How do you pronounce your name? Improving G2P with transliterations

Aditya Bhargava and Grzegorz Kondrak
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada, T6G 2E8
{abhargava,kondrak}@cs.ualberta.ca

Abstract

Grapheme-to-phoneme conversion (G2P) of names is an important and challenging problem. The correct pronunciation of a name is often reflected in its transliterations, which are expressed within a different phonological inventory. We investigate the problem of using transliterations to correct errors produced by state-of-the-art G2P systems. We present a novel re-ranking approach that incorporates a variety of score and n-gram features, in order to leverage transliterations from multiple languages. Our experiments demonstrate significant accuracy improvements when re-ranking is applied to n-best lists generated by three different G2P programs.

1 Introduction

Grapheme-to-phoneme conversion (G2P), in which the aim is to convert the orthography of a word to its pronunciation (phonetic transcription), plays an important role in speech synthesis and understanding. Names, which comprise over 75% of unseen words (Black et al., 1998), present a particular challenge to G2P systems because of their high pronunciation variability. Guessing the correct pronunciation of a name is often difficult, especially if they are of foreign origin; this is attested by the *ad hoc* transcriptions which sometimes accompany new names introduced in news articles, especially for international stories with many foreign names.

Transliterations provide a way of disambiguating the pronunciation of names. They are more abundant than phonetic transcriptions, for example when news items of international or global significance are reported in multiple languages. In addition, writing scripts such as Arabic, Korean, or Hindi are more consistent and easier to identify than various phonetic transcription schemes. The process of transliteration, also called *phonetic translation* (Li et al., 2009b), involves "sounding out" a name and then finding the closest possible representation of the sounds in another writing script. Thus, the correct pronunciation of a name is partially encoded in the form of the transliteration. For example, given the ambiguous letter-to-phoneme mapping of the English letter *g*, the initial phoneme of the name *Gershwin* may be predicted by a G2P system to be either /ɡ/ (as in *Gertrude*) or /ʤ/ (as in *Gerald*). The transliterations of the name in other scripts provide support for the former (correct) alternative.

Although it seems evident that transliterations should be helpful in determining the correct pronunciation of a name, designing a system that takes advantage of this insight is not trivial. The main source of the difficulty stems from the differences between the phonologies of distinct languages. The mappings between phonemic inventories are often complex and context-dependent. For example, because Hindi has no /w/ sound, the transliteration of *Gershwin* instead uses a symbol that represents the phoneme /ʋ/, similar to the /v/ phoneme in English. In addition, converting transliterations into phonemes is often non-trivial; although few orthographies are as inconsistent as that of English, this is effectively the G2P task for the particular language in question.

In this paper, we demonstrate that leveraging transliterations can, in fact, improve the grapheme-to-phoneme conversion of names. We propose a novel system based on discriminative re-ranking that is capable of incorporating multiple transliterations. We show that simplistic approaches to the problem

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 399–408,
Portland, Oregon, June 19-24, 2011. ©2011 Association for Computational Linguistics

fail to achieve the same goal, and that transliterations from multiple languages are more helpful than from a single language. Our approach can be combined with any G2P system that produces n-best lists instead of single outputs. The experiments that we perform demonstrate significant error reduction for three very different G2P base systems.

2 Improving G2P with transliterations

2.1 Problem definition

In both G2P and machine transliteration, we are interested in learning a function that, given an input sequence x, produces an output sequence y. In the G2P task, x is composed of graphemes and y is composed of phonemes; in transliteration, both sequences consist of graphemes but they represent different writing scripts. Unlike in machine translation, the monotonicity constraint is enforced; i.e., we assume that x and y can be aligned without the alignment links crossing each other (Jiampojamarn and Kondrak, 2010). We assume that we have available a base G2P system that produces an n-best list of outputs with a corresponding list of confidence scores. The goal is to improve the base system's performance by applying existing transliterations of the input x to re-rank the system's n-best output list.

