Dialect and variant identification as a multi-label classification task: a proposal based on near-duplicate analysis
Bernier-Colborne, Gabriel; Goutte, Cyril; Leger, Serge

Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023), 2023-05-05

Dialect and Variant Identification as a Multi-Label Classification Task: A Proposal Based on Near-Duplicate Analysis

Gabriel Bernier-Colborne and **Cyril Goutte** and **Serge Léger**
National Research Council Canada
{Gabriel.Bernier-Colborne | Cyril.Goutte | Serge.Leger}@nrc-cnrc.gc.ca

Abstract

We argue that dialect identification should be treated as a multi-label classification problem rather than the single-class setting prevalent in existing collections and evaluations. In order to avoid extensive human re-labelling of the data, we propose an analysis of ambiguous near-duplicates in an existing collection covering four variants of French. We show how this analysis helps us provide multiple labels for a significant subset of the original data, therefore enriching the annotation with minimal human intervention. The resulting data can then be used to train dialect identifiers in a multi-label setting. Experimental results show that on the enriched dataset, the multi-label classifier produces similar accuracy to the single-label classifier on test cases that are unambiguous (single label), but it increases the macro-averaged F1-score by 0.225 absolute (71% relative gain) on ambiguous texts with multiple labels. On the original data, gains on the ambiguous test cases are smaller but still considerable (+0.077 absolute, 20% relative gain), and accuracy on non-ambiguous test cases is again similar in this case. This supports our thesis that modelling dialect identification as a multi-label problem potentially has a positive impact.

1 Introduction

In this paper, we argue that dialect[1] identification should be treated as a multi-label classification problem unless it can be shown that every text in a given dataset belongs to only one dialect or language variant. This feels like a natural hypothesis, as it seems reasonable that some utterances are equally valid in more than one dialect or variant. However, most datasets for, and evaluations of this task rely on single-label classification, where each utterance is annotated as belonging to a single variant.[2]

Previous work shows that manually identifying the language variety of a text is difficult, and that it is actually easier for native speakers to identify texts that are *not* in their variety (Goutte et al., 2016, sec. 4.4). Accordingly, proper multi-label manual annotation requires multiple annotators with complementary skills, and therefore massive annotation budget, when run at the usual scale of tens-to-hundreds of thousands of utterances.

In this work, we focus instead on analyzing and processing already existing dialect identification data, with minimal annotation need. We argue that automatically assessing differences between two similar texts, as done here, is an easier task. We explore empirically how the data can be enriched with multiple labels, and how switching to the multi-label classification paradigm can potentially improve performance in identifying dialects and variants.

We start by analyzing the duplicates and near-duplicates in an existing dataset built for French dialect identification. We search for instances that are identical or highly similar textually, but are annotated with different labels. We find that a considerable number of near-duplicates have different labels, but no obvious differences that could be considered dialectal in nature.

We further show that near-duplicate analysis is useful in at least two ways. First, it allows us to inspect and refine a dataset, in a manner similar to *measuring data* (Wang et al., 2022; Mitchell et al., 2022, inter alia), by identifying phenomena that might otherwise go unnoticed, e.g. texts that are assigned to different classes but have no actual dialectal differences or spotting artefacts due to the selection of text sources or to the processing

[1] In this paper, we use the terms "dialect" and "language variant" somewhat interchangeably. In the FreCDo dataset, language variants are specifically delimited by national origin, as determined by the top-level domain of the original webpage.

[2] A notable exception is this year's "True Labels" shared task at VarDial (https://sites.google.com/view/vardial-2023/shared-tasks).

Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023), pages 142–151
May 5, 2023 ©2023 Association for Computational Linguistics

pipeline (e.g. boiler plate removal, sentence splitting, etc.). Second, by spotting similar texts that have no obvious dialectal differences, it allows us to convert an existing dataset in single-label format into a multi-label dialect classification format.

Using the results of this analysis, we combine the labels of near-duplicates to create what we argue is a more accurate representation of the data. For further empirical validation of this approach, we use this data to train a multi-label classifier for dialect identification. We compare those results to single-label classification and show that the overall classification performance stays at a similar level, while the performance on the subset of examples that have multiple labels is greatly improved.

The experimental code developed in this work is available at https://github.com/gbcolborne/vardial2023.

