Evaluating the Reliability and Interaction of Recursively Used Feature Classes for Terminology Extraction

Anna Hätty
Robert Bosch GmbH
Anna.Haetty@
de.bosch.com

Michael Dorna
Robert Bosch GmbH
Michael.Dorna@
de.bosch.com

Sabine Schulte im Walde
IMS, University of Stuttgart
schulte@
ims.uni-stuttgart.de

Abstract

Feature design and selection is a crucial aspect when treating terminology extraction as a machine learning classification problem. We designed feature classes which characterize different properties of terms, and propose a new feature class for components of term candidates. By using random forests, we infer optimal features which are later used to build decision tree classifiers. We evaluate our method using the ACL RD-TEC dataset. We demonstrate the importance of the novel feature class for downgrading termhood which exploits properties of term components. Furthermore, our classification suggests that the identification of reliable term candidates should be performed successively, rather than just once.

1 Introduction

Terms are linguistic units which characterize a specific topic domain. For example, in the area of Computational Linguistics *Parsing*, *Machine Translation* and *Natural Language Generation* are candidates for single and multi-word terms. Automatic Term Recognition (ATR) is the task of identifying such terms in domain-specific corpora. ATR is an Information Extraction subtask and is used i.a. for compiling dictionaries and for ontology population (Maynard et al., 2008). A typical ATR system comprises two steps: First, term candidates are selected from text, e.g. by extracting sequences which match certain part-of-speech (POS) patterns in text (c.f. Justeson and Katz, 1995). Secondly, term candidates are scored and ranked with regard to their unithood and termhood.

Unithood denotes to what degree a linguistic unit is a collocation. *Termhood* expresses to which extent an expression is a term, i.e to which extent it is related to domain-specific concepts (Kagueura and Umino, 1996). Among a large number of measures, association measures like *Pointwise Mutual Information* (PMI) (Church and Hanks, 1989) are used to determine unithood whereas term-document measures like *tf-idf* (Salton and McGill, 1986) are used to determine termhood. Such measures use distinctive characteristics of terms on how they and their components are distributed within a domain or across domains.

We address term extraction as a machine learning classification problem (c.f. da Silva Conrado et al., 2013). Most importantly, we focus on the interpretability of a trained classifier to understand the contributions of feature classes to the decision process. For this task, we use random forests to automatically detect the best features. These features are used to build simple decision tree classifiers.

For the classification, we use features based on numeric measures which are computed from occurrences of term candidates, its components and derived symbolic information like POS tags. We call these *distributional features*. The advantage of relying on such features is that they are simple to compute and easy to compare. By combining machine learning with those features we get a flexible system which only needs little further information to be applicable on different kinds of text. In this work, we investigate the contributions of the different features to term extraction and experimentally test with our system if these features are mutually supportive. We also point out the limit of a system solely relying on distributional features.

The paper is organized as follows. Section 2 introduces related work. The data used for training and evaluation is presented in Section 3, followed by the feature selection and classification method.

Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 113–121, Valencia, Spain, April 3-7 2017.

Our feature classes are motivated and defined in Section 4. In Section 5, we investigate the design of our models with a subsequent presentation of experiments and evaluation results in Section 6. In Section 7, we present a second experiment with term candidates which share a component to further explore their contribution to termhood.

