Semantic Label Representations with Lbl2Vec: A Similarity-Based Approach for Unsupervised Text Classification

Tim Schopf[1(✉)], Daniel Braun[2], and Florian Matthes[1]

[1] Department of Computer Science, Technical University of Munich, Boltzmannstrasse 3, Garching, Germany
{tim.schopf,matthes}@tum.de

[2] Department of High-tech Business and Entrepreneurship, University of Twente, Drienerlolaan 5, Enschede, The Netherlands
d.braun@utwente.nl

Abstract. In this paper, we evaluate the Lbl2Vec approach for unsupervised text document classification. Lbl2Vec requires only a small number of keywords describing the respective classes to create semantic label representations. For classification, Lbl2Vec uses cosine similarities between label and document representations, but no annotation information. We show that Lbl2Vec significantly outperforms common unsupervised text classification approaches and a widely used zero-shot text classification approach. Furthermore, we show that using more precise keywords can significantly improve the classification results of similarity-based text classification approaches.

Keywords: Natural language processing · Unsupervised text classification · Text representations · Text similarity · Semantic label representations

1 Introduction

Supervised text classification has gained a lot of attention recently, due to the succes of Pretrained Language Models (PLMs). Training supervised classification algorithms or even fine-tuning PLMs requires a large amount of labeled data. However, high-quality annotated datasets often do not exist, particularly in industrial settings. Annotating datasets usually requires a lot of manual effort and causes high expenses. Unsupervised text classification approaches, however, can significantly reduce annotation costs since they can be trained on unlabeled datasets. Despite this opportunity, supervised text classification approaches based on transformer models such as BERT [6] or XLNet [28] are significantly more studied than unsupervised text classification approaches. In this work, we contribute to the less researched field of unsupervised text classification by evaluating the Lbl2Vec [18] approach.

The general approach for unsupervised text classification is to map text to labels based on their textual description. Thereby, classification is based on semantic similarities of text representations and thus avoids the need for annotated training data. Usually, this kind of approach is applied when dealing with a large corpus of unlabeled text documents that need to be classified into topics of interest. These types of tasks

M. Marchiori et al. (Eds.): WEBIST 2020/2021, LNBIP 469, pp. 59–73, 2023.
https://doi.org/10.1007/978-3-031-24197-0_4

are becoming increasingly common, considering the ever growing amount of unlabeled text data. To illustrate the problem, we assume the following scenario as an example: we collected a large number of tech-related text articles from various websites. From this corpus, we want to classify articles based on their relatedness to certain companies such as Apple, Google or Microsoft. Since we do not possess any metadata about the text articles, we can only rely on the texts themselves for this purpose. What appears to be a simple text classification task initially, may actually turn out to be more complex than expected. To use a conventional supervised classification approach, we would need to annotate the text articles first, since they require a large amount of labeled training data [33]. As already mentioned, this likely involves high annotation expenses.

In this work, we evaluate the similarity-based Lbl2Vec approach, which is able to perform unsupervised classification on a large corpus of unlabeled text documents. This approach enables us to classify a text document corpus without having to annotate any data. For classification, Lbl2Vec uses semantic similarities between documents and keywords describing a certain class only. Intuitively, using semantic meanings matches the approach of a human being. In addition, this approach can significantly reduce annotation costs since only a small number of keywords are needed instead of a large number of labeled documents.

Lbl2Vec creates jointly embedded word, document and label representations. The label representations are obtained from the manually predefined keywords. Because vector representations of documents and labels share the same embedding space, their semantic relationship can be measured using cosine similarity. Eventually this similarity can be used to assign a certain class to a text document.

The contributions of our work can be summarized as follows:

- We provide a comprehensive explanation of Lbl2Vec, based on the original paper [18] and additional illustrations.
- We evaluate Lbl2Vec against commonly used unsupervised text classification approaches and against a state-of-the-art zero-shot learning (ZSL) approach.
- We conduct experiments with different Lbl2Vec hyperparameter settings and examine which hyperparameter values yield good label vectors.
- We examine the role of keywords used to describe classes for similarity-based text classification.

