Do we need bigram alignment models? On the effect of alignment quality on transduction accuracy in G2P

Steffen Eger
Text Technology Lab
Goethe University Frankfurt am Main
steeger@em.uni-frankfurt.de

Abstract

We investigate the need for *bigram* alignment models and the benefit of *supervised* alignment techniques in grapheme-to-phoneme (G2P) conversion. Moreover, we quantitatively estimate the relationship between alignment quality and overall G2P system performance. We find that, in English, bigram alignment models do perform better than unigram alignment models on the G2P task. Moreover, we find that supervised alignment techniques may perform considerably better than their unsupervised brethren and that few manually aligned training pairs suffice for them to do so. Finally, we estimate a highly significant impact of alignment quality on overall G2P transcription performance and that this relationship is linear in nature.

1 Introduction

Grapheme-to-phoneme (G2P) conversion is the problem of converting a string of letters into a string of phonetic symbols. Closely related to G2P are other string transduction problems in natural language processing (NLP) such as transliteration (Sherif and Kondrak, 2007), lemmatization (Dreyer et al., 2008), and spelling error correction (Brill and Moore, 2000). The classical learning paradigm in each of these settings is to train a model on pairs of strings $\{(\mathbf{x},\mathbf{y})\}$ and then to evaluate model performance on test data. While there are exceptions (e.g., (Rao et al., 2015)), most state-of-the-art modelings (e.g., (Jiampojamarn et al., 2007; Bisani and Ney, 2008; Jiampojamarn et al., 2008; Jiampojamarn et al., 2010; Novak et al., 2012)) view string transduction as a two-stage process in which string pairs $(\mathbf{x},\mathbf{y})$ in the training data are first *aligned*, and then a subsequent (e.g., sequence labeling) module is learned on the aligned data.

ph	oe	n	i	x
f	i	n	ɪ	ks

Table 1: Sample monotone many-to-many alignment between $\mathbf{x} =$ phoenix and $\mathbf{y} =$ finɪks.

State-of-the-art alignments in G2P are characterized by the following properties:

(i) Alignments are *monotone* in that the ordering of characters in input and output sequences is preserved by the alignments. Furthermore, they are *many-to-many* in the sense that several $\mathbf{x}$ sequence characters may be matched up with several $\mathbf{y}$ sequence characters as illustrated in Table 1.

(ii) The alignment is a *latent variable* and learnt in an *unsupervised* manner from pairs of strings in the training data.

(iii) The unsupervised alignment models are *unigram alignment* models insofar as the overall score that the alignment model assigns an alignment is the same for all orderings of the matched-up subsequences (context independence).

To illustrate point (iii), consider, in the field of lemmatization, the case of aligning an inflected word form with the extended infinitive in German, such as *absagt* ('rejects') with *abzusagen* ('to reject'). Critically, the insertion *-zu-* appears in infixal position and a plausible alignment might be as in Table 2. Then, correctly aligning certain analogous forms such as *zusagt* ('accepts') with

a	b	ϵ	s	a	g	t
a	b	zu	s	a	g	en

Table 2: Alignment between *absagen* and *abzusagen*. Empty string denoted by ϵ.

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1175–1185,
Lisbon, Portugal, 17-21 September 2015. ©2015 Association for Computational Linguistics.

their corresponding extended infinitive *zuzusagen* ('to accept') is beyond the scope of a unigram alignment model since this cannot distinguish the linguistically correct alignment from the following linguistically incorrect alignment

ϵ	z	u	s	a	g	t
zu	z	u	s	a	g	en

precisely because it has no notion of context.

In this work, we **firstly** address bigram alignment models in G2P. We investigate whether there are phenomena in G2P that require bigram alignment models and, more generally, whether bigram alignment models produce better alignments — with respect to a human gold standard — than unigram alignment models within the G2P setting. We do so, **secondly**, in a *supervised* setting where the model learns from gold-standard alignments. While this may seem an odd scenario at first sight, modern alignment toolkits in the related field of machine translation typically include the possibility to learn both in a supervised and unsupervised manner (Liu et al., 2010; Liu and Sun, 2015). The rationale behind supervised learning models may be that they perform better than unsupervised models, and if alignment quality has a large impact upon subsequent string translation performance, then a supervised model may be a suitable alternative. **Thirdly**, we investigate how alignment quality affects overall G2P performance. This allows us to address whether it is worthwhile to work on better alignment models, which bigram and supervised alignment models promise to be. To our knowledge, all three outlined aspects of alignments — bigram models, supervised learning, and *systematically* estimating the relationship between alignment quality and overall string transduction performance — are novel in the G2P setting and its related fields as outlined; however, see also the related work section.

This work is structured as follows. Section 2 presents definitions and algorithms for uni- and bigram alignment models. Section 3 surveys related work. Section 4 presents our data and Section 5 our experiments. We conclude in Section 6.

