Chapter 5
The Bootstrapping Approach to Developing Reinforcement Learning-based Strategies

Fig. 5.1 Baron Münchhausen escaping from a swamp by pulling himself up by his own hair; after a novel by Raspe (1785), illustration by Hosemann (1840) – The first ‘bootstrapping’

This Chapter motivates and introduces a procedural method for automatic strategy learning from Wizard-of-Oz (WOZ) data. Note that this Chapter presents the

V. Rieser and O. Lemon, *Reinforcement Learning for Adaptive Dialogue Systems*,
Theory and Applications of Natural Language Processing, DOI 10.1007/978-3-642-24942-6_5,

approach on a conceptual level, while the concrete practical realisation of the individual steps will be further elaborated in the subsequent chapters.

It should also be noted that the presented steps are not unique to the method introduced in this book, but most of them are required for any simulation-driven approach to strategy learning (though the last step is specific to our method). A contribution of this book is that all these steps are now performed starting with a limited WOZ data set, and that specific methods are introduced to build and validate the obtained simulations.

5.1 Motivation

Statistical learning approaches, such as Reinforcement Learning (RL), for Spoken Dialogue Systems offer several potential advantages over the standard rule-based hand-coding approach to dialogue systems development (as further explained in Chapter 3.1): a data-driven development cycle, provably optimal action policies, a precise mathematical model for action selection, possibilities for generalisation to unseen states, and automatic optimisation of competing trade-offs in the objective function.

In the previous Chapter we showed that RL-based strategies outperform hand-coded strategies with manually tuned thresholds for a wide range of application scenarios. One of the major strengths of RL-based strategies is that they can "intelligently" adapt their strategies to the (local) representation of the dialogue environment in order to satisfy an overall objective. Thus, the correct representation of the learning environment is one of the key challenges for this framework, and data-driven methods to construct such an environment should be preferred over hand-crafting (as done for the proof-of-concept study in Chapter 4).

One of the major limitations of this approach is that it relies on a large quantity of data being available. In cases when a fixed data set is used for learning, e.g. (Henderson et al, 2008; Singh et al, 2002; Walker, 2000), the optimal policy can only be discovered when it is present in the data set. [1] To overcome this problem, simulated learning environments are being used to explore optimal policies which were previously unseen in the data, e.g. (Ai et al, 2007b; Eckert et al, 1997; Young et al, 2009). However, several aspects of the components of these simulated environments are usually hand-crafted, and thus limit the scope of policy learning. In particular, the optimization (or reward) function is often manually set (Paek, 2006). In order to build simulation components from real data, annotated in-domain dialogue corpora have to be available, which explore a range of dialogue management decisions. Collecting dialogue data without a working prototype is problematic, leaving the developer with a classic "chicken-and-egg" problem. We therefore propose to learn

[1] Note, by a policy being "present in a data set" we mean that the set of state-action mappings which define the policy is contained in that data set. When a policy is not present in a data set, either some states covered by the policy are not seen at all in that data, or the actions chosen by the policy in some states are different to those seen in the data.

dialogue strategies using simulation-based RL, where the simulated environment is learned from small amounts of Wizard-of-Oz (WOZ) data.

In contrast to preceding work, our approach enables strategy learning in domains where no prior system is available. Optimised learned strategies are then available from the first moment of online-operation, and handcrafting of dialogue strategies is avoided. This independence from large amounts of in-domain dialogue data allows researchers to apply RL to new application areas beyond the scope of existing dialogue systems. We call this method “bootstrapping”.

In addition, our work is the first using a data-driven simulated environment. Previous approaches to simulation-based dialogue strategy learning usually handcraft some of their components.

Of course, some human effort is needed in developing the WOZ environment and annotating the collected data, although automatic dialogue annotation could be applied (Georgila et al, 2009). The alternative – collecting data using hand-coded dialogue strategies – would still require annotation of the user actions, and has the disadvantage of constraining the system policies explored in the collected data. Therefore, WOZ data allows exploration of a range of possible strategies, as intuitively generated by the wizards, in contrast to using an initial system which can only explore a pre-defined range of options.