2 Data

For this project, we used the FreCDo corpus (Găman et al., 2022),[3] which was used for the Cross-Domain French Dialect Identification (FDI) shared task at the VarDial 2022 evaluation campaign (Aepli et al., 2022). It contains 413,522 short texts belonging to one of four varieties of French from Belgium (BE), Canada (CA), Switzerland (CH), and France (FR), cf. Table 1. The data is unbalanced, with a much lower number of CA texts (8.5% overall, $< 1\%$ on Dev). The training, development, and test sets were compiled from several public news websites using different keywords, in order to create a cross-domain split. Furthermore, tokens that are part of a named entity were replaced with the special token “NE”.

	BE	CA	CH	FR
Train	121,746	34,003	141,261	61,777
Dev	7,723	171	5,244	4,864
Test	15,235	944	9,824	10,730

Table 1: Number of text segments in the original FreCDO corpus.

We selected this dataset for several reasons. First, we wanted to follow up on the results of the shared task at VarDial 2022 that exploited this dataset. The results of that shared task pointed to various properties of the dataset that could explain some of the errors made by the submitted systems, and the generally low scores of both the baselines and the submitted systems (Bernier-Colborne et al., 2022). These include the presence of duplicates both within classes and across classes. In this work, we extend the analysis of the data to include near-duplicates.

Second, this dataset features four different dialects of French, which seemed promising in terms of identifying texts that belong to more than one dialect. In particular, the four-variant setting seems more flexible than the situation where only two variants are considered (e.g. Portuguese from Brazil and Portugal), in which case the only multi-label configuration is essentially all labels.

Third, the authors of this paper are all fluent in (one or more variants of) French and were therefore able to analyze the texts and identify possible dialectal differences between texts or dialectal markers in a given text.

It is important to note that this dataset was created using methods that are common for dataset compilation for dialect identification tasks (aside from the cross-domain split). These methods include scraping texts from the Internet and assigning them to a language variety based on the top-level domain name of the source. This practice naturally leads to a single-label formulation of the problem, if each unique text is only present in one of the sources.

The limitations of this practice was a motivating factor for the DSL-TL (Discriminating Between Similar Languages - True Labels) shared task at this year’s VarDial evaluation campaign:

> The DSLCC was compiled under the assumption that each instance’s gold label is determined by where the text is retrieved from. While this is a straightforward (and mostly accurate) practical assumption, previous research has shown the limitations of this problem formulation as some texts may present no linguistic marker that allows systems or native speakers to discriminate between two very similar languages or language varieties.[4]

The solution proposed in DSL-TL was therefore to curate a higher-quality, human-annotated subset of an existing collection of dialect identification data, DSLCC[5], such that some of the resulting examples

[3] https://github.com/MihaelaGaman/FreCDo

[4] https://sites.google.com/view/vardial-2023/shared-tasks

[5] http://ttg.uni-saarland.de/resources/DSLCC/

have multiple labels (Zampieri et al., 2023). This is in line with our proposal to reformulate the problem as a multi-label classification task. However, although DSL-TL provides high-quality annotation on a subset of data, we focus on the use of semi-automatic near-duplicate analysis in order to minimize the annotation burden. Also, as mentioned earlier, the dataset used in this work contains four different dialects of French, whereas the DSL-TL dataset uses only two dialects for each of three different languages: American and British English, Brazilian and European Portuguese, and Argentinian and Peninsular Spanish.

It is also important to note that deduplication is often applied to datasets for dialect identification, although we have observed duplicates both within and across classes in several such datasets. If deduplication is somewhat common, near-duplicate analysis is not a common step in dataset development as far as we can tell.[6] We argue that it is a useful tool in the context of dialect identification. It can be carried out efficiently and provides useful additional information. In fact, our analysis shows that many highly similar near-duplicates vary only in minor aspects that have nothing to do with dialectal variation or lexical choice, such as slight changes in punctuation or formatting (for example the choice of double quotes), which are typically missed by standard deduplication pipelines.

In the following experiments, we used our own, random split of the texts, because the cross-domain nature of the original split was not relevant for our purposes. We also wanted to eliminate the small amount of leakage of texts between the training, development, and test portions of the original dataset. We therefore created an 85/5/10 split, as this was approximately the size of the partitions in the original dataset, by randomly sampling the train/dev/test from the entire original collection.

