Expert Systems With Applications 199 (2022) 117179

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

NMF-based approach to automatic term extraction

Aliya Nugumanova [a,1], Darkhan Akhmed-Zaki [b,2], Madina Mansurova [c,3], Yerzhan Baiburin [a,4], Almasbek Maulit [a,5,*]

[a] *Sarsen Amanzholov East Kazakhstan University, Ust-Kamenogorsk, Kazakhstan*
[b] *Astana IT University, Nur-Sultan, Kazakhstan*
[c] *Al-Farabi Kazakh National University, Almaty, Kazakhstan*

ARTICLE INFO

Keywords:
Automatic term extraction
Probabilistic topic modeling
NMF
Unsupervised term extraction
ACTER dataset
TermEval shared task

ABSTRACT

This work describes automatic term extraction approach based on the combination of the probabilistic topic modelling (PTM) and non-negative matrix factorization (NMF). Topic modeling algorithms including NMF-based ones do not require expensive and time-consuming manual annotations for domain terms, but only a corpus of domain documents. The topics emerge from the corpus documents without any supervision as sets of most probable words. This work is aimed to investigate how fully and precisely these most probable words from topics can reflect domain terminology. We run a series of experiments on the novel, qualitatively annotated dataset ACTER that was first used in the TermEval 2020 Shared Task. We compare five different NMF algorithms and four different NMF initializations when changing the number of topics extracted from documents and the number of most probable words extracted from topics in order to determine optimal combinations for best performance of term extraction. Finally, we compare the obtained optimal combinations of NMF with the competitive methods in TermEval 2020 and prove that our approach is second only to two much more sophisticated, domain-dependent supervised methods.

1. Introduction

Automatic term extraction, also known as automatic term recognition, is a task aimed at detecting domain terms in a given corpus of documents. Traditionally, methods for solving this problem include three stages: 1) preprocessing and term candidates extracting, 2) term candidates scoring, and 3) term candidates ranking (Astrakhantsev et al., 2015). Preprocessing and term candidates extracting is the most standardized stage that is aimed to transform input texts into a set of words (i.e., tokens, lemmas, n-grams) to obtain term candidates. At this stage, all words and phrases are extracted from texts that can be considered as term candidates with a formal (linguistic) point of view. As a rule, nouns and noun groups are chosen as candidates. Term candidates scoring is the stage that is aimed to assign a value (i.e., score) to each candidate. At this stage, termhood scores quantifying the relevance of candidates are formed, based on some algorithms or criteria. Term candidates ranking is the final stage that is aimed to order candidates according to their scores and thus cut off candidates with lowest scores. At this stage, the final set of domain terms is formed from the top-N candidates with the highest scores.

There are at least two challenges faced by the automatic term extraction task. The first is the low recognizing ability of existing algorithms and criteria, because of which many false terms are extracted from the texts (noise), and several true terms are skipped (silence). The second challenge is the lack of large-scale, well-annotated domain-specific text corpora that allow comparison of different approaches in different domains and in different languages. Both challenges are largely due to the same reason – the absence of a clear and consistent definition of what a domain term is and what its distinctive features are in a text.

In response to these challenges, in 2020, the LREC conference launched the shared TermEval platform, designed to unite researchers in the field of automatic term extraction in order to compare their

* Corresponding author.
E-mail addresses: yalisha@yandex.kz (A. Nugumanova), darkhan.akhmed-zaki@astanait.edu.kz (D. Akhmed-Zaki), maulit.almas@yandex.ru (A. Maulit).

[1] 0000-0001-5522-4421.
[2] 0000-0001-8100-8263.
[3] 0000-0002-9680-2758.
[4] 0000-0002-1583-9912.
[5] 0000-0002-0519-3222.

https://doi.org/10.1016/j.eswa.2022.117179
Received 3 June 2021; Received in revised form 13 January 2022; Accepted 1 April 2022
Available online 6 April 2022

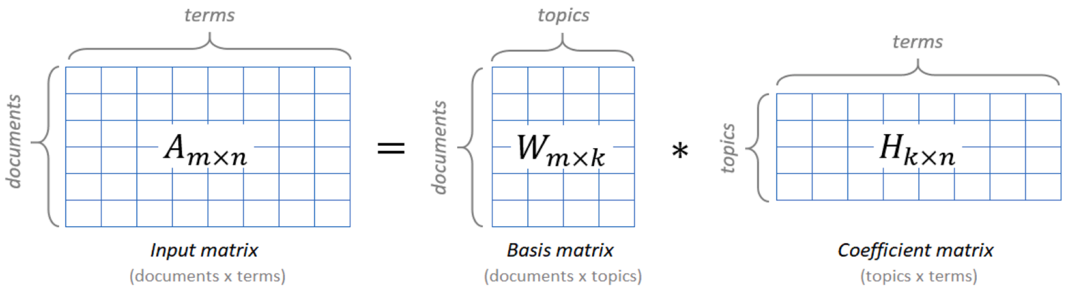

Fig. 1. Non-negative matrix factorization of the documents-terms matrix.

achievements on a single, large and diverse ACTER dataset (Terryn et al., 2020). ACTER is a collection of corpus texts in three languages (English, French and Dutch) and in four domains (corruption, dressage, heart failures and wind energy). Each corpus is accompanied by carefully and manually generated annotations – lists of unique terms contained in the texts of the corpus. Thus, all researchers are provided with a single framework and a single assessment methodology to collaborate on the task of automatic term extraction, allowing them to compare different approaches.

In this research, we use the ACTER dataset to evaluate the performance of the term extraction approach based on non-negative matrix factorization. The approach does not require training data and is invariant both to the domain and to the language. We compare it with five term extraction methods competed in TermEval 2020 (Hazem et al., 2020; Oliver & Vàzquez, 2020; Pais & Ion, 2020) and the four popular keyword extraction methods (TF-IDF, TextRank, RAKE, and YAKE) adapted to the term extraction task.

Let us consider a domain-oriented corpus of m documents with a dictionary of n words and phrases of this corpus as candidates for the domain terms. The corpus and the dictionary are used to construct the “documents-terms” matrix, which stores the frequencies of occurrences of candidates in the corresponding documents. It should be noted that the name of this matrix (“documents-terms”), which is well-established among NLP specialists, contains the word “terms”, which mean not only the sought-for domain terms, but all the candidates under consideration. Therefore, terms will be referred to the entire set of candidates, and the true terms to be extracted from the corpus will be referred as domain terms.

The idea of our method is to decompose the m × n “documents-terms” matrix into two non-negative $m \times k$ and $k \times n$ matrices, respectively, where $k \ll min(m, n)$. An illustrative scheme of such a decomposition, called non-negative matrix factorization, is shown in Fig. 1. The first of the two obtained matrices, called the basis matrix, specifies the representations of documents in a new k-dimensional space, the dimensions of which are no longer terms, but hidden factors, i.e., topics. The second of the matrices, called the coefficient matrix, stores the expansion coefficients of the vectors of the new k-dimensional basis in terms of the vectors of the original n-dimensional basis, i.e., describes the decomposition of topics into terms. We assume that terms contributing the most to the decomposition of topics, i.e., having the highest coefficients, can reflect domain terminology with sufficient precision and recall. Thus, this study is aimed to confirm this assumption, find the optimal parameters of the NMF-based approach, and demonstrate the performance, superior or comparable to the performance of baseline methods of automatic term extraction. Our main contribution is, accordingly, the exploration and comparison of the multiple existing NMF algorithms with multiple parameters for constructing the domain- and language-invariant topic model for automatic term extraction.

The rest of the manuscript is organized as follows. Section 2 describes related work on automatic term extraction. Section 3 provides the necessary theoretical information regarding NMF algorithms. Section 4 presents a term extraction methodology based on NMF topic modeling approach. Section 5 presents the experimental estimates and comparisons with the methods that participated in TermEval 2020, as well as with the methods for extracting the keywords (TF-IDF, TextRank, RAKE and YAKE). Section 6 formulates conclusions and a plan for future work.

2. Related work

A typical term extraction workflow consists of two principal subprocesses: (1) extracting candidate terms using linguistic and/or statistical filters, and (2) scoring and ranking candidate terms in order to select true terms (Nakagawa & Mori, 2003; Terryn et al., 2020; Zhang et al., 2018a). Depending on what type of filters (linguistic, statistical or both) are used in the workflow, the underlying methodology can be classified respectively as linguistic, statistical or hybrid. It has been established that hybrid methodology appears to outperform the other two, and nowadays, most advanced term extraction systems include linguistic filtering as an integral part of their structure (Drouin et al., 2020; Terryn et al., 2020). For this reason, we do not consider linguistic methods as a separate, stand-alone class in our taxonomy of term extraction methodologies.

We divide all these methodologies into four classes: (1) approaches based on contrastive distribution of terms, (2) approaches based on keyword extraction, (3) approaches based on supervised machine learning, (4) approaches based on topic modeling. Certainly, there are many other classification schemes (Astrakhantsev et al., 2015; Drouin et al., 2020; Korkontzelos et al., 2008), but we follow our own taxonomy since it covers the presented method and all the baseline methods considered in this study.

2.1. *Approaches based on contrastive distribution of terms*

Before we start to present the current status of the related work, we need to take a closer look to the definition of “domain term”. Domain terms or just terms are often defined as “linguistic expressions which characterize a domain” (Hätty, 2018; Kageura & Marshman, 2019; Nenadic et al., 2004; Vivaldi & Rodríguez, 2007), but such definitions leave room for questions what should be the measurable features that distinguish terms from ordinary words (Terryn et al., 2020). Vivaldi and Rodríguez (2007) assert that these termhood features can only be measured indirectly by means of other features easier to operationalize and assess, like frequency, statistical association measures, syntactical context exploration, etc.

One of the most intuitively simple idea to indirectly measure the termhood of candidate terms is to compare their distribution in the domain corpus with that in the generic reference corpus. This idea arose out of an observation that words or expressions which are often used in texts of a given domain and – in contrast – are rarely or not at all used in texts from other areas, most likely are terms of this domain (Nugumanova et al., 2016). Contrastive approaches work well in the extraction of so-called basic domain terms. In one of the early works using the contrastive approach, the Weirdness measure is used to assess termhood, which is the ratio of the frequency of use of a word in the domain corpus to the frequency of its use in the corpus of general topics (Ahmad et al., 1999). For ordinary words, this value is close to 1, and for terms it is much more than 1, because in this case, the denominator is close to 0. In (Basili et al., 2001), Contrastive Weight measure is proposed, which is directly proportional to the frequency of use of a word in the domain corpus and inversely proportional to the relative frequency of its use in contrastive corpora. The relative frequency is understood as the ratio of the total frequency of the use of a word in contrastive corpora to the total frequency of use in all corpora, including the domain one.

In later works on contrastive approach, termhood measures became more complex. For example, in (Meijer et al., 2014; Sclano & Velardi, 2007), it is proposed to use a linear combination of four indicators: Domain pertinence, Domain consensus, Lexical cohesion and Structural relevance. Domain pertinence is a measure of Weirdness generalized for the case of many contrastive corpora; it is equal to the ratio of the word frequency in the domain corpus to its highest frequency in contrastive corpora. Domain consensus considers the distribution of words in individual documents; it is the higher, the more evenly the word is distributed in the documents of the domain corpus. Lexical cohesion enhances the appreciation of verbose expressions, which are syntactically and semantically complete units of text. Structural relevance assigns higher scores to terms that appear in a title of a domain corpus document.

In (Mykowiecka et al., 2018) a measure of termhood based on the Context diversity coefficient is proposed, which checks the contexts of words in the general corpus. The measure is based on the observation that the domain terms in contrastive corpora usually do not occur on their own, but together with other terms from the same domain.

In (Kim et al., 2009; Lopes et al., 2016), varieties of the TF-IDF formula, widely used in information retrieval, are proposed, called Term frequency - inverse domain frequency (TF-IDF) and Term frequency - disjoint corpora frequency (TF-DCF), respectively. In the formula used in (Lopes et al., 2016), the relative frequency of the word in the domain corpus is taken as TF, and the logarithm of the ratio of the number of documents in all corpora to the number of documents in which this word is used at least once is taken as IDF. Thus, candidates are rewarded for high frequency of occurrence in the domain and penalized for occurrence in many documents. In the formula used in (Kim et al., 2009), the absolute frequency of the word in the domain corpus is taken as TF, and the product of the absolute frequencies of the word in Contrastive corpus is taken as DCF. Using a product instead of a sum allows the penalty to grow exponentially.

Despite the wide variety of contrastive approaches to term extraction, they all share a common limitation - reliance on well-balanced, contrastive corpora. In (Matsuo & Ishizuka, 2004), this limitation is partially overcome by comparing the occurrence of terms not in different corpora, but in different contexts within one corpus or document. The authors of the work analyze the joint occurrence of the word with the most frequent words of the document and conclude that if this distribution is biased towards a certain subset of the most frequent words, then this word is the keyword in the document.

2.2. *Keywords extraction approaches*

We have cited (Matsuo & Ishizuka, 2004) among works on term extraction, when in fact it is devoted to the keyword extraction. As noted in (Lossio-Ventura et al., 2013), the difference between term and keyword extraction is the same as between terms and keywords: the former are intended for describing domains, and the latter are for annotating individual documents. Accordingly, term extraction requires a large domain corpus, while keyword extraction requires only one document. Terms are always considered in relation to a specific domain, while keywords can refer to different domains if the content of the document is interdisciplinary.

Despite these differences, keyword extraction methods are often adapted to the task of term extraction (Lossio-Ventura et al., 2013; Zhang et al., 2018a). The simplest, although perhaps not the most efficient way, is to treat the domain corpus as one large document and extract keywords from it as terms. This method is used, for example, in (Zhang et al., 2016), where the RAKE and Chi-square methods, originally developed for extracting keywords from a single document, are adapted. The adaptation is done by replacing the word frequencies at the document level with those at the corpus level.

Another way is to extract keywords separately from each document in the corpus, and then combine them into a resulting set of terms. For example, in (Zhang et al., 2018a), the following adaptation of the TextRank keyword extraction method to the task of term extraction is proposed. First, in each document, the weights of all words included in this document are calculated based on the TextRank formula. Then, for each word, all its TextRank weights are summed over all documents where this word is included. The authors refer to the sums obtained as Corpus-level TextRank scores. Words with the highest Corpus-level TextRank scores form the final set of domain terms.

In (Zhang et al., 2018b), the original TextRank algorithm, which builds a document graph and connects two words (graph vertices) with edges, if they appear in this document in the context of each other, undergoes two changes. First, words are taken as vertices not from one document, but from the entire corpus. Secondly, words that appear in the same context not within a specific document, but within any document of the corpus, are connected by edges. The final refined term candidate score is a non-linear combination of the score calculated by one of the basic term extraction algorithms and the TextRank scores of its constituent words (if the term candidate consists of two or more words, then the TextRank scores of each word are added together).

