Letter-based speech synthesis

Oliver Watts, Junichi Yamagishi, Simon King

Centre for Speech Technology Research, University of Edinburgh, UK

O.S.Watts@sms.ed.ac.uk jyamagis@inf.ed.ac.uk Simon.King@ed.ac.uk

Abstract

Initial attempts at performing text-to-speech conversion based on standard orthographic units are presented, forming part of a larger scheme of training TTS systems on features that can be trivially extracted from text. We evaluate the possibility of using the technique of decision-tree-based context clustering conventionally used in HMM-based systems for parameter-tying to handle letter-to-sound conversion. We present the application of a method of compound-feature discovery to corpus-based speech synthesis. Finally, an evaluation of intelligibility of letter-based systems and more conventional phoneme-based systems is presented.

Index Terms: Statistical parametric speech synthesis, HMM-based speech synthesis, letter-to-sound conversion, graphemes.

1. Introduction

This paper presents initial attempts at performing text-to-speech (TTS) conversion based on standard orthographic units. It forms part of a larger scheme of training TTS systems on "naive" features: features that can be trivially extracted from text. We contrast this approach with the one conventionally followed in TTS, where some intermediate representation is constructed to bridge the gap between text and speech; this representation will here be called a "linguistic specification". This specification is given in terms of features based on linguistic knowledge, such as phonemes, syllables, intonational phrases, *etc.*. It can be derived from text by means of a lexicon and a set of classifiers, which will here be collectively termed a "front end". Our motivation for seeking to avoid the need to use such an intermediate representation is the expense associated with constructing a front end. This is a far from trivial task, involving someone with knowledge of the language in question either writing rules or annotating surface forms with the corresponding feature to be used in the linguistic specification. For example, words might be labelled with phonemes in the lexicon, or with syntactic category in a corpus for training a part-of-speech classifier, and syllables might be labelled with pitch accents in a corpus for training an intonation module. This annotated data will here be called "secondary data" to distinguish it from what we will call "primary data": recorded speech, aligned on the utterance level with a transcription in standard orthography.

In HMM-based synthesis, there is not a one-to-one mapping between the unit types detailed in the linguistic specification and the units whose acoustic parameters are estimated during training. Speech is typically modelled at the phoneme level, each phoneme being represented by a speech unit having attributes specifying its phonetic and prosodic context (e.g. neighbouring phonemes, place in syllable, whether the current syllable bears stress or a pitch accent *etc.*). This context-dependency results in a vast number of possible units: almost all units in the training corpus will be of a unique type and at synthesis time, most models that are required to be synthesised will be of unseen types. Therefore a method is needed to map from the vast set of possible logical models to a set that is small enough that there are sufficient data to estimate model parameters during training, and general enough to represent unseen units at synthesis time. The technique generally employed for this purpose is decision-tree based clustering [1, 2]. Our intention in this paper is to evaluate the possibility of using this technique for handling letter-to-sound conversion in addition.

A similar experiment is reported in [3] in the context of cluster-based unit selection synthesis. The target language in that case was Spanish; the notoriously complex and irregular letter-to-sound correspondences of English make using it as our target language very ambitious. This is also shown by findings such as those reported in [4], where the performance of grapheme- and phoneme-based systems on speech recognition tasks in German, English and Spanish are compared. Word error rates for grapheme systems are slightly higher than for phoneme systems in the case of German and Spanish, but significantly higher in the case of English. However, the advantage of starting with something like a worst case scenario among languages with alphabetic writing systems is that we expect any techniques we find to improve synthesis based on these noisy orthographic units to give more marked improvements in languages where the letter-to-sound correspondence is more straightforward.

2. Systems Built

We assembled four systems to evaluate the possibility of performing TTS in English using plain orthography features: two letter-based systems (L-BAS and L-SER) and, for comparison, two more conventional phoneme-based systems making use of a pronouncing dictionary (P-FUL and P-LIM). The distinguishing characteristics of these systems are summarised in Table 1 and explained in the following paragraphs.

2.1. Data

The data used for these experiments was the SLT part of the ARCTIC database [5], of which only the audio and text transcription were used. The transcription was checked before use and manually preprocessed, all numerals and abbreviations being correctly expanded.

