Job Title Classification Strategies for the German Labor Market

Masterthesis

submitted by

Rahkakavee Baskaran

at the

Universität Konstanz

Department of Politics and Public Administration

Center for Data and Methods

1.Gutachter: Prof. Dr. Susumu Shikano

2.Gutachter: JunProf Juhi Kulshresthra

Konstanz, December 1, 2021

Contents

Abbreviations

1 Introduction

2 Related work

2.1 Textclassification

Text classification, a highly researched area, is the process of classifying text documents or text segments into a set of predefined classes. During the last decades, researches developed a various number of classifiers. As Kowsari et al. (2019) summarize in their survey of classifiers, we can group the approaches mainly into three groups. The first group contains traditional methods like Naive Bayes (NV), Support Vector Machines (SVM), K-nearest neighbors (KNN), Logistic Regressions (LR) or Decision Trees (DT) (Vijayan et al., 2017; Colas and Brazdil, 2006; Kowsari et al., 2019; Sebastiani, 2001). Deep learning methods like Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN), which are currently the most advanced algorithms for NLP, form the second group. The last group consists of ensemble learning techniques like Boosting and Bagging.

2.2 Short Text classification

Another potential issue is the length of input documents for classification. Job titles are clearly short text with often not more than 50 characters. Short texts suffer from sparseness, few word co-occurrences, missing shared context, noisiness and ambiguity. Traditional methods, however, are based on word frequency, high word co-occurrence and context, which is why they often fail to achieve high accuracy for short texts (Song et al., 2014; Wang et al., 2017, 2014). In their overview, Song et al. (2014) present three approaches to solve this. First, since short text data often suffers from unlabeled data in the context of online text data, such as Twitter postings, they suggest using semi-supervised approaches. Second, they recommend to use ensemble learning methods, which focus on the sparseness of the data. Third, Song et al. (2014) propose feature dimensionality reduction and extraction of semantic relationship methods. Based on the latter more recent work on short text classification criticizes the use of the "Bag of Word" concept for feature representation as it only reflects the appearance of words in the text. Instead, they represent short texts with semantically similar and conceptual information (Bouaziz et al., 2014; Wang et al., 2014; Chen et al., 2019). Another question concerning the representation of short texts is whether to represent them as dense or sparse vectors. In their comparison between tf-idf/counter vectorizer and the dense vectorizer word2vec and doc2vec, Wang et al. (2017) conclude that among the classifiers Naive Bayes, Logisitic Regression and SVM, the sparse vectorizers achieve the highest accuracy. Chen et al. (2019), conversely, see limitations in sparse representation as it cannot capture the context information. In their work, they integrate sparse and dense representation into a deep neural network with Knowledge powered Attention, which outperform state-of-art deep learning methods, like CNN, for Chinese short texts. Concerning the classifiers, there is no consensus approach for short text classification. For traditional approaches Wang et al. (2017)'s results indicate that logistic regression and SVM perform best, while KNN seems to achieve best accuracy in Khamar (2013)'s work. Similar to job title specific work, more recent work prefers deep learning methods, mostly CNN (Chen et al., 2019).

2.3 Domain-specific

Each text classification task presents different challenges. One challenge is that domain-specific problems may arise. There is some work that deals with job classification in the English speaking job market. In terms of classifiers, the corresponding work can be categorised into traditional classifiers or deep learning methods. **?** for example, use a KNN classifier in combination with document embedding as feature selection strategy. **?** rely on traditional methods as well, by combining a SVM classifier and a KNN classifier for their job recommendation system. In contrast, the approaches of **?**, **?** and **?** are based on Deep Learning methods. From a higher perspective, there is another dividing line between the approaches. As mentioned earlier, job title normalization can be considered as a typical text classification task (**???**). **?** and **?**, however, formulate the task as a string representation approach of similar job titles.

