ZHAW-InIT - Social Media Geolocation at VarDial 2020

Fernando Benites
benf@zhaw.ch
Zurich University of Applied Sciences, Switzerland

Manuela Hürlimann
hueu@zhaw.ch
Zurich University of Applied Sciences, Switzerland

Pius von Däniken
vode@zhaw.ch
Zurich University of Applied Sciences, Switzerland

Mark Cieliebak
mc@spinningbytes.com
SpinningBytes AG, Switzerland

Abstract

We describe our approaches for the Social Media Geolocation (SMG) task at the VarDial Evaluation Campaign 2020. The goal was to predict geographical location (latitudes and longitudes) given an input text. There were three subtasks corresponding to German-speaking Switzerland (CH), Germany and Austria (DE-AT), and Croatia, Bosnia and Herzegovina, Montenegro and Serbia (BCMS). We submitted solutions to all subtasks but focused our development efforts on the CH subtask, where we achieved third place out of 16 submissions with a median distance of 15.93 km and had the best result of 14 unconstrained systems. In the DE-AT subtask, we ranked sixth out of ten submissions (fourth of 8 unconstrained systems) and for BCMS we achieved fourth place out of 13 submissions (second of 11 unconstrained systems).

1 Introduction

The 7th Workshop on NLP for Similar Languages, Varieties and Dialects (Găman et al., 2020) introduced a new task on *Social Media Geolocation (SMG)*: Given a social media post, a system has to predict the latitude and longitude of where it was written. This is an extension to previous evaluation campaigns (Zampieri et al., 2019; Zampieri et al., 2018; Zampieri et al., 2017), which focused on dialect identification, assigning a discrete label – usually corresponding to a geographic region – to a piece of text. Geolocation prediction allows for a more fine-grained assessment of dialectal varieties without the need to define hard and somewhat arbitrary boundaries within dialect continua.

Our motivation for participating in the SMG shared task was to gain more knowledge about real-world, noisy, digital data. More specifically, we seek to mine written texts for different Swiss German Dialects and would profit from being able to place them geographically, particularly in the context of our other projects on Swiss German.

We submitted solutions to all three sub-tasks (see Results in Section 4) and, in light of our motivation, focused specifically on the Swiss sub-task during development. Our submissions are based on three different models (see Section 3): an SVM meta-classifier combining different classifiers based on word and character features for CH (see Section 3.4); a single SVM with fewer features and no meta-classifer for DE-AT and BCMS (see Section 3.6); and a language modelling approach (see Section 3.5) which was applied to all subtasks. We furthermore experimented with character-level Convolutional Neural Networks (CNNs) (see Section 3.7). For all systems, we cluster geolocations to get a number of discrete labels to predict.

2 Related Work

The central focus of the evaluation campaign at VarDial is to identify dialects of various languages. There have been three previous editions, which laid the basis for dialect identification in Swiss German (Zampieri et al., 2019; Zampieri et al., 2018; Zampieri et al., 2017). Dialect classification is useful for many tasks and applications, e.g. for POS-tagging of dialectal data (Hollenstein and Aepli, 2014), for

Proceedings of the 7th VarDial Workshop on NLP for Similar Languages, Varieties and Dialects, pages 254–264
Barcelona, Spain (Online), December 13, 2020

compilation of German dialect corpora (Hollenstein and Aepli, 2015), or for automatic speech recognition of Swiss German.

Past VarDial campaigns have led to the creation of diverse datasets for language and dialect identification, for example: Samardžić et al. (2016) provide a Swiss German dialect data set based on the ArchiMob corpus, Jauhiainen et al. (2019) present a collection of cuneiform texts derived from a larger open access collection, and Huang et al. (2000) and McEnery and Xiao (2003) created data sets for Taiwanese and Mandarin Chinese. The 2020 SMG task is based on social media posts from Twitter (Ljubešić et al., 2016) and Jodel (Hovy and Purschke, 2018), annotated with geolocations (see Section 3.1).

Many studies addressed the problem of language and dialect identification, creating a noticeable amount of related work, summarised in the evaluation campaign reports (Zampieri et al., 2019; Zampieri et al., 2018; Zampieri et al., 2017) and Jauhiainen et al. (2018b). A typical approach uses Support Vector Machines (SVMs) with different feature extraction methods. The use of character language models for language identification has previously been studied by Vatanen et al. (2010).

Over the years various models have been proposed for text-based geolocation prediction (Han et al., 2014; Kinsella et al., 2011; Rahimi et al., 2017b; Rahimi et al., 2017a).

As for discretization of geolocations, Wing and Baldridge (2014) propose a hierarchical approach to divide the earth into a grid with different levels of granularity. Similarly to Duong-Trung et al. (2017), we use a K-Means clustering approach to subdivide the space, which is more data-driven than a grid.