2.2. Initial alignment

Separate initial alignments of the audio and text-derived units were prepared for the two pairs of systems (the L and P systems). The P alignment used phonemes obtained from the plain orthographic transcription by look-up in the CMU pronouncing dictionary [6] as its basic units (phoneme inventory of 54 units, including 15 stressed variants of vowels), whereas the L alignment used a "naive lexicon", mapping tokens to sequences com-

Table 1: Summary of the systems built.

Identifier	Description	Modelling unit	Run-time lexicon and CART training data	Decision Tree Method
L-BAS	Letter-based baseline	Letter	*n/a*	Standard 1-pass
L-SER	Letter-based, serial tree-building	Letter	*n/a*	Serial tree-building
P-FUL	Phoneme-based with full lexicon	Phoneme	Full CMU lexicon	Standard 1-pass
P-LIM	Phoneme-based with limited lexicon	Phoneme	CMU lexicon entries for training set items	Standard 1-pass

posed of the 26 lowercase letters of English (see Table 2). In the case of the P-alignment, all out-of-vocabulary words found in the *training* data were added manually to the lexicon. In all other respects the procedure used for deriving the P and L alignments was identical. In both cases, the location of punctuation marks was used to initialise a silence model, and later the insertion of silence between words (orthographic spaces) was allowed where supported by the audio; selection of alternative pronunciations from the lexicon was also allowed during alignment, although in the case of the naive lexicon there were obviously no variants to choose from. Other details of model structure, parameterisation *etc.* used to obtain the alignment can be found in [7]. Informal visual comparison of the two alignments shows that at the word level they are very similar, and that reasonable assignments of letters to acoustic segments are generally made in the case of the L-alignment.

2.3. Letter-to-sound rules

The L systems require no extra letter-to-sound (LTS) rules beyond the decision trees that are constructed during voice building. For the P systems, however, LTS modules are needed to deal with out-of-vocabulary (o.o.v.) words at synthesis time. We decided to build two different LTS modules, and it is the difference between these modules that distinguishes between systems P-FUL and P-LIM. In both cases, classification trees were constructed using tools from the Edinburgh Speech Tools Library [8]. In the case of the P-FUL tree, the whole of the CMU dictionary was used as training data; in the case of P-LIM, however, the tree was trained on only those lexical entries used to label the training corpus during forced alignment. At synthesis time, both systems attempt look-up in their lexicon: P-FUL in the complete CMU lexicon and P-LIM in the much smaller training lexicon (2333 entries), and fall back to their respective Classification and Regression Trees (CARTs) in the case of o.o.v. words. The decision to handle o.o.v. words differently in these two systems was motivated by the fact the L systems are very limited in the amount of LTS training examples they are exposed to, and we wanted a phoneme-based system that is similarly limited for comparison. In this way, it is possible to determine to what extent the expected superior performance of phoneme-based systems is due their use of linguistically plausible modelling units, and on the other hand to what extent it is due to their reliance on the lexicon's encoding of the pronunciation of unseen words.

2.4. Contextual Features

From the transcriptions obtained during initial alignment, context-dependent label files were constructed for both the P and L voices. Other than the fact that the P labels use phones and the L labels letters, the labels are of identical form and encode the same set of contexts: the identity of units in each position of a 7-letter context window, the number of units since the start of the word, and the number of units until the end of the word. Neither system made use of features above the word level (relating to e.g. position in phrase or utterance). The use of a wider context window than the standard five units is inspired by features typically used in building CART trees for LTS. Note that unlike in LTS trees, the context units in the window may also be taken from neighbouring words, as the features are expected to deal not only with LTS correspondences but also with the type of co-articulatory effects for which decision-tree-based context clustering is conventionally used.

The questions used to query units' features in decision-tree construction were a conventional set of phonetically-motivated categories in the case of the P-voices. In the case of the L-voices, however, the questions were the most naive possible, assuming no knowledge of any natural classes into which letters might fall (i.e. all questions refer to single letters). The automatic discovery of useful categories of units for tree-building questions has been addressed by several researchers in speech recognition [9, 10, 11], and although it forms a part of our ongoing research, such techniques are not evaluated here.