2 Lbl2Vec

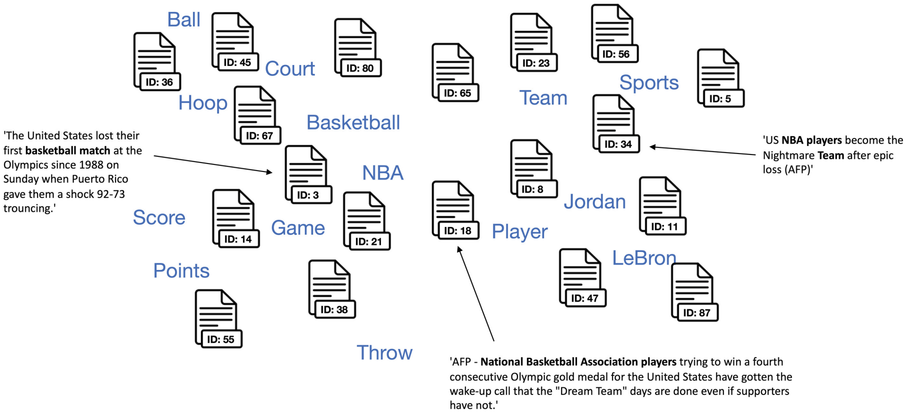
Lbl2Vec [18] is a similarity-based approach for unsupervised text classification. It creates jointly embedded label, document, and word vector representations from a given text document corpus. The semantic label representations are derived from predefined keywords for each class and used to classify text documents. The intuition of this approach is that many semantically similar keywords can represent a class. First, Lbl2Vec generates jointly embedded document and word vector representations. Then, the algorithm learns label vectors from the predefined keywords. Finally, Lbl2Vec classifies documents based on similarities between the document and label representations. Since label and document representations share the same embedding space, their cosine similarities can be used as a classification indicator. The authors made Lbl2Vec publicly available as a ready-to-use tool under the 3-Clause BSD license[1].

In addition to the text document corpus, Lbl2Vec requires manually predefined keywords as input. For each class, a set of semantically coherent keywords will be used to create the label vector representation later. Table 1 shows example keywords representing different classes.

Table 1. Manually predefined example keywords for different sports classes.

Class	Keywords
Basketball	NBA, Basketball, LeBron
Soccer	FIFA, Soccer, Messi
Baseball	MLB, Baseball, Ruth

Given the predefined keywords and the unlabeled text document corpus as input, Lbl2Vec initially learns jointly embedded word and document vector representations using Doc2Vec [10]. Specifically, Lbl2Vec uses the distributed bag of words version of paragraph vector (PV-DBOW) and interleaves it with Skip-gram [13] training to learn jointly embedded document and word representations. After learning jointly embedded vectors, representations of semantically similar documents are located close to each other in embedding space and also close to representations of the most distinguishing words. Figure 1 illustrates the jointly embedded document and word representations.



'The United States lost their first **basketball match** at the Olympics since 1988 on Sunday when Puerto Rico gave them a shock 92-73 trouncing.'

'US **NBA players** become the Nightmare **Team** after epic loss (AFP)'

'AFP - **National Basketball Association players** trying to win a fourth consecutive Olympic gold medal for the United States have gotten the wake-up call that the "Dream Team" days are done even if supporters have not.'

Fig. 1. Example illustration of jointly embedded document and word vector representations, learned by Lbl2Vec [19].

Following the learning of jointly embedded document and word representations, Lbl2Vec uses the class keywords to train semantic label representations. For each class, Lbl2Vec uses the cosine similarities between the average of the keyword vector representations and the document vector representations to find a set of most similar

[1] https://github.com/sebischair/Lbl2Vec.

candidate documents. To include only the document representations most similar to the predefined keywords in the set of candidate documents, Lbl2Vec requires the following parameters:

- s as cosine similarity threshold between the average of the keyword vector representations and the document vector representations. Only documents that exceed s are included in the candidate documents.
- d_{min} as the minimum number of documents for each set of candidate documents. This parameter prevents the selection of an insufficient number of documents in case s is chosen too restrictive.
- d_{max} as the maximum number of documents for each set of candidate documents.

Figure 2 illustrates candidate documents for some example classes.

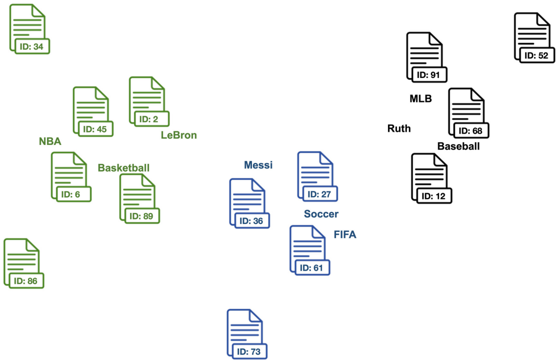


Fig. 2. Example illustration of class keyword representations with their respective set of candidate document representations in embedding space. Each color represents a different class [19].

To remove noise, Lbl2Vec cleans outlier documents from each set of candidate documents using local outlier factor (LOF) [2]. Thereby, Lbl2Vec removes documents with significantly lower local density than their neighbors. The intuition of this cleaning step is to ensure a more accurate label embedding in subsequent steps by removing candidate documents that are related to the keywords but do not align with the intended classification category. Figure 3 illustrates the outlier cleaning process of Lbl2Vec.

