Letter-to-sound Pronunciation Prediction Using Conditional Random Fields

Dong Wang, *Associate Member, IEEE,* Simon King, *Senior Member, IEEE,*

***Abstract*—Pronunciation prediction, or letter-to-sound (LTS) conversion, is an essential task for speech synthesis, open vocabulary spoken term detection and other applications dealing with novel words. Most current approaches (at least for English) employ data-driven methods to learn and represent pronunciation "rules" using statistical models such as decision trees, hidden Markov models (HMMs) or joint-multigram models (JMMs). The LTS task remains challenging, particularly for languages with a complex relationship between spelling and pronunciation such as English. In this paper, we propose to use a conditional random field (CRF) to perform LTS because it avoids having to model a distribution over observations and can perform global inference, suggesting that it may be more suitable for LTS than decision trees, HMMs or JMMs. One challenge in applying CRFs to LTS is that the phoneme and grapheme sequences of a word are generally of different lengths, which makes CRF training difficult. To solve this problem, we employed a joint-multigram model to generate aligned training exemplars. Experiments conducted with the AMI05 dictionary demonstrate that a CRF significantly outperforms other models, especially if n-best lists of predictions are generated.**

***Index Terms*—letter-to-sound, conditional random field, joint multigram model, speech synthesis, spoken term detection**

I. Introduction

PREDICTING pronunciations for novel words – commonly called letter-to-sound (LTS) conversion – is most commonly employed in text-to-speech (TTS) synthesis. Early LTS systems comprised sets of hand-crafted rules [1, for example] but modern LTS systems more often use a data-driven approach in which both widely-applicable phonological rules and exceptions can be learned from training data and represented as a model. Data-driven extensions of the rule-based approach attempt to formulate rules using training data [2, for example]. Exemplar-based approaches attempt to generalize from stored examples. Examples of these approaches include instance-based reasoning [3] and analogy-based reasoning [4, for example]. Artificial neural networks have also been applied to the problem, starting with NETtalk [5].

The HMM was applied to LTS in [6] and recently enhanced in [7]. With this model, phonemes are regarded as states that form a Markov chain, from which grapheme observations are drawn independently. The task of pronunciation prediction is then to find the optimal state sequence within the HMM, given a spelling of a word as the observation sequence. The decision tree is another widely used LTS model. This model determines the pronunciation of each grapheme in turn by examining its local grapheme context. In its simplest form, the tree is used to organize and compactly store training exemplars [3], [8]. More complex configurations include ID3 and its successor C4.5 [9], which determines the questioned grapheme context in a dynamic way as the tree grows [10], [11]. Another type of tree is the classification and regression tree (CART), proposed by [12] and thoroughly studied by [13]. This kind of tree is grown by testing a set of binary questions for each node, and choosing the best question according to some measures.

The last statistical model for LTS in our concern is the joint-multigram model (JMM), originally proposed by [14] and applied to LTS by various researchers, such as [15], [16]. In this model, a word's spelling and its pronunciation are assumed to have been derived from some single underlying process of human language, meaning that graphemes and phonemes should be modelled together via their joint probabilities. Recently, [17] discussed various issues when building joint-multigram models and reported that the joint-multigram model consistently outperformed other models on the LTS task.

This paper presents a new LTS approach based on a conditional random field (CRF). In general, the task of LTS can be expressed as:

$$\tilde{Q} = \arg\max_{Q} P(Q|G) \tag{1}$$

where G is the spelling (grapheme sequence) of the word and Q is a candidate pronunciation. The LTS task amounts to searching for the $\tilde{Q}$ that maximizes $P(Q|G)$. Of the models described above, HMMs and joint-multigrams are generative models, i.e., they model the joint probabilities of graphemes and phonemes and derive the posterior probability $P(Q|G)$ according to the Bayes rule; on the other hand, ANNs and decision trees are discriminative models which estimate the posterior probability directly. From another perspective, HMMs and joint-multigram models perform global inference, meaning that they search for the optimal pronunciation as an entire phoneme sequence (even if this is based only on a sliding localised window of grapheme observations), while ANNs and decision trees perform piece-wise inference to generate pronunciations for individual graphemes and then compose the word pronunciation by concatenation. A discriminative model may be superior to a generative model because it does not need to model the (possibly complex) distributions over observations; on the other hand, global inference will probably be superior to the piece-wise inference. Therefore, an ideal LTS model should be a discriminative model that performs global inference, which none of the above models does.

