

Data Scarcity: Methods to Improve the Quality of Text Classification

Ingo Glaser¹ ^a, Shabnam Sadegharmaki¹, Basil Komboz² and Florian Matthes¹

¹Chair of Software Engineering for Business Information Systems, Technical University of Munich, Boltzmannstrasse 3, 85748 Garching bei München, Germany

²Allianz SE, Munich, Germany

Keywords: Data Scarcity, Natural Language Processing, Text Classification, Legal Text Analytics.

Abstract: Legal document analysis is an important research area. The classification of clauses or sentences enables valuable insights such as the extraction of rights and obligations. However, datasets consisting of contracts or other legal documents are quite rare, particularly regarding the German language. The exorbitant cost of manually labeled data, especially in regard to text classification, is the motivation of many studies that suggest alternative methods to overcome the lack of labeled data.

This paper experiments the effects of text data augmentation on the quality of classification tasks. While a large amount of techniques exists, this work examines a selected subset including semi-supervised learning methods and thesaurus-based data augmentation. We could not just show that thesaurus-based data augmentation as well as text augmentation with synonyms and hypernyms can improve the classification results, but also that the effect of such methods depends on the underlying data structure.

1 INTRODUCTION


With the burst of available textual data, automation of certain processes in various areas, such as advertisement, risk evaluation, or translation, is becoming more and more attractive. As a result, text classification became one of the essential tasks in natural language processing (NLP) and knowledge discovery. Classification as a supervised learning (SL) technique has been applied widely in different areas such as language modeling, sentiment analysis, topic modeling, and named entity recognition (Allahyari et al., 2017). However, the other side of the coin is training data in certain domains. Supervised techniques have to acquire enough amount of labeled data to be able to generalize a model fitted to the labeled target. Hence, data is not equal to training data. The dominant source of these annotated training data is human experts. However, it is not achievable easy to create annotated corpora. The process of annotating is time-consuming, expensive, and more importantly error-prone. This challenge is even more relevant when it comes to deep learning (DL) techniques that require labeled data on a massive scale.

Many studies are addressing this challenge. As an instance, the lack of training data was the primary mo-

tivation behind the advent of semi-supervised learning (SSL) methods. The most basic approach, self-training (ST) was introduced back in 1960s (Chapelle et al., 2006). Recently, graph-based SSL approaches have gained popularity due to the flexibility and ease of interpretation (Sawant and Prabukumar, 2018). Also, state-of-the-art literature considers multi-instance or transfer learning (Cheplygina et al., 2019). However, domains such as the legal domain rely on a vast amount of domain knowledge and as a result, require extensive feature engineering. Furthermore, for these domains an explainable classification result is crucial. The goal of this paper is to not just support the traditional domains suited for NLP, but also these highly specific domains. Hence, the scope of this paper is limited to ST, label propagation (LP), and thesaurus-based data augmentation as traditional machine learning (ML) techniques.

As mentioned, many studies address the problem of data scarcity, but yet there is no state-of-the-art way to overcome it. This leads to our hypotheses behind this paper: *The effect of methods to improve classification tasks despite data scarcity depends on the characteristics of the underlying dataset.*

For that reason, three different scenarios in German text classification have been considered: (1) classification of economic news, (2) classification of le-

^a <https://orcid.org/0000-0002-5280-6431>

gal norms and regulations, and (3) classification of tweets.

The remainder of the paper is structured as follows: Section 2 provides a short overview of the related work, the experimental setup along with the used datasets are discussed in Section 3, finally, the approaches and its performance is evaluated in Section 4 before Section 5 closes with a conclusion and outlook.

2 RELATED WORK

Data scarcity is one of the most important obstacles in many research areas, involving SL, but also in particular concerning real-world problems. A vast amount of approaches to overcome this hurdle exist. These can be divided into the categories (1) SSL, (2) data augmentation, (3) multi-instance learning, and (4) transfer learning.

