An Improved Method for Class-specific Keyword Extraction: A Case Study in the German Business Registry

Stephen Meisenbacher[♠1], **Tim Schopf**[♠1], **Weixin Yan**[1], **Patrick Holl**[2], **and Florian Matthes**[1]
[1]Technical University of Munich
School of Computation, Information and Technology
Department of Computer Science, Garching, Germany
[2]Fusionbase GmbH, Munich, Germany
{first.last}@tum.de, patrick.holl@fusionbase.com, matthes@tum.de

Abstract

The task of *keyword extraction* is often an important initial step in unsupervised information extraction, forming the basis for tasks such as topic modeling or document classification. While recent methods have proven to be quite effective in the extraction of keywords, the identification of *class-specific* keywords, or only those pertaining to a predefined class, remains challenging. In this work, we propose an improved method for class-specific keyword extraction, which builds upon the popular KeyBERT library to identify only keywords related to a class described by *seed keywords*. We test this method using a dataset of German business registry entries, where the goal is to classify each business according to an economic sector. Our results reveal that our method greatly improves upon previous approaches, setting a new standard for *class-specific* keyword extraction.

1 Introduction

As the amount of information created daily continues to rise in the age of big data (Chen et al., 2014), a core challenge becomes how to extract valuable structured information from largely unstructured text documents (Tanwar et al., 2015; Song et al., 2023). An important first step in the process of Information Retrieval (IR) is often the extraction of keywords (or phrases) from documents, which can provide an initial clue about the information stored within the document (Firoozeh et al., 2020; Xie et al., 2023). With the extraction of meaningful keywords, NLP tasks such as Topic Modeling or Document Classification can be bootstrapped.

Over the past few decades, a number of unsupervised keyword extraction approaches have been proposed in the literature, ranging from frequency-based methods to statistics-based methods (Firoozeh et al., 2020), and more recently, methods using graphs or leveraging the capabilities of transformer-based language models (Nomoto, 2022; Tran et al., 2023). Supervised approaches have been proposed, with the downside of requiring reliable training data (Firoozeh et al., 2020).

While a myriad of keyword extraction approaches has appeared in the literature, they are often of the *unguided* nature, where any relevant keywords are extracted regardless of the downstream goal. As such, there has been a scarcity of research in the direction of *class-specific* keyword extraction, where only keywords adhering to a particular *class* are extracted. Presumably, this type of keyword extraction would be useful in settings where a targeted set of keywords is desired, rather than any relevant keyword in a document.

To address this open research challenge, we devise a novel class-specific keyword extraction pipeline, which builds upon the popular open-source package KeyBERT[*] (Grootendorst et al., 2023). We envision an iterative process which is guided by user-provided *seed keywords*. With these, candidate keywords are ranked according to a two-part scoring scheme, and the seed keywords are augmented by top candidates from each iteration.

We evaluate our approach on a dataset of German business registry (*Handelsregister*) entries, where the goal is to extract as many *class-specific* keywords according to *economic sectors*, as defined by an existing classification scheme. In this evaluation, we show that our method greatly outperforms previous keyword extraction methods, demonstrating the strength of our approach in extracting class-specific keywords.

The contributions of our work are as follows:

1. We address the task of *class-specific* keyword extraction with a case study in the German business registry.
2. We propose a class-specific keyword extraction pipeline that improves upon an existing

♠Equal contribution

[*]https://maartengr.github.io/KeyBERT/

Proceedings of the 20th Conference on Natural Language Processing (KONVENS 2024), pages 159–165
September 10-13, 2024

transformer-based method. Our code is found at https://github.com/sjmeis/CSKE.

3. We achieve a new standard for extracting class-specific keywords, measured in a comparative analysis with multiple metrics.

2 Related Work

A recent survey structures 167 keyword extraction approaches from the literature (Xie et al., 2023). We focus on unsupervised extraction approaches, which can generally be characterized as either statistics-, graph-, or embedding-based, while *TF-IDF* is a common frequency-based baseline method (Papagiannopoulou and Tsoumakas, 2019).

