Renforcements naturels pour la collaboration homme-machine

Esther Hoare Nicart

Normandie Université

Pour obtenir le diplôme de doctorat

Spécialité Informatique

Préparée au sein de l'Université de Caen Normandie

Renforcements naturels pour la collaboration homme-machine
(Qualitative reinforcement for man-machine interactions)

Présentée et soutenue par
Esther NICART

Thèse soutenue publiquement le 6 février 2017 devant le jury composé de		
Mme. ABEL Marie-Hélène	Professeur des universités : Université de Technologie de Compiègne	Rapporteur
Mme. GRIGORI Daniela	Professeur des universités : Université Paris Dauphine	Rapporteur
M. CHEVALEYRE Yann	Professeur des universités : Université Paris 13, LIPN	Examinateur
M. GRILHÈRES Bruno	Airbus Defence & Space Élancourt	Coencadrant industriel de la thèse CIFRE
M. SAVAL Arnaud	Cordon Electronics DS2i Val de Reuil	Coencadrant industriel de la thèse CIFRE
M. ZANUTTINI Bruno	Maître de conférences HDR : Université de Caen Normandie	Directeur de thèse

Thèse dirigée par Bruno ZANUTTINI, laboratoire GREYC

ED SIMEM

UNICAEN UNIVERSITÉ CAEN NORMANDIE

GREYC MAD

Abstract

Information extraction (IE) is defined as the identification and extraction of elements of interest, such as named entities, their relationships, and their roles in events. For example, a web-crawler might collect open-source documents, which are then processed by an IE treatment chain to produce a summary of the information contained in them. We model such an IE document treatment chain as a Markov Decision Process, and use reinforcement learning to allow the agent to learn to construct custom-made chains "on the fly", and to continuously improve them. We build a platform, BIMBO (Benefiting from Intelligent and Measurable Behaviour Optimisation) which enables us to measure the impact on the learning of various models, algorithms, parameters, *etc.* We apply this in an industrial setting, specifically to a document treatment chain which extracts events from massive volumes of web pages and other open-source documents. Our emphasis is on minimising the burden of the human analysts, from whom the agent learns to improve guided by their feedback on the events extracted. For this, we investigate different types of feedback, from numerical rewards, which requires a lot of user effort and tuning, to partially and even fully qualitative feedback, which is much more intuitive, and demands little to no user intervention. We carry out experiments, first with numerical rewards, then demonstrate that intuitive feedback still allows the agent to learn effectively. Motivated by the need to rapidly propagate the rewards learnt at the final states back to the initial ones, even on exploration, we propose Dora: an improved version of one of the most standard of reinforcement learning algorithms, Q-Learning. Finally, we apply BIMBO and Dora to a different setting - that of the detection of objects in an image.

Acknowledgements

"Acknowledgement" doesn't sum up the depth of gratitude that I feel towards my directors, colleagues, friends and family. If I were to list all the people who have had an impact on this thesis, and thank everyone to the depth I feel they deserve, then this section would be longer than the rest of the manuscript put together. By necessity, therefore, I have to keep it shorter than I would like.

Firstly, I would like to thank Stéphan, who welcomed me into his team as a Masters internship student, and encouraged me not only to go on to tackle a thesis, but to finally learn the crawl.

Over three years, I built up a a wonderful collection of thesis directors, each of whom contributed something different. Thank you very much to Bruno Z, Bruno G, Patrick, Arnaud, Fred and Stéphan.

Thanks too, to my misogynic colleagues for their good natured teasing, and above all for their help and support in the darkest moments (especially "Tata" Véro). Special mention to Clément (and his Hippo, Amandine) and Guillaume for their late night proof-reading of my français execrable just before the conference deadline, and to Guillaume again for the 1000's of kms driven. Arnaud should also be singled out for his meticulous proof-reading of this manuscript, helpful suggestions, and technical support. Bruno G, thank you for hitting me in the car park. The internship students, Francis-Guillaume and Vincent-Valentin did a great job integrating BIMBO with the image analysis services, and taught me a lot about GIT, lighting barbeques, and image manipulation. To the Cordon DS2i managers, thank you so much for letting me work uninterrupted on my thesis during the final sprint.