2 Related Work

There are several studies investigating linguistic and numeric features, machine learning or a combination of both to extract collocations or terms. Pecina and Schlesinger (2006) combined 82 association measures to extract Czech bigrams and tested various classifiers. The combination of measures was highly superior to using the best single measure. Ramisch et al. (2010) introduced the *mwetoolkit* which identifies multi-word expressions from different domains. The tool provides a candidate extraction step in advance, descriptive features (e.g. capitalisation, prefixes) and association measures can be used to train a classifier. The latter ones are extended for multi-word expressions of indefinite length and only comprise measures which do not depend on a contingency table. Karan et al. (2012) extract bigram and trigram collocations for Croatian by relying on association measures, frequency counts, POS-tags and semantic similarities of all word pairs in an *n*-gram. They found that POS-tags, the semantic features and PMI work best. With regard to terms, Zhang et al. (2008) compare different measures (e.g. tf-idf) for both single- and multi-word term extraction and use a voting algorithm to predict the rank of a term. They emphasize the importance of considering unigram terms and the choice of the corpus. Foo and Merkel (2010) use RIPPER (Cohen, 1995), a rule induction learning system to extract unigram and bigram terms, by using both linguistic and numeric features. They show that the design of the ratio of positive and negative examples while training governs the output rules. Da Silva Conrado et al. (2013) investigate features for the classification of Brazilian Portuguese unigram terms. They use linguistic, statistical and hybrid features, where the context and the potential of a candidate representing a term is investigated. Regarding the features, they find tf-idf essential for all machine learning methods tested.

3 Data and Classification Method

3.1 Corpus and Gold Standard

The underlying data set for the experiments is the ACL RD-TEC 1.0[1], a corpus designed for the evaluation of terminology extraction in the area of Computational Linguistics (Zadeh and Handschuh, 2014). It extends ACL ARC, an automatically segmented and POS-tagged corpus of 10,922 ACL publications from 1965 to 2006. ACL RD-TEC adds a manual annotation of 22,044 valid terms and 61,758 non-terms. The term annotations are further refined with a labeling of terminology terms which are defined as means to accomplish a practical task, like methods, systems and algorithms used in Computational Linguistics. We take the valid terms as our gold standard terms. We cleaned the corpus by applying a language detection tool (*langdetect*[2]) to each sentence, in order to remove sentences which are too noisy. A drawback of the corpus is that about 42,000 sentences could not be connected to a document. Thus, if no document was found for a certain term, its term-document measures were set to a default value outside of a feature's range, or to an extreme value.

3.2 Feature Reduction and Classification

Unigrams, bigrams and trigrams which appear at least ten times in the text are extracted from the corpus as term candidates. For all candidates, features are computed (see Section 4). As a preprocessing step, a **random forest classifier** (Breiman, 2001) with 100 estimators is used for feature reduction. To prevent overfitting, each of these decision trees is trained on a subset of the data, and a randomly chosen subset of features (here the square root of the number of features) is considered for splitting a node. Considering all internal decision trees, the contribution of the features to the classification is evaluated and averaged. In this way, we get good estimates of the importances of each feature and can use them for feature reduction: the classifier returns the importance scores for the features, and feature selection is performed by only taking those features whose score is greater than the mean. Subsequently, a **decision tree classifier** (Breiman et al., 1984) is trained with those features that provide a single representation for the decisions. The training set

[1] http://atmykitchen.info/nlp-resource-tools/the-acl-rd-tec
[2] https://pypi.python.org/pypi/langdetect?

was balanced for terms and non-terms to prevent a bias in the classifier. In the first step, everything which is not marked as term is treated as non-term. We only allowed POS patterns also occurring in the term class and chose randomly to get a representative sample of non-terms. In the second step, we use the explicitly annotated non-term class.
Both classifiers produce binary decision trees and an optimized version of the CART algorithm[3] is used.
As split-criterium for the decision trees we used *entropy* and we only allowed trees to evolve up to five levels, since otherwise they overfit. In addition, trees are very difficult to understand when getting deeper than five levels and we explicitly chose decision trees because of their clear interpretability. For the interpretation and evaluation in the following, the construction of the final decision trees for each *n*-gram and their classification performances will be used.

4 Feature Classes

A salient attribute of terms is how they distribute in text. Our feature classes are motivated by three perspectives on that: a) measuring unithood involving the distribution of term candidates and their components, b) measuring termhood involving candidate term distributions in different texts and c) recursively measuring unithood and termhood of term candidate components independently of each other. Concerning the classes defined in the following, point a) is covered by the *association measures*, b) by *term-document* and *domain specificity measures* and c) by the *features of components*. In addition, we designed *count-based measures* and a *linguistic feature* to address unithood and termhood. However, we expect them to be weaker than the feature classes of a) and b) since they do not relate two distributions. They merely serve for filtering, ruling out very unlikely term candidates.