Our main focus is the CH subtask, where our approach is, from a text classification point of view, most similar to MAZA, which was proposed at VarDial 2017 (Malmasi and Zampieri, 2017). MAZA uses Term Frequency (TF) on character n-grams and word unigram features to train several SVMs. Then it uses a Random Forest meta-classifier with 10-fold cross-validation on the predictions of the SVMs. We extended this approach and used Term Frequency-Inverse Document Frequency (TF-IDF) on word and on character level. We used an SVM as a meta-classifier, and concatenated the output of the base classifiers (see Section 3.4). This solution approach was motivated by the fact that we have already applied similar architectures successfully in a wide range of tasks (Benites de Azevedo e Souza et al., 2019; Benites et al., 2018b; Benites, 2019), especially in (Benites et al., 2018a) we established empirically that for (Swiss German) dialect recognition TF-IDF is better than just TF.

For the BCMS and DE-AT subtasks, we used a single SVM with word- and character-level TF-IDF features (see Section 3.6). We also made submissions using a variant of the HeLI method by Jauhiainen et al. (2016; Jauhiainen et al. (2018a), which we extended with a voting mechanism that takes the centre of the top predicted coordinates in case of low confidence (see Section 3.5).

3 Method

3.1 Task Definition

The shared task data was collected from the social media platforms *Jodel*[1] and *Twitter*[2]. Jodel posts were collected from Germany and Austria (DE-AT), as well as German-speaking Switzerland (CH) (Hovy and Purschke, 2018). Tweets were sourced from Bosnia and Herzegovina, Croatia, Montenegro, and Serbia (BCMS) (Ljubešić et al., 2016). Every sample contains, in addition to the text, latitude and longitude coordinates as set by the users of the respective platform (Jodel or Twitter).

While Tweets are usually authored by a single person, the Jodel samples consist of short conversations involving multiple speakers. This leads to some samples containing multiple dialects. Similarly, we observed samples containing indirect speech in non-local dialects.

For evaluation, two metrics were defined by the organizers: the *median* and the *mean* distances between predicted and real geolocations across all texts in the test set, with the former being the official metric of the SMG shared task.

We opted to frame the task as a text classification problem, by combining locations into discrete clusters and predicting cluster identities.

[1] https://jodel.com/

[2] https://twitter.com/

3.2 Label Clustering

In order to obtain a small number of classes, we use K-Means clustering (Lloyd, 1982) to cluster the geolocations. This allows standard classification methods to tackle the problem, since a certain number of samples per class can then be guaranteed. Generalization is increased, while resolution suffers from the somewhat coarser view. We experimented with different values of k, which will be discussed in subsequent sections. In order to generate coordinates for prediction, we used the centroid coordinate of the predicted cluster.

3.3 Text Preprocessing

The basic preprocessing step common to all systems consisted in splitting the sentences into words on whitespaces. No stopword removal or lemmatization was performed since these steps have been shown to erase features which are useful for differentiating between the dialects (Maharjan et al., 2014). Afterwards, multiple feature extraction methods were applied, as explained in the next sections.

3.4 System 1: SVM-CV

3.4.1 Feature Extraction

Feature Set	Token Type	Case-Sensitive	N-gram Range	Number of Features
1	word	no	1 - 3	70000
2	word	no	1 - 5	70000
3	char	no	1 - 7	30000
4	char	no	2 - 3	50000
5	char	yes	2 - 3	50000
6	char_wb	no	1 - 5	60000
7	char_wb	no	1 - 7	60000
8	char_wb	yes	2 - 3	50000

Table 1: Overview of the different feature sets used by the SVM-CV system. See text for details.

We use a collection of different feature sets based on the TF-IDF representation (Manning et al., 2008). They vary by the type of tokens considered (words, characters, and characters ignoring whitespace), case-sensitivity, the range of n-grams, and the maximum number of features in the set. Table 1 gives an overview of the feature sets that were used. Note that the token type *char_wb* refers to character tokens ignoring whitespace between words. We use the implementation provided by the *scikit-learn*[3] library to extract these features.

3.4.2 Classifiers

For every feature set we train separate linear one-vs-rest SVM classifiers with the discrete cluster identities as target labels. We then use the distances to the decision boundaries of every classifier for every feature set as a new feature vector for another linear SVM meta-classifier.

During training every base classifier is trained via 5-fold cross-validation, and predictions on the held-out fold are used to train the meta-classifier.

Figure 1 illustrates the approach, and we refer to Benites et al. (2018a) for a detailed description.

During prediction, we usually output the geolocation corresponding to the result of our meta-classifier. However, if a sample is below a certain confidence threshold (see also Section 3.8), we assign it the mean latitude and longitude from the complete training data, instead of the location of the predicted cluster center, so the error would be equally distributed and not skewed.