A CRF is a conditional discriminative model that performs global inference; this appears to be eminently well-suited to the problem of letter-to-sound conversion. In the next section, we first introduce the CRF and discuss how to apply it to

LTS. In Section III we show how to prepare aligned, labeled data for model training using a joint-multigram model. Section IV gives the experimental conditions and results, followed by some conclusions and ideas for future work in Section V.

II. Conditional random field for pronunciation prediction

A CRF models the conditional probability distribution of a label sequence given an observation sequence. As a discriminative and conditional model, the CRF is a powerful tool for modeling sequential data and has received much interest in applications as diverse as text processing [18], [19], bioinformatics [20], computer vision [21] and speech recognition [22], [23].

The LTS task can be regarded as a labeling procedure whereby the word spelling (a grapheme sequence) is observed and the pronunciation (a phoneme sequence) is the label sequence to be inferred. Compared to other models, the CRF has a number of properties that make it well-suited for this task. First, the CRF is a conditional model that relaxes the conditional independence assumption required by generative models such as HMMs; second, the CRF performs inference over entire label sequences, unlike the piecewise inference conducted by other conditional models such as decision trees and ANNs; third, the CRF is a discriminative model and thus does not need to model the joint probability distribution of observations (graphemes) as in a JMM. Finally, the loss function of CRFs is convex, guaranteeing convergence to the global optimum [24].

Using the definition from [24], a CRF applied to LTS can be written as

$$P(Q|G) = \frac{1}{Z(G)} exp\{\sum_{k=1}^{K} \lambda_k F_k(Q,G)\} \tag{2}$$

where G is the grapheme sequence of the word, Q is a candidate pronunciation, F_k is the k-th aggregated feature and λ_k is a factor to scale its contribution. $Z(G)$ is a normalization quantity given by

$$Z(G) = \sum_{Q} exp\{\sum_{k=1}^{K} \lambda_k F_k(Q,G)\}. \tag{3}$$

Considering the Markov assumption, the undirected graph of the CRF can be separated into cliques, each of which contains two consecutive phonemes and the entire grapheme sequence. Therefore, the aggregated feature $F_k(Q,G)$ can be factored into feature functions of cliques, given by

$$F_k(Q,G) = \sum_{j=1}^{n-1} \{f_k(Q_j, Q_{j-1}, G, j)\} \tag{4}$$

where $f_k(Q_j, Q_{j-1}, G, j)$ is the k-th feature function of the j-th clique and n is the length of the grapheme sequence.

A commonly used family of features are binary functions that return binary values by examining the graphemes and phonemes at various positions in the clique. For example, the following feature function returns a non-zero value if and only if the current and previous graphemes are H and I respectively and the current phoneme is $/i/$

$$f(Q_j, Q_{j-1}, G, j) = \begin{cases} 1 & \text{if } G_{j-1} = H, G_j = I, Q_j = /i/ \\ 0 & \text{otherwise} \end{cases}$$

We used the toolkit CRF++ v0.52 provided by Taku Kudo of NTT Communication Science Laboratories in Japan [25] for training and inference. A number of features make the tool easily to use. First, features can be defined easily by specifying the concerned graphemes and phonemes; second, the limited memory BFGS (LBFGS) algorithm makes the training fast with moderate memory usage, allowing more features to be taken into account (up to 139 000 000 in our experiment); third, posterior probabilities of candidate pronunciations are available, based on which n-best predictions can be achieved.

III. Joint-multigram model-based alignment

To train a CRF model, labeled exemplars are required. In the LTS task, however, the phoneme sequence and grapheme sequence of a word are generally of different lengths, which means an alignment must be found before an exemplar can be used for training. We employ a JMM to perform this task as it has shown good performance in LTS applications [26].

A joint-multigram model for LTS represents the probability distribution over sequences of phoneme-grapheme joint units. Following the notation of Bisani and Ney [15], we call a grapheme-phoneme joint unit a *graphone*, denoted by $u = (\tilde{g}, \tilde{q})$ where $\tilde{g}$ and $\tilde{q}$ are the grapheme and phoneme component of u respectively. Both $\tilde{g}$ and $\tilde{q}$ contain a sequence of symbols whose length is from N_{min} to N_{max}. With graphones defined, the joint probability of spelling G and pronunciation Q can be written in graphones U as:

$$P(G,Q) = \sum_{U;G(U)=G,Q(U)=Q} P(U) \tag{5}$$

$$= \sum_{U;G(U)=G,Q(U)=Q} P(u_1, u_2, ..., u_K) \tag{6}$$

where $G(U)$ and $Q(U)$ denote the grapheme and phoneme component of U, respectively.