3 Methods

In this work, we first identify ambiguous near-duplicates that are present in an existing single-label dataset for dialect identification. We perform a light manual inspection (Section 3.2), then create an enriched version of the data by combining the labels of near-duplicate texts. Finally, we train and evaluate classifiers on the resulting data.

3.1 Identification of Ambiguous Near-duplicates

We used two different text similarity measures to identify near-duplicates. Then, by checking their respective labels, we focus on the near-duplicate pairs that have different label sets.

The first similarity measure is the character-level Levenshtein edit ratio. This is computed by normalizing the Levenshtein distance by the sum of the length of the two texts, and turning that into a similarity by subtracting the result from 1. We used the `Levenshtein` library[7] for Python to compute this, using an arbitrary cutoff at 0.8 to speed up the computation and extract only the most similar text pairs. Given the large size of the pairwise similarity matrix, we used a sparse matrix representation to limit memory usage.[8]

The second similarity measure is what we refer to as the *Manhattan similarity* of the word bigram frequency count vectors of the two texts. This is the absolute difference between the two count vectors divided by the sum of the two vectors, then turned into a similarity again by subtracting from 1. Our motivation for using word bigrams was that these were the most useful features for sparse vector-based classifiers according to the results of the shared task (Aepli et al., 2022; Bernier-Colborne et al., 2022). In order to limit memory requirements, we computed similarities in mini-batches, and kept the 1000 highest similarities for each text.

We are aware that we could integrate additional statistics such as the length of the texts in the similarity measure used to identify interesting near-duplicates. However, we have chosen to explore two text similarities that use very different information, one relying on character sequences and the other on word bigram counts, instead of engineering a more complex measure.

Note that we also considered testing sentence embedding methods, but we prioritised methods that are focused on surface similarity, whereas sentence embedding methods are designed to model semantic similarity beyond surface characteristics.

3.2 Manual Inspection

A sample of the most similar text pairs with different labels, which we will call *ambiguous near-duplicates*, was manually inspected and annotated

[6] We are not aware of a single dataset where such analysis was described in the documentation.

[7] `https://github.com/maxbachmann/Levenshtein`

[8] We use `scipy.sparse` for this purpose, (`https://docs.scipy.org/doc/scipy/reference/sparse.html`).

by the authors.[9] The goal was to estimate the proportion of near-duplicates that showed no obvious dialectical differences or markers. We also used the results of this inspection to establish a minimum similarity threshold above which it was unlikely that true dialectal differences were present. For the classification experiments we conduct later, we then assume that all ambiguous text pairs with similarity above that threshold can be considered valid in each of their respective dialects, so we combine their labels (as explained in Section 3.3) before training a multi-label classifier.

The visual inspection was done using an interface that highlights the differences between two similar texts, so that we could quickly locate those differences and assess their nature. We also developed a simple annotation protocol with three possible judgments or categories for each pair of ambiguous near-duplicates. In practice, for each of the two similarity measures, we randomly sampled 260 ambiguous near-duplicates, above an arbitrary threshold on the similarity measure (0.8 for Levenshtein, 0.6 for Manhattan). Out of these 260 examples, 20 were annotated by all human judges, to calibrate their judgments and have a rough estimate of inter-annotator agreement. The other samples were split evenly and annotated by one judge each. We defined a simple annotation protocol for this task, which we refined on one of the common sets of 20 samples. For each sample, the annotator had to pick one of three categories:

1. No lexical differences (e.g. minor changes to punctuation, function words, number of NE tokens, span of NE tokens, numbers, etc.).

2. Minor differences, like something an editor might do to a text, with no potentially dialectal differences.

3. Potentially dialectal differences (including differences in content, such as lexical choice, or addition/removal of entire clauses or sentences).

Examples in the first two categories are very unlikely to present actual dialectal differences or markers, therefore if a pair of texts falls in this category, it is likely justified to combine their label sets, as we do following the method explained in Section 3.3. In the third case, where there *might* be actual dialectal differences between the two texts, combining the labels might introduce noise. Examples are provided in Section 4.1

Note that this simple protocol could likely be improved in the future to ensure higher agreement between annotators.

