ISSN 0361-7688, Programming and Computer Software, 2015, Vol. 41, No. 6, pp. 336–349. © Pleiades Publishing, Ltd., 2015.
Original Russian Text © N.A. Astrakhantsev, D.G. Fedorenko, D.Yu. Turdakov, 2015, published in Programmirovanie, 2015, Vol. 41, No. 6.

Methods for Automatic Term Recognition in Domain-Specific Text Collections: A Survey

N. A. Astrakhantsev, D. G. Fedorenko, and D. Yu. Turdakov

Institute for System Programming, Russian Academy of Sciences,
ul. Solzhenitsyna 25, Moscow, 109004 Russia
e-mail: astrakhantsev@ispras.ru, fedorenko@ispras.ru, turdakov@ispras.ru

Received April 6, 2015

Abstract—Applications related to domain specific text processing often use glossaries and ontologies, and the main step of such resource construction is term recognition. This paper presents a survey of existing definitions of the term and its linguistic features, formulates the task definition for term recognition, and analyzes presently-available methods for automatic term recognition, such as methods for candidates collection, methods based on statistics and contexts of term occurrences, methods using topic models, and methods based on external resources (such as text collections from other domains, ontologies, and Wikipedia). This paper also provides an overview of standard methodologies and datasets for experimental research.

DOI: 10.1134/S036176881506002X

1. INTRODUCTION

Term recognition is essential for many applications related to domain-specific texts processing like, for example, machine translation, information retrieval, and document classification. To process texts of a particular domain, the application generally uses a glossary or ontology of this domain, and the first step of constructing the glossary is term recognition [1].

Presently, there are great many methods for automatic term recognition, and the number of researches in this field is increasing. Therefore, it is difficult to overrate the value of surveys for current and future investigations, especially taking into account the insufficient formalization of the concepts "term" and "domain," as well as differences in the corresponding methods at the level of task definition. Nevertheless, the major part of present surveys on this topic either consider the term from the linguistic or philosophic perspectives, thus ignoring term recognition methods, or confine themselves to a detailed analysis of these methods.

This survey is intended to reconcile these contradictory views by analyzing both the basic concepts of automatic term recognition and the presently-available methods for solving the term recognition problem. The paper is organized as follows. Section 2 presents an overview of the existing definitions for the concepts "term" and "domain." The present surveys, including experimental comparisons of term recognition methods, are discussed in Section 3. Methods for term recognition are analyzed in Section 4. Section 5 considers approaches to experimental investigation of the methods and presents a brief overview of open datasets for evaluation. The potential development prospects of term recognition methods are discussed in conclusion.

2. DEFINITION OF THE TERM

The history of terminology science has more than 80 years. During this time, a great number of researches were published, and most of them discussed—in one way or another—the definition of the term. According to K. Myakshin, the ongoing discussions on this topic are due to the "manysidedness of this phenomenon" and due to the fact that the term is a "linguistic universal" [2]. However, despite the elaboration of this topic and a great number of existing definitions of the term, many scientists note that there is no common, universal definition of this phenomenon: "repeated attempts by linguists to formulate a definition of the term that would satisfy the whole research community proved to be underproductive" [2]; "the notion itself of term is still not clear, both from a pure linguistic and a computational point of view" [3]; "there is no unit that is so indefinite and has so many faces as the term, moreover there are several approaches to defining the term: some researches try to give it a rather logical definition, others try to reveal the intension of the term descriptively by assigning some features to it, yet others define the term by its opposition to a certain negative unit, still others search for contradictory procedures of term recognition to arrive at a rigorous definition of this concept, and

[finally] others try to provide, for a time, at least an operational definition of the term" [4].

Despite the diffuse boundaries between the approaches in the last quote, it seems reasonable to consider the existing definitions of the term in correspondence with these approaches. Below, we present a brief overview of the discussions concerning the status of the term, term features (including the features that differentiate the term from the other lexical constructions), operational definitions (first of all, those employed in computational linguistics), and present formulations of the term recognition problem.

2.1. Discussions on the Status of the Term

Myakshin separates the substantial and functional viewpoints on the concept "term" [2]. According to the substantial point of view, the terms are specific words and word combinations that possess a certain set of features, such as monosemy, independence from the context, neutrality, etc. The supporters of this viewpoint believe that "any word can play a role of the term" and that "terms are words in a specific function rather than specific words" [5]. This standpoint seems more logical, but it shifts emphasis to the definition of the concept "term function," which is still debated among linguists [2].

In western linguistics, the problem of the term is considered at a different angle: the main question to be answered is concerned with the relationships between the lexical unit representing the term and the concept expressed by the term.

E. Wuster, one of the founders of terminology science, believed that domains consist of sets of concepts, or mental constructs, while the terms serve as textual representations of these concepts [6]. H. Felber also separates the term and the concept denominated by the term [7], but he believes that one term can denominate several concepts, and a particular meaning of the term (a concept) depends on its position in the system of concepts. This differentiates the term from common words, the meanings of which are fully determined by the context.

According to the ISO 1087 (vocabulary of terminology) [8], the term is also defined via concepts. As correctly pointed out by J. Pearson [9], this definition can hardly be called adequate, since it almost coincides with the definition of the word, which is given in the same standard.

By the term, G. Rondeau means a combination of the notion denominated by the term and the denomination itself [10]. Rondeau also tries to separate terms and the other words, but he confines himself to a remark that terms are used in specific domains.

Having analyzed the existing definitions of the term in detail, Pearson concludes that these definitions—particularly, the attempts to separate terms from common words—are based on the assumption that terms can be recognized by intuition.

To demonstrate the fallacy of this assumption, the so-called "communication attitudes" (in which words can act like terms) are adduced to show that terms are more likely used only in some attitudes; often, it is impossible to assert with certainty that a given word resembling a term is actually used as a term.

2.2. Term Features

Definition of the term by description of its features, which usually distinguish the term from common words, is of particular interest in this work, since such features can serve as a basis for methods of automatic term recognition.

By now, a great number of term features have been formulated. In [11], Myakshin describes more than ten features. Moreover, in [11], the classification of term features according to three aspects of the term—syntactic, semantic, and pragmatic—is proposed (these aspects were suggested by A. Khayutin in [12]). Below, term features are described according to this classification.

Syntactic features, which are due to the form of the term:

1. Nominativeness: "nouns or noun-based word combinations are generally regarded as terms (specific linguistic units)" [13].

2. Normativeness: conformity with language norms.

3. Terminological invariance [12]: absence of diversity in writing and pronouncing the term, since this—Myakshin cites Khayutin—"can hinder communication between specialists, not to speak of the fact that formal difference can cause semantic differentiation" [11].

4. Motivationess, or self-explanability, of the term: "maximum correspondence between the structure of the term and the intensional structure of the concept expressed by the term" [11]. It should be noted that some terminologists believe that the inverse feature is valid, i.e., the undeducibility of term meanings from constituent parts of the term. This opinion, however, is less common, since the absence of motivationess results in the absence of systemacy (see below).

Semantic features, which are due to the intention of the term:

1. Systemacy: the term belongs to a certain terminology, i.e., to a system of concepts of a specific domain or field of knowledge.

2. Correspondence to the concept denominated: the absence of contradictions between the lexical meaning of the words constituting the term and the meaning of the term in a given terminology (domain).

3. Unambiguity, or monosemy, of the term: uniqueness of the term in a given terminology (domain).

4. Intensional exactness: exactness and boundedness of the term meaning.

Pragmatic features, which are due to the specificity of the term behavior:

1. Introducability, or commonness, or understandability, or acceptability, or internationality: taking into account the number of synonyms, the definition seems to be unnecessary; note only that many researchers regard this feature as "the most systemically important criterion."

