Leveraging supplemental representations for sequential transduction

Aditya Bhargava
Department of Computer Science
University of Toronto
Toronto, ON, Canada, M5S 3G4
aditya@cs.toronto.edu

Grzegorz Kondrak
Department of Computing Science
University of Alberta
Edmonton, AB, Canada, T6G 2E8
kondrak@cs.ualberta.ca

Abstract

Sequential transduction tasks, such as grapheme-to-phoneme conversion and machine transliteration, are usually addressed by inducing models from sets of input-output pairs. Supplemental representations offer valuable additional information, but incorporating that information is not straightforward. We apply a unified reranking approach to both grapheme-to-phoneme conversion and machine transliteration demonstrating substantial accuracy improvements by utilizing heterogeneous transliterations and transcriptions of the input word. We describe several experiments that involve a variety of supplemental data and two state-of-the-art transduction systems, yielding error rate reductions ranging from 12% to 43%. We further apply our approach to system combination, with error rate reductions between 4% and 9%.

1 Introduction

Words exist independently of writing, as abstract entities shared among the speakers of a language. Those abstract entities have various representations, which in turn may have different realizations. Orthographic forms, phonetic transcriptions, alternative transliterations, and even sound-wave spectrograms are all related by referring to the same abstract word and they all convey information about its pronunciation.

Figure 1 shows various representations of the word *Dickens*. The primary (canonical) orthographic representation of the word corresponds to the language to which the word belongs. The secondary orthographic representations in different writing scripts are transliterations of the word, which exhibit phonological adaptation to the target language. The various phonetic transcriptions consist of sequences of phonemes representing the pronunciation of the word. Transcription schemes may differ in the number and coverage of various phonemes, as well as the choice of the underlying speech variety. The spoken pronunciation (represented by the waveform) presents a common latent influence on the representations.

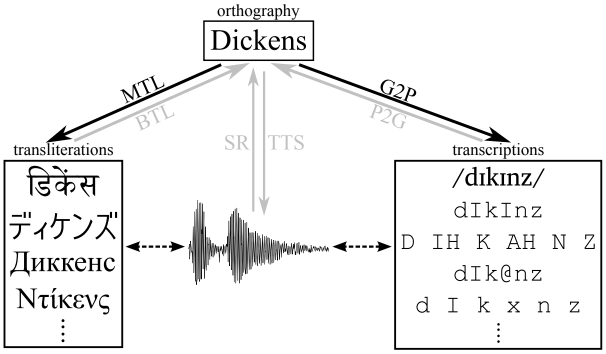

Figure 1: Several NLP tasks involve conversion between various word representations. The tasks on which we focus are shown in black.

Several well-known NLP tasks involve matching, alignment, and conversion between different word representations. Grapheme-to-phoneme conversion (G2P) aims at generating a transcription of a word from its orthographic representation. The reverse task is phoneme-to-grapheme conversion (P2G). Machine transliteration (MTL) deals with conversion between orthographic representations in different writing scripts; forward transliteration proceeds from the primary representation to secondary representations, while the reverse task is called back-transliteration (BTL). The conversion between a sound and an orthography is the goal of text-to-speech synthesis

2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 396–406,
Montréal, Canada, June 3-8, 2012. ©2012 Association for Computational Linguistics

(TTS) and speech recognition (SR), where transcriptions are often used as intermediate forms.

Although both MTL and G2P take orthographic representations as input, the two tasks are rarely considered in conjunction. Traditionally, G2P has been investigated in the context of text-to-speech systems, while MTL is of interest to the information retrieval and machine translation communities. In addition, unlike phonetic transcription schemes, which are often specific to a particular pronunciation lexicon, writing systems are well-standardized, with plenty of transliteration examples available in text corpora and on the Web (Kumaran et al., 2010b). While the goal of G2P is producing a maximally faithful representation of the pronunciation, transliterations are often influenced by other factors, such as the phonological constraints of the target language, the orthographic form in the source language, the morphological adaptations related to the translation process, and even the semantic connotations of the output in the case of logographic scripts. In spite of those differences, both transcriptions and transliterations contain valuable information about the pronunciation of the word.

In this paper, we show that it is possible to improve the accuracy of both G2P and MTL by incorporating supplemental representations of the word pronunciation. Our method is based on SVM reranking of the n-best output lists of a base transduction system, with features including similarity scores between representations and n-grams derived from accurate alignments. We describe a series of experiments in both G2P and MTL contexts, demonstrating substantial reductions in error rate for these base tasks when augmented with various supplemental representations. We then further test the effectiveness of the same approach for combining the results of two independent base systems.