Term-Document Measures (TD) The term-document measures deal with the distribution of term candidates in certain documents and contrast it to their distribution in the whole corpus. It is assumed that terms appear more frequently in only a few documents. We include a range of features dealing with that contrast: variants of tf-idf (Salton and McGill, 1986), i.e. *tf-idf* (without logarithm), *tf-logged-idf* for the document in which the term candidate occurs most often. Furthermore, *corpus maximum frequency* and *corpus maximum frequency & term average frequency (cmf-taf)* as defined in Tilley (2008), and *term variance* and *term variance quality* as described in Liu et al. (2005) are used. Da Silva Conrado et al. (2013) describe the latter features as useful for term extraction. In addition, we experimented with features describing the relative occurrence of a term in a document or the corpus. For example, the percentiles of document or corpus frequencies are used as features, to which the frequency of the term under consideration can be assigned. Another example is the percentile of the document with the term candidate's first position. In the later experiments, these features are assigned little weight by the classifiers which is why we will not go into further detail regarding them.

Domain Specificity Measures (DS) Measures of domain specificity treat the occurrence of a term in a general corpus and relate it to its occurrence in a domain-specific one. As domain-specific corpus, we simply chose the document with the most frequent occurrence of a term candidate. By doing that, the problem is omitted that the vocabulary of these corpora differs too drastically due to aspects of style. As features *weirdness ratio for domain specificity, corpora-comparing log-likelihood (corpComLL), term frequency inverse term frequency (TFITF)* and *contrastive selection of multi-word terms (CSmw)* are used (as defined in Schäfer et al., 2015).

Association Measures (AM) Association measures express how strongly words are associated in a complex expression, they measure unithood. 27 association measures defined in Evert (2005) were computed for bigrams, for example *Local Mutual Information (LocalMI)* and *Maximum Likelihood Estimation (MLE)*. For trigrams, we selected nine association measures (*MLE, PMI, Dice, T-score, Poisson-Stirling, Jaccard,* χ^2*, Simple Log Likelihood* and *true MI*) which are described as useful for trigram association in Lyse and Andersen (2012), Ramisch et al. (2010) and the *nltk*-documentation [4].

Count-based Measures (Count) Wermter and Hahn (2006) compare co-occurrence frequencies

[3] http://scikit-learn.org/stable/modules/tree.html#tree

[4] www.nltk.org/_modules/nltk/metrics/association.html

and association measures and show that not association measures but only linguistically motivated features outperform frequency counts for collocation and terminology extraction. Therefore *frequencies* of the term candidates are included in the feature set. As described, we do not consider them as being as powerful as association measures (and they only play a minor role in our later models). The second count-based measure is *word length*.

Linguistic Feature (Ling) As linguistic feature, *Part-Of-Speech*-tags (POS) of the candidates are used to represent distributions over POS patterns.

Features of Components (Comp) The components of a term phrase have frequently played a role in termhood extraction (e.g. Nakagawa and Mori, 2003; Zhang et al., 2012). Our approach differs from the previous ones by adding all feature information of the candidate term components to the candidates's feature set. I.e., for bigrams the features of its unigrams, and for trigrams the features of its uni- and bigrams are included. The features will be characterized with the following scheme: [POSITION IN TERM]-[COMPONENT IS A UNI- OR BIGRAM]-[FEATURE]. Examples would be *0-uni-CSmw* denoting the CSmw-feature for the first word X in bigram XY or *1-bi-CSmw* denoting the CSmw-feature for second bigram YZ in trigram XYZ. *1-bi-POS != NN NN* expresses that the second bigram YZ in trigram XYZ does not consist of nouns.