5.1.1 Term Definition

The term to ‘bootstrap’ has various different meanings in the fields of computer science (to ‘boot’ a system), linguistics (theory of language acquisition), statistics (sample with replacement for statistical inference), physics, law, business and many more. We use the term in a sense which is closer to its original meaning of “pulling oneself up by one’s own bootstraps”. The term is said to have come from a tale from the adventures of Baron Münchhausen who, according to the story, escaped from a swamp by pulling himself up by the straps of his boots. Although in other versions of the story he hoisted himself using his own hair (Raspe, 1785), see Figure 5. In this book the term is used to describe the problem of how to learn an optimal strategy before a working system or prototype exists, and thus circumvent the chicken-and-egg problem. The bootstrapping method conceptually contrasts with dialogue strategies being manually “uplifted” by a by a human expert.

Note that “bootstrapping” was also used to determine other aspects of dialogue system design: Weilhammer et al (2006) “bootstrap” the ASR language model from WOZ data. Fabbrizio et al (2004, 2008) “bootstrap” various dialogue components from out-of-domain data. In general, the term “bootstrapping” is used to describe the problem of how to build system components in a data-driven manner without having in-domain data available.

5.1.2 Related Work

In general, RL-based dialogue policy learning is based on trial-and-error learning from seeing as many interactions as possible (see Sections 2.3.4 and 3.2.2.3). There are three main techniques for addressing this problem: generalisation of state spaces, e.g. (Henderson et al, 2005, 2008) , rapid learning from small data sets (Gasic and Young, 2011; Pietquin et al, 2011b), and simulation-based learning, also see Section 2.3.4. Nevertheless, both techniques require initial data to start with. In cases where a system is designed from scratch, however, there is often no suitable in-domain data.

Learning dialogue strategies from human-human interaction data (if available) is not an option, since humans behave differently with machines than with other people (Doran et al, 2001; Jönsson and Dahlbäck, 1988; Moore and Morris, 1992). Furthermore, human-human interaction is usually less affected by channel noise since humans are much better at handling noise and uncertainty (see Section 2.1).

It is also not considered an option to use several disparate data sets from which to build different simulated components. This approach assumes that simulated dialogue components are independent from each other, which clearly is not the case for most of the simulations. For example, it assumes that the user behaviour is independent from the channel noise. It is common practise in current research to obtain all the simulated components from one data set, e.g. (Henderson et al, 2005, 2008; Schatzmann, 2008; Singh et al, 2002). The work of (Prommer et al, 2006) also experiments with using data from an isolated data collection to retrain some of the simulated components. However, the results indicate that this can easily lead to experimental conditions which are inconsistent with the environment of the final task setup. Nevertheless, some sub-components can be learned using out-of-domain data. For example, a phoneme confusion matrix for ASR modeling can be learned on any (big and representative enough) data set (Stuttle et al, 2004).

In sum, there is a strong indication that we need one consistent data set of human-machine interaction for building a simulated learning environment. When building a system from scratch, however, there is often no suitable in-domain data available. Different approaches have been suggested to address the problem of lacking initial training data: handcrafting the simulated components (Schatzmann et al, 2007a), online learning (Chickering and Paek, 2007), transfering policies from another domain (Lemon and Liu, 2007; Lemon et al, 2006a), and also starting learning from a limited amount of WOZ data (Prommer et al, 2006; Williams and Young, 2004b).

Schatzmann et al (2007a) suggest to manually set the initial parameters of a simulated environment to learn a policy. This policy is then used to gather initial data, which then can be used to re-train the parameters of the simulation and then re-train the policy. We argue that that data-driven methods should be preferred in order to ensure consistent, realistic behaviour (see discussion Section 4.6).

Chickering and Paek (2007) circumvent the data problem by using online RL. However, online learning requires many iterations with real users to have noticeable effect. In addition, it requires the continuous use of exploration during online operation, which can result in a quite confusing and frustrating interaction for the system's users. Furthermore, it is not clear how to infer the reward function during online operation. Hence, online learning can be used to constantly improve/adapt a reasonably well-performing strategy, but it is not suited to design a strategy from scratch.