2 Uni- and bigram alignment models

We first formally define the problem of aligning two strings $\mathbf{x}$ and $\mathbf{y}$ over arbitrary alphabets in a *monotone and many-to-many* manner. Let $\ell_x = |\mathbf{x}|$ and $\ell_y = |\mathbf{y}|$ denote the lengths of $\mathbf{x}$ and $\mathbf{y}$, respectively. Let $\mathbb{N} = \{0, 1, 2, \ldots\}$, and let $S \subseteq \mathbb{N}^2 \backslash \{(0, 0)\}$ be a set defining the valid match-up operations between $\mathbf{x}$ characters and $\mathbf{y}$ characters. In other words, when $(s, t) \in S$, then this means we allow matches of subsequences of $\mathbf{x}$ of length s and subsequences of $\mathbf{y}$ of length t.[1]

It is convenient to define a monotone many-to-many alignment of $\mathbf{x}$ and $\mathbf{y}$ as a $2 \times k$ (for $k \geq 1$ arbitrary) nonnegative integer matrix $\mathbf{A}_{\mathbf{x},\mathbf{y}} \in \mathbb{N}^{2 \times k}$ satisfying $\mathbf{A}_{\mathbf{x},\mathbf{y}} \mathbb{1}_k = \begin{pmatrix} \ell_x \\ \ell_y \end{pmatrix}$, i.e., the two rows of $\mathbf{A}_{\mathbf{x},\mathbf{y}}$ sum up to the lengths of the respective strings,[2] and where each column of $\mathbf{A}_{\mathbf{x},\mathbf{y}}$ lies in S. For any such alignment, we let $(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ be the corresponding induced segmentation of $\mathbf{x}$ and $(\mathbf{y}_1, \ldots, \mathbf{y}_k)$ be the corresponding induced segmentation of $\mathbf{y}$.

Example. For any $S \supseteq \{(1, 1), (1, 2), (2, 1)\}$, the alignment of $\mathbf{x}$ = phoenix and $\mathbf{y}$ = finɪks shown in Table 1 may be represented by the matrix $\mathbf{A}_{\mathbf{x},\mathbf{y}} = \begin{pmatrix} 2 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 \end{pmatrix}$. The corresponding induced segmentations are (ph,oe,n,i,x) and (f,i,n,ɪ,ks).

Let $\mathcal{A}_S(\mathbf{x}, \mathbf{y})$ denote the class of all alignments of $\mathbf{x}$ and $\mathbf{y}$. We call a function $f : \mathcal{A}_S(\mathbf{x}, \mathbf{y}) \to \mathbb{R}$ an *alignment model*. We call an alignment model f a *unigram alignment model* if f takes the form, for any $\mathbf{A}_{\mathbf{x},\mathbf{y}} \in \mathcal{A}_S(\mathbf{x}, \mathbf{y})$,

$$f(\mathbf{A}_{\mathbf{x},\mathbf{y}}) = \sum_{i=1}^{k} \mathsf{sim}_1(\mathbf{x}_i, \mathbf{y}_i) \tag{1}$$

where sim_1 is an arbitrary (real-valued) similarity function measuring similarity of two subsequences. We call an alignment model f a *bigram alignment model* if f takes the form

$$f(\mathbf{A}_{\mathbf{x},\mathbf{y}}) = \sum_{i=1}^{k} \mathsf{sim}_2\Big((\mathbf{x}_i, \mathbf{y}_i), (\mathbf{x}_{i-1}, \mathbf{y}_{i-1})\Big) \tag{2}$$

where sim_2 is an arbitrary (real-valued) similarity function measuring similarity of successive pairs of subsequences.

Example. Let $\mathsf{sim}_1(\mathbf{u}, \mathbf{v})$ be equal to $|\mathbf{u}| \cdot |\mathbf{v}|$ and let $f_{\text{uni}}(\mathbf{A}_{\mathbf{x},\mathbf{y}})$ be as in Eq. (1). Then, f_{uni} is a unigram alignment model that assigns the score

[1] This is sometimes denoted in the manner M-N (e.g., 3-2, 1-0), indicating that M characters of one string may be matched up with N characters of the other string. Analogously, we could write here s-t rather than (s, t).

[2] Here, $\mathbb{1}_k$ denotes the unit vector of dimension k.

$1 + 1 + 0 + 1 + 1 + 1 + 2 = 7$ to the alignment given in Table 2.

Example. Let $\mathsf{sim}_2\big((\mathbf{u},\mathbf{v}),(\mathbf{u}',\mathbf{v}')\big) = (|\mathbf{u}| \cdot |\mathbf{v}|)^{|\mathbf{v}'|}$ if $|\mathbf{u}| = |\mathbf{u}'| - 1$ or $\mathbf{u} = \mathbf{v}$ and -2 otherwise. Let $f_{\text{bi}}(\mathbf{A}_{\mathbf{x},\mathbf{y}})$ be as in Eq. (2). Then, f_{bi} is a bigram alignment model assigning the score $(1\cdot 1)^0 + (1\cdot 1)^1 + (0\cdot 2)^1 + (1\cdot 1)^2 + (1\cdot 1)^1 + (1\cdot 1)^1 - 2 = 3$ to the alignment in Table 2.

In statistical alignment modeling, the task is to find an *optimal* alignment (i.e., one with maximal score) given strings $\mathbf{x}$ and $\mathbf{y}$ and given the alignment model f. When f is a unigram model, this can be solved efficiently via dynamic programming (DP). When f is a bigram alignment model, then finding the optimal alignment can still be solved via DP, by introducing a variable M_{ijqw} denoting the score of the best alignment of $\mathbf{x}(1:i)$ and $\mathbf{y}(1:j)$ that ends in the match-up of $\mathbf{x}(q:i)$ with $\mathbf{y}(w:j)$.[3] The variable M_{ijqw} satisfies a recurrence leading to a DP algorithm, shown in Algorithm 1. The actual alignment can be found by storing pointers to the maximizing steps taken. Running time of the algorithm is $\mathcal{O}(\ell_{\mathbf{x}}^2\ell_{\mathbf{y}}^2|S|)$. Note also that the sketched algorithm is supervised insofar as it assumes that the similarity values $\mathsf{sim}_2(\cdot,\cdot)$ are known. Typically, such alignment algorithms can be converted into unsupervised algorithms in which similarity measures sim are learnt iteratively, e.g., in an EM-like fashion (cf., e.g., Eger (2012), Eger (2013)); however, in this paper, we only investigate the supervised base version as indicated.