3.3 Combining Labels

Instead of representing the label of each text as a single integer representing a class identifier, we use a set containing the classes that were observed for that text. So, at first, the vast majority of texts have a single class in their label set. The only exceptions are the texts that appear more than once in the original dataset, and with more than one unique label (i.e. ambiguous *exact* duplicates). This version of the data is referred to as the 'Original' data below. We also initialize a 'Combined' version of the data by copying the Original version.

Once the similarity threshold for near-duplicates has been set, as explained in Section 3.2, we identify all pairs of texts $(x_i,\ x_j)$ with $i < j$ and a similarity greater or equal to that threshold. For each of these pairs, we add the Original label set of each text in the pair to the Combined label set of the other text.[10]

So, given two texts x_1 and x_2 with Original label sets $\{y_1\}$ and $\{y_2\}$ respectively, if $y_1 \neq y_2$ and the similarity of x_1 and x_2 is above the threshold, then the Combined labels sets of both texts becomes $\{y_1, y_2\}$.

Note that in this process, the same text may receive labels from more than one other text, if it has more than one neighbour given the similarity threshold. So, if text x_3 with Original label set $\{y_3\}$ is also a neighbour of x_1, then the Combined label set of x_1 becomes $\{y_1, y_2, y_3\}$ (assuming $y_2 \neq y_3$, otherwise the label set is unchanged, as y_2 was already in it), and the Combined label set of of x_3 becomes $\{y_1, y_3\}$ (assuming $y_1 \neq y_3$).

3.4 Training and Evaluating Classifiers

We developed a pipeline to train and evaluate single-label and multi-label classifiers.

For the multi-label setting, it takes the source data, a pairwise similarity matrix for the texts, and a minimum similarity threshold, and produces a

[9] All native French speakers, two from Canada and one from France.

[10] A slightly different method would be to first identify sets of neighbouring texts, and assign the combined label set to all of these. This might increase the average number of labels per text, but it would also assume that texts belonging to the same neighbour set should be treated as neighbours even if their similarity measure is below the threshold.

dataset for multi-label classification, by combining the labels of duplicates and near-duplicates that have more than one unique label. It also produces a single-label representation of that data, by creating duplicates both within and across classes, as in the original data. Finally, it creates a single-label version without in-class duplicates. We also create these three representations of the data using the original labels rather than the combined labels.

The texts are randomly split into training, development and test sets (85%, 5% and 10%, respectively). The same split of texts is used for single-label and multi-label settings.

On each of the training sets, we fine-tuned a pre-trained French language model, namely CamemBERT (Martin et al., 2020), which uses the RoBERTa architecture and training procedure (Liu et al., 2019). This was the most successful approach on the FDI shared task at VarDial 2022 (Aepli et al., 2022). We downloaded the `camembert-base` checkpoint from the HuggingFace repository of pre-trained models.[11] This model has 110 million parameters, and was pre-trained on the French portion of the OSCAR corpus (Ortiz Suárez et al., 2019; Ortiz Suárez et al., 2020; Abadji et al., 2021).

Given that we use a transformer architecture, training a multi-label classifier rather than a single-label one only involves a few changes to the output layer (or head) and the representation of the targets.

For the single-label classifiers, we add a randomly initialized softmax output layer and use the cross-entropy loss function. Targets are represented as a single integer class ID for each example.

For the multi-label classifiers, we feed the output logits to a sigmoid activation function and use the binary cross-entropy loss function. Targets are represented as a binary vector indicating which classes a given example belongs to.

The models are fine-tuned using the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of 5×10^{-5} and a batch size of 8 for 3 epochs. These were the hyperparameter settings used by Bernier-Colborne et al. (2022) in the FDI shared task to fine-tune their open run 2 model that achieved the highest score (without ensembling) on the development set.

In the single-label setting, the model produces a probability distribution over all classes, and predicts the most likely class for each example. In the multi-label setting, the model produces a probability for each class, and the predicted labels are all classes for which that probability is greater than 0.5. We do not apply any calibration methods to either the single-label or the multi-label classifiers that we trained.

Both single-label and multi-label models were evaluated on the same test examples, by computing the F1-score of each class, as implemented in `sckikit-learn`.[12] Note that for class-wise F1-scores, the predicted and gold labels are binary, and the score is computed in exactly the same way for single-label and multi-label settings. We also report the macro-averaged F1-score (class-wise average) and weighted F1-score (class-wise average weighted by the support of each class). Macro-averaged F1 is the more common evaluation measure for language identification, but we also report weighted average for completeness.