Each one of them addresses a specific problem. SSL techniques affect the algorithm directly by enabling it to consume unlabeled data as well as labeled data. Data augmentation, on the other hand, transforms and expands the data even before feeding it to the algorithm. Multi-instance learning enables the utilization of labels for a bag of instances instead of each one separately. Transfer learning can apply the knowledge in another domain with enough samples to process the domain with less training data. As already briefly touched in the introduction, this paper focuses on traditional ML approaches and thus only investigates SSL and data augmentation.

The following sections describe relevant related work.

2.1 Semi-supervised Learning

Through SSL, both labeled and unlabeled data are feed to the learning algorithm. The main idea behind it is the fact that, despite scarce labeled data, there is a large amount of unlabeled data available for many applications (Zhu and Goldberg, 2009). In the following, the most popular techniques in the SSL paradigm and its applications in NLP are discussed.

2.1.1 Self-training

ST is the most common technique in SSL. In this approach, first, a prediction model is learned based on available labeled data. The model then is used to predict the unlabeled data. These pseudo-labeled data alongside the original labeled ones will be later fed to a new model to be retrained. If the second

model is different from the base one, it is also called co-training (Zhu, 2005). Various approaches in self-training differ in the selection of these pseudo-labeled data. As an instance of text classification, (Pavlinek and Podgorelec, 2017) applied a threshold on the results to filter the more confident labels for the next round of training.

2.1.2 Label Propagation

Among SSL methods, graph-based approaches gained popularity recently because of their scalability but with the cost of higher complexity (Sawant and Prabukumar, 2018). In graph-based SSL, labeled and unlabeled data are represented as vertices in a weighted graph, with edge weights encoding the similarity between instances (Zhu et al., 2003). Labeling is done by smooth regularization of these weights in a process called LP.

The graph is constructed in two steps: (1) the adjacency matrix is constructed based on the k-nearest neighbor with radius ϵ , and (2) the weight of each edge is calculated by similarity functions such as gaussian or the inverse Euclidean distance function. In the next step, the classification problem can be represented as optimization of the normalized graph laplacian (Zhou et al., 2004).

2.1.3 Semi-supervised Learning in Text Classification

Text classification is the task of assigning a category to a sentence or document. These categories vary over many applications such as automatic email reply (Kannan et al., 2016), news classification (Howard and Ruder, 2018), question answering (Cer et al., 2018), or sequence modeling (Clark et al., 2018) among others.

SSL has hosted many novel pieces of research in NLP. ST, for instance, has been applied widely in language modeling techniques such as part-of-speech tagging and parsing (McClosky et al., 2006). Besides, (Pavlinek and Podgorelec, 2017) applied ST for increasing the training data size which improved the performance of text classification. However, some papers doubted the fact that self-training can be helpful as the errors are amplified in each iteration (Clark et al., 2003).

On the other hand, SSL had been part of the state-of-the-art classifiers in different applications. Johnson and Zhang (Johnson and Zhang, 2016) exploited unlabeled data to categorize texts by driving the region embeddings from an LSTM network. LP also has been shown to be effective in sentiment analysis (Yang and Shafiq, 2018). Moreover, Google's

smart reply project takes advantage of LP in an automatic email reply (Kannan et al., 2016). Another application is the classification of legal data. (Waltl et al., 2017) applied active machine learning (AML) to approach legal norm classification. (Savelka et al., 2015) utilized AML for the analysis of statutes.

2.2 Data Augmentation

Data augmentation techniques have addressed the problem of a lack of labeled data as well. The terminology comes originally from image processing, where more data can be crafted by adding noise or transforming existing images (Perez and Wang, 2017).

To adapt this definition to text, given a text or sentence, a variation of the text is created without affecting the meaning. The first hurdle is that a meaning of a text is rather subjective and therefore hard to train. Hence this technique has not been applied in NLP as extensively as image or signal processing. However, there are some breakthroughs recently, such as (Wang and Yang, 2015), who proposed a novel data augmentation approach.

The ideal way of varying a text can be paraphrasing, but it is a labor-intensive task. One alternative is replacing the words with synonyms or similar words, either using a thesaurus (Zhang and LeCun, 2015) or embeddings (Miyato et al., 2016).