2 Related work

Because of its crucial role in speech synthesis, grapheme-to-phoneme conversion has been researched extensively. Most out-of-vocabulary words are names, which often exhibit idiosyncratic pronunciation (Black et al., 1998). Excepting languages with highly transparent orthographies, the number of letter-to-sound rules appears to grow geometrically with the lexicon size, with no asymptotic limit (Kominek and Black, 2006). A number of machine learning approaches have been proposed for G2P, including neural networks (Sejnowski and Rosenberg, 1987), instance-based learning (van den Bosch and Daelemans, 1998), pronunciation by analogy (Marchand and Damper, 2000), decision trees (Kienappel and Kneser, 2001), hidden Markov models (Taylor, 2005), joint n-gram models (Bisani and Ney, 2008), and online discriminative learning (Jiampojamarn et al., 2008). The current state-of-the-art is represented by the latter two approaches, which are available as the SEQUITUR and DIRECTL+ systems, respectively.

Machine transliteration has also received much attention (Knight and Graehl, 1998; Li et al., 2004; Sproat et al., 2006; Klementiev and Roth, 2006; Zelenko and Aone, 2006). In the last few years, the Named Entities Workshop (NEWS) Shared Tasks on Transliteration have been the forum for validating diverse approaches on common data sets (Li et al., 2009; Li et al., 2010; Zhang et al., 2011). Both SEQUITUR and DIRECTL+, originally G2P systems, have been successfully adapted to MTL (Finch and Sumita, 2010; Jiampojamarn et al., 2010b).

Most of the research on both G2P and MTL assumes the existence of a homogeneous training set of input-output pairs. However, following the pivot approaches developed in other areas of NLP (Utiyama and Isahara, 2007; Cohn and Lapata, 2007; Wu and Wang, 2009; Snyder et al., 2009), the idea of taking advantage of other-language data has recently been applied to machine transliteration. Khapra et al. (2010) construct a transliteration system between languages A and B by composing two transliteration systems $A \rightarrow C$ and $C \rightarrow B$, where C is called a *bridge* or *pivot* language, resulting in a relatively small drop in accuracy. Zhang et al. (2010) and Kumaran et al. (2010a) report that combinations of pivot systems $A \rightarrow C \rightarrow B$ with direct systems $A \rightarrow B$ produce better results than using the direct systems only. The models, which are composed using a linear combination of scores, utilize a single pivot language C, and require training data between all three languages A, B, and C. However, such a pivot-based framework makes it difficult to incorporate multiple pivot languages, and has shown most promising results for cases in which data for the original $A \rightarrow B$ task are limited. Lastly, Finch and Sumita (2010) developed an MTL approach that combined the output

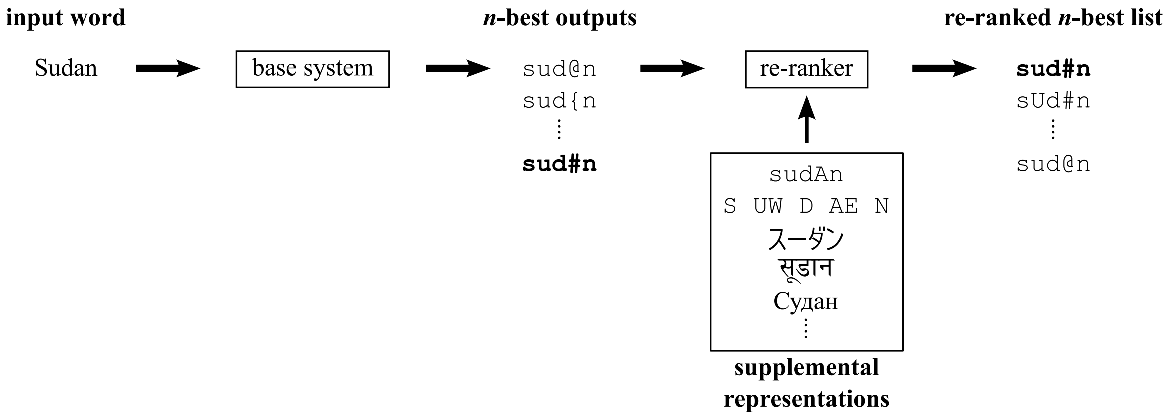

Figure 2: An overview of our approach on an example from the G2P task. The correct output is shown in bold.

of two systems using a linear combination of system scores and a manually-tuned weight.

On the G2P side, Loots and Niesler (2009) investigate the problem of leveraging transcriptions from a different dialect of English, while Bhargava and Kondrak (2011) focus on leveraging transliterations from multiple writing scripts. Bhargava et al. (2011) show that the reranking method proposed by Bhargava and Kondrak (2011) can increase the accuracy of MTL as well. In this paper, we aim to confirm the generality of the same method by testing it on a broad range of tasks: *a*) leveraging *transcriptions* for both G2P and MTL; *b*) utilizing supplemental transcriptions and transliterations *simultaneously*; *c*) improving G2P *in general*, rather than just G2P of names; and *d*) combining different transduction systems.