2.4 Multiclass

A last challenge of text classification tasks comes with the number of classes. As **?** show in their classification of tissue, multiclass classification is more difficult than binary classification problems. Partly, because most of classification algorithms were designed for binary problems (**?**). Approaches for multiclassification can be grouped into two types. Binary algorithms can handle multiclassification naturally. This is, for example, the case for Regression, DT, SVM, KNN and NV. The second type is the decomposition of the problem into binary classification tasks (for the different subtypes see ?). The literature so far does not have a clear answer to solve multiclassification problems. Different approaches, like Boosting (?) or CNN (?) are applied. It is noticeable, however, that many works use variations of SVM (Guo and Wang, 2015; Tomar and Agarwal, 2015; Tang et al., 2019).

3 Data and taxonomy

3.1 KldB 2010 Taxonomy

The "Klassifikation der Berufe 2010 (KldB)" is structured hierarchically with 5 levels. On each level there is a different number of classes. On level 1 each class has an id of length one with a number from 0 to 9. Table 1 shows the 10 classes of level 1 with their class names. On level 2, then, each of the 10 classes are divided into one or more subclasses having a class id of length 2 with the first digit indicating the class of level 1 and the second digit the class of level 2. An overview of the all 5 levels with an example of classes is given in table 2. Note that the example in table 2 does not show on level 2 to level 5 all classes. Thus on level 2 there exists also, e.g. the class id 41 with "Mathematik-, Biologie- Chemie- und Physikberufe", which in turn is divided into other classes on level 3 etc.. With this procedure this ultimatley leads to class ids of length 5 on level 5. An occupation can be classified on every level in the Taxonomy. Considering the classes of the example in table 2, the job title "Java Developer" could be classified on level 5 to the class 43412. From this id, it is also derivable that the jobtitle belongs, for example, on level 3 to the class "Sofwareentwicklung" (Bundesagentur für Arbeit, 2011a,b; Paulus and Matthes, 2013)

The KldB contains of two dimension. The first dimension, the so-called "Berufsfachlichkeit" structures jobs according to their similarity in knowledge, activties and jobs. This is reflected in the first 4 levels. Considering the again the example from above and the job title "Fullstack PHP-Entwickler" it is reasonable to classify both on level 1 to "Naturwissenschaft, Geografie and Information", because both of them are related to computer science. It also make sense to classifiy them for example to 4341, because both are about sofware development. On level 5, then, a second dimension is introduced. the "Anforderungsniveau". This dimension gives information on the level of requirement for a job and 4 possible requirements. In table 3 they are summarized. From the class id

IDs KldB 2010	Berufsbereich (Level 1)
1	Land-, Forst- und Tierwirtschaft und Gartenbau
2	Rohstoffgewinnung, Produktion und Fertigung
3	Bau, Architektur, Vermessung und Gebäudetechnik
4	Naturwissenschaft, Geografie und Informatik
5	Verkehr, Logistik, Schutz und Sicherhe
6	Kaufmännische Dienstleistungen, Warenhandel, Vertrieb, Hotel und Tourismus
7	Unternehmensorganisation, Buchhaltung, Recht und Verwaltung
8	Gesundheit, Soziales, Lehre und Erziehung
9	Sprach-, Literatur-, Geistes-, Gesellschafts- und Wirtschaftswissenschaften, Medien, Kunst, Kultur und Gestaltung
0	Militär

Table 1: Overview of classes Level 1 - Berufsbereiche (edited after (Bundesagentur für Arbeit, 2011b))

of job title "Java Developer", we can see that the job has been assigned to the second requirement level, since the last digit is a 2 (Bundesagentur für Arbeit, 2011a,b; Paulus and Matthes, 2013)

Name	Level	Number of classes	Example
Berufsbereiche	1	10	4: Naturwissenschaft, Geografie und Informatik
Berufshauptgruppen	2	37	43: Informatik-, Informations- und Kommunikationstechnologieberufe
Berufsgruppen	3	144	434: Sofwareentwicklung
Berufsuntergruppen	4	700	4341: Berufe in der Sofwareentwicklung
Berufsgattungen	5	1286	43412: Berufe in der Sofwareenetwicklung - fachlich ausgerichtete Tätigkeiten