Class	1	2,3	Feature Examples
TD	+	+	tf-idf, cmf-taf, term variance
DS	+	+	weirdness ratio, corpComLL, TFITF
AM	-	+	PMI, LocalMI, Chi2
Count	+	+	frequency, word length
Ling	+	+	POS pattern
Comp	-	+	0-uni-POS, 1-bi-tf-idf

Table 1: Overview of Feature Classes

An overview of the classes is given in Table 1. The labels 1, 2 and 3 in the table denote uni- to trigrams, + and - express if a class can be applied or not. For unigram terms (SWT) not all feature classes can be applied.

5 Inspecting the Models

Combining all previously mentioned features with our classification method (i.e. unigrams, bigrams and trigrams) provides three decision trees. For ease of visualization and interpretation, only the first three decision levels are shown in the following figures (Figures 1 to 3). The tree is only allowed to evolve further if the distinction between terms and non-terms could not be made to that point. Furthermore, splitting a node is stopped if there are less than 10 elements in a leaf for one of the classes (even if the tree limit has not been reached yet).

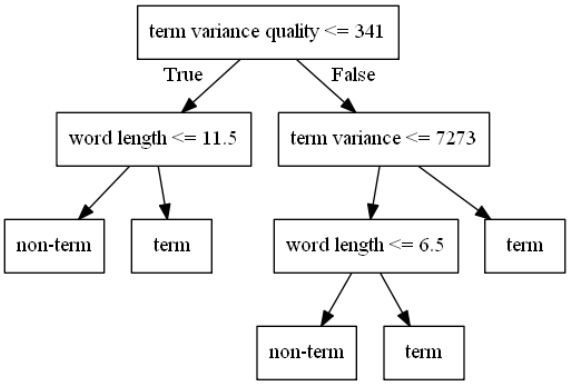

Figure 1: Decision Tree for Unigrams

Unigrams The decision tree for unigram classification based on 1608 unigram terms and non-terms is shown in Figure 1. Term variance quality and term variance best classify terms; In the resulting leaf node (rightmost node) 90% of the 324 elements are correct terms. When looking at the false positives in that node, it is striking that the few non-terms remaining in that class are unexpectedly "usual" (*'czech', 'newspaper', 'chain', 'travel', 'situation'*). The reason for this unexpected classification might result from the context in which the study is conducted: there might be papers which are limited to Czech data or only to newspaper texts.

The construction of the whole decision tree reveals that the classifier tries to identify clear-cut sets of terms using decision thresholds with extreme values. Following the path on the right-hand side, the subset of elements with the highest termhood scores is isolated. If the term-document measure values are not distinctive anymore (taking left branches) non-terms are singled out by filtering via word length. The less distinctive termhood measures are, the less word length is limited on filtering extremely short and therefore extremely unlikely term-candidates. This is an on-demand filtering step: term candidates are not only filtered in advance, but the threshold is adjusted to how