After obtaining the cleaned sets of candidate documents, Lbl2Vec computes the average of candidate document representations for each class as semantic label vector representations. Experiments showed it is difficult to classify text documents based on similarities to keywords, even if their representations share the same embedding space [18]. Therefore, Lbl2Vec computes the label vectors as averages of document representations rather than averages of keyword representations. Figure 4 illustrates examples of label vector representations.

For classification, Lbl2Vec uses the cosine similarities between the label vector representations and document vector representations. Text documents are assigned to the class where their vector representations are most similar to the respective semantic label representation. Figure 5 illustrates an example classification result.

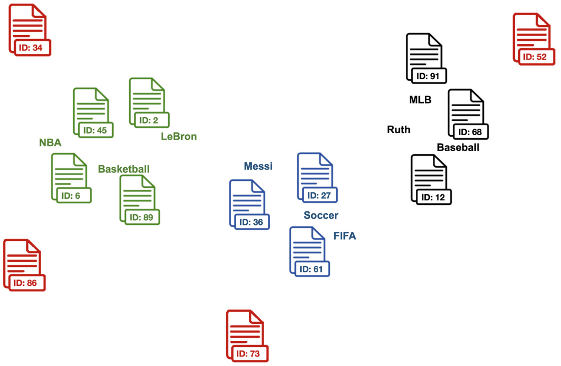


Fig. 3. Example illustration of the Lbl2Vec outlier cleaning step. Red documents are outliers that are removed from the candidate documents [19] (Color figure online).

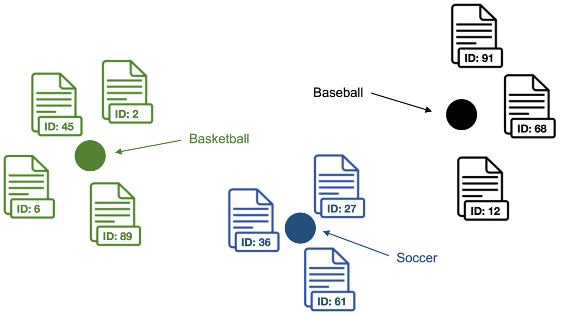


Fig. 4. Example illustration of the label vector representations, calculated as the average of the respective set of cleaned candidate document representations [19].

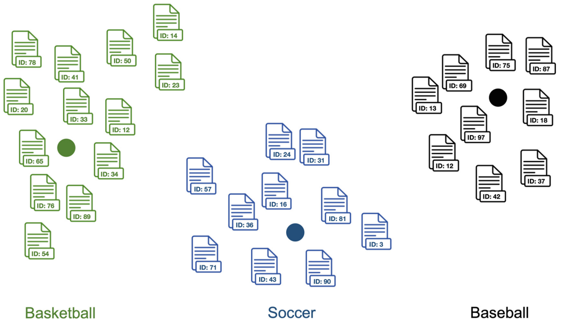


Fig. 5. Example illustration of a Lbl2Vec classification result. Circles represent the label vectors of classes. Colors represent the classes [19].

3 Experimental Design

3.1 Datasets

To conduct unsupervised text classification experiments, we use the 20Newsgroups and AG's Corpus datasets. The 20Newsgroups dataset is a common text classification dataset which consists of approximately 20,000 different news articles, equally distributed across 20 different classes [9]. The AG's Corpus dataset consists of 127,600 news articles, equally distributed among four different classes [32]. Table 2 shows a summary of the used datasets.

Table 2. Summary of the used text classification datasets [18].

Datasets	#Training documents	#Test documents	#Classes
20Newsgroups	11,314	7,532	20
AG's Corpus	120,000	7,600	4

3.2 Label Keywords

We adopt the expert knowledge approach [8] to define keywords for each class in the respective datasets. Therefore, we define some initial keywords based on the class names. Afterwards, we select some random documents from each class to derive more salient keywords. Table 3 and Table 4 show some of the resulting keywords.

Table 3. AG's Corpus class names and label keywords.

Class name	Label keywords
World	government, election, state, president, politics, democracy, war, ...
Sports	sports, football, baseball, rugby, basketball, championship, ...
Business	business, company, market, oil, consumers, price, products, ...
Science/Technology	science, technology, web, google, microsoft, software, laboratory,

3.3 Text Classification Approaches

For evaluation, we conduct experiments with the following text classification approaches:

Word2Vec: We use Word2Vec [13] to create semantic word vector representations for each dataset. To learn the word representations, we use a Skip-gram model with a vector size of 300 and a surrounding window of 5. Then, we use the average of word vectors as document and label representations. For classification, we use the cosine similarities between the resulting document and label representations. Documents are assigned to the class where the cosine similarity to the class label representation is the highest.