In recent work the use of WOZ data has been proposed in the context of Reinforcement Learning (Levin and Passonneau, 2006; Prommer et al, 2006; Williams and Young, 2004b). Williams and Young (2004b) use WOZ data to discover the state and action space for the design of a Markov Decision Process (MDP). Prommer et al (2006) use WOZ data to build a simulated user and noise model for simulation-based RL. While both studies show promising first results, their simulated environments still contain many hand-crafted aspects, which makes it hard to evaluate whether the success of the learned strategy indeed originates from the WOZ data. Schatzmann et al (2007a) propose to 'bootstrap' with a simulated user which is entirely handcrafted. In the following we propose what is currently the most strongly data-driven approach to these problems. We also show that the resulting policy performs well for real users.

In the following sections we discuss the general advantages and challenges of RL-based dialogue strategy learning when the simulated learning environment is obtained from WOZ data.

5.2 Advantages for Learning from WOZ Data

There are several advantages when learning RL-based dialogue strategies from WOZ data, which makes this approach an attractive alternative to the other approaches outlined above.

First of all, the data collection in a WOZ experiment does not require a working prototype, as discussed before. This allows us to learn optimal strategies for domains where no working dialogue system already exists. Optimised learned strategies are then available from the first moment of online-operation, and handcrafting of dialogue strategies is avoided. This independence from large amounts of in-domain dialogue data allows researchers to apply RL to new application areas beyond the scope of existing dialogue systems.

Furthermore, WOZ data can be used to explore human strategies as a basis for automatic optimisation. For example, the state and action set for learning can then be defined based on observing human behaviour first, as also suggested by Williams and Young (2004b) and Levin and Passonneau (2006). A hand-coded strategy for data-collection would not explore different dialogue policies. Even if some random elements could be added into a hand-coded strategy, this would still be constrained variation. Use of human wizards allows less restricted exploration of dialogue policies.

In addition, a WOZ experiment includes a controlled experimental setup which can be designed to closely anticipate the final system setup. Subjects are also asked to fill out a questionnaire (see Section 3.3.1). In conventional system development these user ratings are analysed by an expert and used to guide the design process. Alternatively, these scores can also be utilised to learn an automatic evaluation function using the PARADISE framework (see Section 2.2.2). In RL-based strategy development this function can then be directly used to train and test the dialogue strategy.

5.2.1 Challenges for Learning from WOZ Data

However, using WOZ data for RL-based strategy development also generates some challenges. In the following we discuss two problems in particular: how to learn simulated components from small data sets, and how to account for the fact that a WOZ study only simulates real Human-Computer Interaction.

First of all, WOZ experiments are expensive and usually only produce small amounts of data. Thus, the bootstrapping approach requires methods to learn simulated environments from little data. In order to find methods which are "appropriate" one needs to remind oneself of the purpose of a simulated environment for strategy learning. This purpose is usually two-fold. On the one hand it has to provide *realistic* feedback to the learner. This "experience" gathered in the simulated environment is then reflected in the learned Q-function (see Section 3.2.2). On the other hand, a simulated environment also needs to *cover* a wide state-action space in order for the learner to explore all possible situations in the dialogue. Exploration is necessary to guarantee robust strategies (see Section 2.2.4).

In previous work, the simulated components of these learning environments are either hand-crafted or learned via Supervised Learning (SL). The problem when applying SL to small data set is that the resulting models often suffer from "data sparsity", which is defined as follows Alpaydin (2004).[2] In general, SL is learning to infer a general mappings from seen example (see Section 2.3.1). If training examples are rare, the learner tends to "overfit" the data, i.e. it adjusts to very specific features of the training data. Models which overfit do not generalise well to unseen events and are also not realistic, in a sense that they are only representing the behaviour of a small population. Thus, the major challenge when constructing simulated learning environments from small amounts of WOZ data is to construct simulated components which generate realistic and wide-coverage behaviour.