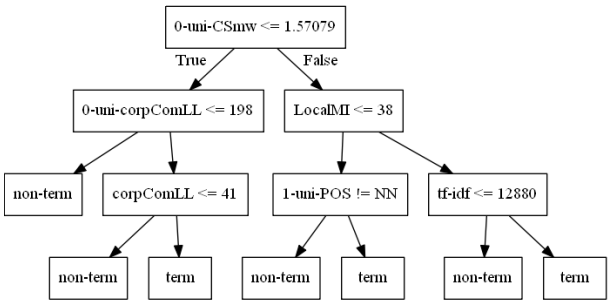

Figure 2: Decision Tree for Bigrams

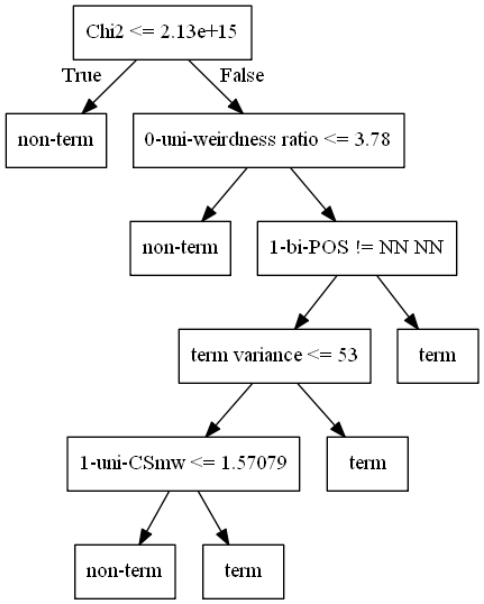

Figure 3: Decision Tree for Trigrams

significant the termhood measures are.

Bigrams The decision tree for the 10,562 extracted bigram candidates is depicted in Figure 2. Features for the first component like 0-uni-CSmw are good indicators for termhood. When inspecting how the bigrams are distinguished by the root node it seems that if the first word of a bigram is a general-language word, the whole bigram is unlikely to be a term. There are quite obvious examples like *this specification, the parser, a hurry* or *another expression* but also more interesting ones like *earlier paper, particular cluster* or *general scheme*. Nevertheless, in other term leaves there are still quite a few expressions whose first words are not terminological (e.g. *simple formalism, common description, good hypothesis*), so there is still room for improvement.

Trigrams The decision tree for trigram classification of 1706 trigram candidates is shown in Figure 3. The association measure χ^2 (*Pearson's chi-squared test*; c.f. Evert, 2005) is by far the most important feature here and the sets are nearly completely distinguished by that feature. Thus, unithood nearly merges to termhood here. Besides that, it is again striking that expressions with non-terminological first components are ruled out correctly by the system, e.g., *possible syntactic category*, *other natural language*, *new grammar formalism*. There are also misclassifications (false negatives) like *first order logic*. The rightmost path produces the purest right-most node compared to all previous ones for uni- and bigrams: 94% of the 636 elements are correct terms.

Comparison Across the decision trees different features dominate the tree, which shows that uni-, bi- and trigram terms behave differently and should be treated differently. Nevertheless, they have in common that the trees are dominated by termhood and unithood features and that features for filtering noise like POS patterns and word length occur lower in the tree. This supports the already mentioned claim that several filtering steps should be performed at different stages of the classification. As a second commonality, the trees combine features from various classes in their first decision steps. Especially in the rightmost path, in which terms are separated best in the experiments, term-document measures, association measures and domain-specificity measures of components are combined. This shows that features from different feature classes interact for achieving a good result.

6 Experiments and Results

Our system is implemented in Python. For the classifications, we used the *RandomForestClassifier* and the *DecisionTreeClassifier* which are included in the Python module *sklearn* (Pedregosa et al., 2011).

Baselines For each *n*-gram class, the best-working feature is chosen as a baseline. These are the root nodes of the decision trees for all features because these ones are chosen first, given that they make the best decision. The baselines are *term variance quality* for unigrams, *0-uni-CSmw* for bigrams and *Chi2* for trigrams.

Performance of Individual Feature Classes As a first evaluation step, the different feature classes are compared. For that, decision trees

are separately trained for each feature class. We do 10-fold cross-validation with a balanced set of terms and non-terms in every step. The performances of the different classes for unigrams, bigrams and trigrams are shown in Table 2. When considering the overall results (F1-score), it is striking that for bigrams and trigrams the component features (Comp) achieve the best score, middle-ranking groups are the count-based features (Count) and the linguistic feature (Ling), and the term-document (TD) and domain-specific features (DS) are in the lower area. This is quite a surprising result since these are the termhood features and therefore the ones to be expected to perform best. For unigrams, in contrast, term-document features and domain specificity are good indicators for classification. However, when considering precision, the domain specificity features lag behind. They do not seem to be competitive to term-document metrics in that respect. All in all, domain specificity features do not reach the expected performance here. This is an interesting result because when the domain specificity features are used for the components of an *n*-gram they appear in the upper part of the tree. We conclude that the features for domain specificity applied to components receive the unexpected application of downgrading the termhood of a term candidate if a component under consideration is unlikely to be terminological.

