Effort and Accuracy during Language Resource Generation: A Pronunciation Prediction Case Study

M. Davel and E. Barnard

Human Language Technologies Research Group
Meraka Institute, Pretoria, 0001
mdavel@csir.co.za, ebarnard@csir.co.za

Abstract

When developing a language resource, there is generally a trade-off between the amount of effort invested in the resource creation process and the quality of the resulting resource. We argue that, in the developing world with its many resource-scarce languages, a 'usable' resource in multiple languages may be more valuable than a highly accurate resource for one language only. From this perspective we investigate the resource validation process – determining whether a resource is sufficiently accurate – using the creation of a pronunciation dictionary as case study. We show that the amount of effort required to validate a 20,000-word pronunciation dictionary can be reduced substantially by employing appropriate computational tools, when compared to both a fully manual validation process and a competing automatic process.

1. Introduction

Speech and language technology development typically relies on the existence of extensive speech and language resources: comprehensive electronic word lists, annotated speech corpora, parallel texts, and so forth. Many of the languages in the developing world, however, can be classified as 'resource-scarce'; that is, for these languages limited or no language resources are available. This lack of appropriate language resources is a significant obstacle in realising the potential of speech and language technologies in the developing world.

When developing language resources, there is a trade-off between the *volume of resources* that can be developed with a given amount of effort invested in the resource creation process and the *quality* of the resulting resource. For environments where multiple languages are prevalent, a number of questions arise: How much effort should be invested in resource development for a single language? When is a language resource 'sufficiently accurate'? We argue that usable resources in many languages contribute more value towards development goals than highly accurate resources in a single language. Towards this end, expensive techniques that are appropriate for the development of highly accurate resources in the developed world may not be as appropriate within a developing-world context.

The efficient development of 'usable' resources can be seen to consist of two components: (1) ensuring the efficiency of the resource creation process, and (2) ensuring the efficiency of the resource validation process. The generic resource creation process can be made more efficient through techniques such as bootstrapping and cross-language utilisation of language resources [1, 2]. In this paper we focus on the resource validation process: using automated techniques to identify outliers (and therefore potential errors), manually verifying the flagged portions of the resource and correcting errors found, and finally, manually verifying a further portion of the resource in order to estimate its current accuracy.

We apply this general approach to the task of developing pronunciation dictionaries. We demonstrate how the validation process can be used to ensure a resource that is highly usable but created at less cost than an optimally accurate version of the same resource.

The paper is structured as follows: In section 2 we provide background with regard to the pronunciation prediction task and the various tools utilised in this study. In section 3 we discuss the dictionary validation process that results in the development of a 'usable' resource. In section 4 we evaluate the effectiveness of the validation strategy and compare this approach with an alternative approach for the development of an optimally accurate version of the same resource. Section 5 contains some concluding remarks.

2. Background

A pronunciation dictionary provides a mapping between the written (orthographic) form of a word and its pronunciation, typically specified in terms of a series of phonemes. This resource is a core component of text-to-speech and automatic speech recognition systems. It has previously been shown [2, 3] that pronunciation dictionaries can be developed efficiently using bootstrapping. Bootstrapping systems utilise automated techniques to extract grapheme-to-phoneme prediction rules from an existing dictionary and apply these rules to predict additional entries, typically in an iterative fashion. Predicted entries are verified and – if necessary – corrected by a human verifier before being added to the dictionary. Upon completion, the final pronunciation dictionary is used to extract a set of grapheme-to-phoneme rules that can be used to deal with out-of-vocabulary words in speech processing systems.

A variety of techniques are available for the extraction of grapheme-to-phoneme prediction rules from pre-existing dictionaries. Approaches include decision trees [4], pronunciation-by-analogy models [5], instance-based learning algorithms [6, 7] and the algorithm used in this study: *Default&Refine* [8]. The *Default&Refine* algorithm is very competitive in terms of both learning efficiency (that is, the accuracy achieved with a limited number of training examples) and asymptotic accuracy, when compared to alternative approaches [8].