2. Definiteness: since the intensional exactness of the term (see above) is generally achieved by finding a scientific definition, the definition itself can serve as a feature of the term.

3. Independence from the context: this feature follows from monosemy of the term; it can be said that the terminology to which the term belongs serves as a context of the term, which defines its meaning.

4. Variational stability: repeatability of words and word combinations that form the term in texts of a given domain, i.e., high term frequency in these texts.

5. Euphony: convenient pronunciation and absence of undesirable associations.

2.3. Operational Definitions of the Term

Starting from the 1970s, "the view has become increasingly popular, according to which the term is a word or word combination that denominates a concept of a certain field of knowledge or activity" [2]; this definition formed the basis for most of works in the field of term recognition.

This definition, however, can hardly be called comprehensive; it is rather an operational definition in terms of Komarova, which also leaves a number of questions unanswered.

The main question: what is the "field of knowledge or activity", or the "domain" as a more common synonym? Note that, even if one does not try to define the concept "domain" and regards it as intuitive, a practical question arises: how to find out (verify) whether a given concept is specific to a particular domain?

As a rule, in present works on automatic term recognition, the question of whether the concept denominated by the term is specific to a particular domain remains under the jurisdiction of experts in the corresponding domain. As a task formulation, the guides containing the most important term features and examples are often written for the experts [14, 15]. Since these examples and many of the features characterize only a particular domain, the definition of the term becomes dependent on it.

Some researchers [16] extend the concept "domain specificity" to "domain relevancy": the term "medical negligence" maybe not specific to the domain "jurisprudence" but is surely relevant to it. This allows one to avoid the most complex problem: analyzing the concepts at the relative boundary of the domain by automatically regarding them as correct terms.

In other works (for example, [17]), the concept "specificity level" is introduced and the emphasis is placed on the terms of "average specificity." The authors confine themselves to several examples of different specificity while assuming its intuitive comprehensibility.

Sometimes, the concept of specificity is applied to domains rather than to terms [14]: the domain "emergency protective circuit arrangements" is more specific (in [14], the term "narrow" is used) than "electricity", which, in turn, is more specific than the domain "technology." The authors suggest analyzing the latter (widest) domain.

The analysis of only average-specific terms and wide domains makes it possible, first, to reduce the requirements for the level of expertise in the domain, as well as to improve the coordination of expert actions, and, second (the most important), to increase the effectiveness of applications that use recognized terms, since different applications require terms of different specificity. For example, for the problems of expert search and keyphrase extraction, terms of lower specificity are required as compared to those for the problem of ontology enrichment.

Thus, applications can impose additional constraints on terms; in other words, practically speaking, the definition of the term depends on the application, which was noted, for example, by G. Bernier-Colborne and P. Drouin [15]. In particular, this dependence was confirmed in the following experiment [18]: four groups of users (terminologists, domain experts, translators, and information scientists) were suggested to manually extract terms from a document collection; as a result, four lists of terms were made that considerably differed in the number and type of terms.

2.4. Scenarios of Term Recognition

It is convenient to analyze the dependence of the term on the application by passing to the practical level: problem formulation or term recognition scenario. Explicit separation and explicit formulation of term recognition scenarios will also provide a more adequate comparison of the present methods.

So, depending on the application, the following categories of term recognition scenarios, or formulations of the term recognition problem, can be distinguished:

1. According to the interpretation of term frequency:

(a) scenarios that consider (classify) each individual occurrence of the term;

(b) scenarios that do not distinguish between occurrences of one term.

2. According to the number of terms to be recognized:

(a) scenarios that recognize a predetermined number of terms;

(b) scenarios in which the number of terms to be recognized is determined by the algorithm for each input collection.

3. According to the length of a term candidate:

(a) scenarios that recognize one-word terms only;

(b) scenarios that recognize two-word terms only;

(c) scenarios that recognize multi-word terms only;

(d) scenarios that recognize terms of any length.

There are many more types of scenarios, but most of the present methods for term recognition fit into this categorization.

3. PRESENT SURVEYS AND EXPERIMENTAL COMPARISONS

One of the first surveys on term recognition [19] analyzes two directions: automatic indexing and term recognition itself. This survey is focused on the TF-IDF methods. The authors are among the first to introduce the aspects of the term—unithood (word relations in multi-word terms) and termnhood (relatedness of the term to the domain)—and analyze term recognition methods according to the aspect which is characteristic of the corresponding method. This survey also separates two classes of methods: linguistic and statistical.

M. Pazienza et al. [3], however, note that the present works regard linguistic methods as sets of filters and do not explicitly distinguish between these classes. In [3], the emphasis is placed on word association measures (Dice Factor, z test, t test, χ^2 test, MI, MI^2, MI^3, and likelihood ratio) and on the simplest methods for determining domain specificity of the term (term frequency, C-value, and co-occurrence).

The experimental comparison carried out in [20] shows that, despite the fact that word association measures are based on the theory of mathematical statistics, their efficiency is comparable to that of the standard term frequency.

Z. Zhang et al. [21] experimentally compared the following methods, which are capable of recognizing both one-word and multi-word terms: TF-IDF [22], Weirdness [23], C-value [24], Glossex [25], and TermExtractor [26]. The authors report that the results differed depending on the datasets used, despite the relative affinity of the domains "biomedicine" and "zoology." Moreover, this survey demonstrates the superiority of the voting algorithm (see Section 4.8) as a method that combines several features.

P. Braslavskii and E. Sokolov [27] compared four methods for recognition of two-word terms: term frequency, t test, χ^2 test, and likelihood ratio. The authors report that the first two methods showed the best (comparable) results; they also point to the main type of errors: "recognition of common collocations that satisfy some patterns."

In the recent work [28], the same authors compared five methods for recognizing terms of arbitrary structure: MaxLen [29], C-value [24], k-factor [30], Window [31], and AOT [32]. The comparison showed that "the methods generally yield similar results"; however, the authors note that the C-value and the k-factor have the highest efficiency, while the AOT has the lowest efficiency. For efficiency evaluation, the combination of the expert evaluation and the formal evaluation according to a preselected vocabulary ("reference list") was used; the important conclusion is as follows: "formal methods [of efficiency evaluation] are suited for comparison between large lists of term candidates."

In [33], two methods based on combination of several features are compared—voting algorithm and method based on supervised machine learning (logistic regression and Random forest)—and the conclusion is made that the second method outperforms the first one.

M. Nokel and N. Loukachevitch [34] compared methods for recognizing one-word and two-word terms for the problem of thesaurus construction and information retrieval. The authors used the gradient boosting algorithm and analyzed most of the known features.

Based on results of experimental comparison, the following four important conclusions were made:

(1) the best features for recognition of one-word terms are based on topic models;

(2) in all cases, the combination of several features yields a considerable increase in efficiency as compared to the use of individual features;

(3) features based on the external corpus offer the most significant increase in efficiency for recognition of two-word terms;

(4) word association measures provide no increase in efficiency.

4. TERM RECOGNITION METHODS

For the majority of term recognition methods, the following general scheme:[1] is applicable:

1. Candidates collection: filtration of words and word combinations, which are extracted from the document collection, according to statistical and linguistic criteria.

2. Feature computation: transformation of each term candidate into a vector of the feature space.

[1] It should be noted that this scheme corresponds to the scenario that does not distinguish between occurrences of one term.

3. Feature-based inference: estimation of the probability of being the term for each candidate on the basis of feature values.