3 Leveraging supplemental data

Incorporating supplemental information directly into an existing system is not always feasible. With generative approaches, one would have to find some way of modelling the relationship between the system inputs, outputs, and the supplemental data. Discriminative approaches are not necessarily easier: DirecTL+, a discriminative G2P system, needs to be able to generate features on-the-fly for partial grapheme-phoneme sequence pairs during the decoding. Instead, we integrate an existing system as a black box that generates n-best lists of candidate outputs, resulting in a modular and general *post hoc* approach that can be applied to multiple tasks and settings.

3.1 Task definition

The task is to convert an input string s into a target string t, where both strings are representations of a word w. In G2P, s is a string of graphemes while t consists of phonemes; in MTL, both s and t are grapheme sequences, although in different scripts. We assume that we have a base system $T\,(s)$ that attempts this task and produces an n-best list of outputs $\widehat{t_1}, \widehat{t_2}, \ldots, \widehat{t_n}$ for the input s. T is imperfect, i.e., the correct output t may appear in a position in the list other than the first. It is reasonable to expect that such a system also provides a list of scores corresponding to the outputs. We further assume that we have access to supplemental representations of w; both transliterations and transcriptions may serve this purpose. Our objective is to improve the accuracy on the task in question with respect to the base system $T(s)$.

3.2 Reranking

For the purpose of exposition, we reiterate here the particulars of the reranking approach of Bhargava and Kondrak (2011) that we apply to the various tasks and supplemental data sources. The method uses SVM reranking of the n-best lists produced by the base system in order to to move the correct output to the top of the list using supplemental data. SVM reranking (Joachims, 2002) facilitates the exploitation of multiple sources of supplemental data, as shown in Figure 2. The feature construction process is performed for each candidate output in the n-best list, as well as each pairing of a candidate output with a supplemental representation. The features used for reranking may or may not overlap with the features used by the base system. While we focus on the G2P and MTL tasks in this paper, this method is general enough as to potentially be applied to other sequence prediction tasks.

3.3 Alignment

In order to construct the feature vectors, we need the alignments between the alternative representations of the same word. For the alignment of supplemental data with candidate outputs, we apply M2M-ALIGNER (Jiampojamarn et al., 2007). We use the same method for the alignment between the input and the candidate outputs, unless the base system already provides this information.

M2M-ALIGNER is a generalization of the learned edit distance algorithm of Ristad and Yianilos (1998). It iteratively refines the alignment of a set of string pairs in an unsupervised manner using the expectation-maximization (EM) approach. In addition to the alignment, M2M-ALIGNER produces an alignment probability, which reflects the similarity between two strings. Intuitively, if two strings contain symbols or n-grams that often co-occur in the training data, their alignment score will be higher. The strings in question are often of completely different scripts, which precludes the application of standard similarity measures such as Levenshtein distance.

3.4 Score features

The similarity of candidate outputs to alternative representations of a word is probably the most intuitive feature for reranking. We include a real-valued similarity feature for each pairing between a supplemental representation and a candidate output, which is set according to the M2M-ALIGNER alignment score.

Another important set of features are the confidence scores assigned to each candidate output by the base system. In addition to the original scores, we also include a set of features that indicate the *differences* between scores for all pairs of outputs in the n-best list. This allows the reranker to incorporate a notion of relative confidence with respect to the other candidate outputs. Similarly, we compute differences between the similarity scores of candidate outputs and supplemental representations.

3.5 N-gram features

Following (Jiampojamarn et al., 2010a), we include several types of n-gram features. The features are defined on substrings occurring in pairs of aligned strings. Each feature is binary, indicating the presence or absence of the particular feature type in the given aligned pair, which could be either the original base system's input and output, or else a candidate output and a supplemental representation.

We can divide the n-gram features into four categories. *Context features* bind an output symbol with input n-grams in a focus window centred around the input-output alignment; the input n-grams represent the context in which the output character is generated. *Markov features* are n-grams of output symbols, which allow previously generated output characters to influence the current output character. *Linear chain features* associate the context and Markov features. *Joint n-gram features* combine aligned input and output n-grams of the same length on both sides.

In the standard string transduction task, the output string t is generated incrementally from the input s. In contrast, in the reranking setting, both strings are complete and available. This allows us to *reverse* the direction of the context and linear chain features, allowing us to associate output n-grams with single input symbols. In addition, we can apply those features in both directions across candidate outputs and supplemental representations, further increasing the amount of information provided to the reranker.

4 Experiments

Our experiments aim at comprehensive evaluation of the reranking approach on both MTL and G2P tasks, employing various supplemental representations. Relevant code and scripts associated with our experimental results are available online[1].