Table 2: Overview of KldB (edited after (Bundesagentur für Arbeit, 2011b))

With the KldB 2010, an useful and information-rich occupational classification was created for Germany that reflects the current trends in the labor market (Paulus and Matthes, 2013). One strength relies in the construction of the KldB. Instead of just including expert knowledge into the Taxonomy the developement process is based on systematical consideration of information about occupations, as well as statistical procedures for taxonomy developement. Furthermore, the taxonomy was reviewed qualitatively several times in relation to professions and revised. Considering the expressiveness, the KldB has some more benefits. Since the taxonmy is quite recent, it reflects new job classes and market trends very adequately. Further, by including the second dimension, the taxonomy provides a powerful tool to organize job titles into simple requirement classes. In addition, the taxonomy also distinguishes between managerial, supervisory, and professional employees, which is also valuable information. Finally, the taxonomy

Level of requirement	Class ID	Name long	Name short
1	xxxx1	Helfer- und Anlerntätigkeit	Helfer
2	xxxx2	fachlich ausgerichtete Tätigkeiten	Fachkraft
3	xxxx3	komplexe Spezialstentätigkeiten	Spezialist
4	xxxx4	hoch komplexe Tätigkeiten	Experte

Table 3: Overview of Level of requirements on Level 5
(edited after (Bundesagentur für Arbeit, 2011b))

also convinces with the possibility to switch to “International Standard Clasification of Occupations (ISCO)” through its IDS and thus to normalize jobs to a global standard (Bundesagentur für Arbeit, 2011b)

4 Pipeline

5 Preprocessing

6 Evaluation metrics

There exists several metrics for the evaluation of classification approaches in the literature (Fatourechi et al., 2008). The choice of appropriate measurements is a crucial step for obtaining a qualitative comparison in the performance between the baseline algorithms and the new approaches. Often researchers rely on popular metrics like overall accuracy (OA). However, especially for multiclass and inbalanced dataset tasks it is difficult to rely only on one measure like OA In order to select appropriate metrics for comparison in the following the most important metrics will be discussed focussing on multiclass classification and imbalanced data sets.

Most metrics rely on a confusion matrix. For the multiclass case this confusion matrix is defined as follows (Kautz et al., 2017):

	positive examples			
positive prediction	$c_{1,1}$	$c_{1,2}$	$\dots$	$c_{1,n}$
	$c_{2,1}$	$c_{i,j}$		
	$\vdots$		$\ddots$	$\vdots$
	$c_{n,1}$		$\dots$	$c_{n,n}$

Table 4: Confusion Matrix (edited after (Kautz et al., 2017, 113)

From the confusion matrix follows that $c_{i,j}$ defines examples which belong to class j and are predicted as class i. Given that k is the current class, True positives (TP) is defined as $tp_k = c_{k,k}$, thus examples which are correctly predicted as the current class k. False negatives (FN) are defined as those examples which not belonging to the current class k, but are predicted as k. Formally $fn_k = \sum_{i=1,i\neq k}^{n} c_{i,k}$. Next, True negatives (TN), are examples belonging to the current class m, but are not predicted as m. Formally $tn_k = \sum_{i=1,i\neq k}^{n} \sum_{j=1,j\neq k}^{n} c_{i,j}$. Last, False positives (FP) are defined as examples not belonging to class k, but are predicted as such. Formally this can be expressed as: $fp_k = \sum_{i=1, i\neq k}^{n} c_{k,i}$ (Kautz et al., 2017)

