Corpus-based computational dialectology: Data, methods and results

Yves Scherrer
(Joint work with Olli Kuparinen and Aleksandra Miletić)

LTG, University of Oslo
14 September 2023

Corpus-based computational dialectology

Core idea: discover dialectal variation patterns in corpora

- What kind of variation-rich corpora are there?
  - Transcribed speech
  - User-generated content / social media
- How to make the texts in a corpus comparable?
  (Different texts will talk about different topics and thus contain topic-specific linguistic forms.
  → Disentangle topic-specific and dialect-specific forms.)
  - Unsupervised classification
  - Automatic normalization
- How to visualize and interpret the resulting variation patterns?

A multilingual dataset for dialect-to-standard normalization

A multilingual normalization dataset

Existing benchmarks and shared tasks:

- Historical text normalization (Bollmann, 2019)
  - 8 languages
- Social media normalization (MultiLexNorm, van der Goot et al. 2021)
  - 12 languages

Our work:

- Dialect-to-standard normalization
  - 4 languages: Finnish, Norwegian, Swiss German, Slovene
  - Reuse of existing datasets
  - Unified format and train/dev/test splits

Some examples

Finnish – SKN:

mä	oon	syänys	seittemän	silakkaa	aiva	niin	häntä	erellä
minä	olen	syönyt	seitsemän	silakkaa	aivan	niin	häntä	edellä

‘I have eaten seven herrings, that’s right, tail first’

Norwegian – NDC:

å	får	eg	sje	sjøra	vår	bil	før	te	påske
og	får	jeg	ikke	kjøre	vår	bil	før	til	påske

‘and I don’t get to drive our car until Easter’

Swiss German – ArchiMob:

ich	ha	das	ales	inere	kasette	won	ich	de	schlüssel	nüme	ha	dezue
ich	habe	das	alles	in einer	kassette	wo	ich	den	schlüssel	nicht mehr	habe	dazu

‘I have it all in a case for which I don’t have the key anymore’

Slovene – GOS:

se	zjemla	je	prpravlena	pugnujena	pa	ubdajlana	pa	puvlajčena
saj	zemlja	je	pripravljena	pognojena	pa	obdelana	pa	povlečena

‘because the soil is prepared, fertilised and tilled and harrowed’

Some numbers

Corpus	Creation	Texts	Speakers	Locations	Sentences	Tokens
SKN / fin	1960s–70s	99	99	50	41,407	630,665
The corpus has two levels of transcriptions. We use the simplified ones.						
NDC / nor	2006–2010	684	438	111	126,460	1,684,059
The provided sentence and word alignments between transcriptions and normalizations are broken. We (hopefully) fixed them. *https://github.com/Helsinki-NLP/ndc-aligned*						
ArchiMob / gsw	1999–2001	43	43	~22	80,228	581,974
		6	6	5	10,183	82,658
The corpus contains a total of 43 texts, but only 6 of them are manually normalized. We only use those for normalization-based experiments.						
GOS / slv	2008–2010	24	36	10	8,621	84,199
The corpus contains 287 texts, but we only extracted those with >30% non-standard tokens.						

Case studies

	SKN fin	NDC nor	ArchiMob gsw	GOS slv	Normalization layer used
1. Topic modelling	✓	✓	✓	✗	✗
2. Character alignment	✓	✓	✓	✗	✓
3. Speaker embeddings	✓	✓	✗	✗	✓
(4. Normalization evaluation)	✓	✓	✓	✓	✓

Topic modelling

Topic modelling

A traditional method of text mining:

- Each document in a collection is represented by a distribution over topics.
- Each topic is defined by a distribution over words.

