CHAPTER THREE

BOOTSTRAPPING MODEL

3.1 INTRODUCTION

In this chapter we sketch a basic framework for the analysis of the bootstrapping process. We describe the bootstrapping model in Section 3.2, and discuss the factors to consider when evaluating the efficiency of the bootstrapping process in Section 3.3. In Section 3.4 we show how this model applies to the pronunciation modelling task in particular.

3.2 MODEL DESCRIPTION

As introduced in Section 2.3, we use the term ‘bootstrapping’ to describe *an iterative process whereby a model is improved via a controlled series of increments, at each stage utilising the previous model to generate the next one*. During bootstrapping the model is grown systematically, becoming increasingly accurate from one increment to the next. When analysing the bootstrapping process, it soon becomes apparent that the process relies on an automated or semi-automated mechanism to convert among various representations of the model considered. Each representation describes the same task in a format that provides a specific benefit: either because the representation is amenable to automated modelling and analysis, or because it describes the current model in a way that is convenient for a human to verify and improve. The remainder of this section contains a definition of the various components of a bootstrapping system, a description of the bootstrapping process, and examples of bootstrapping applications.

3.2.1 COMPONENTS

The general bootstrapping concept utilising two model representations is depicted in Figure 3.1. The number of representations is limited to two for the sake of simplicity – three or more representations can also be included in the model.

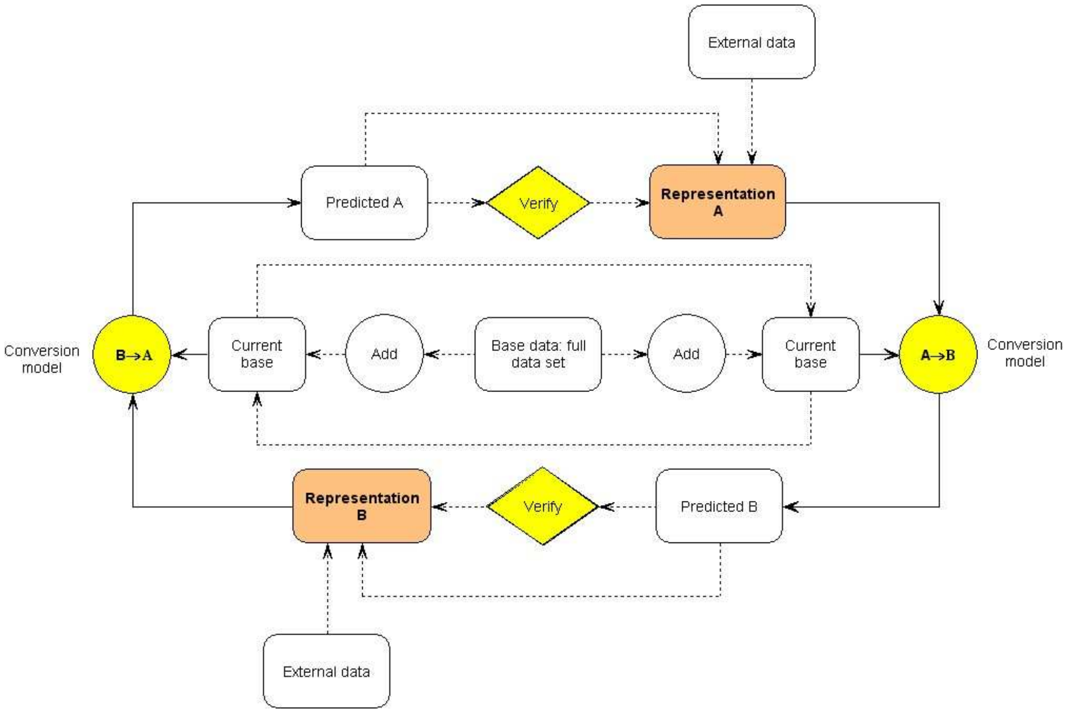

Figure 3.1: *General bootstrapping concept, utilising two model representations.*

The following components play a role during bootstrapping:

- *Alternative representations:* Two or more representations of the same model lie at the heart of the bootstrapping process. In the Fig. 3.1 these are indicated as *A* and *B*.
- *Conversion mechanisms:* Each conversion mechanism (indicated as $A \rightarrow B$ and $B \rightarrow A$) provides an automated or semi-automated means to convert data from one representation to another.
- *Verification mechanisms:* Once converted to a specific representation, the model can be improved via automated or human (manual) verification, indicated in the figure by the *Verify* components.
- *Base data:* This term is used to refer to the domain of the model. The *current base* indicates the domain that has been used in training the current model, and consists of a subset of, or the

full base data set. The current base data is implicitly or explicitly included in each of the two representations.

