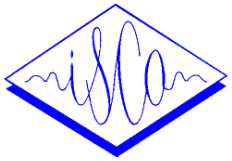

7th International Conference on Spoken Language Processing [ICSLP2002]
Denver, Colorado, USA
September 16-20, 2002

PRONUNCIATION OF PROPER NAMES WITH A JOINT N-GRAM MODEL FOR BI-DIRECTIONAL GRAPHEME-TO-PHONEME CONVERSION

Lucian Galescu, James F. Allen

Department of Computer Science
University of Rochester
{galescu,james}@cs.rochester.edu

ABSTRACT

Pronunciation of proper names is known to be a difficult problem, but one of great practical importance for both speech synthesis and speech recognition. Recently a few data-driven grapheme-to-phoneme conversion techniques have been proposed to tackle this problem. In this paper we apply the joint n-gram model for bi-directional grapheme-to-phoneme conversion, which has already been shown to achieve excellent results on general tasks, to the more specific task of converting between name pronunciations and spellings.

The performance of our technique on generating name pronunciations exceeds that of other techniques even when they use additional information. We find the reverse task, of generating orthographic transcriptions from phonemic input, to be a much more difficult task for names than for common words. However, we derive valuable information from our results about the potential of sub-lexical recognition of novel proper names.

1. INTRODUCTION

Proper names make a large proportion of the new words that text-to-speech and speech recognition systems may encounter that are outside the pronunciation dictionaries used in such systems. This is a serious problem for name-intensive applications like voice-mail, directory assistance, but also for applications for reading or transcribing news. For example, on part of the WSJ corpus [11], it was found that names comprised more than 76% of the words not covered by the otherwise large OALD dictionary [3].

The Donnelly corpus, a collection of over 1.5 million distinct surnames covering 72 million households in the US, reveals that the surname distribution follows roughly Zipf's law; that is, a few names occur in a large number of households, whereas a large number of the surnames are extremely rare (about 650,000 of them occur in just one household each) [10]. Thus, having a pronunciation dictionary to cover all the proper names is a very remote possibility. This makes it imperative to develop automated techniques for name pronunciation.

For US English, the name pronunciation problem is particularly challenging compared to automated pronunciation of common words because surnames come from a variety of ethnic backgrounds, and thus they don't follow the usual grapho-phonotactic constraints of the language. Rather, many foreign proper names retain very peculiar orthography while they often acquire an anglicized pronunciation, which may differ dramatically from the native pronunciation.

In this paper we propose a technique for generating name pronunciations, based on the joint n-gram model described in [8]. In previous evaluations it was found that this technique performs better than other data-driven grapheme-to-phoneme conversion techniques. We will show here that this technique has very good performance on name pronunciation as well. We use data and testing conditions similar to those used by Font Llitjos and Black [7], which will make our results comparable to theirs.

We are also interested in the reverse task, of obtaining the name spelling from its pronunciation. The technique described in this paper is bi-directional, in that the same model can be used to convert orthographic input to pronunciations and phonemic input to spellings. Of course, recognition of names from spoken input is a more difficult task because of the additional challenge of obtaining the correct phonemes (assumed in phoneme-to-grapheme conversion). Note, however, that speech recognizers may use contextual information for better accuracy. Moreover, language models for speech recognition use frequency information, so that more frequent names tend to be transcribed more accurately. Thus, the spelling transcription results reported here cannot be taken as an estimation of the name error rate. Nonetheless, we can extract valuable information from our results about the potential of sub-lexical recognition of novel proper names.

2. STATE OF THE ART

High-quality commercial systems exist, but they use large sets of language-specific rules and sizable exception dictionaries [5, 14, 16]. Given the openness of the set of surnames in US English, maintaining a large set of pronunciation rules becomes a difficult task itself. We believe that a better, more general solution is to learn automatically the pronunciation rules that an educated American speaker may use when encountering a new name.

Similar goals stand behind other name-pronunciation systems reported in the literature. Golding used a clever evaluation procedure to show that a pronunciation by analogy system based on general transcription rules performs almost as well as commercial systems [10]. Unfortunately, given the subjective nature of that evaluation, it is difficult to compare his

results with those obtained with other methods. Yvon [18] proposes two other pronunciation by analogy techniques that achieve high accuracy on pronouncing French patronyms; we are not aware of any evaluation of these techniques for the decidedly more difficult task of pronouncing American English surnames. A similar approach [1] is reported to have achieved 93.9% phoneme accuracy on transcription of English names from the ONOMASTICA database [13]. No word accuracy is reported, and the design of the evaluation is not given (a relatively small set is used for training and testing, but it is unclear how the data was selected). Using a grapheme-to-phoneme conversion technique based on weighted rules automatically learned from the CNET lexicon (the set of correspondences was, however, hand-written), Bagshaw reports 49.21% word accuracy and 14.6% phoneme error rate on 68,046 names in the British section of the ONOMASTICA database [2].