Concretely:

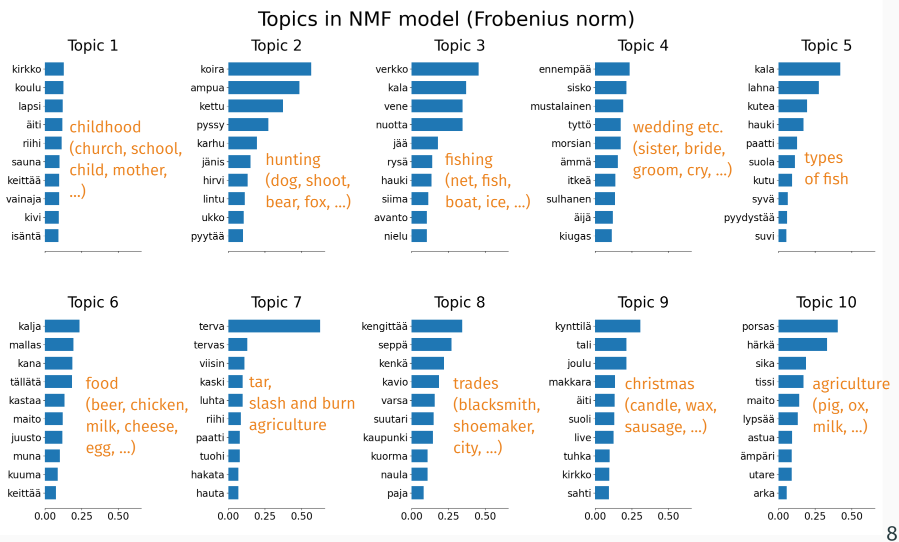
- Creates term-document matrix from plain text
- Uses some dimensionality reduction method (LDA, NMF, PCA, ...) to infer topic and word distributions
- Number of topics needs to be given as parameter

The typical usage focuses on semantic topics:

- Lemmatization to remove morphological variation
- Stopword removal

Semantic topic models

Example: SKN, normalized, lemmatized, no stopwords



Structural topic models

The typical usage focuses on **semantic topics**.

But instead of semantic topics, we are interested in **structural topics** (phonetics, morphology).

How to “remove semantics” from text?

- No normalization
- No lemmatization
- No stopword removal
- Chop up the words
  - Character n-grams
  - Morfessor subwords

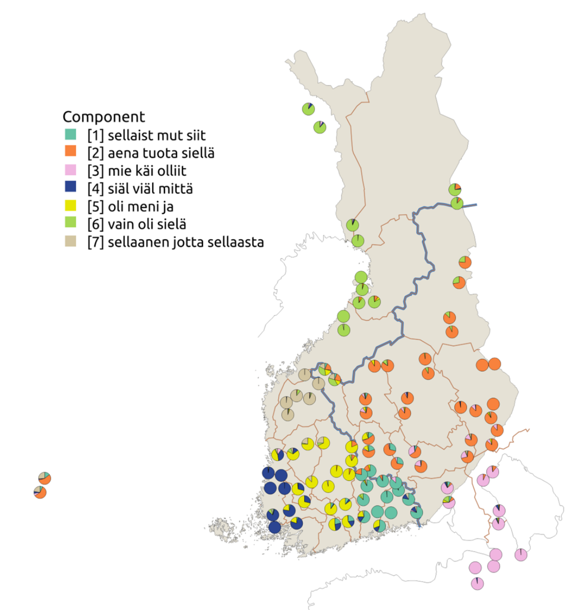
Words	mini eltere händ es zwäifamiliehuus ghaa
Morfessor	mini eltere händ e s zwäi familie huus ghaa
Bigrams	_m mi in ni i_ _e el lt te er re e_ _h hä än nd d_ _e es s_ _z zw wä äi if fa am mi il
Trigrams	_mi min ini ni_ _el elt lte ter ere re_ _hä hän änd nd_ _es es_ _zw zwä wäi äif ifa
Fourgrams	_min mini ini_ _elt elte lter tere ere_ _hän händ änd_ _es_ _zwä zwäi wäif äifa ifam
Gloss	‘my parents had a two-family house’

