КОМБИНИРОВАНИЕ ПРИЗНАКОВ ДЛЯ ИЗВЛЕЧЕНИЯ ОДНОСЛОВНЫХ ТЕРМИНОВ

Нокель М. А. (mnokel@gmail.com),
МГУ, Москва, Россия

Большакова Е. И. (eibolshakova@gmail.com),
МГУ, Москва, Россия

Лукашевич Н. В. (louk_nat@mail.ru),
НИВЦ МГУ, Москва, Россия

Ключевые слова: однословные термины, извлечение терминов, комбинирование признаков, машинное обучение

COMBINING MULTIPLE FEATURES FOR SINGLE-WORD TERM EXTRACTION

Nokel M. A. (mnokel@gmail.com)
Moscow State University, Moscow, Russia

Bolshakova E. I. (eibolshakova@gmail.com)
Moscow State University, Moscow, Russia

Loukachevitch N. V. (louk_nat@mail.ru)
Research Computing Center, Moscow State University,
Moscow, Russia

The paper describes experiments on automatic single-word term extraction based on combining various features of words, mainly linguistic and statistical, by machine learning methods. Since single-word terms are much more difficult to recognize than multi-word terms, a broad range of word features was taken into account, among them are widely-known measures (such as TF-IDF), some novel features, as well as proposed modifications of features usually applied for multi-word term extraction.
A large target collection of Russian texts in the domain of banking was taken for experiments. Average Precision was chosen to evaluate the results of term extraction, along with the manually created thesaurus of terminology on banking activity that was used to approve extracted terms.
The experiments showed that the use of multiple features significantly improves the results of automatic extraction of domain-specific terms. It was proved that logistic regression is the best machine learning method for single-word term extraction; the subset of word features significant for term extraction was also revealed.

Key words: single-word, term extraction, combining features, machine learning

1. Introduction

Term extraction is a field of language technology that involves extraction of relevant terms from domain-specific language corpora. To date, the research in this field has tended to focus on extraction of multi-word terms rather than on single-word ones mainly because most terms are multi-word. At the same time it is argued that single-word terms are much more difficult to recognize (Sclano & Velardi, 2007).

An important current trend in the research consists in applying machine learning methods that combine various features of words for term extraction.

In the work (Vivaldi et al., 2001) for extraction of medical terms features of words are combined with AdaBoost algorithm. (Azé et al., 2005) combines 13 various statistical criteria measures via the supervised learning genetic algorithm ROGER. In (Pecina and Schlesinger, 2006) the combination of multiple statistical characteristics of phrases is used to extract multi-word expressions from the Czech text collection. (Zhang et al., 2008) propose a combined method based on five term recognition algorithms that are capable of handling both single-word and multi-word terms. (Foo and Markel, 2010) apply the rule induction learning system Ripper to automatic term extraction from Swedish patent texts. (Dobrov and Loukachevitch, 2011) combine multiple features for two-word term extraction from texts of two different domains: the broad domain of natural science and technologies and the domain of banking.

Our study continues the described works aiming to apply machine learning in order to improve automatic term extraction, but in contrast to them we deal with single-word terms.

The overall process of extracting single-word terms consists of the following steps:

1) Extraction of term candidates from domain-oriented texts. In our study we consider only nouns and adjectives because they cover the majority of the terms; for our machine learning experiments they are extracted from a target collection of Russian banking texts taken from various electronic magazines such as Analytical Banking Magazine and Auditor.
2) Reordering the list of extracted candidates with the purpose to get more approved terms in the top of the list. To reorder the list, certain word features that measure "termhood" are used. In our study a variety of features (mainly, linguistic and statistical) is considered.

To evaluate the results of the reordering we need a way to approve terms from the candidate list. For these purposes we use the banking thesaurus manually created for the Central Bank of the Russian Federation. It includes about 15 thousand terms and comprises the terminology of banking activity. We consider a given candidate from the list as a term if it belongs to the thesaurus.

In the paper we first characterize the set of chosen features, and then describe experiments with machine learning.