SBERT: Sentence-BERT (SBERT) is a modification of BERT [6] that uses siamese and triplet network structures to derive semantically meaningful sentence embeddings [15].

Table 4. 20Newsgroups class names and label keywords.

Class name	Label keywords
alt.atheism	atheism, god, atheists, religion, atheist, belief, believe, jesus, ...
comp.graphics	image, graphics, jpeg, images, gif, tiff, quicktime, animation, ...
comp.os.ms-windows.misc	windows, microsoft, win, driver, computer, ...
comp.sys.ibm.pc.hardware	bus, drives, bios, disk, dos, motherboard, floppy, cpu, port, ...
comp.sys.mac.hardware	mac, apple, hardware, monitor, powerbook, macintosh, ...
comp.windows.x	computer, windows, program, openwindows, application, ...
misc.forsale	sale, shipping, forsale, price, sell, offer, trade, ...
rec.autos	cars, engine, ford, dealer, oil, toyota, driver, tires, ...
rec.motorcycles	motorcycles, bike, ride, bmw, helmet, honda, harley, ...
rec.sport.baseball	sport, baseball, game , team, hit, pitcher, hitter, sox, ...
rec.sport.hockey	sport, hockey, season, nhl, cup, playoffs, ...
sci.crypt	encryption, key, privacy, algorithm, nsa, security, ...
sci.electronics	electronics, wire , battery, voltage, power, amp, ...
sci.med	medical, disease, cancer, patients, health, doctor, medicine, ...
sci.space	space, nasa, orbit, moon, earth, solar, satellite, mars, ...
soc.religion.christian	religion, christians, god , church, bible, jesus, christ, believe, ...
talk.politics.guns	guns, fbi, firearms, weapons, militia, crime, violence, ...
talk.politics.mideast	israel, armenia, turkey, arab, muslim, ...
talk.politics.misc	president, government, clinton, jobs, tax, insurance, state, ...
talk.religion.misc	religion, jesus, god, bible, lord, moral, judas,

We use the average of SBERT sentence embeddings as document representations and the average of SBERT keyword embeddings as class representations. Then, we classify the text documents according to the highest cosine similarity of the resulting SBERT representations of documents and classes. For our experiments, we use the pretrained general purpose *all-mpnet-base-v2* SBERT model.

Zero-Shot Text Classification: In general, zero-shot text classification (0SHOT-TC) approaches use labeled training instances of seen classes to predict testing instances of unseen classes [26]. Although 0SHOT-TC approaches use annotated data for training, they do not use label information about the target classes and generalize their learned knowledge to classify instances of unseen classes. Because pretrained 0SHOT-TC models do not require training or fine-tuning on labeled instances from target classes, they can be classified as a type of unsupervised text classification strategy. Traditional text classifiers usually struggle to understand the underlying classification problem because class names are converted to simple indices [30]. This makes it difficult for them to generalize to unseen classes. Therefore, a 0SHOT-TC approach similar to that of humans is required, which classifies instances based on semantic class meanings. This is precisely the intuition behind the idea of modeling 0SHOT-TC as an entailment problem [30]. The zero-shot entailment model uses the class label descriptions as hypotheses and is therefore able to understand the semantic meanings of classes [30]. This approach allows the classifier to generalize to unseen classes and currently produces state-of-the-art results in the *label-fully-unseen* 0SHOT-TC setting. In the *label-fully-unseen* setting, 0SHOT-TC aims at learning a classifier $f(\cdot) : X \rightarrow Y$, where classifier $f(\cdot)$ never sees Y-specific labeled data in its model development [30].

For our experiments, we choose a DistilBART zero-shot entailment model, trained on the MultiNLI dataset [27] to classify the respective whole document corpora. As hypotheses we use the respective keywords lists concatenated with "and".

KE + LSA: This refers to an approach that uses keyword enrichment (KE) and subsequent unsupervised classification based on Latent Semantic Analysis (LSA) [5] vector cosine similarities [8]. For this approach, we do not conduct any experiment ourselves, but use the results reported in the original paper [8] for evaluation.

Lbl2Vec: We train Lbl2Vec [18] models, using the respective datasets described in Sect. 3.1. We conduct experiments with different d_{min} values, while $s = 1$, and $d_{max} =$ *the maximum number of documents in the respective dataset*. The detailed results of the experiments using different d_{min} values are shown in Sect. 4.2. The respective best F1-scores on both data sets obtained with Lbl2Vec are shown in Table 5.

For each approach, we conduct experiments with two different keywords sets. First, we use the manually predefined keywords described in Sect. 3.2. Then, we use the respective class names as keywords. For the 20Newsgroups dataset, we separate the class names at the dots and replace the abbreviations with their full names. Section 4.1 shows the results of the experiments that use the manually predefined keywords. The results of the experiments that use the class names as keywords are shown in Sect. 4.3.