A *Default&Refine* rule set is extracted in a straightforward fashion: for every letter (grapheme), a default phoneme is derived as the phoneme to which the letter is most likely to map. 'Exceptional' cases – words for which the expected phoneme is not correct – are handled as refinements. The smallest pos-

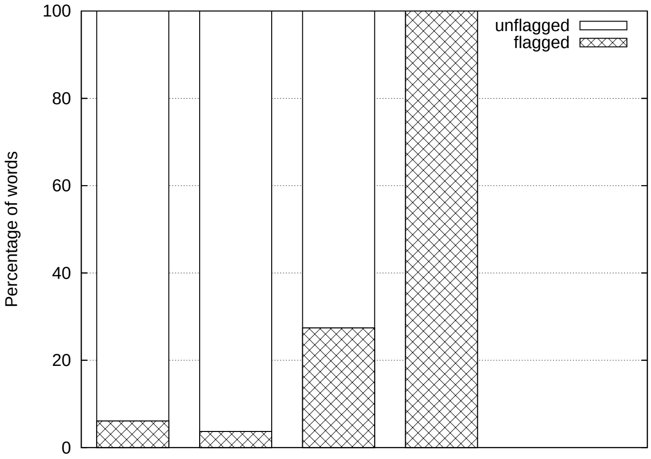

Figure 1: *Estimated percentage of all words, correct words, variants and actual errors flagged as possible errors during validation.*

sible context of letters that can be associated with the correct phoneme is extracted as a refined rule. Exceptions to this refined rule are similarly represented by further refinements, and so forth, leading to a rule set that describes the training set with complete accuracy. Further details can be found in [8]. The *Default&Refine* algorithm has been integrated into our pronunciation dictionary bootstrapping system *DictionaryMaker* [9], which is used in this study both during dictionary creation and dictionary validation, as described further in Section 3.

3. Dictionary validation

In order to analyse the effectiveness of the resource validation process for pronunciation dictionaries we use an approximately 20,000-word Afrikaans dictionary created through bootstrapping using the *DictionaryMaker* system. The system was initialised with a pre-existing 7,782-word dictionary, and 12,422 new words were added through a bootstrapping session.

To validate the accuracy of the new dictionary, we use a technique described in [10] to identify possible errors: The rules extracted by *Default&Refine* are ordered according to generality, with rules that describe a large set of words earlier in the rule list, and rules that describe fewer words later in the rule list. Since errors are likely to result in rules that are applicable to only a few words, these 'specialised' rules may be used to indicate possible errors. In [10] this process was tested using an artificially corrupted dictionary. In this study we apply the technique to a dictionary containing accidental errors introduced during an actual dictionary creation process.

The dictionary validation process consists of the following steps:

- A *Default&Refine* rule set is extracted from the dictionary, and every rule generated by a single word is identified. (Typically, a set of words create a rule.) The words associated with this set of specialised rules are flagged as potential errors, and referred to as the *flagged* set in the rest of this paper.
- A *control* set of 200 words are selected at random from the unflagged section of the dictionary. These words are selected to have the same distribution of word lengths as the words occurring in the *flagged* set.
- Both the *flagged* and *control* sets are manually verified by a linguist. Words are marked as 'invalid' if the word itself is invalid, 'error' if the word is valid but the pronunciation is erroneous, 'variant' if more than one pronunciation is acceptable, and 'correct' if the single correct pronunciation was given.
- Errors found in the previous step are corrected in the dictionary.
- The results of the validation of the *control* set are used to estimate the accuracy of the resulting dictionary.

Table 1: *Number of errors and variants found during validation.*

Number of words in dictionary	20,204
Number of words flagged	1,238
Errors found in flagged set	97
Variants found in flagged set	457
Errors found in control set	0
Variants found in control set	13

Table 2: *Distribution of errors and variants among sets of words generating different numbers of exceptional rules.*

	total	errors	%errors	variants	%variants
4 or more	8	1	12.5	5	62.5
3 or more	40	3	7.5	19	47.5
2 or more	189	22	11.6	85	45.0
1 or more	1238	97	7.8	457	36.9

4. Results

4.1. Validation process

When the *flagged* set was verified, it was found that it contained a number of errors but also a large number of variants (words that have multiple pronunciations, only one of which was included in the dictionary previously). For example, the Afrikaans word 'vertikaal' may be pronounced as both */v eh r t iy k aa l/* and */f eh r t iy k aa l/* by different speakers (using ARPABET symbols). If the dictionary creator were not consistent with the choice of variant for similar words (such as 'vertikaal' and 'vertikale') the rule extraction process would need to extract highly specialised rules in order to accommodate these words, which would then flag these variants as potential errors.

Table 1 contains a summary of the number of errors and variants found in both the *flagged* and *control* sets. Using the results from the *control* set we can estimate how many additional variants and errors have not yet been found in the dictionary: 6.5% variants and 0% errors (an optimistic estimate given the size of the control set.) The effectiveness of the automated error detection process is more clearly illustrated in Figure 1. As can be seen, only a small percentage of all words have to be validated, while the majority of errors are found in this small subset of *flagged* words.

