Incorporating Target Language Models into Discriminative String Transduction

Anonymous ACL submission

Abstract

Many character-level tasks can be addressed as sequence-to-sequence transduction, where the target is a word from a natural language. We propose enhancements that allow discriminative transduction models to leverage large unannotated corpora for the purpose of guiding the candidate generation process. We demonstrate substantial improvements on several tasks and languages, including new state-of-the-art results in cognate projection and phoneme-to-grapheme conversion.

1 Introduction

Many natural language tasks, particularly those involving character-level operations, can be viewed as sequence-to-sequence transduction (Figure 1). Under the discriminative transduction paradigm, after the sequences are aligned, a learning algorithm assigns weights to features defined on source and target pairs. At test time, an input sequence is converted into the highest-scoring output sequence.

One of the most successful discriminative transduction tools is DirecTL+ (Jiampojamarn et al., 2010). It has been shown to achieve state-of-the-art results on several NLP tasks, including grapheme-to-phoneme conversion, transliteration, and inflection generation. Although neural sequence-to-sequence models (Bahdanau et al., 2014; Kann and Schütze, 2016) have since led to substantial improvements when sufficient amounts of training data are available, DirecTL+ is still competitive in low-data scenarios (Cotterell et al., 2017).

In spite of its effectiveness, DirecTL+ suffers from two deficiencies. First, in order to reduce the complexity of the transduction process, its set of edit operations excludes insertion. In practice, the insertions are implicitly incorporated into one-to-many substitutions by an unsupervised EM-based aligner (for example, consider the letter ‘l’ in Figure 2). However, in low-data scenarios, especially when the target strings are longer than the source strings, there is often insufficient evidence to derive correct alignments. This has a negative effect on the quality of subsequent transduction.

The second, even more critical, deficiency of DirecTL+ is that its target language modeling is limited to a set of binary \( n \)-gram features, which are based exclusively on the forms included in the parallel training data. In the low-data condition, the resulting model is too weak to rule out many ill-formed outputs. This shortcoming can be remedied by taking advantage of large unannotated corpora that contain thousands of examples of valid target words. In particular, Nicolai et al. (2015a) apply a discriminative reranker to the top-\( k \) candidates generated by the transducer, employing features such as a character-level language model
score, and unigram corpus presence/absence indicators. However, such a pipeline approach suffers from error propagation, and cannot produce output forms that are not already present in the top-\textit{k} list. In addition, training a reranker requires a held-out set that substantially reduces the amount of training data in low-data scenarios.

In this paper, we address these two deficiencies, and make the following contributions:

- We explicitly handle insertion operations during the alignment of the training data.
- We incorporate a stronger target language model derived from a large unannotated corpus. Specifically, we augment DirecTL+ with two new feature sets:
  - A character-level \textit{n}-gram model
  - A word-level unigram model
- We release to the community two coughly projection datasets, and DirecTLM, our augmented version of DirecTL+.

We report the results of experiments on several transduction tasks and language datasets, which demonstrate that the enhanced version of DirecTL+ dramatically improves transduction accuracy, achieving state-of-the-art results in phoneme-to-grapheme conversion and cognate projection.

2 Baseline approach

DirecTL+ is a feature-rich, discriminative character transducer, which searches for a model-optimal sequence of character transformation rules for its input. The core of the engine is a dynamic programming algorithm capable of transducing many consecutive characters in a single operation, also known as a semi-Markov model. Using a structured version of the MIRA algorithm (McDonald et al., 2005), the training process assigns weights to each feature, in order to achieve maximum separation of the gold-standard output from all others in the search space.

DirecTL+ uses a number of feature templates to assess the quality of a rule: source context, target \textit{n}-gram, and joint \textit{n}-gram features. Context features conjoin the rule with indicators for all source character \textit{n}-grams within a fixed window of where the rule is being applied. Target \textit{n}-grams provide indicators on target character sequences, describing the shape of the target as it is being produced, and may also be conjoined with the source context features. Joint \textit{n}-grams build indicators on rule sequences, combining source and target context, and memorizing frequently-used rule patterns. An additional copy feature generalizes the identity function from source to target.