As mentioned the OA is one of most common metric for performance evaluation. It represents how well the classifier classifies across all classes correctly. Formally, given that N is number of examples and K the number of all classes, this can be expressed as (Branco et al., 2017):

$$OA = \frac{1}{K}\sum_{i=1}^{K}\frac{tp_k + tn_k}{N}$$

Following the formula an accuracy of 1 means that all examples are correctly classified, while a 0 mean that each example is classified with the wrong class. (Berthold et al., 2020) Although OA is a widley used metric it is critized for favouring the majority classes, thus not reflecting minority classes appropriatly in unbalanced datasets (Berthold et al., 2020; Fatourechi et al., 2008)

Two more popular metrics are precision and recall. Precision respresents how well the classifier detects actual positive examples among the positive predicted examples. Recall, also called sensitivity, in contrast, represents how many examples are labelled as positive among the actual positive examples (Berthold et al., 2020). For the multiclass scenario, two different calculation approaches for each of the metrics are proposed: micro and macro average (Branco et al., 2017). In the macro approach first the metric is calculated for each class k against all other classes. The average of all of them is built. Formally:

$$precision_{macro} = \frac{1}{K}\sum_{i=1}^{k}\frac{tp_i}{tp_i + fp_i}$$

$$recall_{macro} = \frac{1}{K}\sum_{i=1}^{k}\frac{tp_i}{tp_i + fn_i}$$

In contrast the micro approach aggregates the values, which can be formally expressed as follows:

$$precision_{micro} = \frac{\sum_{i=1}^{K} tp_i}{\sum_{i=1}^{K} tp_i + fp_i}$$

$$recall_{micro} = \frac{\sum_{i=1}^{K} tp_i}{\sum_{i=1}^{K} tp_i + fn_i}$$

There is a trade-off between precision and recall (Buckland and Gey, 1994). The F-measure capture both precision and recall by taking the harmonic mean between both.

It is calculated as follows (Branco et al., 2017; Pan et al., 2016):

$$F_{micro} = 2 \cdot \frac{precision_{micro} \cdot recall_{micro}}{precision_{micro} + recall_{micro}}$$

$$F_{macro} = 2 \cdot \frac{precision_{macro} \cdot recall_{macro}}{precision_{macro} + recall_{macro}}$$

Apart from the trade-off between recall and precision, there is also a tradeoff between sensitivity and specificty (1- sensitivity). Using a receiver operating characteristics (ROC), which plots the specifity against the sensitivity the trade-off can be visualized for different thresholds. The area under the curve then can be used to obtain the performance of the classifier. A large area indicates a better classifier (Berthold et al., 2020; Espíndola and Ebecken, 2005).

As shown above, there are several metrics for evaluating the performance of a classifier, with the metrics having different focuses. Since the job title classification involves multiclass classification and the descriptive analysis show that the data is clearly unbalanced, at least for some classes in level 5, it is not reasonable to base the evaluation solely on the OA. Taking precision, recall and the harmonic mean into account would capture the performance of the minority classes as well. The ROC curve does gives, due to its visualization a good impression for the performance, but it is not feasible for high number of classes. Following this argumentation the performance of the classifiers will be evaluated with accuracy, precision, recall, F-measure and Cohen's Kappa.

7 Baseline Algorithms

In order to compare different feature selection methods solid baselines are necessary. As pointed out in the literature review NB (NB), LR (LR) and Support Vector Machine (SVM) have several advantages for text classifcation tasks. In the following based on a theoretical discussion of each classifier, the exact modeling of the baseline classifiers is justified.

7.1 Naive Bayes Classifier

The NB, a family of probabilistic classifiers, uses Bayes' rule in order to determine the most likely class for each document (Schneider, 2005). All NB classifiers rely on the conditional independence assumptions which means, that "features are independent of each other, given the category variable" (Xu, 2018, 48). Depending on whether the features are discrete or continous, different distributions, so-called event models are proposed. While from a theoretical perspective for continous features Gaussian distribution is well-suited, for discrete features usually Bernoulli or multinomial distibutions are applied (Xu, 2018). Although, popular practical implementations, like the one from sklearn, allow as well fractional counts for multinomial distributions (**?**). Trying different event models, the multinomial NB shows indeed for both Count Vectorizer and for TFIDF the best results, which is why I choose it as event model for the baseline.

