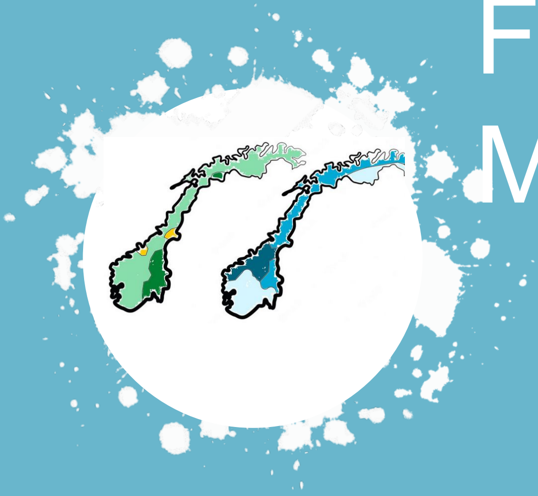
Identifying Token-Level Dialectal Features in Social Media



Jeremy Barnes, Samia Touileb, Petter Mæhlum, Pierre Lison

eman ta zabal zazu

Universidad del País Vasco / Euskal Herriko Unibertsitatea

HiTZ
Hizkuntza Teknologiako Zentroa
Basque Center for Language Technology

Media Futures

University of Oslo

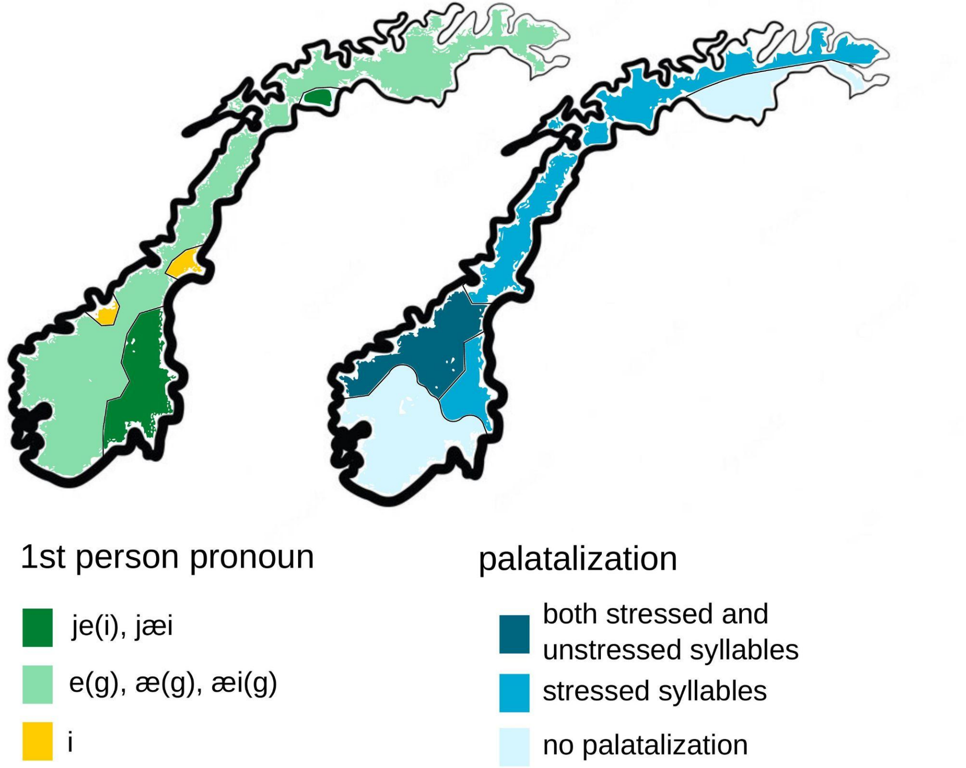
NR Norsk Regnesentral
Norwegian Computing Center

Dialectal variation in Norway

- Dialect (geolect, topolect) - a language variety the indicates *where* a speaker is from