In order to incorporate language-model information from an unannotated corpus, Nicolai et al. (2015a) train a reranker on a held-out set. The reranker features include a corpus presence indicator, a normalized language model score, and the rank and normalized confidence score generated by the initial algorithm.

3 Methods

In this section, we describe our novel extensions: insertion handling, character-level target language model, and corpus frequency.

3.1 Insertion handling

Before DirecTL+ can learn a transduction model, the training source-target pairs are aligned using the unsupervised M2M aligner (Jiampojamarn et al., 2007). The alignment involves every source and target character. Because DirecTL+ does not support the insertion operation, the M2M aligner combines insertions with the neighboring substitutions, producing 1-to-many (1-M) links. In particular, 1-M links are strictly necessary when the target is longer than the source. Unfortunately, in low-data scenarios, the resulting alignments are often sub-optimal.

In order to improve the alignment quality, we allow the aligner to match target characters to null symbols (0-1 links), which explicitly model insertions, and make use of two post-processing steps to re-configure these alignments. We first make use of the method that Jiampojamarn and Kondrak (2010) refer to as “alignment aggregation”. This method restricts alignments to 1-1, but produces a set of \textit{n}-best alignment alternatives. These alignments are scored, normalized, and thresholded, and then combined in such a way that any disagreeing sequences between alignment candidates...
are merged. The original paper only considers aggregation for 1-1 and 1-0 links (i.e., substitution and deletion), but we modify their algorithm for 0-1 links, as well.

For example, the source sequence for the word *walkers* in Figure 2 consists of 5 phonemes, which must be aligned to 7 target letters. In this instance, the baseline approach is confused by the uncommon pronunciation of *al* in *walkers*, and incorrectly links the letter ‘a’ with the phoneme /w/ (the left-most alignment in the diagram). If insertions are allowed (the middle alignment), ‘a’ is correctly aligned to the /ə/ and ‘l’ is treated as an insertion. Finally, our algorithm finds more support for merging the insertion with the substitution operation that precedes it, rather than with the one that follows it. By contrast, the second insertion, which involves /ə/, is merged with the substitution that follows it (the right-most alignment).

### 3.2 Character-level language model

In order to incorporate a stronger character language model into DirecTL+, we propose an additional set of features that directly reflect the probability of the generated subsequences. We train a character-level language model on a list of types extracted from a raw corpus in the target language, applying Witten-Bell smoothing and backoff for unseen n-grams. During the generation process, while the transducer incrementally constructs target sequences character-by-character, the normalized log-likelihood score of the current output sequence is computed according to this character language model. For consistency with other sets of features, we convert those real-valued scores into binary indicators by means of binning. Features fire in a cumulative manner, and a final feature fires only if no bin threshold is met. For example, if a sequence has a log-likelihood of -0.85, the feature for -0.9 fires, as does the one for -0.975, and -1.05, etc.

For example, the candidate prediction *piuss* is scored higher than the correct *pierce* by the baseline DirecTL+ (Figure 3). However, our character language model assigns a low score to any output form that starts with the trigram *piu*, which causes *piuss* to fall out of the top-k list.

### 3.3 Corpus frequency counts

Our final extension can be described as a unigram word-level language model. The objective is to bias DirecTL+ towards generating output sequences that correspond to words observed in a large corpus. Since the output sequence can only be matched against a word list after the generation process is complete, we propose to estimate the final frequency count for each prefix considered during the generation process. Following Cherry and Suzuki (2009) we use a prefix trie to store partial words for reference in the generation phase. We modify their solution by calculating partial counts of each prefix in a word, using Equation 1, where *p* is a prefix, *w* is a word, and *Cw* is the count of a word.

\[
P_c = \sum_{w \ni \text{Prefix}(w, p)} \frac{C_w \ast \text{len}(p)}{\text{len}(w)}
\]

The pseudo-count of a prefix is equivalent to a weighted sum of all of the words in which the prefix occurs. Each word is weighted by the ratio of the prefix to the word. An example prefix trie is

![Figure 3: An example of how a character 4-gram model and word unigram model can improve transduction. Note that although we portray the extensions as part of a pipeline, their scores are incorporated jointly with DirecTL+’s other features. Black cells represent firing features.](image-url)
Figure 4: An example prefix trie. The approximated count of each prefix is a weighted sum of all leaf nodes that have the prefix as an ancestor.

shown in Figure 4. The prefix ‘f’ receives a partial count of 2.75 from “four”, 2.5 from “five”, and 0.4 from “fifty”, for a total count of 5.65. As with our language model features, unigram features are binned. A unigram feature fires if the count of the generated sequence surpasses the bin threshold, in a cumulative manner.