YAKE uses a set of different statistical metrics, including word casing, word position, word frequency, and more, to extract keyphrases from text (Campos et al., 2020). *TextRank* uses Part of Speech (PoS) filters to extract noun phrase candidates that are added to a graph as nodes while adding an edge between nodes if the words co-occur within a defined window (Mihalcea and Tarau, 2004; Page et al., 1999). *SingleRank* improves upon the TextRank approach by adding weights to edges based on word co-occurrences (Wan and Xiao, 2008). *RAKE* leverages a word co-occurrence graph and assigns a number of scores to aid in ranking keyword candidates (Rose et al., 2010). Knowledge Graphs can also be used to incorporate semantics for keyword or keyphrase extraction (Shi et al., 2017). *EmbedRank* leverages Doc2Vec (Le and Mikolov, 2014) and Sent2Vec (Pagliardini et al., 2018) embeddings to rank candidate keywords for extraction (Bennani-Smires et al., 2018). In a similar way, *PatternRank* uses a combination of sentence embeddings and *POS* filters (Schopf et al., 2022). Further, Language Model-based approaches have been introduced, for example using BERT (Devlin et al., 2019), for automatic extraction of keywords and keyphrases (Sammet and Krestel, 2023; Song et al., 2023).

3 A Class-Specific Keyword Extraction Pipeline

In this section, we outline in detail our proposed class-specific keyword extraction pipeline. The pipeline is illustrated in Figure 1.

Preliminaries For our pipeline, we assume three preliminary requirements:

1. **Document corpus**: unstructured text documents from any domain, from which meaningful information can be extracted.
2. **Pre-defined classes**: a set of one or more *classes*, each of which represents a distinct and well-defined concept.
3. **Class-specific seed keywords**: for each defined class, a set of *seed keywords* is available. Seed keywords are keywords that are representative of a particular class and can be used as a foundation for guided keyword extraction.

An Iterative Method Given a sizeable document corpus, we propose to process the corpus in *batches*, allowing for an iterative method, where each iteration "learns" from the previous.

For each iteration (on one batch), the first step is to extract keywords from the batch's documents in a *guided* manner. For this, we modify the popular KeyBERT package, specifically the *guided* functionality. In the current version of KeyBERT the guided functionality by default takes a set of seed keywords as input parameters, and uses a weighted average of seed keyword embeddings and document embeddings to extract candidate keywords. As we place a focus on *class-specific* seed keywords, we make the modification for *KeyBERT* to focus 100% on the seed keyword embeddings. After this modified version is run on the entire batch, the output is a list of *guided* candidate keywords (i.e., from the seed keywords).

Following the above, we employ a two-part scoring scheme to "reorder" the candidates. In particular, we use the following two scores:

- **Average Scoring**: the embedding of each candidate is compared against each seed keyword embedding, using cosine similarity, and these results are averaged for the *average score*.
- **Max Scoring**: similar to *average scoring*, but only the maximum cosine similarity score is kept, resulting in the *max score*.

We use the mean of *average score* and *max score* for the final candidate score, and all candidates for a batch are reordered based on this final score. The intuition behind such a scoring scheme is that an ideal keyword is both similar in meaning to one seed keyword, but also generally similar to all seed keywords, suggesting that such a keyword is also representative of the class in question.