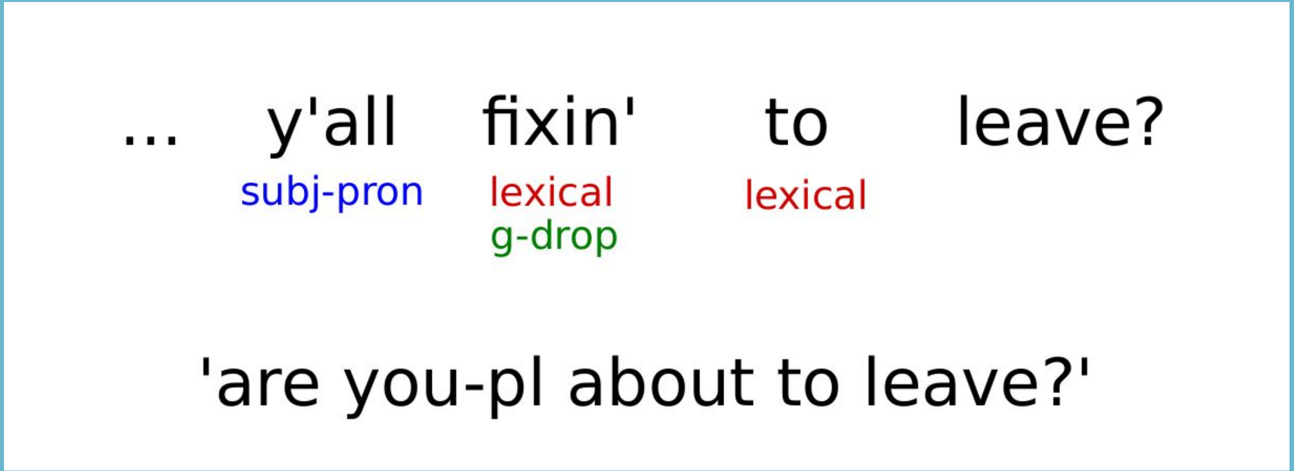
- In Norwegian, such variation is common and the distribution of dialectal features often overlap imperfectly, making it difficult to define dialect zones

- We avoid this controversial step by trying to predict dialect features directly



Motivation

- Most NLP datasets with dialectal variation:
  - Classification of dialects into predefined categories
  - Geolocation
  - Other NLP tasks performed on the dialectal data
- But we wanted to model the distribution of features themselves
  - In text, so we decided on token-level (although some of the features can span several tokens)



Tweet collection

- Collected 2,455 dialectal tweets (human annotated)
- First attempt using Twitter API + confining search to Norway didn't work
- Instead queried for dialect features
  - Found users who commonly used these features
  - Gathered their tweets and expanded by collecting tweets from their followers
- Annotators then filtered non-dialectal tweets

Annotated dialectal features

- Taken from features found in the Store Norske Leksikon

- Did not include features that are not observable in written texts
  - variation in toneme patterns
  - pronunciation of ‘L’

- Added a few more to deal with social media use

Annotated dialectal features

1. Subject pronoun
2. Object pronoun
3. Copula
4. Contraction
5. Palatalization
6. Present marker deletion
7. Apocope
8. Voicing
9. Vowel shift
10. Lexical variation
11. Demonstrative pronoun use
12. Shortening
13. Grammatical gender
14. Marked
15. H-V changes
16. Adjectival declension
17. Nominal declension
18. Verb conjugation
19. Functional words
20. Phonemic spelling
21. Interjection

Subject pronoun

(1) ... og **dem**/**de** blir aldrig eldre ...

'... and they never get older ...'

Copula

(2) Det **e**/**er** rart at ...

'It is weird that ...'

Apocope

(5) Æ e her for å **vinn**/**vinne**

'I am here to win' ...

Voicing

(6) Eg kommer ikkje **tebage**/**tilbake**

'I won't come back' ...