In previous studies, JMMs have been used to perform the prediction task directly [15], [16], [26]. In the CRF-based LTS, instead of making predictions, we employ the JMM simply to align phoneme and grapheme sequences that are of different lengths, by finding the graphone sequence $\hat{U}$ that satisfies

$$\hat{U} = \arg\max_{U} P(U). \tag{7}$$

As in the work of others [16], [17], we factor the probability $P(U)$ using a graphone n-gram model:

$$P(U) = \prod_{j=1}^{|U|} P(u_j|h_j) \tag{8}$$

where h_j is the graphone history of u_j. In our experiment, a 4-gram model gave the best accuracy on an LTS task, so this configuration was chosen for performing grapheme/phoneme alignment.

In this work, we used 0-1 graphones ($N_{min} = 0$ and $N_{max} = 1$) for alignment, meaning that either one or zero phoneme is allowed to be aligned to one grapheme, and vice versa. Larger graphones did not work, since the explosive increase in the number of features and labels they produce led to excessive memory requirements.

Together, the JMM-based alignment and the CRF-based prediction comprise a fully automatic LTS system. Given a dictionary as training data, the system learns a statistical model that is optimized with respect to the task objection function (1) and which predicts globally optimal pronunciations.

IV. Experiments

A. Experimental settings

We performed our experiments on the dictionary used by the AMI RT05s LVCSR system [27], with 36575 words randomly selected for training, 4064 words for parameter tuning and 8000 words for evaluation. Three LTS systems were built and compared:

1) CART, as used by the Festival speech synthesizer [28];
2) JMM, since this exhibited good performance [17], [26];
3) CRF, as proposed here.

For the CART-based system, we followed the training process in [29] in which an *allowable table* that specifies the allowable pronunciations of each letter is manually created using trial & error, and a CART is learned for each letter, with configuration settings (especially the "stop value") chosen to optimize performance on the development set. For the JMM-based system, we built two models. The first model was used to align the training data, for which we set N_{min}=0, $N_{max} = 1$. The second model was used to perform LTS, for which we set N_{min}=1, $N_{max} = 2$. This model represented the best performance that we could obtain with a JMM [26]. For the CRF-based system, the most important thing is to choose suitable feature sets because this has a large effect on prediction accuracy. The CRF++ tool provides an easy way to define feature sets by allowing users specifying a grapheme/phoneme context, from which all possible choices of graphemes/phonemes within this context constitute a set of binary functions, which are in turn used as features. We explored a wide range of contexts to optimize the configuration (based on the performance on the development set).

In the following experiments, we choose the word error rate (WER) as the major evaluation metric: a prediction is correct only if the predicted pronunciation matches the canonical pronunciation exactly. If a word has multiple pronunciations (which is the case of the AMI05 dictionary), the prediction is assumed to be correct if it matches any of the canonical pronunciations. The phone error rate (PER) is also provided in the case that it reveals additional information.

B. 1-best prediction

In our first experiment, we employed the various models to predict 1-best pronunciations. The results are shown in Table I. For the JMM, we report both the best configuration (1-2 model) and the performance of the model used for alignment (0-1 model) – the latter is presented only for comparison with the CRFs, since they are based on the same alignment of the training examples.

For the CRF, we examined different widths of grapheme contexts that the features may cover. For example, (-2,2) means that the features may cover two graphemes preceding and following the current position. The context always includes exactly two phonemes in each feature function (4).

TABLE I
Results of 1-best pronunciation prediction

Model	WER (%)	PER (%)
CART	35.2	8.7
JMM (1-2 model)	31.3	7.8
JMM (0-1 model)	41.3	11.2
CRF (-1,+1)	67.6	18.5
CRF (-2,+2)	40.9	9.6
CRF (-3,+3)	29.7	6.8
CRF (-4,+4)	25.4	5.8

We observe that the prediction accuracy of the CRF-based system rapidly increases as more features are involved. In this experiment, the (-4,4) grapheme context achieved the best result; wider contexts were prohibited by memory limitations. The best CRF achieved far better performance than the CART or JMMs. A pairwise t-test shows that this is highly significant ($p < 10^{-14}$ over JMM and $p < 10^{-31}$ over CART). This supports our hypothesis that a global and conditional model (i.e., the CRF) is more suitable for LTS. The CRF far outperformed the (0-1) JMM, using the same alignment of the training examples. The JMM must use larger graphones to gain respectable performance, which suggests that the CRF and JMM are substantially different with respect to describing context dependency among phoneme and grapheme sequences, as well as the conditional relationship between them.