As already noted, in this study, we consider four baseline methods TF-IDF, RAKE, YAKE and TextRank, which were originally created for keyword extraction and then adapted to term extraction (Pais & Ion, 2020). Although implementations of these methods are available through ‘pke’ free python library (Boudin, 2016), we do not use it in our experiments. We obtain the existing estimates of performance of these methods on ACTER Dataset directly from work (Pais & Ion, 2020).

2.3. *Supervised machine learning approaches*

The primary task of the approaches based on supervised machine learning is the selection of features in use and/or in the environment of words that would indicate their termhood (Zhang et al., 2018a). For example, these can be such signs as the frequency of the use of a word, spelling a word with a capital letter not at the beginning of a sentence, using quotation marks or special introductory words, etc. In (Nokel et al., 2012), a step-by-step greedy algorithm is used to select the most significant features. The algorithm starts with an empty set of features, and then at each step it adds a feature that maximizes the mean average precision measure. As a result, the algorithm selects the most effective combination of eight features for training the classifier.

The authors (Terryn et al., 2019) use 152 features, distributed into six groups: 1) morphological/formal (word length, capital letters, special characters); 2) frequency (word frequency in specialized corpora, for example, in the Wikipedia corpus); 3) statistical (termhood measures); 4) lexical (for example, information about the presence of words with the same lemma); 5) linguistic (POS tags); 6) corpus (information about the original corpus, etc.). All features are scaled and displayed in the interval [0, 1], and then fed to the input to the binary Decision Tree classifier. The authors train the classifier on the Dutch Wind Energy and Heart Failure corps in the ACTER dataset, and then, after analyzing the learning outcomes, they leave 136 features that are used to extract terms from the Dressage corpus. As expected by the authors, their approach allows achieving higher F-measure by combining dissimilar features. It is less dependent on frequency features, which results in fewer false positives when extracting terms with low frequency or false positives when extracting non-terms with high frequency. However, this approach is not without its drawbacks, the main of which are: 1) the need for a large amount of training data; 2) dependence on the content

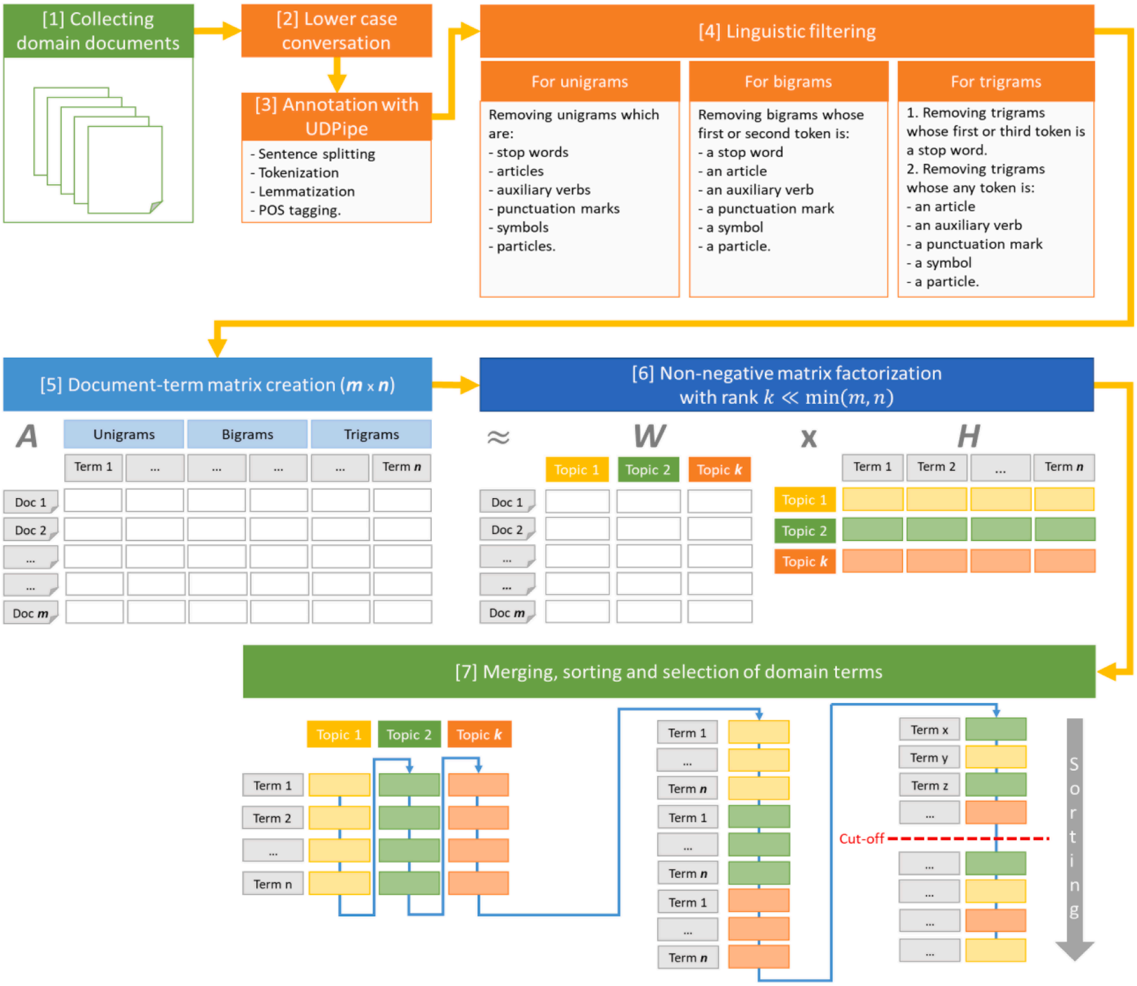

Fig. 2. The overall workflow of the used NMF-based methodology.

Table 1
UDPipe models for annotating the corpora.

No	Language	Model
1	English	English-gum-ud-2.5–191206.udpipe
2	French	French-gsd-ud-2.5–191206.udpipe
3	Dutch	Dutch-alpino-ud-2.5–191206.udpipe

Table 2
Examples of erroneous n-gram markup containing words with ’s and s’ endings.

No	Created keyword	Ngram	Pattern	Keyword after recoding
1	women ’s health	3	NNS POS NN	women’s health
2	protein ’s efficacy	3	NN POS NN	protein’s efficacy
3	veterans ’ affair	3	NNS POS NN	veterans’ affair

Table 3
Applications of NMF in the proposed study design.

NMF algorithm	Initialization algorithm			
	Fixed values (FV)	nndSVD	SPA	Fuzzy C-means (FCM)
FR	x	x	x	x
KL	x	x	x	x
PE	x	x	x	x
SL	x	x	x	x
SR	x	x	x	x

of training data, threatening to overfitting the classifier.

Overfitting is one of the most important reasons why approaches based on semi-supervised and weakly supervised learning are becoming more popular. In (Astrakhantsev, 2014), a bootstrap approach based on a partial learning algorithm is proposed and does not requires the labeled data. The idea is to extract the top 100–300 term candidates on Wikipedia and then use these candidates as positive examples for building a training model. Experiments on four domains (board games, biomedicine, computer science, agriculture) show the superiority of the proposed method. In (Repar et al., 2019; Šandrih et al., 2020), bilingual text corpora are used, which compensate for the absence of labeled data for low-resource languages.

In (Amjadian et al., 2016, 2018), it is proposed to use local–global word embeddings to represent words. Global embeddings are pre-trained on the general corpus, while local embeddings are pre-trained on the domain corpus. It is shown that only nine positive examples are enough to train the classifier on local–global vectors. The authors offer two options for using local–global vectors: a) as a filter for any high precision (even with very low recall) term extraction tool; b) as an independent method of term extraction. In the first option, a third-party high-precision algorithm extracts several term candidates, for which global–local embeddings are then built. These attachments are used as features for training the classifier.

(Hazem et al., 2020) use the transformer-based model BERT which has demonstrated high performance on the ACTER dataset during TermEval 2020 competition. As the authors claim, the capability of the attention mechanism of BERT to learn hidden features is less laborious

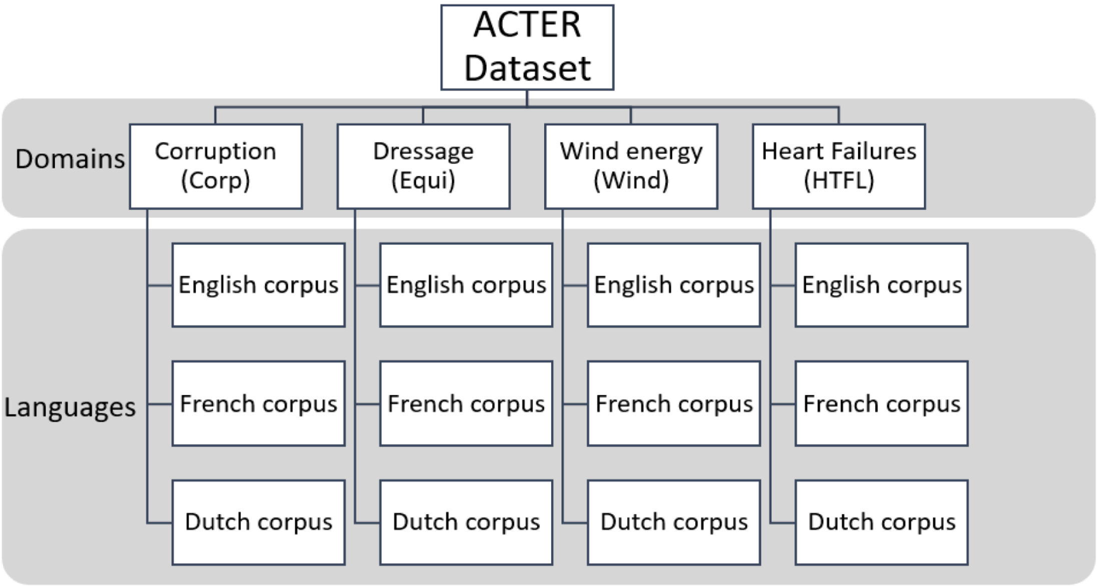

Fig. 3. ACTER Dataset structure.

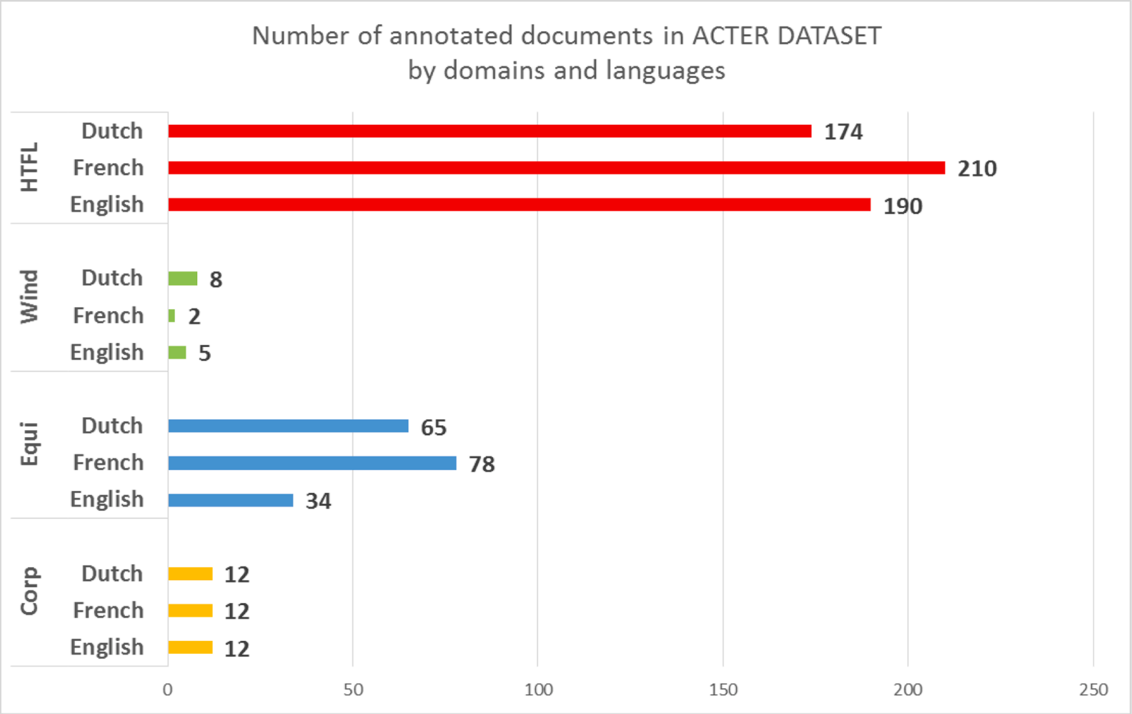

Fig. 4. Dimensions of the annotated parts of ACTER Dataset corpora.

and more efficient compared to machine learning methods which require feature extraction. They prove that transformer-based models can successfully coped with automatic term extraction challenge, without the need for text preprocessing or feature extraction (Lang et al., 2021). also confirm this finding by proposing three transformer-based models which are evaluated on the ACTER dataset. The proposed models outperform previous baselines of TermEval 2020 competition, including the above BERT-based model from (Hazem et al., 2020). All three proposed models demonstrate excellent zero-shot transfer learning capabilities, i.e., they are trained on one language and show strong test scores on another.

2.4. *Topic modeling approaches*

The approach we apply in this study exploits the same idea as the approaches collected in this group. The essence of this idea is to map the existing document corpus of the domain into a semantic space consisting of several topics. Then the probabilities of the distribution of words in topics or the weight of words in topics (this depends on which model is taken as a basis) are used to assess the termhood of words. In (Bougouin et al., 2013; Liu et al., 2010; Sterckx et al., 2015; Teneva & Cheng, 2017), topic models are combined with graph models, which leads to the emergence of interesting metrics for evaluating termhood such as Topical PageRank. Although these metrics are calculated for keyword

Table 4
Numerical characteristics of the corpora used in the experiments.

Domain	Language	Number of reference (gold) domain terms extracted by human experts		Document-term matrix dimensions		Number of candidate terms matched with gold domain terms (in case of without NEs)	Maximum limit of Recall (%) (in case of without NEs)
		With NEs	Without NEs	Number of rows (documents)	Number of columns (candidate terms)		
Corp	English	1174	927	12	11 891	862	92.99
	French	1207	979	12	14 222	822	83.96
	Dutch	1295	1 047	12	14 709	981	93.70
Equi	English	1575	1 155	34	13 127	1 056	91.43
	French	1181	961	78	18 350	808	84.08
	Dutch	1544	1 393	65	15 070	1 254	90.02
Wind	English	1534	1 091	5	15 294	1 033	94.68
	French	968	773	2	15 397	619	80.08
	Dutch	1245	940	8	19 529	895	95.21
HTFL	English	2585	2 361	190	19 367	2 084	88.27
	French	2374	2 228	210	17 544	1 982	88.96
	Dutch	2254	2 074	174	19 487	1 948	93.92

Table 5
Best NMF run results on ACTER Dataset corpora (without NEs).