Annotation procedure

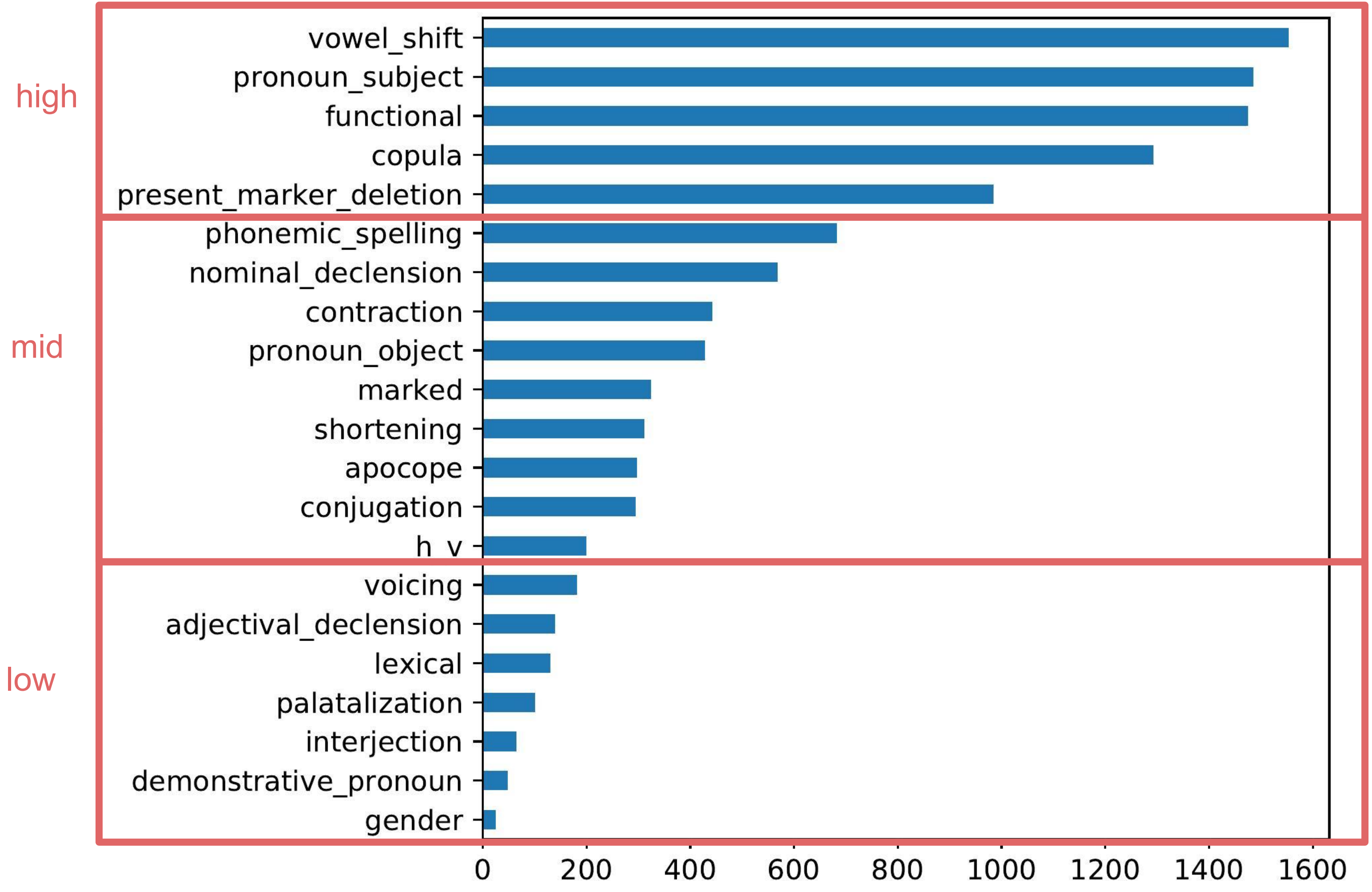
- Annotators - three hired research assistants with background in linguistics
- First 50 tweets annotated independently by two annotators
  - Group discussion to find sources of disagreement and refine guidelines
  - These were then adjudicated by a third annotator
- Regular group meetings to discuss difficulties and further refine guidelines