In turn, methods for candidates collection also consist of several steps. At the first step, linguistic filters are applied to select only nouns and nominal groups (word combinations in which the noun is the main word) according to the feature "nominativeness" (see Section 2.2). For this purpose, either shallow parsing (chunking) [14] or, more frequently, N-gram filtration according to predefined part-of-speech patterns [24, 33, 35, 36] is performed.

At the next steps of candidates collection, to reduce the noise, additional filtration is performed:

(1) according to term frequency: candidates with the number of occurrences less than 2 or 3 are generally eliminated, since many statistical features become inapplicable;

(2) according to stop words from a preset list [36]: many words, such as "good" or "interesting," are quite rarely included into terms while having a high frequency of occurrence (for example, "good method");

(3) according to the length of candidates or special symbols contained in candidates [37]: non-alphabetic symbols and words composed of one letter are generally eliminated.

The second stage (computation of features for term candidates) is of special interest and is considered below (see subsections 4.1–4.7) in detail.

The third stage is considered in subsection 4.8.

The difference between the terms "feature" and "method" should be clarified: by the feature, a mapping of a candidate into a certain number is usually meant; by the method, we mean a sequence of actions to obtain a ranked list of candidates for a given document collection, which involves calculating one or several features. Nevertheless, in practice, these two terms are often used interchangeably, since any method can be regarded as a feature and, in turn, most of features were originally developed as individual methods; moreover, the method has a more general meaning: a way of calculating the feature. In this paper, the terms "feature" and "method" are also used interchangeably provided that the ambiguity will not result.

4.1. Methods Based on Statistics of Term Occurrences

This subsection describes methods that take into account only the frequency of candidate occurrences in the document collection and, sometimes, components of these occurrences.

The first, simplest, and relatively effective method is the term frequency (TF). Taking into account the obviousness of this feature, it does not seem possible to determine the authorship.

The TF-IDF method, a classical method for information retrieval, can also be classified as an early feature:

$$TF \cdot IDF(t) = TF(t) \cdot \log \frac{1}{TF_d(t)}, \qquad (1)$$

where $TF_d(t)$ is the number of the documents containing the term candidate t.

This feature has high values for the terms that frequently occur only in a small number of documents. D. Evans and R. Lefferts [22] were among the first to use this feature for term recognition.

It is interesting to note that the opposite (in some sense) feature—Domain Consensus—is also used [38], which is designed for recognition of terms uniformly distributed over the whole collection:

$$DC(t) = -\sum_{d \in Docs} \frac{TF_d(t)}{TF(t)} \log_2 \frac{TF_d(t)}{TF(t)}. \qquad (2)$$

A separate group called "word association measures" is formed by the features that estimate how strong the words constituting the term are related (unithood) or how random is the combination of these words.

Since these methods can be applied only to multiword terms (often, only to two-word terms) and taking into account that these methods were shown [20, 34] to provide no increase in efficiency, we confine ourselves only to mentioning the most common methods of this group: z test [39], t test [40], χ^2 test, likelihood ratio [41], mutual information (MI [42], MI^2, and MI^3 [43]), lexical cohesion [44], and term cohesion [25].

The most popular feature—C-value [24]—is also classified among the methods based on statistics of term occurrences:

$$C\text{-}Value(t) = \begin{cases} \log_2 |t| \cdot f(t) \\ \text{if } \{s : t \subset s\} = \emptyset; \\ \log_2 |t| \cdot \left(f(t) - \dfrac{\sum_s f(s)}{|\{s : t \subset s\}|} \right) \\ \text{otherwise}, \end{cases} \qquad (3)$$

where t is the term candidate, $|t|$ is the length of the candidate t (in words), $f(t)$ is the frequency of t in the text collection, and s is the set of the candidates that enclose the candidate t, i.e., the candidates such that t is their substring.

In this feature, the weight of the candidate is reduced if this candidate is a part of other candidates, since the candidate frequency in this case is added to the frequency of enclosing candidates: for example, the frequency of the word combination *point arithmetic* is not less than that of the term *floating point arithmetic*, although the former is obviously not a term.

It should be noted that the C-value is meant for recognition of multi-word terms only: otherwise, the expression under logarithm sets the feature value to zero.

In [36], the C-value is generalized to the case of one-word terms by adding a constant to the logarithm:

$$C\text{-}Value(t) = \begin{cases} c(t) \cdot TF(t) \\ \text{if } \{s : t \subset s\} = \emptyset; \\ c(t) \cdot \left(TF(t) - \dfrac{\sum_s TF(s)}{|\{s : t \subset s\}|} \right) \\ \text{otherwise}, \end{cases} \quad (4)$$

where $c(t) = i + \log_2|t|$. The authors report that the value $i = 0.1$ was initially used to minimize the distortion of the original formula; however, during the experiments, it was found that the value $i = 1$ offers the best efficiency.

J. Ventura et al. [35] suggest adding 1 before taking the logarithm:[2]

$$C\text{-}Value(t) = \begin{cases} \log_2(|t|+1) \cdot TF(t) \\ \text{if } \{s : t \subset s\} = \emptyset; \\ \log_2(|t|+1) \cdot \left(TF(t) - \dfrac{\sum_s TF(s)}{|\{s : t \subset s\}|} \right) \\ \text{otherwise}. \end{cases} \quad (5)$$

G. Bordea et al. [17] propose the method called Basic[3] which is a modification of the C-value for recognizing terms of average specificity:

$$Basic(t) = |t| \log f(t) + \alpha e_t, \quad (6)$$

where e_t is the number of the candidates enclosing the candidate t.

Just as the C-value, the Basic is applied only to multi-word terms. However, in contrast to the C-value (in which the frequency of a candidate is reduced if it is part of other candidates), in the Basic, the candidates that contain a given candidate increase its feature value, since average-specific terms are often used to form more specific terms.

As an example, the authors suggest the term information retrieval, which can be used to construct more specific terms such as *information retrieval system*, *information retrieval metric*, etc.

It should be noted that the Basic is a part of the method called PostRankDC (see the following subsection).

[2] Note that this is somewhat close to the Laplace smoothing.

[3] This name is used in the recent work [37]; in the original work [17], this method was called Baseline.

4.2. Methods Based on Contexts of Term Occurrences

Methods of this group, particularly, the NC-value [24], are based on the assumption that the contexts of terms and common words are different. By the context, following G. Grefenstette, the authors of the NC-value mean nouns, verbs, or adjectives that immediately precede or follow the term.

The feature computation consists of three steps. At the first step, the best 200 terms are recognized using the C-value; however, the authors note that any other method can be used, including manual labeling.

At the second step, weights of context words are calculated by the formula

$$weight(w) = \frac{t(w)}{n}, \quad (7)$$

where w is the context word (noun, verb, or adjective); $t(w)$ is the number of terms, in the context of which the context word occurs (not to be confused with the term frequency); and n is the total number of terms.

At the third step, the final value is calculated by the formula

$$NC(t) = 0.8 \cdot C\text{-}Value(t) + \sum_{w \in C_t} f_t(w) weight(w), \quad (8)$$

where t is the term candidate, C_t is a set of the words occurring in the context of the candidate t, w is the word from C_t, and $f_t(w)$ is the frequency of the word w in the context of the candidate t.

In [17], the method called DomainCoherence, which is a modification of the NC-value for the case of average-specific terms, is proposed.

The authors impose the following constraints, which are called the "domain model," on context words:

(1) occurrence in at least a quarter of the input document collection;

(2) belonging to nouns, verbs, or adjectives;

(3) semantic relatedness to many specific terms.