4.1 Data

We extract transcriptions from two lexica: Combilex (Richmond et al., 2009), which includes both Received Pronunciation (RP) and General American (GA) pronunciation variants, and CELEX (Baayen et al., 1996), which includes RP only. After discarding duplicates and letter diacritics, the total number of

[1] http://www.cs.toronto.edu/~aditya/g2p-tl-rr

Language	Corpus size	Japanese Overlap
Bengali	13,624	2,152
Chinese	40,214	14,056
Hebrew	10,501	3,997
Hindi	13,427	2,507
Japanese	28,013	—
Kannada	11,540	2,170
Korean	7,778	7,733
Persian	12,386	4,047
Tamil	11,642	2,205
Thai	28,932	10,378

Table 1: The number of unique entries in each transliteration corpus, and the number of common single-word entries (overlap) with the Japanese corpus.

word-transcription pairs are 114,094 for Combilex, and 66,859 for CELEX. We use 10% of the data for development, 10% for testing, and the remaining 80% for training. The development set is merged with the training set for final testing.

Our transliteration data come from the shared tasks of the 2011 NEWS workshop (Zhang et al., 2011). The number of entries in each transliteration corpus is shown in the middle column of Table 1.

4.2 Base systems

In order to verify the generality of our approach, we perform all experiments using two different base transduction systems described in Section 2: SEQUITUR and DIRECTL+. Both systems are set to provide 10-best output lists along with scores for each output.[2] SEQUITUR is modified to provide log-probabilities instead of regular probabilities. DIRECTL+ is run with the complete set of features described by Jiampojamarn et al. (2010a). System parameters, such as maximum number of iterations, are determined during development.

M2M-ALIGNER is used throughout for the alignment of various representations. The aligner is trained on an intersection of a relevant pair of data sets. For example, the intersection of the English-to-Japanese and English-to-Hindi corpora on the basis of common entries on the English side yields a corpus matching Japanese transliterations with Hindi transliterations. M2M-ALIGNER, after having been trained on this corpus, is able to produce a similarity score for an arbitrary Japanese-Hindi pair. We set a lower limit of -100 on the M2M-ALIGNER log-probabilities, and use the default of 2-2 alignments; deletions are enabled for the supplemental data side of the alignment.

4.3 MTL experiments

When faced with the task of transliterating a word from the original script to a secondary script, we would like to leverage the information encoded in transliterations of the same word that are available in other scripts. For example, consider the problem of automatically generating a Wikipedia stub article[3] in Hindi about guitarist John Petrucci. We assume that we have access to an MTL system trained on the English-Hindi transliterations, but we also want to take advantage of the existing transliterations of the name that are easy to extract from the corresponding articles on the topic in Japanese and other languages. In this case, the orthography of the last name reflects its Italian origins, but the pronunciation depends on its degree of assimilation to English phonology. This type of information is often difficult to determine even for humans, and we posit that it may be inferred from other transliterations.

Similarly, phonetic transcriptions more directly encode the pronunciation and thus present an important resource for exploitation. In fact, some transliteration systems use a phonetic transcription as an intermediate representation (Knight and Graehl, 1998), although these methods do not generally fare as well as those that perform the transliteration process directly (Al-Onaizan and Knight, 2002; Li et al., 2009). Transcriptions are often available; larger pronunciation dictionaries contain tens of thousands of entries, including some proper names (for which machine transliteration is most relevant), and many names in Wikipedia are accompanied by an IPA transcription.

Our first experiment aims at improving the transliteration accuracy from English to Japanese Katakana. The English-Japanese corpus has one of the largest overlaps (number of entries with a common input)

[2] While running times prevented us from extensively analyzing reranking performance vs. n-best list size, our initial tests produced almost identical results for $n = 5$, $n = 10$, and $n = 20$.

[3] A stub article is a skeleton article with little content.

	SEQUITUR		DIRECTL+	
	Acc.	ERR	Acc.	ERR
BASE	49.6		51.1	
RERANKED	56.2	13.5	57.3	12.7
ORACLE	85.0	70.3	80.4	60.0

Table 2: Word accuracies and error rate reductions (ERR) in percentages for English-to-Japanese MTL augmented by corresponding transliterations from other languages. BASE is the base system while RERANKED represents the same system with its output reranked using supplemental transliterations. ORACLE represents an oracle reranker.

with the other transliteration and transcription corpora, the former of which is shown in Table 1. In total, there are 18,505 entries for which at least one transliteration from a non-Japanese language is available and 6,288 for which at least one transcription is available. The reranker is trained on an intersection of the English-Japanese training set and the supplemental data; similarly, the reranking test set is an intersection of the English-Japanese test set and the supplemental data. Note that we compute word accuracy on these intersected sets, so the results of the base systems that we report here may not represent their performance on the full data set.

Table 2 shows the results[4] on the test set of 1,891 entries, including the performance of an oracle (perfect) reranker for comparison. This same approach applied to the English-to-Hindi transliteration task yields an error rate reduction of 9% over the base performance of DIRECTL+ (Bhargava et al., 2011)[5], which confirms that our reranking method's applicability is not limited to a particular language.