Dataset Statistics

	train	dev	test	total
number of tweets	1,655	300	500	2,455
number of tokens	40,483	7,563	12,597	60,643
average number of tokens per tweet	24.5	25.2	25.2	24.7
average number of annotations per tweet	4.5	4.4	4.5	4.5
average number of annotations per token	0.2	0.2	0.2	0.2
average number of labels per annotated token	1.2	1.2	1.2	1.2

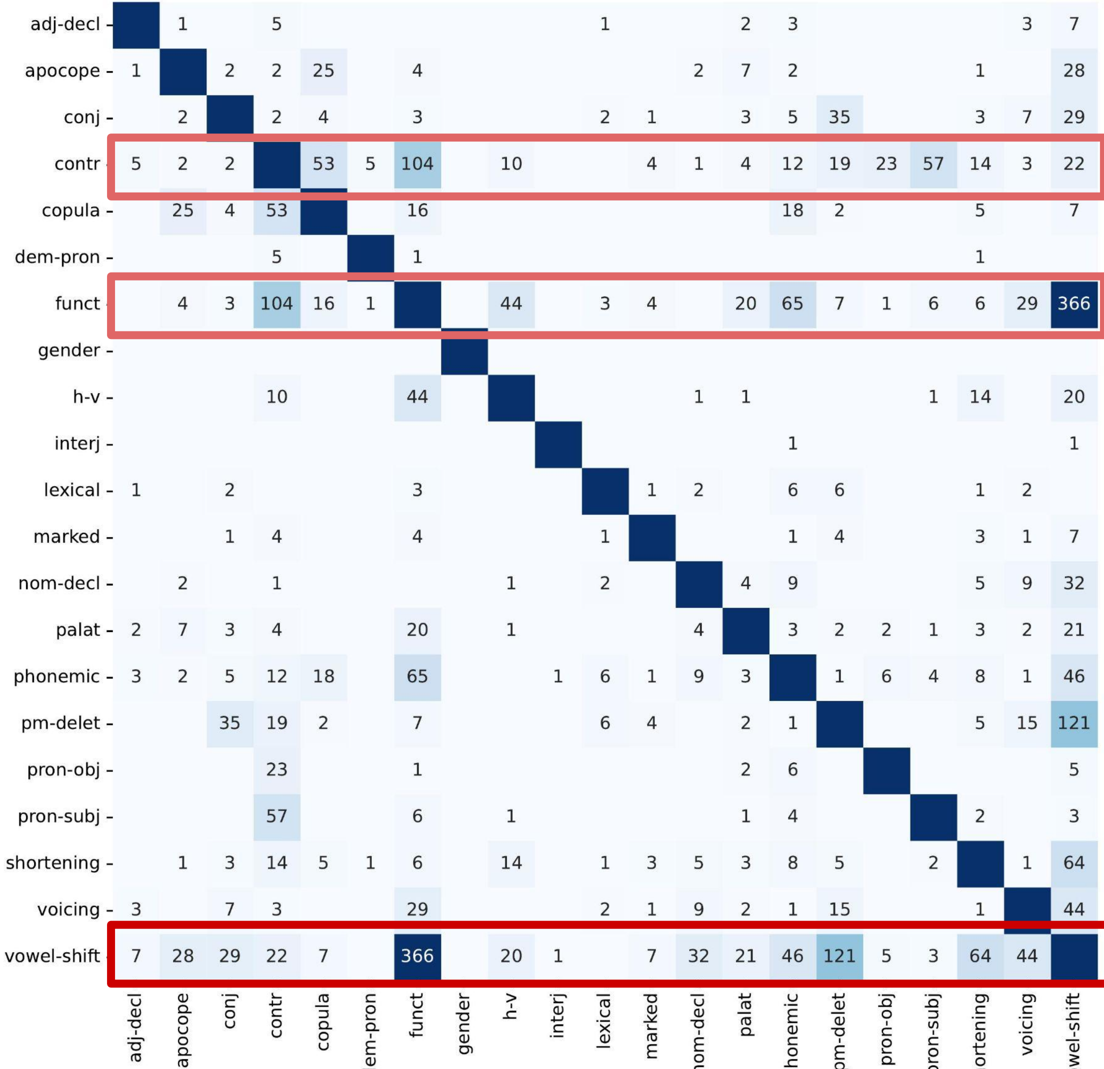
IAA: $\gamma = 0.63$, and $\gamma = 0.64$