	English			French			Dutch		
	F1	Algorithm	k	F1	Algorithm	k	F1	Algorithm	k
Corp	23.98	FR + nndSVD FR + SPA FR + FCM	7, 9	19.83	FR + nndSVD FR + SPA FR + FCM	7	24.13	FR + SPA FR + FCM	9
Equi	31.17	KL + FV	5	25.60	KL + nndSVD	9, 15	33.07	KL + SPA	3
Wind	24.69	FR + nndSVD	3	17.69	SL + nndSVD	2	18.84	KL + nndSVD	5
HTFL	33.51	KL + SPA	5, 7	30.89	KL + FCM	2	30.12	KL + FV	2

Table 6
Best NMF run results on ACTER Dataset corpora (with NEs).

	English			French			Dutch		
	F1	Algorithm	k	F1	Algorithm	k	F1	Algorithm	k
Corp	25.72	FR + nndSVD	7	21.91	FR + nndSVD	7	25.76	FR + SPA	9
Equi	33.28	KL + SPA	5	27.20	KL + nndSVD	9	32.73	KL + SPA	3
Wind	26.14	FR + nndSVD	3	18.37	SL + nndSVD	2	20.35	KL + nndSVD	5
HTFL	33.71	KL + SPA	5	30.67	KL + SPA	7	30.25	KL + FV	2

Table 7
Comparison of the best results of NMF-based approach with results of the baseline methods (TF-IDF, RAKE, YAKE and TextRank) on English corpora (relative to the list of reference terms with NEs).

	The best result of NMF		TF-IDF		RAKE		YAKE		TextRank	
	F1%	Rank	F1%	Rank	F1%	Rank	F1%	Rank	F1%	Rank
Corp	**25.72**	1	11.01	4	21.6	2	10.76	5	17.70	3
Equi	**33.28**	1	13.71	5	24.95	3	22.30	4	26.04	2
Wind	**26.14**	1	12.34	4	22.79	2	5.94	5	13.86	3
HTFL	**33.71**	1	13.95	5	29.27	2	18.47	4	25.17	5

extraction, as noted above, they can be used for term extraction as well.

In (Bolshakova et al., 2013), for automatic extraction of one-word terms, several topic models are used at once: K-means, NMF, LDA and hierarchical clustering algorithms Single-linkage, Complete-linkage, and Average-linkage. Then, well-known termhood measures such as TF and TF-IDF are redefined based on the distribution of words by topic. In particular, the frequency of a word (TF) in the corpus is replaced by the total probability or the total weight of the word for all topics, and the number of documents (DF) containing the given word is replaced by the number of topics containing the given word. The best performance is demonstrated by the NMF model with the Kullback-Leibler distance and the term score, which is an extended version of the TF-IDF measure (Bolshakova et al., 2013).

In (Li et al., 2013), a topic i-SWB model is proposed, designed to map the corpus of documents of the domain into a hidden semantic space, consisting of some general topics, the main topic of the corpus and topics specific to documents. For each of the listed topics, the 200 most probable (typical) words are extracted. The idea of the authors is that a good term candidate should be composed of typical words representing a specific topic.

Table 8
Comparison of the best results of NMF-based approach with results of methods competed in TermEval 2020 task (on the HTFL test corpora).

Track	Rank	Team/approach	Scores with NEs (%)			Scores without NEs (%)		
			Pr	Re	F1	Pr	Re	F1
English	1	TALN-LS2N	34.8	70.9	46.7	32.6	72.7	45.0
	2	RACAI	42.4	40.3	41.3	38.6	40.1	39.3
	3	NMF-based approach	30.47	37.72	33.7	29.5	38.8	33.5
	4	NYU	43.5	23.6	30.6	42.2	25.1	31.5
	5	e-Termhood	34.4	14.2	20.1	34.4	15.5	21.4
	6	NLPLab UQAM	21.4	15.6	18.1	20.1	16.0	17.8
French	1	TALN-LS2N	45.2	51.5	48.1	41.9	50.9	45.9
	2	NMF-based approach	27.5	34.7	30.7	26.5	36.9	30.9
	3	e-Termhood	36.3	13.5	19.7	36.3	14.4	20.6
	4	NLPLab UQAM	16.1	11.2	13.2	15.1	11.2	12.9
Dutch	1	NMF-based approach	23.4	42.6	30.3	22.7	44.8	30.1
	2	NLPLab UQAM	18.9	18.6	18.7	18.1	19.3	18.6
	3	e-Termhood	29.0	9.6	14.4	29.0	10.4	15.3

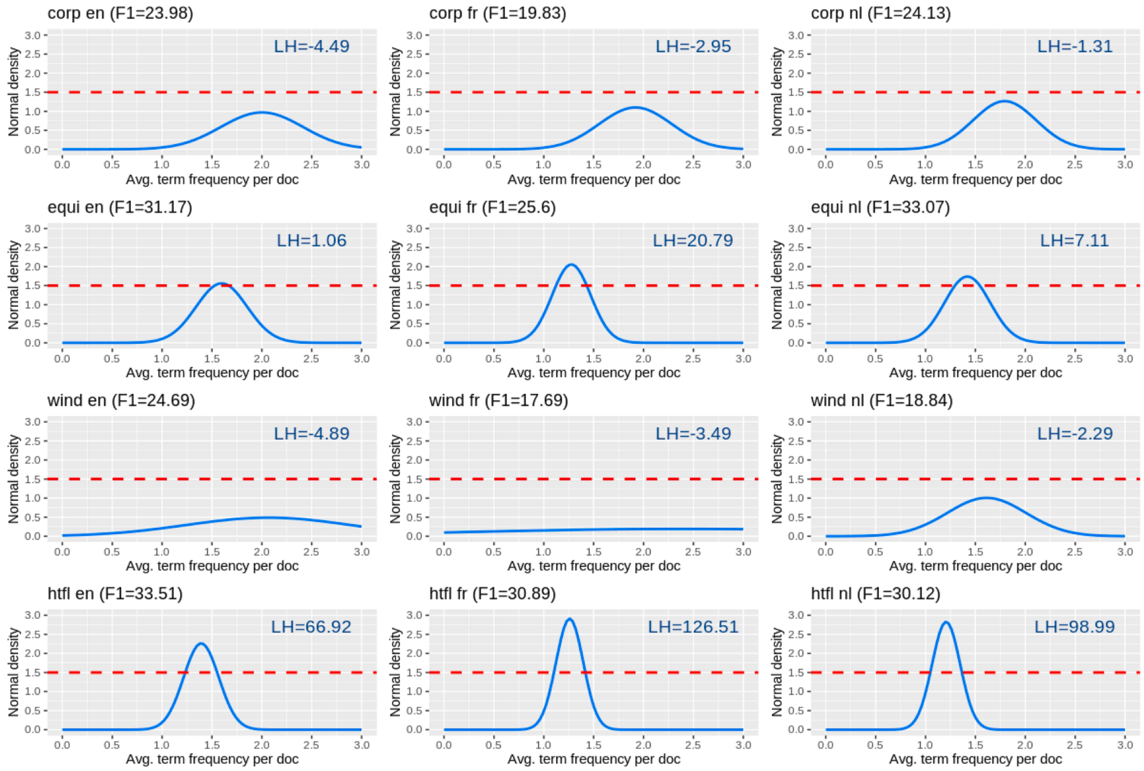

Fig. 5. The normal distributions approximating term distributions in various corpora of ACTER Dataset (the best NMF results, i.e., F1-scores are shown in brackets next to the corpora names).

In (Süzek, 2017) the LSA topic model is used to extract keywords from a single document. Like other keyword extraction methods, the approach can be adapted to term extraction as well. First, a term-sentence matrix $A_{n\times s}$ is formed, the rows of which correspond to words and the columns to sentences in the document. The elements of this matrix are the frequencies of the use of words in the corresponding sentences. Then, for $A_{n\times s}$ matrix, its optimal singular value decomposition is formed in the form of the product of three matrices $U_{n\times k}$, $S_{k\times k}$ and $(V_{s\times k})^T$. In this decomposition, the author of the work is only interested in the first column of the matrix U, which is considered the main topic of the document. The most important keywords annotating a document are determined by selecting the top N values in this first column.

In (Abuzayed & Al-Khalifa, 2021) probabilistic topic modeling is considered from the perspective of deep learning. The authors present BERTopic, a novel transformer-based topic modeling framework. They start with LDA and NMF as baselines and then use BERTopic with various word embedding representations and with different topics numbers. This pilot study shows promising results of BERTopic on popular news datasets, but the authors do not use their model for automatic term extraction.

In the very recent study by (Wang & Zhang, 2021) it is proposed to apply a non-neural-network unsupervised deep model to probabilistic topic modeling. The authors design a deep NMF topic modeling framework to alleviate the issues of bad local minima and high computational complexity intrinsic to NMF models. Although the authors do not apply their framework to automatic term extraction task, the proposed methodology is able to cover all the applications of probabilistic topic modeling including term extraction. Experimental results illustrate that

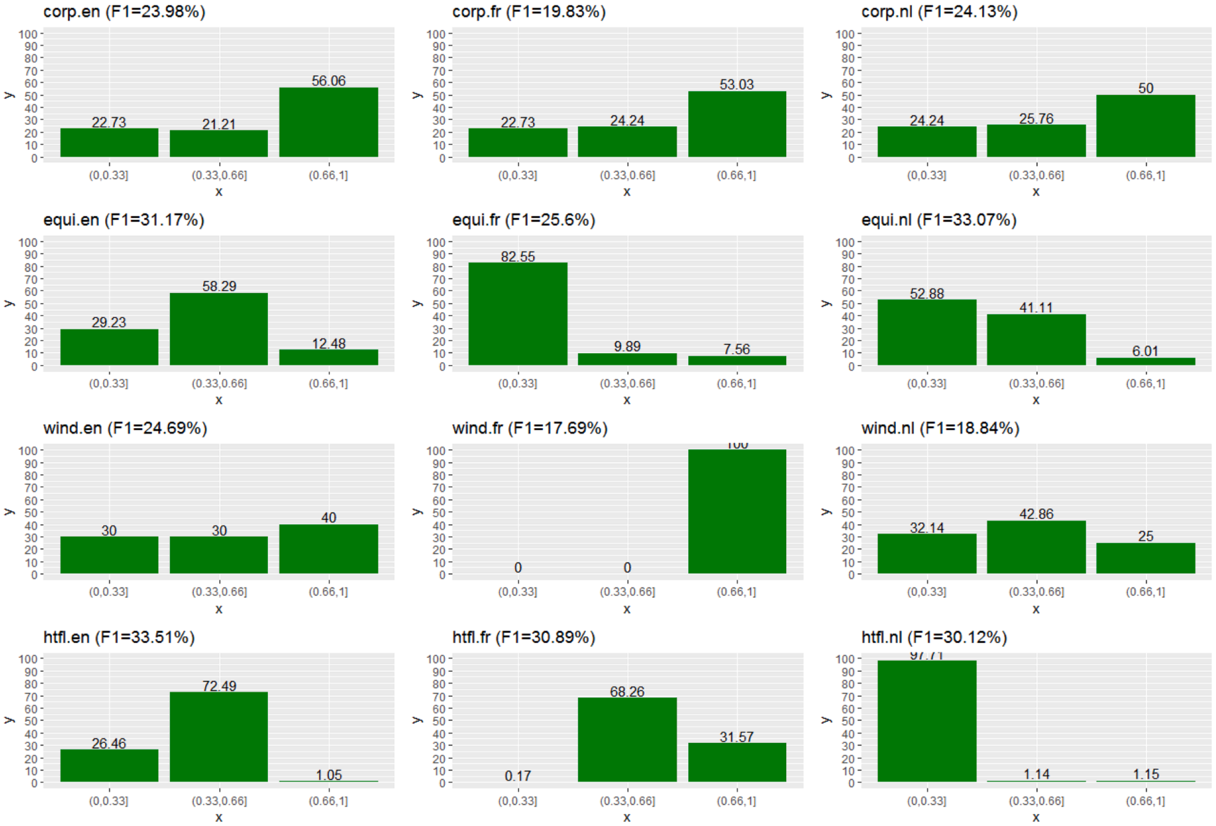

Fig. 6. Distribution of close, medium, and far distances between documents in the various corpora of ACTER Dataset (best NMF results from Table 5 are shown in brackets next to the corpora names).

the deep NMF topic modeling methods outperform conventional shallow topic modeling methods significantly.

Finally, another recent and large study by (De Handschutter et al., 2021) presents the main models, algorithms, and applications of deep matrix factorization through a comprehensive literature review. According to the authors, "deep matrix factorization is likely become an important paradigm unsupervised learning in the next few years". These recent studies give a new impetus to research in the field of document topic modeling since it seems that the existing fundamental limitations of NMF can be overcome using deep learning models.

3. Background

Non-negative matrix factorization (NMF) has proven to be highly useful in data representation and topic modeling. In this Section, we provide the minimal theoretical background necessary to follow our approach. We start in Section 3.1 by briefly representing the classical NMF problem statement. Then, in Section 3.2 we describe how to enhance the classical NMF problem statement by incorporating regularization functions in the objective function. In Section 3.3, we outline some strategies to initialize NMF algorithms in order to provide better interpretable results or better convergence.

3.1. Classical NMF

In general, NMF is used to represent the original non-negative $A \in \mathbb{R}^{m\times n}$ matrix in the form of two non-negative matrices of lower rank, $W \in \mathbb{R}^{m\times k}$ and $H \in \mathbb{R}^{k\times n}$, which, when multiplied, approximately restore A:

$$A \approx WH \tag{1}$$

Without loss of generality, the rows of the matrix A can be considered objects or samples (in our case, documents), and the columns – features (in our case, terms). As noted above, the matrix W is called the basis matrix, and the matrix H is called the coefficient matrix.

The goal of the classical NMF is to minimize the loss function L arising from the transition from the original matrix A to the matrix WH:

$$\min_{W\geq 0, H\geq 0} L(A, WH) \tag{2}$$

The loss function L is often chosen as the standard deviation, i.e. the Frobenius norm (3), or as the Kullback-Leibler divergence (4):

$$L(A,\ B) = \|A - B\|^2 = \sum_{ij}(a_{ij} - b_{ij})^2, \tag{3}$$

$$L(A,\ B) = \sum_{ij}\left(a_{ij}log\frac{a_{ij}}{b_{ij}} - a_{ij} + b_{ij}\right) \tag{4}$$

where a_{ij}, b_{ij} are elements of matrices A and B, respectively.