2. Features for term candidate ranking

2.1. Linguistic features

We improve the results of the first step in the term extraction process via the following simple post morphology techniques:

1) We apply a simple morphological disambiguation procedure to extract only those initial forms of the nouns and adjectives that are consistent in text with other context words. Thus, the combinations such as *Preposition + Noun* and *Preposition + Adjective* should be consistent in the case, while the combinations such as *Adjective + Noun*, *Participle + Noun*, *Possessive Pronoun + Noun*, *Ordinal Number + Noun* should be in agreement with the gender, number and case.
2) Term candidates that had the same initial forms with the words with POS other than nouns and adjectives were excluded from the consideration.

For resulted set of term candidates we propose to apply four linguistic features that do not rank the term candidates and are used for the purposes of machine learning: **Ambiguity**, **Novelty**, **POS** and **Specificity**. The first one determines whether the word has multiple initial forms, the second determines whether the word is known for a morphological analyser, the third determines whether the word is a noun or an adjective, and the last determines whether the word exists in the reference text collection.

We also make an attempt to take into account the subjects in the sentences because they are more likely to represent some domain-specific information. All words in the nominative case (according to the morpohological analyzer) are considered as the subjects.

2.2. Orthographic features

Supporting the proposal of (Conrado et al., 2011) we consider the number of occurrences of words beginning with the capital letters. However, we also consider the subset of these words that did not start the sentences because such words are more likely to represent the named entities in the subject domain (they are called non-initial words further in the paper).

2.3. Statistical features

The most of the features of our set are statistical. They may be divided into four groups:

1) Features based only on the target corpus;
2) Features based on the target and reference corpora;
3) Features based on the statistical and contextual information;

4) Features for the words that stand near the most frequent ones in texts.

2.3.1. Used notations

While describing the statistical features, we use the following notations:

– w is the word from the target corpus;
– $TF_t(w)$ and $TF_r(w)$ are the frequencies of the word w in the target and reference corpora;
– $|W_t|$ and $|W_r|$ are the total numbers of words in the target and reference corpora;
– $DF_t(w)$ and $DF_r(w)$ are the numbers of documents containing the word w in the target and reference corpora;
– $|D_t|$ and $|D_r|$ are the total numbers of documents in the target and reference corpora;

2.3.2. Features based only on the target corpus

The most basic features in the group are **Term Frequency** (TF) and **Document Frequency** (DF). The former is the number of occurrences of term candidates in the target corpus, while the latter is the number of documents where a term candidate occurs. These features reflect the assumption that the terms are much more frequent than other words in the target corpus.

The more complex feature is **Term Frequency — Inverse Document Frequency** (TF-IDF) that was originated and is widely used in Information Retrieval. This measure encourages words that occur many times within a small number of documents. Primarily, this feature (Manning and Schutze, 1999) was calculated via the general collection. Later, it was adapted to use only the target corpus. Therefore, we consider these two versions of TF-IDF measure:

$$TF\text{-}IDF^{reference}(w) = TF_t(w) \times \log\frac{|D_r|}{DF_r(w)}; \quad TF\text{-}IDF(w) = TF_t(w) \times \log\frac{|D_t|}{DF_t(w)}$$

We also use an important extension of TF-IDF measure called **Term Frequency — Residual Inverse Document Frequency** (TF-RIDF) proposed by (Church and Gale, 1995):

$$TF\text{-}RIDF(w) = TF_t(w) \times \left(\log\frac{|D_t|}{DF_t(w)} - \left(-\log\left(1 - e^{-\frac{TF_t(w)}{|D_t|}} \right) \right) \right)$$

TF-RIDF is based on the observation that the Poisson model can only fairly predict the distribution of non-content words. Therefore, the deviation from Poisson can be used to predict term informativeness.

The last feature in the group is **Domain Consensus** (DC) (Navigli and Velardi, 2002). This measure simulates the consensus that a term must gain in a community before considered a relevant domain term. It is an entropy-related feature:

$$DC(w) = -\sum_{d_k \in D_t} \left(freq(w, d_k) \times \log\left(freq(w, d_k) \right) \right)$$

where d_k is a document from the target corpus D_t and $freq(w, d_k)$ is the frequency of the word w in a document d_k divided by the total number of the words in d_k.