2.2 Similarity-based methods

A simple and intuitive approach to improving G2P with transliterations is to select from the n-best list the output sequence that is most similar to the corresponding transliteration. For example, the Hindi transliteration in Figure 1 is arguably closest in perceptual terms to the phonetic transcription of the second output in the n-best list, as compared to the other outputs. One obvious problem with this method is that it ignores the relative ordering of the n-best lists and their corresponding scores produced by the base system.

A better approach is to combine the similarity score with the output score from the base system, allowing it to contribute an estimate of confidence in its output. For this purpose, we apply a linear combination of the two scores, where a single parameter λ, ranging between zero and one, determines the relative weight of the scores. The exact value of λ can be optimized on a training set. This approach is similar to the method used by Finch and Sumita (2010) to combine the scores of two different machine transliteration systems.

2.3 Measuring similarity

The approaches presented in the previous section crucially depend on a method for computing the similarity between various symbol sequences that represent the same word. If we have a method of converting transliterations to phonetic representations, the similarity between two sequences of phonemes can be computed with a simple method such as normalized edit distance or the longest common subsequence ratio, which take into account the number and position of identical phonemes. Alternatively, we could apply a more complex approach, such as ALINE (Kondrak, 2000), which computes the distance between pairs of phonemes. However, the implementation of a conversion program would require ample training data or language-specific expertise.

A more general approach is to skip the transcription step and compute the similarity between phonemes and graphemes directly. For example, the edit distance function can be learned from a training set of transliterations and their phonetic transcriptions (Ristad and Yianilos, 1998). In this paper, we apply M2M-ALIGNER (Jiampojamarn et al., 2007), an unsupervised aligner, which is a many-to-many generalization of the learned edit distance algorithm. M2M-ALIGNER was originally designed to align graphemes and phonemes, but can be applied to discover the alignment between any sets of symbols (given training data). The logarithm of the probability assigned to the optimal alignment can then be interpreted as a similarity measure between the two sequences.

2.4 Discriminative re-ranking

The methods described in Section 2.2, which are based on the similarity between outputs and transliterations, are difficult to generalize when multiple transliterations of a single name are available. A linear combination is still possible but in this case optimizing the parameters would no longer be straightforward. Also, we are interested in utilizing other features besides sequence similarity.

The SVM re-ranking paradigm offers a solution

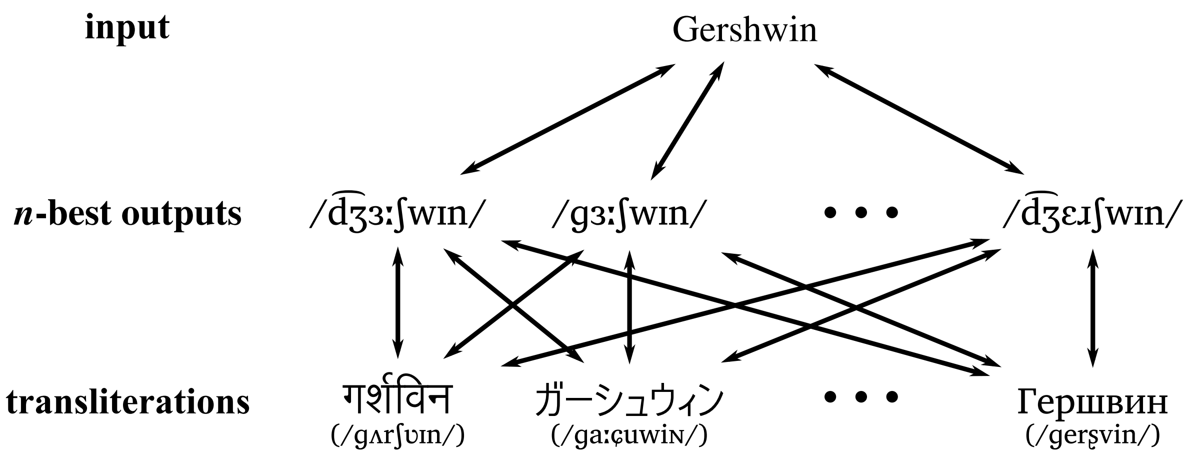

Figure 1: An example name showing the data used for feature construction. Each arrow links a pair used to generate features, including n-gram and score features. The score features use similarity scores for transliteration-transcription pairs and system output scores for input-output pairs. One feature vector is constructed for each system output.

to the problem. Our re-ranking system is informed by a large number of features, which are based on scores and n-grams. The scores are of three types:

1. The scores produced by the base system for each output in the n-best list.
2. The similarity scores between the outputs and each available transliteration.
3. The differences between scores in the n-best lists for both (1) and (2).