Feat. Class	TD	DS	Assoc	Count	Ling	Comp
Unigrams						
Precision	**0.75**	0.67	-	0.73	0.63	-
Recall	0.71	0.73	-	0.66	0.81	-
F1-Score	**0.72**	0.70	-	0.69	0.70	-
Bigrams						
Precision	0.72	0.65	0.72	**0.73**	0.67	**0.73**
Recall	0.71	0.79	0.65	0.79	0.88	0.88
F1-Score	0.71	0.71	0.68	0.76	0.76	**0.80**
Trigrams						
Precision	0.67	0.59	0.85	0.75	0.80	**0.88**
Recall	0.72	0.72	0.96	0.82	1.0	0.97
F1-Score	0.69	0.65	0.90	0.78	0.89	**0.92**

Table 2: Precision, Recall and F1-Scores for Feature Classes

Evaluating All Features As a last step, we evaluate if the combination of different features outperforms the best single feature. For that we do 10-fold cross-validation with a balanced set of terms and non-terms in every step. The results are shown in Table 3. All systems which combine features outperform the baselines. In addition, they also outperform the best systems which only use one feature class at a time (Table 2). All these improvements are significant, [5] except for the comparison of the overall model for trigrams to the model of its best-working class (*features of components*). This shows that a combination is not only superior to a baseline but also information from several classes is needed. Term recognition works best for trigrams and is most difficult for unigrams.

Method	Precision	Recall	F-score
Baseline	0.62	0.85	0.70
Unigrams	0.75	0.79	**0.77**
Baseline	0.60	0.89	0.72
Bigrams	0.78	0.87	**0.81**
Baseline	0.84	0.97	0.90
Trigrams	0.89	0.96	**0.93**

Table 3: Results

7 The Relevance of the Component Class

In the previous experiments we investigated how terms can be distinguished from candidates in the scientific text which are restricted by POS but which are otherwise randomly chosen. For bigrams and trigrams, the component class performs best. Since the components of candidate terms seem to have a major influence on their termhood, we further investigate the components. For that, candidates are not chosen randomly anymore, but are taken from the class explicitly annotated as non-terms by Zadeh and Handschuh (2014). The reason for this is that the elements of the provided annotated term and non-term expressions have identical components in many cases. Like that term candidates with components which are not uniquely terminological or non-terminological are used for training the classifier. Subsets of the classes are compared three times: Only those elements are allowed where either the first, the second or the third component (in case of trigrams) appears in both classes. The results are presented in Table 4.

The results indicate that a clearly terminological or non-terminological first component has more effect on the termhood of the whole expression than for the last component. If the first component is fixed and thus is not relevant for scoring termhood, results decrease.

[5] χ^2, $p<0.01$

	Bigrams			Trigrams		
Feature Class	**P**	**R**	**F1**	**P**	**R**	**F1**
last component	0.69	0.83	0.76	0.76	0.77	0.76
mid component	-	-	-	0.73	0.75	0.74
first component	0.66	0.70	0.68	0.73	0.71	0.72

Table 4: Results for identical elements for different components in term- and non-term class

This is also reflected in the decision trees: For identical heads, the most important feature is the component feature of the first unigram and of the first bigram. For identical modifiers no component feature is chosen as most important feature.