Experiments

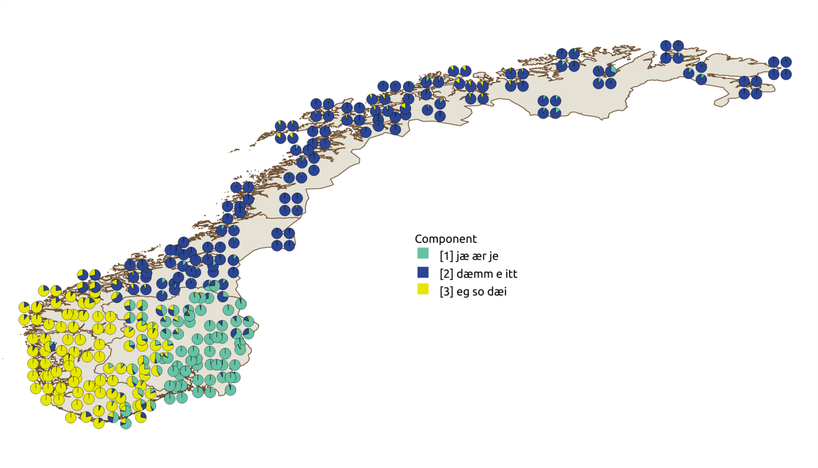
- 3 corpora
- 2 algorithms
  - LDA (Latent Dirichlet allocation; Blei et al. 2003)
  - NMF (Non-negative matrix factorization; Lee & Seung 1999)
- 5 word segmentation settings
- 8 numbers of components (3–10)

I’ll show you 3 of the 240 maps…

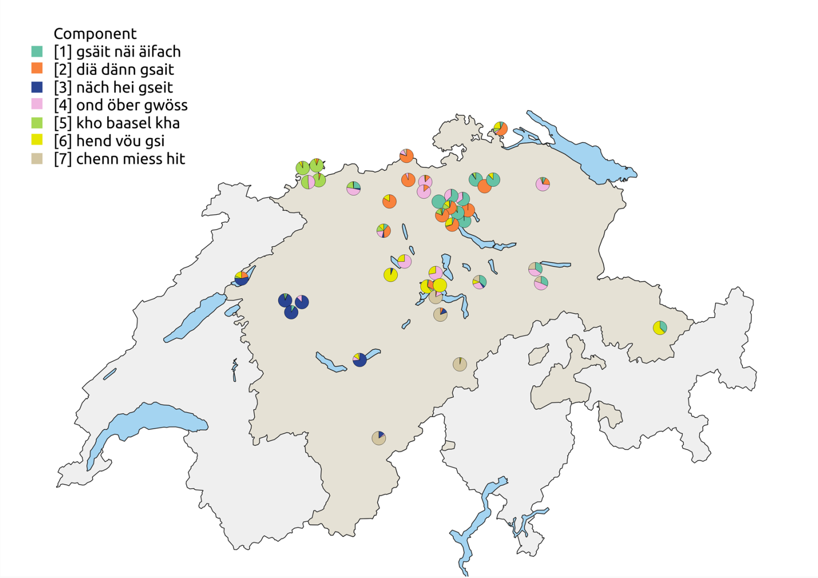
SKN | NMF | full words | 7 components



NDC | LDA | full words | 3 components



ArchiMob | NMF | Morfessor | 8 components



Conclusions

- A simple approach that works surprisingly well
  - Topics are mostly defined by function words and devoid of semantic content
  - Topics match well with existing dialect classifications
- Methods:
  - Both NMF and LDA give reasonable results for dialect data
  - (and so would probably any other dimensionality reduction algorithm)
- Word segmentation:
  - Using full words works fine in most cases
  - N-gram decomposition does not provide major benefits
  - Somewhat surprisingly, Morfessor works best for Swiss German

Conclusions

- Dimensionality reduction is commonly used in atlas-based dialectometry
  - We try to bring together two completely different research areas
  - The method provides a significant simplification of the dialectometric workflow
- Eisenstein et al. (2010) uses topic models to detect dialect variation in US English tweets
  - Simultaneously learns geographic and semantic topics
  - In US English, dialectal variation and topic variation are both mainly expressed on the lexical level
  - Our data permits a much simpler approach

Olli Kuparinen & Yves Scherrer (accepted): Corpus-based dialectometry with topic models. In *Journal of Linguistic Geography*.

Character alignment

Character alignment in dialect-to-standard normalization

Data: Utterances with two transcription layers (phonetic and orthographic transcription)

Task: Align the two layers character by character

```
_ j a _ s e _ k u a l i _ s i ä l ä _ , _ A m e r i i k k à s A _
| | | | | | | | | | | | | | | | |\ \ \ \ \ \ \ \ \ \| |/ / /| | |
_ j a _ s e _ k u o l i _ s i e l l ä _ , _ A m e r i k a s s a _
```

Why is this interesting?