3 Related work

Monotone alignments have a long tradition in NLP. The classical Needleman-Wunsch algorithm (Needleman and Wunsch, 1970) computes the optimal alignment between two sequences when only single character matches, mismatches, and skips are allowed. It is a special case of the unigram model (1) for which $S = \{(1,0),(0,1),(1,1)\}$ and sim_1 takes on values from $\{0,-1\}$, depending on whether compared subsequences match or not. As is well-known, this alignment specification is equivalent to the edit distance problem (Levenshtein, 1966) in which the minimal number of insertions, deletions and substitutions is sought that transforms one string into another. *Substring-to-substring edit operations* — or equivalently, (monotone) many-to-many alignments — have appeared in the NLP context, e.g., in Deligne et al. (1995), Brill and Moore (2000), Jiampojamarn et al. (2007), Bisani and Ney (2008), Jiampojamarn et al. (2010), or, significantly earlier, in Ukkonen (1985), Véronis (1988). *Learning* edit distance/monotone alignments in an *unsupervised* manner has been the topic of, e.g., Ristad and Yianilos (1998), Cotterell et al. (2014), besides the works already mentioned. All of these approaches are special cases of our unigram model — i.e., they consider particular S (most prominently, $S = \{(1,0),(0,1),(1,1)\}$) and sim_1.[4] Eger (2015b), Yao and Kondrak (2015), and Eger (2015a) generalize to alignments of multiple strings, but likewise only consider unigram alignment models in their experiments.

Probably the most closely related work to ours is Jiampojamarn and Kondrak (2010). There, older and specialized alignment techniques such as ALINE (Kondrak, 2000) (as well as partly heuristic/semi-automatic alignment methods) are compared with variants of the M2M alignment algorithm, which we also survey. This work does not consider supervised alignments or bigram alignments, as we do. Moreover, Jiampojamarn and Kondrak (2010) also evaluate the impact of alignment quality on overall G2P system accuracy by running a few experiments, finding that better alignment quality does not always translate into better G2P accuracy, but that there is a "strong correlation" between the two. We more thorougly investigate this question, using, arguably, more heterogeneous aligners, and many more experiments. We also quantitatively estimate how alignment quality influences G2P system accuracy on two different languages via linear regression.

Goldwater et al. (2006) study the effect of context in (unsupervised) word/sequence segmentation, which may be considered the one-dimensional specialization of sequence alignment, using a Bayesian method. They find that bigram models greatly outperform unigram models for their task.

Of course, our study is also related to the field of machine translation and its studies on the rela-

[3] We denote by $\mathbf{x}(a:b)$ the substring $x_a x_{a+1} \cdots x_b$ of the string $x_1 x_2 \cdots x_t$.

[4] In Cotterell et al. (2014), context influences alignments, so that the approach goes beyond the unigram model sketched in (1) (but does not allow for many-to-many match-ups). The contextual dependencies in this model are set up differently from the bigram dependencies in our paper.

Algorithm 1

```
procedure BIGRAM-ALIGN($\mathbf{x} = x_1 \ldots x_n, \mathbf{y} = y_1 \ldots y_m; S, \mathsf{sim}_2$)
    $M_{ijqw} \leftarrow -\infty$ for all $(i, j, q, w) \in \mathbb{Z}^4$
    $M_{0000} \leftarrow 0$
    for $i = 0 \ldots n$ do
        for $j = 0 \ldots m$ do
            for $q = 0 \ldots i + 1$ do
                for $w = 0 \ldots j + 1$ do
                    if $(i, j, q, w) \neq (0, 0, 0, 0)$ then
                        if $(i - q + 1, j - w + 1) \in S$ then
                            $M_{ijqw} = \max_{(a,b) \in S} M_{q-1,w-1,q-a,w-b} + \mathsf{sim}_2\Big(\big(\mathbf{x}(q{:}i), \mathbf{y}(w{:}j)\big), \big(\mathbf{x}(q-a{:}q-1), \mathbf{y}(w-b{:}w-1)\big)\Big)$
```

tionship between alignment quality and translation performance (Ganchev et al., 2008). In machine translation, the monotonicity assumption of string transduction does typically not hold, however, rendering alignment and translation techniques different and more heuristic in nature.

4 Data and systems

4.1 Data

For English, we conduct experiments on the General American (GA) variant of the Combilex data set (Richmond et al., 2009). This contains about 128 000 grapheme-phoneme pairs as exemplified in Table 3. Importantly, Combilex provides gold-standard alignments, which we will make use of for the supervised alignment models as well as for measuring alignment quality. For German, we randomly extract 3 000 G2P string pairs from CELEX (Baayen et al., 1995). We had a native speaker manually align them so that gold standard alignments are available here, too. Both data sets contain quite complex match-ups of character subsequences such as (2,3) as in English *s-oi-r-ee-s*/s-wOA-r-P-z or (4,1) as in *w-eigh-t*/w-P-t but the majority of match-ups are of type (1,1), (2,1), and, to a lesser degree, (1,2) and (3,1).

Grapheme string	Phoneme string
g-e-n-e-r-a-l	dZ-E-n-@-r-@-l
p-r-o-b-a-t-ion-a-r-y	p-r-@U-b-eI-S-n=-E-r-i
w-oo-d-e-d	w-U-d-@-d
M-u-r-m-a-n-s-k	m-U@-r-m-A-n-s-k

Table 3: Sample grapheme-phoneme string pairs in Combilex, using Combilex notation for the phoneme strings. Gold-standard alignments indicated in an intuitive manner.