(Sun and He, 2018) have introduced multi-granular data augmentation for sentiment analysis by incorporating synonyms and word vectors as word level and also some transformation for phrase and sentence-level. The synonyms often are derived from a thesaurus such as WordNet (Fellbaum, 2010). Unlike WordNet, the German version, GermaNet (Hamp and Feldweg, 1997; Henrich and Hinrichs, 2010) is not open-source and requires a licence. (Zhang and LeCun, 2015), who introduced a random selection algorithm for replacing a synonym. Another nice discussion about data augmentation for NLP has been made most recently by Wei and Zou (Wei and Zou, 2019).

3 EXPERIMENTAL SETUP

3.1 Objective

As already briefly touched in the introduction, we assume that the effect of methods to overcome data scarcity depends on the respective dataset. Therefore, we utilize three different datasets with varying characteristics. However, when talking about data scarcity, it

Table 1: Distribution of labels in the LN dataset.

Semantic type	Occurrences	Rel occur. (%)
Duty	117	19
Indemnity	8	1
Permission	148	25
Prohibition	18	3
Objection	98	16
Continuation	21	3
Consequence	117	19
Definition	18	3
Reference	56	9

can be distinguished between two different problems. (1) the label problem, and (2) the data problem. While in the former case enough data is existent, but just labels are missing, the latter problem even misses sufficient data instances. This paper investigates methods to overcome both problems on different datasets.

3.2 Data

As mentioned, three diverse datasets were used to show, that a generalization of the effects of the examined methods is not possible. The remainder of this Subsection deals with the utilized data.

3.2.1 Legal Norms (LN)

This dataset has been introduced in (Glaser et al., 2018). It contains 601 sentences of the German tenancy law which were manually labeled according to a taxonomy, constituting 9 semantic types. Table 1 shows the distribution of the different semantic types. For more information about the legal definition of these semantics, please have a look at (Waltl et al., 2019).

As representation for formal and technical German sentences, this dataset has been used. In other words, exactness in the meaning of technical words plays an essential role in this case. Moreover, the classification of semantics in the LN dataset is a multi-class problem, while the next two datasets represent binary classification.

Pre-processing of LN. In terms of pre-processing, words were lemmatized, after their part-of-speech tags had been extracted using the spaCy library (Honibal and Montani, 2017). For example, the word "booked" is converted to the phrase "book_v". In the next step, the documents were transformed into numeric vectors. There are many techniques in this regard, such as TF-IDF, term-frequency, binary-frequency, or different embeddings. The utilization of a binary vectorizer leads to the best performance. This setup was used for further experiments. The

Table 2: Distribution of labels in the NB dataset.

Label	Occurrences	Rel occur. (%)
Critical	282	12%
Non-critical	1996	88%

model achieved the best result when binary-frequency was applied.

3.2.2 GermEval18: Offensive Tweets (GE18)

For the sake of analyzing the effect of different augmentation methods in the social network context, we have employed the GermEval-2018 dataset (Wiegand et al., 2018). It is a publicly available dataset of tweets in German with a binary label, providing the information whether the tweet contains offensive content. The authors offer two label sets for this purpose, a coarse-grained and a more fine-grained. In this paper, the coarse-grained label set was chosen. The dataset includes 5.009 tweets, whereof 1.688 are labeled as offensive. Due to its nature, this dataset contains informal short texts in comparison to the formal content of the LN dataset.

Pre-processing of GE18. Rule-based approaches in order to remove superfluous special characters, such as hashtags, the so-called mentions, or links have been removed or replaced. Afterward, the same pre-processing steps from the above were applied.

3.2.3 News Bulletin (NB)

This is a private dataset provided by a big German insurance company, which contains 2.278 news regarding the German economy and industry. The dataset has been labeled manually by experts into whether it contains critically important information for the company or not. Being important is subject to different criteria such as target company, industry, and any other signals affecting the market of the companies insured by the insurance company. The frequency of labels is shown in Table 2.