8 Discussion and Future Work

There are two main points why a system like ours only based on distributions reaches its limit. One aspect is the unexpected fluctuations of general-language terms shown especially for unigram term extraction. We found words being classified as terms because they often appear in the context of a special experimental setting. Secondly, our results show that it is harder for such a system to distinguish term candidates with shared components than to distinguish terms from a representative part of the other in-domain text as done in the first experiment (Table 3 vs. Table 4).
However, the advantages of our model suggest that it can be applied to extract terms from forum text, a topic which has not received much attention yet. The information used in the model, the features and their application on components of the term candidates, can be easily computed on the text and additional resources are not necessarily needed. Another advantage of our model is that it is dynamic. Uni-, bi- and trigrams are quite different in nature which is reflected in the models. It filters improbable term candidates by making several decision steps adapted to the data seen in training. Thus, we might not need a preprocessing step to filter good candidates. In both experiments, with and without an explicitly annotated non-term class, applying the features to components of the candidates improves the extraction. We find that especially the features for the first parts, mostly the modifier, are good dividers for the term and the non-term class. Since the number of non-terminologic modifiers (like judging adjectives) will be higher in forum texts, this aspect will be a further advantage.

9 Conclusion

In this work, term extraction was approached as a classification problem using uni-, bi- and trigram term candidates. We used a decision tree classifier to model term recognition with focus on the distribution of terms and of its components in text. Different classifier setups were compared: classifiers for the single best feature, different feature classes and a combination of all features. In each of those steps classification improves. Neither a feature class nor a special feature constantly dominates the classification in all models. The construction of the decision trees reveals that there is an interaction of features of different classes. Features from the most adequate classes to recognize terms, i.e. features which measure termhood and unithood, interact to find the purest term class.
The resulting decision trees from the experiments indicate that there should not be a rigid pipeline of two steps, where candidate extraction and filtering noise comes first, and subsequently the terms should be scored and ranked. Our results indicate that there should rather be an on-demand filtering step, where filtering is performed successively during the classification and the threshold for ruling out extremely unlikely candidates is adjusted to the decisions made before.
The most interesting finding is that measures of domain specificity perform unexpectedly low for bigram and trigram recognition but when being applied to their unigram components they appear in the upper parts of the tree. When looking into the data, the reason for this seems to be that there is a downgrading of multi-word term candidate phrases (bigrams and trigrams) if a component (preferably the first) is too common to belong to a term. A second experiment, in which we compare term candidates with shared components confirms this finding. The components of terms are addressed in several studies (Erbs et al, 2015; Frantzi el al., 2000; Nakagawa and Mori,2003; Zhang et al., 2012), but to our knowledge this aspect of termhood has not been considered yet.
Since our model is flexible and the feature selection easily adapts to different types of text data, we plan to apply it to forum texts and see how the results differ from the ones in this study. In addition, we aim to explore whether the results are reproducible for terms from other technical domains.

References

Leo Breiman, Jerome Friedman, Richard Olshen, and Charles Stone. 1984. Classification and Regression Trees. Wadsworth Publishing Company, Belmont, CA.

Leo Breiman. 2001. Random Forests. *Machine Learning*, 45(1):5-32.

Kenneth W. Church and Patrick Hanks. 1990. Word Association Norms, Mutual Information, and Lexicography. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, pages 76-83, Vancouver, British Columbia, Canada.

Jonathan Cohen. 1995. Highlights: Language- and Domain-Independent Automatic Indexing Terms for Abstracting. *Journal of the Association for Information Science and Technology*, 46(3):162-174.

Merley da Silva Conrado, Thiago S. Pardo, and Solange O. Rezende. 2013. A Machine Learning Approach to Automatic Term Extraction using a Rich Feature Set. In *Proceedings of the 2013 NAACL HLT Student Research Workshop*, pages 16-23, Atlanta, Georgia.

Nicolai Erbs, Pedro B. Santos, Torsten Zesch and Iryna Gurevych. 2015. Counting What Counts: Decompounding for Keyphrase Extraction. in *Proceedings of the ACL 2015 Workshop on Novel Computational Approaches to Keyphrase Extraction*, pages 10–17, Beijing, China.

Stefan Evert. 2005. The Statistics of Word Cooccurrences: Word Pairs and Collocations. Dissertation, Institut für maschinelle Sprachverarbeitung, University of Stuttgart.