4.2 Alignment toolkits/models

The **M2M aligner** (Jiampojamarn et al., 2007), which is based on EM maximum likelihood estimation of alignment parameters, is the classical unsupervised unigram many-to-many aligner in G2P. As has been pointed out (Kubo et al., 2011), M2M greatly overfits the data.[5] This means that when the M2M aligner is given the freedom to align two sequences without restrictions, it matches them up as a whole. The reason is that a (probabilistic) unigram alignment model adds log-probabilities of matched-up subsequences, which, if not appropriately corrected for, makes alignments with few match-ups a priori more likely than alignments with many match-ups, when probabilities of individual match-ups are uniformly or randomly initialized (as is typically the case for EM maximum likelihood estimation in unsupervised models). To address this, M2M must artifically restrain, in our language, the set S to be $\{(1,1),(1,2),(2,1)\}$. In contrast, the **Mpaligner** (Kubo et al., 2011) introduces a prior (or penalty) in the alignment model which favors 'short' matches (s,t) over 'long' ones. Finally, the **Phonetisaurus** aligner (Novak et al., 2012) modifies the M2M aligner by adding additional soft constraints.

Our own alignment model is, as indicated, supervised. We implement a **unigram** alignment model where we specify $\mathsf{sim}_1(\mathbf{u}, \mathbf{v})$ as

$$\alpha \cdot \text{logp}((\mathbf{u}, \mathbf{v})) + \beta \cdot \text{logp}((|\mathbf{u}|, |\mathbf{v}|)) \\ + \gamma \cdot \text{logp}(\mathbf{u}) + \delta \cdot \text{logp}(\mathbf{v}).$$

Here, $\text{logp}(\mathbf{z})$ denotes the log-probability — estimated from the training data — of observing the

[5] See also the discussion in (Goldwater et al., 2006) for the related word segmentation problem.

object $\mathbf{z}$, and α, β, γ and δ are parameters. This specification says that the subsequences $\mathbf{u}$ and $\mathbf{v}$ are similar insofar as (i) $\mathbf{u}$ and $\mathbf{v}$ have been paired frequently in the training data, (ii) the length of $\mathbf{u}$ and the length of $\mathbf{v}$ have been paired frequently, (iii)/(iv) $\mathbf{u}/\mathbf{v}$ by itself is likely. We refer to this unigram alignment model as $\mathsf{uni}_{\alpha,\beta,\gamma,\delta}$. We also implement a **bigram** alignment model where we specify $\mathsf{sim}_2\big((\mathbf{u},\mathbf{v}),(\mathbf{u}',\mathbf{v}')\big)$ as

$$\begin{aligned} &\alpha\cdot\mathrm{logp}\big((\mathbf{u},\mathbf{v})\,|\,(\mathbf{u}',\mathbf{v}')\big)\\ &+\beta\cdot\mathrm{logp}\big((|\mathbf{u}|,|\mathbf{v}|)\,|\,(|\mathbf{u}'|,|\mathbf{v}'|)\big)\\ &+\gamma\cdot\mathrm{logp}\big(\mathbf{u}|\mathbf{u}'\big)+\delta\cdot\mathrm{logp}\big(\mathbf{v}|\mathbf{v}'\big). \end{aligned}$$

Here, $\mathrm{logp}(\mathbf{z}\,|\,\mathbf{z}')$ denotes the logarithm of the conditional probability of observing the object $\mathbf{z}$ following the object $\mathbf{z}'$. We refer to this bigram alignment model as $\mathsf{bi}_{\alpha,\beta,\gamma,\delta}$.

4.3 Transduction systems

We use two string transduction systems for our experiments. The first one is **DirecTL+** (Jiampojamarn et al., 2010), a discriminative string-to-string translation system incorporating joint n-gram features. DirecTL+ is an extension of the model presented in Jiampojamarn et al. (2008) which treats string transduction as a source sequence segmentation and subsequent sequence labeling task. In addition, we use **Phonetisaurus** (Novak et al., 2012), a weighted finite state-based joint n-gram model employing recurrent neural network language model N-best rescoring in decoding. Both systems take aligned pairs of strings as input and from this construct a monotone translation model.[6]

4.4 Measuring alignment quality

We employ two measures of alignment quality. First, we use **word accuracy**, defined as the fraction of correctly aligned sequence pairs in a test sample. This is a very strict measure that penalizes even tiny deviations from the gold standard. Additionally, we measure the **edit distance** between the true alignment $\mathbf{A}_{\mathbf{x},\mathbf{y}}$ and the predicted alignment $\hat{\mathbf{A}}_{\mathbf{x},\mathbf{y}}$. To implement this, we view the two induced segmentations that constitute an alignment — e.g., (ph,oe,n,i,x) and (f,i,n,ɪ,ks) — as strings including splitting signs. Thus, we can compute the edit distance between the gold-standard segmented $\mathbf{x}$ string and the predicted segmentation, and analogously for the $\mathbf{y}$ sequence. Then, we define the edit distance between $\mathbf{A}_{\mathbf{x},\mathbf{y}}$ and $\hat{\mathbf{A}}_{\mathbf{x},\mathbf{y}}$ as the *sum* of these two string edit distances. For a test sample, we indicate so-defined *average* edit distance, averaged over all pairs in the sample.

5 Experiments

5.1 Alignment quality

To measure alignment quality for the different systems, for English, we run experiments on sets of size x+5 000, where $x = 1\,000$, 2 000, 5 000, 10 000, and 20 000. For the supervised models, we consider x as the training data and the 5 000 additional string pairs as test data.[7] To quantify effects when training data is very little, we let x also range over 100 and 500 string pairs for the supervised models. For the unsupervised models, we simply take all x+5 000 string pairs as data to learn from (but evaluate performance only on the 5 000 string pairs, for comparability).