Creating a model to extract critical news saves the cost and time of experts in insurance industries by reducing the risk of omission through crucial pieces of information. Moreover, this dataset is particularly interesting for the present research, because news involves long texts which are mostly edited and formalized in a standardized way. Hence, this dataset provides the opportunity to evaluate the methods of this paper on longer texts, too.

Preprocessing of NB. After the removal of special characters as well as links, the methods from the LN dataset were applied.

3.3 Experiments

We implemented all the experiments in this research in Python and scikit-learn (Pedregosa et al., 2011). The code will be published on Github.

3.3.1 Effect of Graph-based SSL

The first experiment aims to investigate how well graph-based SSL performs compared to classic SL methods on textual data. The experiment was designed by having different training sizes and consistent test size.

LP with regularization is implemented in scikit-learn by a function named *LabelSpreading*. This function follows the work of (Zhou et al., 2004) which suggested an affinity matrix based on the normalized graph Laplacian and soft clamping across the labels. There are two parameters that we tuned for each dataset. (1) the parameter of the RBF kernel defining how spread the decision region is (Gamma), and (2) the parameter which configures the label propagation and is the relative amount that an instance should adopt the information from its neighbors as opposed to its original label (Alpha).

3.3.2 Effect of Self-training SSL

For the second experiment, we investigated another approach in SSL, called ST. The goal of this experiment is to examine how much ST can compensate for the lack of training data in the textual context. To achieve this goal, each dataset was divided into three parts: (1) constant-size test set, (2) constant-size training set, and (3) variable-size augmented set (pseudo-labeled).

After the pre-processing steps, as described in Section 3.2, we implemented the self-training framework by means of scikit-learn models. Moreover, a custom k-fold validator was required to adapt to the implemented framework. Therefore, the evaluation is repeated five times, and then the average of F_1 score is reported as the performance of the model instead of employing a built-in cross validator. Moreover, a threshold was introduced to filter the pseudo-labels with confidence above it. The value of the threshold was tuned during the training process.

As a base of comparison, first, we evaluated the model without the presence of unlabeled data. Then we evaluated the highest cut-off possible. It means we fit the model by both augmented and training set with correct labels in order to omit the first model error. In other words, an ST model cannot achieve a better result than this cut-off.

In the next step, we incrementally increased the number of unlabeled data inserted to the next model. We investigated the hypothesis if a larger number of unlabeled data increases the performance of classification. Moreover, the effect of the threshold on the performance is reported.

3.3.3 Effect of Thesaurus-based Data Augmentation: Synonyms and Hypernyms

As mentioned in the previous chapter, data augmentation is coming from the area of the image processing where to create a more generalized model, different variations of an image, e.g. picture of an object from different angles, are added to the training set.

During text augmentation, for each training sample, the different variations are created by the replacement of words with their synonyms or hypernyms. The goal is to expand the training dataset to catch similar words around a topic. To achieve this goal, XML-based German synsets provided by GermaNet (Hamp and Feldweg, 1997) are employed. However, due to the licensing situation, it could not be applied to the NB dataset. Therefore, this experiment is tested and reported only on the two remaining datasets.

To better understand the effect of thesaurus-based data augmentation, the following example is considered:

The weather is nice, labeled as +.

The weather is awful, labeled as -.

Assuming a classification model has been trained with the sentences above, the polarity of the following sentence shall be predicted:

The weather is decent.

For simplification, let's assume the binary classification is determined by the cosine similarity between the binary vectors. In that case, the word "decent", which is not among the training vocabularies, is ignored by the binary vectorizer. As a result, the model determines the similarities incorrectly:

$$\cos_similarity(sent1,unseen) = 0.86$$

$$\cos_similarity(sent2,unseen) = 0.86$$

Using a synonym thesaurus, the training set can be augmented to include "decent" as a synonym of "nice". Most of the studies replace synonyms randomly and add new documents to the training. However, in the following experiment, all alternatives were compared. For better understanding, the simple example is expanded: Let "decent" stand as the synonym of "nice" and the adjective "bad" as the synonym of "awful", we get different possibilities for data augmentation. The remainder of this section describes these.