The final step within one iteration includes taking the top-scoring candidates and adding them

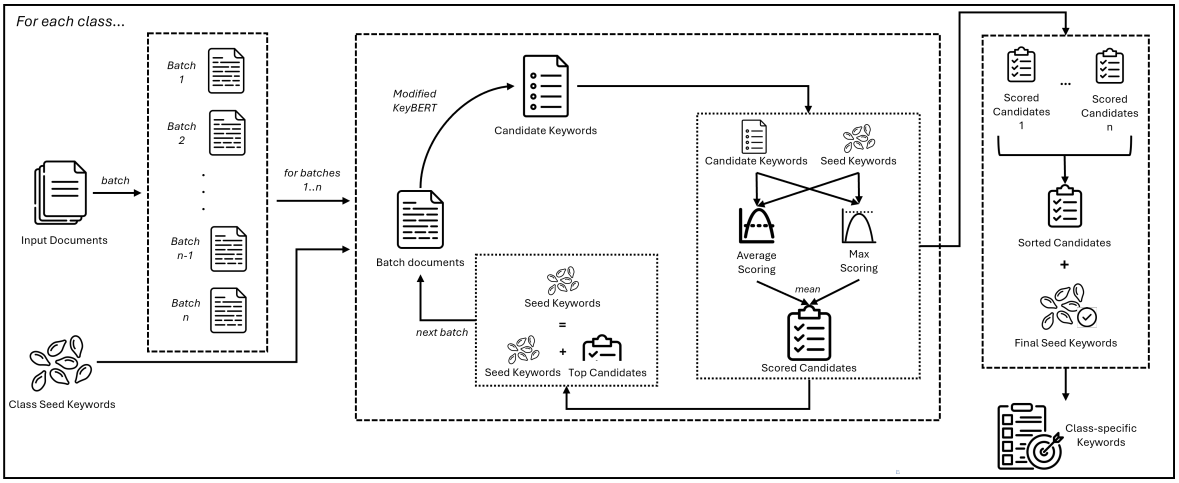

Figure 1: Our class-specific keyword extraction pipeline. With a document corpus and class-specific keyword sets as inputs, we iterate sequentially over batches of the corpus, using a modified KeyBERT and a two-part scoring scheme. Top keywords are added to the seed keywords for the next iteration, until a final set of keywords is achieved.

to the set of seed keywords. In doing so, we can iteratively "expand" the class-specific seed keywords, thus also expanding the comprehensiveness of these seeds. To do this, we define two parameters: (1) *percentile_newseed*, defining above which percentile of scores to consider (default: 99), and (2) *number_newseed*, defining how many new seed keywords to add per iteration (default: 3). Thus in the default setting, after each iteration (except the last), a maximum of 3 keywords from the top 99th percentile are added to the set of seed keywords.

Class-specific Keyword Set The output of each iteration is a set of scored candidate keywords. After all batches are processed, all scored candidates are merged and sorted. A *topk* parameter governs how many of the keywords to return, with seed keywords always being placed at the top of the list.

4 Experimental Setup and Results

Our experimental setup aims to evaluate the ability of our proposed method to extract class-specific keywords, in comparison to previous approaches. As opposed to typical keyword extraction evaluations, our evaluation tests the ability of a method to extract a set of class-specific keywords from a corpus, rather than generic keywords from documents.

Dataset We use a dataset of the German business registry (*Deutsches Handelsregister*) records, which contains 2.37 million business purpose records structured by Fusionbase[†]. The goal is to classify each business into an economic sector, according to the scheme proposed by the German Ministry of Statistics (*Statistiches Bundesamt*), called the *WZ 2008* (*Klassifikation der Wirtschaftszweige, Ausgabe 2008*)[‡]. In this work, we model the evaluation on the above dataset as a class-specific keyword extraction task, where the goal is to extract meaningful keywords for each of the 21 top-level economic sectors in the WZ 2008. For evaluation purposes, we use a random sample of 10,000 rows from the larger dataset[§].

It should be noted that we only investigate the extraction of unigram keywords. For the extraction of German keywords, this is advantageous due to the relatively high frequency of nominal compounds in the German language. Thus, meaningful keywords can be extracted in an efficient manner. However, this comes with two limitations: (1) not all *keyphrases* will be caught, thus sometimes leading to incomplete keywords (see "Dicke" in Listing 1, which means *thick* translated to English), and (2) the results achieved for German language datasets may not be directly generalizable to English.