Results are shown in Tables 4, 5, and 6. We first note (Table 4) that the *unsupervised* models perform decently, obtaining accuracy rates of 80% and beyond under appropriate parametrizations. We also observe the M2M aligner's deterioration in performance as we increase its degrees of freedom (allowing it to match subsequences of larger length), confirming our previous remarks. The Mpaligner does not suffer from this problem as it penalizes large matches. Phonetisaurus suffers from the same problems as M2M, but to a lesser degree. Overall, we find that, under optimal parametrizations, Phonetisaurus produces best alignments, followed by Mpalign and M2M. However, peak performances of all three unsupervised aligners are close. Unsurprisingly, the *supervised* alignment models perform better than the unsupervised ones (Tables 5 and 6). Surprisingly, however, they do so with very little training data; fewer than 100 aligned string pairs suffice to outperform the unsupervised models under good calibrations. When there is sufficient training data, the supervised models perform splendidly, with a peak accuracy of 99.43% for the bigram alignment model that includes appropriate features (scoring lengths of aligned subsequences,

[6] We run both systems with parameters determined by some manual tuning, without trying to systematically optimize their individual performances, however.

[7] For all our below experiments involving the supervised aligners, we set S to a ('pessimistically' large) value of $\{(a,b)\,|\,1\le a\le 6, 1\le b\le 6\}$. Also, for the bigram models, we add special sequence boundary markers.

etc.). We also note that the bigram alignment model is almost consistently better than the unigram alignment model, with a surplus of about 1% point, depending on specific parametrizations.

We performed an analogous analysis for the German data. Results are quite similar except that unigram and bigram alignment model have indistinguishable performance on the German data, indicating (the known fact) that G2P is a more complex task in English, apparently not requiring bigram alignment models.

x	$\text{uni}_{0,0,1,1}$	$\text{uni}_{1,0,0,0}$	$\text{uni}_{1,1,1,1}$
100	70.34	58.13	**87.22**
500	81.94	84.64	**95.60**
1000	84.56	90.38	**96.17**
2000	85.41	93.47	**97.13**
5000	86.56	96.11	**97.72**
10000	86.13	97.07	**98.14**
20000	86.60	97.90	**98.34**

Table 5: Unigram model and its alignment accuracies in % for various training sizes.

x	$\text{bi}_{0,0,1,1}$	$\text{bi}_{1,0,0,0}$	$\text{bi}_{1,1,1,1}$
100	73.96	58.02	**87.28**
500	87.62	85.31	**95.26**
1000	91.87	90.73	**97.32**
2000	93.29	94.11	**97.96**
5000	95.58	97.01	**99.03**
10000	96.07	98.12	**99.17**
20000	97.21	98.73	**99.43**

Table 6: Bigram model and its alignment accuracies in % for various training sizes.

Error analysis Concerning errors that the unigram model commits and the bigram model does not, the majority of errors (roughly 80%) involve match-ups of *ed/d* and d. For example, the unigram model aligns as in

t	w	i	n	k	le	d
t	w	I	N	k	@l	d

while the gold-standard alignment is

t	w	i	n	k	l	ed
t	w	I	N	k	@l	d

While all match-ups in both alignments are plausible, the bigram model assigns here higher probability to the correct *ed/d* match-up in terminal position (consistently favored in the data set), which has a particular meaning there, namely, that of a suffix marker for past tense.[8,9] In the German data, there is a single instance where the unigram and bigram alignment model disagree, namely, in the alignment of *s-t-o-ff-f-l-a-sch-e*/S-t-O-f-f-l-&-S-@, which the unigram model falsely aligns as *s-t-o-f-ff-l-a-sch-e*/S-t-O-f-f-l-&-S-@; note that in the correct alignment *f* must follow *ff*, not vice versa, which depends on context information, e.g., that o/O signifies a short vowel which is followed by a double consonant, not a single consonant.

All remaining errors that the bigram alignment models commits are, for the best considered parametrization and training set size, typically due to match-up types not seen in the training data, and thus mostly concern foreign names or writings (e.g., *Bh-u-tt-o*/b-u-t-F, falsely aligned as *B-hu-tt-o*/b-u-t-F). A few other errors might be corrected when the feature coefficients $\alpha, \beta, \gamma, \delta$ were optimized on a development set rather than set manually. We find no indication that our G2P data, either for English or German, would further benefit from n-gram alignment models of order $n > 2$.

5.2 Alignment quality vs. overall G2P performance

Next, we estimate the relationship between alignment quality and overall G2P performance (transcription accuracy). To this end, for the English data, we use the 5 000 aligned string pairs from the previous experiment on alignment quality and feed them in — as training data — to either DirecTL+ or Phonetisaurus as outlined in Section 4. We then evaluate G2P performance — in terms of **word accuracy** (fraction of correctly transcribed strings) — on a distinct test set of size 10 000. Figure 1 shows a plot of overall G2P accuracy vs. *training set size for the aligner* (ranging over the x values in the last section); and a second plot that sketches G2P accuracy as a function of corresponding alignment accuracy. We first note that, as the supervised aligner receives more training

[8] Similar cases are, e.g., alignments of the type *f-ee-d-b-a-ck*/f-i-d-b-a-k, which the unigram model falsely aligns as *f-e-ed-b-a-ck*/f-i-d-b-a-k. Here, too, the unigram is unable to account for the almost exclusive terminal position of the *ed/d* match-up in the data.