The process of solving Eq. (2) usually begins with the initialization of the matrices W and H with random values, which are updated iteratively. In the case of Frobenius norm, Eq. (3), these multiplicative update rules look as follows (Lee & Seung, 2000):

$$H_{[i,j]} \leftarrow H_{[i,j]} \bullet \frac{(W^T A)_{[i,j]}}{(W^T WH)_{[i,j]}} \tag{5}$$

$$W_{[i,j]} \leftarrow W_{[i,j]} \bullet \frac{(AH^T)_{[i,j]}}{(WHH^T)_{[i,j]}} \tag{6}$$

The iterations are repeated until the W and H values stabilize. Updating the matrices is carried out element by element, so that each zero element of the W and H matrices remains zero throughout the entire iteration process, which makes it easy to impose constraints on factorization, i.e. to adjust matrices W and H as desired (Lee & Seung, 2000).

Table A1
NMF results on Corp corpus for English at k = 2,3,5,7,9 (without NEs).

k = 2	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	15.24	41.10	22.24	15.36	41.42	22.41	15.24	41.10	22.24	15.24	41.10	22.24
KL	16.85	36.35	23.03	**15.10**	**50.49**	**23.25**	14.87	49.73	22.89	14.54	43.91	21.85
PE	15.95	36.14	22.13	15.36	41.42	22.41	15.24	41.10	22.24	15.24	41.10	22.24
SL	15.70	38.94	22.38	NA	NA	NA	NA	NA	NA	15.70	38.94	22.38
SR	15.62	40.45	22.54	NA	NA	NA	15.62	40.45	22.54	15.62	40.45	22.54
k = 3	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	14.80	39.91	21.59	15.36	41.42	22.41	15.78	39.16	22.50	15.78	39.16	22.50
KL	15.29	39.59	22.06	**15.67**	**45.63**	**23.33**	16.40	35.38	22.41	14.59	42.50	21.72
PE	15.00	38.83	21.64	15.36	41.42	22.41	16.29	36.89	22.60	16.09	38.19	22.64
SL	14.79	38.30	21.34	NA	NA	NA	NA	NA	NA	15.30	33.01	20.91
SR	14.75	38.19	21.28	NA	NA	NA	15.35	33.12	20.98	15.35	33.12	20.98
k = 5	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	16.25	35.06	22.21	15.36	41.42	22.41	16.29	36.89	22.60	16.09	38.19	22.64
KL	15.50	40.13	22.36	**14.64**	**52.10**	**22.86**	16.30	35.17	22.28	16.05	34.63	21.93
PE	15.00	42.07	22.11	15.36	41.42	22.41	15.08	42.29	22.23	16.65	35.92	22.75
SL	14.92	38.62	21.52	NA	NA	NA	NA	NA	NA	16.24	36.79	22.53
SR	15.08	39.05	21.76	NA	NA	NA	15.08	39.05	21.76	15.08	39.05	21.76
k = 7	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	**17.55**	**37.86**	**23.98**	15.36	41.42	22.41	**17.55**	**37.86**	**23.98**	**17.55**	**37.86**	**23.98**
KL	17.40	37.54	23.78	14.88	51.35	23.07	16.25	35.06	22.21	16.60	35.81	22.68
PE	15.04	42.18	22.17	15.36	41.42	22.41	16.65	35.92	22.75	16.65	35.92	22.75
SL	15.76	42.50	22.99	NA	NA	NA	NA	NA	NA	15.76	42.50	22.99
SR	16.10	34.74	22.00	NA	NA	NA	16.10	34.74	22.00	14.81	41.53	21.83
k = 9	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	**17.55**	**37.86**	**23.98**	15.36	41.42	22.41	**17.55**	**37.86**	**23.98**	**17.55**	**37.86**	**23.98**
KL	17.40	37.54	23.78	14.88	51.35	23.07	16.25	35.06	22.21	16.60	35.81	22.68
PE	15.04	42.18	22.17	15.36	41.42	22.41	16.65	35.92	22.75	16.65	35.92	22.75
SL	15.76	42.50	22.99	NA	NA	NA	NA	NA	NA	15.76	42.50	22.99
SR	16.10	34.74	22.00	NA	NA	NA	16.10	34.74	22.00	14.81	41.53	21.83

There are other ways to find the matrices W and H, for example, the method of least squares (Kim et al., 2007). At each step of this method, first H is fixed, W is found using the least squares method, then W is fixed, and H is found using the least squares method. It is worth noting that the NMF decomposition, Eq. (1), is not unique (Xu et al., 2003). If we replace matrix W with matrix $W \bullet X^T$ and matrix H with matrix $X \bullet H$ we obtain the same value of A. To the best of our knowledge, the issue of rotations is poorly explored in the studies devoted to NMF-based topic modeling.

At present, all the developed NMF algorithms are suboptimal in the sense that they guarantee that only the local rather than the global minimum of the objective function is found. In the near future, the creation of NMF algorithm capable of finding the global minimum is not expected, since it is shown that the NMF problem is NP-complete (Vavasis, 2009). However, as numerous NMF applications show, even suboptimal solutions provide tremendous benefits in data mining, natural language processing, and other data decomposition tasks.

3.2. Regularized NMF

In the extended (regularizable) NMF model, additional restrictions are imposed on the solution of problem (1) in the form of $J_W(W)$ and $J_H(H)$ functions, designed to encourage the desired properties of W and H matrices:

$$\min_{W\geq 0, H\geq 0}[L(A, WH) + J_W(W) + J_H(H)] \tag{7}$$

These can be properties such as smoothness or sparsity of the matrix, lower magnitude of values, or better orthogonality of rows/columns. $J_W(W)$ and $J_H(H)$ functions are called regularizations, and in general they are written as follows:

$$J_W(W) = \alpha_1 J_1(W) + \alpha_2 J_2(W) + \alpha_3 J_3(W) \tag{8}$$

$J_H(H) \ = \beta_1 J_1(H) \ + \beta_2 J_2(H) + \beta_3 J_3(H)$, (9).where J_1 is commonly used to control the magnitude and smoothness of the matrix, also helps to stabilize numerical algorithms; J_2 is used to minimize correlations between columns, that is, to maximize the linear independence of the columns; J_3 is used for LASSO regularizations, which controls the sparsity of the matrix.

There are many ways to regularize NMF. For example, in (Mirzal, 2014), the Tikhonov regularization is used, which ensures the smoothness of the obtained solutions:

$$\min_{W\geq 0, H\geq 0}\left[\frac{1}{2}\|A - WH\|^2 + \frac{1}{2}\beta\|W\|^2 + \frac{1}{2}\alpha\|H\|^2\right] \tag{10}$$

where α, β are regularization parameters, the selection of which is a separate problem. If the parameters are too large, then the approximation error increases, most of the information in the original data is lost, and if the parameters are too small, then the noise characteristic of unregulated models prevails in the solutions.

In (Ang & Gillis, 2019), a volume regularization is proposed that minimizes the volume of the convex hull spanned by the basis matrix:

Table A2
NMF results on Corp corpus for French at k = 2,3,5,7,9 (without NEs).

k = 2	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	12.47	38.20	18.80	13.50	30.34	18.69	12.47	38.20	18.80	12.47	38.20	18.80
KL	12.59	34.73	18.48	**13.52**	**31.77**	**18.97**	13.21	32.38	18.76	13.50	27.58	18.13
PE	12.47	38.20	18.80	13.50	30.34	18.69	12.47	38.20	18.80	12.47	38.20	18.80
SL	13.17	32.28	18.71	NA	NA	NA	NA	NA	NA	13.17	32.28	18.71
SR	12.81	34.01	18.61	NA	NA	NA	12.81	34.01	18.61	12.81	34.01	18.61
k = 3	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	13.55	30.44	18.75	13.50	30.34	18.69	13.55	30.44	18.75	13.55	30.44	18.75
KL	12.44	34.32	18.26	**14.05**	**30.13**	**19.16**	13.32	29.93	18.44	12.84	32.79	18.45
PE	13.25	32.48	18.82	13.50	30.34	18.69	13.76	29.52	18.77	13.76	29.52	18.77
SL	13.15	26.86	17.66	NA	NA	NA	NA	NA	NA	13.15	26.86	17.66
SR	12.67	31.05	18.00	NA	NA	NA	13.15	26.86	17.66	13.15	26.86	17.66
k = 5	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	14.65	29.93	19.67	13.50	30.34	18.69	**14.27**	**32.07**	**19.75**	14.27	32.07	19.75
KL	12.25	40.04	18.76	13.68	30.75	18.94	12.16	38.51	18.48	12.20	37.39	18.40
PE	14.14	30.34	19.29	13.50	30.34	18.69	12.25	25.03	16.45	13.86	31.15	19.18
SL	12.22	33.71	17.94	NA	NA	NA	NA	NA	NA	12.59	28.29	17.43
SR	12.84	32.79	18.45	NA	NA	NA	13.27	29.83	18.37	13.27	29.83	18.37
k = 7	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	**13.96**	**34.22**	**19.83**	13.50	30.34	18.69	**13.96**	**34.22**	**19.83**	**13.96**	**34.22**	**19.83**
KL	13.52	29.01	18.44	13.57	31.87	19.04	14.45	29.52	19.40	14.45	29.52	19.40
PE	14.40	29.42	19.34	13.50	30.34	18.69	13.29	32.58	18.88	13.29	32.58	18.88
SL	12.52	37.08	18.72	NA	NA	NA	NA	NA	NA	12.52	37.08	18.72
SR	13.30	31.26	18.66	NA	NA	NA	13.30	31.26	18.66	13.30	31.26	18.66
k = 9	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	12.33	37.79	18.59	13.50	30.34	18.69	12.33	37.79	18.59	12.33	37.79	18.59
KL	13.41	30.13	18.56	14.25	29.11	19.13	**14.50**	**29.62**	**19.47**	**14.50**	**29.62**	**19.47**
PE	14.14	30.34	19.29	13.50	30.34	18.69	13.52	31.77	18.97	13.52	31.77	18.97
SL	13.80	28.19	18.53	NA	NA	NA	NA	NA	NA	13.80	28.19	18.53
SR	13.13	30.85	18.42	NA	NA	NA	13.13	30.85	18.42	13.13	30.85	18.42

$$\min_{W\geq0,H\geq0,\,H^T1_r\leq1_n}\left[\frac{1}{2}\|A-WH\|^2+\lambda V(W)\right] \tag{11}$$

where $\lambda \geq 0$ is the regularization parameter, a compromise between the standard objective function and the volume regularizer *V (W)*. $H^T1_k \leq 1_n$ constraint relaxes the $H^T1_k = 1_n$ constraint, which requires that the sum of the elements of each column of the matrix H be equal to 1.

In (Zhang et al., 2008), PE-NMF regularization is proposed providing effective blind source separation. The objective function in the PE-NMF model is a special case of function (7), the first term of which tries to form as orthogonal column vectors of W matrix as possible, and the second term is the uniform representation of all column vectors of H matrix in the data expression:

$$J_W(W) = \alpha\sum_{i,j,i\neq j} W_i^T W_j \tag{12}$$

$$J_H(H) = \beta\sum_{i,j} h_{ij} \tag{13}$$

where W_i is the i-th column of the matrix W, h_{ij} is the element of the matrix H at the intersection of the i-th row and j-th column, and α and β are parameters that determine the compromise between $J_W(W)$ and $J_H(H)$.

3.3. NMF initialization

NMF algorithms are practically unsupervised, they perform blind decomposition, which sometimes calls into question the meaning of the result (Zhang et al., 2008). On the one hand, this can limit the use of unsupervised methods in areas where high interpretability is critically important (for example, in biomedical research); on the other hand, decomposition without using prior knowledge can be ineffective, especially with a small sample size (Ang & Gillis, 2019).

To overcome these problems, it is recommended to run NMF algorithms not with random initial values of the matrices W and H, but with values obtained based on prior experience or assumptions about the nature of the data. Such initialization is often capable of "directing" the search for the minimum value of the objective function in the right direction, i.e. can provide faster convergence to a local minimum (Wild, 2003).

In (Langville et al., 2014), as an initialization method, it is proposed to use an optimal singular value decomposition, which allows representing the original matrix A as a product of three matrices $U \in \mathbb{R}^{m\times r}$, $\Sigma = diag(\sigma_1, \sigma_2, \cdots, \sigma_r) \in \mathbb{R}^{r\times r}$ and $V^T \in \mathbb{R}^{r\times n}$:

$$A \approx U\Sigma V^T \tag{14}$$

where $\sigma_1, \sigma_2, \cdots, \sigma_r$ are the singular values of the matrix A, ordered in descending order, the columns of the matrices U and V are called left and right singular vectors, respectively. It is proposed to initialize the values of the matrices W and H with the absolute values of the matrices U andΣV^T.

$$W_0 = |U| H_0 = \left|\Sigma V^T\right| \tag{15}$$

Table A3
NMF results on Corp corpus for Dutch at k = 2,3,5,7,9 (without NEs).

k = 2	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	15.52	40.02	22.37	15.44	39.83	22.25	15.52	40.02	22.37	15.52	40.02	22.37
KL	**16.00**	**41.26**	**23.06**	15.93	41.07	22.96	16.43	36.10	22.58	14.82	46.70	22.50
PE	15.58	38.68	22.21	15.44	39.83	22.25	15.69	38.97	22.37	15.73	39.06	22.43
SL	15.96	35.05	21.93	NA	NA	NA	NA	NA	NA	15.96	35.05	21.93
SR	16.09	35.34	22.11	NA	NA	NA	16.09	35.34	22.11	16.09	35.34	22.11
k = 3	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	15.46	38.40	22.04	15.44	39.83	22.25	15.50	38.49	22.10	15.50	38.49	22.10
KL	15.83	43.84	23.26	**15.68**	**46.42**	**23.44**	15.19	46.42	22.89	16.04	35.24	22.05
PE	15.22	39.26	21.94	15.44	39.83	22.25	15.50	38.49	22.10	15.35	38.11	21.89
SL	14.68	35.05	20.69	NA	NA	NA	NA	NA	NA	16.90	32.28	22.19
SR	15.62	31.33	20.85	NA	NA	NA	15.62	31.33	20.85	15.62	31.33	20.85
k = 5	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	15.25	40.78	22.20	15.44	39.83	22.25	**16.86**	**35.43**	**22.85**	**16.86**	**35.43**	**22.85**
KL	15.60	37.25	21.99	15.63	40.31	22.53	16.43	36.10	22.58	15.27	37.92	21.77
PE	15.19	39.16	21.89	15.44	39.83	22.25	15.23	37.82	21.72	15.23	37.82	21.72
SL	14.92	37.06	21.27	NA	NA	NA	NA	NA	NA	16.17	37.06	22.52
SR	14.92	35.63	21.03	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 7	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	17.55	36.87	23.78	15.44	39.83	22.25	**17.13**	**39.26**	**23.85**	**17.13**	**39.26**	**23.85**
KL	17.33	34.77	23.13	16.38	37.54	22.81	15.23	37.82	21.72	16.48	33.05	21.99
PE	16.50	31.52	21.66	15.44	39.83	22.25	15.12	37.54	21.56	15.12	37.54	21.56
SL	16.04	39.83	22.87	NA	NA	NA	NA	NA	NA	15.17	42.02	22.29
SR	16.50	34.67	22.36	NA	NA	NA	16.23	34.10	21.99	16.23	34.10	21.99
k = 9	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	16.85	41.83	24.02	15.44	39.83	22.25	17.12	40.88	24.13	17.12	40.88	24.13
KL	17.50	33.43	22.97	16.35	35.91	22.47	17.24	34.57	23.01	17.24	34.57	23.01
PE	16.65	31.81	21.86	15.44	39.83	22.25	16.10	32.28	21.48	16.10	32.28	21.48
SL	17.70	33.81	23.24	NA	NA	NA	NA	NA	NA	17.70	33.81	23.24
SR	17.29	34.67	23.07	NA	NA	NA	17.29	34.67	23.07	17.29	34.67	23.07

In this case, because of replacing negative elements of the matrices U and ΣV^T with positive ones, the approximation error increases. However, some part of the initial information is stored in the initial matrices W_0 and H_0, obtained from the positive components of the singular vectors, and this gives reason to discuss more efficient initialization than initialization based on random values.