Previous work on learning simulated environments from WOZ data used handcrafted components in combination with "low conditioned" models (Prommer et al, 2006), i.e. models which do not cover the entire policy space. For example, the bigram user simulation used by Prommer et al includes a lot of zero frequencies (see Section 7.8.1.2). In Chapter 7 we will introduce appropriate methods for building and evaluating simulated components from small data sets.

Another major problem in this framework is that a WOZ study only simulates real HCI. That is, a simulated environment learned from this data is a "simulation of

[2] Note that corpus size is not the only factor which can cause data sparsity. A corpus also need to be representative of the real underlying distribution, which does not necessarily depend on the size.

a simulation". We thus explicitly show that the learned simulations are a sufficient estimate of real HCI by post-evaluating the simulated environment in Section 8.5. We also show that a policy trained in this simulated environment "transfers" to real HCI, i.e. we test whether the obtained results are compatible. We do this by including an extra post-evaluation step in the general simulation-based RL framework.

In sum, developing RL-based dialogue strategies from WOZ data offers a number of important advantages over previous approaches, such as learning without a working prototype, the ability to study wizard behaviour, and the availability of user ratings. The major challenges are the data sparsity problem for SL and the fact that a WOZ study only mimics real HCI. In the next Section we describe a general method which addresses these challenges.

5.3 The Bootstrapping Method

In this book we introduce a method for training an optimised policy from WOZ data. We call this method bootstrapping, because an optimised strategy exists even before a working prototype system (see Section 5.1.1).

In particular, we follow a 5-step procedure (see Figure 5.3): We start by collecting data in a WOZ experiment. From this data we train and test different components of our simulated environment (such as the noise simulation and the simulated user) using Supervised Learning techniques. We then train and evaluate dialogue policies by interacting with the simulated environment using Reinforcement Learning. Once the learned policies are "good enough" we test them with real users. In addition, we introduce a final phase where we explicitly evaluate whether the models and policies obtained by bootstrapping from WOZ data are a valid estimate of real HCI (see step 5 in Figure 5.3).

The next Sections provide an overview of the specific methods used in each of the steps. We also include pointers to the respective Chapters which report on these steps in more detail.

5.3.1 Step 1: Data Collection in a Wizard-of-Oz Experiment

The first step in our framework is to collect data in a WOZ experiment. In order to use WOZ data as an initial corpus for dialogue strategy learning we apply the following changes to the conventional WOZ setup (as described in Section 3.3.1).

First of all, we are not only interested in the users' behaviour, but also what kind of strategies human wizards apply. That is, the wizard also becomes a subject of our study. (This approach has also been described as a "Ghost in the Machine" method.) We therefore have several different wizards participating in our study and explore "intuitive" strategies applied by the wizards. We do not restrict the wizard to follow a script (as done by other WOZ experiments, e.g. (Prommer et al, 2006; Türck, 2001)),