On an academic level, my wholehearted thanks go to Hugo for our discussions on qualitative feedback. I would also like to thank the reviewers of our work for their thought-provoking questions and comments. I really appreciate the time you took to read the articles and respond in such depth. Bruno Z, you put up with my stupid questions, idiotic remarks, and appalling French, and still you smiled and stayed patient. I don't know how you did it, but I thank you.

For her linguistic advice and weather analysis, I have to thank Anneke of Stellenbosch University (she will sulk otherwise), who became an instant friend when she insulted me over breakfast.

And last, but absolutely not least, my family:

Florent, my lovely marmotte, I can't tell you how much your support has meant. You understood when I shut myself away for days / weeks / months / years on end, spending more time with BIMBO and Dora than with you. You also uncomplainingly took on way more than your fair share of all the household tasks. Without you, I would have starved, had nothing to wear, and would have given up long ago (actually, I wouldn't even have started!) <3 lftcyvs.

Thank you mum, dad and Peter. Despite the physical distance, you've been right by my side emotionally with your supportive emails, cards and 'phone calls. Let's raise a glass of port to bank clerks everywhere.

For the things we have to learn before we can do them,
we learn by doing them

Aristotle, The Nicomachean Ethics

This manuscript was written using LaTeX, Tikz, gnuplot and Inkscape, and edited and compiled under Kile. BIMBO sent her progress report text messages using the free.fr API.

Foreword

This thesis was carried out under the French *ANRT* (*Association Nationale de la Recherche et de la Technologie* or National Association for Research and Technology) system, specifically the *CIFRE* (*Conventions Industrielles de Formation par la REcherche* or Industrial Agreements for Training through Research). The *CIFRE*'s objectives are to develop partnerships between public research laboratories and businesses, and to help doctors to find employment outside academia.

The proportion of time that the student should spend working on the thesis as opposed to working for the company is not specified by the ANRT, but in my case, I was able to spend around one half to two thirds of my time on my thesis or related project work.

During the three years of this thesis, I officially changed company several times. At first it was just name changes from *Cassidian* (originally *EADS Defence and Security*) to *Airbus Defence and Space*. Then a major change occurred during the second year of my thesis, when Airbus Defence and Space were faced with the option of either closing or selling the site on which I worked. A buyer, *Cordon Electronics*, was eventually found, and the transfer finalised in October 2015 when we became *Cordon Electronics DS2i*.

This transfer posed a technical challenge as the in-house WebLab services such as translation, and in-depth document analysis (see chapter 4) that we originally planned to introduce into the chain belong to Airbus Defence and Space. They kindly granted me special authorisation to keep the service encapsulating GATE and the WOOKIE ontology developed by Serrano (2014) that I had started using for my tests. All other services that I used are open-source.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Imagine a car factory that produces cars with one wheel on the roof, one sticking out at the side, and two on the ground. The customers of this rather odd factory have to change the wheels every time they buy a car. It sounds improbable, but this can be the case with Information Extraction (IE) systems. Vast amounts of data are trawled, and the system tries to provide a summary of the useful information. Any flaws in the extraction of this information mean that the summary is inaccurate, and the end user will need to correct it.

The factory realises that the customers are not happy, and so they provide an expert who can produce custom cars. The customer just needs to take the time to come into the factory and explain exactly what he wants. The IE system (often a chain of treatments) can likewise be configured by an expert in discussion with the user (hoping that the user doesn't change his mind once he sees the results).

The factory decides to increase user satisfaction by providing tools that allow the customer to change the factory layout themselves, and to give the customers training in car-assembly. The user of the IE system has been given special languages to interact with the IE chain, but he can lack the expertise to appreciate the impact of his choices.

The factory then takes a radical decision. It creates a new customer feedback department. They issue opinion polls, asking users to mark the features of their cars from 1 to 1000, or to rank them in order of preference. They then use these questionnaires to build tailor-made cars for each customer. We can see that obtaining feedback from the users could help to improve the IE services, but is it really natural to ask them to provide it explicitly and numerically?