3.5 System 2: LM

Our second approach is a language modelling system and is heavily modelled on the HeLI submission to the VarDial 2018 GDI task (Jauhiainen et al., 2018a). The full method is described in Jauhiainen et al.

[3] `https://scikit-learn.org/stable/`

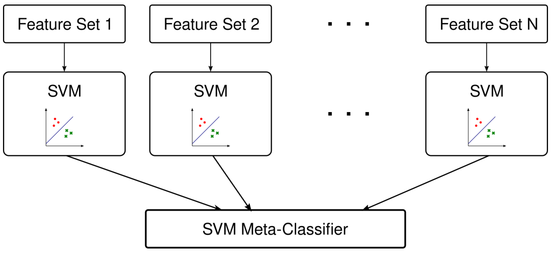


Figure 1: Overview of the SVM-CV classifier

(2016), to which we refer the interested reader.

3.5.1 Corpora and Language Models

We first created local corpora using the same K-Means clustering procedure as outlined in 3.2. We then create character-level language models for each of the corpora using the scoring procedure defined in Jauhiainen et al. (2018a): the text is split into words at whitespaces and relative n-gram frequencies are calculated within each word (including the preceding and following space characters). The score associated with an n-gram of a dialect is the negative decadic logarithm of its relative frequency within that dialect's subcorpus, meaning that n-grams with a high relative frequency have low scores.

3.5.2 Prediction

In order to make a prediction for an unseen input text, a score is calculated for each dialect based on the language models. The text is split into words at whitespaces, and for each word (again including leading and trailing space) the mean of its n-gram scores is calculated. If an n-gram is not present in the model of this dialect, a penalty term is assigned instead. The score of the text is calculated as the mean of its word-level scores, and the dialect with the lowest score is selected as output.

3.5.3 Voting Mechanism

We define the confidence of a prediction in line with Jauhiainen et al. (2018a) as the difference in scores between the second best and the best dialect. For samples that have low confidence, we introduce a voting mechanism where we use the centre of the V highest-confidence clusters as the prediction, whose coordinate is represented by the mean of the V longitudes and the mean of the V latitudes. In section 4.1.2, the value V is represented by v.

3.5.4 Parameters and Tuning

The tunable hyperparameters of this method are: the number of clusters (k), the n-gram order of the language models (n), whether case is preserved in the input to the language models, the penalty term (we assume the same term for all languages) (p), the confidence threshold below which to apply voting, and the number of clusters to use when determining the centre during voting.

We briefly experimented with using a maximum number of features per dialect (called "cutoff" by Jauhiainen et al. (2018a)) but found no improvement.

We used neither the backoff procedure to lower-order n-grams from Jauhiainen et al. (2016) nor the highly promising semi-supervised language model adaptation (Jauhiainen et al., 2018a) due to lack of time.

3.6 System 3: SVM-Base

For the larger DE-AT and BCMS datasets, we did not have sufficient time to train and tune SVM-CV. Instead, we used a simple linear SVM classifier for these languages with the feature sets shown in Table 2. Feature sets 1, 2 and 3 are also used for SVM-CV, corresponding to rows 1, 2 and 5 in Table 1, while set 4 is unique to SVM-Base.

Feature Set	Token Type	Case-Sensitive	N-gram Range	Number of Features
1	word	no	1 - 3	70000
2	word	no	1 - 5	70000
3	char	yes	2 - 3	50000
4	char_wb	yes	1 - 3	150000

Table 2: Overview of the different feature sets used by the SVM-Base system.

3.7 System 4: CNN

We also experimented with a character-wise Convolutional Neural Network (CNN) (Zhang et al., 2015), which we did not submit. We include it as a neural baseline to compare our other approaches against. The network was composed of multiple convolutions in parallel with filter size and width of {(128,2), (96,2), (96,4), (64,3), (64,4)} with dropout set at 0.1 and maxpooling. The output of the convolutional layers are subsequently concatenated. Afterwards a 3-layer fully connected network is applied with 100, 100 and 50 neurons per layer, respectively. The activation function on all layers was ReLU, except for the last where softmax was applied. As output, and so as the number of classes, the number of clusters is used, similarly to the approach of SVMs. We used the Adam optimizer (Kingma and Ba, 2014) with the learning rate set to 0.001 and minimizing the binary cross entropy loss. The network is then trained for 100 epochs.

3.8 Handling Outliers

We discovered one text in French in the development set and decided to use the language detection library *langdetect*[4]. If the language is detected as French we set the coordinates to (46.67, 7.0), the center of the French-speaking part of Switzerland. In case the prediction score is very low (below -0.9 for SVM-CV and -0.8 for SVM-Base) we assign the text to the center of the training data with coordinates (47.26, 8.3).