The last constraint is essentially a way of weighting; in contrast to the ordinary computation of relations between the terms before or after which the word occurs (NC-value), the DomainCoherence uses the PMI metric:

$$s(w) = \sum_{t \in T} PMI(t, w) = \sum_{w \in W} \log\left(\frac{P(t, w)}{P(t)P(w)} \right), \quad (9)$$

where w is the word regarded as a candidate for the domain model, T is the set of the best 200 terms recognized by the Basic (see the previous subsection), $P(t, w)$ is the probability that the word w occurs in the context of the term t, and $P(t)$, $P(w)$ are the probabilities of occurrence of the term t and the word w, respectively. These probabilities are estimated based on the

term frequency in the input document collection; the window comprising five words is regarded as a context.

To find the final value of the DomainCoherence, the PMI metric is also used, which is calculated between each term candidate (t) and the word from the domain model (w).

In addition, the authors report that, during the experimental research, the best results were shown by a linear combination of the Basic and DomainCoherence, which was called PostRankDC.

4.3. Methods Based on Topic Models

Owing to the progress in methods for topic modeling (word clustering according to topics and topic clustering according to documents), in recent years, several features were developed for term recognition on the basis of topic modeling. Nokel and Loukachevitch [34] note that the majority of features are modifications of the standard methods that use the probability distribution by topics of words (term candidates) instead of the term frequency. In particular, this implies that such methods can be applied only to recognition of one-word and (more rarely) two-word terms.

Such features are Term Score, which was originally developed for topic visualization [45] and was used by E. Bolshakova et al. for term recognition [46]; term frequency; maximum term frequency; TF-IDF; and Domain Consensus [46].

S. Li et al. [47] propose the method called Novel Topic Model, which is capable of recognizing term of any length. To calculate this feature, one needs distributions of words over the following topics:

- ϕ^t, particular topics of the domain ($1 \leqslant t \leqslant T$; the authors set $T = 20$);
- ϕ^B, background topic; and
- ϕ^D, topic specific to the document.

Then, the most probable 200 words are recognized for each topic (V_t, V_B, and V_D), and the weight of each candidate c_i, which consists of L_i words ($w_{i1}\, w_{i2} \ldots w_{iL_i}$), is taken as a sum of maximum probabilities of the words constituting this candidate (from the distributions found):

$$NTM(c_i) = \log(if_i) \cdot \sum_{1 \leq j \leq L_i,\ w_j \in \cup\{V_t\}_{t \in T \cup \{B,D\}}} \phi_{w_j}^{mt_{w_j}}, \quad (10)$$

where

$$mt_{w_j} = \arg\max_{t \in T \cup \{B,D\}} \phi_{w_j}^{t}.$$

4.4. Methods Based on External Corpora

Methods of this group are based on the observation that terms of a certain domain occur in texts of this domain much more frequently than in texts of other domains, particularly, in texts of the so-called general domain or texts that do not belong to any domain.

Such texts, which are usually called the external or reference corpus, include document collections of other domains [26, 48], sets of electronic books and magazines [17], news collections [49], and corpora created by linguists, like, for example, Open American National Corpus[4] (OANC) [17] and British National Corpus[5] (BNC) [23].

One of the simplest ways to implement the observation mentioned above is to modify the TF-IDF [50], which is sometimes called TF-RIDF [34]: when calculating the number of documents in which the term occurs (IDF (RIDF)), the external corpus is used instead of the domain collection.

The feature called Domain Pertinence [48] is also based on a simple formula

$$DR(t) = \frac{TF_{target}(t)}{TF_{reference}(t)}, \quad (11)$$

where $TF_{target}(t)$ is the frequency of the candidate t in the input domain-specific document collection and $TF_{general}$ is the frequency in the general corpus.

The feature called Domain Relevance [26] uses a similar formula

$$DR(t) = \frac{TF_{target}(t)}{TF_{target}(t) + TF_{reference}(t)}. \quad (12)$$

The feature called Weirdness [23] additionally takes into account the size of the collection:

$$W(t) = \frac{TF_{target}(t) \cdot |Corpus_{reference}|}{TF_{reference}(t) \cdot |Corpus_{target}|}. \quad (13)$$

The feature called Relevance [49] reduces the weight of the candidates that rarely occur in documents of a given domain, or occur in a very small percentage of documents, or frequently occur in the external corpus:

$$Rel(t) = 1 - \frac{1}{\log_2\left(2 + \frac{TF_{target}(t) \cdot DF_{target}(t)}{TF_{reference}(t)}\right)}. \quad (14)$$

The feature called Domain Specificity, which is a part of the GlossEx system [25], takes into account frequencies of the individual words constituting the term:

$$DomainSpecificity(t) = \frac{\sum_{w_i \in t} \log \frac{P_d(w_i)}{P_c(w_i)}}{|t|}, \quad (15)$$

where $|t|$ is the number of words in the candidate t; $P_d(w_i)$ is the probability that the word w_i, which is part of the candidate t, occurs in the domain-specific text

[4] http://www.americannationalcorpus.org.
[5] http://www.natcorp.ox.ac.uk.

collection; and $P_c(w_i)$ is the probability that the word occurs in the external corpus. As in the other methods, the probability is estimated as a number of occurrences normalized by the size of the collection in words.

4.5. Methods Based on Retrieval Engines

A separate group is formed by methods that use retrieval engines. In [51], for recognition of two-word terms, several features—iFreq, TF-IDF, freq/iFreq, and coherence—were used.

The authors report that these methods are applicable not to every domain and put forward a hypothesis that "the method will be more likely applicable to domains with specific terminology (which differs for the most part from common expressions and rarely uses expressions for universal concepts), which are not presented too widely in the Web."

For filtration of two-word terms, D. Golomazov [52] constructs the following requests to retrieval engines: "A" (the term itself), "A is a term," "A is a concept," "A_1," "A_2," and "A_1 AND A_2," where A_1 and A_2 are the words of which the term A is composed.

Then, for the term to pass the filtration, at least one of the following conditions[6] must be fulfilled:

$$(1)\ \frac{hits(\text{“}A \text{ is a term”})}{hits(A)} > C_1,$$

$$(2)\ \frac{hits(\text{“}A \text{ is a concept”})}{hits(A)} > C2,$$

$$(3)\ \frac{hits(\text{“}A_1 \text{ AND } A_2\text{”})}{\min(hits(A_1), hits(A_2))} > C_3,$$

where $hits(A)$ is the number of pages returned by the retrieval engine on the request A and $C_1, C_2, C_3 \in [0, 1]$ are the parameters of the algorithm.

B. Dobrov and N. Loukachevitch [53] also consider the recognition of only two-word terms with the use of retrieval engines. However, instead of the frequency of candidate occurrence in the Web, the authors extensively use snippets returned on the request consisting of the whole candidate and individual words of the candidate. In particular, the authors analyze candidate frequencies in snippets (FreqBySnip); the number of predetermined words that are characteristic of the domain (Markers) in snippets; the number of words that frequently occur in dictionary definitions (NearDefWords); relatedness of the snippets obtained for the whole candidate and for its individual words (Scalar Features); and so on.

In [53], other types of features are also considered and, based on experimental results, the conclusion is made that the maximum efficiency is reached when using all types of features.

[6] Strictly speaking, there can be one more condition described in subsection 4.7.

4.6. Methods Based on Ontologies

For the problem of term recognition, ontologies are used more rarely than other external resources, since general ontologies insufficiently cover domains and include only the most general terms. Moreover, domain-specific ontologies are available only for a few domains, and the format and structure of such ontologies often depend on a particular domain.