Keyword Extraction Methods For a comparative analysis, we test our method against four methods: (1) RAKE (Rose et al., 2010), (2) YAKE (Campos et al., 2020), (3) KeyBERT, and (4) Guided KeyBERT. Note that RAKE and YAKE do not offer any mechanism for guided keyword extraction, and thus the result-

[†] https://fusionbase.com

[‡] https://www.destatis.de/DE/Methoden/Klassifikationen/Gueter-Wirtschaftsklassifikationen/klassifikation-wz-2008.html

[§] This sample can be found in our code repository.

		Precision@10	Precision@25	Precision@50	Precision@100	Average
Exact Match	RAKE	0.95	1.33	1.71	1.42	1.36
	YAKE	3.33	3.24	2.38	1.81	2.69
	KeyBERT	1.90	1.71	1.71	1.05	1.60
	Guided KeyBERT	2.38	1.90	1.90	1.24	1.86
	Ours	**28.10**	**22.67**	**13.62**	**8.33**	**18.23**
Lemma Match	RAKE	1.43	1.52	1.90	1.76	1.65
	YAKE	2.38	3.24	2.67	2.33	2.65
	KeyBERT	1.90	1.90	1.81	1.29	1.73
	Guided KeyBERT	2.38	1.90	2.10	1.48	1.96
	Ours	**21.43**	**20.76**	**13.43**	**9.00**	**16.15**
Fuzzy Match	RAKE	62.60	61.63	61.95	59.95	61.29
	YAKE	65.91	65.10	62.98	59.58	63.39
	KeyBERT	60.42	60.49	59.55	57.16	59.41
	Guided KeyBERT	60.67	60.62	59.95	57.42	59.67
	Ours	**78.19**	**75.21**	**72.93**	**67.54**	**73.47**
CS Match	RAKE	77.54	79.48	79.73	79.83	79.14
	YAKE	82.48	83.52	82.69	82.05	83.13
	KeyBERT	76.73	77.39	77.09	76.86	77.02
	Guided KeyBERT	77.30	77.76	77.66	77.36	77.52
	Ours	**86.32**	**86.82**	**86.02**	**85.36**	**86.13**
Average Match	RAKE	35.70	36.02	36.37	35.52	35.91
	YAKE	38.90	38.69	37.62	36.40	37.90
	KeyBERT	34.88	35.14	34.99	34.16	34.79
	Guided KeyBERT	35.21	35.31	35.38	34.44	35.08
	Ours	**53.51**	**51.41**	**46.50**	**42.56**	**48.49**

Table 1: Class-specific Keyword Extraction Evaluation Results. For each scoring scheme, the highest score for each k is **bolded**. The average in the right column represents the average of the four evaluated k values. *Average Match* denotes the average score achieved by a method for one k but across all four scores. Examples of extracted keywords for each approach are provided in Appendix A.

ing keywords are the same for each class. We test our proposed method with the parameter *n_iterations* (number of batches) set to 5. *Guided* KeyBERT refers to the use of the optional `seed_keywords` parameter, which serves as a direct comparison point to our proposed method (denoted *ours*). For KeyBERT and our method, we use the DEUTSCHE-TELEKOM/GBERT-LARGE-PARAPHRASE-COSINE language model. Note that for comparability, KeyBERT was set only to extract unigram keywords.