Finally, the factory installs a monitoring device inside the car, which automatically tells the feedback department exactly how the car is used, how it is modified, which colour the customer paints it on which day, which features make the customer smile most, *etc.* At last, they can produce tailor-made cars for the customer, without having to interrogate them, train them, or get them to fill in lengthy questionnaires.

Our objective is to install a "monitoring device" into an IE chain. It will observe the users actions, and requires only intuitive natural feedback to improve the chain (although, of course, it can be given numerical feedback too).

1.1 Context

Many people rely on IE systems, including *Open Source INTelligence* (OSINT) analysts. In the past, the OSINT analyst's task was finding hidden information. Now, faced with ever-increasing volumes of web pages and other open-source multimedia, the challenge is to find pertinent information from the huge wealth of data available.

Many specialised treatment chains have been developed to ease this task, but in the particular platform that we examine[1], open-source multimedia are inserted continuously into a series of web services which aim to extract events (*e.g.*, terrorist attacks) and their characteristics (date, place, agents, *etc.*) for OSINT assessment and surveillance.

The chain, defined by experts, consists of a series of individual treatments, such as the detection of the multimedia format, transcription, sentiment analysis, language recognition, translation, extraction of named entities and events using nouns and verbs as triggers, *etc.* An example of a possible multimedia treatment chain is shown in Figure 1.1, and we invite you to see section 4.1 for more details of the document treatment chain that we used for our tests.

It is clear that the output of this chain, the named entity and event extractions, cannot be perfect. Firstly, because of the huge diversity of multimedia on the Web, there cannot be a single optimal chain for all of them. Secondly, imagine, for example, a school blog recounting "the *bombardment* of a gold *target* by ions during an *atomic* physics demonstration at the school science fair". The treatment chain could erroneously recognise and extract an atomic bomb attack. Thirdly, the need to treat multimedia coming from all over the world in diverse languages entails the use of dictionaries and translation services whose quality is variable. Finally, an error early in the chain can be amplified as it goes further down the chain, for instance an error in the transcription of a voice recording will probably cause errors in the translation of that transcription, *etc.* The analysts are therefore typically forced to correct *a posteriori* the events extracted. These corrections are

[1] As part of the industrial challenge underlying the CIFRE contract.

stored in the database, but are never fed back into the extraction process, and are thus "wasted".

