Character Alignment Methods for Dialect-to-Standard Normalization

Yves Scherrer
Department of Digital Humanities
University of Helsinki
Helsinki, Finland
yves.scherrer@helsinki.fi

Abstract

This paper evaluates various character alignment methods on the task of sentence-level standardization of dialect transcriptions. We compare alignment methods from different scientific traditions (dialectometry, speech processing, machine translation) and apply them to Finnish, Norwegian and Swiss German dialect datasets. In the absence of gold alignments, we evaluate the methods on a set of characteristics that are deemed undesirable for the task. We find that trained alignment methods only show marginal benefits to simple Levenshtein distance. On this particular task, *eflomal* outperforms related methods such as GIZA++ or *fast_align* by a large margin.

1 Introduction

In recent research, a wide range of character transduction tasks (Wu and Cotterell, 2019) have been studied, such as modernization of historical spellings, correction of non-standard spellings in user-generated content, lemmatization, or grapheme-to-phoneme conversion (G2P). While most work aims at creating and improving generative models that produce the target representation given its source representation, we focus in this paper on the task of aligning characters when both representations are given. Character alignment is a key step in the training pipeline of certain character transduction models such as those based on the statistical machine translation (SMT) paradigm.

Other lines of research have been concerned with finding distances between strings, e.g., to compare different dialectal pronunciations (dialectometry) or to identify cognate pairs in corpora of related languages. While most research in these areas focuses on finding the optimal distance metric for a given task, we rather look at the alignments produced by these distance metrics here. Indeed, character alignments are a by-product of distance computations and readily available.

In most cases, both character transduction and distance computation are performed at word level, i.e., one word at a time. However, we argue that it is beneficial to carry them out at sentence level (if appropriate corpora are available) to enable contextual disambiguation, to avoid relying on pre-existent tokenization and to capture assimilation effects at word boundaries.

In this work, we focus on sentence-level standardization of dialect transcriptions. We compare character alignment methods from different scientific traditions and apply them to corpora of transcribed dialectal speech from three languages, namely Finnish, Norwegian and Swiss German. In the absence of gold alignments, we evaluate the alignment methods on a set of characteristics (e.g., the amount of vowel-to-consonant alignments) that are deemed undesirable for dialect-to-standard character alignment.

2 Alignment Methods

Character alignment methods have been proposed for different purposes in different fields, but all of them can be meaningfully applied to sentence-level dialect-to-standard alignment.

Dialectometry The core idea of dialectometry is to obtain abstract representations of dialect landscapes from large numbers of individual features (see e.g. Nerbonne and Kretzschmar, 2003; Wieling and Nerbonne, 2015). One way to achieve this is to compute distances between phonetic transcriptions of a given word in different dialects, followed by aggregating the distances over all words of the dataset. Levenshtein distance (Levenshtein, 1966) is generally used as a starting point for such undertakings, but over the years, several extensions have been proposed, such as vowel-sensitive Levenshtein distance, or the possibility to learn the edit weights from a corpus (Heeringa et al., 2006). While most work focuses on the obtained distance

Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 110–116

	docs	sents	words	sents/doc	words/doc	words/sent	chars/word	$\|C_\cup\|$	$\|C_\cap\|$
SKN	99	51,254	841,859	518	8504	16.4	5.7	243	70
NDC	648	145,961	1,937,905	225	2991	13.3	4.4	93	84
ArchiMob	6	11,959	93,450	1993	15575	7.8	5.3	49	33

Table 1: Key figures of the three datasets. The table shows the absolute number of documents, sentences and words, as well as the average number of sentences per document, words per document, words per sentence, and characters per word. $|C_\cup|$ refers to the size of the union of dialectal and standardized character sets, $|C_\cap|$ to their intersection.

values and their correlation to existing dialectological findings, Wieling et al. (2009) specifically evaluate the alignments obtained by such distance metrics.