4 Evaluation

4.1 Classification Results

We classify the documents from the datasets described in Sect. 3.1 using the keywords described in Sect. 3.2 and the approaches described in Sect. 3.3. Since we do not need label information to train the classifiers, we use the entire concatenated datasets for training and testing respectively. Table 5 shows the classification results of our experiments.

Table 5. F1-scores (micro) of text classification approaches on different datasets. For all experiments, the keywords described in Sect. 3.2 are used. The best results on the respective dataset are displayed in bold. Since we use micro-averaging to calculate our classification results, we realize equal F1, Precision, and Recall scores respectively.

	20Newsgroups	AG's corpus
Word2Vec	22.68	34.52
SBERT (all-mpnet-base-v2)	63.42	79.39
Zero-shot Entailment (DistilBART)	12.42	32.54
KE + LSA	61.0	76.6
Lbl2Vec	**77.03**	**82.96**

We observe that Lbl2Vec yields best F1-scores among all approaches by a large margin on both datasets. It even outperforms the SBERT approach, although SBERT currently generates sentence embeddings that achieve state-of-the-art results in text similarity tasks. Although the SBERT approach performs worse than Lbl2Vec, it nevertheless shows that the more advanced SBERT embeddings perform significantly better than Word2Vec embeddings in this similarity-based classification task. Surprisingly, the zero-shot entailment approach performs significantly worse than the Word2Vec approach and even worst of all approaches examined. The KE + LSA approach, which uses basic LSA embeddings, performs comparatively well and only slightly worse than the SBERT approach, which uses more sophisticated transformer-based embeddings.

4.2 Lbl2Vec Hyperparameter Analysis

To examine the effect of the number of documents used to compute the label vectors on the Lbl2Vec classification results, we conduct experiments with different d_{min} parameter values. The higher the d_{min} parameter values, the more document representations are used for calculating the label vector as their average. To conduct the experiments, we use the datasets described in Sect. 3.1 and the keywords described in Sect. 3.2. The results are shown in Fig. 6 and Fig. 7.

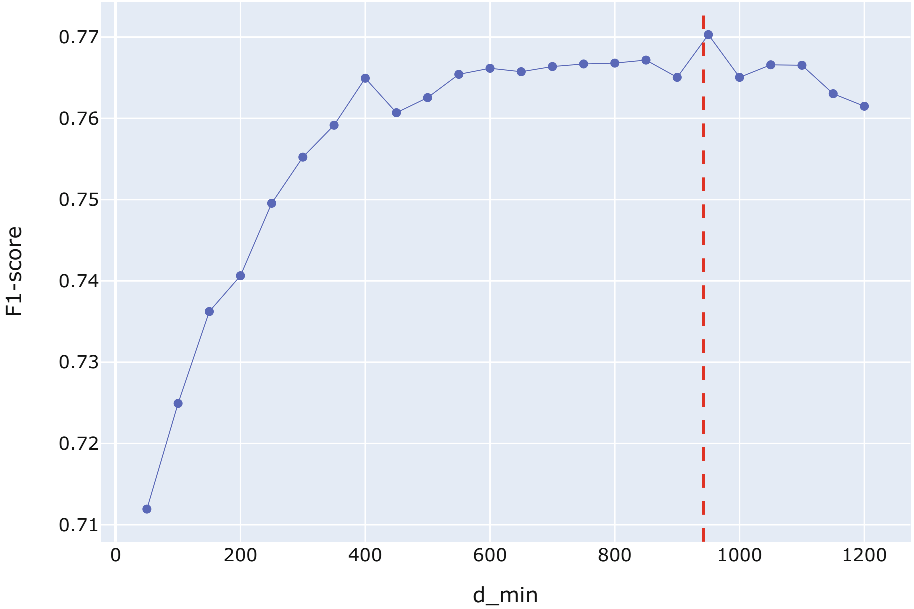


Fig. 6. F1-scores of Lbl2Vec on the 20Newsgroups dataset with different d_{min} parameter values, while $s = 1$ and $d_{max} = 18,846$ are fixed. For the experiments, the keywords described in Sect. 3.2 are used. The red line indicates the average number of documents per class. (Color figure online)

On both datasets, we observe that F1-scores improve with increasing d_{min} parameter values until a peak is reached. After the peak, the F1-scores get worse with increasing d_{min} parameter values. For the 20Newsgroup dataset, the peak occurs after d_{min} is higher than the average number of documents per class. However, the F1-scores already reach a high plateau after d_{min} is about 60% of the average number of documents per class. For the AG's Corpus dataset, the peak occurs at $d_{min} = 22,000$. This is about 69% of the average number of documents per class. However, almost similar F1-scores are achieved at $d_{min} = 18,000$, which is about 56% of the average number of documents per class.