Let us consider an example of how our new features benefit a transduction model, shown in Figure 3. The top-n list produced by the baseline DirecTL+ for the input phoneme sequence /pİ@s/ fails to include the correct output pierce. However, after the new language model features are added, the correct form makes its way to the top predictions. Note that the new features combine with the original features of DirecTL+, so that the high unigram count of piece is not sufficient to make it the top prediction on the right side of the diagram. Only when both sets of new features are incorporated does the system manage to produce the correct form, as seen at the bottom of the diagram.

4 Related work

Sequence-to-sequence prediction is a general task that encompasses many important NLP processes.

The task of converting a word from a source to a target script on the basis of the word’s pronunciation is known as transliteration. Nicolai et al. (2015b) propose a combination of feature-rich transduction systems for transliteration. Recently, neural transliteration approaches follow the sequence-to-sequence model originally proposed for Neural Machine Translation (Jadidnejad, 2016; Rosca and Breuel, 2016). Similarly, the best submissions from the 2016 NEWS Shared Task on Transliteration (Duan et al., 2016) were neural methods.

The task of inflection generation has attracted much interest in recent years (Dreyer and Eisner, 2011; Durrett and DeNero, 2013; Nicolai et al., 2015a; Ahlberg et al., 2015). The most successful systems in the shared tasks on morphological reinflection (Potterell et al., 2016, 2017) were based on neural methods. However, in the low-resource track, non-neural systems were still competitive. Similarly, Aharoni and Goldberg (2017) augment an RNN with hard attention and explicit alignments between source and target, but have difficulty consistently improving upon the results of DirecTL+, even on larger datasets. Moreover, the strategies adopted by the neural approaches to deal with data sparsity often depend on modeling the identity operations, and cannot be easily transferred to tasks such as transliteration or phoneme-to-grapheme generation, where the source and target symbol sets have little or no overlap.

Ruzsics and Samardzic (2017) incorporate a language model trained on unannotated data into an RNN Seq2Seq model for the task of canonical word segmentation. However, their language model is only incorporated during decoding, whereas we jointly learn the influence of the language model with the DirecTL+’s other features.

Cognate projection, also referred to as cognate production, is the task of predicting the spelling of a hypothetical cognate word in a related language. Mulloni (2007) apply an SVM tagger trained on a list of pairs, which are aligned using the minimal edit-distance algorithm. Beinborn et al. (2013) use a lexicon list of the target language to filter non-words from the n-best list output of a character-level SMT model. Ciobanu (2016) combines sequence labelling and maximum-entropy reranker. We discuss and compare their results in Section 5.6.

Phoneme-to-grapheme conversion is the task
of predicting the spelling of a word from a sequence of phonemes that represent its pronunciation (Rentzepopulos and Kokkinakis, 1996). To the best of our knowledge, the state of the art on this task is the joint n-gram approach of Bisani and Ney (2008), who report improvements on the results of Galescu and Allen (2002) on NetTalk and CMUDict. We perform a comparison with their results on three lexicons in Section 5.9.

5 Experiments

In this section, we describe our experiments on four character-level sequence-to-sequence tasks: transliteration, inflection generation, cognate projection, and phoneme-to-grapheme conversion (P2G).

5.1 Setup

We evaluate DirecTLM (DTLM), an implementation of DirecTL+ modified with the extensions described in Section 3, against the baseline DirecTL+ (DirecTL+) as well as DirecTL+ augmented with reranking (DTLM+RR), as described in Section 2. The source-target pairs in the training sets are aligned with the M2M aligner (Jimpojamar et al., 2007). For both systems, the maximum allowed alignment link size is 2-2, without the extra insertion handling described in Section 3.1. The reranker is trained using 10-fold cross validation on the training set, using the method of Joachims (2002).