[9] Other errors involve 'unusual/foreign' spelling/pronunciation pairs such as *Ph-oe-n-i-c-ia*/f-@-n-i-S-@ (wrongly aligned as *Ph-o-en-i-c-ia*/f-@-n-i-S-@ by the unigram model) or *m-a-d-e-m-oi-s-e-ll-e-'s*/m-a-d-@-m-w@-z-E-l-0-z (*m-a-d-e-m-o-i-s-e-ll-e-'s*/m-a-d-@-m-w-@-z-E-l-0-z), where the bigram alignment model has apparently gathered the more appropriate statistics.

x	Mpalign	M2M$_{2,2}$	M2M$_{3,3}$	M2M$_{6,6}$	Phon$_{2,2}$	Phon$_{3,3}$	Phon$_{6,6}$
1000	76.48	77.87	34.59	18.96	**78.27**	78.15	11.70
2000	78.05	78.03	34.45	18.87	**79.24**	77.07	12.43
5000	76.68	77.93	35.09	19.72	79.77	**80.47**	17.63
10000	78.86	77.97	35.03	21.35	79.60	**81.30**	23.57
20000	79.87	78.60	37.09	22.90	80.09	**83.37**	34.61

Table 4: Unsupervised aligners and their alignment accuracies in % for various data sizes as described in the text. Subscripts a, b denote restrictions on maximal lengths of subsequences allowed in match-ups (a/b corresponds to $\mathbf{x}/\mathbf{y}$ subsequences).

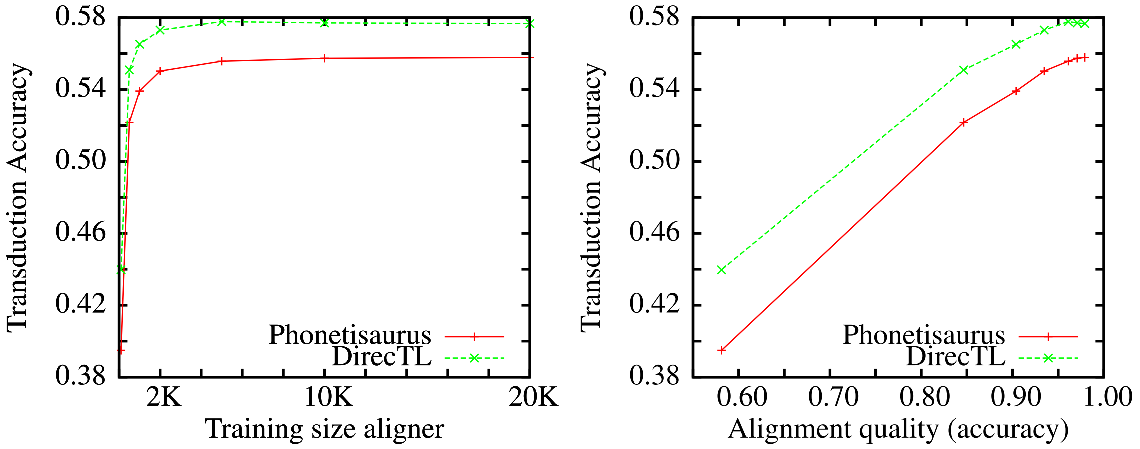

Figure 1: Left: Overall G2P accuracy as a function of training set size of supervised aligner $\texttt{uni}_{1,0,0,0}$. Right: G2P accuracy as a function of alignment quality (measured in accuracy).

data from which to align the 5 000 string pairs, the overall G2P accuracy of both DirecTL+ and Phonetisaurus increase substantially (and as a convex function of training set size). Apparently, the better alignments produced by more training data for the particular supervised aligner considered directly translate into better overall G2P accuracy. The other plot in the figure shows that, indeed, there seems to be a *linear* trend coupling alignment quality with overall G2P performance. Table 7 pairs G2P accuracy with alignment accuracy of selected systems, all run in the $x = 20\,000$ setting. While, in the table, better alignments do not necessarily imply better overall G2P performance, the two best alignments also lead to the two best overall G2P performances (although, in this case, the second best alignment is paired with the best overall G2P performance); conversely, the worst alignment quality is coupled with the worst overall G2P performance.

	Alignment acc.	Phon.	DirecTL+
Mpalign	79.87	55.48	57.54
M2M$_{3,3}$	37.09	49.25	53.71
Phon$_{3,3}$	83.37	54.05	56.11
$\texttt{uni}_{0,0,1,1}$	86.60	53.19	55.49
$\texttt{uni}_{1,1,1,1}$	98.34	**55.72**	**57.78**
$\texttt{bi}_{1,1,1,1}$	**99.43**	55.71	57.71

Table 7: Systems, alignment accuracies of corresponding produced alignments and transcription accuracy of Phonetisaurus and DirecTL+ when trained with the respective alignments.

Overall, we ran 249 experiments (including the German data) in which we trained DirecTL+ or Phonetisaurus with alignments of specific qualities obtained from particularly parametrized aligners. In each of these cases, we obtained an alignment quality score and a subsequent overall G2P system performance. The English part of this data is sketched in Figure 2. This figure seems to corroborate the linear relationship (apparently present in Figure 1) between alignment quality and overall G2P system accuracy, particularly, when alignment quality is measured in the more fine-grained metric of edit distance. To formally test