Seed Keywords For the selection of seed keywords, specifically for Guided KeyBERT and our method, we utilize an existing collection of keywords (*Stichwörter*) provided by the creators of the WZ 2008[‡]. As we aim only to extract unigrams, we truncate all keyphrases to the first word if they are longer than one word. From this gold set, we randomly select 10 keywords from each class to serve as the seeds for that class. The rest of the gold set is then used for evaluation. The seed keywords from two classes are presented in Listings 1 and 2.

```
['Schweinehaltung', 'Holztaxierung',
 'Austernzucht', 'Teichwirtschaft',
 'Tabak'*, 'Dicke',
 'Fischerei'*, 'Seidenraupenzucht',
 'Wild', 'Kassava']
```

Listing 1: Seed Keywords for Class A: *Land- und Forstwirtschaft, Fischerei*. Seed keywords marked with an asterisk (*) denote those found in our dataset sample.

```
['Heizkraftwerke'*, 'Elektrizitaetserzeugung',
 'Blockheizkraftwerk'*, 'Waermeversorgung',
 'Solarstromerzeugung', 'Bereitstellung'*,
 'Energieversorgung'*, 'Windparks'*,
 'Spaltgaserzeugung', 'Kokereigasgewinnung']
```

Listing 2: Seed Keywords for Class D: *Energieversorgung*. Seed keywords marked with an asterisk (*) denote those found in our dataset sample.

Metrics With the keywords sets from each of the tested methods, we evaluate the accuracy of the keywords on two dimensions: (1) *precision@K*, where the number of correct keywords amongst the top K output keywords is counted, and (2) *matching method*, where the meaning of "correct" is varied. For K, we choose $K \in \{10, 25, 50, 100\}$, and for matching method, we use four approaches:

- **Exact string match**: a correct keyword is counted if the extracted keyword is found *exactly* in the gold set of keywords.
- **Lemma match**: a correct keyword is counted if the *lemmatized* version of the keyword is found in the *lemmatized* gold set of keywords (Zesch and Gurevych, 2009).
- **Fuzzy string match**: the "correctness" of a keyword is not binary, but rather is represented by the closest fuzzy string match score, using the Python package THEFUZZ.
- **Cosine similarity match**: the correctness of

a keyword is measured by its highest cosine similarity to any of the gold keywords.

For cosine similarity, the DEEPSET/GBERT-BASE model is used, so as not to use the same base model used with the keyword extraction process.

Results Table 1 presents the results of the above-described experiments. Note that for the evaluation of extracted keywords against the gold set, we only include keywords in the gold set that appear (in lemmatized form for *lemma match*) in the 10k sample of the German business registry data.

We can observe that our approach outperforms all other methods in *class-specific* keyword extraction. The performance of our approach is particularly strong in the exact match and lemma match evaluations, indicating it is well suited to extract class-specific gold keywords as defined by the creators of the WZ 2008[‡] classification scheme. Notably, even the Guided KEYBERT method, designed to extract keywords similar to provided seed keywords, performs significantly worse than our approach. Looking to the results, we see that the guided version of KEYBERT often only shows improvements over the base version when more extracted results are considered. This implies that while some class-specific keywords are found, they are not ranked as high as other keywords. Ultimately, we conclude that our approach achieves state-of-the-art results for *class-specific* keyword extraction, a point that is supported by a qualitative analysis of example outputs in Appendix A.

5 Conclusion

We present a class-specific keyword extraction pipeline which outperforms previous methods in identifying keywords related to a predefined *class*. Our evaluation results exhibit the strong performance of our method in the task of retrieving keywords specific to particular German economic sectors. These results make a compelling case for the continued study of class-specific keyword extraction as an improvement to non-guided approaches.

As points for future work, we propose more rigorous evaluation of our method from two perspectives: (1) an ablation study on the effect of the *n_iterations*, *number_newseed*, *percentile_newseed*, and *topk* parameters, in particular to study their relevance for class-specific keyword extraction, and (2) evaluation of our method beyond the German language, firstly with English.

Acknowledgments

This work has been supported by the BayVFP Digitalization grant DIK-2210-0028//D1K0475102 (CreateData4AI) from the Bavarian Ministry of Economic Affairs, Regional Development and Energy. The project is performed in collaboration with Fusionbase GmbH, whom we thank for the Business Registry data access and for the guidance.