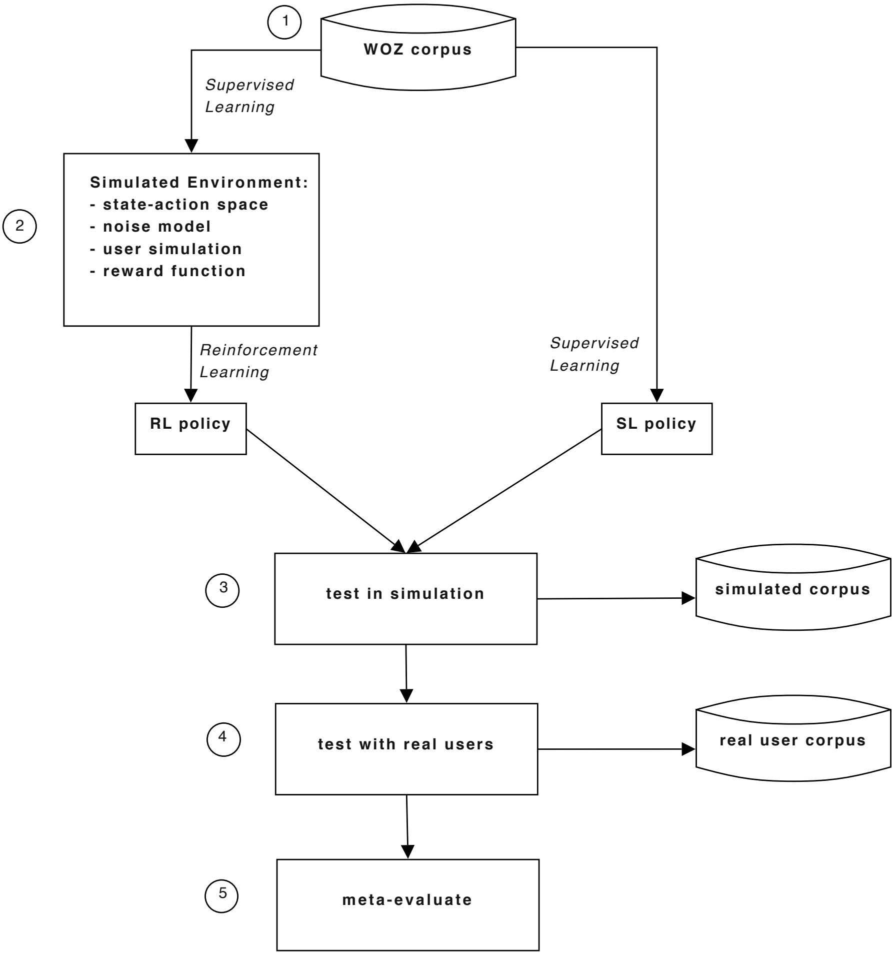

Fig. 5.2 Data-driven methodology for simulation-based dialogue strategy learning for new applications

but the wizard can interact freely with the user. Based on these initial insights about human behaviour we define the state and action set for learning, following Williams and Young (2004b).

These initial strategies should also account for noise in the interaction, as they should be suited to be applied in spoken dialogue systems (which are much more sensitive to noise). Error or uncertainty handling strategies (such as implicit or explicitly confirming what was said) are much less frequent in standard human-human communication, as we previously found in a comparative corpus study of human-human interaction (Rieser and Moore, 2005). We therefore artificially introduce simulated noise by randomly deleting words in the user transcripts before presenting them to the wizard.

The application domain of the experiment is multimodal information-seeking dialogue with an in-car music-player (see Section 3.4). In total we collected data from 21 sessions, containing 72 dialogues with approximately 1600 turns in this setup. A detailed overview of the experiment follows in Chapter 6 (also see (Rieser et al, 2005)). Note that the experiments were conducted in the larger context of the TALK project.[3] The resulting corpus is known as the SAMMIE-2[4] corpus (also see (Kruijff-Korbayová et al, 2005a,b, 2006a,b)).

5.3.2 Step 2: Build a Simulated Learning Environment

In the second step we use the WOZ data to construct a simulated learning environment. In particular, we model the following components: The action set and state space for learning, the user and noise simulation, and the objective function for learning and evaluation. All of these components are constructed using data-driven methods. The action set and state space are retrieved by exploring the wizards' actions. The user and noise simulations are both constructed using frequency-based approaches. The objective function is a predictive model of user ratings obtained by a regression analysis, following the PARADISE framework (see Section 2.2.2). Here one of the major questions is how to construct simulations from small amounts of data, as discussed before in Section 5.2.1. onstruct models with full More detail will be provided in Chapter 7 (also see (Rieser and Lemon, 2006a,b,c, 2011)).

5.3.3 Step 3: Train and test a strategy in simulation

In the third step of our framework, we train and test a policy by interacting with the simulated environment using Reinforcement Learning. We formulate the problem as a hierarchical Markov Decision Process (MDP) which reflects the structure of information-seeking dialogues (i.e. the information acquisition and the subsequent presentation phase). For strategy training we use the SHARSHA algorithm with linear function approximation. We test the RL-based strategy against a baseline which reflects the human wizard behaviour as observed in the data. This baseline is constructed by using Supervised Learning.