4 Results

In the following, we evaluate the performance of our four systems plus the two simple baseline systems. Since we were primarily focusing on the CH subtaks, we present the detailed analysis for these data. Later on, we briefly present how our systems performed on the other subtasks.

4.1 CH subtask

4.1.1 Optimizing Number of K-Means Centroids on CH data

One of the most important parameters when using clustering to discretize geographical data, is the number of centroids k for the K-Means algorithm. This determines the upper bound on performance as well as the number of samples per class and the number of classes. Usually, classification performance decreases rapidly with an increasing number of classes, which most probably negatively affects the median distance[5], the main metric of this competition.

We analyzed the reconstruction error with different numbers of clusters on the training set, over ten runs for each setting, i.e. we clustered the locations of the training samples and then calculate the distance between the cluster centroid to the actual location of the sample assigned to this centroid. The results are depicted in Table 3, where we show median and mean distances for 10, 20, 35, 50, 75 and 100 clusters, along with their variances. We see that the largest relative drop is between 50 and 75 ($\frac{0.79}{0.09}$=878%), but the difference is almost negligible in terms of geographical dialectal difference. The drop between 35 and 50 is also interesting, although it might be difficult to argue that there are about 50 dialectal hotspots. We chose 35 to use as k parameter for the K-Means algorithm, since it promised the least error for the most generalization capacity, i.e. lower risk of overfitting.

[4] https://github.com/Mimino666/langdetect

[5] We judge the probability very low for the case when a finer-grained division (more clusters, e.g. cluster a is subdivided into subcluster b, c and d), allows a finer resolution, and a misclassification might still decrease the mean distance (e.g. b is right, but d is predicted, however subcluster c cause that the center of a is far distant than the target (which would lie within b)).

Nr. Clusters	Median	Var	Mean	Var
10	10.61	± 0.16	11.33	± 0.01
20	5.65	± 0.29	6.91	± 0.04
35	3.19	± 0.02	3.91	± 0.01
50	0.79	± 0.10	1.72	±0.00
75	0.09	± 0.00	0.42	± 0.00
100	0.04	± 0.00	0.15	± 0.00

Table 3: Reconstruction error depending on the number of clusters for CH-subtask training data, for 10 runs

Nr.	k	p	med-dv	mean-dv
1	60	5.5	18.33	27.27
2	60	5.6	18.33	27.30
3	70	5.8	18.33	27.42
4	70	5.6	18.42	27.52
5	70	5.9	18.49	27.37
6	70	5.5	18.64	27.61
7	70	5.7	18.64	27.64
8	60	5.8	18.70	27.47
9	60	5.9	18.70	27.41
10	60	5.4	18.73	27.28

Table 4: Tuning results for LM-CH: first step, n=4, cs=no

				Dev		Test	
Nr.	cs	v	vt	median	mean	median	mean
	no	0	n/a	18.33	27.27	19.05	27.97
1	yes	3	0.01	**17.17**	26.06	17.66	26.21
2	yes	3	0.02	17.30	25.76	17.75	25.79
3	no	3	0.01	17.41	25.87	17.84	26.44
4	no	3	0.02	17.42	25.33	**17.56**	25.77
5	yes	2	0.01	17.47	26.38	18.35	26.63
6	yes	2	0.02	17.49	26.09	18.44	26.41
7	no	4	0.01	17.53	25.81	18.07	26.30
8	yes	4	0.01	17.78	26.16	17.87	26.15
9	no	4	0.02	17.81	25.32	18.04	25.55
10	yes	4	0.02	18.01	25.87	18.24	25.74

Table 5: Tuning results for LM-CH: second step; best relevant results marked in bold

4.1.2 LM Parameter Tuning

We proceeded in two steps for tuning the parameters of the LM system. First, we searched over the n-gram-level ($n \in \{4, 5, 6\}$), number of clusters ($c \in \{35, 40, 50, 60, 70\}$), penalty ($p \in \{5, 5.1, 5.2, \ldots, 6\}$), and case-sensitivity (cs), of which we selected the best configuration. Using this parameter set, we fine-tuned the parameters relating to voting (see Section 3.5.3) in a second step , i.e. the number of voters ($v \in \{0, 2, 3, 4, 5\}$) and the voting confidence threshold ($vt \in \{0.001, 0.01, 0.02, 0.05, 0.1\}$).

Please refer to Section 3.5.4 for the description of the parameters.

In Table 4, we report the results of the first step, showing the ten configurations with the best results in descending order by median distance error. The best-performing n-gram order is 4, which is in line with results obtained by Jauhiainen et al. (2018a). We can also see that larger numbers of clusters and penalties above 5.4 are beneficial. The best models are not sensitive to case; we hypothesize that this is because lower-casing helps overcome data sparsity.