Horizontal Augmentation by Synonyms. In this case, the number of training data is consistent while the extra words are concatenated to the sentences. Notably, in this case, the unseen or test data should also be transformed. For example:

The weather is nice **decent**, +

The weather is awful **bad**, -

The weather is decent **nice**, ?

Consequently, the unlabeled sentence moves toward the correct label:

$$\cos_similarity(set1,unseen) = 1$$

$$\cos_similarity(sent2,unseen) = 0.6$$

It is essential to consider the feature space is increased. In reality, words have more than one synonym. This fact can degrade the similarity of close sentences. Moreover, synonym relations are not transitive. As an example, in WordNet, the word "nice" is a synonym of "decent" and "decent" is a synonym of "clean". However, "nice" does not count as the synonym of "clean". This fact can enforce concatenating irrelevant words. The following sentences are created by synonyms extracted from WordNet:

The weather is nice **decent good pleasant**, +

The weather is awful **bad**, -

The weather is decent **nice adequate modest**, ?

This example shows adding too many irrelevant words can have side effects for the similarity function:

$$\cos_similarity(set1,unseen) = 0.72$$

$$\cos_similarity(sent2,unseen) = 0.54$$

Besides, the increase of dimensionality can affect the assumptions made in ML algorithms which should be revisited. In the next section, we show how this method performs in the mentioned datasets.

Vertical Augmentation by Synonyms. As the second alternative, the original document remains intact, and combinations of synonyms for the word in the document are added to the training data. This approach is more related to the original concept of data augmentation.

The weather is nice, +

The weather is **decent**, +

The weather is awful, -

The weather is **bad**, -

The weather is decent, ?

Mainly, it generalizes the training data and is able to catch similar words. However, this approach has a significant downside when it comes to text processing. TF-IDF is a common technique for vectorizing the text. DF in the denominator normalizes the frequency of the words which repeats in different documents. This approach affects TF-IDF dramatically. In the above sample, for instance, "The weather is" is more likely to be degraded in the final vector. This is one of the reasons we utilized a binary vectorizer instead of TF-IDF.

Another important observation is that, in this case, the unlabeled data is not transformed. Still, the most critical challenge of this approach is the fact that all combination of synonyms of the words implies a vast number of variations. Following the work of (Zhang and LeCun, 2015), we introduced a parameter n_random which selects a specific number of variations.

Let's assume each document has $|W|$ number of words and each one has in average N_{syn} number of synonyms. Hence, there are $|W| * N_{syn}$ varieties for each document. Therefore, n_random of these combinations is selected. In the next step, the same label of the original sentence is assigned to these augmented set.

The last consequence is the fact that we can not easily apply cross validators to the augmented training dataset. Because an augmented version of a document should not appear in the test set. Otherwise, it results in over-fitting.

Generalization with Hypernyms. Another possibility to transform the text data is the utilization of hypernyms. GermaNet offers a similar structure as synonyms. Instead of co-meaning, they represent the generalization or abstract version of a word. As an example, the word "color" is the hypernym for the words "red", "blue", and "green". In the previous example, assuming *pos_adj* would be the hypernym for "nice" and *neg_adj* to be the hypernym for "awful" then the transformed data will look like:

```
The weather is pos_adj, +
The weather is neg_adj, -
The weather is pos_adj, ?
```

The disadvantages of the latter approaches are less relevant here. Yet, hypernyms are not transitive and therefore can increase the chance of adding irrelevant words. Although, compared to the synonyms, fewer words will be added to the feature space.

For the sake of implementation, we developed a transformer to add the respective words given the original data. The transformer searches the synset

structure in GermaNet and finds the most probable synset of the word from which synonyms and hypernyms are extracted. Moreover, in the case of vertical augmentation, it selects n variation of sentences by randomly combination synonyms and hypernyms. For extraction of the synonyms though, we repeated this process to find synonyms of synonyms and therefore extend the possible alternatives. This helps the method by increasing the chance that two similar words have enough common synonyms. However, there is a trade-off between enforcing irrelevant words and an increasing number of common synonyms between two words. Finally, the same process as the previous approaches is taken into account to transform the text documents into binary vectors.