Limitations

The primary limitation of our work is the lack of evaluation of the various parameters of our method, as discussed in Section 5. Evaluating a range of values would strengthen the work in determining the individual effect of each parameter.

The second limitation involves the relatively limited scope both in domain and language. In particular, we focus our case study only on the German Business Registry, and we do not generalize beyond this to different domains or languages.

Ethics Statement

An ethical consideration comes with the use of the German Business Registry dataset, which is directly tied to real-world businesses, potentially raising privacy concerns. However, this is mitigated by the fact that the data is public and business owners are aware of this when drafting their entries.

References

Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple unsupervised keyphrase extraction using sentence embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 221–229, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289.

Min Chen, Shiwen Mao, and Yunhao Liu. 2014. Big data: A survey. *Mobile networks and applications*, 19:171–209.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages

4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Nazanin Firoozeh, Adeline Nazarenko, Fabrice Alizon, and Béatrice Daille. 2020. Keyword extraction: Issues and methods. *Natural Language Engineering*, 26(3):259–291.

Maarten Grootendorst, Abhay Mishra, Art Matsak, OysterMax, Priyanshul Govil, Yuki Ogura, Vincent D Warmerdam, and yusuke1997. 2023. Maartengr/keybert: v0.8.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China. PMLR.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Tadashi Nomoto. 2022. Keyword extraction: a modern perspective. *SN Computer Science*, 4(1):92.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking : Bringing order to the web. In *WWW 1999*.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540, New Orleans, Louisiana. Association for Computational Linguistics.

Eirini Papagiannopoulou and Grigorios Tsoumakas. 2019. A review of keyphrase extraction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, pages 1–20.

Jill Sammet and Ralf Krestel. 2023. Domain-specific keyword extraction using BERT. In *Proceedings of the 4th Conference on Language, Data and Knowledge*, pages 659–665, Vienna, Austria. NOVA CLUNL, Portugal.

Tim Schopf, Simon Klimek, and Florian Matthes. 2022. Patternrank: Leveraging pretrained language models and part of speech for unsupervised keyphrase extraction. In *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2022) - KDIR*, pages 243–248. INSTICC, SciTePress.

Wei Shi, Weiguo Zheng, Jeffrey Xu Yu, Hong Cheng, and Lei Zou. 2017. Keyphrase extraction using knowledge graphs. *Data Science and Engineering*, 2:275–288.

Mingyang Song, Yi Feng, and Liping Jing. 2023. A survey on recent advances in keyphrase extraction from pre-trained language models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2153–2164, Dubrovnik, Croatia. Association for Computational Linguistics.

Mona Tanwar, Reena Duggal, and Sunil Kumar Khatri. 2015. Unravelling unstructured data: A wealth of information in big data. In *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, pages 1–6.

Hanh Thi Hong Tran, Matej Martinc, Jaya Caporusso, Antoine Doucet, and Senja Pollak. 2023. The recent advances in automatic term extraction: A survey. *arXiv preprint arXiv:2301.06767*.

Xiaojun Wan and Jianguo Xiao. 2008. CollabRank: Towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 969–976, Manchester, UK. Coling 2008 Organizing Committee.

Binbin Xie, Jia Song, Liangying Shao, Suhang Wu, Xiangpeng Wei, Baosong Yang, Huan Lin, Jun Xie, and Jinsong Su. 2023. From statistical methods to deep learning, automatic keyphrase prediction: A survey. *Information Processing & Management*, 60(4):103382.

Torsten Zesch and Iryna Gurevych. 2009. Approximate matching for evaluating keyphrase extraction. In *Proceedings of the International Conference RANLP-2009*, pages 484–489, Borovets, Bulgaria. Association for Computational Linguistics.