Table 5 shows the results of the second step using n-gram-level of 4 (n=4), cluster size of 60 (k=60), and penalty to 5.5 (p=5.5), with the best ten results by median distance on the development set. We tune the voting-related parameters v and vt. We also tune case-sensitivity cs again, since the voting scenario could equalize the more sparse data. We can see that the voting mechanism significantly improves performance on both development and test set. The most successful configuration for CH uses three voters and a confidence threshold of 0.01, leading to a median distance of 17.66 km on the test set, which corresponds to the fourth best submission for this subtask.

		Dev		Test		
System	Clusters	Median	Mean	Median	Mean	Submitted
Baseline: Center	-	43.13	48.10	43.13	48.47	
Baseline: SVM-Base-Unigram	10	21.29	28.58	19.99	27.94	
Baseline: SVM-Base-Unigram	20	**19.53**	29.23	19.02	28.06	
Baseline: SVM-Base-Unigram	35	19.86	29.30	**18.64**	28.70	
Baseline: SVM-Base-Unigram	50	19.93	29.65	18.83	28.54	
Baseline: SVM-Base-Unigram	100	20.13	29.67	19.06	29.06	
System 1: SVM-CV	20	17.80	25.60	17.83	25.46	
System 1: SVM-CV	35	16.83	26.36	15.93	25.05	x
System 1: SVM-CV	50	**16.68**	25.27	**15.59**	24.39	
System 1: SVM-CV	100	16.83	25.65	15.93	24.30	
System 2: LM	10	19.74	27.81	19.55	28.25	
System 2: LM	20	19.05	27.48	19.69	27.76	
System 2: LM	35	18.97	27.22	18.33	26.97	
System 2: LM	50	17.50	26.87	**17.51**	26.47	
System 2: LM	60	**17.17**	26.06	17.66	26.21	x
System 2: LM	70	17.53	26.40	18.07	26.49	
System 2: LM	100	17.62	26.39	18.27	26.44	
System 3: SVM-Base	20	**19.62**	28.63	19.69	28.18	
System 3: SVM-Base	35	19.68	29.22	18.80	28.17	
System 3: SVM-Base	50	19.90	29.39	**18.32**	28.02	
System 3: SVM-Base	100	20.03	28.83	19.06	28.48	
System 4: CNN	20	**24.66**	33.12	24.68	33.21	
System 4: CNN	35	25.78	35.19	23.30	32.00	

Table 6: Results for CH subtask for different systems on development and test sets

4.1.3 Comparison of the Different Systems on the CH subtask

We report the results for the various systems with different numbers of clusters in Table 6. In addition to the systems described in Section 3 we include 2 baselines: *Center* predicting the geographic center of the training set for every sample, and *SVM-Base-Unigram* which is a version of SVM-Base using only unigram word features. The parameters of LM are set according to the best setting presented in 4.1.2 and only the number of clusters is varied.

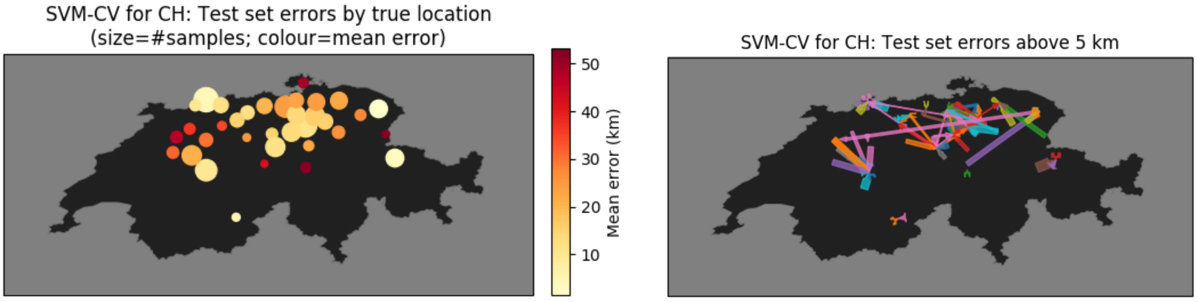
CNNs give relatively good results which would score about 10-11th place in the competition. A simple SVM-TF-IDF baseline with Unigram feature extraction would already be among the best 10 places with a median distance of about 20km. Increasing the number of clusters from 10 to 20 makes it better, but then the error distance increases for *SVM-Base-Unigram*. *SVM-Base* has comparable performance to *SVM-Base-Unigram* which could point to simple word/features being already good indications of geographic locations.

The LM method (System 2) benefits from a larger number of clusters than the SVM- and CNN based ones, peaking at 50 clusters on the test set and 60 on the development set.

For the SVM-CV system we see a drop of about 2 points compared to *SVM-Base*. Using the optimum number of clusters we get very close to the winning system.