The results show, that a sufficiency amount of candidate document representations are needed to compute good label vectors. Furthermore, the results indicate that a minimum d_{min} value of approximately 60% of the average number of documents per class yields good label vectors for classification.

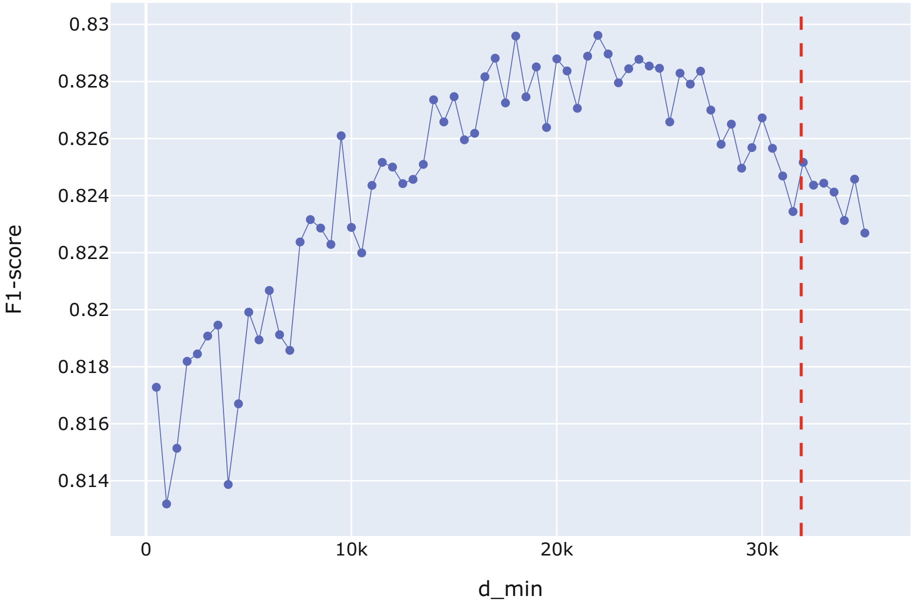


Fig. 7. F1-scores of Lbl2Vec on the AG's Corpus dataset with different d_{min} parameter values, while $s = 1$ and $d_{max} = 127,600$ are fixed. For the experiments, the keywords described in Sect. 3.2 are used. The red line indicates the average number of documents per class. (Color figure online)

4.3 Keywords Analysis

Table 6. F1-scores (micro) of text classification approaches on different datasets. For all experiments, the class names are used as keywords. The best results on the respective dataset are displayed in bold. Since we use micro-averaging to calculate our classification results, we realize equal F1, Precision, and Recall scores respectively.

	20Newsgroups	AG's Corpus
Word2Vec	11.71	26.61
SBERT (all-mpnet-base-v2)	52.98	62.75
Zero-shot Entailment (DistilBART)	43.29	64.70
Lbl2Vec	**67.64**	**66.02**

To examine how the use of different keywords affects the classification results, we conduct experiments using the class names as label keywords instead of the manually predefined ones. For the 20Newsgroups dataset, we separate the class names at the dots and replace the abbreviations with their full names. The results of these experiments are shown in Table 6.

Overall, Lbl2Vec outperforms all other text classification approaches examined. Furthermore, we observe that in comparison to the experiments using the manually predefined keywords in Sect. 4.1, the F1-scores for the Word2Vec, SBERT, and Lbl2Vec approaches decrease significantly. The pure class names, in comparison to the manually predefined keywords, contain fewer and less precise class descriptions, which affects the label vectors and the classification results negatively. However, this only applies to the similarity-based text classification approaches. In contrast, the zero-shot entailment approach yields significantly improved classification results using the class names as hypotheses instead of the manually predefined keywords. The experiments show that the simultaneous use of many keywords as hypothesis confuses the zero-shot entailment approach. As a result, this affects the 0SHOT-TC performance negatively.

5 Related Work

Most unsupervised text classification approaches leverage semantic text similarities. Thereby, these approaches generate semantic representations of texts as well as of label descriptions, and then aim to align the texts with the labels using similarity metrics. In one of the earlier works, this approach was described as "Dataless Classification" [3]. Thereby, text and label descriptions were embedded in a common semantic space using Explicit Semantic Analysis (ESA) [7] and the label with the highest matching score was selected for classification [3]. Further, ESA was applied in a dataless hierarchical classification approach that exploited the hierarchical structure of labels [23]. The general idea of "Dataless Classification" is based on the assumption that, for text classification, label representations are equally important as text representations and already was studied extensively [4,11,24].