All these features were calculated four times: for all term candidates, only for subjects, only for words beginning with capital words and only for non-initial words.

2.3.3. Features based on the target and reference corpora

The first feature in this group is **Weirdness** (Ahmad et al., 2007). It compares term frequencies in the target and reference corpora and reflects the basic idea for all measures in the group that these frequencies significantly differ:

$$Weirdness(w) = \frac{TF_t(w)}{|W_t|} \Big/ \frac{TF_r(w)}{|W_r|}$$

Relevance feature (Peñas et al., 2007) is based on the similar idea:

$$\mathrm{Re}levance(w) = 1 - \frac{1}{\log_2\left(2 + \frac{TF_t(w) \times DF_t(w)}{TF_r(w)}\right)}$$

This weight is high for high frequent terms in the target corpus, unless they are also frequent in the reference corpus or appear in a very small number of documents in the target corpus.

Next, we consider several extensions of TF-IDF measure. **Contrastive Weight** (CW) was proposed in (Basili et al., 2001) as a more accurate weight that reflects the specificity of terms with respect to the target domain. It is based on the heuristic that general words should spread similarly across different domain corpus:

$$CW(w) = \log\left(TF_t(w)\right) \times \left(\log \frac{|W_t| + |W_r|}{TF_t(w) + TF_r(w)} \right)$$

Next, Domain Tendency (DT) and Domain Prevalence (DP), which are slight modifications of Weirdness and CW respectively, contribute to the weight, known as **Discriminative Weight** (DW) (Wong et al., 2007). A term that appears frequently in the target corpus will have a low overall DW if it is more specific in the reference corpus:

$$DW(w) = DP(w) \times DT(w), \text{ where } DT(w) = \log_2\left(\frac{TF_t(w) + 1}{TF_r(w) + 1} + 1\right)$$

$$DP(w) = \log_{10}\left(TF_t(w) + 10\right) \times \log_{10}\left(\frac{|W_t| + |W_r|}{TF_t(w) + TF_r(w)} + 10\right)$$

One more extension of TF-IDF measure is **KF-IDF** (Kurz and Xu, 2002). This feature considers a simple term candidate as relevant if it appears more often than other candidates in the target domain, but occasionally in the reference domain. This weight is defined as follows:

$$KF\text{-}IDF(w) = DF_t(w) \times \log\left(\frac{2}{|D_w|} + 1\right)$$

where $|D_w|$=2, if the word w exists in the reference collection, and $|D_w|$=1 otherwise.

The last feature in the group is **Loglikelihood** that was originally designed for multi-word term extraction and then adapted by (Gelbukh et al., 2010) to single-word term extraction. Since only term candidates whose relative frequency is greater in the target corpus than in the reference one are taken into account: $\frac{TF_t(w)}{|W_t|} > \frac{TF_r(w)}{|W_r|}$, Log-likelihood is defined as follows:

$$Loglikelihood(w) = 2 \times \left(TF_t(w) \times \log\left(\frac{TF_t(w)}{TF_t^{\exp ected}(w)}\right) + TF_r(w) \times \log\left(\frac{TF_r(w)}{TF_r^{\exp ected}(w)}\right)\right)$$

where $TF_t^{\exp ected}(w) = |W_t| \times \frac{TF_t(w) + TF_r(w)}{|W_t| + |W_r|}$; $TF_r^{\exp ected}(w) = |W_r| \times \frac{TF_t(w) + TF_r(w)}{|W_t| + |W_r|}$

2.3.4. Features based on the statistical and contextual information

First of all, C-value and its modifications and improvements are included into the group. Originally, **C-value** was proposed to extract multi-word terms (Ananiadou, 1994), but we use its modified version adapted by (Nakagawa and Mori, 2002) for single-word term extraction:

$$MC\text{-}value(w) = TF_t(w) - \frac{\sum_{p \in P_w} TF_t(p)}{|P_w|}$$

where $|P_w|$ is the set of all phrases in the text collection that contain the word w and $|P_w|$ is its cardinality.