C. N-best prediction

In the second experiment, we consider n-best predictions because these are sometimes desirable to compensate for potential errors (see [26] for applications in spoken term detection). For a CART, inference is based on individual phonemes and the confidence (posterior probability) is rather difficult to smooth, so n-best prediction is not very reliable. For the JMM, the confidence is based on the entire candidate pronunciation, which allows global n-best predictions; however the confidence scores have to be derived from joint probabilities by applying Bayes' rule. This is usually achieved by calculating the posterior probabilities of the pronunciation in the lattices, which is only an approximation and might be inaccurate. CRFs, on the other hand, provide a simple and straightforward way to compute the posterior probabilities of each candidate pronunciation.

We predicted between 1 and 50 best pronunciations with the JMM and CRF model and compared their performance in terms of n-best WER, i.e., the proportion of the words for whom none of the n-best predictions is correct. Figure 1 presents the results, where we can see the CRF provided higher quality n-best prediction than the JMM.

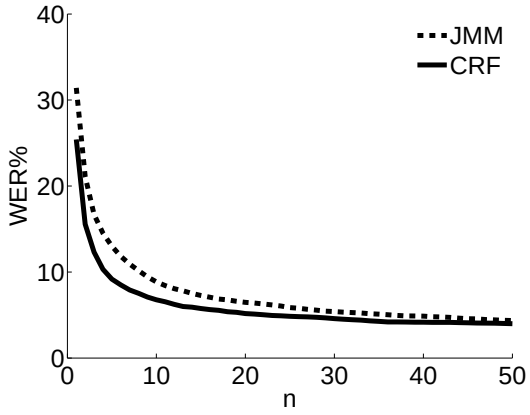

Fig. 1. WER% of n-best predictions with JMM and CRF.

V. Conclusion

This paper proposed using a CRF to perform pronunciation prediction for novel words. We found a significant accuracy improvement over both CART and JMM in our experiments. The CRF also provided better n-best pronunciation lists than the JMM. An interesting idea for the future would be to model units derived from larger graphones using CRFs.

Acknowledgment

Dr. Dong Wang was a Fellow of the EdSST Marie Curie training programme when this work was carried out. This work was partially supported by the French Ministry of Industry (Innovative Web call) under contract 09.2.93.0966, "Collaborative Annotation for Video Accessibility" (ACAV), and by the Adaptable Ambient Living Assistant (ALIAS) project funded through the joint national Ambient Assisted Living (AAL) programme.

References

[1] D. H. Klatt, "Review of text-to-speech conversion for English," *Journal of the Acoustical Society of America*, vol. 82, no. 3, pp. 737–793, 1987.

[2] P. C. Bagshaw, "Phonemic transcription by analogy in text-to-speech synthesis: novel word pronunciation and lexicon compression," *Computer Speech & Language*, vol. 12, no. 2, pp. 119–142, April 1998.

[3] W. Daelemans, A. van den Bosch, and T. Weijters, "IGTree: Using trees for compression and classification in lazy learning algorithms," *Artificial Intelligence Review, Special Issue on Lazy Learning*, vol. 11, pp. 407–423, 1997.

[4] R. Damper and J. Eastmond, "Pronunciation by analogy: Impact of implementational choices on performance," *Language and Speech*, vol. 40, no. 1, pp. 1–23, 1997.

[5] T. J. Sejnowski and C. R. Rosenberg, "Parallel networks that learn to pronounce English text," *Complex Systems*, vol. 1, no. 1, pp. 145–168, 1987.

[6] S. H. Parfitt and R. A. Sharman, "A bi-directional model of English pronunciation," in *Proc. Eurospeech'91*, Genoa, Italy, September 1991, pp. 801–804.

[7] P. Taylor, "Hidden Markov models for grapheme to phoneme conversion," in *Proc. Interspeech'05*, Lisbon, Portugal, September 2005, pp. 1973–1976.

[8] K. Torkkola, "An efficient way to learn English grapheme-to-phoneme rules automatically," in *Proc. ICASSP'93*, Minneapolis, MN, USA, April 1993, pp. 199–202.

[9] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, February 1992.

[10] V. Pagel, K. Lenzo, and A. W. Black, "Letter-to-sound rules for accented lexicon compression," in *Proc. ICSLP'98*, vol. 5, Sydney, Australia, November 1998, pp. 2015–2018.