The majority of works devoted to ontology enrichment are focused on extraction of relations between concepts and do not use (at the term recognition stage) any information contained in the ontology being enriched. For example, K. Meijer et al. [48] use the above-described features Domain Consensus, Domain Pertinence, and Lexical Cohesion; F. Xu et al. [54] use the TF-IDF and various word association measures (*MI*, *t* test, and likelihood ratio).

Here, the work [53] by Dobrov and Loukachevitch (see above) is worth mentioning, in which an actual domain ontology is used (particularly, a thesaurus for information retrieval). The features proposed in this work, however, can be used only for two-word terms. These are the binary feature SynTerm, which equals to one if and only if, for each word constituting the term, there is a synonym in the thesaurus, and the feature Completeness, which sums up the synonyms and relations for descriptors, which, in turn, are also found in the thesaurus for individual words of the term.

4.7. Methods Based on Wikipedia

Wikipedia, a multi-language Internet encyclopedia, possesses some unique characteristics that can be used for term recognition: its articles describe both universal and specific concepts for narrow domains, and their coverage increases permanently; Wikipedia contains structural information in the form of hyperlinks between its articles; and it is very large and is daily updated by the user community.

In [55], D. Milne et al. compare the coverage of the domain "agriculture" by Wikipedia and by the Agrovoc thesaurus created by experts.[7] The authors show that about half of all thesaurus terms, including the most common ones, are contained in Wikipedia. Note that, over the past eight years, Wikipedia considerably grew in size: by the time of writing this paper, the number of English Wikipedia articles exceeded 4.6 million (in 2006, this number was about 1.1 million), and there is reason to believe that its size will increase far beyond that.

However, despite the extensive use of Wikipedia for solving various problems of knowledge mining [56–58], so far, there are few methods developed for term recognition based on Wikipedia.

[7] http://aims.fao.org/agrovoc.

In [52], Wikipedia is used only for filtration of terms: if there is a Wikipedia article that describes a given term, then the term passes filtration.

In [16], terms are recognized only in Wikipedia, rather than in domain-specific text collections, and it is required to manually select several concepts (Wikipedia articles) as positive examples of domain-specific terms.

More precisely, the authors construct a weighted graph, in which nodes are Wikipedia articles and categories, while edges are hyperlinks between them. Then, using manually selected concepts, a random walk algorithm is applied to the graph. The weight assigned by the algorithm to each concept is taken as an estimate that the corresponding concept is expressed by a domain-specific term.

J. Vivaldi et al. [59, 60] propose the following method: in a domain-specific text collection, term candidates are recognized and, then, are estimated by applying path searching algorithms to the graph of Wikipedia categories. As in [16], additional input information must be specified: one or several Wikipedia categories (called domain borders) that describe a given domain as precisely and comprehensively as possible.

The algorithm for estimating term candidates is as follows. For each candidate, all its concepts are found, i.e., Wikipedia articles with the same title (generally, there can be several articles for one candidate, which is due to lexical polysemy); then, for each article, all categories are determined to which this article belongs. From all estimates obtained, the best one is selected for each term candidate.

Next, for each category, the graph of categories is recursively traversed (following only the links to the top-level category) until the specified domain border or the topmost-level category is reached. Finally, the properties of the paths found are used to estimate term candidates based on one of the following criteria.

1. The number of paths (NC)

$$NC(t) = \frac{NP_{domain}(t)}{NP_{total}(t)}, \quad (16)$$

where $NP_{domain}(t)$ is the number of paths from the categories of the candidate to the domain border and $NP_{total}(t)$ is the number of paths from the categories of the candidate to the top-level category.

2. Length of paths (LC)

$$LC(t) = \frac{LP_{total}(t) - LP_{domain}(t)}{LP_{total}(t)}, \quad (17)$$

where $LP_{domain}(t)$ is the (total) length of paths from the categories of the candidate to the domain border and $LP_{total}(t)$ is the (total) length of paths from the categories of the candidate to the top-level category.

3. Average length of paths (LMC)

$$LC(t) = \frac{ALP_{total}(t) - ALP_{domain}(t)}{ALP_{total}(t)}, \quad (18)$$

where $ALP_{domain}(t)$ is the average length of paths from the categories of the candidate to the domain border and $ALP_{total}(t)$ is the average length of paths from the categories of the candidate to the top-level category.

Based on results of the experimental research carried out for one- and two-word terms, the authors report that the NC criterion possesses the maximum efficiency.

In [61], two methods are proposed that use Wikipedia hyperlinks. The first method called LinkProbability is a normalized frequency with which the term candidate is a hyperlink in Wikipedia articles:

$$LinkProb_T(t) = \begin{cases} 0 & \text{if } t \text{ is not contained} \\ & \text{in Wikipedia or } \dfrac{H(t)}{W(t)} < T; \\ \dfrac{H(t)}{W(t)} & \text{otherwise,} \end{cases} \quad (19)$$

where t is the term candidate (word or word combination filtered according to parts of speech, frequency, and stop list); $H(t)$ shows how often the candidate t occurs in Wikipedia articles in the form of a hyperlink caption; $W(t)$ shows how often t occurs in Wikipedia in total; and T is the parameter of the method, which is used to filter too small values, since they generally are layout errors ($T = 0.018$ was selected experimentally).

The value of this feature will be close to zero for the words and word combinations that belong to the general vocabulary, i.e., are not specific to any domain. Thus, this method is useful for filtering such words and word combinations, since they most likely are not specific to the domain the terms of which are to be recognized.

The second method proposed in [61]—KeyConceptRelatedness—is based on the following interpretation of the term: "the term is a word or word combination that denominates *at least one concept*, which [in turn] is *specific to* a given *domain*," where the "concept" is interpreted as a concept contained in Wikipedia in the form of an article, the "domain" is the set of concepts closely related in their meaning, the "specificity to a domain" is the relatedness in terms of meaning to the domain, and the "relatedness in terms of meaning" is the semantic relatedness that is a function defined for any pair of concepts on the interval [0, 1] (the closer the function value is to 1, the more the concepts have in common). The explicit formulation "that denominates at least one concept" makes it possible to solve the problem of polysemy, i.e., a situation when the term candidate has several meanings (Wikipedia concepts), by selecting the closest concept from those denominated by the term. Note that such an interpretation is valid only for the scenario that does not dis-

tinguish between different occurrences of a term candidate.

In detail, the algorithm of this method is given below.

1. Find key concepts in a given document collection:

(a) recognize d key concepts in each document of the collection ($d = 3$);

(b) select N key concepts with the highest frequency ($N = 200$).

2. For a given term candidate, find all Wikipedia concepts such that their captions coincide with the term candidate.

3. For each concept found for the term candidate, calculate its semantic relatedness to the key concepts found by using the weighted kNN method adapted for the case of positive examples only:

$$sim_k(c, C_N) = \frac{1}{k}\sum_{i=1}^{k} sim(c, c_i), \quad (20)$$

where c is the concept of the term; C_N is the set of key concepts ranked in the descending order of semantic relatedness to c; $sim(c, c_i)$ is the semantic relatedness function found by the Dice formula, where the articles connected by at least one hyperlink are regarded as neighbors; and k is a constant that defines the number of the nearest concepts, which are used for calculating the resultant semantic relatedness.

4. Select the maximum value over all concepts of the term candidate.

Thus, the value of this feature will be close to zero for the words and word combinations that denominate the concepts with low relatedness to domain-specific key concepts.

4.8. Methods of Feature-Based Inference

In the case of several features, the last stage—feature-based inference—is implemented in one of the following ways.