- *Increment mechanisms:* The *Add* components are used to increase the current base during bootstrapping. At the one extreme, all model instances can be included in a single increment; at the other, a single instance can be added per bootstrapping cycle. The increment mechanisms may utilise active learning techniques [69, 70] in order to select an appropriate set of instances to add.
- *External data:* This term refers to additional data sources that are utilised during bootstrapping. Typically, external data is used to initialise a bootstrapping system with models that were developed on a related task.

3.2.2 PROCESS

Prior to bootstrapping, the various representations are initialised in preparation for the first iteration. Typically only a single representation requires initialisation (A in this instance). External data may be included in this process, or the bootstrapping process starts without any initial knowledge of the task not included in the base data. The increment mechanism chooses the first base set to use. Once initialised, the bootstrapping process consists of the following steps, many of which are optional, as indicated:

1. The current base, as well as the current representation A is used to generate the next representation B.
2. B is verified, either manually or automatically. (Optional)
3. Based on the current state of the bootstrapping system, the increment mechanism increases the current base set. (Optional if (6) is not)
4. The current base, as well as the current representation B is used to generate representation A.
5. A is verified, either manually or automatically. (Optional)
6. Based on the current state of the bootstrapping system, the increment mechanism increases the current base set. (Optional if (3) is not)

This cycle is repeated until a sufficiently accurate and/or comprehensive model is obtained.

3.2.3 EXAMPLES

Two typical examples of bootstrapping are illustrated in Figures 3.2 and 3.3. The first example (Fig. 3.2) illustrates the automated bootstrapping scenario described in Section 1.2. For this task, the base data consists of audio data and phonemic transcriptions (initially not aligned with the audio

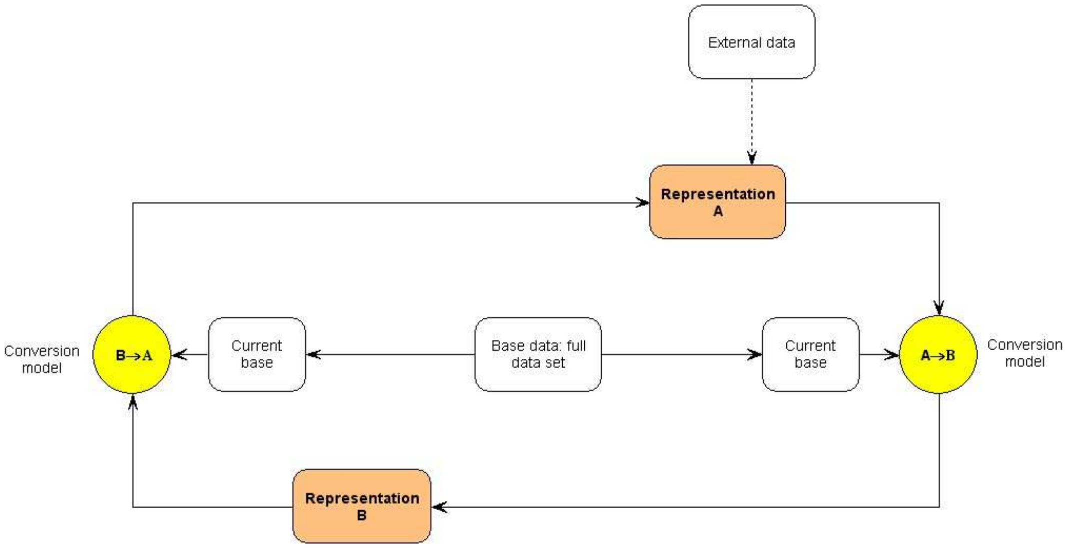

Figure 3.2: *An example of automated bootstrapping.*

data). A represents the phonemic segmentation of the audio data, and B the acoustic models derived from the segmentations. The focus is on the refinement of the acoustic models: the segmentations themselves are only important to the extent that they influence the quality of the acoustic models. The $A \rightarrow B$ mechanism consists of the training, re-clustering, and re-training of acoustic models, and the $B \rightarrow A$ mechanism of automatic Viterbi alignment of the phonemic transcriptions, utilising the current acoustic models.