Statistics



high
mid
low
vowel_shift
pronoun_subject
functional
copula
present_marker_deletion
phonemic_spelling
nominal_declension
contraction
pronoun_object
marked
shortening
apocope
conjugation
h_v
voicing
adjectival_declension
lexical
palatalization
interjection
demonstrative_pronoun
gender
0
200
400
600
800
1000
1200
1400
1600

Coocurrence of labels



Experimental setup

1. Rule-based labeling functions applied directly to test set

2. Weak labeling - Labeling functions applied to larger set of unlabeled tweets
   a. Use aggregation function (majority vote or HMM) to create silver data
   b. Train a BERT model on this silver data

3. Train a Pretrained Language Model directly on the available training data
   a. To determine the effect of the contextual embeddings, we also train an SVM using the non-contextualized word embeddings from the same BERT model as features

Rule-based weak learning

Particularly useful for low-resource settings: no training data, only expert knowledge

We created 39 hand-crafted functions of 3 types:

1. Heuristic functions
   a. Labels that can be detected programmatically
2. Lexicon functions
   a. Labels to be applied to relatively small, closed classes
3. Dictionary-based functions
   a. Labels that require checking with standard dictionaries

Heuristic examples

```python
def dem_pro(doc):
    i = 0
    while i < len(doc):
        tok = doc[i]
        if tok.pos_ in ["PROPN"]:
            if i-1 >= 0:
                prev_tok = doc[i-1]
                if prev_tok.text.lower() in ["han", "n", "hun", "hu", "ho", "a"]:
                    yield i-1, i, "demonstrative_pronoun"
            if i-2 >= 0:
                prevv_tok = doc[i-2]
                if prevv_tok.text.lower() in ["han", "n", "hun", "hu", "ho", "a"]:
                    yield i-2, i-1, "demonstrative_pronoun"
        i += 1
```