Our set of binary n-gram features includes those used for DIRECTL+ (Jiampojamarn et al., 2010). They can be divided into four types:

1. The context features combine output symbols (phonemes) with n-grams of varying sizes in a window of size c centred around a corresponding position on the input side.
2. The transition features are bigrams on the output (phoneme) side.
3. The linear chain features combine the context features with the bigram transition features.
4. The joint n-gram features are n-grams containing both input and output symbols.

We apply the features in a new way: instead of being applied strictly to a given input-output set, we expand their use across many languages and use all of them simultaneously. We apply the n-gram features across all transliteration-transcription pairs in addition to the usual input-output pairs corresponding to the n-best lists. Figure 1 illustrates the set of pairs used for feature generation.

In this paper, we augment the n-gram features by a set of *reverse* features. Unlike a traditional G2P generator, our re-ranker has access to the outputs produced by the base system. By swapping the input and the output side, we can add reverse context and linear-chain features. Since the n-gram features are also applied to transliteration-transcription pairs, the reverse features enable us to include features which bind a variety of n-grams in the transliteration string with a single corresponding phoneme.

The construction of n-gram features presupposes a fixed alignment between the input and output sequences. If the base G2P system does not provide input-output alignments, we use M2M-ALIGNER for this purpose. The transliteration-transcription pairs are also aligned by M2M-ALIGNER, which at the same time produces the corresponding similarity scores. (We set a lower limit of -100 on the M2M-ALIGNER scores.) If M2M-ALIGNER is unable to produce an alignment, we indicate this with a binary feature that is included with the n-gram features.

3 Experiments

We perform several experiments to evaluate our transliteration-informed approaches. We test simple similarity-based approaches on single-transliteration data, and evaluate our SVM re-ranking approach against this as well. We then test our approach using all available transliterations. Relevant code and scripts required to reproduce our experimental results are available online[1].

3.1 Data & setup

For pronunciation data, we extracted all names from the Combilex corpus (Richmond et al., 2009). We discarded all diacritics, duplicates and multi-word names, which yielded 10,084 unique names. Both the similarity and SVM methods require transliterations for identifying the best candidates in the n-best lists. They are therefore trained and evaluated on the subset of the G2P corpus for which transliterations available. Naturally, allowing transliterations from all languages results in a larger corpus than the one obtained by the intersection with transliterations from a single language.

For our experiments, we split the data into 10% for testing, 10% for development, and 80% for training. The development set was used for initial tests and experiments, and then for our final results the training and development sets were combined into one set for final system training. For SVM re-ranking, during both development and testing we split the training set into 10 folds; this is necessary when training the re-ranker as it must have system output scores that are representative of the scores on unseen data. We ensured that there was never any overlap between the training and testing data for all trained systems.

Our transliteration data come from the shared tasks on transliteration at the 2009 and 2010 Named Entities Workshops (Li et al., 2009a; Li et al., 2010). We use all of the 2010 English-source data plus the English-to-Russian data from 2009, which makes nine languages in total. In cases where the data provide alternative transliterations for a given input, we keep only one; our preliminary experiments indicated that including alternative transliterations did not improve performance. It should be noted that these transliteration corpora are noisy: Jiampojamarn et al. (2009) note a significant increase in English-to-Hindi transliteration performance with a simple cleaning of the data.