More recently, decision trees have been used for automatic generation of name pronunciations. Ngan *et. al*. report on such a system achieving a 54.5% word accuracy, with a phoneme error rate of 13.28% on a database of 18,500 surnames (with close to 24,000 different pronunciations) collected and transcribed by the ISIP group at Mississippi State University [12]. The results were obtained with a three-fold cross-evaluation in which the data was split into 19,500 spelling-pronunciation pairs for training and 4,500 spelling-pronunciation pairs for testing.

Better results were obtained with another decision-tree technique by Font Llitjos and Black, on the more difficult task of predicting both the phonetic transcription and the stress assignment for names included in the CMU dictionary [17]. Specifically, their technique achieved 54.08% word accuracy and 89.02% letter accuracy on a set of 56,000 names (of which 90% was used for training and 10% was used for testing) [7]. This compares with 57.80% word accuracy and 91.99% letter accuracy on the full CMU dictionary, under similar testing conditions. Moreover, this study shows that knowing the language of origin can increase significantly the accuracy of their technique. By adding to their model language classification features from a letter trigram language identifier, they obtain a word accuracy of 61.72%, with a letter accuracy of 91.23%. We believe this is the best result on name pronunciation by grapheme-to-phoneme conversion for US English, although direct comparison with other methods is impossible due to differences in the data used for evaluation.

3. THE JOINT N-GRAM MODEL

The joint n-gram model is described in detail in [8]. We summarize here the approach, which we used without modification.

The first step in building the model is aligning the spelling and pronunciation for each entry in the dictionary. This step is carried out automatically using a version of the EM algorithm [6]. As a result of the alignment phase, the training data is segmented into grapheme-to-phoneme (g-p) correspondences. Thus, this approach differs substantially from most other grapheme-to-phoneme conversion methods, which usually employ hand-build sets of allowable g-p correspondences. Also, note that the alignment procedure allows for many-to-many correspondences; this way there is no need to modify the raw data by introducing pseudo-graphemes and pseudo-phonemes (nulls, doubles, etc.).

The sequences of g-p correspondences are used to train a back-off n-gram model. The model is a joint n-gram model because it predicts simultaneously the graphemic and the phonemic part of a correspondence. Thus, it can be used for both spelling-to-pronunciation (S2P) and pronunciation-to-spelling (P2S) conversion. For the mathematical formulation the reader is again referred to [8].

4. EXPERIMENTAL RESULTS

4.1. Data

For the experiments reported in this paper we used version 0.6 of the CMU pronunciation dictionary [17], which consists of more than 119,000 words, with a total of over 127,000 pronunciations. The phone set is composed of 39 phones. Lexical stress is indicated with one of three stress markers (0 = no stress, 1 = primary stress, 2 = secondary stress).

The list of proper names is the one included in the version 0.4 of the CMU dictionary, comprising the most frequent names and surnames in the US compiled from Bell Labs directory listings. It consists of 53,421 unique words, with over 54k pronunciations, as found in the full dictionary. These are not all the names in the CMU dictionary – foreign names gleaned out of news sources were not included in the name database.

In the following experiments, we randomly selected 10% of the full vocabulary for testing, and used the other 90% for training. The name portion of the training and test data follows the same distribution. We will refer to the full training data set (comprising both names and non-names) as FTRN, and to the names-only part of the training data as NTRN. Similarly, we will use FTST and NTST to refer to the corresponding test sets.

4.2. Models

The language models evaluated here are back-off n-gram models with Witten-Bell discounting, built with the CMU-Cambridge Toolkit [4].

For comparison, we experimented with joint 4-gram models built on FTRN and NTRN, with accented phonemes and non-accented phonemes. We also evaluated 5-gram models, but they performed slightly worse than 4-gram models (probably due to overtraining), so we won't report here those results.

The sets of grapheme-to-phoneme correspondences were learned separately for accented and non-accented phonemes on the FTRN data.

4.3. Spelling-to-pronunciation conversion

Results for spelling-to-pronunciation conversion are given in Tables 1 to 3. For comparison, we repeat the results obtained with the FTRN 4-gram model on the full test. Language models are identified according to the data set used for training. Reported are: word accuracy (WACC), phoneme accuracy (PACC), and phoneme error rate (PER).