- Character alignment methods have been used in various fields for various purposes, but not thoroughly compared
- Character alignments are required to train some automatic normalization systems
- The mappings between phonetic and orthographic strings are a valuable source for corpus-based dialectology

Character alignment methods

Various traditions and applications:

- Dialectometry (Heeringa et al. 2006, Wieling et al. 2009)
  - Levenshtein distance
  - Vowel-sensitive Levenshtein distance
  - Levenshtein distance with PMI-based edit weights
- Grapheme-to-phoneme conversion
  - Stochastic memoryless transducers (Ristad & Yianilos 1998, Jiampojamarn et al. 2007)
  - HMMs
- Cognate identification (Mann & Yarowsky 2001)
- Character-level statistical machine translation (Tiedemann 2009)
  - GIZA++ (Och & Ney 2000)
  - fast_align (Dyer et al. 2013)
  - eflomal (Östling & Tiedemann 2016)

Experimental setup

Three datasets:

Corpus	Language	Documents	Locations	Sent/doc	Words/sent
SKN	Finnish	99	50	418	15
NDC	Norwegian	438	111	289	13
ArchiMob	Swiss German	6	5	1697	8

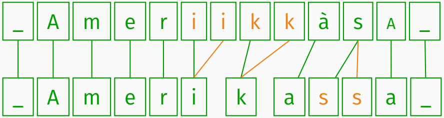
Eight alignment methods:

Method	Training	Disjoint alphabet	Swaps	1-to-n links
Levenshtein	Untrained	✗	✗	✗
Levenshtein+PMI	Corpus-level	(✓)	✗	✗
Unigram transducer	Doc-level	✓	✗	✗
Bigram transducer	Doc-level	✓	(✓)	(✓)
GIZA	Doc-level	✓	✓	✓
fast_align	Doc-level	✓	✓	✓
eflomal	Doc-level	✓	✓	✓
eflomal+priors	Corpus-level	✓	✓	✓

Experimental setup

Extension 1: add adjacent identicals

- Applied to Levenshtein-based and unigram methods



Extension 2: symmetrization with *grow-diag-final-and*

- Standard practice for SMT word aligners
- For consistency applied to all methods

Evaluation

Ideally, we should compare the output of the automatic alignment methods with gold alignments. **In practice**, we do not have gold alignments on character level in our datasets.

Instead, we gather statistics about four phenomena that we consider **“undesirable”** for the given task:

U-src the proportion of unaligned source characters,

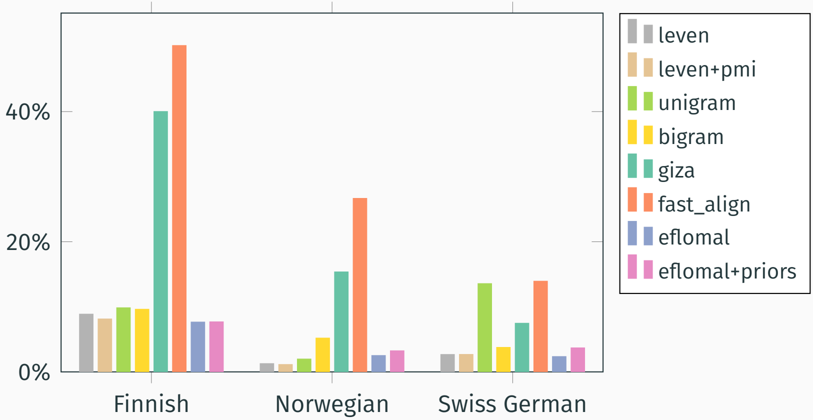
U-tgt the proportion of unaligned target characters,

V-C the proportion of vowel-to-consonant and consonant-to-vowel alignments (disregarding semi-vowels, nasals, laterals and suprasegmentals),

X the proportion of crossing alignment pairs (swaps / metatheses).