In (Qiao, 2015), initialization based on SVD is carried out in two stages, which gave rise to its name "non-negative double singular value decomposition" (nndSVD). At the first stage, the initial data matrix is approximated using SVD; at the second step, the positive components of the resulting singular vectors are also approximated using SVD.

In (Boutsidis & Gallopoulos, 2008), the SQR-NMF algorithm is proposed as an initialization algorithm, which uses the spectral decomposition of the square matrix $A' \in R^{p\times p}$ $(p = \max(m,n))$ obtained from the original matrix $A \in \mathbb{R}^{m\times n}$ by adding missing rows/columns with elements equal to 0, 1 or the average of the remaining elements in columns/rows, respectively:

$$A' \approx V\Lambda V^{-1} \tag{16}$$

where V is a matrix the columns of which are the eigenvectors of the matrix A, Λ is a diagonal matrix with the corresponding eigenvalues on the main diagonal. It is proposed to initialize the values of the matrices W and H with the absolute values of the matrices V andΛV^{-1}.

$$W_0 = |V| = \left|\Lambda V^{-1}\right| \tag{17}$$

Fuzzy clustering based on C-means is used as an initialization algorithm in (Yueyang & Shafai, 2018). The Fuzzy C-means algorithm seeks to split the original data (in our case, the document-term matrix) into C fuzzy clusters. Thus, the degree of membership of document $i(i = \overline{1,m})$ in cluster $j(j = \overline{1,r})$ is determined by the weight value w_{ij}. To initialize the matrix W, the centroids of the clusters are taken, they are used as the columns of the matrix W_0. To initialize the matrix H, the weights w_{ij} are taken, they serve as rows of the matrix H_0.

In (Rezaei et al., 2011), the sequential projection algorithm (SPA) is proposed as an initialization algorithm, the first step of which is to select the point $p1$ with the largest $\lambda 2$-norm. All data points are then projected onto the orthogonal complement to $p1$. Then the point $p2$ with the highest $\lambda 2$-norm in this projected subspace is chosen. The next point $p3$ is chosen as the data point with the highest $\lambda 2$-norm after projection onto the orthogonal complement to $p1$ and $p2$, and so on. To initialize the matrix W, the obtained points $p1, p2, \ldots, pr$ are used as columns of the matrix W_0. The SPA algorithm does not initialize the matrix H; the least squares method is used to find the initial matrix H_0.

4. Methodology

The overall workflow of the methodology used is shown in Fig. 2. The workflow involves seven stages (marked in the figure by numbers). Stages 1–5 are preparatory. The workflow input receives a set of text documents of the domain, which, after going through stages 1, 2 and 3, is converted into an annotated corpus. At stage 4, all unigrams, bigrams and trigrams are extracted from the annotated corpus, which are then passed through linguistic filters. Those of them that have passed the

Table A4
NMF results on Equi corpus for English at k = 2,3,5,7,9,15,20 (without NEs).

k = 2	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	22.43	44.68	29.87	**24.05**	**43.72**	**31.03**	22.35	44.50	29.76	22.35	44.50	29.76
KL	23.14	44.07	30.35	23.95	43.55	30.90	23.45	40.61	29.73	22.91	45.63	30.50
PE	23.00	39.83	29.16	24.05	43.72	31.03	23.70	41.04	30.05	23.65	40.95	29.98
SL	22.70	39.31	28.78	NA	NA	NA	NA	NA	NA	22.70	39.31	28.78
SR	22.60	39.13	28.65	NA	NA	NA	22.60	39.13	28.65	22.60	39.13	28.65
k = 3	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	22.57	44.94	30.05	**24.05**	**43.72**	**31.03**	22.57	44.94	30.05	22.57	44.94	30.05
KL	23.75	41.13	30.11	23.17	46.15	30.85	23.41	44.59	30.70	24.25	41.99	30.74
PE	23.40	40.52	29.67	**24.05**	**43.72**	**31.03**	23.50	40.69	29.79	23.45	40.61	29.73
SL	22.50	38.96	28.53	NA	NA	NA	NA	NA	NA	22.50	38.96	28.53
SR	22.80	39.48	28.91	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 5	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	24.30	42.08	30.81	24.05	43.72	31.03	23.32	44.42	30.58	24.25	41.99	30.74
KL	24.45	42.34	31.00	**23.77**	**45.28**	**31.17**	23.32	44.42	30.58	23.52	42.77	30.35
PE	23.70	41.04	30.05	24.05	43.72	31.03	23.80	41.21	30.17	23.80	41.21	30.17
SL	23.25	40.26	29.48	NA	NA	NA	NA	NA	NA	22.14	42.16	29.03
SR	22.50	38.96	28.53	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 7	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	24.15	41.82	30.62	**24.05**	**43.72**	**31.03**	23.20	40.17	29.41	23.20	40.17	29.41
KL	23.85	41.30	30.24	23.90	43.46	30.84	22.20	38.44	28.15	21.96	43.72	29.24
PE	23.75	41.13	30.11	24.05	43.72	31.03	22.14	40.26	28.57	22.14	40.26	28.57
SL	22.60	39.13	28.65	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	21.70	37.58	27.51	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 9	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	23.10	41.99	29.80	**24.05**	**43.72**	**31.03**	23.45	40.61	29.73	23.45	40.61	29.73
KL	21.79	45.28	29.42	24.25	41.99	30.74	21.30	36.88	27.00	20.62	46.41	28.55
PE	23.45	40.61	29.73	**24.05**	**43.72**	**31.03**	21.70	43.20	28.89	21.95	41.82	28.79
SL	23.85	41.30	30.24	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	22.15	38.35	28.08	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 15	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	23.70	41.04	30.05	**24.05**	**43.72**	**31.03**	22.45	38.87	28.46	22.45	38.87	28.46
KL	22.70	45.19	30.22	24.40	42.25	30.93	19.70	34.11	24.98	20.36	38.79	26.70
PE	23.75	41.13	30.11	**24.05**	**43.72**	**31.03**	21.70	37.58	27.51	21.65	37.49	27.45
SL	23.60	40.87	29.92	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	22.91	43.64	30.05	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 20	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	23.14	42.08	29.86	**24.05**	**43.72**	**31.03**	21.45	37.14	27.19	21.45	37.14	27.19
KL	21.48	46.49	29.38	24.40	42.25	30.93	19.55	33.85	24.79	19.04	39.57	25.71
PE	23.65	40.95	29.98	**24.05**	**43.72**	**31.03**	21.85	37.84	27.70	21.90	37.92	27.76
SL	22.22	44.24	29.58	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	22.32	42.51	29.27	NA	NA	NA	NA	NA	NA	NA	NA	NA

linguistic filtering are fed to the input of stage 5 as columns for the formation of the documents-terms matrix. Stages 6 and 7 directly implement the term extraction process. The documents-terms matrix is fed to the input of stage 6, where as a result of NMF two new matrices are obtained: the matrix of bases and the coefficient matrix containing the weights of the terms. The transposed coefficient matrix enters the input of stage 7, where it is straightened into one general list, which is sorted in descending order of weights of terms. Then, from this list, the top N terms are cut off without repetitions, which, according to our hypothesis, form the desired set of domain terms.

4.1. *Annotating the corpus*

To annotate the corpus, we use UDPipe linguistic tool (Straka, 2018). UDPipe is a pipeline with trained language models that performs the following pipeline operations of text processing: segmentation (isolating sentences and paragraphs), tokenization (isolating words and indivisible combinations), lemmatization (bringing words to normal form), POS tagging (defining a grammatical category words or tokens) and sentence analysis. For some languages, UDPipe offers several language models; in this case we choose the best model in terms of its performance (Kim et al., 2007). Table 1 shows the selected models for all three languages represented in the ACTER dataset.

When using the UDPipe tool at the annotation stage, we encountered

Table A5
NMF results on Equi corpus for French at k = 2,3,5,7,9,15,20 (without NEs).

k = 2	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	14.50	30.18	19.59	14.95	31.11	20.20	14.50	30.18	19.59	14.50	30.18	19.59
KL	17.43	41.73	24.59	**18.30**	**38.09**	**24.72**	18.15	37.77	24.52	17.95	37.36	24.25
PE	14.40	29.97	19.45	14.95	31.11	20.20	14.00	29.14	18.91	13.85	28.82	18.71
SL	14.35	29.86	19.38	NA	NA	NA	NA	NA	NA	12.85	26.74	17.36
SR	14.35	29.86	19.38	NA	NA	NA	13.00	27.06	17.56	13.00	27.06	17.56
k = 3	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	14.85	30.91	20.06	14.95	31.11	20.20	14.85	30.91	20.06	14.85	30.91	20.06
KL	**18.00**	**37.46**	**24.32**	18.15	37.77	24.52	17.75	36.94	23.98	16.70	34.76	22.56
PE	14.60	30.39	19.72	14.95	31.11	20.20	14.35	29.86	19.38	14.00	29.14	18.91
SL	14.65	30.49	19.79	NA	NA	NA	NA	NA	NA	NaN	0.00	NaN
SR	14.50	30.18	19.59	NA	NA	NA	14.50	30.18	19.59	14.50	30.18	19.59
k = 5	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	16.25	33.82	21.95	14.95	31.11	20.20	16.25	33.82	21.95	16.25	33.82	21.95
KL	**17.70**	**42.35**	**24.97**	17.59	40.27	24.48	17.80	37.04	24.04	16.55	34.44	22.36
PE	14.00	29.14	18.91	14.95	31.11	20.20	13.86	31.74	19.29	13.45	30.80	18.72
SL	14.90	31.01	20.13	NA	NA	NA	NA	NA	NA	NaN	0.00	NaN
SR	14.15	29.45	19.12	NA	NA	NA	NaN	0.00	NaN	NaN	0.00	NaN
k = 7	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	17.35	36.11	23.44	14.95	31.11	20.20	14.29	31.22	19.61	14.29	31.22	19.61
KL	**18.45**	**38.40**	**24.92**	18.25	37.98	24.65	16.20	33.71	21.88	16.00	33.30	21.61
PE	13.82	31.63	19.24	14.95	31.11	20.20	13.80	28.72	18.64	13.75	28.62	18.58
SL	17.00	35.38	22.97	NA	NA	NA	NA	NA	NA	NaN	0.00	NaN
SR	16.76	36.63	23.00	NA	NA	NA	NaN	0.00	NaN	NaN	0.00	NaN
k = 9	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	16.23	37.15	22.59	14.95	31.11	20.20	16.70	34.76	22.56	16.70	34.76	22.56
KL	**18.95**	**39.44**	**25.60**	17.50	40.06	24.36	17.25	35.90	23.30	16.95	35.28	22.90
PE	14.10	29.34	19.05	14.95	31.11	20.20	12.95	26.95	17.49	12.85	26.74	17.36
SL	17.33	37.88	23.78	NA	NA	NA	NA	NA	NA	NaN	0.00	NaN
SR	15.05	31.32	20.33	NA	NA	NA	NaN	0.00	NaN	NaN	0.00	NaN
k = 15	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	16.90	35.17	22.83	14.95	31.11	20.20	16.9	35.17	22.83	17.10	35.59	23.10
KL	**18.95**	**39.44**	**25.60**	18.00	37.46	24.32	16.3	33.92	22.02	16.30	33.92	22.02
PE	14.10	30.80	19.34	14.95	31.11	20.20	14.1	29.34	19.05	13.75	28.62	18.58
SL	17.50	36.42	23.64	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	16.60	34.55	22.43	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 20	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	16.30	39.02	22.99	14.95	31.11	20.20	16.76	36.63	23.00	16.29	35.59	22.35
KL	17.15	35.69	23.17	**18.00**	**37.46**	**24.32**	16.25	33.82	21.95	16.10	33.51	21.75
PE	14.14	30.91	19.40	14.95	31.11	20.20	14.30	29.76	19.32	13.70	28.51	18.51
SL	17.55	36.52	23.71	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	15.65	32.57	21.14	NA	NA	NA	NA	NA	NA	NA	NA	NA

a problem that the English-language model defines 's and s' endings as independent tokens, and as a result, the bigrams formed with their participation are mistakenly marked as trigrams. To get rid of this problem, we used UDPipe's n-gram decode function, which allows us to append 's and s' endings to the main word (Table 2).

4.2. *Linguistic filtering and document-term matrix construction*

Linguistic filtering narrows down the list of candidate terms to terms that match to certain morphosyntactic patterns. For example, strong linguistic filters can require candidate terms to be only nouns or noun phrases, while soft filters can require to exclude only stop words and functional words. The use of strong linguistic filters increases precision but reduces recall, while the use of soft filters, on opposite, increases recall but reduces precision (Pazienza et al., 2005; Zhang et al., 2018a). In the broad sense, linguistic filtering can be involved at any stage of linguistic processing, including lemmatization, POS-tagging, N-gram extraction, removing stop words etc.