In the second experiment, instead of supplemental transliterations, we use supplemental transcriptions from the RP and GA Combilex corpora as well as CELEX. The number of common elements with the English-Japanese transliteration corpus was 6,288 for Combilex and 2,351 for CELEX; in total, there were 6,384 transliteration entries for which at least one transcription was available. Table 3 shows the results, giving a similar error rate reduction as for using supplemental transliterations.

	SEQUITUR		DIRECTL+	
	Acc.	ERR	Acc.	ERR
BASE	57.9		58.6	
RERANKED	65.6	18.4	63.9	12.8
ORACLE	89.9	51.5	84.6	62.6

Table 3: Word accuracies and error rate reductions (ERR) in percentages for English-to-Japanese MTL augmented by corresponding transcriptions.

Surprisingly, if we proceed to the next logical step and use both transcriptions and transliterations as supplemental representations *simultaneously*, the error rate reduction is slightly lower than in the above two experiments. This difference is so small as to be statistically insignificant. We have no convincing explanation for this phenomenon, although we note that, in general, significant heterogeneity in data can increase the difficulty of a given task.

4.4 G2P experiments

Consider the example of an automatic speech synthesis system tasked with generating an audio version of a news article that contains foreign names. Often, foreign versions of the same news article already exist; in these, the name will have been transliterated. These transliterations could then be leveraged to guide the system's pronunciation of the name. The same is conceivable of other types of words, although transliterations are generally mostly available for names only.

On the other hand, transcription schemes are not consistent across different pronunciation lexica. Their phonemic inventories often differ, and it is not always possible to construct a consistent mapping between them. In addition, because of pronunciation variation and dialectal differences, a substantial fraction of transcriptions fail to match across dictionaries. Nevertheless, if a phonetic transcription is already available, even in an alternative format, it could facilitate the task of generating a new pronunciation.

The first G2P experiment concerns the application of supplemental transcriptions. The goal is to quantify the improvements achieved using our reranking

[4] Unless otherwise noted, all improvements reported in this paper are statistically significant with $p < 0.01$ using the McNemar test.

[5] Note that this result is computed over the full English-Hindi data set, so is in fact slightly diluted compared to the results we present here.

approach, and to compare reranking to two other methods of utilizing supplemental transcriptions, to which we refer as MERGE and P2P, respectively.

MERGE implements the most intuitive approach of merging different lexica into a single training set. In order to make this work, we first need to make sure that all data is converted to a single transcription scheme. Combilex and CELEX make different distinctions among phonemes, making it unclear how some phonemes might be mapped from CELEX into Combilex; fortuitously, the opposite conversion is more agreeable.[6] This allows us to convert Combilex to CELEX's format via a simple rule-based script and then merge the two corpora together. This provides an alternative method against which we can compare our reranking-based approach which would treat Combilex as a source of supplemental representations.

P2P is a phoneme-to-phoneme conversion approach inspired by the work of Loots and Niesler (2009). In that approach, a phoneme-to-phoneme model is derived from a training set of phonetic transcription pairs representing two different pronunciation lexicons. We use such model to convert the Combilex transcriptions to the scheme used by CELEX for the words that are missing from CELEX. Where Loots and Niesler (2009) use decision trees for both the base system and the corpus converter, we use the much higher-performing state-of-the-art SEQUITUR and DIRECTL+ systems.

The two transcription corpora have 15,028 entries in common. As with the MTL experiments, the reranker is trained on an intersection of the Combilex G2P data and the supplemental data.

The results on the intersected set of 1,498 words are shown in Table 4. We can see that merging the corpora provides a clear detriment in performance for data where an alternative transcription is available from another corpus. Even if we look at the full CELEX test set (as opposed to the intersected subset used in Table 4), DIRECTL+ trained only on CELEX achieves 93.0% word accuracy on the CELEX test set where DIRECTL+ trained on CELEX merged with Combilex achieves 87.3%. Evidently, the disparate conventions of the two corpora "confuse" the base G2P systems. In contrast, our reranker performs well, yielding spectacular error reductions of 32% and 42%.

	SEQUITUR		DIRECTL+	
	Acc.	ERR	Acc.	ERR
BASE	87.3		88.1	
MERGE	74.2	—	71.6	—
P2P	85.7	—	87.0	—
RERANKED	92.7	42.9	92.0	32.6
ORACLE	97.6	81.2	96.7	72.5

Table 4: Word accuracies and error rate reductions (ERR) in percentages for CELEX G2P augmented by Combilex transcriptions.

The differences between the two corpora account for the inadequate performance of the P2P approach. Inducing a full transduction model requires much more training data that simply reranking the existing outputs, but in this case models for these two approaches (P2P and reranking) are trained on the same amount of data. Furthermore, when the supplemental transcription is radically different from the n-best outputs, the alignment simply fails, and the reranking approach gracefully falls back to the original G2P model. In contrast, the P2P approach has no such option.