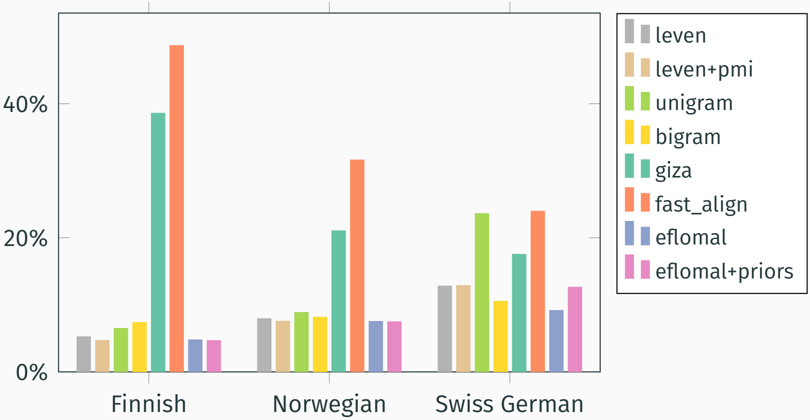
Lower values are indicative of better alignment quality.

Results – Unaligned source characters



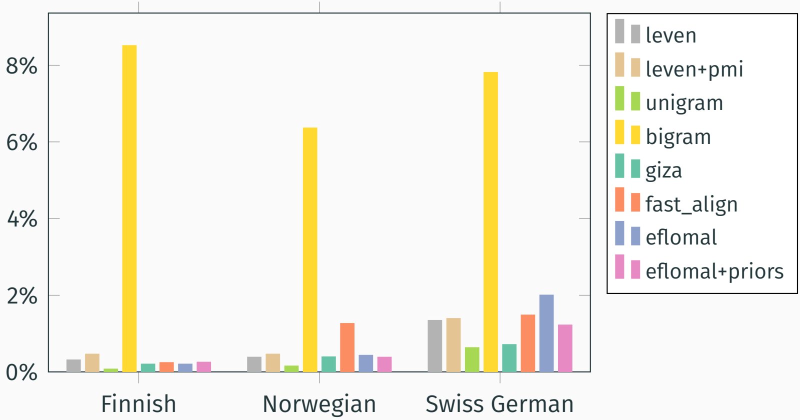
- GIZA++ and fast_align perform poorly
- Inconsistent results with unigram

Results – Unaligned target characters



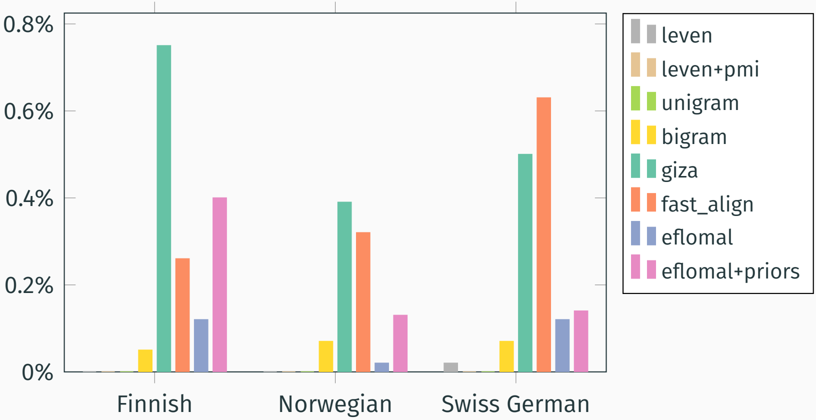
- GIZA++ and fast_align perform poorly
- Inconsistent results with unigram

Results – Vowel-consonant alignments



- The bigram transducer produces unintuitive alignments
- Best results for unigram transducer

Results – Crossing alignments (swaps)



- leven(+pmi) and unigram do not allow crossing alignments (or only through symmetrization)
- Best non-zero results by bigram and eflomal

Conclusions

- GIZA++, fast_align and the bigram transducer produce unintuitive results and cannot be recommended
- Corpus-level training is not better than document-level training
  - See unigram vs leven+pmi, eflomal vs eflomal+priors
  - We expected this to be useful for corpora with short documents (SKN, NDC)

Recommendations:

- Eflomal
  - Allows swaps, can handle disjoint alphabets
- Unigram transducer
  - Best phonological consistency
- Levenshtein distance
  - Untrained, most efficient

Dialectological analysis

"The mappings between phonetic and orthographic strings are a valuable source for corpus-based dialectology."

How exactly?