In this study, at the stage of linguistic filtering, we select *n*-grams (*n* = 1,2,3) to form a list of candidate terms. From the list of unigrams, we exclude stop words and service tokens, i.e. tokens with the following POS-tags: DET (articles), NUM (numbers, alphanumeric combinations), PUNCT (punctuation), SYM (symbols), PART (particles), AUX (auxiliary verbs). Similarly, from the list of bigrams, we exclude bigrams containing stop words or service tokens. With regard to trigrams, we act differently: we exclude only trigrams containing stop words in the first

Table A6
NMF results on Equi corpus for Dutch at k = 2,3,5,7,9,15,20 (without NEs).

k = 2	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	24.70	35.46	29.12	25.10	36.04	29.59	24.70	35.46	29.12	24.70	35.46	29.12
KL	25.75	36.97	30.36	26.62	40.13	32.01	**27.75**	**39.84**	**32.71**	27.14	40.92	32.63
PE	25.05	37.76	30.12	25.10	36.04	29.59	25.00	37.69	30.06	25.00	37.69	30.06
SL	24.86	37.47	29.89	NA	NA	NA	NA	NA	NA	24.86	37.47	29.89
SR	24.86	37.47	29.89	NA	NA	NA	24.86	37.47	29.89	24.86	37.47	29.89
k = 3	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	25.45	36.54	30.00	25.1	36.04	29.59	24.20	34.75	28.53	24.20	34.75	28.53
KL	26.60	38.19	31.36	27.2	39.05	32.07	**28.05**	**40.27**	**33.07**	28.00	40.20	33.01
PE	24.59	38.84	30.11	25.1	36.04	29.59	21.19	41.06	27.95	21.19	41.06	27.95
SL	22.39	36.97	27.89	NA	NA	NA	NA	NA	NA	22.39	36.97	27.89
SR	22.35	36.90	27.84	NA	NA	NA	NaN	0.00	NaN	NaN	0.00	NaN
k = 5	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	24.75	35.53	29.18	25.1	36.04	29.59	24.75	35.53	29.18	24.75	35.53	29.18
KL	21.24	44.22	28.70	27.2	39.05	32.07	**27.70**	**39.77**	**32.66**	27.45	39.41	32.36
PE	24.90	35.75	29.35	25.1	36.04	29.59	21.13	34.89	26.32	21.13	34.89	26.32
SL	17.65	46.88	25.64	NA	NA	NA	NA	NA	NA	NaN	0.00	NaN
SR	17.51	44.01	25.05	NA	NA	NA	NaN	0.00	NaN	NaN	0.00	NaN
k = 7	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	24.85	35.68	29.30	25.10	36.04	29.59	24.85	35.68	29.30	24.85	35.68	29.30
KL	21.58	40.27	28.10	27.15	38.98	32.01	**27.85**	**39.99**	**32.83**	27.20	39.05	32.07
PE	24.70	35.46	29.12	25.10	36.04	29.59	21.78	35.97	27.13	21.48	35.46	26.75
SL	21.67	32.66	26.05	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	20.70	29.72	24.40	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 9	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	22.73	35.89	27.83	25.10	36.04	29.59	25.20	36.18	29.71	25.20	36.18	29.71
KL	24.45	35.10	28.82	27.15	38.98	32.01	25.08	45.01	32.21	**27.90**	**40.06**	**32.89**
PE	24.60	35.32	29.00	25.10	36.04	29.59	21.26	35.10	26.48	21.77	34.39	26.66
SL	21.55	30.94	25.41	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	21.55	30.94	25.41	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 15	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	25.35	36.40	29.89	25.10	36.04	29.59	24.85	35.68	29.30	24.85	35.68	29.30
KL	26.75	38.41	31.54	26.67	40.20	32.07	25.25	43.50	31.95	**26.95**	**40.63**	**32.41**
PE	22.20	39.84	28.51	25.10	36.04	29.59	23.00	33.02	27.11	23.30	33.45	27.47
SL	24.10	34.60	28.41	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	22.80	32.74	26.88	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 20	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	25.80	37.04	30.41	25.10	36.04	29.59	25.55	36.68	30.12	25.55	36.68	30.12
KL	27.20	39.05	32.07	26.86	40.49	32.30	24.04	44.87	31.31	**27.60**	**39.63**	**32.54**
PE	24.10	36.32	28.97	25.10	36.04	29.59	23.40	33.60	27.59	23.70	34.03	27.94
SL	25.95	37.26	30.59	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	24.35	34.96	28.71	NA	NA	NA	NA	NA	NA	NA	NA	NA

or third place (this allows keeping trigrams with prepositions in the middle), and exclude trigrams with service tokens anywhere.

As noted above, because of linguistic filtering, we discard a large number of irrelevant words and phrases, but at the same time, we lose some of the domain terms. For example, when working with the Heart failures (HTFL) corpus, after exclusion of terms with NUM POS-tag, we lose such domain terms as cxl-1020, erk1/2, arg16/gln27. This problem is common to all term extraction systems (Ittoo & Bouma, 2013). For this reason the developers of the TALN-LS2N system (Hazem et al., 2020), who won first place in the TermEval 2020 competition, abandoned the stage of linguistic filtering, fearing to lose potentially correct variants. As a result, they increased recall of term extraction, but decreased precision, because false positives appeared in the form of verbose terms starting or ending with service words.

We place all unigrams, bigrams and trigrams that have successfully passed through the linguistic filters, in the columns of documents-terms matrix, and write the names of the corpus documents in the rows. In the cells of the matrix, we place the frequencies of the use of the corresponding terms in the corresponding documents. Based on the number of columns of the resulting matrix n (it is the number of candidates to the domain terms) and the number of true domain terms T, we can calculate the limiting value of the recall of the extraction of terms:

$$Re_{lim} = \frac{T}{n} \tag{18}$$

Obviously, it is impossible to increase this value at subsequent stages,

Table A7
NMF results on Wind corpus for English at k = 2,3 (without NEs).

k = 2	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	17.71	34.10	23.31	17.20	39.41	23.95	17.71	34.10	23.31	17.71	34.10	23.31
KL	17.37	38.22	23.88	18.18	36.66	24.31	**16.63**	**45.74**	**24.39**	16.63	45.74	24.39
PE	18.35	33.64	23.75	17.20	39.41	23.95	18.33	35.29	24.13	18.33	35.29	24.13
SL	17.62	33.91	23.19	NA	NA	NA	NA	NA	NA	17.62	33.91	23.19
SR	17.62	33.91	23.19	NA	NA	NA	17.62	33.91	23.19	17.62	33.91	23.19
k = 3	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	**18.76**	**36.1**	**24.69**	18.75	34.37	24.26	17.50	35.29	23.40	17.50	35.29	23.40
KL	17.09	36.02	23.18	**18.57**	**35.75**	**24.44**	18.10	33.18	23.42	18.10	33.18	23.42
PE	18.15	33.27	23.49	18.75	34.37	24.26	18.65	34.19	24.13	18.65	34.19	24.13
SL	16.71	36.76	22.98	NA	NA	NA	NA	NA	NA	16.71	36.76	22.98
SR	16.58	36.48	22.80	NA	NA	NA	16.58	36.48	22.80	16.58	36.48	22.80

Table A8
NMF results on Wind corpus for French with k = 2 (without NEs).

k = 2	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	10.96	36.87	16.90	10.92	35.32	16.68	10.92	35.32	16.68	10.92	35.32	16.68
KL	12.15	31.44	17.53	11.70	30.27	16.88	11.70	30.27	16.88	11.70	30.27	16.88
PE	11.55	29.88	16.66	10.92	35.32	16.68	10.92	35.32	16.68	10.92	35.32	16.68
SL	**12.10**	**32.86**	**17.69**	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	10.92	35.32	16.68	NA	NA	NA	NA	NA	NA	NA	NA	NA

Table A9
NMF results on the Wind corpus for Dutch at k = 2,3,5 (without NEs).

k = 2	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	12.71	28.40	17.56	9.64	22.55	13.51	12.71	28.40	17.56	12.71	28.40	17.56
KL	**11.28**	**43.19**	**17.89**	11.14	43.83	17.76	10.61	25.96	15.06	10.61	25.96	15.06
PE	11.35	27.77	16.11	9.64	22.55	13.51	10.59	24.79	14.84	10.59	24.79	14.84
SL	12.71	28.40	17.56	NA	NA	NA	NA	NA	NA	12.71	28.40	17.56
SR	12.85	27.34	17.48	NA	NA	NA	12.85	27.34	17.48	12.85	27.34	17.48
k = 3	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	10.71	38.72	16.78	9.64	22.55	13.51	10.71	38.72	16.78	10.71	38.72	16.78
KL	**11.87**	**40.43**	**18.35**	10.71	46.70	17.42	12.04	32.02	17.50	11.92	32.98	17.51
PE	11.55	27.02	16.18	9.64	22.55	13.51	11.52	25.74	15.92	11.52	25.74	15.92
SL	12.48	27.87	17.24	NA	NA	NA	NA	NA	NA	12.48	27.87	17.24
SR	12.57	28.09	17.37	NA	NA	NA	12.57	28.09	17.37	12.57	28.09	17.37
k = 5	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	12.11	34.79	17.97	9.64	22.55	13.51	12.11	34.79	17.97	12.11	34.79	17.97
KL	**12.37**	**39.47**	**18.84**	10.97	44.36	17.59	11.54	34.36	17.28	11.54	34.36	17.28
PE	11.70	28.62	16.61	9.64	22.55	13.51	11.70	28.62	16.61	11.50	29.36	16.53
SL	11.86	35.32	17.76	NA	NA	NA	NA	NA	NA	11.86	35.32	17.76
SR	11.93	35.53	17.86	NA	NA	NA	11.93	35.53	17.86	11.93	35.53	17.86

since the denominator can only be decreased, and the numerator is fixed.

4.3. Non-negative matrix factorization and term extraction

In this research, we use the following algorithms for non-negative matrix factorization, which we denote as FR, KL, PE, SL, and SR, respectively:

1) FR: NMF algorithm from Lee (Lee & Seung, 2000) that minimizes the Frobenius norm and uses the stationarity of the objective value as a stopping criterion;
2) KL: NMF algorithm from Brunet (Brunet et al., 2004) that minimizes the Kullback-Leibler divergence and uses the stationarity of the objective value as a stopping criterion;
3) PE: Pattern-Expression NMF algorithm (Zhang et al., 2008) that minimizes an Euclidean-based objective function which is regularized for effective expression of patterns with basis vectors. We use this algorithm with parameters $\alpha = 0, \beta = 1$ (see formulas (12) and (13)).
4) SL: Alternating Least Square NMF algorithm from Kim et al. (Kim et al., 2007). that minimizes an Euclidean-based objective function, which is regularized to favor sparse basis matrices;

Table A10
NMF results on HTFL corpus for English at k = 2,3,5,7,9,15,20 (without NEs).

k = 2	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	23.20	39.31	29.18	22.88	40.70	29.29	23.20	39.31	29.18	23.20	39.31	29.18
KL	20.03	51.76	28.88	27.67	38.67	32.26	**28.03**	**41.55**	**33.48**	27.44	41.85	33.15
PE	23.41	38.67	29.16	22.88	40.70	29.29	23.25	39.39	29.24	23.25	39.39	29.24
SL	23.28	39.43	29.28	NA	NA	NA	NA	NA	NA	23.28	39.43	29.28
SR	23.25	39.39	29.24	NA	NA	NA	23.25	39.39	29.24	23.25	39.39	29.24
k = 3	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	26.81	30.66	28.61	22.88	40.70	29.29	26.81	30.66	28.61	26.81	30.66	28.61
KL	30.43	29.65	30.03	24.24	47.23	32.04	29.32	38.50	33.29	**27.64**	**42.14**	**33.38**
PE	23.13	38.20	28.81	22.88	40.70	29.29	21.24	41.38	28.07	21.24	41.38	28.07
SL	21.36	40.70	28.02	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	24.78	33.59	28.52	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 5	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	27.04	29.78	28.34	22.88	40.70	29.29	27.04	29.78	28.34	27.04	29.78	28.34
KL	23.18	44.18	30.41	24.95	44.39	31.94	**28.06**	**41.59**	**33.51**	27.94	41.42	33.37
PE	21.31	37.91	27.28	22.88	40.70	29.29	23.03	32.19	26.85	23.03	32.19	26.85
SL	23.43	27.78	25.42	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	18.88	39.18	25.48	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 7	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	26.56	30.37	28.34	22.88	40.7	29.29	26.56	30.37	28.34	26.81	30.66	28.61
KL	24.90	42.19	31.32	28.57	36.3	31.97	**29.52**	**38.75**	**33.51**	29.48	38.71	33.47
PE	21.17	37.65	27.10	22.88	40.7	29.29	23.93	28.38	25.97	20.50	34.73	25.78
SL	20.67	34.14	25.75	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	17.47	44.39	25.07	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 9	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	26.56	30.37	28.34	22.88	40.7	29.29	25.89	30.71	28.09	25.89	30.71	28.09
KL	24.14	42.95	30.91	26.16	42.1	32.27	26.57	41.63	32.44	**26.65**	**41.76**	**32.54**
PE	19.52	38.03	25.80	22.88	40.7	29.29	19.28	37.57	25.48	20.00	35.58	25.61
SL	20.07	34.86	25.47	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	20.78	31.68	25.10	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 15	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	26.36	27.91	27.11	22.88	40.70	29.29	27.54	28.00	27.77	26.04	28.67	27.29
KL	24.79	44.09	31.74	25.37	44.05	32.20	27.18	37.99	31.69	**26.50**	**42.65**	**32.69**
PE	18.77	38.16	25.16	22.88	40.70	29.29	17.84	41.55	24.96	18.08	39.05	24.72
SL	18.50	36.04	24.45	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	16.63	43.67	24.09	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 20	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	25.96	29.69	27.70	22.88	40.70	29.29	24.57	29.14	26.66	25.22	28.84	26.91
KL	24.83	43.12	31.51	24.29	46.29	31.86	**29.06**	**38.16**	**32.99**	31.62	32.15	31.88
PE	17.85	39.31	24.55	22.88	40.70	29.29	17.00	39.60	23.79	18.26	35.58	24.13
SL	16.25	41.30	23.32	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	16.59	39.35	23.34	NA	NA	NA	NA	NA	NA	NA	NA	NA

5) SR: Alternating Least Square NMF algorithm from Kim et al. (Kim et al., 2007) that minimizes an Euclidean-based objective function, which is regularized to favor sparse coefficient matrices.

The choice of these algorithms is due, first, to the availability of their implementation in R, as well as their performance parameters - speed and low resource consumption. The same is true for NMF initialization algorithms. In addition to initialization with random values, which we do not include in our evaluation, we use four more initialization options, which are denoted as FV, SPA, nndSVD, and FCM, respectively:

1) FV: initialization with fixed values, we use the initialization of matrices W and H with matrices consisting of 1;
2) SPA: initialization based on SPA algorithm (Sauwen et al., 2017);
3) nndSVD: initialization based on nndSVD algorithm (Qiao, 2015);
4) FCM: initialization based on Fuzzy C-means algorithm (Yueyang & Shafai, 2018).

Thus, the design of our study includes 20 NMF applications (Table 3).