It is important to note that the scores reported in this paper can not be compared to the scores achieved on the shared task, as our random split of the data is different. In particular, we did not keep the cross-domain split in the original data, because it was not relevant to the problem explored in this paper. As a consequence, our scores are considerably higher.

We evaluate the classifiers both on unambiguous examples, i.e. examples that belong to only one class in the original dataset, and on ambiguous examples, including the near-duplicates with high similarity that belong to more than one class.

Note that training a multi-label classifier incurs no extra cost compared to a single label classifier. However, our procedure for identifying near-duplicate pairs of texts, which we use to enrich an existing dataset, does incur additional cost, as mentioned in the Limitations section below.

4 Results

4.1 Identification of Ambiguous Near-duplicates

Analyzing the exact duplicates in the dataset shows that there are 81 texts that belong to more than one dialect. However, if we extend this analysis to include near-duplicate text pairs, the number of pairs that have different label sets increases sharply. Using the Levenshtein edit ratio with a cutoff at 0.8, we obtain 615,932 near-duplicate text pairs, and

[11] `https://huggingface.co/camembert-base`

[12] `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html`

6044 of those belong to different classes. Using the Manhattan similarity with a cutoff at 0.6, we obtain 576,722 near-duplicates, 3567 of which are ambiguous.

If we look at the most frequent edit operations, using both similarity measures, the most frequent edit operations by far are those that remove/add/replace punctuation or named entity tokens, all of which seem very unlikely to be dialectal in nature.

Manual inspection of a sample of ambiguous near-duplicates resulted in a disagreement rate, between the three annotators, around 15-20% on the common sets (i.e. 3 or 4 examples out of 20).

To illustrate the three categories we established for annotation purposes, consider the following two examples, where additions and deletions are within square brackets, and deletions are striked out.

An example of category 3 (potentially dialectal changes) is shown below. The first text is labelled CH, the second BE, and their edit ratio is 0.919. The first text contains a short phrase at the beginning that is completely absent from the second text. Note that, were this not the case, this example would likely have been annotated as category 1.

> [~~« Nous avons commencé », a-t-il ajouté. «~~]["]Des collaborateurs (du ministère) sont venus prendre leurs affaires personnelles[~~,~~] mais nous les avons mises sous scellés et nous ne laisseront personne entrer tant que la situation ne se normalise pas dans le pays [~~»~~][""], a indiqué l[~~’~~][']un des militants à [~~NE NE NE~~][l’agence Interfax]. NE[~~,~~] dont le centre est occupé depuis fin novembre par les manifestants pro[-]européens après la volte - face du pouvoir sur un rapprochement avec [~~NE~~][l’]NE NE au profit de la NE[~~,~~] est le théâtre de heurts violents entre manifestants radicaux et forces de [~~NE~~][l’]ordre depuis dimanche qui ont fait cinq morts.["]

Another example of category 3 is shown below. The first text is labelled CH, and the second BE, and their edit ratio is 0.924.

> [~~Une~~][L’]inconnu[~~e~~] subsiste quant aux réelles intentions de NE NE qui [~~NE~~][n’]a dit mot lundi des troupes russes [~~présent~~][déployé]es aux frontières de [~~NE~~][l’]NE. Il a en revanche une fois encore vilipendé le refus occidental de lui céder sur la fin de la politique d[~~’~~][']élargissement de [~~NE~~][l’]NE et le retrait de ses moyens militaires d[~~’NE de l’Est~~]['NE NE NE NE]. La NE a présenté ces exigences comme étant les conditions d[~~’~~][']une désescalade.

An example of category 2, where only the adverb “notamment” was deleted, is shown below. The Manhattan similarity of these texts is 0.973. The first text was labelled CH, and the second BE.

> Ce phénomène météorologique violent touche particulièrement les immenses plaines américaines. Sur des vidéos amateur prises vendredi soir, on voit ces immenses colonnes noires balayant le sol, illuminées par des éclairs intermittents. Le NE a [~~notamment~~]été balayé sur plus de 200 miles (320 kilomètres) par NE une des plus longues tornades jamais enregistrées aux NE, selon son gouverneur.