The second example (Fig. 3.3) illustrates a simple bootstrapping scenario where machine learning and human intervention are combined, as would be the case, for example, when bootstrapping audio segmentations for Text-to-Speech purposes. The base data again consists of audio data and phonemic transcriptions; A represents the human-readable segmentation of the audio data, and B the acoustic models derived from the segmentations. The $A \rightarrow B$ mechanism consists of acoustic model training, and the $B \rightarrow A$ mechanism of automatic alignment. Here the focus is on achieving optimal segmentations and these are hand-verified until the acoustic models are stable enough to support accurate alignments (and possibly even after that, if high quality segmentations are required).

3.3 EFFICIENCY OF BOOTSTRAPPING PROCESS

The main aim of a bootstrapping system is to obtain as accurate a model as possible from available data. When human intervention is used to supplement or create the training data itself, the aim shifts towards *minimising the amount of human effort required during the process*. This is the focus of our

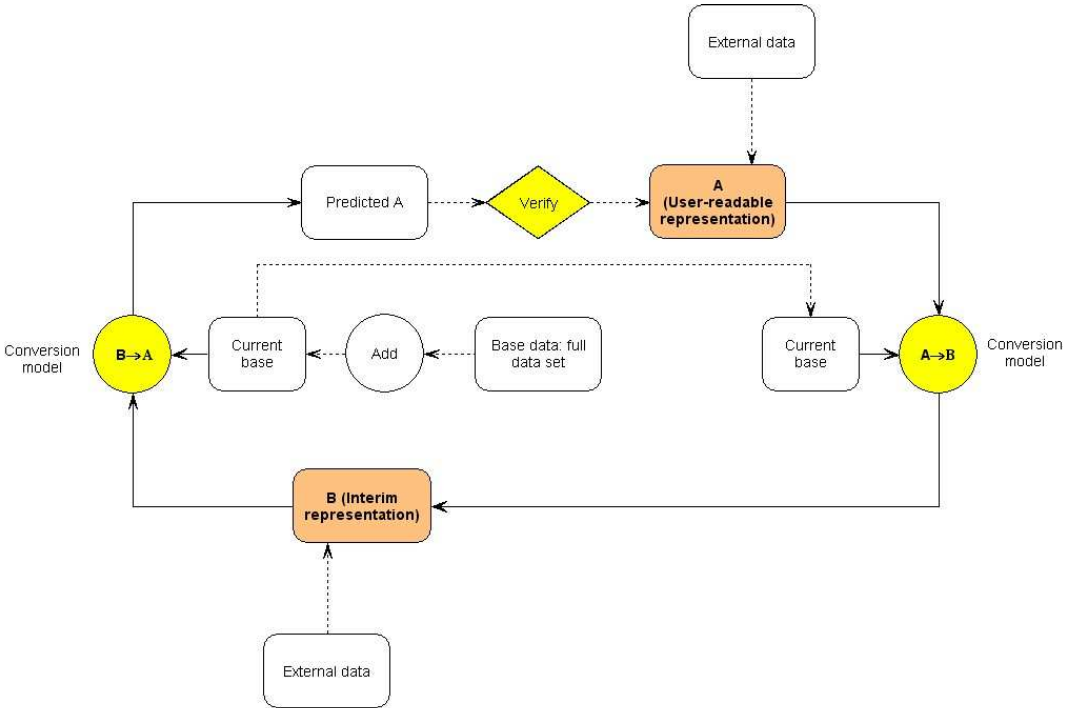

Figure 3.3: *An example of bootstrapping where machine learning and human intervention are combined.*

analysis, and we therefore measure bootstrapping efficiency as a function of model accuracy:

$$efficiency(a) = \frac{t_{bootstrap}(a)}{t_{manual}(a)} \tag{3.1}$$

where a is the accuracy of the current model as measured against an independent test set and $t_{bootstrap}(a)$ and $t_{manual}(a)$ specify the time (measured according to amount of human intervention) required to develop a model of accuracy a with and without bootstrapping respectively.