As noted above, NMF results in two matrices, but we use only the matrix H. We concatenate all the columns of the transposed matrix H^T into one long list and sort it in descending order of coefficients (weights of terms in topics). Since each term appears in this list k times by the number of topics, we leave only one value corresponding to the maximum weight of the term. Then we extract the top N most significant terms from this list, which, according to our hypothesis, form the desired

Table A11
NMF results on HTFL corpus for French at k = 2,3,5,7,9,15,20 (without NEs).

k = 2	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	21.65	30.12	25.19	21.61	32.00	25.80	21.65	30.12	25.19	21.65	30.12	25.19
KL	23.15	41.56	29.74	22.59	44.61	29.99	26.45	36.80	30.78	**26.55**	**36.94**	**30.89**
PE	22.14	28.82	25.04	21.61	32.00	25.80	22.10	29.76	25.36	22.10	29.76	25.36
SL	17.93	35.41	23.81	NA	NA	NA	NA	NA	NA	17.93	35.41	23.81
SR	21.25	26.71	23.67	NA	NA	NA	21.25	26.71	23.67	21.25	26.71	23.67
k = 3	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	22.66	29.49	25.63	21.61	32.0	25.80	22.66	29.49	25.63	22.66	29.49	25.63
KL	23.81	39.54	29.72	22.64	44.7	30.06	**23.27**	**42.82**	**30.15**	23.84	40.66	30.06
PE	22.95	22.67	22.81	21.61	32.0	25.80	21.56	26.12	23.62	21.56	26.12	23.62
SL	22.47	30.25	25.79	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	23.74	28.77	26.01	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 5	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	21.17	27.56	23.95	21.61	32.00	25.80	21.21	27.60	23.99	21.21	27.60	23.99
KL	24.14	39.00	29.82	**23.50**	**42.19**	**30.19**	25.71	35.77	29.92	24.66	38.73	30.13
PE	18.53	29.94	22.89	21.61	32.00	25.80	22.62	24.37	23.46	22.54	24.28	23.38
SL	17.67	30.92	22.49	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	18.47	31.51	23.29	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 7	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	21.14	27.51	23.91	21.61	32.00	25.80	20.81	28.95	24.21	20.81	28.95	24.21
KL	25.97	34.96	29.80	22.63	43.67	29.81	**28.11**	**34.07**	**30.80**	27.81	33.71	30.48
PE	17.90	32.94	23.20	21.61	32.00	25.80	18.94	29.76	23.15	17.63	30.07	22.23
SL	16.13	34.02	21.88	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	16.04	35.28	22.05	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 9	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	20.15	30.75	24.35	21.61	32.0	25.80	19.30	32.05	24.09	19.30	32.05	24.09
KL	25.03	35.95	29.51	22.64	44.7	30.06	27.37	33.17	29.99	**27.44**	**33.26**	**30.07**
PE	19.29	29.44	23.31	21.61	32.0	25.80	19.77	27.51	23.01	20.16	28.95	23.77
SL	15.60	38.51	22.20	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	17.80	27.96	21.75	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 15	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	17.85	32.85	23.13	21.61	32.00	25.80	16.65	35.86	22.74	16.96	35.01	22.85
KL	26.48	32.09	29.02	**22.59**	**44.61**	**29.99**	25.97	33.80	29.37	25.90	33.71	29.29
PE	16.22	35.68	22.30	21.61	32.00	25.80	17.12	32.27	22.37	17.61	30.03	22.20
SL	14.81	45.87	22.39	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	14.65	41.43	21.65	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 20	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	16.44	38.38	23.02	21.61	32.00	25.80	22.42	24.15	23.25	21.75	23.43	22.56
KL	28.12	31.55	29.74	**23.05**	**43.45**	**30.12**	27.60	30.97	29.19	27.32	30.66	28.89
PE	14.02	50.99	21.99	21.61	32.00	25.80	16.74	32.32	22.06	16.43	33.93	22.14
SL	16.17	33.39	21.79	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	13.68	49.73	21.46	NA	NA	NA	NA	NA	NA	NA	NA	NA

set of domain terms.

5. Experimental evaluation

In this section, we evaluate the performance of our approach. Section 5.1 presents the ACTER Dataset corpora used in the experiments. Section 5.2 describes the experiments using proposed NMF approach. Section 5.3 compares the performance of our approach with the performance of algorithms that competed in the TermEval 2020, as well as the performance of four keyword extraction methods adapted to term extraction.

5.1. *ACTER Dataset*

The ACTER Dataset consists of twelve corpora and covers four domains and three languages (Fig. 3). A detailed description of the structure and content of the ACTER Dataset corpora is presented in (Terryn et al., 2020). In this work, we conduct experiments on all twelve ACTER Dataset corpora. We only use the annotated parts of these corpora, i.e., documents for which annotators have already specified the domain terms they contain. This allows us to match the terms extracted using our approach with the reference domain terms manually selected by the compilers of ACTER. It should be noted that the compilers of ACTER for each corpus formed two lists of reference terms: an extended list - with named entities (NEs), and a regular list - without NEs.

Table A12
NMF results on HTFL corpus for Dutch at k = 2,3,5,7,9,15,20 (without NEs).

k = 2	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	14.53	43.44	21.78	15.93	42.24	23.14	14.53	43.44	21.78	14.53	43.44	21.78
KL	25.31	31.73	28.16	**22.68**	**44.84**	**30.12**	15.09	61.86	24.26	15.24	61.72	24.44
PE	14.53	43.44	21.78	15.93	42.24	23.14	14.53	43.44	21.78	14.53	43.44	21.78
SL	14.76	44.12	22.12	NA	NA	NA	NA	NA	NA	14.76	44.12	22.12
SR	14.56	43.54	21.82	NA	NA	NA	14.56	43.54	21.82	14.56	43.54	21.82
k = 3	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	19.40	28.06	22.94	15.93	42.24	23.14	19.40	28.06	22.94	19.40	28.06	22.94
KL	17.77	48.84	26.06	**23.53**	**40.84**	**29.86**	23.16	41.32	29.68	23.77	40.12	29.85
PE	13.91	45.61	21.32	15.93	42.24	23.14	13.77	46.48	21.25	13.77	46.48	21.25
SL	14.31	48.99	22.15	NA	NA	NA	NA	NA	NA	14.31	48.99	22.15
SR	14.23	48.70	22.02	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 5	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	17.30	33.37	22.79	15.93	42.24	23.14	17.22	33.22	22.68	17.22	33.22	22.68
KL	17.53	49.04	25.83	**22.68**	**43.73**	**29.87**	17.71	50.39	26.21	17.61	50.10	26.06
PE	13.48	52.65	21.46	15.93	42.24	23.14	13.37	52.22	21.29	13.41	52.36	21.35
SL	20.54	25.75	22.85	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	20.76	25.02	22.69	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 7	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	18.88	29.12	22.91	15.93	42.24	23.14	18.97	29.27	23.02	18.97	29.27	23.02
KL	26.48	26.81	26.64	**22.87**	**43.01**	**29.86**	17.52	50.68	26.04	17.59	49.18	25.91
PE	13.77	47.16	21.32	15.93	42.24	23.14	16.25	31.34	21.40	14.73	39.05	21.39
SL	19.47	28.16	23.02	NA	NA	NA	NA	NA	NA	NaN	0.00	NaN
SR	19.67	25.60	22.25	NA	NA	NA	NaN	0.00	NaN	NaN	0.00	NaN
k = 9	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	15.26	36.79	21.57	15.93	42.24	23.14	19.86	26.81	22.82	19.59	27.39	22.84
KL	27.15	26.18	26.66	**23.05**	**43.35**	**30.10**	26.43	29.32	27.80	26.00	27.58	26.77
PE	16.20	31.24	21.34	15.93	42.24	23.14	16.00	30.86	21.07	16.00	30.86	21.07
SL	22.35	21.55	21.94	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	16.47	28.59	20.90	NA	NA	NA	NA	NA	NA	NA	NA	NA
k = 15	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	21.65	24.01	22.77	15.93	42.24	23.14	21.16	25.51	23.13	21.16	25.51	23.13
KL	20.23	38.04	26.41	**21.91**	**45.42**	**29.56**	27.05	27.39	27.22	21.38	38.14	27.40
PE	16.49	32.59	21.89	15.93	42.24	23.14	16.15	31.92	21.45	16.34	32.3	21.70
SL	14.33	53.91	22.64	NA	NA	NA	NA	NA	NA	NA	0	NA
SR	16.49	35.78	22.58	NA	NA	NA	NA	0	NA	NA	0	NA
k = 20	nndSVD			FV			SPA			FCM		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
FR	21.04	28.40	24.17	15.93	42.24	23.14	22.91	25.41	24.09	22.16	26.71	24.22
KL	23.79	33.27	27.74	**22.32**	**44.12**	**29.64**	24.21	32.69	27.82	26.33	30.47	28.25
PE	16.63	32.88	22.09	15.93	42.24	23.14	22.14	23.48	22.79	22.09	23.43	22.74
SL	15.42	39.39	22.16	NA	NA	NA	NA	NA	NA	NA	NA	NA
SR	14.98	41.9	22,07	NA	NA	NA	NA	NA	NA	NA	NA	NA

The distribution of annotated documents in the corpus is shown in Fig. 4. The dimensions of the document-term matrices formed for all corpora are presented in Table 4. It also lists the following for each corpus: 1) the number of reference (gold) domain terms, manually extracted by annotators, with and without NEs; 2) the number of term candidates that match the reference terms (without NEs); 3) the limiting recall of term extraction, which is equal to the ratio of the number of domain terms (without named entities) to the number of term candidates. The Dutch Wind corpus has the highest limiting recall of 95.21%, i.e., less than 5% of true domain terms were lost during preprocessing of this corpus.

5.2. Evaluating the performance of the proposed NMF-based approach

We evaluated our NMF-based approach on all twelve corpora of the ACTER Dataset, using 20 different combinations of NMFs and varying the rank of factorization k. The total number of NMF runs was 1260.

We changed the k values based on the requirement $k \ll \min(m, n)$, where m is the number of rows (documents), and n is the number of columns (terms) in the documents-terms matrix. In particular, for all three corpora of the Corp domain ($m = 12$), we used $k = 2,3,5,7,9$. For the Wind domain corpus in English ($m = 5$), we used $k = 2,3$; for the French corpus of the Wind domain ($m = 2$), we used the only possible $k = 2$; for the Dutch corpus of Wind domain ($m = 8$), we used $k = 2,3,5$. For all six corpora of Equi and HTFL domains, for which m greater than

20, we used $k = 2,3,5,7,9,15,20$.

We evaluated the results of NMF runs using the metrics of precision, recall and F1-measure, calculated in relation to the reference domain terms without named entities (Tables A.1–A.12 in the Appendix). We marked in bold the values corresponding to the highest F1-measure obtained on this corpus for a given *k*, and in bold and underlined - the values corresponding to the highest F1-measure obtained on this corpus among all *k* values. The NA values opposite to the corresponding algorithms meant that these algorithms are inapplicable on a given corpus for a given *k*.

We combined the best results for all twelve corpora from Tables A.1 to A.12 into Table 5. From the analysis of this table, it follows that the algorithm with FR code turned out to be the most efficient on the Corp domain corpus, which we denote the Lee algorithm minimizing the Frobenius norm and using the stationarity of the objective value as a stopping criterion. On the rest of the corpora, except for the Wind domain corpus in French, the algorithm with KL code turned out to be the most efficient, which we denote the Brunet algorithm minimizing the Kullback-Leibler distance and using the stationarity of the objective value as a stopping criterion. As for the initialization algorithms working in conjunction with NMF algorithms, nndSVD and SPA algorithms turned out to be the most effective. The fact that the simplest initialization algorithm FV was able to compete with such complex initialization algorithms as nndSVD and SPA deserves a separate study.

Since the teams that participated in the TermEval 2020 used lists of reference terms both without and with named entities to evaluate the performance of their approaches, we also recalculated the precision, recall and F1-measure of our best NMF runs relative to extended lists with the inclusion of named entities (Table 6).

The differences in performance indicators (from 17.69% on the French Wind corpus to 33.51% on the English HTFL corpus in Table 5 and from 18.37% on the French Wind corpus to 33.71% on the English HTFL corpus in Table 6) were explained by differences in the structures of considered corpora. To confirm this assertion, we approximated for each corpus the distribution of mean frequencies of candidate terms in corpus documents to the normal distribution, with the help of Maximum Spacing Estimation (MSE). We plotted the fitted distributions and calculated the log-likelihood values (LH) as shown in Fig. 5. Comparing the obtained plots with the NMF performance results from Table 5, we found that NMF demonstrated the low performance on imbalanced corpora whose structure is poorly fitted to the normal distribution (which corresponds to negative values of LH). These imbalanced corpora have a low peak value of the density function (below the red line on the plot).

KL-NMF algorithm turned out to be the undoubted leader in our experiments, with the exception of the runs on English, French and Dutch corpora of the Corp domain and the runs on English and French corpora of the Wind domain. FR-NMF algorithm showed the best performance on all three corpora of the Corp domain and on the English corpus of the Wind domain. In related studies on evaluating the performance of NMF applied to texts, we found the only comparison of algorithms based on the Kullback-Leibler and Frobenius norms (Svensson & Blad, 2020) . In this study, it is noted that NMF algorithms based on the Kullback-Leibler norm “are characterized by poorer coherence scores compared to the Frobenius norm but achieved significantly better scores in assigning specific topics to documents”. Armed with this fact, we estimated how close documents are to each other in all corpora of ACTER Dataset. We estimated the pairwise distances between corpus documents in the Euclidean space of the candidate terms, and then built histograms of densities, conventionally dividing all estimated distances into three groups: close, medium, and far. It turned out that in all three corpora of the Corp domain and in the English and French corpora of the Wind domain, the proportion of far distances between documents is more than 40%, while in the rest of the corpora close or medium distances prevail (see Fig. 6). Thus, we can conclude that KL-NMF algorithm performs better when the documents are semantically close.

5.3. Benchmarking the NMF-based approach against baseline methods

Although at TermEval the results of different approaches were compared only on the HTFL test corpora, and the Corp, Equi and Wind corpora were considered training ones, one of the participating teams evaluated the performance of four baseline keyword extraction methods (TF-IDF, RAKE, YAKE and TextRank) adapted for term extraction, in all ACTER corpora in English (Pais & Ion, 2020). The results of a comparison of best results of NMF-based approach with results of these baseline methods are shown in Table 7. NMF-based approach has surpassed all the methods indicated.

Finally, we compared the best results of NMF-based approach with the performance of the methods competed in TermEval 2020 (Table 8). The final of TermEval 2020 was attended by five teams (TALN-LS2N, RACAI, NYU, e-Termhood and NLPLab_UQAM) (Terryn et al., 2020). All of them offered solutions for HTFL test corpus in English, three of them also offered solutions for HTFL test corpus in French, and only two teams offered solutions for HTFL test corpus in Dutch. NMF-based approach was ranked first on the Dutch HTFL corpus, second on the French HTFL corpus and third on the English HTFL corpus. We consider this a promising result since the methods to which our approach yielded used sophisticated machine learning technologies. Our approach does not require training data and is easily adaptable to changing languages and domains. On the other hand, we cannot argue that our approach has completely abandoned the results of machine learning, since the UDPipe tool that we used for part-of-speech tagging of terms uses pre-trained models.