4 EVALUATION

4.1 Results of Graph-based SSL

To evaluate the performance of classifiers, a 5-fold cross validator was employed, while the data was shuffled and then split by a stratified method to ensure the ratio of the labels is intact.

Designed the experiment as mentioned in Section 3, the results are shown in Figure 1. For each dataset, the results of LP are compared to a supervised method, linear support vector classification, or logistic regression, implemented by scikit-learn. The tuned parameters for each set of data are shown in Table 3. Logistic regression has been chosen for the NB dataset to be comparable with the previous results, which were achieved internally on the NB dataset. Nevertheless, logistic regression and linear support vector classification share a similar optimization function which yields to the same hyperplanes as the solution, and therefore our results are comparable.

The results show, by increasing the training size, both SSL and SL performances improve. Moreover, at some point, the amount of data does not add any information to the classification problem, and therefore the performance reaches a ceiling. Comparing SSL and SL, the results do not show a clear superiority of LP over the linear models. Only in NB, it shows a marginally increase in performance. Moreover, we observed that the LP technique is susceptible to the configuration of parameters, despite the linear models.

4.2 Results of Self-training

Figure 2 compares the performance of ST in the different datasets given the various number of unlabeled

Table 3: Comparison of F_1 between SSL and SL.

Labeled	NB		LN		GermEval18	
%	LogReg	LP	L SVC	LP	L SVC	LP
100	0.75	0.7	0.81	0.73	0.71	0.59
50	0.67	0.63	0.75	0.63	0.7	0.58
25	0.56	0.58	0.69	0.51	0.66	0.59
12.5	0.4	0.58	0.58	0.49	0.64	0.56
Tuned Params	C=35.38	$\gamma=30$ $\alpha=0.7$	C=1	$\gamma=10$ $\alpha=0.2$	C=1	$\gamma=20$ $\alpha=0.2$

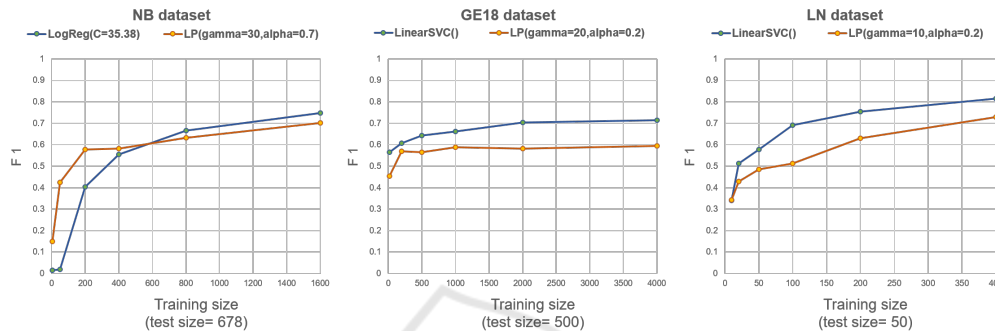


Figure 1: Effect of SSL LP by increasing the training size.

data. The solid yellow line shows the performance of the model without considering any unlabeled data. The dashed yellow lines show maximum performance that the model can reach assuming all data is labeled. The right axis, as well as the lines, show the performance of self-training and the left axis and the bars are showing the gradual increase of unlabeled data while the base labeled remains intact.

The result shows in presence of a threshold, ST boosts the performance. That is a very positive result, as usually there is a large number of unlabeled data available in different applications, which could be used for training as well now. Interestingly, ST improves the performance of each dataset.

4.3 Results of Data Augmentation

Table 4 shows the performance of text classification on two datasets, LN and GE18, compared to the different augmentation techniques. The horizontal technique using synonyms performed poorly in comparison to the other techniques and decreased the performance. This was expected as we discussed it in the previous section. It must be noted that in the horizontal synonym method, despite the horizontal hypernym, new data is not transformed. However, for hypernyms, we have to transform the new sets as the categories are not necessarily meaningful word units.