Bootstrapping is analysed according to bootstrapping cycles. While bootstrapping, all base instances do not result in valid data that can be included in the model training process. Of the instances that define valid base data, some will be correctly represented by the initial representation (B), and others will contain errors. We define a number of variables to assist us in the analysis of these instances: At the start of cycle x of the bootstrapping process, we define $n(x)$ as the number of instances included in the current base, $n_{invalid}(x)$ as the number of instances that are invalid, $n_{correct}(x)$ as the number of instances that are valid and correct, and $n_{error}(x)$ as the number of instances that are valid and incorrect. For these variables, the following will always hold:

$$\begin{aligned} n(x) &= n_{invalid}(x) + n_{valid}(x) \\ n_{valid}(x) &= n_{correct}(x) + n_{error}(x) \end{aligned} \tag{3.2}$$

Related incremental variables are used to represent the increase during cycle x, namely $inc\_n(x)$, $inc\_n_{invalid}(x)$, $inc\_n_{valid}(x)$, $inc\_n_{correct}(x)$ and $inc\_n_{error}(x)$. The same intervention mechanism may have different cost implications based on the $status$ of the instance. In the simplest case, the status of an instance may simply be correct, incorrect or invalid, but subtler differences are possible, e.g. the number of changes required to move from an incorrect to a correct version. The expected status of a newly predicted instance changes as the system becomes more accurate. Prior to human intervention at stage x of the bootstrapping process, the number of instances of each status within the current increment is given by:

$$inc\_n(x) = \sum_{s \in status} inc\_n(s, x) \tag{3.3}$$

Combining machine learning and human intervention in a way that minimises the amount of human effort required during the process can be achieved in two ways: (a) by minimising the effort required by the human verifier to identify errors accurately, and (b) by optimising the speed and accuracy with which the system learns from the human input. This section describes the various factors that influence the efficiency of the bootstrapping process from both these perspectives.

3.3.1 HUMAN FACTORS

The first human factor that impacts on the efficiency of the bootstrapping process relates to *required user expertise:* whether the task requires expert skills, or whether a limited amount of task-directed training is sufficient. If is assumed that the user has the skills required, the following measurements provide an indication of the efficiency of the bootstrapping process for a specific user:

- *User learning curve:* The time it takes for a specific user to become fully proficient using the bootstrapping system. Measured as t_{train}, initial training data is assumed to be discarded.
- *Cost of intervention:* The average amount of user time required per intervention i when an instance is in status s, for a fully trained user using the bootstrapping system. Measured as $t_{verify}(i, s)$ a different average cost may be associated with different types of interventions. If more than one intervention is used to generate a single instance during one cycle of bootstrapping, the combination of mechanisms is modelled as an additional (single) mechanism. Depending on the bootstrapping process, it may be more realistic to measure this value for a set of instances.
- *Task difficulty:* The average number of errors for a fully trained user using the bootstrapping system. Indicated by $error\_rate_{bootstrap}(i, s)$, this is measured in percentage as the average number of errors per 100 instances generated using intervention mechanism i to verify an instance initially in state s.

- *Quality and cost of user verification mechanisms:* Implicit in the above two measurements are the cost and effect on error-rate of additional assistance provided during user intervention. Rather than modelling additional user assistance provided during existing interventions separately, the combined intervention is again modelled as an additional type of intervention. In the same way, automated verification mechanisms are modelled as additional interventions.

- *Difficulty of manual task:* The average number of errors for a fully trained user developing instances manually. Indicated by $error\_rate_{manual}$, this is measured in percentage as the average number of errors per 100 manual instances developed, where each manual instance can be associated with an individual base data instance in the bootstrapped system.

- *Manual development speed:* The average amount of time per instance development for a fully trained user performing this task manually, measured as $t_{develop}$; this value can also be analysed separately per types of instance development as $t_{develop}(s)$, if so required.

- *Initial set-up cost:* The time it takes for a user to prepare the initial system for manual development or bootstrapping; measured in time as t_{setup_manual} and $t_{setup\_bootstrap}$ respectively.

3.3.2 MACHINE LEARNING FACTORS

The faster a system learns between verification cycles, the fewer corrections are required from a human verifier, and the more efficient the bootstrapping process becomes. From a machine learning perspective, learning speed and accuracy are directly influenced by:

- *Predictive accuracy of current base:* modelled as the expected number of instances of each status at a specific cycle of the bootstrapping process, and indicated by $E(inc\_n(s,x))$. Implicit to this measurement are four factors:
  - *Accuracy of representations:* The ability of the chosen representations to model the specific task.
  - *Set sampling ability:* The ability to identify the the next ‘best’ instance or instances to add to the knowledge base, possibly utilising active learning techniques.
  - *System continuity:* The speed at which the system updates its knowledge base. This has a significant effect on system responsiveness, especially during the initial stages of bootstrapping.
  - *Robustness to human error:* The stability of the conversion mechanisms and chosen representations in the presence of noise introduced by human error.