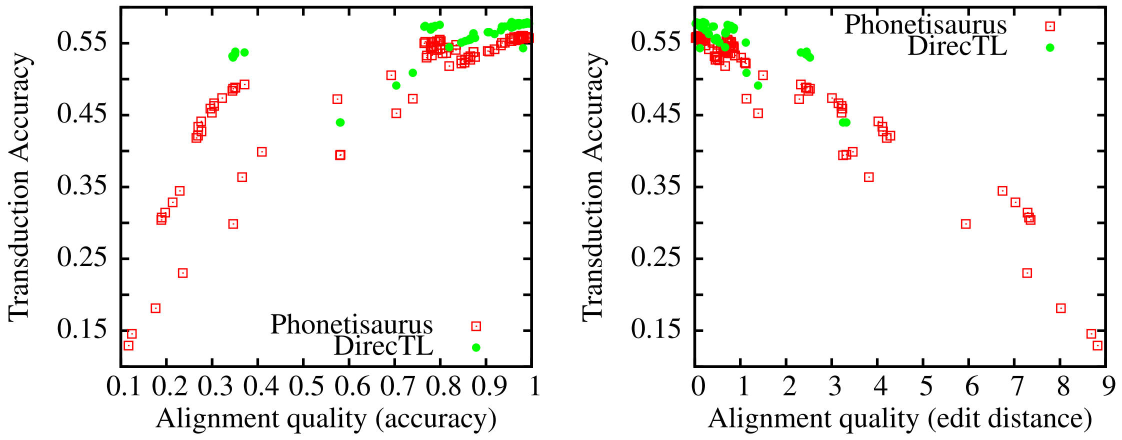

Figure 2: Overall G2P accuracy vs. alignment quality. Left: Alignment quality measured in accuracy. Right: Alignment quality measured in edit distance. English data only.

this, we regress overall G2P system performance (measured in word accuracy) on edit distance and other variables.[10] This yielded the coefficients as given in Table 8; in each case, the goodness-of-fit of the linear model was quite large, with R^2 values above 90% for the English data and about 84% for the German data. Also, the coefficients on alignment quality were highly significantly different from zero. The table shows that the coefficients are on the order of about -3.80% to -4.70%, meaning that, all else being equal, increasing alignment quality by 1 edit distance to the gold-standard alignment increases overall G2P by about 3.80 to 4.70%.

	DirecTL+	Phonetisaurus
English	-3.80^{***}	-4.14^{***}
German	-	-4.68^{***}

Table 8: Coefficients on edit distance in the regression of G2P accuracy on edit distance and further variables. For German, DirecTL+ is omitted due to its long run times.

So far, we have estimated the effects of alignment quality on overall G2P system performance for a **fixed** size of training data, namely, 5 000 aligned string pairs. To see whether this relationship changes when we vary the amount of training data, we run several more experiments. In these, we align training sets of sizes 100, 500, 1 000, 2 000, 10 000, 20 000, 40 000 and 60 000 via our several alignment systems. Then we feed the aligned data to the Phonetisaurus system (we omit DirecTL+ here because of its long run times) and compute overall G2P accuracy on a disjoint test set of size 28 000 approximately. This time, we only use the unsupervised aligners and the gold-standard alignments directly, omitting results for our various supervised aligners. Note, however, that these aligners could, in principle, imitate the gold-standard alignments with a very high degree of precision, as previously seen. Table 9 shows that training G2P systems from the human gold standard alignments in each case yields better overall G2P transcriptions than training them from either of the three unsupervised alignments considered here. However, we note that the surplus over the unsupervised alignments decreases as training set size increases. This may be due to the fact that the unsupervised aligners themselves create better alignments once they are boot-

	$\text{M2M}_{3,3}$	Mpalign	$\text{Phon}_{3,3}$	Gold
100	5.38	6.43	0.19	**9.60**
500	16.80	22.43	5.08	**23.93**
1K	25.79	31.46	18.70	**33.37**
2K	35.31	42.01	37.74	**43.64**
10K	58.44	64.05	63.06	**64.60**
20K	67.70	71.70	71.51	**72.21**
40K	74.69	78.45	78.13	**78.65**
60K	78.00	81.07	80.92	**81.17**

Table 9: Overall G2P accuracy in % as a function training size of aligned data and alignment system.

[10] These include binary dummy variables for the specific systems as well as alignment consistency and its square — measured in conditional entropy $H(Y|X)$ (Pervouchine et al., 2009) — in the regression.

strapped from larger data sets (cf. Table 4). Additionally, the effect of alignment quality on overall G2P system performance may simply vanish as training set sizes become large enough because the translation modules can better accomodate 'noisy' data as long as its size is sufficiently large. Figure 3 sketches the decreasing influence of alignment system on overall G2P system performance as size of the aligned data increases.

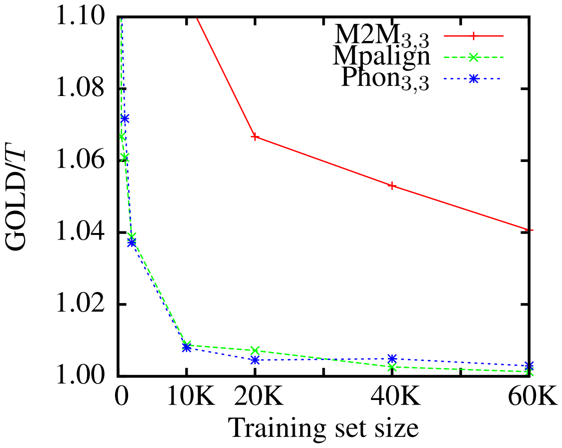

Figure 3: Ratio of transcription accuracy when using gold standard alignments (GOLD) and when using alignments generated by T = M2M$_{3,3}$, Mpalign, and Phon$_{3,3}$, respectively, as a function of size of aligned training set.

6 Conclusion

We have investigated the need for *bigram* alignment models and the benefit of *supervised* alignment techniques in G2P. We have also quantitatively estimated the relationship between alignment quality and overall G2P system performance. We have found that, in English, bigram alignment models do perform better than unigram alignment models on the G2P task (we find almost no differences between unigram and bigram models for the German sample of G2P data we considered). Moreover, we have found that supervised alignment techniques may perform considerably better than their unsupervised brethren and that few manually aligned training pairs suffice for them to do so. Finally, we have estimated a highly significant impact of alignment quality on overall G2P transcription performance and that this relationship is linear in nature. At a particular training size, a linear regression model has estimated that improving alignment quality by 1 edit distance toward the gold standard alignments leads to an 3.80-4.70% increase in G2P transcription accuracy. However, we have also found that the importance of good alignments on G2P accuracy appears to dimish as data set size increases, possibly because the translation modules can accomodate more 'noisy' data in this scenario.