2.5. Voice Building Procedure and Serial Tree Building

The procedure followed for building voices L-BAS, P-FUL and P-LIM was the same as that used to build the HMM Speech Synthesis System (HTS) group's entry in the 2005 Blizzard Challenge [12]. The procedure used for L-SER was the same except for the addition of a *serial tree-building* procedure at the final iteration of context clustering of spectral envelope parameters; this procedure is motivated and described below.

2.5.1. Tree-building and data fragmentation

A possible weakness for tree-based methods which becomes apparent when the input feature vectors have high-dimensionality and the structure to be uncovered has a Boolean structure is over-fragmentation of the data, which can disguise the data's structure [13, pp. 136ff]. Such Boolean structure is obviously present in sets of rules which capture English LTS correspondences, to a much greater extent, for example, than the sorts of rules necessary to predict co-articulatory effects. Take for example the set of words shown in node 0 of the tree in Figure 1A, and the sort of rule necessary to encode the pronunciation of ⟨a⟩ in these words as either [a] or [ei] (represented in the diagram by green and red respectively; note that this diagram could represent either a CART tree for LTS rules or an HTS state clustering tree where letter-based features are used). The question 'is the letter 2 places to the right an *e*?' is not sufficient to split the set of words appropriately because of the exceptional pronunciation of the ⟨a⟩ in *have*; this exception means that only a Boolean combination of features can split the set appropriately. In standard tree-building procedures, however, questions are asked one at a time leading either to impure nodes if splitting stops in the state depicted in Figure 1A, or over-fragmentation if splitting continues till the nodes are pure (as in Figure 1B, where items that should be together are split apart, both in nodes 2 and 5 and

Table 2: Sample entries from dictionaries used in experiments.

Naive Lexicon	CMU Lexicon
a a	a ah
abandonment a b a n d o n m e n t	a ey1
able a b l e	abandonment ah b ae1 n d ah n m ah n t
abnormal a b n o r m a l	able ey1 b ah l
about a b o u t	abnormal ae b n ao1 r m ah l
abstractions a b s t r a c t i o n s	about ah b aw1 t
...	abstractions ae b s t r ae1 k sh ah n z
	...

nodes 4 and 6).

Empirical investigation shows that heavy fragmentation is not detrimental to the predictive performance of CART trees built for LTS and that splitting till total node purity gives the best results [14]. Such is not the case, however, in the context of the rather different problem of decision tree building for state-tying of acoustic models. As with CART building for LTS, decision-tree-based clustering involves building a classifier for unseen models in future. Unlike CART for LTS, however, it also needs to solve the model-selection problem: the number and extent of the classes to which input examples are to be assigned is not pre-determined. Therefore, an explosion in the number of leaf nodes is an explosion in the number of classes chosen to partition the training set (unlike in LTS tree building, where many different leaves can share a single class). Over-fragmentation of data in DT building will lead to models poorly estimated due to shortage of training data. A phenomenon we have observed in real trees is that such over-fragmentation is often accompanied by under-fragmentation in other parts of the same tree. This is understandable as we use a Minimum Description Length criterion to determine at which point tree-building should cease [2]. This criterion is designed to balance the increasingly good fit of the model to the data and the concomitant increasing complexity of the model in an appropriate way. However, Description Length is computed globally over the tree as a whole. In effect, by creating many pure but fragmented clusters early in tree-building, we are getting bad value in terms of increased likelihood for the extra model parameters used. If free parameters are wasted through fragmentation in one part of the tree, it is understandable that splitting could stop in a locally premature way in another part of the tree.

We hypothesise that this under-fragmentation is one of the causes of the general degradation in the quality of synthetic speech we have observed from models built using orthographic features. The problem of inappropriate averaging in HMM-based synthesis is well-recognised generally (e.g. [15]), and we consider the general degradation in speech quality to be an especially heightened case of such inappropriate averaging, heightened because of the poor clusters that the naive orthographic features allow to form.

2.5.2. Serial Tree-Building

Various researchers have proposed methods to overcome these problems with tree-building, e.g. [16]; the one we adopt here is closely based on that explained in [17]. This approach can be characterised as finding 'compound questions': questions that query the values of more than one linguistic attribute simultaneously. Tree-building proceeds iteratively: a tree is built that clusters the units, and the leaf nodes of this tree are added as features to the names of the models that have passed through them. The tree is then put to one side, but questions can now be asked about the new features it has provided in subsequent iterations. The tree produced in the final iteration is the tree that is finally used in the normal way. In effect, this allows questions to be asked (indirectly) about several linguistic attributes simultaneously: the new features represent Boolean combinations of the original questions with the AND and NOT operators.