The multinomial NB classifies according to the most likely class. Given that a document d has $t = 1, ..., k$ terms and can be assigned to $c = 1, ..., j$ classes, the probability of a term in a document given a class is calculated as (Manning et al., 2008):

$$P(t_k, c_j) = \frac{occurence(t_k, c_j) + 1}{\sum occurence(t, c_j) + |V|}$$

where $|V|$ is the cardinality of the vocabulary. In the denominator 1 is added, so-called Laplace smoothing, in order to avoid zeros, which is the case if the number of terms in a document for one class is zero. (Manning et al., 2008). Further given that $N_c = count(c_j)$ is the number of documents belonging to class c_j and N is number of documents the probability of c_j is defined as $\frac{N_c}{N}$. The probability of a document d belonging to a class c_j can then formulated as follows (Manning et al., 2008, 258):

$$P(c_j|d) \propto P(c_j) \prod_{i=1}^{k} P(t_i|c_j)$$

Then the most likely classes can be determined by (Manning et al., 2008):

$$\arg\max_{c \in C} P(c_j) \prod_{i=1}^{k} P(t_i|c_j)$$

7.2 Logistic Regression

7.3 Support Vector Machines

However, SVM also performed well for text classification. Especially for multiclass tasks, as mentioned in the literature review, often different versions of the algorithm are used and showed good performance (Aiolli and Sperduti, 2005; Angulo et al., 2003; Benabdeslem and Bennani, 2006; Guo and Wang, 2015; Mayoraz and Alpaydm, 1999; Tang et al., 2019; Tomar and Agarwal, 2015). In general SVM has several advantages for text classifcation. First, text classifcation usually has a high dimensional input space. SVM can handle these large features since they are able to learn independently of the dimensionality of the feature space. In addition SVMs are known to perform well for dense and sparse vectors, which is usually the case for text classification (Joachims, 1998). Empirical results, for example Joachims (1998) or Liu et al. (2010) confirm the theoretical expectations. It is, therefore, a reasonable option to use a basic version of the SVM algorithm as a baseline.

The general idea of a SVM is to map "the input vectors x into a high-dimensional feature space Z through some nonlinear mapping chosen a priori [...], where an optimal separating hyperplane is constructed" (Vapnik, 2000, 138). In SVM this optimal hyperplane maximizes the margin, which is simply put the distance from the hyperplane to the closest points, so called Support Vectors, across both classes (Han et al., 2012). Formally, given a training data set with n training vectors $x_i \in R^n, i = 1, ..., n$ and the target classes $y_1, ...y_i$ with $y_i \in \{-1, 1\}$, the following quadratic programming problem (primal) has to be solved in order to find the optimal hyperplane:

$$\min_{w,b} \frac{1}{2} w^T w$$

$$\text{subject to } y_i(w^T \phi(x_i) + b) \geq 1$$

where $\phi(x_i)$ transforms x_i into a higher dimensional space, w corresponds to the weight and b is the bias (Chang and Lin, 2001; Jordan et al., 2006) The given optimzation function assumes that the data can be separated without errors. This is not always possible, which is why Cortes et al. (1995) introduce a soft margin SVM, which allows for missclassfication (Vapnik, 2000). By adding a regularization parameter C with $C > 0$ and the corresponding slack-variable ξ the optimization problem changes to (Chang and Lin, 2001; Han et al., 2012):

$$\min_{w,b} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i$$

$$\text{subject to } y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i,$$

$$\xi_i \geq, i = 1, ..., n$$

Introducing Lagrange multipliers α_i and converting the above optimization problem into a dual problem the optimal w meets (Chang and Lin, 2001; Jordan et al., 2006):

$$w = \sum_{I=1}^{n} y_i \alpha_i \phi(x_i)$$

with the decision function (Chang and Lin, 2001):

$$\text{sgn}\ (w^T \phi(x) + b) = sgn(\sum_{i=1}^{n} y_i \alpha K(x_i, x) + b)$$

$K(x_i, x)$ corresponds to a Kernel function, which allows to calculate the dot product in the original input space without knowing the exact mapping into the higher space (Han et al., 2012; Jordan et al., 2006).