The manual annotation of samples of ambiguous near-duplicates indicates that between 6.25% and 11.25% of near-duplicates identified using the Levenshtein edit ratio exhibited *potentially* dialectal differences (i.e. category 3), though most of these were cases where one text had significant additions compared to the other, such that they might *potentially* contain dialectal markers. As noted above, the examples in category 3 might introduce some level of noise when we combine the labels of near-duplicates. As for “editorial” type changes (i.e. category 2), they represent between 0 and 8.75% of the samples.

As for the Manhattan similarity, the number of texts containing potentially dialectal differences was much higher, 36.25% and 46.25%. Additions with potential dialectal markers account for the vast majority of these. The number of “editorial” type changes was between 0 and 2.5%.

The two similarity measures identified different kinds of differences. The edit ratio was more effective for identifying slight, character-level changes between texts. The Manhattan similarity identified a large number of text pairs where one text had an additional trailing or leading sentence, which might indicate that the data should be split at sentence level rather than paragraph level, or that the paragraph splitting method could be improved.

The classification tests described in the next section were only carried out using the Levenshtein edit ratio as similarity measure, because there was a much higher proportion of potentially dialectal differences in the samples we annotated for the Manhattan similarity, and therefore a higher likelihood of introducing noise in the enriched dataset. We set the minimum similarity threshold at 0.8, which was the cutoff used when the near-duplicates were initially computed.

Using the Levenshtein edit ratio with a minimum of 0.8, we identified 615,932 pairs of similar texts. 6044 of these near-duplicate pairs had different sets of unique labels, and were therefore ambiguous. Among these 6044 pairs of ambiguous near-duplicates, there are 2901 unique texts. 74% of these have only one neighbour (i.e. they appear in only one pair), but the number of neighbours reaches as high as 241 for one of the texts. As for the number of new, unique labels each text will receive from its neighbours, 85% of texts receive only one new, unique label, but almost 15% receive two, and 10 texts (0.34%) receive three. There are also 8 texts (0.28%) that receive no new, unique labels.[13]

The distribution of the number of unique labels in the original dataset and the one we created by combining the labels of near-duplicates are shown in Table 2.

Labels/Text	Original	Combined
1	325,182	322,297
2	77	2,516
3	4	439
4	0	11

Table 2: Distribution of label counts according to the original labels and the combined labels.

The number of texts for each of the training, development, and test partitions we created using the original labels and the combined labels is shown in Table 3.

The most frequently confused pairs of dialects in the training sets, according to the original labels and our combined labels, are shown in Table 4.

Partition	Subset	Original	Combined
Train	Unambig	276,408	273,929
	Ambig	66	2545
Dev	Unambig	16,256	16,132
	Ambig	7	131
Test	Unambig	32,518	32,236
	Ambig	8	290

Table 3: Number of texts using original labels and combined labels.

Pairs	Original	Combined
(BE, FR)	54	1377
(CH, FR)	13	531
(BE, CH)	11	1381
(CA, FR)	0	19
(CA, CH)	0	18
(BE, CA)	0	13

Table 4: Most frequently confused classes in the training sets, using the original labels and the combined labels.

4.2 Classification

The classifiers were compared in the following ways. Using either the original labels of the dataset or the enriched (combined) labels resulting from our analysis of near-duplicates, we train classifiers on all the training data, and evaluate them on two subsets of the test data: ambiguous texts, that belong to more than one dialect, and unambiguous texts. In the single-label setting, ambiguous texts in the training set are represented by duplicating the text for each of its labels.[14] In this case, the model is evaluated on a test set that contains no in-class duplicates, as evaluating on in-class duplicates serves no purpose. In the multi-label setting, both the training and test data is represented in a multi-label format.

It is important to note that, on ambiguous test cases, single-label classifiers are obviously at a disadvantage, as they can only predict one class for a given text.

The results of this experiment are shown in Table 5 and Table 6 for the original labels and the combined labels respectively. When inspecting these results, it is important to remember that there

[13] These are texts that had *exact* duplicates with different labels. If such a text is in an ambiguous near-duplicate pair, and the other text's label set is a subset of this text's label set, then it will "give" one or more new labels to it, but will not receive any.

[14] We also tried training single-label classifiers without any in-class duplicates in the training data, but this made very little differences to the scores. We do not report these scores to avoid unnecessary confusion.