As a toy example, take the tree in Figure 1C. We start by placing all model names in the root node (0), and extending them with features indicating through which nodes they have passed on a previous iteration of tree-building (i.e. the tree in 1B). For example, to the 'cat' model are appended the features 0 and 2, indicating that the model traversed those nodes of the previous tree (1B). Querying these features is equivalent to querying multiple original features of the model simultaneously. At node 1 of 1C this is done, and results in a less fragmented tree than 1B. The procedure can be repeated, as in 1D: the models are renamed with the compound features found by traversing 1C, and reference to them leads in 1D to a final, perfect split of the data.

We use 5 iterations of this procedure for the final clustering of spectral parameters of system L-SER. In Table 3 it can be seen that the number of parameters estimated for L voices increases when serial tree building is introduced, approaching and in some cases surpassing the number of parameters estimated for the P voices. We suppose this to be a result of decreased under-fragmentation enabled by the discovery of compound features.

3. Evaluation

A web-based evaluation of the intelligibility of the voices built was conducted on Amazon's Mechanical Turk.[1] This is a web-based platform that allows short tasks requiring human intelligence to be posted and completed on the web for payment. Several language experiments have been reported that use the service (e.g. [18]). 40 listeners were obtained in this way to evaluate Semantically Unpredictable Sentences (SUS: [19]) synthesised by the systems. 40 such sentences were produced using each system, 20 of which where the content words were not to be found in the systems' training vocabulary (the OOV portion of the test-set), and the other 20 so that all the content words had been 'seen' by systems during training (the INV portion). Listeners were assigned to one of 4 groups (each with 10 listeners); the groups were designed so that each group's listeners heard a different set of system–sentences, but so that the same sentences were heard for each system over the whole test. SUS sentences were interspersed with four short natural samples of SLT's speech in order that the reliability of listeners' responses could be gauged; the responses to these samples were not used for evaluation of the systems. Stimuli were presented in random order to the listeners, and the listeners were asked to type

[1] https://www.mturk.com/mturk/

what they heard. Word error rates (WERs) were then computed on the listeners' responses, with reference to the text used to generate the nonsense sentences in the first place.

4. Results

Results of the evaluation are summarised in Figure 2. WERs are given over all test sentences (left), sentences with in-training-vocabulary content words only (middle), and sentences with out-of-training-vocabulary content words only (right). Differences between system WERs were compared in a pairwise fashion using the bootstrap procedure outlined in [20]: bootstrap-*t* confidence intervals were calculated over system differences. Differences found to be non-significant in this analysis (with α = 0.05 and Bonferroni correction) are indicated with arcs in the figures.

On both the INV portion of the test set (centre plot of Figure 2) and on the OOV portion (right-hand plot of same figure), the phoneme-based systems acheive lower WERs than the letter-based ones, as expected. For the INV set, the two phoneme-based systems receive the same WER as we would expect, as they are essentially the same system when producing this 'seen' vocabulary. On the OOV set, the limited-lexicon phoneme-based voice P-LIM has a higher WER than counterpart P-FUL, although this difference between the P voices is not found to be significant.

The serial tree-building method produces a significant improvement to the baseline letter-based system in both the overall evaluation (left-hand plot of Figure 2) and evaluation on the INV portion of the test-set (middle plot in same figure). Also on the OOV portion of the test-set (right-hand plot of Figure 2), L-SER achieves a lower WER than L-BAS, although in this case it is not found to be significant. In no case does performance of the L systems approach that of the full phoneme-based system, P-FUL. On the OOV test-set, though, the addition of serial tree-building allows the letter-based system to close a part of the gap in performance between the baseline system L-BAS and the phoneme-based system with limited lexicon, P-LIM. Here, although there remains a gap between L-SER and P-LIM, it is not found to be significant (though as noted above, neither is the gap in performance between L-BAS and L-SER in this case).