The other techniques, on the other hand, could increase the performance slightly in the LN dataset. However, they are competing closely, and they are not

showing any better results for the GE18. This is a larger dataset that provides one explanation. Furthermore, the data in social networks is rather diverse and informal, whereas the news data as well as legal norms constitute more formal data.

Table 4: Effect of different methods of data augmentation on the F_1 in text classification.

DA	LN	GermEval18
total # data	601	5009
% training	0.9	0.8
original	0.819	0.736
syn. horizontal	0.799	0.734
hypernym horizontal	0.822	0.734
syn. vert. random 5	0.841	0.728
syn. vert. random 10	0.837	0.726

5 CONCLUSION & OUTLOOK

This work examined the effects of methods to improve the quality of text classification despite a lack of data. We divided the issue in two distinct problem: (1) the label problem, and (2) the data problem. For the former problem, we investigated the application of SSL in different datasets. The latter problem was tackled by means of data augmentation.

We could show, LP, although promising, cannot improve the performance in either dataset. This method is very sensitive to the parameters and noises

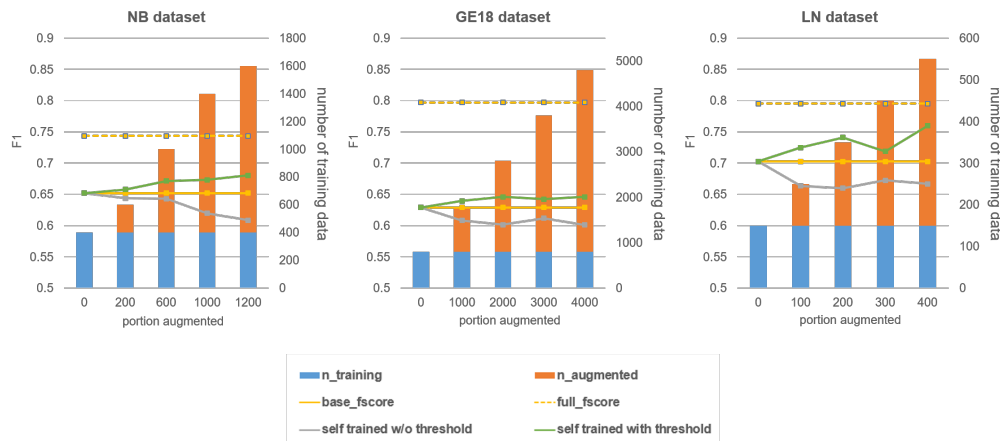


Figure 2: Effect of ST on the three datasets.

compared to classical linear models. Besides, we showed ST with consideration of a threshold can increase the performance and enables the model to take advantage of a vast number of unlabeled data. On the other hand, a self- or co-training method without a threshold has undoubtedly a negative impact.

Utilizing thesaurus-based data augmentation, a new variation of documents is created by replacing synonyms or hypernyms. Our experiments revealed, that data augmentation can be useful only in formal contexts. Furthermore, the experiment with horizontal data augmentation shows, that it was enforcing more irrelevant data which caused a negative impact on the classification. Finally, we could show that text augmentation with both, synonyms and hypernyms, can slightly improve the classification performance. However, the parameters must be fitted specific to each application and dataset. Also, it is essential to note that vertical data augmentation affects the vectorizer technique. TF-IDF as an instance has an adverse effect on the words which do not have a synonym. Hence, the augmentation techniques should be applied with binary vectorization.

Last but not least, the varying results observed during this work confirm the initial hypotheses. Methods to overcome data scarcity depend a lot on the characteristics of the used dataset.