As expected, generating pronunciations for names is significantly harder than that for non-names: the drop in word accuracy is over 7% absolute for non-accented phonemes and over 3% absolute for accented phonemes. On the other hand,

Model	Test data	WACC [%]	PACC [%]	PER [%]
FTRN	FTST	71.5	93.6	7.0
FTRN	NTST	67.4	91.9	9.1
NTRN	NTST	68.0	92.2	8.8
FTRN+NTRN	NTST	68.4	92.2	8.8

Table 1: S2P results for predicting non-accented phonemes

Model	Test data	WACC [%]	PACC [%]	PER [%]
FTRN	FTST	72.3	93.8	6.8
FTRN	NTST	68.3	92.1	8.9
NTRN	NTST	68.5	92.2	8.8

Table 2: S2P results for predicting non-accented phonemes with models based on accented phonemes

Model	Test data	WACC [%]	PACC [%]	PER [%]
FTRN	FTST	62.6	91.0	9.6
FTRN	NTST	60.8	89.5	11.5
NTRN	NTST	60.6	89.7	11.3

Table 3: S2P results for predicting accented phonemes

there are no significant performance differences between the FTRN models and the NTRN models. In the case of non-accented phonemes, the latter are slightly better, but this is not the case for the case of predicting accented phonemes. Just as in the general case, there is a small advantage in using stress information in the context, but stress assignment is not as reliable as phoneme prediction.

When we compared the results obtained with the FTRN and NTRN models, we found that they differ in over 13% of the test set (see Table 4). This suggests that specific knowledge of name pronunciation is good, but more general knowledge of English pronunciation rules can be helpful, too. Thus, perhaps better results can be obtained by adequately combining the two sources of knowledge. Indeed, when we trained a joint 4-gram model on the combined NTRN and FTRN data (that is, names were represented twice as often as non-names), the performance on non-accented phonemes increased to 68.4% word accuracy (the FTRN+NTRN model in Table 1). It appears that small improvements are possible by optimizing the mix of name and non-name training data, or, equivalently, by interpolating general-purpose models with name-specific models; however, for this study we haven't pursued this avenue.

4.4. Pronunciation-to-spelling conversion

Results for pronunciation-to-spelling conversion are given in Tables 5 and 6. Again, we repeat the results obtained with the FTRN 4-gram model on the full test. Reported are: word accuracy (WACC), letter accuracy (LACC), and letter error rate (LER).

		NTRN	
		Correct	Wrong
FTRN	Correct	3270	337
	Wrong	369	1372

Table 4: Confusion matrix between model trained on names and non-names (FTRN) and model trained on names only (NTRN)

Note that, just as for the full dictionary, language models based on accented phonemes perform slightly better. However, in the case of names, which seems a much more difficult task than common words, the increase in performance is significantly larger.

The drop in the P2S performance between non-names and names is about 18% absolute, much larger than the difference on the S2P task. This leads us to believe that, while text-to-speech systems may be able to effectively pronounce names at the same quality level as common words, sub-lexical recognition of names will be a much more difficult task. However, even in the absence of large pronunciation dictionaries for proper names, graphotactic models built on spellings from large directory listings may be used in conjunction with our P2S technique to achieve better transcription performance.

We did not test this hypothesis directly. Instead, we trained another joint 4-gram model on the NTRN data augmented with all the test names' pronunciations obtained automatically with our S2P technique. Although almost 33% of these pronunciations are erroneous, the model (NTRN+S2P in Table 6) achieves an impressive 57% word accuracy, which approaches the performance of the FTRN model on non-names. Since the set of g-p correspondences is not changed, we attribute this large improvement to the additional grapho-phonotactic regularities gleaned out of the test data within the constraints imposed by the existing set of g-p correspondences.

We conclude from this experiment that training models on

Model	Test data	WACC [%]	LACC [%]	LER [%]
FTRN	FTST	50.3	91.2	11.5
FTRN	NTST	40.2	88.2	15.5
NTRN	NTST	40.6	88.8	15.0
NTRN+S2P	NTST	57.0	91.9	10.9

Table 5: P2S results for models based on non-accented phonemes

Model	Test data	WACC [%]	LACC [%]	LER [%]
FTRN	FTST	50.6	91.6	11.2
FTRN	NTST	41.4	89.0	14.8
NTRN	NTST	41.5	89.4	14.4

Table 6: P2S results for models based on accented phonemes

large lists of names (which can be obtained from directory listings or genealogy lists on the internet) for which pronunciations are obtained automatically is a promising way to increase the accuracy of a sub-lexical recognizer for proper names.

5. CONCLUSION

In this paper we evaluated an existing technique for bi-directional grapheme-to-phoneme conversion on the task of obtaining automatically pronunciations for names. Our baseline results compare favorably with the performance of the decision-tree method of Font Llitjos and Black (see Section 2) even without using information about the names' ethnic origin, which boosted their method's word accuracy by more than 7% absolute. Nonetheless, we concur with them in finding that the pronunciation problem is more difficult than the S2P task for non-names. This result mirrors the conclusion of Surprenant *et. al.*, that educated American speakers have more difficulty pronouncing names than common nouns [15].