5. Conclusions

Our experiments have shown that, fairly obviously, it is beneficial to use phonemic representations when they are available to us. The improvement in WER obtained when serial tree building is introduced encourages us, however, in that it demonstrates that ways exist to improve on the baseline letter-based system without resorting to manually compiled resources such as lexicons and letter-to-sound rules. As noted at the beginning of this paper, English has an especially difficult orthography for this type of work, and we suspect that techniques like the ones presented here may, if developed, enable us to close the smaller gap between a baseline letter-based system and phoneme-based systems in languages with more regular letter-to-sound correspondences. This is planned for future work.

6. Acknowledgements

The authors would like to thank Karl B. Isaac for his generous help with setting up the online evaluation described in this paper.

This work has made use of the resources provided by the Edinburgh Compute and Data Facility (ECDF: http://www.ecdf.ed.ac.uk/). The ECDF is partially supported by the eDIKT initiative (http://www.edikt.org.uk).

7. References

[1] S. Young, J. Odell, and t. . Woodland, in *Proc. ARPA Human Language Technology Workshop*, Mar. 1994, pp. 307–312.

[2] K. Shinoda and T. Watanabe, "MDL-based context-dependent subword modeling for speech recognition," *Acoustical Science and Technology*, vol. 21, no. 2, pp. 79–86, 2000.

[3] A. Black and A. Font Llitjos, "Unit selection without a phoneme set," in *IEEE TTS Workshop 2002*, 2002.

[4] M. Killer, S. Stüker, and T. Schultz, "Grapheme based speech recognition," in *Proc. Eurospeech*, 2003, pp. 3141–3144.

[5] J. Kominek and A. Black, "The CMU Arctic speech databases," in *Proc. 5th ISCA speech synthesis workshop*, Pittsburgh, USA, Jun. 2004, pp. 223–224.

[6] *The Carnegie Mellon University Pronouncing Dictionary*. [Online]. Available: http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[7] 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.

[8] *Edinburgh Speech Tools Library*. [Online]. Available: http://www.cstr.ed.ac.uk/projects/speech_tools/manual-1.2.0

[9] K. Beulen and H. Ney, "Automatic question generation for decision tree based state tying," in *ICASSP '98*, vol. 2, May 1998, pp. 805–808 vol.2.

[10] R. Singh, B. Raj, and R. Stern, "Automatic clustering and generation of contextual questions for tied states in hidden Markov models," in *Proc. ICASSP '99*, vol. 1, Mar 1999, pp. 117–120.

[11] C. Chelba and R. Morton, "Mutual information phone clustering for decision tree induction," in *Proc. Int. Conf. on Spoken Language Processing 2002*, 2002.

[12] H. Zen, T. Toda, M. Nakamura, and K. Tokuda, "Details of Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005," *IEICE Trans. Inf. & Syst.*, vol. E90-D, no. 1, pp. 325–333, Jan. 2007.

[13] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Chapman and Hall, 1993.

[14] A. Black, K. Lenzo, and t. . Pagel, in *Proc. of the 3rd ESCA Workshop on Speech Synthesis*, 1998, pp. 77–80.

[15] Z.-J. Yan, Y. Qian, and F. K. Soong, "Rich context modeling for high quality HMM-based TTS," in *Proc. Interspeech*, Brighton, U.K., sep 2009, pp. 1755–1758.

[16] F. Questier, R. Put, D. Coomans, B. Walczak, and Y. V. Heyden, "The use of CART and multivariate regression trees for supervised and unsupervised feature selection," *Chemometrics and Intelligent Laboratory Systems*, vol. 76, no. 1, pp. 45–54, 2005.

[17] I. Shafran and M. Ostendorf, "Acoustic model clustering based on syllable structure," *Computer Speech & Language*, vol. 17, no. 4, pp. 311–328, 2003.

[18] M. I. Tietze, A. Winterboer, and J. D. Moore, "The effect of linguistic devices in information presentation messages on comprehension and recall," in *ENLG '09: Proceedings of the 12th European Workshop on Natural Language Generation*. Morristown, NJ, USA: Association for Computational Linguistics, 2009, pp. 114–117.