Cognate identification Similar distance metrics have been employed for identifying cognates in large corpora of related languages (e.g. Mann and Yarowsky, 2001; Kondrak and Sherif, 2006).

Grapheme-to-phoneme conversion Many text-to-speech systems contain a G2P component that turns words spelled in conventional orthography into sequences of phoneme symbols that correspond to the actual pronunciation of the word. Before neural sequence-to-sequence models were used, the standard approaches for G2P relied on stochastic transducers or HMMs with weights learned from training data using expectation-maximization (EM). For example, Ristad and Yianilos (1998) introduced a stochastic memoryless transducer. Jiampojamarn et al. (2007) proposed an extension to this model that also covers multi-character graphemes and phonemes.

Statistical machine translation Word alignment is a crucial ingredient of the SMT paradigm introduced at the beginning of the 1990s (Brown et al., 1993). GIZA++, an open-source aligner that has become standard over the years, uses a pipeline of increasingly complex word alignment models (Och and Ney, 2000). Follow-up work such as *fast_align* (Dyer et al., 2013) and *eflomal* (Östling and Tiedemann, 2016) introduced simpler, faster and less memory-hungry alignment approaches with only minor sacrifices in accuracy.

Although designed to align words in sentence pairs, the word alignment models can also operate on single characters. This approach has become popular as character-level SMT and has been used e.g. to translate between closely-related languages (Tiedemann, 2009) or for historical text modernization (Scherrer and Erjavec, 2013).

3 Data

We use existing dialect corpora from Finnish, Norwegian and Swiss German for our experiments:

SKN – Finnish The Samples of Spoken Finnish corpus (*Suomen kielen näytteitä*, hereafter SKN) (Institute for the Languages of Finland, 2021) consists of 99 interviews conducted mostly in the 1960s. It includes data from 50 Finnish-speaking locations, with two speakers per location (with one exception). The interviews have been transcribed phonetically on two levels of granularity (detailed and simplified) and normalized manually by linguists. We use the detailed transcriptions here.[1]

NDC – Norwegian The Norwegian Dialect Corpus (Johannessen et al., 2009, hereafter NDC) was compiled between 2006 and 2010 in the context of a larger initiative to collect dialect data of the North Germanic languages. Typically, four speakers per location were recorded, and each speaker appears both in an interview with a researcher and in an informal conversation with another speaker. The recordings were transcribed phonetically and thereafter semi-automatically normalized to the Bokmål standard.[2]

ArchiMob – Swiss German The ArchiMob corpus (Scherrer et al., 2019) consists of oral history interviews conducted between 1999 and 2001. It contains 43 phonetically transcribed interviews, but only six of them were normalized manually. We only use these six documents for our experiments.

Some quantitative information about the datasets is given in Table 1. One may note that ArchiMob has the longest documents and NDC the shortest. On the other hand, ArchiMob has the shortest sentences. SKN has the most detailed transcriptions

[1]Details about the availability of the corpora are given in Table 6 in the appendix.

[2]The publicly available phonetic and orthographic transcriptions are not well aligned. We use (and provide) an automatically re-aligned version of the corpus, cf. Table 6.

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'
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'
ArchiMob: aber meer hènd den furchpaari finanzijelli schwirigkaite gchaa aber wir haben dann furchtbare finanzielle schwierigkeiten gehabt 'but then we had terrible financial difficulties'

Table 2: Example sentence pairs from the three datasets. The top row presents the phonetic dialectal transcription, the middle row the standardized version, and the bottom row provides an English gloss. Although the number of transcribed and standardized tokens is the same in the three shown examples, we do not presuppose this for our experiments. Likewise, we do not presuppose that the data is aligned at token level.

and therefore the largest character vocabulary. Table 2 provides some example sentences.

4 Experimental Setup

4.1 Data Preparation

We reformat the three datasets in such a way that the utterances are split into sequences of characters and that the word boundaries are marked with a special symbol (_), as exemplified in Figure 1.

```
_ å _ f å r _ e g _ s j e _ s j ø r a _
_ o g _ f å r _ j e g _ i k k e _ k j ø r e _
```

Figure 1: Tokenized example sentence, dialectal transcription above and orthographic normalization below.