Language	Corpus size	Overlap
Bengali	12,785	1,840
Chinese	37,753	4,713
Hindi	12,383	2,179
Japanese	26,206	4,773
Kannada	10,543	1,918
Korean	6,761	3,015
Russian	6,447	487
Tamil	10,646	1,922
Thai	27,023	5,436

Table 1: The number of unique single-word entries in the transliteration corpora for each language and the amount of common data (overlap) with the pronunciation data.

Our tests involving transliterations from multiple languages are performed on the set of names for which we have both the pronunciation and transliteration data. There are 7,423 names in the G2P corpus for which at least one transliteration is available. Table 1 lists the total size of the transliteration corpora as well as the amount of overlap with the G2P data. Note that the base G2P systems are trained using all 10,084 names in the corpus as opposed to only the 7,423 names for which there are transliterations available. This ensures that the G2P systems have more training data to provide the best possible base performance.

For our single-language experiments, we normalize the various scores when tuning the linear combination parameter λ so that we can compare values across different experimental conditions. For SVM re-ranking, we directly implement the method of Joachims (2002) to convert the re-ranking problem into a classification problem, and then use the very fast LIBLINEAR (Fan et al., 2008) to build the SVM models. Optimal hyperparameter values were determined during development.

We evaluate using word accuracy, the percentage of words for which the pronunciations are correctly predicted. This measure marks pronunciations that are even slightly different from the correct one as incorrect, so even a small change in pronunciation that might be acceptable or even unnoticeable to humans would count against the system's performance.

[1] http://www.cs.ualberta.ca/~ab31/g2p-tl-rr

3.2 Base systems

It is important to test multiple base systems in order to ensure that any gain in performance applies to the task in general and not just to a particular system. We use three G2P systems in our tests:

1. FESTIVAL (FEST), a popular speech synthesis package, which implements G2P conversion with CARTs (decision trees) (Black et al., 1998).

2. SEQUITUR (SEQ), a generative system based on the joint n-gram approach (Bisani and Ney, 2008).

3. DIRECTL+ (DTL), the discriminative system on which our n-gram features are based (Jiampojamarn et al., 2010).

All systems are capable of providing n-best output lists along with scores for each output, although for FESTIVAL they had to be constructed from the list of output probabilities for each input character.

We run DIRECTL+ with all of the features described in (Jiampojamarn et al., 2010) (i.e., context features, transition features, linear chain features, and joint n-gram features). System parameters, such as maximum number of iterations, were determined during development. For SEQUITUR, we keep default options except for the enabling of the 10 best outputs and we convert the probabilities assigned to the outputs to log-probabilities. We set SEQUITUR's joint n-gram order to 6 (this was also determined during development).

Note that the three base systems differ slightly in terms of the alignment information that they provide in their outputs. FESTIVAL operates letter-by-letter, so we use the single-letter inputs with the phoneme outputs as the aligned units. DIRECTL+ specifies many-to-many alignments in its output. For SEQUITUR, however, since it provides no information regarding the output structure, we use M2M-ALIGNER to induce alignments for n-gram feature generation.

3.3 Transliterations from a single language

The goal of the first experiment is to compare several similarity-based methods, and to determine how they compare to our re-ranking approach. In order to find the similarity between phonetic transcriptions, we use the two different methods described in Section 2.2: ALINE and M2M-ALIGNER. We further test the use of a linear combination of the similarity scores with the base system's score so that its confidence information can be taken into account; the linear combination weight is determined from the training set. These methods are referred to as ALINE+BASE and M2M+BASE. For these experiments, our training and testing sets are obtained by intersecting our G2P training and testing sets respectively with the Hindi transliteration corpus, yielding 1,950 names for training and 229 names for testing.

Since the similarity-based methods are designed to incorporate homogeneous same-script transliterations, we can only run this experiment on one language at a time. Furthermore, ALINE operates on phoneme sequences, so we first need to convert the transliterations to phonemes. An alternative would be to train a proper G2P system, but this would require a large set of word-pronunciation pairs. For this experiment, we choose Hindi, for which we constructed a rule-based G2P converter. Aside from simple one-to-one mapping (romanization) rules, the converter has about ten rules to adjust for context.