We train 4-gram character language models using the CMU language modeling toolkit\footnote{Our code will be made publicly available upon acceptance} with Witten-Bell smoothing. Word counts are determined from the first one million lines of the appropriate Wikipedia dumps, while the language models are constructed from the appropriate corpora from unimorph.org, except where noted otherwise. The Unimorph corpora consist of inflected word forms, and vary in size from 55,000 forms for Dutch to 383,000 for Spanish.

5.2 Comparison systems

We compare our results against two different systems: Sequitur (SEQ), which is a generative string transcription tool based on joint source and target n-grams (Bisani and Ney, 2008), and a character-level neural baseline implementation (RNN). Parameter tuning was performed on the same development sets as for DTLM.

Our baseline neural model uses the encoder-decoder architecture of Sutskever et al. (2014). The encoder is a bi-directional RNN applied to randomly initialized character embeddings. For low-resource training, we employ a hard-monotonic attention mechanism where the decoder at time step \( t \) only attends to the encoder’s hidden vector \( h_t \). We observe that with few training examples, hard attention produces a more accurate model than various soft attention mechanisms (Luong et al., 2015). The neural models are trained for two different random seeds using the Adam optimizer, embeddings of 100 dimensions, hidden units of size 200, and a beam of size 4. We report the average performance of the two models on the final test set.

5.3 Transliteration

Our transliteration data comes from the 2016 NEWS Shared Task on Transliteration. We evaluate on three language pairs: Japanese Katakana to English (JaEn), Thai to English (ThEn), and Arabic to English (ArEn). Since the original datasets contain thousands of training instances, we simulate a low-resource setting by randomly sampling 100 instances from each training set. Likewise, we randomly select 1000 instances as a development set, and 1000 as a test set. If there are more than one gold English target for a source word, we consider a transliteration to be correct if it produces any correct form.

The results in Table 1 show that our proposed extensions have a dramatic impact on the generation accuracy. In particular, the seamless incorporation of the target language model not only simplifies the model but also greatly improves the results with respect to the reranking approach. Although the absolute word accuracy numbers appear low, many of the generated forms that are counted as errors are actually very close to the gold references in the test set.

The Arabic data is different from the other two sets in that the source words are mostly native Arabic names and places, rather than English words and names expressed in a foreign script. In an additional experiment, we train the target character language models from transliterated Arabic, rather than the English Unimorph. We obtain this data by querying DBPedia for all personal names from
Table 1: Word-level accuracy on transliteration (in %) with 100 training instances.

<table>
<thead>
<tr>
<th>System</th>
<th>JaEn</th>
<th>ThEn</th>
<th>ArEn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc</td>
<td>F1</td>
<td>Acc</td>
</tr>
<tr>
<td>DirecTL+</td>
<td>3.5</td>
<td>64.6</td>
<td>4.3</td>
</tr>
<tr>
<td></td>
<td>14.1</td>
<td>84.8</td>
<td></td>
</tr>
<tr>
<td>DTL+RR</td>
<td>6.7</td>
<td>65.2</td>
<td>7.3</td>
</tr>
<tr>
<td></td>
<td>15.8</td>
<td>84.7</td>
<td></td>
</tr>
<tr>
<td>DTLM</td>
<td>11.9</td>
<td>65.1</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>20.2</td>
<td>86.9</td>
<td></td>
</tr>
<tr>
<td>RNN</td>
<td>0.3</td>
<td>2.9</td>
<td>7.0</td>
</tr>
<tr>
<td></td>
<td>6.3</td>
<td>46.5</td>
<td></td>
</tr>
<tr>
<td>SEQ</td>
<td>2.9</td>
<td>58.6</td>
<td>7.0</td>
</tr>
</tbody>
</table>

Table 2: Word-level accuracy (in %) on inflection generation with 100 training instances.