It may be interesting to note what happens when the P2P model is replaced with our rule-based Combilex-to-CELEX converter. Such an approach has the advantage of being fast and not dependent on the training of any base system. However, it achieves only 64.8% word accuracy, which is lower than any of the results in Table 4. Clearly, a simple mapping script fails to capture the differences between the corpora.

Turning to supplemental transliterations, Bhargava and Kondrak (2011) have already shown that supplemental transliterations can improve G2P accuracy on *names*. It is interesting to verify whether this conclusion also applies to other types of words that occur in the G2P data set. Performing this test with DIRECTL+ as the base system shows good error rate reduction on names (about 12%) as reported, but a much smaller statistically insignificant error rate re-

[6] In particular, Combilex distinguishes between [l] and the velarized ("dark") [ɫ]. These can be collapsed into the single /l/ phoneme for CELEX, but it is not clear how to handle the conversion in the reverse direction.

duction on core vocabulary words (around 2%). In other words, the supplemental transliterations are able to help only for names.

This discrepancy is attributable to the fact that names (and, more generally, named entities) are the *raison d'être* of transliterations. Because the process of transliteration occurs primarily for names that must be "translated" phonetically, we expect transliterations' utility as supplemental representations to apply mostly for names. The smaller number of transliterations for core vocabulary words also makes it difficult for any system to learn how to apply transliterations of such words.

4.5 Base system matters

While our SVM reranking approach demonstrates significant improvements for all tasks and all tested base systems, the *magnitude* of the performance increase *is* dependent on the base system. In particular, we see a common thread recurring throughout all experiments: SEQUITUR sees higher improvements than does DIRECTL+. Although reranking treats the base system as a black box, we are limited by the amount of room for improvement available in the base system's outputs. Our results above show that the performance of an oracle reranker (a reranker that automatically selects the correct output from the n-best list) is consistently higher for SEQUITUR than for DIRECTL+. Higher oracle reranker scores indicate greater reranking potential, and we observe a corresponding higher error reduction, sometimes leading SEQUITUR to outperform DIRECTL+ after reranking despite having been the lower performer prior to reranking.

We hypothesize that another reason for the greater influence of reranking on SEQUITUR is the fact that the reranker's features are related to those used in DIRECTL+. Because SEQUITUR implements a diametrically different, generative approach to transduction, it benefits more from reranking. However, DIRECTL+ still sees significant performance increases despite the feature similarity, which demonstrates that the supplemental representations do provide useful additional information.

4.6 System combination

Although the reranking approach was developed for the purpose of leveraging supplemental data, it can also increase the accuracy when no genuine supplemental data is available. The idea is to perform system combination by treating the output of one of the systems as the supplemental data for the other system, effectively casting the system combination problem into our reranking framework. In our last experiment, we test the combination of DIRECTL+ and SEQUITUR for English-to-Japanese MTL by designating either of them as the base system. Since the supplemental data are generated, we are not limited to a particular subset, and can conduct the experiment on the entire English-to-Japanese set, with the test set having 2,801 entries. For comparison, we also test a linear combination of the (normalized) system scores with a manually tuned weight parameter (LINCOMB). This baseline is similar to the system combination method of Finch and Sumita (2010).

	SEQUITUR		DIRECTL+	
	Acc.	ERR	Acc.	ERR
BASE	45.5		47.3	
LINCOMB	49.4	7.2	49.4	4.0
RERANKED	50.2	8.7	49.2	3.7
ORACLE	82.4	67.7	77.3	56.9

Table 5: Word accuracies and error rate reductions (ERR) in percentages for English-to-Japanese MTL augmented by predicted transliterations from the other base system.

Table 5 contains the results for English-to-Japanese transliterations, which indicate a significant increase in accuracy in both cases, thereby demonstrating the viability of our approach for system combination. This experiment extends the system combination result on English-to-Hindi transliteration reported by Bhargava et al. (2011), in which DIRECTL+ served as the base system while SEQUITUR provided the supplemental data. The system in question yielded nearly a 4% error rate reduction, which made it the top-ranking submission at the NEWS 2011 Shared Task on Transliteration.

On the other hand, LINCOMB turns out to be a strong baseline, which is evidenced by the fact that the differences between our reranking approach and LINCOMB are statistically insignificant. This is likely because LINCOMB can take advantage of the full n-best lists provided by both systems, whereas the reranking approach uses only the top-1 result from the "supplemental" system. Combining the two n-best lists in this way also gives a higher oracle score of 86.4%, suggesting that this may be a good and computationally cheap first step prior to reranking using proper supplemental data as described above.