[11] J. Häkkinen, J. Suontausta, S. Riis, and K. J. Jensen, "Assessing text-to-phoneme mapping strategies in speaker independent isolated word recognition," *Speech Communication*, vol. 41, no. 2, pp. 455–467, 2003.

[12] J. Lucassen and R. Mercer, "An information theoretic approach to the automatic determination of phonetic baseforms," in *Proc. ICASSP'84*, San Diego, California, USA, March 1984, pp. 42.5.1–42.5.4.

[13] L. Breiman, J. Friedman, C. J. Stone, and R. Olshen, *Classification and regression trees*. Chapman & Hall;, January 1984.

[14] S. Deligne, F. Yvon, and F. Bimbot, "Variable-length sequence matching for phonetic transcription using joint multigrams," in *Proc. Eurospeech'95*, Madrid, Spain, September 1995, pp. 2243–2246.

[15] M. Bisani and H. Ney, "Investigations on joint-multigram models for grapheme-to-phoneme conversion," in *Proc. ICSLP'02*, Denver, USA, September 2002, pp. 105–108.

[16] S. F. Chen, "Conditional and joint models for grapheme-to-phoneme conversion," in *Proc. Eurospeech'03*, Geneva, Switzerland, September 2003, pp. 2033–2036.

[17] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, May 2008.

[18] F. Sha and F. Pereira, "Shallow parsing with conditional random fields." in *Proc. HLT/NAACL-03*, Edmonton, Canada, 2003.

[19] X. Xin, J. Li, J. Tang, and Q. Luo, "Academic conference homepage understanding using constrained hierarchical conditional random fields," in *Proc. the 17th ACM conference on Information and knowledge management*, Napa Valley, California, USA, 2008, pp. 1301–1310.

[20] A. Culotta, D. Kulp, and A. McCallum, "Gene prediction with conditional random fields," University of Massachusetts, Amherst, Tech. Rep., 2005.

[21] C. Smimchisescu, A. Kanaujia, Z. Li, and D. Metaxus., "Conditional models for contextual human motion recognition," in *Proc. the International Conference on Computer Vision, (ICCV 2005)*, Beijing, China, 2005.

[22] Y. Hifny and S. Renals, "Speech recognition using augmented conditional random fields," *ASLPA05*, vol. 17, no. 2, pp. 354–365, Feburary 2009.

[23] Y.-H. Sung and D. Jurafsky, "Hidden conditional random fields for phone recognition," in *Proc. ASRU'09*, Merano, Italy, 2009.

[24] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th International Conf. on Machine Learning*, San Francisco, CA, USA, 2001, pp. 282–289.

[25] T. Kudo, 2009. [Online]. Available: http://crfpp.sourceforge.net/

[26] D. Wang, "Out-of-vocabulary spoken term detection," Ph.D. dissertation, The Center for Speech Technology Research, Edinburgh University, December 2009.

[27] T. Hain, L. Burget, J. Dines, G. Garau, M. Karafiat, M. Lincoln, J. Vepa, and V. Wan, "The AMI meeting transcription system: Progress and performance," in *Machine Learning for Multimodal Interaction*. Springer Berlin/Heidelberg, 2006, vol. 4299/2006, pp. 419–431.

[28] R. A. J. Clark, K. Richmond, and S. King, "Multisyn: Open-domain unit selection for the Festival speech synthesis system," *Speech Communication*, vol. 49, no. 4, pp. 317–330, 2007.

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

Dong Wang received the B.Sc. and M.Sc. in computer science at Tsinghua Univ. in 1999 and 2002, and then worked for Oracle China in 2002-2004 and IBM China in 2004-2006. He joined CSTR, University of Edinburgh in 2006 as a research fellow and PhD student supported by a Marie Curie fellowship, from where he received his Ph.D. in 2010. He is now working in EURECOM, France as a post-doc researcher. His research focuses on speech recognition and spoken term detection and extends to stochastic modelling and machine learning generally.

Simon King (M'95, SM'08) received M.A.(Cantab) and M.Phil. degrees in Engineering from the University of Cambridge in 1992 and 1993 and a Ph.D. from the University of Edinburgh in 1998. He is a Reader in Linguistics and English Language and his interests include speech synthesis, recognition and signal processing. He serves on ISCA SynSIG committee, co-organises Blizzard Challenge, was recently an assoc. ed. of IEEE Trans. Audio, Speech & Lang. Proc., is on the IEEE SLTC and the editorial board of Computer Speech and Language.