The most widely known modification of C-value is **NC-value** (Frantzi and Ananiadou, 1997). This weight incorporates contextual information into C-value for the extraction of multi-word terms and counts how independently the given multi-word term is used in the target corpus. We adapt NC-value to single-word term extraction as follows:

$$NC\text{-}value(w) = \frac{1}{|W_t|} \times MC\text{-}value(w) \times cweight(w)$$

$$\text{where } cweight(w) = \sum_{c \in C_w} weight(c) + 1; \quad weight(c) = \frac{1}{2}\left(\frac{|W_c|}{|W_t|} + \frac{\sum_{e \in W_c} freq(e)}{TF_t(c)}\right)$$

where C_w is the set of context words of the word w, W_c is the set of the term candidates that have c as a context word, $\sum_{e \in W_c} freq(e)$ is the sum of the frequencies of the term candidates that appear with the word c.

We also consider another form of the original NC-value proposed by (Frantzi and Ananiadou, 1999) and modify it to single-word term extraction:

$$MNC\text{-}value(w) + 0.8 \times MC\text{-}value(w) + 0.2 \times CF(w)$$

where $CF(w) = \sum_{c \in C_w} freq(c)$ is the context factor for the word w, C_w is the set of context words of the word w, $freq(c)$ is the frequency of the term candidate c as the context word of the word w.

Next feature in the group is LR (Nakagawa and Mori, 2003) that is based on the intuition that some words are used as term units more frequently than others. It was originally proposed for multi-word term extraction, but we adapt it for single-word term extraction by simply replacing the term units in its definition by context words. We consider two versions of this score: **Token-LR** and **Type-LR**:

$$Token\text{-}LR(w) = \sqrt{l_{token}(w) \times r_{token}(w)}; \quad Type\text{-}LR(w) = \sqrt{l_{type}(w) \times r_{type}(w)}$$

where the left score $l_{token}(w)$ of the word w is defined as the sum of the frequencies of the context words appearing to the left of the word w, the left score $l_{type}(w)$ is the cardinality of the set of such context words and the right scores $r_{token}(w)$ and $r_{type}(w)$ are defined in the same manner.

Since all variants of LR method reflect the numbers of occurrences of the context words, but do not reflect the terms themselves, we also choose FLR method intended to overcome this shortcoming (Nakagawa and Mori, 2003). Similar to LR we consider two variants of FLR score: namely, **Type-FLR** and **Token-FLR**:

$$Token\text{-}FLR(w) = TF_t(w) \times Token\text{-}LR(w); \; Type\text{-}FLR(w) = TF_t(w) \times Type\text{-}LR(w)$$

Additionally we consider several features reflecting the usage of the word in a set of phrases. The first one is **Insideness** (Dobrov and Loukachevitch, 2011). It checks whether the word is used in the same phrase and is intended to reveal truncated word sequences that are the parts of the real terms (note, that the similar phenomenon is modelled by previously described C-value feature). Insideness is defined as follows:

$$Insideness(w) = \frac{F_{\max}}{TF_t(w)}$$

where $F_{\max}$ is the frequency of the most frequent phrase containing the word *w*.

Another feature is SumN proposed by (Loukachevitch and Logachev, 2010), where N is the number of the most frequent phrases containing the considered term candidate. The feature checks productivity of the word for the formation of domain word combinations. We also modify it for single-word term extraction by excluding term frequency from the denominator:

$$SumN(w) = \frac{\sum_{p \in P_w^N} TF_t(p)}{N}$$

where P_w^N is the set of the *N* most frequent phrases containing *w*. We consider **Sum3**, **Sum10** and **Sum50** features.