Test Set	Classifier	BE	CA	CH	FR	Average	Weighted
Unambig	Single-label	0.891	0.722	0.898	0.817	0.832	0.877
	Multi-label	0.894	0.670	0.903	0.826	0.823	0.882
Ambig	Single-label	0.533	-	0.571	0.400	0.376	0.490
	Multi-label	0.727	-	0.800	0.286	0.453	0.575

Table 5: Results using original labels: class-wise F1 scores, macro-average and weighted average. Note that there were no CA examples in the ambiguous test set.

Test Set	Classifier	BE	CA	CH	FR	Average	Weighted
Unambig	Single-label	0.891	0.644	0.901	0.818	0.813	0.878
	Multi-label	0.895	0.690	0.895	0.814	0.824	0.877
Ambig	Single-label	0.519	0.000	0.399	0.357	0.319	0.438
	Multi-label	0.815	0.000	0.800	0.561	0.544	0.739

Table 6: Results using combined labels: class-wise F1 scores, macro-average and weighted average.

are only 8 unique texts in the ambiguous test set using the original labels. None of these were labelled as CA, so the F1-score for this class is actually undefined.

On the enriched dataset (produced by combining labels of near-duplicates), the multi-label classifier produces similar accuracy to the single-label classifier on test cases that are unambiguous. The only class that displays significant difference is CA (up from 0.644 to 0.690), but that class is much smaller so it hardly makes a difference overall. On ambiguous examples, however, the macro-averaged F1-score increases from 0.319 to 0.544, for a 0.225 absolute gain (71% relative gain) on the combined data. Results on the original data are similar. Gains on the ambiguous test cases are smaller but still sizeable (+0.077 absolute, 20% relative gain), and accuracy on non-ambiguous test cases is hardly changed overall. To summarize, on unambiguous texts, the single-label and multi-label classifiers achieve similar accuracy, but on ambiguous texts, the multi-label classifier is considerably more accurate.

Note that we do not report overall performance (on both unambiguous and ambiguous examples), because it is almost identical to the performance on unambiguous examples, given that there is only around 1% of ambiguous examples with multiple labels. The main finding we want to highlight here is that multi-label classification improves accuracy on ambiguous examples without sacrificing accuracy on unambiguous ones, and at no extra cost in terms of modelling.[15]

It is important to note that the multi-label classifiers sometimes predict no dialects at all. Knowing that the test set contains no examples that belong to no classes, we could force the classifier to at least predict the most probable label, but we did not do this. The other option is simply to accept that the classifier does not assign sufficient probability to any dialect.

These results show that multi-label classifiers provide additional predictive information about ambiguous cases without degrading performance on unambiguous ones.

5 Discussion

Based on our analysis and experimental results, we argue that the analysis of near-duplicates and particularly ambiguous near-duplicates, should be an integral part of a dataset creation and validation pipeline, and should be described in the documentation for the collection. In the case of French variant identification, this analysis uncovered a number of features and issues with the dataset, such as differing formatting and typological conventions, which evade traditional deduplication, and may cause further problems, such as inconsistent named entity tagging, especially in terms of span. Another issue is that the segmentation of the original news stories into text fragments may differ between similar instances. This suggests that we may improve the near-duplicate detection and analysis by integrating sentence splitting into the processing, i.e. further split segments into individual sentences to detect more duplicates or near-duplicates.

It is important to remember that we do not believe that the ambiguity of duplicate text pairs and

[15] The only extra costs involved here are those of creating the enriched dataset, by combining labels of near-duplicates.

near-duplicates is unique to this dataset. In fact, we have observed similar issues in several datasets used for dialect identification in the past. However, further testing, e.g. on datasets in other languages, may be required to better establish the validity of the proposed approach.

Although we show that modelling dialect identification as a multi-label problem is useful, the proportion of ambiguous near-duplicates identified by our methods may seem small and therefore of little significance. If another dataset contained more ambiguous near-duplicates, or if a better method of identifying them were to be developed, the utility of this proposal would only be heightened. Note that in the dataset developed for the "True Labels" shared task at this year's VarDial evaluation campaign (Zampieri et al., 2023), the number of ambiguous examples was between 12% and 32%, which is much higher than the $\sim 1\%$ proportion we identified in the FreCDo dataset using near-duplicate analysis. In the proof of concept presented here, we limited ourselves to semi-automatic methods that exploit a sampling-based re-annotation protocol that is simple and inexpensive. Note also that further refinements to this protocol could reduce the number of disagreements between annotators on the sampled cases.