For these experiments, we apply our SVM re-ranking method in two ways:

1. Using only Hindi transliterations (referred to as SVM-HINDI).

2. Using all available languages (referred to as SVM-ALL).

In both cases, the test set is restricted to the same 229 names, in order to provide a valid comparison.

Table 2 presents the results. Regardless of the choice of the similarity function, the simplest approaches fail in a spectacular manner, significantly reducing the accuracy with respect to the base system. The linear combination methods give mixed results, improving the accuracy for FESTIVAL but not for SEQUITUR or DIRECTL+ (although the differences are not statistically significant). However, they perform much better than the methods based on similarity scores alone as they are able to take advantage of the base system's output scores. If we look at the values of λ that provide the best performance

	Base system		
	FEST	SEQ	DTL
Base	58.1	67.3	71.6
ALINE	28.0	26.6	27.5
M2M	39.3	36.2	36.2
ALINE+BASE	58.5	65.9	71.2
M2M+BASE	58.5	66.4	70.3
SVM-HINDI	63.3	69.0	69.9
SVM-ALL	68.6	72.5	75.6

Table 2: Word accuracy (in percentages) of various methods when only Hindi transliterations are used.

	Base system		
	FEST	SEQ	DTL
Base	55.3	66.5	70.8
SVM-SCORE	62.1	68.4	71.0
SVM-N-GRAM	66.2	72.5	73.8
SVM-ALL	67.2	73.4	74.3

Table 3: Word accuracy of the base system versus the re-ranking variants with transliterations from multiple languages.

on the training set, we find that they are higher for the stronger base systems, indicating more reliance on the base system output scores. For example, for ALINE+BASE the FESTIVAL-based system has $\lambda = 0.58$ whereas the DIRECTL+-based system has $\lambda = 0.81$. Counter-intuitively, the ALINE+BASE and M2M+BASE methods are unable to improve upon SEQUITUR or DIRECTL+. We would expect to achieve at least the base system's performance, but disparities between the training and testing sets prevent this.

The two SVM-based methods achieve much better results. SVM-ALL produces impressive accuracy gains for all three base systems, while SVM-HINDI yields smaller (but still statistically significant) improvements for FESTIVAL and SEQUITUR. These results suggest that our re-ranking method provides a bigger boost to systems built with different design principles than to DIRECTL+ which utilizes a similar set of features. On the other hand, the results also show that the information obtained by consulting a single transliteration may be insufficient to improve an already high-performing G2P converter.

3.4 Transliterations from multiple languages

Our second experiment expands upon the first; we use all available transliterations instead of being restricted to one language. This rules out the simple similarity-based approaches, but allows us to test our re-ranking approach in a way that fully utilizes the available data. We test three variants of our transliteration-informed SVM re-ranking approach, which differ with respect to the set of included features:

1. SVM-SCORE includes only the three types of score features described in Section 2.4.
2. SVM-N-GRAM uses only the n-gram features.
3. SVM-ALL is the full system that combines the score and n-gram features.

The objective is to determine the degree to which each of the feature classes contributes to the overall results. Because we are using all available transliterations, we achieve much greater coverage over our G2P data than in the previous experiment; in this case, our training set consists of 6,660 names while the test set has 763 names.

Table 3 presents the results. Note that the baseline accuracies are somewhat lower than in Table 2 because of the different test set. We find that, when using all features, the SVM re-ranker can provide a very impressive error reduction over FESTIVAL (26.7%) and SEQUITUR (20.7%) and a smaller but still significant ($p < 0.01$ with the McNemar test) error reduction over DIRECTL+ (12.1%).