How efficient is the proposed process? Is there a smaller subset that can be evaluated while still finding a similar percentage of errors? We consider two alternative approaches to identifying such a flagged subset of words.

Firstly, we consider the ordering of words according to the number of specialised rules generated. Table 2 indicates the number of words creating 4 or more, 3 or more, 2 or more and 1 or more specialised rules, and shows how both variants

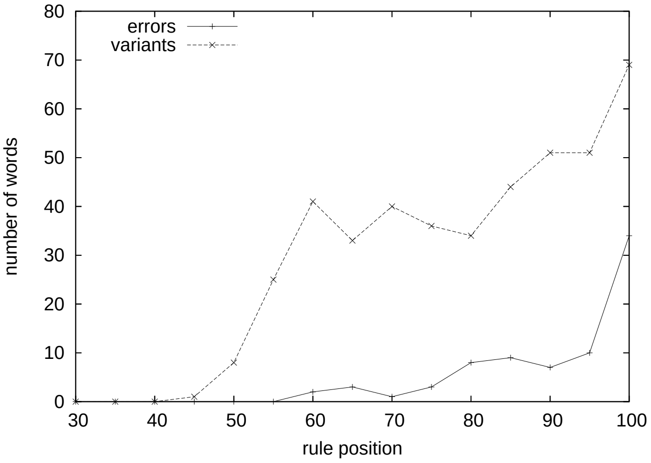

Figure 2: *The position of rules associated with identified errors and variants in the extracted rule list.*

and errors are distributed among these sets. While the smaller flagged sets (such as the '4 or more' set) contains a larger proportion of errors and variants, the full flagged set (indicated by '1 or more') needs to be analysed in order to find a significant percentage of the errors and variants occuring in the full set of pronunciations.

Secondly, we consider where in the rule set the specialised rules that are associated with incorrect words occur. If these rules occur late enough in the rule set, all rules after a threshold may be used to flag possible errors. For each word we find all the rules generated by that specific word (one per grapheme) and select the rule that occurs latest in the rule set as the index rule for that word. In Figure 2 we plot the number of identified errors and variants according to where their index rules occur in the rule set. Note that this is not the full set of errors and variants but only those identified during the validation process. As can be seen from the figure, a significantly larger percentage of rules would need to be validated if specialised words were flagged according to this approach.

In both cases there is not a smaller set of words that can be used for error detection: the original set of words generated by exceptional rules as described in Section 3 is indeed the best set to consider during validation.

4.2. Comparing approaches

In this section we compare the validation process described in the previous sections with an alternative approach, where the full dictionary is created by more than one developer, and the resulting dictionaries compared for consistency.

In order to obtain an estimate for the time taken to create a dictionary, we measure the effectiveness of the dictionary creation process while bootstrapping the 20,204-word dictionary, as described earlier. The dictionary creation process using the *DictionaryMaker* tool is very efficient. The linguist using the system was measured during normal operation which included a number of breaks. (The linguist was asked to concentrate on accuracy rather than speed.) If words that took longer than 30s to correct are excluded from measurements (since these typically indicate breaks), the average speed with which a word was added was 3.9s per word. Figure 3 shows the number of words corrected in time t where t ranges from 0 seconds to 30 seconds. The distribution of correction times shows that the majority of words take only a second or two to correct. While there are few words that take longer than 30s to correct, these numbers are not negligible, as shown in Figure 4. A correction time of 3.9s therefore provides an optimistic lower bound on manual correction time during normal operation.