Since all alignment methods are unsupervised and there are no gold alignments for evaluation, we do not split the data into training and test sets. We train one alignment model per document, using the dialectal transcriptions as the source and the orthographic normalizations as the target.

4.2 Alignment Methods

We apply the following alignment methods:

- Levenshtein distance with default edit operation weights (`leven`).
- Weighted Levenshtein distance using PMI scores as edit operation weights (Wieling et al., 2009). We extract the PMI scores from the concatenation of all Levenshtein-aligned documents of a corpus (`leven-pmi`).
- Stochastic memoryless unigram transducer with weights trained iteratively on single documents (Ristad and Yianilos, 1998) (`unigram`).[3]
- Stochastic memoryless bigram transducer (Jiampojamarn et al., 2007); we override the default settings and allow deletions and insertions, as well as mappings of two bigrams (`bigram`).
- `GIZA++` with default parameters.
- `fast_align` with default parameters.
- `eflomal` with default parameters.
- *eflomal* can extract prior alignment probabilities from a previously aligned dataset to initialize a new alignment model. We concatenate all documents of a corpus to extract the probabilities (`eflomal-priors`).

To summarize, our experiments cover one untrained model (`leven`), five models trained on document-level data (`unigram`, `bigram`, `GIZA++`, `fast_align`, `eflomal`) and two models trained on corpus-level data (`leven-pmi`, `eflomal-priors`).

4.3 Symmetrization

Word alignment algorithms can only produce one-to-many alignments, but no many-to-one alignments. Therefore, it is standard practice to run the models twice, once in the "forward" direction and once in the "reverse" direction. The produced alignments are then symmetrized, e.g., by taking the intersection if precision is favored, or the union if recall is favored. Heuristics such as the popular *grow-diag-final-and* method produce a more balanced result (Och and Ney, 2003). For consistency, we apply symmetrization to all methods.

4.4 Adding Adjacent Identicals

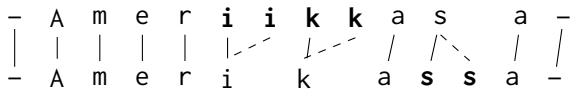

Figure 2: Additional alignments (dashed lines) are added to the initial alignments (solid lines) on the basis of consecutive identical characters (in bold).

Levenshtein-based models only produce one-to-one alignments, but leave inserted and deleted characters unaligned. To reduce the amount of

[3]We use the implementation by (Jiampojamarn et al., 2007) available at https://github.com/letter-to-phoneme/m2m-aligner.

unaligned characters, we add a simple heuristic that identifies two consecutive identical characters on one side and, if one of them is unaligned, introduces a new many-to-one alignment link (see Figure 2 for an example).

4.5 Evaluation Criteria

In a similar study, Wieling et al. (2009) compare various alignment methods with a set of manually verified gold alignments. Unfortunately, such annotations are not available for the three datasets used in this work. Instead, we gather four statistics about various phenomena that we consider undesirable for the given task, and rank the alignment methods according to these phenomena. They include:

U-src proportion of unaligned source characters,
U-tgt proportion of unaligned target characters,
V-C proportion of vowel-to-consonant and consonant-to-vowel alignments (disregarding semi-vowels, nasals, laterals and suprasegmentals),
X proportion of crossing alignment pairs (swaps).

We aggregate these proportions over all documents of a given dataset.

Note that we do not expect the optimal values of these proportions to be 0. The expected values depend on the languages and dialects, and reliable estimates would require access to a gold-aligned development set. However, based on our knowledge of the languages and dialects, we estimate **V-C** to lie below 1% and **X** below 0.2%. **U-tgt** is expected to be higher than **U-src**,[4] but both proportions are unlikely to exceed 15%.