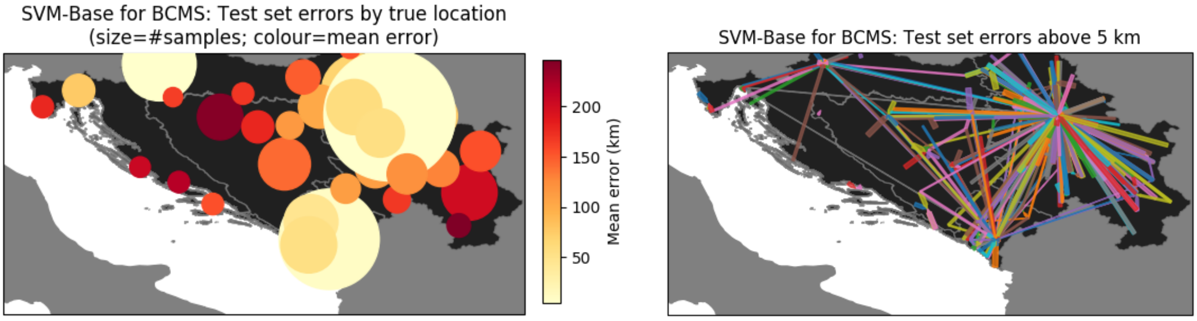
Geographical Error Analysis of SVM-CV We can see from Figure 2a that the hotspots around Zurich with the most texts were predicted with good quality. Problematic were the borders where there were regions containing smaller number of texts. For example, the Basel region (top left) was very well predicted, whereas the regions of Schaffhausen (top most) and St. Galler Rheintal (right most) were often wrongly predicted by a large distance.

In Figure 2b, we can see the confusion of the largest errors (more than 5 km). We can clearly see a confusion between the region of Bern (left most) and St. Gallen (top right). Also St. Gallen and Schwyz (red spot below in the middle).


SVM-CV for CH: Test set errors by true location
(size=#samples; colour=mean error)
50
40
30
20
10
Mean error (km)
SVM-CV for CH: Test set errors above 5 km

(a) Errors of SVM-CV classifier for CH.

(b) Errors of SVM-CV classifier for CH.


SVM-Base for BCMS: Test set errors by true location
(size=#samples; colour=mean error)
200
150
100
50
Mean error (km)
SVM-Base for BCMS: Test set errors above 5 km

(a) Errors of SVM-Base classifier for BCMS.

(b) Errors of SVM-Base classifier for BCMS.

4.2 BCMS subtask

System 2: LM We tuned the same parameters as described in Section 4.1.2 for CH and found the best setting to be identical to CH, except for the absence of case-sensitivity, and using five voters instead of three, which resulted in a development set median distance of 109.86 km. The LM-based approach performed rather poorly in the evaluation, scoring last place out of all submissions with 111.4 km median distance.

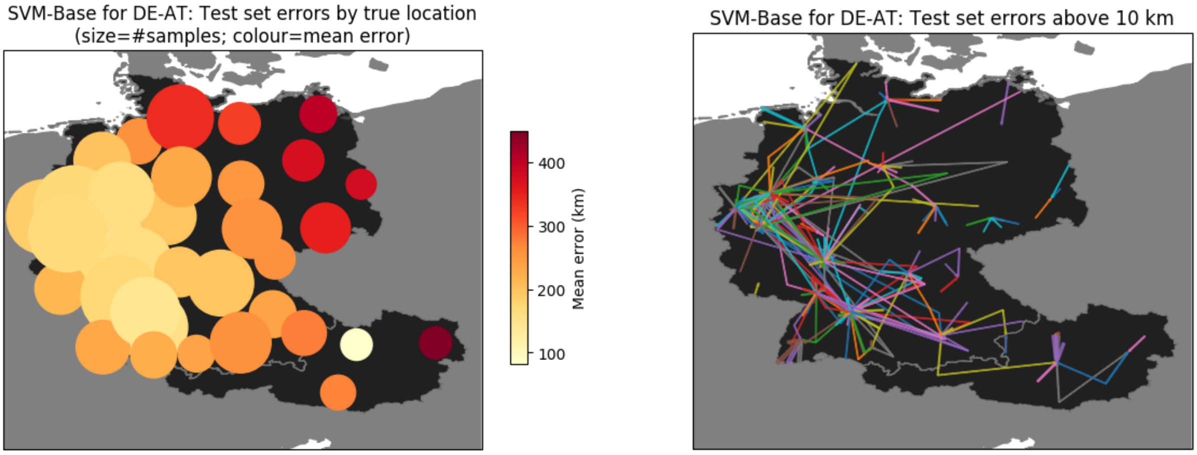
System 3: SVM-Base The SVM-Base system performed somewhat better. We evaluated different numbers of clusters: 25, 35, 50, 75 and 100, which yielded development set results within 3 km (59.02 with 35 clusters to 62.05 km with 100 clusters). Hence, our submission was based on 35 clusters. It achieved fourth rank in terms of submissions (second by teams), with 57.2 km median distance, more than 15 km behind the winning submission of 41.54 km.