As a 'policy' implication, we recommend the use of supervised alignment techniques particularly when the size of the G2P corpus is small or when high quality alignments, as an end in themselves, are required. In this case, constructing a few dozen or few hundred alignments in an unsupervised manner and correcting them by hand (to serve as an input for a supervised technique) may be highly beneficial.

In future work, it may be worthwhile to study the impact of alignment techniques on overall system performance in other string transduction problems such as transliteration, lemmatization, and spelling error correction.

Our supervised uni- and bigram aligners are available via `https://github.com/SteffenEger/`.

Acknowledgments

I thank three anonymous reviewers and Tim vor der Brück for valuable suggestions.

References

H. Baayen, R. Piepenbrock, and L. Gulikers. 1995. The CELEX2 lexical database. ldc96l14.

Maximilian Bisani and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451.

Eric Brill and Robert C. Moore. 2000. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 286–293, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ryan Cotterell, Nanyun Peng, and Jason Eisner. 2014. Stochastic contextual edit distance and probabilistic FSTs. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 625–630, Baltimore, June.

Sabine Deligne, Franois Yvon, and Frédéric Bimbot. 1995. Variable-length sequence matching for phonetic transcription using joint multigrams. In *EUROSPEECH*. ISCA.

Markus Dreyer, Jason Smith, and Jason Eisner. 2008. Latent-variable modeling of string transductions with finite-state methods. In *EMNLP*, pages 1080–1089. ACL.

Steffen Eger. 2012. S-restricted monotone alignments: Algorithm, search space, and applications. In *COLING'12*, pages 781–798.

Steffen Eger. 2013. Sequence alignment with arbitrary steps and further generalizations, with applications to alignments in linguistics. *Inf. Sci.*, 237:287–304.

Steffen Eger. 2015a. Improving g2p from wiktionary and other (web) resources. In *Proceedings of Interspeech*. accepted.

Steffen Eger. 2015b. Multiple many-to-many sequence alignment for combining string-valued variables: A G2P experiment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 909–919.

Kuzman Ganchev, Joo Graa, and Ben Taskar. 2008. Better alignments = better translations? In Kathleen McKeown, Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui, editors, *ACL*, pages 986–993. The Association for Computational Linguistics.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2006. Contextual dependencies in unsupervised word segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 673–680, Sydney, Australia, July. Association for Computational Linguistics.

Sittichai Jiampojamarn and Grzegorz Kondrak. 2010. Letter-phoneme alignment: An exploration. In Jan Hajic, Sandra Carberry, and Stephen Clark, editors, *ACL*, pages 780–788. The Association for Computational Linguistics.

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, April. Association for Computational Linguistics.

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2008. Joint processing and discriminative training for letter-to-phoneme conversion. In *Proceedings of ACL-08: HLT*, pages 905–913, Columbus, Ohio, June. Association for Computational Linguistics.

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2010. Integrating joint n-gram features into a discriminative training framework. In *HLT-NAACL*, pages 697–700. The Association for Computational Linguistics.

Grzegorz Kondrak. 2000. A new algorithm for the alignment of phonetic sequences. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, NAACL 2000, pages 288–295, Stroudsburg, PA, USA. Association for Computational Linguistics.

Keigo Kubo, Hiromichi Kawanami, Hiroshi Saruwatari, and Kiyohiro Shikano. 2011. Unconstrained many-to-many alignment for automatic pronunciation annotation. In *Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2011 (APSIPA2011)*, October.

VI Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.

Yang Liu and Maosong Sun. 2015. Contrastive unsupervised word alignment with non-local features. In *Proceedings of AAAI 2015*.

Yang Liu, Qun Liu, and Shouxun Lin. 2010. Discriminative word alignment by linear modeling. *Computational Linguistics*, pages 303–339.

Saul B. Needleman and Christian D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, March.

Josef R. Novak, Nobuaki Minematsu, and Keikichi Hirose. 2012. WFST-based grapheme-to-phoneme conversion: Open source tools for alignment, model-building and decoding. In *Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*, pages 45–49, Donostia–San Sebastin, July. Association for Computational Linguistics.

Vladimir Pervouchine, Haizhou Li, and Bo Lin. 2009. Transliteration alignment. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 136–144, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kanishka Rao, Fuchun Peng, Hasim Sak, and Franoise Beaufays. 2015. Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. In *ICASSP 2015*.

Korin Richmond, Robert A. J. Clark, and Susan Fitt. 2009. Robust LTS rules with the Combilex speech technology lexicon. In *INTERSPEECH*, pages 1295–1298. ISCA.

Eric Sven Ristad and Peter N. Yianilos. 1998. Learning string-edit distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(5):522–532.

Tarek Sherif and Grzegorz Kondrak. 2007. Substring-based transliteration. In John A. Carroll, Antal van den Bosch, and Annie Zaenen, editors, *ACL*. The Association for Computational Linguistics.

Esko Ukkonen. 1985. Algorithms for approximate string matching. *Information and Control*, 64:100–118.

Jean Véronis. 1988. Computerized correction of phonographic errors. *Computers and the Humanities*, 22(1):43–56.

Lei Yao and Grzegorz Kondrak. 2015. Joint generation of transliterations from multiple representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 943–952, Denver, Colorado, May–June. Association for Computational Linguistics.