2.3.5. Features for the words that stand near the most frequent ones in texts

At last, we hypothesize that the terms are more likely to co-occur with the most frequent ones and introduce the novel feature **NearTermsFreq** defined as the number of the word occurrences in the context window of the several predefined most frequent words. In fact, as such words we take the first ten elements at the top of the term candidate list ordered by TF-RIDF because our experiments showed that it is the best single feature — cf. Table 1). Additionally we apply the original TF-IDF measure, calculated via the general text collection, to NearTermsFreq, thus obtaining the following feature:

$$NearTermsFreq\text{-}IDF_{ref}(\mathrm{w}) = NearTermsFreq(w) \times \log\left(\frac{|D_r|}{DF_r(w)}\right)$$

3. Experiments

3.1. Experimental setup

For experiments we used a target text corpus in the banking domain with 10 422 documents (nearly 15,5 million words) and word frequencies from the reference, more general collection. All described features were calculated for five thousand of the most frequent single-word term candidates extracted from the target collection.

In order to obtain the best combination of the features, we used machine learning methods provided by the freely-available library Weka (*http://www.cs.waikato.ac.nz/ml/weka/*). We performed four-fold cross-validation, which means that every

time the training set was three-quarters of the whole list while the testing set was the remaining part.

Among various methods of evaluation we chose Average Precision (Manning and Schutze, 1999) because this measure allows us to evaluate the quality of the term extraction using a single numerical value. Average Precision of a set of all term candidates D with $D_q \subseteq D$ as a set of approved ones among them is defined as follows:

$$AvP(D) = \frac{1}{|D_q|} \sum_{1 \le k \le |D|} \left(r_k \times \left(\frac{1}{k} \sum_{1 \le i \le k} r_i \right) \right)$$

where r_i=1 if the i-th term $\in D_q$ and r_i=0 otherwise. The formula reflects the fact that the more terms are concentrated in the top of the list, the higher the measure is.

3.2. Experimental results

In order to find the best combination of the features we tested several machine learning methods. It proved that the maximal value of Average Precision is achieved by logistic regression method. So it was taken for further experiments.

Table 1 shows AvP values for single features and their combination obtained with logistic regression (Ambiguity, Novelty, POS and Specificity features are not presented in the table because they do not rank the term candidates and are used only in the combination with the other features).

Table 1. Average Precision for single features and logistic regression

Feature	Average Precision
TF	33,91%
DF	28,7%
TF-IDF	37,84%
TF-RIDF	**40,05%**
DC	32,42%
$TF\text{-}IDF_{reference}$	34,56%
$TF_{subjects}$	29,66%
$DF_{subjects}$	27,92%
$TF\text{-}IDF_{subjects}$	30,56%
$TF\text{-}RIDF_{subjects}$	32,61%
$DC_{subjects}$	28,92%
$TF\text{-}IDF_{reference\ subjects}$	29,61%
$TF_{capital\ words}$	35,49%
$DF_{capital\ words}$	33,42%
$TF\text{-}IDF_{capital\ words}$	35,98%
$TF\text{-}RIDF_{capital\ words}$	37,97%
$DC_{capital\ words}$	34,63%
$TF\text{-}IDF_{reference\ capital\ words}$	35,51%
$TF_{non\text{-}initial\ words}$	36,29%
$DF_{non\text{-}initial\ words}$	36,12%
$TF\text{-}IDF_{non\text{-}initial\ words}$	36,26%
$TF\text{-}RIDF_{non\text{-}initial\ words}$	35,85%
$DC_{non\text{-}initial\ words}$	35,77%
$TF\text{-}IDF_{reference\ non\text{-}initial\ words}$	32,83%

Feature	Average Precision
Weirdness	29,87%
Relevance	32,43%
CW	34,42%
DW	30,37%
KF-IDF	28,68%
Loglikelihood	34,48%
MC-value	33,86%
NC-value	35,1%
MNC-value	34,55%
Token-LR	35,93%
Type-LR	33,21%
Token-FLR	35,44%
Type-FLR	34,02%
Insideness	27,8%
Sum3	36,88%
Sum10	37,22%
Sum50	36,86%
NearTermsFreq	35,76%
$NearTermsFreq\text{-}IDF_{ref}$	36,06%
Logistic Regression	**53,95%**

As we see, the best single feature is TF-RIDF, while logistic regression by combining multiple features gives an increase of 35% compared with the best single feature.