A Extracted Keyword Examples


```
{'rake': ['analyse',
  'entwicklung',
  'software',
  'programmen',
  'weiterentwicklung',
  'verkauf',
  'vermietung',
  'domainadressen',
  'housing',
  'domainverwaltung',
  'peering',
  'administration',
  'saemtliche',
  'handel',
  'insbesondere'],
 'yake': ['uebernahme',
  'dienstleistungen',
  'geschaefte',
  'beteiligung',
  'verkauf',
```

```
 'entwicklung',
 'vermittlung',
 'geschaeftsfuehrung',
 'beratung',
 'herstellung',
 'beteiligungen',
 'taetigkeiten',
 'erbringung',
 'bereich',
 'immobilien'],
'keybert': ['landschaftsbau',
  'photovoltaik',
  'elektroinstallationen',
  'masskleidung',
  'landschaftsmusikfestivals',
  'systemgastronomie',
  'bauleistungen',
  'reisebueros',
  'immobilien',
  'physiotherapie',
  'wasserinstallationsarbeiten',
  'diskothek',
  'nassbaggerarbeiten',
  'druckereierzeugnissen',
  'zahntechnischen'],
'guided_keybert': ['landschaftsbau',
  'elektroinstallationen',
  'photovoltaik',
  'systemgastronomie',
  'landschaftsmusikfestivals',
  'masskleidung',
  'bauleistungen',
  'reisebueros',
  'immobilien',
  'diskothek',
  'wasserinstallationsarbeiten',
  'druckereierzeugnissen',
  'nassbaggerarbeiten',
  'physiotherapie',
  'zahntechnischen'],
'ours': ['zucht',
  'fuger',
  'getreide',
  'spenglerei',
  'verpachtungen',
  'veraeu',
  'frachten',
  'fracht',
  'schalungen',
  'verpachtung',
  'beund',
  'kalk',
  'schalung',
  'holzwaren',
  'haefte']
}
```

Listing 3: Sample extracted keywords for Class A, from the 10:25 top keywords for each method.

```
{'rake': ['analyse',
  'entwicklung',
  'software',
  'programmen',
  'weiterentwicklung',
  'verkauf',
  'vermietung',
  'domainadressen',
  'housing',
  'domainverwaltung',
  'peering',
  'administration',
  'saemtliche',
  'handel',
  'insbesondere'],
'yake': ['uebernahme',
 'dienstleistungen',
 'geschaefte',
 'beteiligung',
 'verkauf',
 'entwicklung',
 'vermittlung',
 'geschaeftsfuehrung',
 'beratung',
 'herstellung',
 'beteiligungen',
 'taetigkeiten',
 'erbringung',
 'bereich',
 'immobilien'],
'keybert': ['immobilien',
  'delaware',
  'verkauf',
  'pizzalieferservices',
  'unternehmens',
  'ambulanten',
  'eingliederungshilfe',
  'gesellschaftsbeteiligungen',
  'bebauung',
  'schulverwaltungssoftware',
  'geschaeftsfuehrung',
  'textilzubehoer',
  'masskleidung',
  'motorradzubehoerteilen',
  'casinobetriebe'],
'guided_keybert': ['immobilien',
  'delaware',
  'kraftfahrzeugen',
  'pizzalieferservices',
  'unternehmens',
  'ambulanten',
  'eingliederungshilfe',
  'gesellschaftsbeteiligungen',
  'bebauung',
  'schulverwaltungssoftware',
  'geschaeftsfuehrung',
  'textilzubehoer',
  'masskleidung',
  'motorradzubehoerteilen',
  'casinobetriebe'],
'ours': ['energieanlagen',
  'energieerzeugungsanlagen',
  'energieerzeugung',
  'energietechnik',
  'energieversorgungs',
  'energietechnischen',
  'energieprodukten',
  'stromerzeugungsanlagen',
  'energiegewinnung',
  'energietraeger',
  'energietraegern',
  'energiequellen',
  'ernergieanlagen',
  'energie',
  'stromerzeugern']}
```

Listing 4: Sample extracted keywords for Class D, from the 10:25 top keywords for each method.