5 Future work

We plan on investigating a more parsimonious method of incorporating supplemental data. There are two aspects to this. First, while our experiments in this paper treated base systems as black boxes for the purposes of examining the effect of the supplemental data in isolation, reranking is limited by its *post hoc* nature. After all, if the correct output does not appear in the base systems' n-best list, even a perfect reranker would be unable to find it. Incorporating the supplemental data earlier in the process would allow us to overcome this limitation at the expense of being a solution specific to the base system.

Second, we would like to be able to incorporate general supplemental *information* rather than being limited by the existence of relevant *data*. In particular, a good transliteration model should encode a general version of the information provided by a single transliteration, so being able to apply that information would allow us to overcome our dependence on existing data as well as provide more potentially useful information even when a transliteration or transcription already exists.

Finally, we plan on examining other potential supplemental resources. Given the success of our approach in the face of sometimes-noisy transliteration data[7], other noisy data may be applicable as well. For example, IPA transcriptions could be mined from Wikipedia despite the fact that different transcriptions may have been written by different people. Similarly, difficult-to-pronounce names or words are often accompanied by *ad hoc* approximately-phonetic re-spellings, which may also prove useful.

6 Conclusion

In this paper, we examined the relevance of alternative, supplemental representations for the tasks of grapheme-to-phoneme conversion and machine transliteration, both of which have pronunciation as an important underlying influence. We applied an SVM reranking approach that leverages the supplemental data using features constructed from n-grams as well as from similarity and system scores. The approach yielded excellent improvements when used with both the SEQUITUR and DIRECTL+ base systems. Over the state-of-the-art DIRECTL+, we achieved significant error rate reductions of **13%** for English-to-Japanese MTL using supplemental transliterations, **13%** using supplemental transcriptions, and **33%** for English G2P using supplemental transcriptions. For system combination, we found a smaller but still significant error rate reduction of **4%**. The fact that the improvements vary systematically by base system help confirm that the supplemental data do provide inherently useful information.

We can also take a step back to take a broader look at our approach. It applies similar features as those used in the standard generation task in a new, orthogonal direction (supplemental data) with successful results. This notion is general enough that it may potentially be applicable to other tasks, such as part-of-speech tagging or machine translation.

Acknowledgements

We thank Sittichai Jiampojamarn and Shane Bergsma for helpful discussions. This research was supported by the Natural Sciences and Engineering Research Council of Canada.

References

Yaser Al-Onaizan and Kevin Knight. 2002. Machine transliteration of names in Arabic texts. In *Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages*, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

R. Harald Baayen, Richard Piepenbrock, and Leon Gulikers. 1996. The CELEX2 lexical database. LDC96L14.

Aditya Bhargava and Grzegorz Kondrak. 2011. How do you pronounce your name? Improving G2P with transliterations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 399–408, Portland, Oregon, USA, June. Association for Computational Linguistics.

[7] Jiampojamarn et al. (2009) found a significant increase in English-to-Hindi transliteration performance after applying a simple rule-based cleaning script.

Aditya Bhargava, Bradley Hauer, and Grzegorz Kondrak. 2011. Leveraging transliterations from multiple languages. In *Proceedings of the 3rd Named Entities Workshop (NEWS 2011)*, pages 36–40, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

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

Alan W. Black, Kevin Lenzo, and Vincent Pagel. 1998. Issues in building general letter to sound rules. In *The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*, Jenolan Caves House, Blue Mountains, New South Wales, Australia, November.

Trevor Cohn and Mirella Lapata. 2007. Machine translation by triangulation: Making effective use of multi-parallel corpora. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 728–735, Prague, Czech Republic, June. Association for Computational Linguistics.

Andrew Finch and Eiichiro Sumita. 2010. Transliteration using a phrase-based statistical machine translation system to re-score the output of a joint multigram model. In *Proceedings of the 2010 Named Entities Workshop*, pages 48–52, Uppsala, Sweden, July. 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, USA, 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, USA, June. Association for Computational Linguistics.

Sittichai Jiampojamarn, Aditya Bhargava, Qing Dou, Kenneth Dwyer, and Grzegorz Kondrak. 2009. DirecTL: a language independent approach to transliteration. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 28–31, Suntec, Singapore, August. Association for Computational Linguistics.

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2010a. Integrating joint n-gram features into a discriminative training framework. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 697–700, Los Angeles, California, USA, June. Association for Computational Linguistics.

Sittichai Jiampojamarn, Kenneth Dwyer, Shane Bergsma, Aditya Bhargava, Qing Dou, Mi-Young Kim, and Grzegorz Kondrak. 2010b. Transliteration generation and mining with limited training resources. In *Proceedings of the 2010 Named Entities Workshop*, pages 39–47, Uppsala, Sweden, July. Association for Computational Linguistics.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142, Edmonton, Alberta, Canada. Association for Computing Machinery.

Mitesh M. Khapra, A Kumaran, and Pushpak Bhattacharyya. 2010. Everybody loves a rich cousin: An empirical study of transliteration through bridge languages. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 420–428, Los Angeles, California, June. Association for Computational Linguistics.