[19] C. Benoit, M. Grice, and V. Hazan, "The SUS test: A method for the assessment of text-to-speech synthesis intelligibility using Semantically Unpredictable Sentences," *Speech Communication*, vol. 18, no. 4, pp. 381 – 392, 1996.

[20] M. Bisani and H. Ney, "Bootstrap estimates for confidence intervals in ASR performance evaluation," in *Proc. ICASSP '04*, vol. 1, 2004, pp. 409–12.

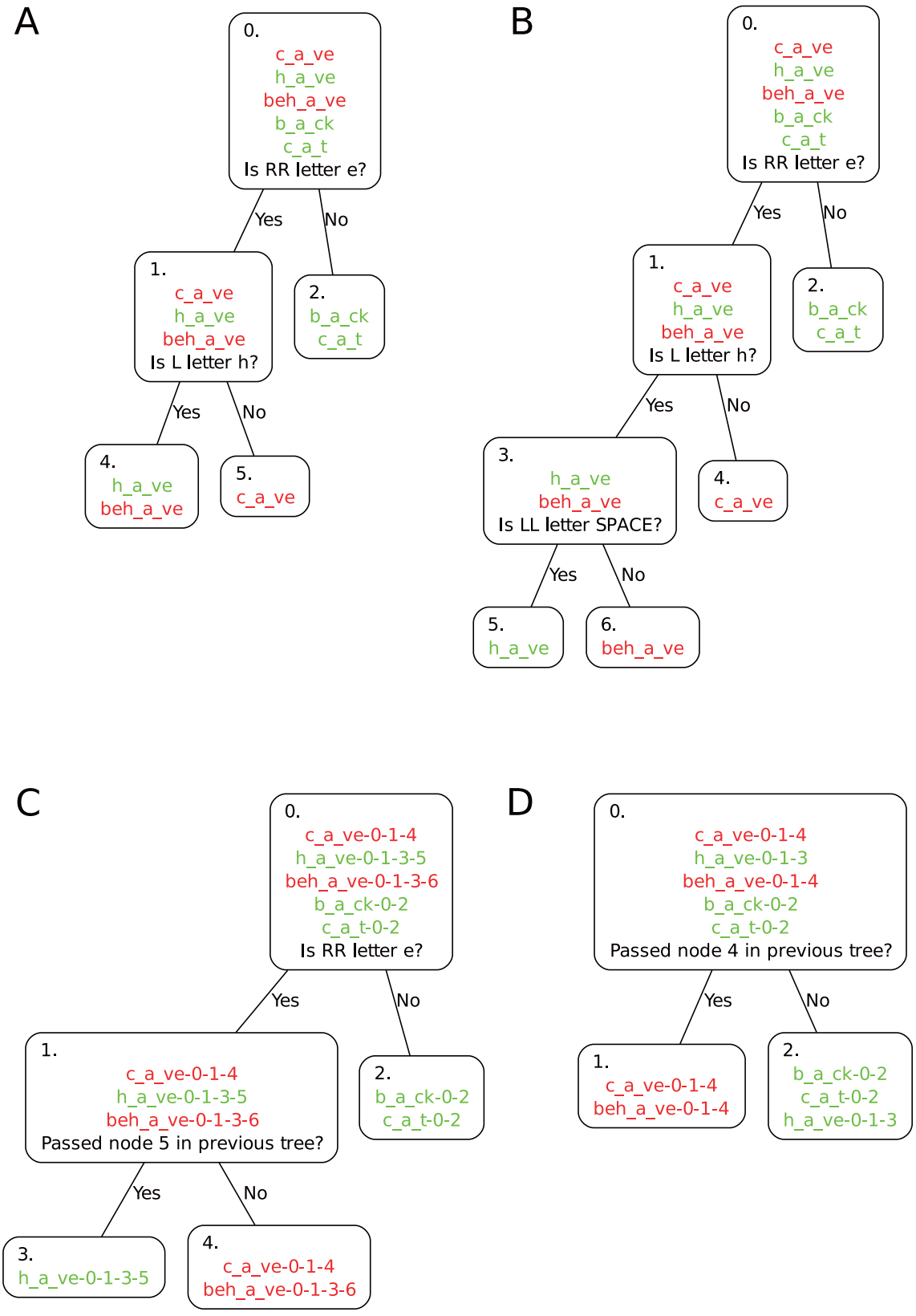

Figure 1: Serial tree building.