<table>
<thead>
<tr>
<th>System</th>
<th>EN</th>
<th>DE</th>
<th>ES</th>
<th>PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DirecTL+</td>
<td>90.6</td>
<td>66.0</td>
<td><strong>68.2</strong></td>
<td>45.2</td>
</tr>
<tr>
<td>DTL+RR</td>
<td><strong>90.9</strong></td>
<td>67.6</td>
<td>67.0</td>
<td>49.7</td>
</tr>
<tr>
<td>DTLM</td>
<td>90.3</td>
<td>67.6</td>
<td>68.1</td>
<td><strong>51.1</strong></td>
</tr>
<tr>
<td>RNN</td>
<td>79.3</td>
<td>12.1</td>
<td>21.4</td>
<td>10.9</td>
</tr>
<tr>
<td>CLUZH</td>
<td>90.4</td>
<td><strong>68.1</strong></td>
<td>66.4</td>
<td>47.9</td>
</tr>
</tbody>
</table>

5.4 Inflection generation

Our morphological data comes from the recent CoNLL-SIGMORPHON Shared Task on Reflection (Cotterell et al., 2017). We use the datasets from the low-resource setting of the inflection generation sub-task, in which the training sets are composed of 100 source lemmas with inflection tags and the corresponding inflected forms. For example, the Spanish infinitive liberar with the tag V;IND;FUT;2;SG should produce the word-form liberaráds. We supplement the training data with 100 synthetic “copy” instances that simply transform the target string into itself. This modification, which is known to help in transduction tasks where the source and target are nearly identical, replaces our modified alignment for the inflection generation experiments only. Along with the 100 training forms provided by the shared task, we also use the task’s development and test sets, each consisting of 1000 instances.

Table 2 shows the results on English (EN), German (DE), Spanish (ES), and Polish (PL). Since Sequitur is ill-suited for this type of transduction, we instead include the best results of the CLUZH team (Makarov et al., 2017), which was the winner of the shared task. The differences between the three variants of DirecTL+ are small, but the results compare favorably to the CLUZH results, which was also represented by several systems in the shared task. On Polish, in particular, the performance of DTLM is outstanding.

From error analysis, we conclude that the introduction of a stronger target language model helps avoid the generation of non-words. For example, the German non-word knechtetet is corrected to knechtet. While ‘tt’ is a valid bigram in German, the 4-gram ‘cht’ has a very low likelihood. On the other hand, the unigram word model is responsible for correcting the non-word Wolkes to Wolke, even though the former violates no phono-tactic constraints.

5.5 Cognate projection

We evaluate the cognate projection systems on three diverse language pairs. The first set, which contains 1013 forms\(^3\), represents a close genetic sibling relationship of two languages from the Germanic family. The second set of 438 forms is similar to the first, but comes from the Slavic family\(^4\), and represents two closely-related languages that are written in different scripts (Roman vs. Cyrillic). The third set, which contains 601 pairs of reconstructed Vulgar Latin and Italian words (Boyd-Bowman, 1980) represents a genetic mother-daughter relationship from the Romance family of languages, where regular sound correspondences can be used by linguists to generate the modern reflexes from the proto-forms. Each set is divided into 100 forms for training and 100 for development, with the rest set aside as a test set. We attach our Germanic and Slavic data to this submission, with the intention of making it available to the community.

The results are shown in Table 3. DirecTL+ performs better on average than SEQ, and much better than our RNN baseline. The incorporation of corpus statistics substantially improves the accuracy, with DTLM outperforming DTL+RR on all datasets.

\(^3\)https://en.wiktionary.org/wiki/Appendix:List_of_German_cognates_with_English
Table 3: Word-level accuracy (in %) on cognate projection with 100 training instances.

<table>
<thead>
<tr>
<th>System</th>
<th>EN-DE</th>
<th>RU-PL</th>
<th>VL-IT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DirecTL+</td>
<td>4.3</td>
<td>23.5</td>
<td>39.2</td>
</tr>
<tr>
<td>DTL+RR</td>
<td>7.1</td>
<td>32.8</td>
<td>43.6</td>
</tr>
<tr>
<td>DTLM</td>
<td><strong>10.9</strong></td>
<td><strong>42.0</strong></td>
<td><strong>49.6</strong></td>
</tr>
<tr>
<td>RNN</td>
<td>3.8</td>
<td>6.3</td>
<td>17.7</td>
</tr>
<tr>
<td>SEQ</td>
<td>9.2</td>
<td>22.3</td>
<td>36.9</td>
</tr>
</tbody>
</table>

Table 4: Word-level accuracy (in %) on cognate projection with many training forms.