Jody Foo and Magnus Merkel. 2010. Using Machine Learning to Perform Automatic Term Recognition. In *Proceedings of the 7th LREC - Workshop on Methods for Automatic Acquisition of Language Resources and their Evaluation Methods*, pages 49–54, Malta.

Katerina Frantzi, Sophia Ananiadou and Hideki Mima. 2000. Automatic Recognition of Multiword-Terms: the C-value/NC-value Method. *International Journal on Digital Libraries*, 3(2): 115-130.

John Justeson and Slava Katz. 1995. Technical Terminology: some Linguistic Properties and an Algorithm for Identification in Text. *Natural Language Engineering*, 1(1):9-27.

Kyo Kagueura and Bin Umino. 1996. Methods of Automatic Term Recognition: A Review. *Terminology*, 3(2): 259-289.

Mladen Karan, Jan Šnajder and Bojana D. Bašić. 2012. Evaluation of Classification Algorithms and Features for Collocation Extraction in Croatian. In *Proceedings of Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 657–662, Istanbul, Turkey.

Luying Liu, Jianchu Kang, Jing Yu, and Zhongliang Wang. 2005. A Comparative Study on Unsupervised Feature Selection Methods for Text Clustering. In *Proceedings of International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE'05)*, pages 597–601, Wuhan, China.

Gunn I. Lyse and Gisle Andersen. 2012. Collocations and Statistical Analysis of n-grams: Multiword Expressions in Newspaper Text. *Exploring Newspaper Language: Using the Web to Create and Investigate a Large Corpus of Modern Norwegian, Studies in Corpus Linguistics*, pages 79–109, John Benjamins Publishing, Amsterdam, Netherlands.

Diana Maynard, Yaoyong Li and Wim Peters. 2003. NLP Techniques for Term Extraction and Ontology Population. In *Proceedings of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 107–127, IOS Press, Amsterdam, Netherlands.

Hiroshi Nakagawa and Tatsunori Mori. 2003. Automatic Term Recognition based on Statistics of Compound Nouns and their Components. *Terminology*, 9(2):201–219.

Pavel Pecina and Pavel Schlesinger. 2006. Combining Association Measures for Collocation Extraction. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 651–658, Sydney, Australia.

Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Carlos Ramisch, Aline Villavicencio, and Christian Boitet. 2010. mwetoolkit: a Framework for Multiword Expression Identification. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta.

Gerard Salton and Michael McGill. 1986. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA.

Johannes Schäfer, Ina Rösiger, Ulrich Heid and Michael Dorna. 2015. Evaluating Noise Reduction Strategies for Terminology Extraction (TIA'15). In *Proceedings of Terminology and Artificial Intelligence*, pages 123–131, Granada, Spain.

Jason Tilley. 2008. *A Comparison of Statistical Filtering Methods for Automatic Term Extraction for Domain Analysis*. Master thesis, University of Virginia.

Joachim Wermter and Udo Hahn. 2006. You Can't Beat Frequency (Unless You Use Linguistic Knowledge) - A Qualitative Evaluation of Association Measures for Collocation and Term Extraction. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 785-792, Sydney, Australia.

Behrang Q. Zadeh and Siegfried Handschuh. 2014. The ACL RD-TEC: A Dataset for Benchmarking Terminology Extraction and Classification in Computational Linguistics. In *Proceedings of the 4th International Workshop on Computational Terminology (Computerm)*, pages 52-63, Dublin, Ireland.

Ziqi Zhang, José Iria, Christopher Brewster and Fabio Ciravegna 2008. A Comparative Evaluation of Term Recognition Algorithms. In *Proceedings of the Sixth Conference on International Language Resources and Evaluation (LREC'08)*, pages 2108-2113, Marrakech, Morocco.

Chunxia Zhang, Zhendong Niu, Peng Jiang and Hongping Fu. 2012. Domain-Specific Term Extraction from Free Texts. *9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'12)*, pages 1290–1293, Chongqing, China.