In the Table 2 the first ten elements from the top of the extracted term candidates lists are presented. The columns correspond to various orderings of the lists: the ordering by Term Frequency, by TF-RIDF feature, and by logistic regression (the real terms among them are given in italics).

Table 2. First ten extracted term candidates

#	Term Frequency	TF-RIDF	Logistic Regression
1	*Банк (Bank)*	*Банк (Bank)*	*Банковский (Banking)*
2	*Банковский (Banking)*	*Кредитный (Credit)*	*Компания (Company)*
3	*Россия (Russia)*	*Банковский (Banking)*	*Рынок (Market)*
4	Год (Year)	Риск (Risk)	Риск (Risk)
5	*Система (System)*	*Кредит (Credit)*	*Пенсионный (Pensionary)*

#	Term Frequency	TF-RIDF	Logistic Regression
6	Организация (Organization)	*Рынок (Market)*	*Аудиторский (Auditing)*
7	*Рынок (Market)*	*Система (System)*	*Страна (Country)*
8	*Кредитный (Credit)*	*Налоговый (Taxing)*	*Налоговый (Taxing)*
9	Банка (Jar)	*Страна (Country)*	*Система (System)*
10	*Российский (Russian)*	Банка (Jar)	*Бухгалтерский (Bookkeeping)*

3.3. Feature selection algorithm

The resulting combination model is too complex in the number of applied features. Some of them may be redundant for machine learning method and have no use in the model, make its training harder. In order to exclude them we applied a stepwise greedy algorithm for selecting the most significant features.

The algorithm starts with the empty set of features, and then at each step it adds the feature that maximizes the overall Average Precision. As a result, the combination of only eight features (namely, TF-IDF, TF-RIDF, KF-IDF, $DF_{non\text{-}initial\ words}$, $TF_{subjects}$, $TF\text{-}IDF_{reference\ subjects}$, Weirdness and NC-value) was found with **53,51 %** of Average Precision. Therefore, the number of the combined features may be considerably reduced with decrease in precision less than 1 %.

4. Conclusions

In the paper we described multiple word features including linguistic, orthographic and statistical ones that were used in machine learning experiments for ordering the set of single-word term candidates extracted from the target text corpus. Several machine learning methods combining the features were tested, and logistic regression proved to be the best with significantly higher values of Average Precision than for any single feature. In addition, it was experimentally found that the number of the combined features can be reduced to eight features without sensible decrease of Average Precision.

References

1. *Ahmad, K., Gillam, L., Tostevin L.* University of Survey participation in Trec8: Weirdness indexing for logical document extrapolation and retrieval. Proceedings of the 8th Text Retrieval Conference. Gaithersburg, USA, 2007.
2. *Ananiadou S.* A Methodology for Automatic Term Recognition. Proceedings of the 15th International Conference on Computational Linguistics, COLING'94. Kyoto, Japan, 1994, pp. 1034–1038.