REFERENCES

- Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., and Kochut, K. (2017). A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919*.
- Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan,

- S., Tar, C., et al. (2018). Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.
- Chapelle, O., Schölkopf, B., and Zien, A. (2006). *Semi-Supervised Learning*. MIT Press, London, England.
- Cheplygina, V., de Bruijne, M., and Pluim, J. P. (2019). Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical Image Analysis*.
- Clark, K., Luong, M.-T., Manning, C. D., and Le, Q. V. (2018). Semi-supervised sequence modeling with cross-view training. *arXiv preprint arXiv:1809.08370*.
- Clark, S., Curran, J. R., and Osborne, M. (2003). Bootstrapping pos taggers using unlabelled data. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 49–55. Association for Computational Linguistics.
- Fellbaum, C. (2010). Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer.
- Glaser, I., Scepankova, E., and Matthes, F. (2018). Classifying semantic types of legal sentences: Portability of machine learning models. In *Proceedings of the 28th Annual Conference on Legal Knowledge and Information Systems (JURIX'15)*, Groningen, The Netherlands.
- Hamp, B. and Feldweg, H. (1997). Germanet-a lexical-semantic net for german. *Automatic information extraction and building of lexical semantic resources for NLP applications*.
- Henrich, V. and Hinrichs, E. (2010). Gernedit-the germanet editing tool. *Proceedings of the ACL 2010 System Demonstrations*, pages 19–24.
- Honnibal, M. and Montani, I. (2017). spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.
- Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.
- Johnson, R. and Zhang, T. (2016). Supervised and semi-supervised text categorization using lstm for region embeddings. *arXiv preprint arXiv:1602.02373*.

- Kannan, A., Kurach, K., Ravi, S., Kaufmann, T., Tomkins, A., Miklos, B., Corrado, G., Lukacs, L., Ganea, M., Young, P., et al. (2016). Smart reply: Automated response suggestion for email. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 955–964. ACM.
- McClosky, D., Charniak, E., and Johnson, M. (2006). Effective self-training for parsing. In *Proceedings of the main conference on human language technology conference of the North American Chapter of the Association of Computational Linguistics*, pages 152–159. Association for Computational Linguistics.
- Miyato, T., Dai, A. M., and Goodfellow, I. (2016). Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.
- Pavlinek, M. and Podgorelec, V. (2017). Text classification method based on self-training and lda topic models. *Expert Systems with Applications*, 80:83–93.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Perez, L. and Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.
- Savelka, J., Trivedi, G., and Ashley, K. D. (2015). Applying an interactive machine learning approach to statutory analysis. In *Proceedings of the 28th Annual Conference on Legal Knowledge and Information Systems (JURIX'15)*. IOS Press.
- Sawant, S. S. and Prabukumar, M. (2018). A review on graph-based semi-supervised learning methods for hyperspectral image classification. *The Egyptian Journal of Remote Sensing and Space Science*.
- Sun, X. and He, J. (2018). A novel approach to generate a large scale of supervised data for short text sentiment analysis. *Multimedia Tools and Applications*, pages 1–21.
- Waltl, B., Bonczek, G., Scepankova, E., and Matthes, F. (2019). Semantic types of legal norms in german laws: classification and analysis using local linear explanations. *Artificial Intelligence and Law*, 27(1):43–71.
- Waltl, B., Muhr, J., Glaser, I., Bonczek, G., Scepankova, E., and Matthes, F. (2017). Classifying legal norms with active machine learning. In *Proceedings of the 28th Annual Conference on Legal Knowledge and Information Systems (JURIX'15)*, pages 11–20.
- Wang, W. Y. and Yang, D. (2015). That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2557–2563.
- Wei, J. and Zou, K. (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.
- Wiegand, M., Siegel, M., and Ruppenhofer, J. (2018). Overview of the germeval 2018 shared task on the identification of offensive language. In *14th Conference on Natural Language Processing KONVENS 2018*.
- Yang, Y. and Shafiq, M. O. (2018). Large scale and parallel sentiment analysis based on label propagation in twitter data. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (Trust-Com/BigDataSE)*, pages 1791–1798. IEEE.
- Zhang, X. and LeCun, Y. (2015). Text understanding from scratch. *arXiv preprint arXiv:1502.01710*.
- Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. (2004). Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328.
- Zhu, X., Ghahramani, Z., and Lafferty, J. D. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919.
- Zhu, X. and Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130.
- Zhu, X. J. (2005). Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.