When we consider our results using only the score and n-gram features, we can see that, interestingly, the n-gram features are most important. We draw a further conclusion from our results: consider the large disparity in improvements over the base systems. This indicates that FESTIVAL and SEQUITUR are benefiting from the DIRECTL+-style features used in the re-ranking. Without the n-gram features, however, there is still a significant improvement over FESTIVAL, demonstrating that the scores *do* provide useful information. In this case there is no way for DIRECTL+-style information to make its way into the re-ranking; the process is based purely on the transliterations and their similarities with the transcriptions in the output lists, indicating that the system is capable of extracting useful information directly from transliterations. In the case of DIRECTL+, the transliterations help through the n-gram features rather than the score features; this is probably because the crucial feature that signals the inability of M2M-ALIGNER to align a given transliteration-transcription pair belongs to the set of the n-gram features. Both the n-gram features and score features are dependent on the alignments, but they differ in that the n-gram features allow weights to be learned for local n-gram pairs whereas the score features are based on global information, providing only a single feature for a given transliteration-transcription pair. The two therefore overlap to some degree, although the score features still provide useful information via probabilities learned during the alignment training process.

A closer look at the results provides additional insight into the operation of our re-ranking system. For example, consider the name *Bacchus*, which DIRECTL+ incorrectly converts into /bækʧəs/. The most likely reason why our re-ranker selects instead the correct pronunciation /bækəs/ is that M2M-ALIGNER fails to align three of the five available transliterations with /bækʧəs/. Such alignment failures are caused by a lack of evidence for the mapping of the grapheme representing the sound /k/ in the transliteration training data with the phoneme /ʧ/. In addition, the lack of alignments prevents any n-gram features from being enabled.

Considering the difficulty of the task, the top accuracy of almost 75% is quite impressive. In fact, many instances of human transliterations in our corpora are clearly incorrect. For example, the Hindi transliteration of *Bacchus* contains the /ʧ/ consonant instead of the correct /k/. Moreover, our strict evaluation based on word accuracy counts all system outputs that fail to exactly match the dictionary data as errors. The differences are often very minor and may reflect an alternative pronunciation. The *phoneme* accuracy[2] of our best result is 93.1%, which provides some idea of how similar the predicted pronunciation is to the correct one.

# TL	# Entries	Improvement
≤ 1	111	0.9
≤ 2	266	3.0
≤ 3	398	3.8
≤ 4	536	3.2
≤ 5	619	2.8
≤ 6	685	3.4
≤ 7	732	3.7
≤ 8	762	3.5
≤ 9	763	3.5

Table 4: Absolute improvement in word accuracy (%) over the base system (DIRECTL+) of the SVM re-ranker for various numbers of available transliterations.

3.5 Effect of multiple transliterations

One motivating factor for the use of SVM re-ranking was the ability to incorporate multiple transliteration languages. But how important is it to use more than one language? To examine this question, we look particularly at the sets of names having at most k transliterations available. Table 4 shows the results with DIRECTL+ as the base system. Note that the number of names with more than five transliterations was small. Importantly, we see that the increase in performance when only one transliteration is available is so small as to be insignificant. From this, we can conclude that obtaining improvement on the basis of a single transliteration is difficult in general. This corroborates the results of the experiment described in Section 3.3, where we used only Hindi transliterations.

4 Previous work

There are three lines of research that are relevant to our work: (1) G2P in general; (2) G2P on names; and (3) combining diverse data sources and/or systems.

The two leading approaches to G2P are represented by SEQUITUR (Bisani and Ney, 2008) and DIRECTL+ (Jiampojamarn et al., 2010). Recent comparisons suggests that the former obtains somewhat higher accuracy, especially when it includes joint n-gram features (Jiampojamarn et al., 2010). Systems based on decision trees are far behind. Our

[2] The phoneme accuracy is calculated from the minimum edit distance between the predicted and correct pronunciations.

results confirm this ranking.

Names can present a particular challenge to G2P systems. Kienappel and Kneser (2001) reported a higher error rate for German names than for general words, while on the other hand Black et al. (1998) report similar accuracy on names as for other types of English words. Yang et al. (2006) and van den Heuvel et al. (2007) post-process the output of a general G2P system with name-specific phoneme-to-phoneme (P2P) systems. They find significant improvement using this method on data sets consisting of Dutch first names, family names, and geographical names. However, it is unclear whether such an approach would be able to improve the performance of the current state-of-the-art G2P systems. In addition, the P2P approach works only on single outputs, whereas our re-ranking approach is designed to handle n-best output lists.