Besides these quality indicators, we also report run times (on 1 CPU) and memory usage of the alignment methods.[5]

5 Results

5.1 Symmetrization Strategies

Table 3 exemplifies the effect of different symmetrization strategies on the basis of *eflomal* and the SKN dataset, but similar results are obtained for the other methods and datasets. It can be seen that recall-focused strategies (union) provide the lowest number of unaligned characters, whereas precision-focused strategies (intersection) show the lowest amounts of vowel-consonant alignments and crossing alignments. The *grow-diag-final-and* (`gdfa`) strategy is largely similar to union, but greatly reduces the number of crossing alignments. We find that `gdfa` provides the best compromise overall and select this symmetrization method for all subsequent experiments.

	forward	reverse	intersect	union	gdfa
U-src	9.39	9.51	13.77	**7.53**	7.63
U-tgt	6.00	7.18	11.11	**4.64**	4.76
V-C	0.17	0.15	**0.11**	0.21	0.20
X	0.50	0.49	**0.02**	1.00	0.12

Table 3: Impact of alignment symmetrization strategies. All values are percentages and refer to *eflomal* alignments on the SKN dataset.

5.2 Adding Adjacent Identicals

Table 4 shows that the adjacent-identicals heuristic effectively reduces the number of unaligned characters on both source and target sides, but leaves the other measures largely unaffected. In the following, we add this heuristic to all Levenshtein- and unigram-based methods and apply it after symmetrization with `gdfa`.

	SKN		NDC		ArchiMob	
	–aai	+aai	–aai	+aai	–aai	+aai
U-src	9.27	**8.85**	5.09	**1.25**	4.57	**2.65**
U-tgt	6.18	**5.22**	8.10	**7.92**	13.76	**12.78**
V-C	0.31	0.31	0.36	0.38	1.37	1.34
X	0.00	0.00	0.00	0.00	0.02	0.02

Table 4: Impact of adding adjacent identicals (`+aai`) on Levenshtein alignment. All values are percentages.

5.3 Method Comparison

The comparison between the eight alignment methods enumerated in Section 4.2 is shown in Table 5.

Two methods, `GIZA++` and `fast_align`, yield unrealistically high proportions of unaligned characters, leaving half of all characters unaligned in the worst case. The same methods also show higher-than-expected amounts of swaps. On the other hand, the `bigram` transducer produces unexpectedly large amounts of vowel-consonant alignments. These three methods can therefore not be recommended for character alignment with the used parameters.

[4] In SKN, **U-src** may be higher than **U-tgt** because of the suprasegmentals occurring in the source.

[5] The code for all experiments is available at https://github.com/Helsinki-NLP/dialect-align-sigmorphon23.

		Leven	Leven+PMI	unigram	bigram	GIZA++	fast_align	eflomal	eflomal+priors
SKN	U-src	8.85	8.11	9.83	9.60	*39.99*	*50.13*	7.63	7.67
	U-tgt	5.22	4.67	6.47	7.35	*38.56*	*48.66*	4.76	4.65
	V-C	0.31	0.46	0.07	*8.51*	0.20	0.24	0.20	0.25
	X	0.00	0.00	0.00	0.05	*0.75*	*0.26*	0.12	*0.40*
NDC	U-src	1.25	1.11	1.95	5.17	*15.34*	*26.64*	2.49	3.22
	U-tgt	7.92	7.54	8.85	8.13	*21.03*	*31.59*	7.51	7.45
	V-C	0.38	0.46	0.15	*6.36*	0.39	*1.26*	0.43	0.38
	X	0.00	0.00	0.00	0.07	*0.39*	*0.32*	0.02	0.13
ArchiMob	U-src	2.65	2.66	13.54	3.74	7.45	13.91	2.33	3.67
	U-tgt	12.78	12.85	*23.59*	10.51	*17.52*	*23.95*	9.14	12.61
	V-C	*1.34*	*1.39*	0.63	*7.81*	0.71	*1.48*	*2.00*	*1.22*
	X	0.02	0.00	0.00	0.07	*0.50*	*0.63*	0.12	0.14
CPU time (hh:mm)		0:30	11:17	20:20	105:27	30:36	0:45	12:32	16:18
Memory (MB)		69	76	1290	2350	58	34	263	268

Table 5: Evaluation of character alignment methods. All values are percentages, lower values are assumed to be better. Values violating our expectations are shown in italics.