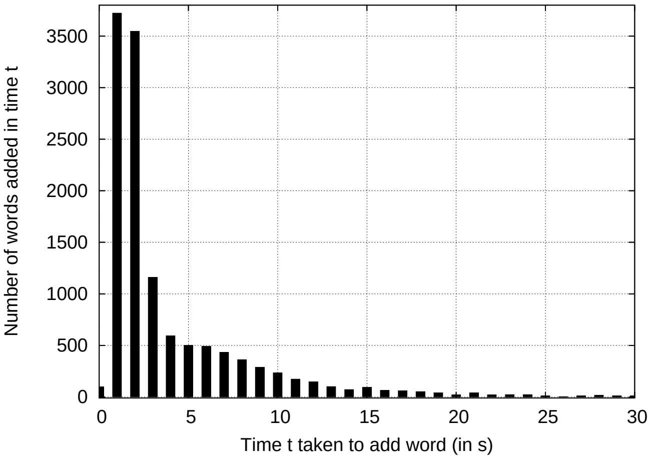

Figure 3: *Number of words that took time t each to correct.*

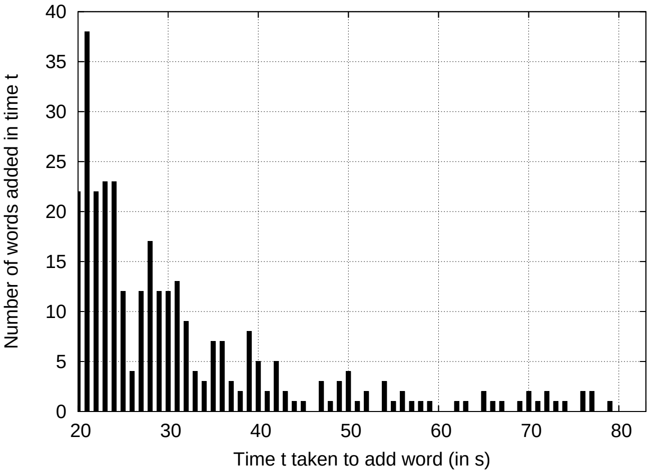

Figure 4: *More detailed view of the tail of Figure 3*

Using 3.9s as an optimistic lower bound, and assuming that a dictionary developer requires a 10-minute break after 20 minutes of validation, we find that a 20,204-word dictionary would take at least 32.8 man hours to validate manually (65.6 hours if two validators are required). This is in comparison with the 2 hours required to validate the smaller subset. Also, since developers tend to make similar mistakes, a validation process as described here can be useful, even if a full validation (redevelopment by different dictionary developers) is also implemented.

5. Conclusion

In this paper we have demonstrated an efficient technique for the validation of a pronunciation dictionary. Efficient validation techniques are important for the development of language resources that are 'usable', even though they may not be optimally accurate. As long as the resource is sufficiently accurate in order to be utilised in speech and language systems, further errors may be found and corrected during utilisation, leading to an increasingly accurate resource over time. This may be of particular importance in the developing world where language resources are scarce and the means to develop new resources constrained.

In this work, the dictionary created through our more efficient evaluation process is expected to be comparable in quality to the dictionary that would be obtained with the labour-intensive manual procedure described in section 4.2, since no erroneous words were found in the unflagged set of test words. However, this beneficial state of affairs will probably be the exception rather than the rule in resource development. That is, one expects that efficient creation and evaluation of resources will typically come at some cost in quality of systems developed using those resources. (For example, using automatic rather than manual alignments in the development of speech synthesizers, or using reduced quantities of training data for speech recognizers, will generally degrade the quality of the resulting system.) The development of a theoretical model which allows one to evaluate the impact of such a trade-off is a crucial topic for further investigation, since such a model is required to develop an appropriate allocation strategy for resource development.

6. References

[1] T. Schultz and A. Waibel, "Language-independent and language-adaptive acoustic modeling for speech recognition," *Speech Communication*, vol. 35, pp. 31–51, Aug. 2001.

[2] M. Davel and E. Barnard, "Bootstrapping for language resource generation," in *Proceedings of PRASA*, South Africa, November 2003, pp. 97–100.

[3] S. Maskey, L. Tomokiyo, and A.Black, "Bootstrapping phonetic lexicons for new languages," in *Proceedings of Interspeech*, Jeju, Korea, October 2004, pp. 69–72.

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

[5] F. Yvon, "Grapheme-to-phoneme conversion using multiple unbounded overlapping chunks," in *Proceedings of NeMLaP*, Ankara, Turkey, 1996, pp. 218–228.

[6] K. Torkkola, "An efficient way to learn English grapheme-to-phoneme rules automatically," in *Proceedings of ICASSP*, Minneapolis, USA, April 1993, vol. 2, pp. 199–202.

[7] W. Daelemans, A. van den Bosch, and J. Zavrel, "Forgetting exceptions is harmful in language learning," *Machine Learning*, vol. 34, no. 1-3, pp. 11–41, 1999.

[8] M. Davel and E.Barnard, "A default-and-refinement approach to pronunciation prediction," in *Proceedings of PRASA*, South Africa, November 2004, pp. 119–123.

[9] "Dictionarymaker user manual, version 2.0 (i)," September 2006, http://dictionarymaker.sourceforge.net/.

[10] M. Davel and E. Barnard, "Bootstrapping pronunciation dictionaries: practical issues," in *Proceedings of Interspeech*, Lisboa, Portugal, September 2005, pp. 1561–1564.