Lexicon-based

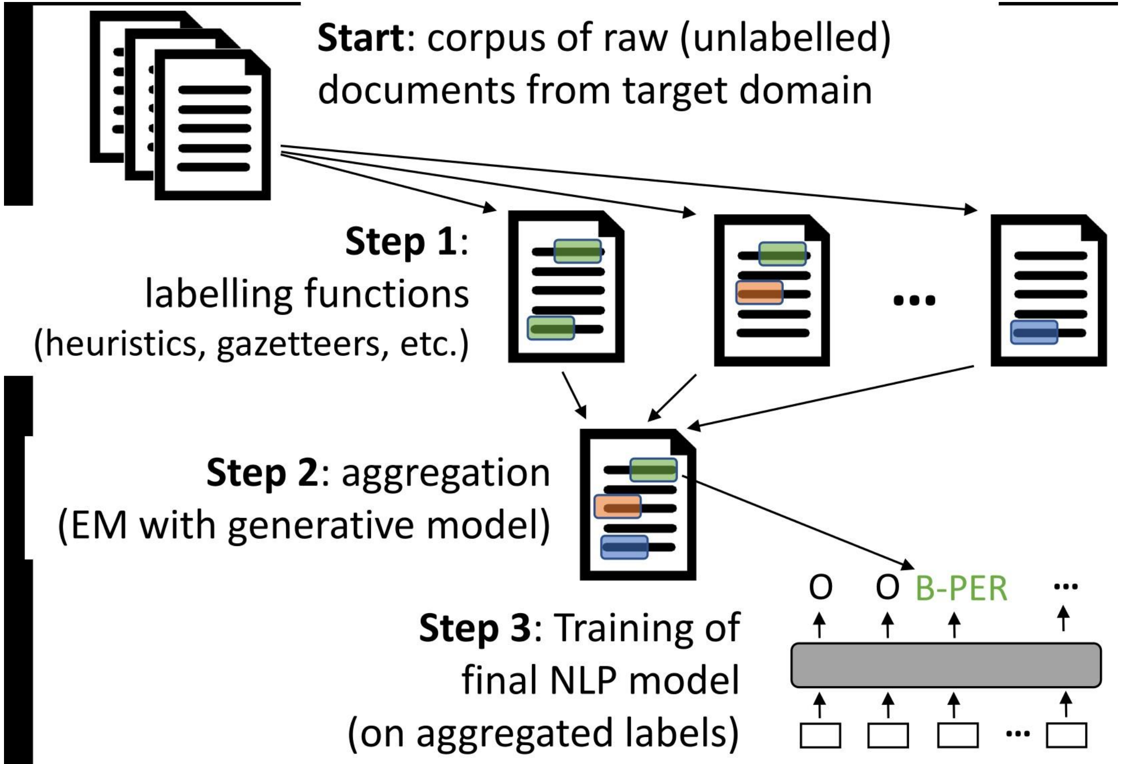
```python
def high_prec_pron_subj(doc):
    subj_pron = ["æ", "æg", "jæ", "jæi", "je", "ej"]
    i = 0
    while i < len(doc):
        tok = doc[i]
        if tok.text.lower() in subj_pron:
            yield i, i+1, "pronoun_subject"
        i += 1
```

Dictionary-based

```python
class VowelshiftAnnotator(SpanAnnotator):
    def __init__(self, name, bokmal, nynorsk):
        super(VowelshiftAnnotator, self).__init__(name)
        self.bokmal = bokmal
        self.nynorsk = nynorsk
        self.shifts = {"au": ["ø", "o"],
                       "jø": ["e"],
                       "øu": ["au"],
                       "æ": ["e", "a"],
                       "a": ["e"],
                       "jæ": ["e"],
                       "je": ["ø"],
                       "o": ["u"],
                       "ø": ["u", "o", "ei"],
                       "jo": ["y"],
                       "y": ["ø"],
                       "ei": ["e"],
                       "e": ["ei", "i"],
                       "ju": ["y"],
                       "øu": ["au"],
                       "å": ["o"]
                       }
    def replacenth(self, string, sub, wanted, n):
        where = [m.start() for m in re.finditer(sub, string)][n-1]
        before = string[:where]
        after = string[where:]
        after = after.replace(sub, wanted, 1)
        newString = before + after
        return newString
    #
    def apply_vowelshift(self, token):
        shifted = []
        for shift, shiftbacks in self.shifts.items():
            if shift in token:
                count = token.count(shift)
                for shiftback in shiftbacks:
                    for i in range(count):
                        shifted.append(self.replacenth(token, shift, shiftback, i))
                        shifted.append(token.replace(shift, shiftback))
        return shifted
```

Weak supervision

Gain recall by applying previous functions to unlabeled data.



Fully supervised models

Best case scenario: we have enough data to train our models

- BiLSTM
- NorBERT
- NB-BERT-base

Although the problem is multi-label, early experiments gave problems.

Therefore, we instead merge multi-labels and predict a total of 159 labels

['functional', 'contraction'] -> 'functional_contraction'

Results

Model	Dev		Test	
'Vowel shift'	3.7		4.4	
Labeling functions (MV-aggregated)	15.6		16.4	
NB-BERT fine-tuned on HMM-aggregated weak labels	14.1	$\pm$ 0.3	21.2	$\pm$ 0.7
NB-BERT fine-tuned on MV-aggregated weak labels	29.7	$\pm$ 0.6	33.3	$\pm$ 0.7
SVM + NB-BERT embeddings (gold labels)	45.5		47.7	
BiLSTM fine-tuned on train (gold labels)	38.5	$\pm$ 3.4	45.5	$\pm$ 0.0
NorBERT fine-tuned on train (gold labels)	42.0	$\pm$ 6.0	52.9	$\pm$ 1.3
NB-BERT fine-tuned on train (gold labels)	54.9	$\pm$ 0.8	58.4	$\pm$ 0.4