6 Conclusion

This contribution is motivated by the hypothesis that dialect identification is best addressed as a multi-label problem. By analyzing the similarity between instances in a four-class, French-language variant identification collection, we showed that there are a significant number of duplicates or near-duplicates with essentially the same surface representation and content, but differing reference labels. This is likely an artefact of the data acquisition pipeline, which focuses on the source of the data and provides a single label. By leveraging this finding, we were able to re-label some instances with multiple labels, and show that taking those into account by training a multi-label classifier produces a large increase in performance on the instances with multiple labels, while maintaining the performance on instances with a single label.

We argue that the analysis of ambiguous near-duplicates should be a standard in dataset creation and validation efforts, hopefully producing data that is labelled in a more informative way than by provenance alone.

Additional investigations may provide more insight on how to best represent dialect and variant classification. For example, we could encode multiple labels in a single-label model by encoding combinations of dialects as classes. Another possibility would be to formulate dialect identification as a word-level sequence tagging problem, identifying parts of a sentence that are dialectal markers, and parts that are not specific. This would likely require much more labelling, modelling and training effort.

Limitations

It must be acknowledged that identifying near-duplicates is a computationally intensive task, as it involves pairwise comparisons of a potentially large number of texts. For instance, processing 350K texts, as we did in this work, involves well over 100B comparisons. It took us about two days to compute the Levenshtein edit ratio matrix on this dataset, using a cutoff of 0.8 to speed up the dynamic program. This was done on a CPU server with large amounts of memory. Scaling this to larger datasets may require more efficient methods.

Furthermore, we have only experimented on dialects of the French language. Our method uses no tools that are specific to French, so that we believe that it may be useful on other dialect identification collections. However we cannot guarantee that any findings will generalize to all or any specific language or language families that have different properties.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful suggestions, which we hope have improved this paper.

References

Julien Abadji, Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2021. Ungoliant: An optimized pipeline for the generation of a very large-scale multilingual web corpus. In *Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-9) 2021. Limerick, 12 July 2021 (Online-Event)*, pages 1 – 9, Mannheim. Leibniz-Institut für Deutsche Sprache.

Noëmi Aepli, Antonios Anastasopoulos, Adrian-Gabriel Chifu, William Domingues, Fahim Faisal, Mihaela Găman, Radu Tudor Ionescu, and Yves Scherrer. 2022. Findings of the VarDial evaluation campaign 2022. In *Proceedings of the Ninth Workshop on NLP* *for Similar Languages, Varieties and Dialects*, pages 1–13, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Gabriel Bernier-Colborne, Serge Léger, and Cyril Goutte. 2022. Transfer learning improves French cross-domain dialect identification: NRC @ VarDial 2022. In *Proceedings of the Ninth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 109–118, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Cyril Goutte, Serge Léger, Shervin Malmasi, and Marcos Zampieri. 2016. Discriminating similar languages: Evaluations and explorations. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1800–1807, Portorož, Slovenia. European Language Resources Association (ELRA).

Mihaela Găman, Adrian-Gabriel Chifu, William Domingues, and Radu Tudor Ionescu. 2022. FreCDo: A large corpus for french cross-domain dialect identification. ArXiv:2212.07707.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. ArXiv:1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Margaret Mitchell, Alexandra Sasha Luccioni, Nathan Lambert, Marissa Gerchick, Angelina McMillan-Major, Ezinwanne Ozoani, Nazneen Rajani, Tristan Thrush, Yacine Jernite, and Douwe Kiela. 2022. Measuring data. ArXiv:2212.05129.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. In *Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019*, pages 9 – 16, Mannheim. Leibniz-Institut für Deutsche Sprache.

Chengwen Wang, Qingxiu Dong, Xiaochen Wang, Haitao Wang, and Zhifang Sui. 2022. Statistical dataset evaluation: Reliability, difficulty, and validity. ArXiv:2212.09272.

Marcos Zampieri, Kai North, Tommi Jauhiainen, Mariano Felice, Neha Kumari, Nishant Nair, and Yash Bangera. 2023. Language variety identification with true labels. ArXiv:2303.01490.