Note that RL is fundamentally different to Supervised Learning (SL): RL is a statistical planning approach which allows us to find an optimal policy (sequences of actions) with respect to an overall goal (Sutton and Barto, 1998); SL in contrast

[3] TALK (Talk and Look: Tools for Ambient Linguistic Knowledge; www.talk-project.org (20. September 2011)) was funded by the EU as project No. IST-507802 within the 6th Framework program.

[4] SAMMIE stands for Saarbrücken Multimodal MP3 Player Interaction Experiment.

is concerned with deducing a function from training data for predicting/classifying events. This book is not concerned with showing differences between SL and RL on a small amount of data, but we use SL methods to capture the average human wizard strategy in the original data, and show that simulation-based RL is able to find new policies, which were previously unseen. The supervised wizard strategy will be explained in Section 7.6 in more detail. The simulated-based learning experiments will be presented in Section 7.11 (also see (Rieser and Lemon, 2008d)).

5.3.4 Step 4: Test with Real Users

In the fourth step of our framework the strategy is tested with real users. We therefore develop a music-player dialogue system using a rapid development tool. The learned strategy is implemented using a table look-up between states and learned actions. In the user tests 17 subjects interact with the system, solving 2×6 tasks with each policy (SL and RL). At the end of each task they also fill out a questionnaire. The experiments will further be described in Chapter 8 (also see (Rieser and Lemon, 2008c, 2011)).

5.3.5 Step 5: Post-Evaluation

In the final step of our framework we address the problem that a WOZ experiment itself is only a simulation of real HCI. We therefore compare different aspects of the 3 corpora gathered so far: the WOZ study, the dialogues generated in simulation, and the final user tests. We first compare dialogue strategy performance obtained in simulation with the results obtained when testing with real users. We also compare the experimental conditions of the different studies, where we discuss the noise model in particular. We furthermore explore whether the objective function used for learning is a realistic estimate of real user preferences. We will provide more detail in Chapter 8.5 (also see (Rieser and Lemon, 2008a, 2011)).

5.4 Summary

This Chapter introduced and motivated a 5-step procedure for bootstrapping an optimised strategy from WOZ data using Reinforcement Learning methods. We start with a data collection in a modified WOZ setup. In this setup the wizard also becomes a subject and channel noise is artificially introduced. We then construct a simulated learning environment from this data, where all the simulated components are constructed using data-driven methods suited for learning from small data sets. The strategy is then trained and tested by interacting with this simulated environment.

We compare RL against a supervised baseline which is derived from the wizards' behaviour. This comparison allows us to measure relative improvements over the training data. We also evaluate the strategy with real users (see Chapter 8). Finally, we post-evaluate the policy and the simulated environment by comparing the WOZ data, the simulated data, and the real user corpus. We also conduct a detailed error analysis, see Section 8.5.

Annotated example dialogues from the 3 different corpora can be found in Appendix A: Appendix A.1 contains dialogues from the WOZ study, Appendix A.2 simulated dialogues, and Appendix A.3 examples from the user study.[5]

We now apply this framework to optimise multimodal information-seeking dialogue strategies for an in-car digital music player. Dialogue Management and multimodal output generation are two closely interrelated problems for information seeking dialogues: the decision of *when* to present information depends on *how many* pieces of information to present and the available options for *how* to present them, and vice versa. We therefore formulate the problem as a hierarchy of joint learning decisions which are optimised together. We see this as a first step towards an integrated statistical model of Dialogue Management and more advanced output planning/Natural Language Generation, see Chapter 9.

[5] Note that we will refer to these example dialogues in different chapters throughout this book . We therefore decided to put the example dialogues in the Appendix in order to facilitate consistent reference and allow the reader to directly compare dialogues from different corpora more easily.

http://www.springer.com/978-3-642-24941-9

Reinforcement Learning for Adaptive Dialogue Systems
A Data-driven Methodology for Dialogue Management
and Natural Language Generation
Rieser, V.; Lemon, O.
2011, XVI, 256 p., Hardcover
ISBN: 978-3-642-24941-9