. Ku C.-H., Leroy G. A decision support system: automated crime report analysis and classification for e-government, Government Information Quarterly, 2014, vol.31, no.4, pp.534–544.
6. Yildiz M. E-government research: reviewing the literature, limitations, and ways forward, Government Information Quarterly, 2007, vol.24, no.3, pp.646–665.
7. Zhao J.J., Zhao S.Y., Zhao S.Y. Opportunities and threats: security assessment of state e-government websites, Government Information Quarterly, 2010, vol.27, no.1, pp.49–56.
8. Wimmer M., Codagnone C., Janssen M. Future e-government research: 13 research themes identified in the eGovRTD2020 project, Proceedings of the 41st Hawaii International Conference on System Sciences, Hawaii, USA, 7–10 January, 2008, pp.1–11.
9. Linders D. From e-government to we-government: defining a typology for citizen coproduction in the age of social media, Government Information Quarterly, 2012, vol.29, no.4, pp.446–454.
10. Aliguliyev R.M. Role of text mining in national security, Problems of Information Technology, 2013, no.1, pp.38–43. (in Russian)
11. Aggarwal C.C., Zhai C.X. Mining text data. Springer New York Dordrecht Heidelberg London. 2014.
12. www.idc.com
13. Miller G.A. WordNet: a lexical database for English, Communications on the ACM, 1995, vol.38, no.11, pp.39-41.
14. Wu Z., Palmer M. Verb semantics and lexical selection, Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, New Mexico, USA, 27–30 June, 1994, pp.133–138.
15. Keselj V., Peng F., Cercone N., Thomas C. N-gram based author profiles for authorship attribution, Proceedings of the Conference of the Pacific Association for Computational Linguistics, Nova Scotia, Canada, August 22–25, 2003, pp.255–264.
16. Last M., Markov A., Kandel A. Multi-lingual detection of terrorist content on the web, Lecture Notes in Computer Science, 2006, vol.3917, pp.16–30.
17. Shapira B., Last M., Elovici Y., Kandel A., Zaafrany O. Using data mining techniques for detecting terror-related activities on the web, Journal of Information Warfare, 2003, vol.3, no.1, pp.17–28.
18. Sharef N.M., Martin T. Evolving fuzzy grammar for crime texts categorization, Applied Soft Computing, 2015, vol.28, pp.175–187.
19. ru.wikipedia.org/wiki/Коэффициент_Симпсона#cite_note-2
20. Abdi A., Idris N., Alguliev R.M., Aliguliyev R.M. Automatic summarization assessment through a combination of semantic and syntactic information for intelligent educational systems, Information Processing & Management, 2015, vol.51, no.4, pp.340–358.
21. Lin D. An information-theoretic definition of similarity, Proceedings of the Fifteenth International Conference on Machine Learning, 1998, pp.296–304.

22. Zhao L., Wu L., Huang X. Using query expansion in graph-based approach for query-focused multi-document summarization, Information Processing & Management, 2009, vol.45, no.1, pp.35–41.
23. Alguliev R.M., Aliguliyev R.M., Mehdiyev C.A. Sentence selection for generic document summarization using an adaptive differential evolution algorithm, Swarm and Evolutionary Computation, 2011, vol.1, no.4, pp.213–222.
24. Aliguliyev R.M. A new sentence similarity measure and sentence based extractive technique for automatic text summarization, Expert Systems with Applications, 2009, vol.36, no.4, pp.7764–7772.
25. Li Y., McLean D., Bandar Z.A., O'shea J.D., Crockett K. Sentence similarity based on semantic nets and corpus statistics, IEEE Transactions on Knowledge and Data Engineering, 2006, vol.18, no.8, pp.1138–1150.
26. Aliguliyev R.M. Effective summarization method of text documents, Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, France, September 19-22, 2005, pp.264–271.
27. Devroye L., Gyorfi L., Lugosi G. A probabilistic theory of pattern recognition, Springer, 1996.