6. Conclusion and future work

In this work, we joined, albeit significantly later and offline, the community of researchers who participated in TermEval 2020 Shared Task competition. The goal of the competition was to evaluate reliably the performance of various automatic term extraction algorithms. We applied NMF-based topic modeling to unsupervised term extraction. We found the optimal parameters of the applied approach, which provide the performance, superior or comparable to the performance of baseline methods and methods competed in TermEval 2020.

The key contributions of our work are summarized as follows:

- We compared five different NMF algorithms with each other and showed that the modified Brunet algorithm was the most efficient. The essence of the modification was that not the stationarity of the connectivity matrix, but the stationarity of the objective value was used as the stopping criterion.
- We considered four initialization methods and showed that the most efficient in our task are initialization based on nndSVD and SPA algorithms.
- The found optimal combinations of NMF outperform four baseline keyword extraction methods (TF-IDF, Rake, Yake and TextRank) and three methods competed in the TermEval 2020.

Although our approach has not outperformed two methods competed in TermEval 2020 (TALN-LS2N and RACAI) we believe that NMF-based term extraction has its own niche. While methods such as TALN-LS2N and RACAI, which use deep learning, take time and resources to fit, NMF can extract terms from scratch.

Our future work will be in a deeper study of the applicability of various NMF algorithms, as well as how they are initialized to the task of automatic term extraction. We plan to answer the question whether it is possible to improve the obtained performance estimates, or this is the ceiling of the NMF-based approaches. We also plan to investigate in more detail how the structure of the corpus affects the performance of NMF algorithms, since this time we used a general assumption that the quality of term extraction decreases with a decrease in the number of documents in the corpus, regardless of how voluminous these documents are in general.

In conclusion, when we look ahead it is worthwhile to note that NMF-based probabilistic topic modeling is increasingly important in the context of the emergence of novel studies on deep matrix factorizations (De Handschutter et al., 2021; Wang & Zhang, 2021). We are impressed with the performance of the novel models combining "the ability to extract hierarchical features as deep learning models, with a high interpretability power, as low-rank matrix approximations" (De Handschutter et al., 2021), and we are motivated to continue our research on unsupervised automatic term extraction using novel deep NMF models.

CRediT authorship contribution statement

Aliya Nugumanova: Conceptualization, Methodology, Software. **Darkhan Akhmed-Zaki:** Supervision. **Madina Mansurova:** Writing – review & editing. **Yerzhan Baiburin:** Data curation, Validation, Writing – original draft. **Almasbek Maulit:** Visualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This paper has been supported by Grant of the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan AP09561483.

Appendix A

References

Abuzayed, A., & Al-Khalifa, H. (2021). BERT for arabic topic modeling: An experimental study on BERTopic technique. –. *Procedia Computer Science, 189*, 191–194.

Ahmad K. et al. (1999). University of Surrey Participation in TREC8: Weirdness Indexing for Logical Document Extrapolation and Retrieval (WILDER) Retrieved from https://dblp.uni-trier.de/rec/conf/trec/AhmadGT99.html?view=bibtex Accessed May 2, 2021.

Amjadian, E., et al. (2016). Local-global vectors to improve unigram termhood extraction. In *Proceedings of the 5th International Workshop on Computational Termhood* (pp. 2–11).

Amjadian, E., et al. (2018). Distributed specificity for automatic termhood extraction. Termhood. *International Journal of Theoretical and Applied Issues in Specialized Communication, 24*(1), 23–40. https://doi.org/10.1075/term.00012.amj

Ang, A. M. S., & Gillis, N. (2019). Algorithms and comparisons of nonnegative matrix factorizations with volume regularization for hyperspectral unmixing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 12*(12), 4843–4853.

Astrakhantsev, N. A., et al. (2015). Methods for automatic term recognition in domain-specific text collections. A survey. *Programming and Computer Software, 41*(6), 336–349. https://doi.org/10.1134/S036176881506002X

Astrakhantsev N. (2014). Automatic term acquisition from domain-specific text collection by using Wikipedia. *Proceedings of the institute for system programming*, 26 (4), 7-20. 10.15514/ISPRAS-2014-26(4)-1.

Basili, A. et al. (2001). A contrastive approach to term extraction. *Proceedings of the 4th Conference on Terminology and Artificial Intelligence* (TIA–2001), Nancy.

Bolshakova, E., et al. (2013). Topic models can improve domain term extraction. In *Advances in Information Retrieval* (pp. 684–687). Springer.

Boudin, F. (2016). Pke: An open source python-based keyphrase extraction toolkit. In *In Proceedings of COLING 2016, the 26th international conference on computational linguistics: System demonstrations* (pp. 69–73).

Bougouin A., et al. (2013) TopicRank: Graph-based topic ranking for keyphrase extraction. *Proceedings of the Sixth International Joint Conference on Natural Language Processing.* Asian Federation of Natural Language Processing. 543–551.

Boutsidis, C., & Gallopoulos, E. (2008). SVD based initialization: A head start for nonnegative matrix factorization. *Pattern recognition, 41*(4), 1350–1362. https://doi.org/10.1016/j.patcog.2007.09.010

Brunet, J.-P., et al. (2004). Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences of the United States of America,* PNAS, 101, 4164–4169. 10.1073/pnas.0308531101.

Drouin, P., et al. (2020). Automatic term extraction from newspaper corpora. Making the most of specificity and common features. In *Proceedings of the 6th International Workshop on Computational Termhood* (pp. 1–7).

De Handschutter, P., Gillis, N., & Siebert, X. (2021). A survey on deep matrix factorizations. *Computer Science Review, 42*, Article 100423.

Hazem A. et al. (2020). TermEval2020: Taln-ls2n system for automatic term extraction. *Proceedings of the 6th International Workshop on Computational Termhood*, – European Language Resources Association (ELRA), 95-100.

Hätty, A., and im Walde, S. S. (2018). Fine-grained termhood prediction for german compound terms using neural networks. In Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018) (pp. 62-73).

Ittoo, A., & Bouma, G. (2013). Term extraction from sparse, ungrammatical domain-specific documents. *Expert Systems with Applications, 40*(7), 2530–2540. https://doi.org/10.1016/j.eswa.2012.10.067

Kageura, K. and Marshman, E. (2019). Terminology Extraction and Management. In O'Hagan, Minako, editor, The Routledge Handbook of Translation and Technology.

Kim, D., et al. (2007). Fast Newton-type methods for the least squares nonnegative matrix approximation problem. In *Proceedings of the SIAM international conference on data mining. – Society for Industrial and Applied Mathematics* (pp. 343–354). https://doi.org/10.1137/1.9781611972771.31

Kim, S. N., et al. (2009). An unsupervised approach to domain-specific term extraction. In *Proceedings of the Australasian Language Technology Association Workshop* (pp. 94–98).

Korkontzelos, I., et al. (2008). In *Reviewing and evaluating automatic term recognition techniques* (pp. 248–259). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-85287-2_24.

Lang C. et al. (2021). Transforming Term Extraction: Transformer-Based Approaches to Multilingual Term Extraction Across Domains. *Findings of the Association for Computational Linguistics. – ACL-IJCNLP-2021*, 3607-3620.

Langville A. N. et al. (2014). Algorithms, initializations, and convergence for the nonnegative matrix factorization. Retrieved from https://arxiv.org/abs/1407.7299v1 Accessed May 2, 2021.

Lee D. D., Seung H. S., (2000). Algorithms for non-negative matrix factorization. *Proceedings of the Neural Information Processing Systems (NIPS),* Denver, CO, USA, 556–562.

Liu, Z., et al. (2010). In *Automatic keyphrase extraction via topic decomposition* (pp. 366–376). Association for Computational Linguistics.

Li, S., et al. (2013). A novel topic model for automatic term extraction. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval* (pp. 885–888). https://doi.org/10.1145/2484028.2484106

Lopes, L., et al. (2016). Estimating term domain relevance through term frequency, disjoint corpora frequency-tf-dcf. *Knowledge-Based Systems, 97*, 237–249. https://doi.org/10.1016/j.knosys.2015.12.015

Lossio-Ventura J.A., et al. (2013). Combining C-value and keyword extraction methods for biomedical terms extraction. In LBM: Languages in Biologyand Medicine, Tokyo, Japan.

Matsuo, Y., & Ishizuka, M. (2004). Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools, 13*(01), 157–169. https://doi.org/10.1142/S0218213004001466

Meijer, K., Frasincar, F., & Hogenboom, F. (2014). A semantic approach for extracting domain taxonomies from text. *Decision Support Systems, 62*, 78–93. https://doi.org/10.1016/j.dss.2014.03.006

Mirzal, A. (2014). Nonparametric Tikhonov regularized NMF and its application in cancer clustering. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 11*(6), 1208–1217. https://doi.org/10.1109/tcbb.2014.2328342

Mykowiecka, A., et al. (2018). Recognition of irrelevant phrases in automatically extracted lists of domain terms. Termhood. *International Journal of Theoretical and Applied Issues in Specialized Communication, 24*(1), 66–90. https://doi.org/10.1075/term.00014.myk

Nenadic, G., Ananiadou, S., & McNaught, J. (2004). Enhancing automatic term recognition through recognition of variation. In *In COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics* (pp. 604–610).

Nakagawa, H., & Mori, T. (2003). Automatic term recognition based on statistics of compound nouns and their components. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication, 9*(2), 201–219.

Nokel, M., et al. (2012). Combining multiple features for single-word term extraction. *Proceedings of Dialog*, 490–501.

Nugumanova, A., et al. (2016). In *A new operationalization of contrastive term extraction approach based on recognition of both representative and specific terms* (pp. 103–118). Cham: Springer.

Oliver A., Vàzquez M. (2020). TermEval2020: Using TSR Filtering Method to Improve Automatic Term Extraction. *Proceedings of the 6th International Workshop on Computational Termhood*, – European Language Resources Association (ELRA),106-113.

Pais V., Ion R. (2020). TermEval2020: Racai's automatic term extraction system. *Proceedings of the 6th International Workshop on Computational Termhood*, – European Language Resources Association (ELRA), 101-105.

Pazienza M. T et al. (2005). Pazienza M. T., Pennacchiotti M., Zanzotto F. M. Terminology extraction: an analysis of linguistic and statistical approaches //Knowledge mining. – Springer, Berlin, Heidelberg, 2005. – C. 255-279.

Qiao H. (2015). New SVD based initialization strategy for non-negative matrix factorization *//Pattern Recognition Letters*, 63, 71-77. 10.1016/j.patrec.2015.05.019.

Rezaei, M., et al. (2011). An efficient initialization method for nonnegative matrix factorization. *Journal of Applied Sciences, 11*(2), 354–359. https://doi.org/10.3923/jas.2011.354.359

Repar, A., et al. (2019). TermEnsembler: An ensemble learning approach to bilingual term extraction and alignment. Termhood. *International Journal of Theoretical and Applied Issues in Specialized Communication, 25*(1), 93–120. https://doi.org/10.1075/term.00029.rep

Šandrih, B., et al. (2020). Two approaches to compilation of bilingual multi-word termhood lists from lexical resources. *Natural Language Engineering, 26*(4), JULY. https://doi.org/10.1017/S1351324919000615

Sauwen N. et al. (2017). The successive projection algorithm as an initialization method for brain tumor segmentation using non-negative matrix factorization. *PLOS ONE*, 12 (8), Article e0180268. 10.1371/journal.pone.0180268.

Sclano, F., & Velardi, P. (2007). Termextractor. A web application to learn the shared termhood of emergent web communities. In *Enterprise Interoperability II* (pp. 287–290). London: Springer. https://doi.org/10.1007/978-1-84628-858-6_32.

Straka M. (2018) UDPipe 2.0 prototype at CoNLL UD Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. *Proceedings of the CoNLL*, 197-207. 10.18653/v1/K18-2020.

Sterckx, L., et al. (2015). Topical word importance for fast keyphrase extraction. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 121–122). https://doi.org/10.1145/2740908.2742730

Süzek, T.Ö. (2017). Using latent semantic analysis for automated keyword extraction from large document corpora. *Turkish Journal of Electrical Engineering & Computer Sciences, 25*(3), 1784–1794. https://doi.org/10.3906/ELK-1511-203

Svensson K., Blad J. Exploring NMF and LDA Topic Models of Swedish News Articles [Internet] [Dissertation]. 2020. (Accessed 2 January 2022).

Teneva, N., and Cheng, W. (2017). Salience rank: Efficient keyphrase extraction with topic modeling. *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2, 530-535.

Terryn, A. R., et al. (2019). Analysing the Impact of Supervised Machine Learning on Automatic Term Extraction. HAMLET vs TermoStat. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing* (pp. 1012–1021). https://doi.org/10.26615/978-954-452-056-4_117

Terryn, A. R., et al. (2020). In *TermEval2020: Shared task on automatic term extraction using the annotated corpora for term extraction research (ACTER) dataset* (pp. 85–94). European Language Resources Association (ELRA).

Vavasis, S. (2009). On the complexity of nonnegative matrix factorization. *SIAM J. Optimization, 20*, 1364–1377. https://doi.org/10.1137/070709967

Vivaldi, J., & Rodríguez, H. (2007). Evaluation of terms and term extraction systems: A practical approach. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication, 13*(2), 225–248.

Wang J.Y., Zhang X.L., (2021). Deep NMF Topic Modeling. – arXiv preprint arXiv: 2102.12998.

Wild, S. (2003). *Seeding non-negative matrix factorizations with spherical k-means clustering.* University of Colorado. Master's thesis.

Xu, W., Liu, X., & Gong, Y. (2003). Document clustering based on non-negative matrix factorization. In *In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval* (pp. 267–273).

Yueyang, W., & Shafai, B. (2018). *New initialization strategy for nonnegative matrix factorization.* Diss: Northeastern University Boston.

Zhang, J., et al. (2008). Pattern expression nonnegative matrix factorization: Algorithm and applications to blind source separation. *Computational intelligence and neuroscience. Article ID*, 168769 | 10.1155/2008/168769.

Zhang, Z., et al. (2018a). SemRe-rank: Improving automatic term extraction by incorporating semantic relatedness with personalised PageRank. *ACM Transactions on Knowledge Discovery from Data (TKDD), 12*(5), 1–41. https://doi.org/10.1145/3201408

Zhang Z., et al. (2018b). Adapted TextRank for Term Extraction: A Generic Method of Improving Automatic Term Extraction Algorithms. SEMANTiCS 2018 – *14th International Conference on Semantic Systems. Procedia Computer Science*, 137, 102-108. 10.1016/j.procs.2018.09.010.

Zhang Z., et al. (2016). JATE2: Java Automatic Term Extraction with Apache Solr. *Tenth International Conference on Language Resources and Evaluation(LREC)*. - European Language Resources Association (ELRA), 2262-2269.