In order to apply SVM to multiclass problems several approaches have been proposed. One stratetgy is to divide the multi-classifcation problem into several binary problems. A common approach here is the one-against-all method. In this method as many SVM classifiers are constructed as there are classes k. The k-th classifier assumes that the examples with the k label are positive labels, while all the other examples treated as negative. Another popular approach is the one-against-one method. In this approach $k(k-1)/2$ classifiers are constructed allowing to train in each classifier the data of two classes (Hsu and Lin, 2002). Besides dividing the multiclass problem into several binary problems, some researches propose approaches to solve the task in one single optimization problem, like Crammer and Singer (2001). [1].

In order to find a strong baseline I checked SVM's with different parameters for the SVM, as well as different multiclass approaches. It appears that a SVM using a soft margin with a $C = 1$ and a one-vs-rest approach has the best results. I also test different kernels, like RBF Kernel or linear kernel. The linear kernel, formally $k(x, x') = x^T x'$, achieved the best results, which is why I choose it for the baseline.

[1] For a detailed overview of all different methods and the method of Crammer and Singer (2001) see Hsu and Lin (2002); Crammer and Singer (2001)

8 Implementation of ...

9 Experimental results

10 Discussion and Limitations

References

Aiolli, F. and Sperduti, A. (2005). Multiclass classification with multi-prototype support vector machines. *Journal of Machine Learning Research*, 6:817–850.

Angulo, C., Parra, X., and Català, A. (2003). K-svcr. a support vector machine for multi-class classification. *Neurocomputing*, 55:57–77.

Benabdeslem, K. and Bennani, Y. (2006). Dendogram based svm for multi-class classification. *Proceedings of the International Conference on Information Technology Interfaces, ITI*, pages 173–178.

Berthold, M. R., Borgelt, C., Höppner, F., Klawonn, F., and Silipo, R. (2020). *Guide to Intelligent Data Science*. Springer, 2 edition.

Bouaziz, A., Dartigues-Pallez, C., da Costa Pereira, C., Precioso, F., and Lloret, P. (2014). Short text classification using semantic random forest. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8646 LNCS:288–299.

Branco, P., Torgo, L., and Ribeiro, R. P. (2017). Relevance-based evaluation metrics for multi-class imbalanced domains. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10234:698–710.

Buckland, M. and Gey, F. (1994). The relationship between recall and precision. *Journal of the American society for information science*, 45.

Bundesagentur für Arbeit, B., editor (2011a). *Klassifikation der Berufe 2010 Band 1: Systematischer und alphabetischer Teil mit Erläuterungen.*

Bundesagentur für Arbeit, B. (2011b). Klassifikation der berufe 2010 (kldb 2010) – aufbau und anwenderbezogene hinweise. Technical report.

Chang, C.-C. and Lin, C.-J. (2001). Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2.3:1–27.

Chen, J., Hu, Y., Liu, J., Xiao, Y., and Jiang, H. (2019). Deep short text classification with knowledge powered attention. *33rd AAAI Conference on Artificial Intelligence, AAAI 2019*, pages 6252–6259.

Colas, F. and Brazdil, P. (2006). Comparison of svm and some older classification algorithms in text classification tasks. *IFIP International Federation for Information Processing*, 217:169–178.

Cortes, C., Vapnik, V., and Saitta, L. (1995). Support-vector networks editor. *Machine Leaming*, 20:273–297.

Crammer, K. and Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292.