Table 3: Systems built: model sizes

System identifier	L-BAS	L-SER	P-{FUL,LIM}
No. leaf nodes (mcep)	479	577	619
No. leaf nodes (logF0)	2110	2431	2397
No. leaf nodes (bndap)	536	531	501
No. leaf nodes (duration)	940	1204	766
No. used questions (mcep)	102	366	204
No. used questions (logF0)	210	498	706
No. used questions (bndap)	118	265	195
No. used questions (duration)	176	379	377

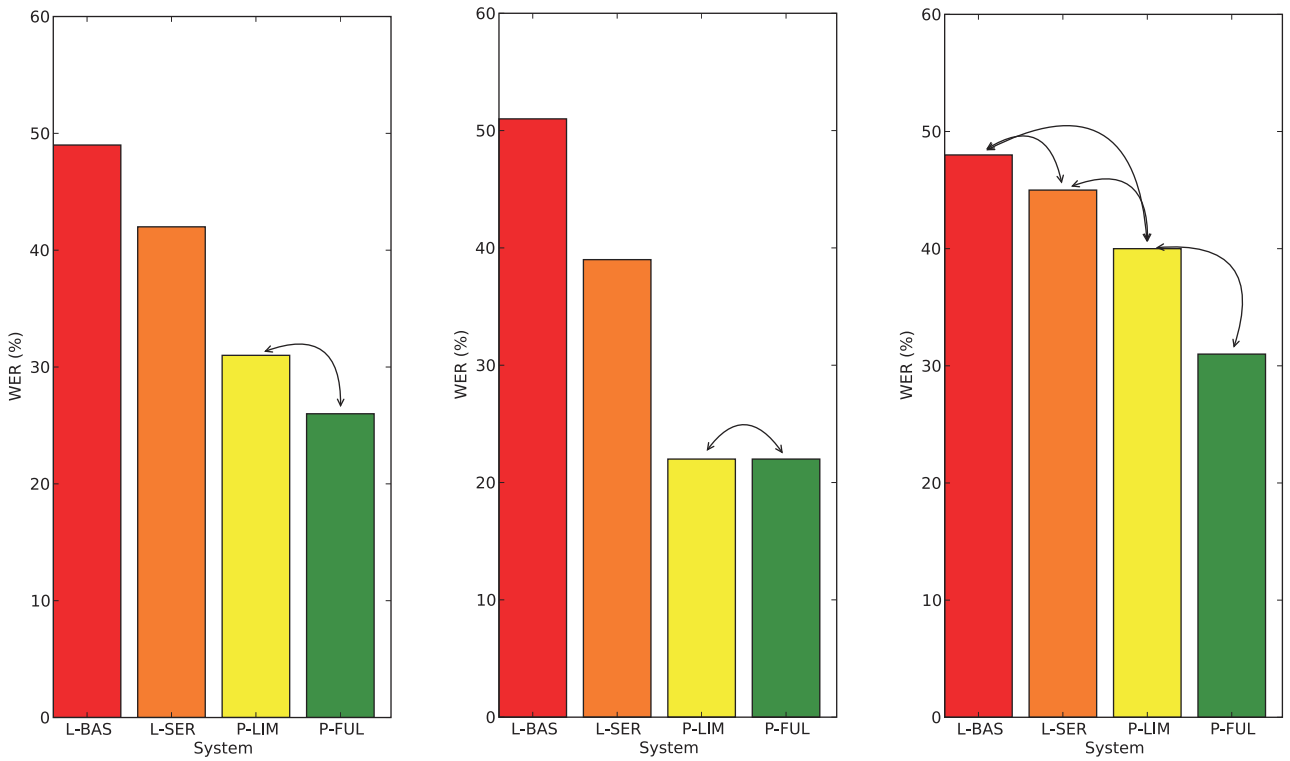

Figure 2: WER for all test sentences (left), sentences with in-training-vocabulary content words only (middle), and sentences with out-of-training-vocabulary content words only (right). Arcs show pairs of systems where bootstrap-*t* confidence intervals over system differences show no significant difference (with $\alpha = 0.05$ and Bonferroni correction).