Anne K. Kienappel and Reinhard Kneser. 2001. Designing very compact decision trees for grapheme-to-phoneme transcription. In *EUROSPEECH-2001*, pages 1911–1914, Aalborg, Denmark, September.

Alexandre Klementiev and Dan Roth. 2006. Named entity transliteration and discovery from multilingual comparable corpora. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 82–88, New York City, USA, June. Association for Computational Linguistics.

Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24(4):599–612, December.

John Kominek and Alan W. Black. 2006. Learning pronunciation dictionaries: Language complexity and word selection strategies. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 232–239, New York City, New York, USA, June. Association for Computational Linguistics.

A. Kumaran, Mitesh M. Khapra, and Pushpak Bhattacharyya. 2010a. Compositional machine transliteration. 9(4):13:1–13:29, December.

A Kumaran, Mitesh M. Khapra, and Haizhou Li. 2010b. Report of news 2010 transliteration mining shared task. In *Proceedings of the 2010 Named Entities Workshop*, pages 21–28, Uppsala, Sweden, July. Association for Computational Linguistics.

Haizhou Li, Min Zhang, and Jian Su. 2004. A joint source-channel model for machine transliteration. In *Proceedings of the 42nd Meeting of the Association* *for Computational Linguistics (ACL'04), Main Volume*, pages 159–166, Barcelona, Spain, July.

Haizhou Li, A Kumaran, Vladimir Pervouchine, and Min Zhang. 2009. Report of NEWS 2009 machine transliteration shared task. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 1–18, Suntec, Singapore, August. Association for Computational Linguistics.

Haizhou Li, A Kumaran, Min Zhang, and Vladimir Pervouchine. 2010. Report of NEWS 2010 transliteration generation shared task. In *Proceedings of the 2010 Named Entities Workshop*, pages 1–11, Uppsala, Sweden, July. Association for Computational Linguistics.

Linsen Loots and Thomas R. Niesler. 2009. Data-driven phonetic comparison and conversion between south african, british and american english pronunciations. In *Proceedings of Interspeech*, Brighton, UK, September.

Yannick Marchand and Robert I. Damper. 2000. A multistrategy approach to improving pronunciation by analogy. *Computational Linguistics*, 26(2):195–219, June.

Korin Richmond, Robert Clark, and Sue Fitt. 2009. Robust LTS rules with the Combilex speech technology lexicon. In *Proceedings of Interspeech*, pages 1295–1298, Brighton, UK, September.

Eric Sven Ristad and Peter N. Yianilos. 1998. Learning string edit distance. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 20(5):522–532, May.

Terrence J. Sejnowski and Charles R. Rosenberg. 1987. Parallel networks that learn to pronounce english text. *Complex Systems*, 1(1):145–168.

Benjamin Snyder, Tahira Naseem, Jacob Eisenstein, and Regina Barzilay. 2009. Adding more languages improves unsupervised multilingual part-of-speech tagging: a bayesian non-parametric approach. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 83–91, Boulder, Colorado, USA, June. Association for Computational Linguistics.

Richard Sproat, Tao Tao, and ChengXiang Zhai. 2006. Named entity transliteration with comparable corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 73–80, Sydney, Australia, July. Association for Computational Linguistics.

Paul Taylor. 2005. Hidden Markov models for grapheme to phoneme conversion. In *Proceedings of Interspeech*, pages 1973–1976, Lisbon, Portugal, September.

Masao Utiyama and Hitoshi Isahara. 2007. A comparison of pivot methods for phrase-based statistical machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 484–491, Rochester, New York, USA, April. Association for Computational Linguistics.

van den Bosch and Walter Daelemans. 1998. Do not forget: Full memory in memory-based learning of word pronunciation. In D.M.W. Powers, editor, *NeMLaP3/CoNLL98: New Methods in Language Processing and Computational Natural Language Learning*, pages 195–204, Sydney, Australia. Association for Computational Linguistics.

Hua Wu and Haifeng Wang. 2009. Revisiting pivot language approach for machine translation. 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*, pages 154–162, Suntec, Singapore, August. Association for Computational Linguistics.

Dmitry Zelenko and Chinatsu Aone. 2006. Discriminative methods for transliteration. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 612–617, Sydney, Australia, July. Association for Computational Linguistics.

Min Zhang, Xiangyu Duan, Vladimir Pervouchine, and Haizhou Li. 2010. Machine transliteration: Leveraging on third languages. In *Coling 2010: Posters*, pages 1444–1452, Beijing, China, August. Coling 2010 Organizing Committee.

Min Zhang, Haizhou Li, A Kumaran, and Ming Liu. 2011. Report of NEWS 2011 machine transliteration shared task. In *Proceedings of the 3rd Named Entities Workshop (NEWS 2011)*, pages 1–13, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.