<table>
<thead>
<tr>
<th>System</th>
<th>EN-DE</th>
<th>EN-RU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>top-1</td>
<td>top-5</td>
</tr>
<tr>
<td>BZG-13</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DirecTL+</td>
<td>51.0</td>
<td>58.1</td>
</tr>
<tr>
<td>DTLM</td>
<td>53.7</td>
<td><strong>61.6</strong></td>
</tr>
</tbody>
</table>

5.6 Cognate projection with larger sets

In this section, we evaluate DTLM on the larger cognate projection datasets from a previous study of Beinborn et al. (2013). The datasets were created by applying romanization scripts and string similarity filters to translation pairs extracted from Bing. The resulting word-lists contain mostly lexical loans from Latin, Greek, and English, and, unfortunately, many compound words that share only one morpheme (e.g., *informatics* and *informationswissenschaft*).

We report the results of experiments on English-German (EN-DE) and English-Russian (EN-RU). The first language pair has been used in other related works, as well as in our low-data experiments described in Section 5.5. The second pair is interesting because it involves languages written in different alphabets. The EN-DE set consists of 7944 training pairs and 1002 test pairs, while the EN-RU set contains 4739 training and 127 test pairs. In each case, we reserve 10% of the training set to tune parameters.

In order to alleviate the noise problem in the EN-DE dataset, we disregard 139 training instances that the M2M aligner fails to align with the max. 2-2 parameter setting. Unfortunately, Beinborn et al. (BZG-13) only report mean reciprocal rank, and top-5 accuracy (i.e., the presence of the correct form in the 5-best list); we also report the word accuracy (top-1) for all experiments.

Table 4 shows the results. DTLM substantially outperforms BZG-13 on both datasets. To provide additional context, Mulloni (2007) achieves a top-1 word accuracy of 30.3% using a different set that includes 1683 English-German training pairs, while Ciobanu (2016) reports an MRR of 28% on the BZG-13 dataset (since the latter uses no target corpus, it should be compared to the baseline DirecTL+ result). We conclude that our results establish a new state of the art on cognate projection.

In terms of error analysis, we are encouraged to observe that even with much larger training sets, the new LM component still helps, and that the more flexible alignments contribute as well. For example, the projection of *Kenyan* improves from *kenyisch* to *kenianisch* thanks to the correct alignment of ‘an’ ‘anisch’ in the training data, which is achieved through a merger of multiple insertion operations.

5.7 Phoneme-to-grapheme conversion

For P2G, we extract word pairs from the CELEX lexical database (Baayen et al., 1995) for English (EN), Dutch (NL), and German (DE). To simulate the low-resource setting, we randomly sample 100 training examples, and construct development and held-out sets of 1000 example pairs, mirroring the setup of the inflection shared task.

Table 5 shows that our modifications yield substantial gains for all three languages, with consistent error reductions of 15-20% over the reranking approach. Despite only training on 100 words, the system is able to convert phonetic transcriptions into completely correct spellings for a large fraction of words, even in English, which is notorious for its idiosyncratic orthography.

The language model corrects unlikely character sequences; for example, *blushi* is replaced by *blushed*. Likewise, the unigram features help disambiguate some hard to predict phonemes, such as the reduced vowels: a and t. For example, given [tɬɪmɒtri] as the input, the baseline DirecTL+ generates *tlemetry* instead of *telemetry*, but the error is corrected after incorporating the corpus unigram features.

5.8 Ablation study

We investigate which of our contributions is most influential to the P2G task. We begin with the complete DTLM system, as reported in Table 5, disabling a single feature in each row. The results are reported in Table 6.
Table 5: Word-level accuracy (in %) on phoneme-to-grapheme conversion with 100 training instances.