System 1: SVM-CV: Addendum After the competition, the gold labels were released and we had time to run the SVM-CV system on all sub-tasks. We also calculated the predictions for System 1 with 35 Cluster which took roughly 2 days. We achieved a better result than the first placed (41.54 km) approach with a median distance of 36.79 km but a worse mean distance of 83.08 km (80.89 km for the first place). An analysis why this system performed better in this dataset in comparison to the other competitors in the other two datasets would be interesting, but we leave this for future work.

Geographical error analysis Figures 3a and 3b visualise the errors of the SVM-Base system on the BCMS subtask. We can see that areas with many samples, mostly around the capital cities of the respective countries, are predicted accurately (Figure 3a), but also that there is a strong trend of assigning False Positives to them (Figure 3b).

4.3 DE-AT subtask

System 2: LM We used the same parameters of BCMS subtask System 2 for the DE-AT. On the development set, this achieved 229.46 km, while on the test set the result was 217.8 km, 8th place among submissions and a large margin behind the best submission of 143.3 km.


SVM-Base for DE-AT: Test set errors by true location
(size=#samples; colour=mean error)
400
300
200
100
Mean error (km)
SVM-Base for DE-AT: Test set errors above 10 km

(a) Errors of SVM-Base classifier for DE-AT.

(b) Errors of SVM-Base classifier for DE-AT.

System 3: SVM-Base Due to lack of time we evaluated only 25, 35 and 50 clusters, of which 25 clusters performed the best on the development set, resulting in a median distance of 200.81 km. The test set result of 205.81 km ranked sixth, markedly behind the top three submissions.

System 1: SVM-CV: Addendum As pointed out before in Section 4.2, after the competition, we ran System 1 on all sub-tasks. This took for DE-AT roughly 5 days to finish. The prediction quality achieved with 35 clusters yielded a median distance of 167.01 km (first place: 143.3 km) and a mean distance of 193.32 km (first place: 166.64 km).

Geographical error analysis We visualize the results of the better submission which was again SVM-Base. As can be seen in Figure 4a, the main difficulties of the system are in the regions of Eastern Germany and Eastern Austria. In Figure 4b, we can see that texts from these problematic regions tend to be assigned more to the west; but also that many smaller errors are accumulated in the more populous areas along the Rhine.

5 Conclusion

We presented our approach to the VarDial 2020 SMG shared task, focusing on the submission for the CH subtask. Despite the expected noise, caused by people moving between different regions without adjusting their writing, a meta algorithm on top of SVMs with different n-grams weighted by TF-IDF performs impressively well, particularly for Switzerland (CH subtask). We achieve the second rank in terms of teams and the third by submissions with a median distance error of only 15.93 km. Deep learning approaches combining CNN and K-Means also showed interesting results but are still far behind a simple Unigram-TF-IDF with K-Means.

Acknowledgements

We thank the task organizers for their support and the reviewers for their detailed and helpful feedback. This research has been funded by the Swiss Innovation Agency project no. 28190.1 PFES-ES and by SpinningBytes AG, Switzerland.

References

Fernando Benites, Ralf Grubenmann, Pius von Däniken, Dirk von Grünigen, Jan Deriu, and Mark Cieliebak. 2018a. Twist Bytes-German Dialect Identification with Data Mining Optimization. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 218–227.

Fernando Benites, Shervin Malmasi, and Marcos Zampieri. 2018b. Classifying patent applications with ensemble methods. *arXiv preprint arXiv:1811.04695*.

Fernando Benites de Azevedo e Souza, Pius von Däniken, and Mark Cieliebak. 2019. Twistbytes-identification of cuneiform languages and german dialects at vardial 2019. In *6th Workshop on NLP for Similar Languages, Varieties and Dialects, VarDial 2019, Minneapolis, United States, 7 June 2019*, pages 194–201. Association for Computational Linguistics.

Fernando Benites. 2019. Twistbytes–hierarchical classification at germeval 2019: walking the fine line (of recall and precision). *arXiv preprint arXiv:1908.06493*.

Nghia Duong-Trung, Nicolas Schilling, Lucas Rego Drumond, and Lars Schmidt-Thieme. 2017. An effective approach for geolocation prediction in twitter streams using clustering based discretization.