The Levenshtein-based and `unigram` models do not permit swaps, leaving the corresponding measure at 0.[6] Since this is a technical limitation of the models, it should not be considered as an argument in their favor.

Learning the weights over the entire corpus (`leven-pmi`, `eflomal-priors`) does not consistently improve (nor worsen) results. We would have expected this approach to be useful especially for SKN and NDC with their short texts. This also contrasts with the findings of Wieling et al. (2009), who obtained significant error rate reductions with PMI-based Levenshtein distance. More thorough inspection of the results will be required to explain this divergence.

Three models (`leven`, `unigram`, `eflomal`) show similar performance over our criteria. They can be recommended in different situations. If crossing alignments (swaps) are expected to occur in the data, `eflomal` is the only recommended solution. If phonological consistency is highly rated, the `unigram` transducer is the method of choice because it produces the lowest rate of vowel-consonant alignments, at the expense of slightly higher amounts of unaligned tokens. Finally, plain Levenshtein distance remains remarkably competitive compared to the trained models. It is also one of the fastest and least memory-hungry approaches.

[6] It is nevertheless possible to obtain swaps through symmetrization. It has also been proposed to add a swap transition to Levenshtein distance, but preliminary experiments have shown that this addition negatively affects the other measures.

6 Discussion

Our evaluation of character alignment methods is based on a set of "undesirable characteristics" of the task. In this section, we would like to discuss some issues arising from this experimental setup.

In Swiss German and Finnish, a common pattern is the lack of final *n* in the dialectal pronunciation. For Swiss German *müesse / müssen*, two solutions are available: (a) a one-to-many alignment containing both *e–e* and *e–n*, and (b) leaving *n* unaligned. Although both options can be considered linguistically equivalent, our evaluation favors solution (a). In the opposite direction, the same argument holds for the suprasegmental symbols in the SKN corpus.

The transcription systems of Norwegian and Swiss German are based on conventional orthography and render some phonemes by multiple characters (e.g. Norwegian ***sj*e / *ikk*e**). It is unclear how alignment errors inside such multi-character graphemes should be evaluated.

Alignment can be performed left-to-right or right-to-left. For Norwegian *æin / en*, the former yields *æ–e* and the latter *i–e*. Although symmetrization minimizes the effects of alignment direction, its impact on the evaluation scores is not entirely clear.

Despite these yet unsolved questions, we believe that our evaluation provides interesting insights into the performance of character alignment methods for sentence-level dialect-to-standard normalization.

Limitations

A major limitation of the current work is the absence of gold alignments for evaluating the different methods. Gold alignments would also enable us to provide more reliable estimates of the prevalence of the evaluated phenomena in the three datasets. We are not aware of any other similar corpora that come with gold character alignments. The work of Wieling et al. (2009) uses word lists, not entire sentences.

Furthermore, our work currently only covers European languages in Latin script. Some of the presented techniques also assume identical writing systems in the transcribed and normalized layers. Our setup may therefore not generalize well to the dialectal variation and writing systems present in other parts of the world. For example, the **V-C** proportion cannot be easily determined in scripts that do not specify all vowels. Although there is an extensive amount of research in particular on Arabic and Japanese dialects and their normalization (e.g., Abe et al., 2018; Eryani et al., 2020), we currently limit our experiments to data written in Latin script.

Ethics Statement

All experiments are based on publicly available corpora. Even though some of the corpora contain personal information, they have been cleared for publication. The reported experiments do not introduce any new artifacts that would be problematic from an ethical point of view.