1. Align transcriptions and normalizations character by character
   - Done, we now know what works best
2. Merge adjacent alignment pairs to n-gram pairs
3. Collect counts and conditional probabilities of target n-grams
   - This is the standard phrase extraction process of SMT
   - Example from Swiss German:

Document 1048:

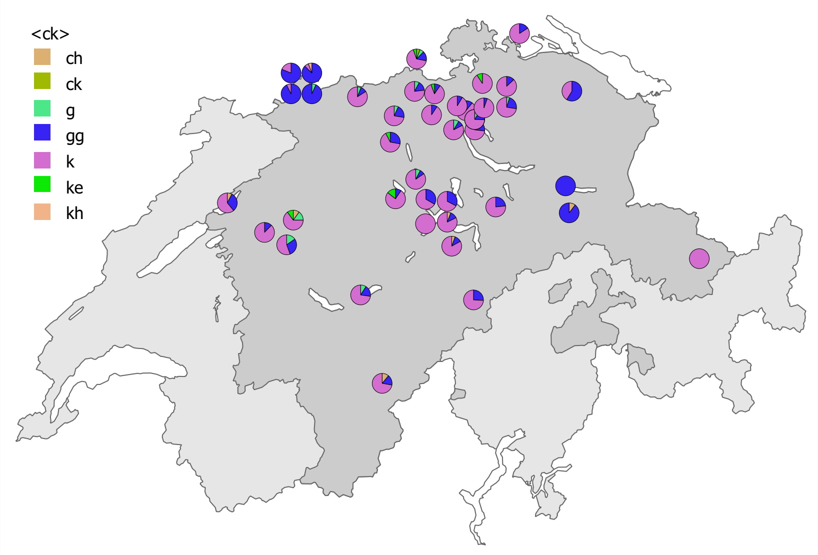
					P(Ph|Or)
ch	ck	2028	52	5	0.09615
gg	ck	176	52	46	**0.88462**
k	ck	122	52	1	0.01923

Document 1244:

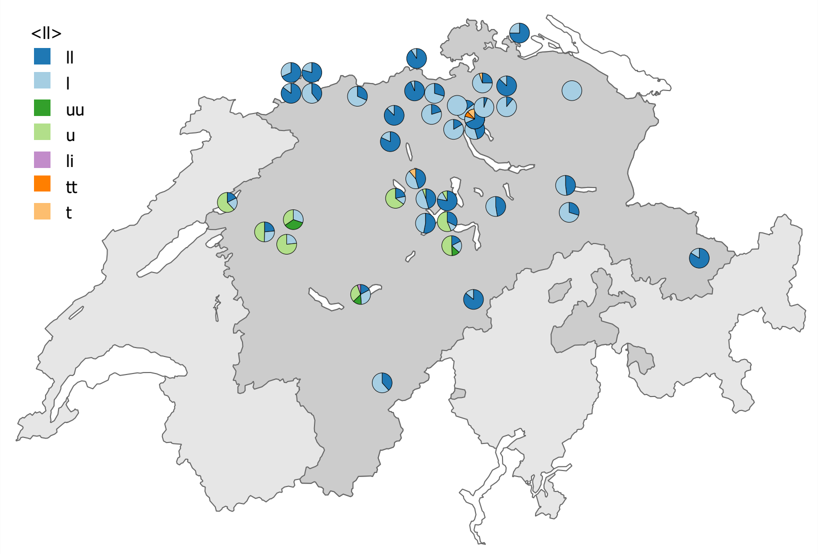
					P(Ph|Or)
g	ck	3126	47	2	0.04256
gg	ck	51	47	2	0.04256
k	ck	566	47	43	**0.91489**

4. Select target n-grams, visualize distribution of variants

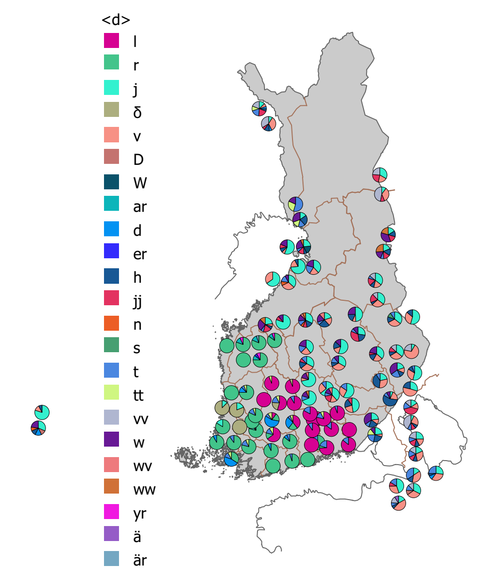
ck in ArchiMob



ll in ArchiMob



d in SKN



Speaker embeddings

Multilingual machine translation

Multilingual NMT: one model, several source and target languages indicated with so-called **language labels**:

<FROM_ES> <TO_FR> Visitaré a los niños.	Je viendrai voir les enfants.
<FROM_EN> <TO_ES> You did well, you did very well.	Bien hecho. Genial.
<FROM_ES> <TO_EN> Llegaremos enseguida.	We will be arriving soon.
<FROM_FR> <TO_ES> C’est la voix de notre âme qui parle.	Es la voz del alma que habla.

We use this idea to inform the normalization model about the source dialect:

- Train NMT on dialect-to-standard normalization task
- Full-sentence Transformer with subword segmentation
- Many-to-one setup: many source dialects, one standardized target variety
- Speaker IDs appended as source labels

<SKN34a> mie poikain kans olen kahen teäl	minä poikani kanssa olen kahden täällä

Speaker label embeddings

After model training, we inspect and analyze the embeddings of the speaker labels.

- Does the normalization model learn which speakers come from the same area?
- Do the embeddings encode information about the dialect areas?

O. Kuparinen & Y. Scherrer (2023): Dialect representation learning with neural dialect-to-standard normalization. In *Proceedings of VarDial*.

Inspired by a similar study on Japanese:

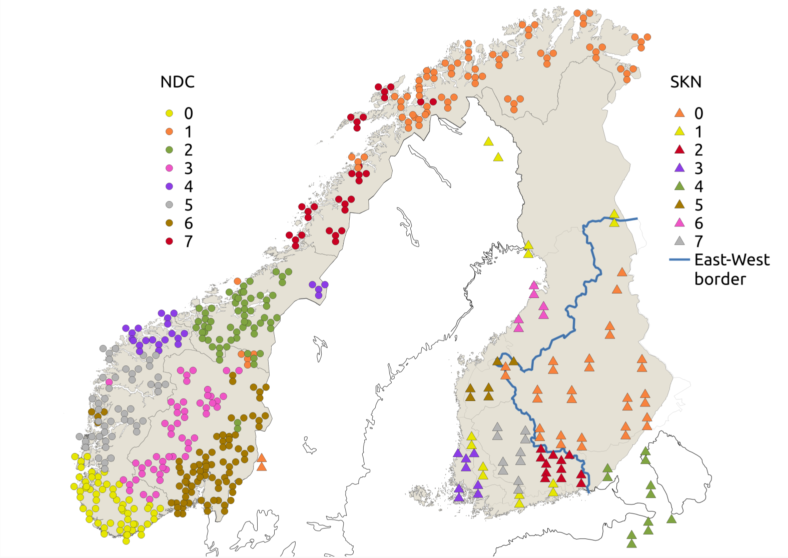
K. Abe, Y. Matsubayashi, N. Okazaki, and K. Inui (2018): Multi-dialect neural machine translation and dialectometry. In *Proceedings of PACLIC*.

Some inspiration from dialectometry

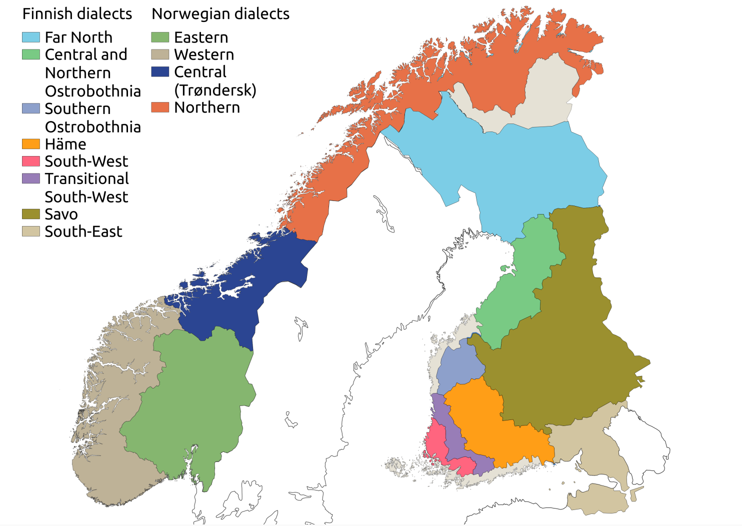
A rough sketch of a dialectometrical experiment:

1. Build a vector characterizing each dialect
    - Data vector: each dimension represents a linguistic item, the value marks presence or absence
    - Distance/similarity vector: each dimension represents the distance/similarity to one other dialect
    - We just use our speaker embedding vectors here.
2. Project these high-dimensional vectors into a lower-dimensional space
    - Cluster analysis
    - Multidimensional scaling, principal component analysis, factor analysis, ...
3. Assign each value a color and plot on a map

Hierarchical clustering (Ward, 8 clusters per language)

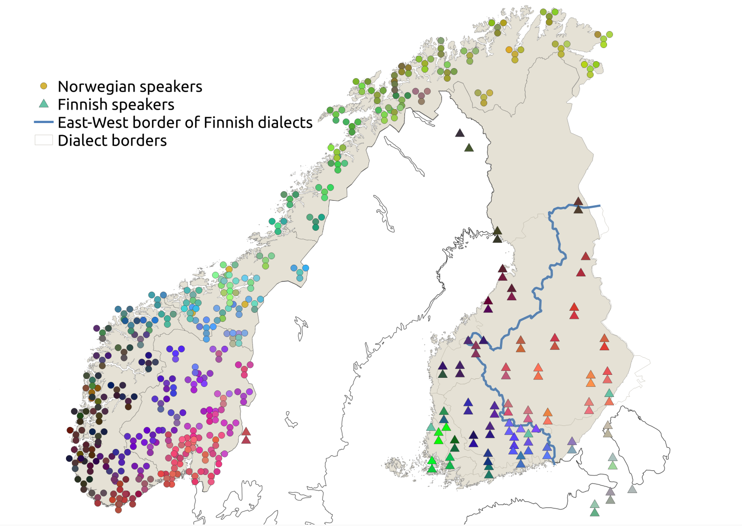


Expected dialect classifications



Norwegian division based on Hanssen (2010–2014).
Finnish division based on Itkonen (1989).

Principal component analysis (3 dimensions → RGB)



Explained variance: 9% for Norwegian, 14% for Finnish.

Discussion

- The speaker labels learned for the normalization task reflect the dialectal (or geographic) origin of the speakers.
  - The model could also have ignored them completely.
  - The model could also have used them for something else.
- Speakers from the same place are almost always placed in the same cluster.
- The major dialect borders are visible in the embeddings.
- The explained variance of the PCA is low. The exact reasons for this remain to be investigated.
- It would be interesting to see **when** and **how** the normalization model makes most use of the labels. This could be achieved by analyzing the attention weights.

Conclusions and perspectives

Conclusions

	SKN fin	NDC nor	ArchiMob gsw	GOS slv	Normalization layer used
1. Topic modelling	✓	✓	✓	✗	✗
2. Character alignment	✓	✓	✓	✗	✓
3. Speaker embeddings	✓	✓	✗	✗	✓
(4. Normalization evaluation)	✓	✓	✓	✓	✓

Perspectives

Dialect-to-standard normalization:

- Implicit assumption:
  Normalization improves downstream task performance
  - Bollmann (2019), van der Goot et al. (2021)
- Is this also true for dialect normalization?
- Is this the right way to go?
  Do we still want such pipeline approaches?
- If not, can we get comparable dialect representations from end-to-end systems?

Perspectives

Reducing variation:

- Different methods reduce different types of variation:
  - Normalization reduces phonetic and spelling variation
  - Lemmatization reduces morphological variation
- Can/should we combine normalization with lemmatization?
  - Normalization – no obvious target standard:
    Low Saxon $\rightarrow$ nds-de, nds-nl, deu, nld?
  - Lemmatization without normalization:
    singsch $\rightarrow$ singe / singä / singa / singu
  - Should we attempt **cross-lingual lemmatization**?
    Low Saxon $\rightarrow$ deu, nld
    Occitan $\rightarrow$ cat, fra