<table>
<thead>
<tr>
<th>System</th>
<th>EN</th>
<th>NL</th>
<th>DE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DirecTL+</td>
<td>13.9</td>
<td>30.6</td>
<td>33.5</td>
</tr>
<tr>
<td>DTL+RR</td>
<td>25.3</td>
<td>32.6</td>
<td>51.5</td>
</tr>
<tr>
<td>DTLM</td>
<td>38.8</td>
<td>46.7</td>
<td>61.8</td>
</tr>
<tr>
<td>RNN</td>
<td>2.8</td>
<td>3.5</td>
<td>4.5</td>
</tr>
<tr>
<td>SEQ</td>
<td>15.9</td>
<td>30.5</td>
<td>28.6</td>
</tr>
</tbody>
</table>

Table 6: Ablation test on P2G data with 100 training instances.

<table>
<thead>
<tr>
<th>System</th>
<th>EN</th>
<th>NL</th>
<th>DE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTLM</td>
<td>38.8</td>
<td>46.7</td>
<td>61.8</td>
</tr>
<tr>
<td>-Language model</td>
<td>30.5</td>
<td>42.5</td>
<td>50.6</td>
</tr>
<tr>
<td>-Freq</td>
<td>25.2</td>
<td>38.3</td>
<td>57.1</td>
</tr>
<tr>
<td>-Insertions</td>
<td>30.9</td>
<td>45.8</td>
<td>52.6</td>
</tr>
</tbody>
</table>

We see that each of our contributions significantly improves P2G accuracy. With the exception of German, we see that the word list leads to the largest increase in accuracy. This decreased (although still noticeable) utility of the German wordlist may be related to the increased morphological complexity of German. German has more inflectional productivity than either Dutch or English, and will thus have a much sparser word list. Contrarily, we note that our other contributions, particularly the language model, have a larger impact on German than the other languages, possibly due to German's more regular spelling system.

We also see an interesting trend with the insertion handling. Recall that insertions can be aligned either implicitly through 1-many links, or explicitly through 0-1 links. We note that while DTLM sees a marked improvement through explicit insertion handling, the same is not true for DirecTL+. Without our corpus features, DirecTL+ performs best with a 1-to-many handling of insertions. We hypothesize that when we incorporate a strong language model, the extra flexibility of a 1-1 alignment benefits the generation module by allowing it to make small changes to better fit the corpus. Without the constraints of a corpus, however, the smaller alignments provide too much flexibility, allowing DirecTL+ to produce unnatural sequences.

### 5.9 Large-scale P2G

The goal of our final experiment is to measure the extent to which our contributions improve transcription when training data is plentiful. We attempt to replicate the P2G experiments reported by (Bisani and Ney, 2008). The data comes from three lexicons on which we conduct 10-fold cross validation: English NetTalk (Sejnowski and Rosenberg, 1993), French Brulex (Mousty and Radeau, 1990), and English CMUDict (Weide, 2005). These corpora contain 20,008, 24,726, 113,438 words, respectively, in both orthographic and phonetic notations. We note that CMUDict differs from the other two lexicons in that it is much larger, and contains predominantly names, as well as alternative pronunciations. When the training data is that abundant, there is less to be gained from improving the alignment or the target language models, as they are already adequate in the baseline approach.

Table 7 shows the comparison of the results. We omit the reranking approach because of the complexity of its setup on large data sets. For similar reasons, we only run our RNN model on NetTalk, obtaining a word accuracy of 55.6%. Sequitur outperforms the baseline DirecTL+, which we attribute to the insertion handling issue.\(^5\) However, DTLM substantially outperforms Sequitur on both the NetTalk and Brulex data sets, with smaller gains on the much larger CMUDict. We conclude that our results advance the state of the art on the task of phoneme-to-grapheme conversion.

### 6 Conclusion

We have presented DTLM, an enhanced discriminative transducer, which is particularly effective in low-data scenarios. The proposed modifications address two shortcomings of DirecTL+: its pro-

---

\(^5\)The P2G results obtained by Sequitur in our experiments are slightly lower than those reported in the original paper, which is attributable to differences in data splits, tuned hyperparameters, and/or the presence of stress markers in the data.
cessing of insertions, and its target-side language model. The modifications allow us to simplify the transduction approach by collapsing the reranking pipeline, which results in a better output accuracy. We have demonstrated that a single tool with a strong language model can yield substantial improvements across multiple transduction tasks and language sets.

References


Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2010. Integrating joint n-gram features into a discriminative training network. In NAACL-HLT.