1. Linear combination of features with manually fitted coefficients (generally, equal) [17, 26, 44].

2. Voting algorithm proposed in [21]:

$$V(t) = \sum_{i}^{n} \frac{1}{rank(F_i(t))}, \quad (21)$$

where n is the number of features and $rank(F_i(t))$ is the ordinal number of the candidate t among all candidates ranked by the value of the feature F_i.

3. Supervised machine learning: a classifier model is constructed based on manually labeled data. For the scenario in which the number of terms to be recognized can be specified, the learning algorithms that support probabilistic classification are used, i.e., the algorithms that return the class membership probability rather than a binary response for each candidate.

The following supervised learning algorithms can be pointed out: Ada Boost [62], logistic regression [33, 53, 63], Random forest [33], and Gradient Boosting [34].

The methods that are based on supervised learning algorithms but require no labeled data merit detailed consideration. In [64], Y. Yang et al. propose the method called Fault-Tolerant Learning, which is a combination of bootstrapping and co-training algorithms.

The authors separate two sets of features: standard TF-IDF and features based on word delimiters, which are specific to Chinese. Using each set, all candidates are sorted to obtain two lists (more precisely, two assortments) of candidates, which consist of the same elements. From each list, the best 500 and the worst 500 candidates are extracted, which are regarded as positive and negative examples, respectively, for subsequent supervised learning. Supervised learning algorithms are support vector machines (SVMs) with five features: candidate frequency, parts of speech for words of the candidate, word delimiters from occurrence contexts of the candidate, the first word of the candidate, and the last word of the candidate.

Trained classifiers are then applied to all term candidates, and the candidates with the maximum and minimum estimates are used again as positive and negative examples for the next training iteration. To avoid degradation of the process, the so-called verification of training sets is used: when different labels (term and non-term) are assigned by two classifiers to the same candidate, this candidate is eliminated from the training set.

A similar idea is followed in [61]: using a special term recognition method, the best S candidates are found (in the experiments, $S = 100$), which are then used as positive examples for constructing a model of the positive-unlabeled (PU) learning algorithm (which is a particular case of the semi-supervised learning algorithm). In this case, unlabeled examples are the other term candidates. The constructed model of the PU learning algorithm is used for probabilistic classification of each term candidate. As a method for recognition of positive examples, a modified Basic is used; moreover, the recognized candidates can be filtered according to their presence in Wikipedia, i.e., only the candidates for which there is a Wikipedia article of the same title are retained.

In [14], a term recognition method is proposed that classifies each occurrence of the term candidate individually, but the basic scheme can also be used for the scenario that does not distinguish between occurrences of one term. The authors use the following heuristics to recognize positive and negative examples: terms are words or word combinations that immediately precede a reference to an illustration in the text of

a patent and non-terms are words or word combinations that occur in patents only once or are either citations or units of measurement.

Having obtained a set of positive and negative examples, the authors use the supervised learning algorithm (logistic regression and conditional random fields) with 74 features, including parts of speech, contexts and statistics of occurrences, and features based on string metrics.

The results of experimental evaluation are rather high (more than 75% for the F1 measure); however, the evaluation technique differs from the commonly accepted one in that each term occurrence is classified and evaluated individually. Therefore, the same term candidate with lots of occurrences has a greater contribution to precision and recall than several candidates with a small number of occurrences; generally, the former candidates are easier to classify correctly. Another important drawback is the impossibility of transfer to other domain and other languages because of the heuristics used for recognizing positive examples.

5. METHODS FOR EFFICIENCY EVALUATION

Bernier-Colborne and Drouin [15] note that the problem of evaluating term recognition systems remains unsolved: efficiency evaluations are regularly published, but the methodology differs from paper to paper, thus making difficult any comparison.

Two principal approaches for estimating term recognition methods can be distinguished:

(1) manual evaluation with the help of experts in the corresponding domain (for example, [65]);

(2) the use of the "gold standard," i.e., a preset list of reference terms ("formal estimation" in terms of Braslavski and Sokolov [27, 28]).

Pros and cons of each approach are obvious: the first one offers the most accurate evaluation, while the second one provides reproducibility of results, tunability of parameters, and comparison between different methods on one dataset. As noted above, given sufficiently large lists of terms, these approaches yield coinciding comparison results for term recognition methods [27, 28, 51].

The second approach can be divided into several evaluation techniques depending on the way of obtaining the list of reference terms:

(1) manual labeling of all documents (for example, [14]);

(2) manual labeling of a small part of documents (for example, [63]);

(3) adaptation of available resources to the term recognition problem (for example, [37, 53]).

The first technique is most accurate but is most time consuming as well, especially given a great number of documents (a small number of documents distorts the effect of statistical features).

The second technique allows one to calculate features based on all documents and to evaluate their efficiency only for the terms that occur in labeled documents.

The applicability of the third technique depends on the corresponding domain and application. For some domains, there are manually-constructed thesauri or vocabularies that can be used as the gold standard [28, 53]. Sometimes, terms are approximated by key phrases or index terms; for example, Bordea [37] uses the union of sets of key words for each paper as reference terms for the collection of papers of one scientific field. For this purpose, subject indexes of books are also used [27, 51].

5.1. Efficiency Evaluation Metrics

The following efficiency metrics are generally used for the scenario under consideration (recognizing a given number of terms of any length in a document collection without distinguishing between occurrences of one term).

1. Precision, which is sometimes called the precision at the level N:

$$P(N) = \frac{|Correct \cap Retrived[1:N]|}{N}, \qquad (22)$$

where N is the number of the best candidates, *Correct* is the set of reference terms, and *Retrieved*$[1:N]$ is the set of the best N candidates according to the weights assigned by the methods being evaluated.

2. Recall:

$$R(N) = \frac{|Correct \cap Retrived[1:N]|}{|Correct|}. \qquad (23)$$

3. Average precision:

$$AvP(N) = \sum_{i=1}^{N} P(i)(R(i) - R(i-1)). \qquad (24)$$

It should be noted that, in practice, the recall is often evaluated implicitly, since it is actually determined by the specified number of recognized terms and by the precision, while the average precision is now being the most popular metric, since it is an integral evaluate on the set of N.

Moreover, in some works, the dependence of the precision [37] and average precision [33, 34] on N is investigated explicitly.

5.2. Datasets

Many papers devoted to term recognition, for example, [15, 34, 47],[8] use closed datasets, which

makes impossible the correct comparison with the corresponding methods.

Well-known open datasets are described below.

GENIA [66] is a collection of 2000 labeled documents on biomedicine; it is one of the most popular datasets for testing term recognition efficiency; results on this collection are presented in six papers at least [21, 33, 37, 61, 67, 68].

FAO [69] consists of 780 manually-labeled reports of the Food and Agriculture Organization (for each report, two terms were recognized). This dataset was used in [37] to test methods for automatic term recognition.

Krapivin [70] comprises 2304 papers on informatics; as a reference set of terms, key words selected by the authors of the papers are used. This dataset was also used in [37].

Patents [14] consists of 16 manually-labeled patents on electrical engineering.

Board games [63] is a collection of 1300 descriptions and reviews of board games, in which 35 documents (out of 1300) are labeled manually. For testing, it is suggested to take into account only the terms that occur in the labeled documents at least once, which allows one to use the occurrence statistics calculated on the whole collection and to evaluate the efficiency of term recognition methods as accurately as possible.

CONCLUSIONS

In this work, the presently-available methods for automatic term recognition are discussed and the definitions of the term are analyzed from both theoretical and practical perspectives. The presented survey shows that there are no commonly-accepted definitions of the concepts "term" and "domain," while the attempts to find such definitions result in a variety of inconsistent and, sometimes, contradictory formulations. The definitions used as operational ones suffer from a lack of formality.