Error analysis

- the model confuses most labels with the label 'Ø'
- 'vowel shift' is regularly over-predicted
- Many of the labels are context sensitive - a non-contextualized baseline performs much worse

	Ø	adj-decl	apocope	conj	contr	copula	dem-pron	funct	gender	h-v	interj	lexical	marked	nom-decl	palat	phonemic	pm-delet	pron-obj	pron-subj	shortening	voicing	vowel-shift
Ø	10391	9	5	12	2	3	1	25		6	5	1	31	42		41	32	3	26	3	2	70
adj-decl	13	7	1	1															1			2
apocope	19		37	2		2								2			1			3		4
conj	25		4	9				1									5					3
contr	12				27	1		1		1	1		1				4		6	2		5
copula	5		1	1	1	257										2			3	1		4
dem-pron	3						6													1		
funct	28			3	2	3		205		8	1		2			8	7	2	7	10	1	52
gender	6																					2
h-v	4							5		29			1			1						2
interj	9										3											
lexical	16			1				1				1	2	2		1				1		16
marked	26	1		1				5					4	12						1		14
nom-decl	75							2						140		8				8		22
palat	3						1	2					1		3					1		5
phonemic	29	1		1				9					5	6		46	2		1	6		18
pm-delet	25		1	2		1		1								1	147					6
pron-obj	16						1	8								1		67	8		1	6
pron-subj	64				2	6	2	3			1							4	243			
shortening	11				2			6			1		2	2			1			19	1	4
voicing	3		1	1																	14	8
vowel-shift	98	2		7		1		39		2			24	14	2	18	16		3	8		296

Error analysis

- correlation between frequency in the training corpus and F1,
  - although there are outliers such as vowel shift.
  - This may be due to the range of heterogeneous contexts in which vowel shift can occur.
- Other labels such as functional or h-v are more difficult than expected, likely due to the number of possible forms
- Frequent multi-label tokens are correctly predicted, but the models struggle to generalize

Label	Precision	Recall	F_1
copula	**94.5**	**94.8**	**94.7**
pron. subj.	**82.9**	**74.3**	**78.4**
pm deletion	**72.4**	**79.9**	**76.0**
pron. obj.	**88.2**	63.8	**74.0**
h-v	67.4	69.0	68.2
functional	**71.2**	63.9	67.3
voicing	**73.7**	58.3	65.1
apocope	**75.5**	53.6	62.7
nom. decl.	66.0	55.6	60.4
dem. pro.	60.0	60.0	60.0
contraction	**77.1**	45.8	57.4
vowel shift	58.4	55.3	56.8
phon. spelling	40.7	36.5	38.5
shortening	41.3	35.2	38.0
adj. decl.	36.8	28.0	31.8
palatalization	**75.0**	20.0	31.6
interjection	30.0	25.0	27.3
conjugation	24.3	15.8	19.1
marked	6.7	8.0	7.3
lexical	50.0	3.0	5.7
gender	0.0	0.0	0.0

Conclusion and future work

- New dataset for token-level dialect feature detection in Norwegian
- For many of these labels, context is necessary to properly identify them

- We would like to / encourage others to use the data in order to
  - explore the distribution of these dialectal features in different online communities using the learned models.
  - predict regional dialects based on the token-level features

Thanks to:

- TekstHub for funding
- Alexandra Wittemann and Marie Emerentze Fleisje for annotation

Contact info

- https://github.com/jerbarnes/nordial

- jeremy.barnes@ehu.eus
- samia.touileb@uib.no
- pettemae@ifi.uio.no
- plison@nr.no