Although our approach is (to the best of our knowledge) the first to use different *tasks* (G2P and transliteration) to inform each other, this is conceptually similar to model and system combination approaches. In statistical machine translation (SMT), methods that incorporate translations from other languages (Cohn and Lapata, 2007) have proven effective in low-resource situations: when phrase translations are unavailable for a certain language, one can look at other languages where the translation *is* available and then translate from that language. A similar pivoting approach has also been applied to machine transliteration (Zhang et al., 2010). Notably, the focus of these works have been on cases in which there are less data available; they also modify the generation process directly, rather than operating on existing outputs as we do. Ultimately, a combination of the two approaches is likely to give the best results.

Finch and Sumita (2010) combine two very different approaches to transliteration using simple linear interpolation: they use SEQUITUR's n-best outputs and re-rank them using a linear combination of the original SEQUITUR score and the score for that output of a phrased-based SMT system. The linear weights are hand-tuned. We similarly use linear combinations, but with many more scores and other features, necessitating the use of SVMs to determine the weights. Importantly, we combine different *data types* where they combine different *systems*.

5 Conclusions & future work

In this paper, we explored the application of transliterations to G2P. We demonstrated that transliterations have the potential for helping choose between n-best output lists provided by standard G2P systems. Simple approaches based solely on similarity do not work when tested using a single transliteration language (Hindi), necessitating the use of smarter methods that can incorporate multiple transliteration languages. We apply SVM re-ranking to this task, enabling us to use a variety of features based not only on similarity scores but on n-grams as well. Our method shows impressive error reductions over the popular FESTIVAL system and the generative joint n-gram SEQUITUR system. We also find significant error reduction using the state-of-the-art DIRECTL+ system. Our analysis demonstrated that it is essential to provide the re-ranking system with transliterations from multiple languages in order to mitigate the differences between phonological inventories and smooth out noise in the transliterations.

In the future, we plan to generalize our approach so that it can be applied to the task of generating transliterations, and to combine data from distinct G2P dictionaries. The latter task is related to the notion of domain adaptation. We would also like to apply our approach to web data; we have shown that it is possible to use noisy transliteration data, so it may be possible to leverage the noisy *ad hoc* pronunciation data as well. Finally, we plan to investigate earlier integration of such external information into the G2P process for single systems; while we noted that re-ranking provides a general approach applicable to any system that can generate n-best lists, there is a limit as to what re-ranking can do, as it relies on the correct output existing in the n-best list. Modifying existing systems would provide greater potential for improving results even though the changes would be necessarily system-specific.

Acknowledgements

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

References

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.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

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 (NEWS 2010)*, pages 48–52, Uppsala, Sweden, July. Association for Computational Linguistics.

Sittichai Jiampojamarn and Grzegorz Kondrak. 2010. Letter-phoneme alignment: An exploration. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 780–788, 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, 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. 2010. 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.

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.

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.

Grzegorz Kondrak. 2000. A new algorithm for the alignment of phonetic sequences. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 288–295, Seattle, Washington, USA, April.

Haizhou Li, A Kumaran, Vladimir Pervouchine, and Min Zhang. 2009a. 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. 2009b. Whitepaper of NEWS 2009 machine transliteration shared task. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 19–26, 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 (NEWS 2010)*, pages 1–11, Uppsala, Sweden, July. Association for Computational Linguistics.

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.

Henk van den Heuvel, Jean-Pierre Martens, and Nanneke Konings. 2007. G2P conversion of names. what can we do (better)? In *Proceedings of Interspeech*, pages 1773–1776, Antwerp, Belgium, August.

Qian Yang, Jean-Pierre Martens, Nanneke Konings, and Henk van den Heuvel. 2006. Development of a phoneme-to-phoneme (p2p) converter to improve the grapheme-to-phoneme (g2p) conversion of names. In

Proceedings of the 2006 International Conference on Language Resources and Evaluation, pages 2570–2573, Genoa, Italy, May.

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.