Mihaela Găman, Dirk Hovy, Radu Tudor Ionescu, Heidi Jauhiainen, Tommi Jauhiainen, Krister Lindén, Nikola Ljubešić, Niko Partanen, Christoph Purschke, Yves Scherrer, and Marcos Zampieri. 2020. A Report on the VarDial Evaluation Campaign 2020. In *Proceedings of the Seventh Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*.

Bo Han, Paul Cook, and Timothy Baldwin. 2014. Text-based twitter user geolocation prediction. *J. Artif. Int. Res.*, 49(1):451–500, January.

Nora Hollenstein and Noëmi Aepli. 2014. Compilation of a Swiss German dialect corpus and its application to PoS tagging. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pages 85–94.

Nora Hollenstein and Noëmi Aepli. 2015. A Resource for Natural Language Processing of Swiss German Dialects. In *GSCL*.

Dirk Hovy and Christoph Purschke. 2018. Capturing regional variation with distributed place representations and geographic retrofitting. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4383–4394, Brussels, Belgium. Association for Computational Linguistics.

Chu-Ren Huang, Feng-Yi Chen, Keh-Jiann Chen, Zhao-ming Gao, and Kuang-Yu Chen. 2000. Sinica treebank: Design criteria, annotation guidelines, and on-line interface. In *Proceedings of the Second Workshop on Chinese Language Processing: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 12*, CLPW '00, pages 29–37, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tommi Sakari Jauhiainen, Bo Krister Johan Linden, Heidi Annika Jauhiainen, et al. 2016. Heli, a word-based backoff method for language identification. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects VarDial3, Osaka, Japan, December 12 2016*.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2018a. Heli-based experiments in swiss german dialect identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 254–262.

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2018b. Automatic language identification in texts: A survey. *arXiv preprint arXiv:1804.08186*.

Tommi Jauhiainen, Heidi Jauhiainen, Tero Alstola, and Krister Lindén. 2019. Language and Dialect Identification of Cuneiform Texts. *arXiv preprint*, arXiv:1903.01891.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Sheila Kinsella, Vanessa Murdock, and Neil O'Hare. 2011. "i'm eating a sandwich in glasgow": Modeling locations with tweets. In *Proceedings of the 3rd International Workshop on Search and Mining User-Generated Contents*, SMUC '11, page 61–68, New York, NY, USA. Association for Computing Machinery.

Nikola Ljubešić, Tanja Samardžić, and Curdin Derungs. 2016. TweetGeo - a tool for collecting, processing and analysing geo-encoded linguistic data. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3412–3421, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Stuart Lloyd. 1982. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

Suraj Maharjan, Prasha Shrestha, and Thamar Solorio. 2014. A Simple Approach to Author Profiling in MapReduce. In *CLEF*.

Shervin Malmasi and Marcos Zampieri. 2017. German Dialect Identification in Interview Transcriptions. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 164–169, Valencia, Spain, April.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

A. M. McEnery and R. Z. Xiao. 2003. The lancaster corpus of mandarin chinese., 12.

Afshin Rahimi, Timothy Baldwin, and Trevor Cohn. 2017a. Continuous representation of location for geolocation and lexical dialectology using mixture density networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 167–176, Copenhagen, Denmark, September. Association for Computational Linguistics.

Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. 2017b. A neural model for user geolocation and lexical dialectology. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 209–216, Vancouver, Canada, July. Association for Computational Linguistics.

Tanja Samardžić, Yves Scherrer, and Elvira Glaser. 2016. ArchiMob – a corpus of spoken Swiss German. In *Proceedings of LREC*.

Tommi Vatanen, Jaakko J. Väyrynen, and Sami Virpioja. 2010. Language Identification of Short Text Segments with N-gram Models. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).

Benjamin Wing and Jason Baldridge. 2014. Hierarchical discriminative classification for text-based geolocation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 336–348.

Marcos Zampieri, Shervin Malmasi, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, Jörg Tiedemann, Yves Scherrer, and Noëmi Aepli. 2017. Findings of the VarDial Evaluation Campaign 2017. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, Valencia, Spain.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shuon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, Chris van der Lee, Stefan Grondelaers, Nelleke Oostdijk, Antal van den Bosch, Ritesh Kumar, Bornini Lahiri, and Mayank Jain. 2018. Language Identification and Morphosyntactic Tagging: The Second VarDial Evaluation Campaign. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, Santa Fe, USA.

Marcos Zampieri, Shervin Malmasi, Yves Scherrer, Tanja Samardžić, Francis Tyers, Miikka Silfverberg, Natalia Klyueva, Tung-Le Pan, Chu-Ren Huang, Radu Tudor Ionescu, Andrei Butnaru, and Tommi Jauhiainen. 2019. A Report on the Third VarDial Evaluation Campaign. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*. Association for Computational Linguistics.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *CoRR*, abs/1509.01626.