We expect to obtain further improvements with our models by using language models dependent on the ethnic origin, similar to topic-dependent language models [9]. We have not yet experimented with this model, which we leave for future work.

Using a bi-directional model gave us the opportunity to evaluate the possibility of performing the reverse task, of converting name pronunciation into spelling. We found this task to be much more difficult, which raises questions about the possibility of automatic transcription of names based on sub-lexical language models, even assuming perfect phoneme recognition (which the current technology is still far from achieving). However, we found that one can obtain dramatic improvements by capitalizing on the much better accuracy of automatic spelling-to-pronunciation techniques.

6. ACKNOWLEDGEMENTS

This work has been supported by ONR grant N00014-01-1-1015, DARPA grant F30602-98-2-0133, and a grant from the W.M. Keck Foundation.

7. REFERENCES

[1] Andersen, O. and Dalsgaard, P., "Multi-lingual testing of a self-learning approach to phonemic transcription of orthography". *Proc. Eurospeech'95*, Madrid, 1995.

[2] Bagshaw, P., "Phonemic transcription by analogy in text-to-speech synthesis: Novel word pronunciation and lexical compression". *Computer Speech and Language*, 12:119–142, 1998.

[3] Black, A.W., Lenzo, K., and Pagel, V., "Issues in building general letter to sound rules". *Proc. ESCA Workshop on Speech Synthesis*, pp. 77–80, Jenolan Caves, Australia, 1998.

[4] Clarkson, P. and Rosenfeld, R., "Statistical Language Modelling using the CMU-Cambridge Toolkit". *Proc. Eurospeech'97*, pp. 2707–2710, 1997.

[5] Coker, C.H., Church, K.W., and Liberman, M.Y., "Morphology and rhyming: Two powerful alternatives to letter-to-sound rules for speech synthesis". *Proc. of the ESCA Conference on Speech Synthesis*, pp. 83–86, Autrans, France, 1990.

[6] Dempster, A.P., Laird, N.M., and Rubin, D.B., "Maximum likelihood from incomplete data via the EM algorithm". *Journal of the Royal Statistical Society*, 39B:1–38, 1977.

[7] Font Llitjos, A. and Black, A.W., "Knowledge of language origin improves pronunciation accuracy of proper names". *Proc. Eurospeech-2001*, Aalborg, Denmark, 2001.

[8] Galescu, L. and Allen, J.F., "Bi-directional conversion between graphemes and phonemes using a joint n-gram model". *Proc. 4th ISCA Tutorial and Research Workshop on Speech Synthesis*, Pertshire, Scotland, 2001.

[9] Gildea, D. and Hofmann, T., "Topic-based language models using EM". *Proc. Eurospeech'99*, Budapest, Hungary, 1999.

[10] Golding, A.R. and Rosenblum, P.S., "A comparison of Anapron with seven other name-pronunciation systems", *Journal of the American Voice Input/Output Society*, 14:1–21, 1993.

[11] Marcus, M., Santorini, B., and Marcinkiewicz, M., "Building a large annotated corpus of English: the Penn Treebank". *Computational Linguistics*, 19:313–330, 1993.

[12] Ngan, J., Ganapathiraju, A., and Picone, J., "Improved surname pronunciations using decision trees". *Proc. International Conference on Spoken Language Processing*, pp. 3285–3288, Sydney, Australia, 1998.

[13] Schmidt, M., Fitt, S., Scott, C., and Jack, M., "Phonetic transcription standards for European names (ONOMASTICA)". *Proc. Eurospeech'93*, vol. 1, pp. 279–282, Berlin, Germany, 1993.

[14] Spiegel, M.F. and Macchi, M.J., "Synthesis of names by a demisyllable-based speech synthesizer (Orator)". *Journal of the American Voice Input/Output Society*, 7:1–10, 1990.

[15] A.M. Surprenant, *et. al.*, "Familiarity and Pronounceability of Nouns and Names: The Purdue Proper Name Database". *Proc. 16th International Congress on Acoustics and 135th Meeting Acoustical Society of America,* pp. 2007–2008, Seattle, WA, 1998.

[16] Vitale, A.J., "An algorithm for high accuracy name pronunciation by parametric speech synthesis". *Journal of Computational Linguistics*, 17(3):257–276, 1991.

[17] Weide, R., "The CMU Pronunciation Dictionary, release 0.6". Carnegie Mellon University, 1998.

[18] Yvon, F., "Self-learning techniques for grapheme-to-phoneme conversion". *Proc. 2nd Onomastica Research Colloquium*, London, 1994.