Espíndola, R. P. and Ebecken, N. F. F. (2005). On extending f-measure and g-mean metrics to multi-class problems. *WIT Transactions on Information and Communication Technologies*, 35:25–34.

Fatourechi, M., Ward, R. K., Mason, S. G., Huggins, J., Schlögl, A., and Birch, G. E. (2008). Comparison of evaluation metrics in classification applications with imbalanced datasets. *Proceedings - 7th International Conference on Machine Learning and Applications, ICMLA 2008*, pages 777–782.

Guo, H. and Wang, W. (2015). An active learning-based svm multi-class classification model. *Pattern Recognition*, 48:1577–1597.

Han, J., Kamber, M., and Pei, J. (2012). *Data Mining: Concepts and Techniques*. Elsevier Inc., 3 edition.

Hsu, C. W. and Lin, C. J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13:415–425.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *European Conference on Machine Learning*, pages 137–142.

Jordan, M., Kleinberg, J., and Schölkopf, B. (2006). *Pattern Recognition and Machine Learning*. Springer.

Kautz, T., Eskofier, B. M., and Pasluosta, C. F. (2017). Generic performance measure for multiclass-classifiers. *Pattern Recognition*, 68:111–125.

Khamar, K. (2013). Short text classification using knn based on distance function. *International Journal of Advanced Research in Computer and Communication Engineering*, 2:1916–1919.

Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L., and Brown, D. (2019). Text classification algorithms: A survey. *Information 2019*, 10:1–68.

Liu, Z., Lv, X., Liu, K., and Shi, S. (2010). Study on svm compared with the other text classification methods. *2nd International Workshop on Education Technology and Computer Science, ETCS 2010*, 1:219–222.

Manning, C., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval.* Cambridge University Press.

Mayoraz, E. and Alpaydm, E. (1999). Support vector machines for multi-class classification. *Lecture Notes in Computer Science*, 1607:833–842.

Pan, W., Narasimhan, H., Protopapas, P., Kar, P., and Ramaswamy, H. G. (2016). Optimizing the multiclass f-measure via biconcave programming. *IEEE 16th International Conference on Data Mining (ICDM)*, pages 1101–1106.

Paulus, W. and Matthes, B. (2013). Klassifikation der berufe : Struktur, codierung und umsteigeschlüssel. *FDZ Methodenreport.*

Schneider, K.-M. (2005). Techniques for improving the performance of naive bayes for text classification. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 682–693. Springer.

Sebastiani, F. (2001). Machine learning in automated text categorization. *ACM Computing Surveys*, 34:1–47.

Song, G., Ye, Y., Du, X., Huang, X., and Bie, S. (2014). Short text classification: A survey. *Journal of Multimedia*, 9:635–643.

Tang, L., Tian, Y., and Pardalos, P. M. (2019). A novel perspective on multiclass classification: Regular simplex support vector machine. *Information Sciences*, 480:324–338.

Tomar, D. and Agarwal, S. (2015). A comparison on multi-class classification methods based on least squares twin support vector machine. *Knowledge-Based Systems*, 81:131–147.

Vapnik, V. N. (2000). *The nature of statistical learning theory.* Springer.

Vijayan, V. K., Bindu, K. R., and Parameswaran, L. (2017). A comprehensive study of text classification algorithms. pages 1109–1113. Institute of Electrical and Electronics Engineers Inc.

Wang, F., Wang, Z., Li, Z., and Wen, J. R. (2014). Concept-based short text classification and ranking. *CIKM 2014 - Proceedings of the 2014 ACM International Conference on Information and Knowledge Management*, pages 1069–1078.

Wang, Y., Zhou, Z., Jin, S., Liu, D., and Lu, M. (2017). Comparisons and selections of features and classifiers for short text classification. *IOP Conference Series: Materials Science and Engineering*, 261:1–8.

Xu, S. (2018). Bayesian naïve bayes classifiers to text classification:. *Journal of Information Science*, 44:48–59.