References

Kaori Abe, Yuichiroh Matsubayashi, Naoaki Okazaki, and Kentaro Inui. 2018. Multi-dialect neural machine translation and dialectometry. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong. Association for Computational Linguistics.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Fadhl Eryani, Nizar Habash, Houda Bouamor, and Salam Khalifa. 2020. A spelling correction corpus for multiple Arabic dialects. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4130–4138, Marseille, France. European Language Resources Association.

Wilbert Heeringa, Peter Kleiweg, Charlotte Gooskens, and John Nerbonne. 2006. Evaluation of string distance algorithms for dialectology. In *Proceedings of the Workshop on Linguistic Distances*, pages 51–62, Sydney, Australia. Association for Computational Linguistics.

Institute for the Languages of Finland. 2021. Samples of Spoken Finnish, VRT Version.

Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and hidden Markov models to letter-to-phoneme conversion. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 372–379, Rochester, New York. Association for Computational Linguistics.

Janne Bondi Johannessen, Joel James Priestley, Kristin Hagen, Tor Anders Åfarli, and Øystein Alexander Vangsnes. 2009. The Nordic Dialect Corpus – an advanced research tool. In *Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009)*, pages 73–80, Odense, Denmark. Northern European Association for Language Technology (NEALT).

Grzegorz Kondrak and Tarek Sherif. 2006. Evaluation of several phonetic similarity algorithms on the task of cognate identification. In *Proceedings of the Workshop on Linguistic Distances*, pages 43–50, Sydney, Australia. Association for Computational Linguistics.

Vladimir I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 8(10):707–710.

Gideon S. Mann and David Yarowsky. 2001. Multipath translation lexicon induction via bridge languages. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

John Nerbonne and William Kretzschmar. 2003. Introducing computational techniques in dialectometry. *Computers and the Humanities*, 37(3):245–255.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong Kong. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *Prague Bulletin of Mathematical Linguistics*, 106:125–146.

Eric Ristad and Peter Yianilos. 1998. Learning string edit distance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20:522 – 532.

Yves Scherrer and Tomaž Erjavec. 2013. Modernizing historical Slovene words with character-based SMT. In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, pages 58–62, Sofia, Bulgaria. Association for Computational Linguistics.

Yves Scherrer, Tanja Samardžić, and Elvira Glaser. 2019. Digitising Swiss German: how to process and study a polycentric spoken language. *Language Resources and Evaluation*, 53(4):735–769.

Jörg Tiedemann. 2009. Character-based PSMT for closely related languages. In *Proceedings of the 13th Annual conference of the European Association for Machine Translation*, Barcelona, Spain. European Association for Machine Translation.

Martijn Wieling and John Nerbonne. 2015. Advances in dialectometry. *Annual Review of Linguistics*, 1(1):243–264.

Martijn Wieling, Jelena Prokić, and John Nerbonne. 2009. Evaluating the pairwise string alignment of pronunciations. In *Proceedings of the EACL 2009 Workshop on Language Technology and Resources for Cultural Heritage, Social Sciences, Humanities, and Education (LaTeCH – SHELT&R 2009)*, pages 26–34, Athens, Greece. Association for Computational Linguistics.

Shijie Wu and Ryan Cotterell. 2019. Exact hard monotonic attention for character-level transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1530–1537, Florence, Italy. Association for Computational Linguistics.

A Appendix

The appendix provides further information about the used datasets (Table 6).

Dataset	Licence	URL
SKN	CC-BY	http://urn.fi/urn:nbn:fi:lb-2021112221
NDC	CC BY-NC-SA 4.0	http://www.tekstlab.uio.no/scandiasyn/download.html
(realigned)	CC BY-NC-SA 4.0	https://github.com/Helsinki-NLP/ndc-aligned
ArchiMob	CC BY-NC-SA 4.0	https://www.spur.uzh.ch/en/departments/research/textgroup/ArchiMob.html

Table 6: Datasets used in the experiments.