3. *Azé, J., Roche, M., Kodratoff, Y., Sebag, M.* Preference Learning in Terminology Extraction: A ROC-based Approach. Proceedings of ASMDA'05 (Applied Stochastic Models and Data Analysis). Brest, France, 2005, pp. 209–219.
4. *Basili, R., Moschitti, A., Pazienza, M., Zanzotto, F.* A Contrastive Approach to Term Extraction. Proceedings of the 4th Terminology and Artificial Intelligence Conference (TIA). Nancy, France, 2001.
5. *Church, K., Gale, W.* Inverse Document Frequency IDF: A Measure of Deviation from Poisson. Proceedings of the Third Workshop on Very Large Corpora. Cambridge, USA, 1995, pp. 121–130.
6. *Conrado, M., Koza W., Díaz-Labrador, J., Abaitua, J., Rezende, S., Pardo, T., Solana, Z.* Experiments on Term Extraction Using Noun Phrase Subclassifications. Proceedings of the 8th Recent Advances in Natural Language Processing Conference (RANLP 2011). Hissar, Bulgaria, 2011, pp. 746–751.
7. *Dobrov, B., Loukachevitch, N.* 2011. Multiple Evidence for Term Extraction in Broad Domains. Proceedings of the 8th Recent Advances in Natural Language Processing Conference (RANLP 2011). Hissar, Bulgaria, 2011, pp. 710–715.
8. *Foo, J., Merkel M.* Using Machine Learning to Perform Automatic Term Recognition. Proceedings of the LREC2010 Aquisition Workshop. Malta, 2010.
9. *Frantzi, K., Ananiadou, S.* Automatic Term Recognition Using Contextual Cues. Proceedings of the IJCAI Workshop on Multilinguality in Software Industry: the AI Contribution. Nagoya, Japan, 1997.
10. *Frantzi, K., Ananiadou, S.* (1999), The C-value/NC-value Domain-independent Method for Multi-word Term Extraction. Journal of Natural Language Processing, Vol. 6, no. 3, pp. 145–179.
11. *Gelbukh, A., Sidorov, G., Lavin-Villa, E., Chanona-Hernandez, L.* Automatic Term Extraction using Log-likelihood based Comparison with General Reference Corpus. Proceedings of the Natural Language Processing and Information Systems, and the 15th International Conference on Applications of Natural Language to Information Systems. Cardiff, UK, 2010, pp. 248–255.
12. *Kurz, D., Xu, F.* Text Mining for the Extraction of Domain Relevant Terms and Term Collocations. Proceedings of the International Workshop on Computational Approaches to Collocations. Vienna, 2002.
13. *Loukachevitch, N., Logachev, IU.* Using Machine Learning for Term Extraction [Ispol'zovanie metodov mashinnogo obucheniia dlia izvlecheniia slov-terminov]. Trudy Dvenadtsatoi Natsional'noi Konferentcii po Iskusstvennomy Intellekty c Mezhdunarodnym Uchastiem: KII 2010 (Proceedings of the 12th National Conference on Artificial Intelligence with International Participation). Tver', Russia, 2010.
14. *Manning, C. D., Schutze, H.* (1999), Foundations of Statistical Language Processing. The MIT Press.
15. *Nakagawa, H., Mori, T.* A Simple but Powerful Automatic Term Extraction Method. COMPUTERM 2002 — Proceedings of the 2nd International Workshop on Computational Terminology. Taipel, Taiwan, 2002, pp. 29–35.
16. *Nakagawa, H., Mori, T.* (2003), Automatic Term Recognition based on Statistics of Compound Nouns and their Components. Terminology, Vol. 9, no.2, pp. 201–219.

17. *Navigli, R., Velardi, P.* Semantic Interpretation of Terminological Strings. Proceedings of the 6th International Conference on Terminology and Knowledge Engineering (TKE 2002). Nancy, France, 2002, pp. 95–100.
18. *Pecina, P., Schlesinger, P.* Combining Association Measures for Collocation Extraction. Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions. Sydney, Australia, 2006, pp. 651–658.
19. *Sclano, F., Velardi P.* TermExtractor: a Web Application to Learn the Common Terminology of Interest Groups and Research Communities. Proceedings of the 9th Conference on Terminology and Artificial Language TIA 2007. Sophia Antipolis, 2007.
20. *Vivaldi, J., Màrquez, L., Rodríguez, H.* Improving Term Extraction by System Combination Using Boosting. Proceedings of the 12th European Conference on Machine Learning (ECML 2001). Freiburg, Germany, 2001, pp. 515–526.
21. *Wong, W., Liu, W., Bennamoun, M.* Determining Termhood for Learning Domain Ontologies using Domain Prevalence and Tendency. Proceedings of the 6th Aistralasian Conference on Data Mining (AusDM). Gold Coast, 2007, pp. 47–54.
22. *Zhang, Z., Iria, J., Brewster, C., Ciravegna, F.* A Comparative Evaluation of Term Recognition Algorithms. Proceedings of the Sixth International Language Resources and Evaluation (LREC'08). Marrakech, Morocoo, 2008.