The formulation of the term recognition problem is also far from being entirely formal, which makes it quite difficult to compare term recognition methods and to evaluate their efficiency. As a result, there are currently no commonly-accepted datasets and methodologies for efficiency evaluation, and developed methods are often domain- and application-specific.

Thus, the development of datasets, experimental research methodologies, and methods for adapting present algorithms to other domains and applications seem to be the most promising directions in the field of term recognition.

[8] It should be noted that [15] is devoted to creating a dataset for experimental investigation of term recognition methods, but the authors cannot distribute labeled data and suggest to send developed methods to them for testing.

ACKNOWLEDGMENTS

This work was supported by the Russian Foundation for Basic Research, project no. 14-07-00692.

REFERENCES

1. Astrakhantsev, N. and Turdakov, D., Automatic construction and enrichment of informal ontologies: A survey, *Program. Comput. Software*, 2013, vol. 39, no. 1, pp. 34–42.
2. Myakshin, K.A., Various approaches to definition of the concept "term," *Al'manakh Sovremennoi Nauki Obrazovaniya, ser. Yazykoznanie Literaturovedenie Sinkhronii Diakhronii Metodika Prepodavaniya Yazyka Literatury*, 2007, vol. 3, no. 3, pp. 175–178.
3. Pazienza, M., Pennacchiotti, M., and Zanzotto, F., Terminology extraction: An analysis of linguistic and statistical approaches, in *Knowledge Mining*, 2005, pp. 255–279.
4. Komarova, R.I., Term system of the heuristics sublanguage (on the material of English), *Extended Abstract of Cand. Phil. Sci. Dissertation*, Odessa, 1991, p. 18.
5. Vinokur, G.O., Grammatical observations in the field of technical terminology, *Tr. Mosk. Inst. Filosofii Literatury Istorii*, 1939, vol. 5.
6. Wüster, E., *Einfuuhrung in die allgemeine Terminologielehre und terminologische Lexikographie (1979)*, Kobenhavn: Handelshojskolen, 1985.
7. Felber, H., Basic principles and methods for the preparation of terminology standards, *Standardization of Technical Terminology: Principle and Practices*, ASTM STP, 1982, vol. 806, pp. 3–13.
8. *Terminology–Vocabulary: Standard*, CH: International Organization for Standardization, Geneva, 1990.
9. Pearson, J., *Terms in Context*, John Benjamins, 1998, vol. 1.
10. Rondeau, G., *Introduction ala terminologie*, Quebec: Gaetan Morin, 1984, 2nd ed.
11. Myakshin, K.A., On the question of main features of the term, *Al'manakh Sovremennoi Nauki Obrazovaniya, ser. Yazykoznanie Literaturovedenie Sinkhronii Diakhronii Metodika Prepodavaniya Yazyka Literatury*, 2008, vol. 2, no. 21, pp. 17–22.
12. Khayutin, A.D., Compound terms: Functional type of complex linguistic units from the perspective of lexicography, in *Otraslevaya terminologiya i leksikografiya* (Industrial Terminology and Lexicography), Voronezh: Voronezh State Pedagogical Univ., 1981.
13. Akhmanova, O.S., Linguistic terminology, *Linguistic Encyclopedic Dictionary*, Moscow: Sov. Entsikl., 1990.
14. Judea, A., Schütze, H., and Bruegmann, S., Unsupervised training set generation for automatic acquisition of technical terminology in patents, *Proc. 25th Int. Conf. Computational Linguistics: Technical Papers (COLING)*, Dublin, 2014, pp. 290–300.
15. Bernier-Colborne, G. and Drouin, P., Creating a test corpus for term extractors through term annotation, *Terminology*, 2014, vol. 20, no. 1, pp. 50–73.
16. Wu, W., Liu, T., Hu, H., et al., Extracting domain-relevant term using Wikipedia based on random walk

model, *Proc. 7th IEEE Int. Conf. Data Mining Workshops*, 2012, pp. 68–75.

17. Bordea, G., Buitelaar, P., and Polajnar, T., Domain-independent term extraction through domain modeling, *Proc. 10th Int. Conf. Terminology and Artificial Intelligence (TIA)*, Paris, 2013.
18. Bagot, R.E., Les unites de signification sprecialisrees relargissant l'objet du travail en terminologie, *Terminology*, 2002, vol. **7**, no. 2, pp. 217–237.
19. Kageura, K. and Umino, B., Methods of automatic term recognition: A review, *Terminology*, 1996, vol. 3, no. 2, pp. 259–289.
20. Wermter, J. and Hahn, U., You can't beat frequency (unless you use linguistic knowledge): A qualitative evaluation of association measures for collocation and term extraction, *Proc. 21st Int. Conf. Computational Linguistic and 44th Annu. Meet. Association for Computational Linguistic*, 2006, pp. 785–792.
21. Zhang, Z., Brewster, C., and Ciravegna, F., A comparative evaluation of term recognition algorithms, *Proc. 6th Int. Conf. Language Resources and Evaluation (LREC)*, Marrakech, 2008.
22. Evans, D.A. and Lefferts, R.G., Clarit-trec experiments, *Inf. Process. Manage.*, 1995, vol. 31, no. 3, pp. 385–395.
23. Ahmad, K., Gillam, L., Tostevin, L., et al., University of surrey participation in trec8: Weirdness indexing for logical document extrapolation and retrieval (wilder), *Proc. 8th Text Retrieval Conf. (TREC)*, 1999.
24. Frantzi, K., Ananiadou, S., and Mima, H., Automatic recognition of multi-word terms: The c-value/nc-value method, *Int. J. Digital Libr.*, 2000, vol. 3, no. 2, pp. 115–130.
25. Kozakov, L., Park, Y., Fin, T., et al., Glossary extraction and utilization in the information search and delivery system for IBM technical support, *IBM Syst. J.*, 2004, vol. 43, no. 3, pp. 546–563.
26. Sclano, F. and Velardi, P., Termextractor: A web application to learn the shared terminology of emergent web communities, *Enterprise Interoperability II*, 2007, pp. 287–290.
27. Braslavskii, P.I. and Sokolov, E.A., Comparison of four methods for automatic recognition of two-word terms in text, in *Komp'yuternaya lingvistika i intellektual'nye tekhnologii* (Computational Linguistics and Intellectual Technologies), 2006, pp. 88–94.
28. Braslavskii, P. and Sokolov, E., Comparison of five methods for recognition of terms of arbitrary length, in *Komp'yuternaya lingvistika i intellektual'nye tekhnologii* (Computational Linguistics and Intellectual Technologies), 2008, no. 7, p. 14.
29. Bourigault, D., Surface grammatical analysis for the extraction of terminological noun phrases, *Proc. 14th Conf. Computational Linguistic*, 1992, vol. 3, pp. 977–981.
30. Baroni, M. and Bernardini, S., BootCaT: Bootstrapping corpora and terms from the Web, *Proc. Conf. Language Resources and Evaluation (LREC)*, 2004, pp. 1313–1316.
31. Dobrov, B.V., Lukashevich, N.V., and Syromyatnikov, S.V., Formation of the base of terminological phrases based on domain texts, *Trudy 5oi Vseross. nauchn. konf. "Elektronnye biblioteki: perspektivnye metody i tekhnologii, elektronnye kollektsii"* (Proc. 5th All-Russ. Sci. Conf. "Electronic Libraries: Promising Methods and Technologies, Electronic Collections"), 2003, pp. 201–210.
32. Automatic Text Processing, Syntactic analysis. http://www.aot.ru/docs/synan.html.
33. Fedorenko, D., Astrakhantsev, N., and Turdakov, D., Automatic recognition of domain-specific terms: An experimental evaluation, *Proc. Spring Researchers Colloquium on Databases and Information Systems (SYRCoDIS)*, 2013, pp. 15–23.
34. Nokel, M. and Loukachevitch, N., An experimental study of term extraction for real information-retrieval thesauri, *Proc. 10th Int. Conf. Terminology and Artificial Intelligence*, 2013, pp. 69–76.
35. Ventura, J.A.L., Jonquet, C., and Roche, M., et al., Combining C-value and keyword extraction methods for biomedical terms extraction, *Proc. Int. Symp. Languages in Biology and Medicine (LBM)*, 2013, pp. 45–49.
36. Barron-Cedeno, A., Sierra, G., Drouin, P., et al., An improved automatic term recognition method for Spanish, in *Computational Linguistics and Intelligent Text Processing*, Berlin: Springer, 2009, pp. 125–136.
37. Bordea, G., Domain adaptive extraction of topical hierarchies for expertise mining, *Ph.D. Dissertation*, Galway: National University of Ireland, 2013.
38. Navigli, R. and Velardi, P., Semantic interpretation of terminological strings, *Proc. 6th Int. Conf. Terminology and Knowledge Engineering*, 2002, pp. 95–100.
39. Dennis, S.F., The construction of a thesaurus automatically from a sample of text, *Proc. Symp. Statistical Association Methods for Mechanized Documentation*, Washington, 1965, pp. 61–148.
40. Church, K., Gale, W., Hanks, P., et al., Using statistics in lexical analysis, in *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*, 1991, p. 115.
41. Dunning, T., Accurate methods for the statistics of surprise and coincidence, *Comput. Linguist.*, 1993, vol. 19, no. 1, pp. 61–74.
42. Church, K.W. and Hanks, P., Word association norms, mutual information, and lexicography, *Comput. Linguist.*, 1990, vol. 16, no. 1, pp. 22–29.
43. Daille, B., Combined approach for terminology extraction: Lexical statistics and linguistic filtering, *Ph.D. Thesis*, Paris: University Paris 7, 1994.
44. Park, Y., Byrd, R., and Boguraev, B., Automatic glossary extraction: Beyond terminology identification, *Proc. 19th Int. Conf. Computational Linguistic*, 2002, vol. **1**, pp. 1–7.
45. Blei, D.M. and Lafferty, J.D., Topic models, *Text Min. Classif., Clustering, Appl.*, 2009, vol. 10, p. 71.
46. Bolshakova, E., Loukachevitch, N., and Nokel, M., Topic models can improve domain term extraction, in *Advances in Information Retrieval*, Berlin: Springer, 2013, pp. 684–687.
47. Li, S., Li, J., Song, T., et al., A novel topic model for automatic term extraction, *Proc. 36th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, 2013, pp. 885–888.