Eventually, the term “Dateless Classification” became less common and currently rather fits into the general concept of similarity-based text classification approaches. A very common similarity-based approach, which is also often used as a baseline for unsupervised text classification, embeds texts and labels with Word2Vec [13] and tries to predict the correct class with cosine similarities [16]. Word2Vec [13] creates semantic word embeddings based on their surrounding context and can be trained specifically for different languages and domains [1,22]. Instead of Word2Vec vectors, similarities between LSA [5] representations were also used for unsupervised text classification. Furthermore, DocSCAN uses Semantic Clustering by Adopting Nearest-Neighbors of text representations for unsupervised text classification [25].

Similar to unsupervised text classification approaches, ZSL approaches also aim to classify instances of unseen classes. Unlike unsupervised approaches, however, ZSL approaches use annotated training data from seen classes to predict instances of unseen classes [26]. Although ZSL models are partly trained on annotated data, they do not require label information about the unseen target classes for prediction and are therefore often considered equivalent to unsupervised approaches [20]. Jointly embedded document, label and word representations were also used in 0SHOT-TC to learn a ranking function for multi-label classification [14]. Additionally, different kinds of semantic knowledge (word embeddings, class descriptions, class hierarchy, and a general knowledge graph) were used for 0SHOT-TC [31], while other approaches tackle the task in a semi-supervised self-training approach [29] or treat 0SHOT-TC as entailment problem [30].

6 Conclusion

In this work, we showed how to effectively use Lbl2Vec for unsupervised text classification. Our experiments demonstrated that Lbl2Vec performs significantly better than other approaches in classifying text documents of unseen classes. Furthermore, our Lbl2Vec hyperparameter analysis indicates, that a minimum d_{min} value of approximately 60% of the average number of documents per class yields good label vectors for classification. In addition, using more accurate keywords can improve the classification performance of similarity-based text classification approaches such as Lbl2Vec. Future work can examine the use of keyphrase extraction approaches [21], knowledge graphs [17], or Masked Language Models (MLMs) for keyword generation. Thereby, a pretrained MLMs predicts what words can replace the class names under most contexts [12]. Usually, the top-50 predicted words have a similar meaning to the masked class name [12]. Therefore, using the top-50 predicted words as label keywords holds the potential to automate the keyword definition process and further improve F1-scores of similarity-based text classification approaches.

References

1. Braun, D., Klymenko, O., Schopf, T., Kaan Akan, Y., Matthes, F.: The language of engineering: training a domain-specific word embedding model for engineering. In: 2021 3rd International Conference on Management Science and Industrial Engineering, MSIE 2021,

pp. 8–12. Association for Computing Machinery, New York (2021). https://doi.org/10.1145/3460824.3460826

2. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LoF: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD 2000, pp. 93–104. Association for Computing Machinery, New York (2000). https://doi.org/10.1145/342009.335388
3. Chang, M.W., Ratinov, L.A., Roth, D., Srikumar, V.: Importance of semantic representation: dataless classification. In: AAAI, pp. 830–835 (2008). https://www.aaai.org/Library/AAAI/2008/aaai08-132.php
4. Chen, X., Xia, Y., Jin, P., Carroll, J.: Dataless text classification with descriptive LDA. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI 2015, pp. 2224–2231. AAAI Press (2015). https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9524
5. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. **41**, 391–407 (1990). https://cis.temple.edu/vasilis/Courses/CIS750/Papers/deerwester90indexing_9.pdf
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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), Minneapolis, Minnesota, pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/N19-1423
7. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In: Proceedings of the 20th International Joint Conference on Artifical Intelligence, IJCAI 2007, San Francisco, CA, USA, pp. 1606–1611. Morgan Kaufmann Publishers Inc. (2007). https://www.ijcai.org/Proceedings/07/Papers/259.pdf
8. Haj-Yahia, Z., Sieg, A., Deleris, L.A.: Towards unsupervised text classification leveraging experts and word embeddings. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp. 371–379. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/P19-1036. https://aclanthology.org/P19-1036
9. Lang, K.: Newsweeder: learning to filter netnews. In: Proceedings of the 12th International Machine Learning Conference (ML 1995) (1995)
10. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Xing, E.P., Jebara, T. (eds.) Proceedings of the 31st International Conference on Machine Learning. Proceedings of Machine Learning Research, Bejing, China, vol. 32, pp. 1188–1196. PMLR (2014). https://proceedings.mlr.press/v32/le14.html
11. Li, Y., Zheng, R., Tian, T., Hu, Z., Iyer, R., Sycara, K.: Joint embedding of hierarchical categories and entities for concept categorization and dataless classification. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, pp. 2678–2688. The COLING 2016 Organizing Committee (2016). https://aclanthology.org/C16-1252
12. Meng, Y., et al.: Text classification using label names only: a language model self-training approach. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 9006–9017. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.emnlp-main.724
13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 26. Curran Associates, Inc. (2013). https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf

14. Nam, J., Mencía, E.L., Fürnkranz, J.: All-in text: learning document, label, and word representations jointly. In: AAAI Conference on Artificial Intelligence (2016). https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12058
15. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, pp. 3982–3992. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/D19-1410. https://aclanthology.org/D19-1410
16. Sappadla, P.V., Nam, J., Mencia, E.L., Fürnkranz, J.: Using semantic similarity for multi-label zero-shot classification of text documents. In: Proceedings of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (2016). https://www.esann.org/sites/default/files/proceedings/legacy/es2016-174.pdf
17. Schneider, P., Schopf, T., Vladika, J., Galkin, M., Simperl, E., Matthes, F.: A decade of knowledge graphs in natural language processing: a survey. In: Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing, pp. 601–614. Association for Computational Linguistics (2022). https://aclanthology.org/2022.aacl-main.46
18. Schopf, T., Braun, D., Matthes, F.: Lbl2Vec: an embedding-based approach for unsupervised document retrieval on predefined topics. In: Proceedings of the 17th International Conference on Web Information Systems and Technologies - WEBIST, pp. 124–132. INSTICC, SciTePress (2021). https://doi.org/10.5220/0010710300003058
19. Schopf, T., Braun, D., Matthes, F.: Lbl2Vec github repository (2021). https://github.com/sebischair/Lbl2Vec
20. Schopf, T., Braun, D., Matthes, F.: Evaluating unsupervised text classification: zero-shot and similarity-based approaches. In: 2022 6th International Conference on Natural Language Processing and Information Retrieval (NLPIR), NLPIR 2022. Association for Computing Machinery, New York (2023)
21. Schopf, T., Klimek, S., Matthes, F.: 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 - KDIR, pp. 243–248. INSTICC, SciTePress (2022). https://doi.org/10.5220/0011546600003335
22. Schopf, T., Weinberger, P., Kinkeldei, T., Matthes, F.: Towards bilingual word embedding models for engineering. In: 2022 4th International Conference on Management Science and Industrial Engineering, MSIE 2022. Association for Computing Machinery, New York (2022). https://doi.org/10.1145/3535782.3535835
23. Song, Y., Roth, D.: On dataless hierarchical text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28, no. 1 (2014). https://ojs.aaai.org/index.php/AAAI/article/view/8938
24. Song, Y., Upadhyay, S., Peng, H., Roth, D.: Cross-lingual dataless classification for many languages. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, pp. 2901–2907. AAAI Press (2016). https://www.ijcai.org/Proceedings/16/Papers/412.pdf
25. Stammbach, D., Ash, E.: DocSCAN: unsupervised text classification via learning from neighbors. arXiv abs/2105.04024 (2021). https://arxiv.org/abs/2105.04024
26. Wang, W., Zheng, V.W., Yu, H., Miao, C.: A survey of zero-shot learning: settings, methods, and applications. ACM Trans. Intell. Syst. Technol. **10**(2) (2019). https://doi.org/10.1145/3293318
27. Williams, A., Nangia, N., Bowman, S.: A broad-coverage challenge corpus for sentence understanding through inference. In: Proceedings of the 2018 Conference of the North

American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, Louisiana, pp. 1112–1122. Association for Computational Linguistics (2018). https://doi.org/10.18653/v1/N18-1101. https://aclanthology.org/N18-1101

28. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V.: XLNet: Generalized Autoregressive Pretraining for Language Understanding. Curran Associates Inc., Red Hook (2019)
29. Ye, Z., et al.: Zero-shot text classification via reinforced self-training. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 3014–3024. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.acl-main.272. https://aclanthology.org/2020.acl-main.272
30. Yin, W., Hay, J., Roth, D.: Benchmarking zero-shot text classification: datasets, evaluation and entailment approach. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, pp. 3914–3923. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/D19-1404. https://aclanthology.org/D19-1404
31. Zhang, J., Lertvittayakumjorn, P., Guo, Y.: Integrating semantic knowledge to tackle zero-shot text classification. 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), Minneapolis, Minnesota, pp. 1031–1040. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/N19-1108. https://aclanthology.org/N19-1108
32. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 28. Curran Associates, Inc. (2015). https://proceedings.neurips.cc/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf
33. Zhang, Y., Meng, Y., Huang, J., Xu, F.F., Wang, X., Han, J.: Minimally Supervised Categorization of Text with Metadata, pp. 1231–1240. Association for Computing Machinery, New York (2020). https://doi.org/10.1145/3397271.3401168