- *On-line conversion speed:* Any additional time costs introduced when computation is performed while a human verifier is required to be present (but idle while waiting for the computation to complete); measured as an average per number of valid instances developed and indicated by $t_{idle}(n)$.

- *Quality and cost of verification mechanisms:* The average amount of time required to utilise additional assistance mechanism j – from a computational perspective – when an instance is in status s, measured as $t_{auto}(j, s)$.
- *Validity of base data:* Using invalid data slows the bootstrapping process, especially if human intervention is required to verify the validity of base data; measured in % of base data, this is indicated by $valid\_ratio$.

Two additional factors that are not included explicitly in the general model, but can be included based on the requirements of the specific bootstrapping task, are:

- *Conversion accuracy:* The ability of the conversion to model convert between representations without loss of accuracy.
- *Effect of incorporating additional data sources:* The ability of the system to boost accuracy by incorporating external data sources at appropriate times.

3.3.3 SYSTEM ANALYSIS

The combined effect of the machine learning factors and human factors provide an indication of the expected cost of using the bootstrapping system. The time to develop a bootstrapping model via N cycles of bootstrapping, utilising a set of interventions I, is given by:

$$\begin{aligned} t_{bootstrap}(N, I) &= t_{setup\_bootstrap} + t_{train} + t_{iterate}(N, I) \\ &= t_{setup\_bootstrap} + t_{train} \\ &\quad + \sum_{x=1}^{N-1} \left(\sum_{i \in I} \sum_{s \in status} (t_{verify}(s, i) + t_{auto}(s, i)) * inc\_n(s, x) \right. \\ &\quad \left. + t_{idle} * inc\_n_{valid}(x + 1) \right) \end{aligned} \tag{3.4}$$

where $t_{iterate}(N, I)$ combines the cost of the various iterations, excluding the cost associated with system setup and user training. The expected value of $inc\_n(s, x)$ depends on the specific conversion mechanism, and is influenced by $valid\_ratio$ and $error\_rate_{bootstrap}(i, s)$.

This cost of bootstrapping can be compared to the expected cost of developing n_{manual} instances via a manual process:

$$t_{manual} = t_{setup\_manual} + t_{develop} * n_{manual} \tag{3.5}$$

If $n_{bootstrap}$ and n_{manual} are chosen such that

$$E[inc\_n(correct, n_{bootstrap})] = E[inc\_n(correct, n_{manual})] \tag{3.6}$$

where the number of valid instances generated during bootstrapping is given by:

$$n_{bootstrap} = \sum_{x=1}^{N-1} inc\_n(valid, x) \tag{3.7}$$

the accuracy of each of the two systems is approximately equivalent, and the values of eq. 3.4 and 3.5 can be combined according to eq. 3.1 in order to obtain a measure of the expected efficiency of the bootstrapping process. We use this measure to analyse a specific bootstrapping system in Chapter 6.

3.4 BOOTSTRAPPING PRONUNCIATION MODELS

The scenario depicted in Fig. 3.3 can be applied to the bootstrapping of pronunciation models. In this case, the base data consists of a word list; A represents an explicit pronunciation dictionary, each instance consisting of a word and pronunciation pair; and B represents a set of grapheme-to-phoneme rules. The $A \rightarrow B$ mechanism represents grapheme-to-phoneme rule extraction, and the $B \rightarrow A$ mechanism grapheme-to-phoneme conversion. Additional verification assistance that can be provided include automated error detection, and audio support during verification.

3.4.1 ALGORITHMIC REQUIREMENTS

An appropriate grapheme-to-phoneme rule extraction and conversion mechanism lies at the heart of the bootstrapping process. From the discussion in 3.3.2 it follows that the following are the most important requirements for a grapheme-to-phoneme formalism to be used in bootstrapping:

1. It should have high predictive ability, even for very small training set sizes.
2. It should be able to represent the word/pronunciation data exactly (in order to prevent conversion loss when switching between representations).
3. It should allow continuous model updating at a low computational cost.
4. Pronunciation prediction should be fast.
5. It should be robust to noise in the training data.

3.5 CONCLUSION

In this chapter we defined a framework and terminology for the analysis of a bootstrapping system. We showed how this model applies to the bootstrapping of pronunciation models and defined the requirements for a grapheme-to-phoneme conversion mechanism suitable for bootstrapping. These requirements are taken into account in the next chapter (Chapter 4) in the search for such a mechanism. The bootstrapping topic itself is revisited in Chapter 6.