48. Meijer, K., Frasincar, F., and Hogenboom, F., A semantic approach for extracting domain taxonomies from text, *Decis. Support Syst.*, 2014, vol. 62, pp. 78–93.

49. Penas, A., Verdejo, F., Gonzalo, J., et al., Corpus-based terminology extraction applied to information access, *Proc. Corpus Linguistics*, 2001, vol. 13.

50. Manning, C. and Schutze, H., *Foundations of Statistical Natural Language Processing*, MIT Press, 1999.

51. Braslavskii, P.I. and Sokolov, E.A., Automatic term recognition using Internet retrieval engines, in *Komp'yuternaya lingvistika i intellektual'nye tekhnologii* (Computational Linguistics and Intellectual Technologies), 2007, pp. 89–94.

52. Golomazov, D.D., Methods and techniques for scientific information management using ontologies, *Cand. Sci. (Phys.–Math.) Dissertation*, Moscow, 2012.

53. Dobrov, B.V. and Loukachevitch, N.V., Multiple evidence for term extraction in broad domains, *Proc. Recent Advances in Natural Language Processing*, 2011, pp. 710–715.

54. Xu, F., Kurz, D., Piskorski, J., et al., A domain adaptive approach to automatic acquisition of domain relevant terms and their relations with bootstrapping, *Proc. Int. Conf. Language Resources and Evaluation*, 2002.

55. Milne, D., Medelyan, O., and Witten, I.H., Mining domain-specific thesauri from Wikipedia: A case study, *Proc. IEEE/WIC/ACM Int. Conf. Web Intelligence*, 2006, pp. 442–448.

56. Strube, M. and Ponzetto, S.P., WikiRelate!: Computing semantic relatedness using Wikipedia, *Proc. 21st AAAI Conf. Artificial Intelligence*, 2006, vol. 6, pp. 1419–1424.

57. Mihalcea, R. and Csomai, A., Wikify!: Linking documents to encyclopedic knowledge, *Proc. 16th ACM Conf. Information and Knowledge Management*, 2007, pp. 233–242.

58. Milne, D. and Witten, I.H., Learning to link with Wikipedia, *Proc. 17th ACM Conf. Information and Knowledge Management*, 2008, pp. 509–518.

59. Vivaldi, J. and Rodriguez, H., Using Wikipedia for term extraction in the biomedical domain: First experiences, *Procesamiento del Lenguaje Natural*, 2010, vol. 45, pp. 251–254.

60. Vivaldi, J., Cabrera-Diego, L.A., Sierra, G., et al., Using Wikipedia to validate the terminology found in a corpus of basic textbooks, *Proc. Conf. Language Resources and Evaluation (LREC)*, 2012, pp. 3820–3827.

61. Astrakhantsev, N., Automatic term recognition in a domain-specific text collection using Wikipedia, *Tr. Inst. Sistemnogo Program. Ross. Akad. Nauk*, 2014, vol. 26, no. 4, pp. 7–20.

62. Patry, A. and Langlais, P., Corpus-based terminology extraction, *Proc. 7th Int. Conf. Terminology and Knowledge Engineering*, Copenhagen, 2005.

63. Astrakhantsev, N., Fedorenko, D., and Turdakov, D., Automatic enrichment of informal ontology by analyzing a domain-specific text collection, *Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue,"* 2014, vol. **13**, pp. 29–42.

64. Yang, Y., Yu, H., Meng, Y., et al., Fault-tolerant learning for term extraction. http://www.aclweb.org/anthology/Y10-1036.

65. Liu, X. and Kit, C., An improved corpus comparison approach to domain specific term recognition, *Proc. Pacific Asia Conf. Language, Information, and Computing (PACLIC)*, 2008, pp. 253–261.

66. Kim, J.-D., Ohta, T., Tateisi, Y., *et al.*, GENIA corpus: A semantically annotated corpus for bio-textmining, *Bioinformatics*, 2003, vol. 19, no. Suppl. 1, pp. 180–182.

67. Nenadie, G., Ananiadou, S., and McNaught, J., Enhancing automatic term recognition through recognition of variation, *Proc. 20th Int. Conf. Computational Linguistics*, 2004, p. 604.

68. Krauthammer, M. and Nenadic, G., Term identification in the biomedical literature, *J. Biomed. Inf.*, 2004, vol. 37, no. 6, pp. 512–526.

69. Medelyan, O. and Witten, I.H., Domain-independent automatic keyphrase indexing with small training sets, *J. Am. Soc. Inf. Sci. Technol.*, 2008, vol. 59, no. 7, pp. 1026–1040.

70. Krapivin, M., Autaeu, A., and Marchese, M., Large dataset for keyphrases extraction. http://eprints.biblio.unitn.it/1671/1/disi09055-krapivin-autayeu-marchese.pdf.

Translated by Yu. Kornienko