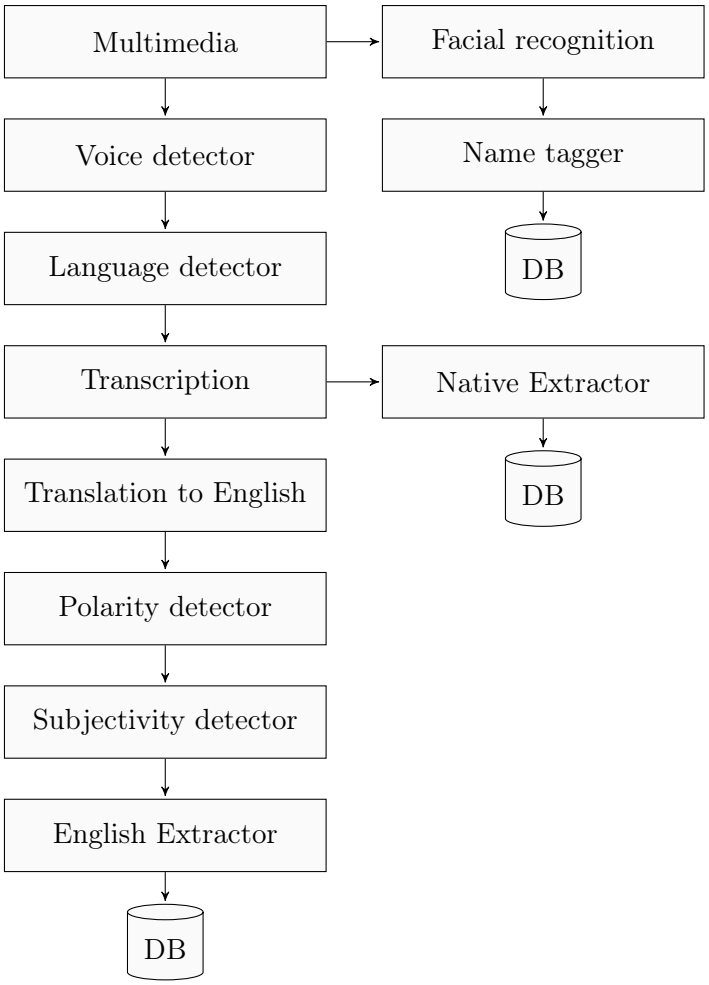


Figure 1.1 – An example of a possible treatment chain which extracts information from a video. The faces are identified and tagged with names. In parallel, the languages spoken are identified and the voice recording is transcribed. The native language extractor is used to summarise the information contained in the transcript, whilst the transcription is also translated. The polarity of the opinions expressed, and their subjectivity is assessed, and an English extractor used. In all cases the information is indexed in a database.

There is currently no way to improve the chain over time without the intervention of an expert in consultation with the analysts.

So why don't we let the analyst reconfigure the chain themselves? First and foremost, they simply do not have the time to make the alterations. Even if they did, they are not system experts, and can lack the expertise to appreciate the impact of their choices. Nor do they have an in depth knowledge of the services to correctly determine which is best to use in a given situation.

Our objective is to remedy this situation by providing a "feedback department" for the chain which receives its feedback as the analysts consult and correct the system output. It is responsible for modifying the behaviour of the chain in real time, ensuring its continuous improvement, and reducing human effort by decreasing the error rate. For example, if the analyst corrects an event, we could give an explicit numeric feedback based on the distance between the event as corrected and that which was extracted. However, this numerical feedback is neither natural nor easy for the analyst to define. It is not intuitive to determine precisely that "Correcting five letters in a partially correct name is worth 0.71, but deleting an incorrectly extracted place is worth 0.69".

We therefore want to advance to improving the chain based on qualitative, intuitive feedback by "monitoring" the user actions, applying naturally expressed user preferences which can be gathered or changed very easily, such as "I prefer a partial, or even a false summary to seeing nothing extracted" and "I prefer the treatment to be as fast as possible", and applying them to the extraction results.

1.2 Contribution

We formalise the improvement of a document treatment chain which extracts events from web pages and other open-source documents as a reinforcement learning (RL) problem (chapter 5), and the treatment chain itself as a Markov Decision Process (MDP) (chapter 6).

We build a platform, *BIMBO* (Benefiting from Intelligent and Measurable Behaviour Optimisation), which can modify the treatment chain in real time, based on the end user feedback (chapter 7).

BIMBO is built in a modular fashion, which means that we easily test different RL algorithms and reward mechanisms, and measure their impact on the learning.

We investigate three different types of user feedback:

- In the first, the agent receives numerical rewards on the quality of the treatment (chapter 8). These experiments show that automatically learning how to chain and parametrise services is perfectly feasible. The problem is that numerical rewards are very difficult for the human user to calibrate.

- In the next two, we use qualitative feedback, for which we present a novel algorithm in chapter 9.

  - We first use a partially qualitative formalisation (chapter 10, subsection 10.1.1). The agent is rewarded with a semi-qualitative feedback which gives an indication of the strength of the user preferences, rather than a specific numeric value. This proves very effective, but still requires some calibration.
  - Then in chapter 10, subsection 10.1.2 we use a fully qualitative and intuitive user feedback. This is less informative for the agent, but requires no calibration and can still give very good results.

You will see in chapter 8 that during the learning process, the AI may try several hundreds of actions during one treatment. This leads to a very long chain of state / action pairs (see chapter 5), or "paths" before it finally receives a reward. Frustrated by the length of time that this reward was taking to back-propagate along these very long paths, we suggest an improvement to the algorithm Q-Learning, Dora, which loses less information when exploring (chapter 12).

We initially apply BIMBO to a document treatment chain, but we demonstrate the flexibility of our approach by applying BIMBO and Dora to the parametrisation of image analysis services (chapter 13). In fact, BIMBO is not dependent on the web services used, nor the type of input, and can therefore equally be used for the study of any process where we can define the states in terms of the system and the "thing" being processed, and where the actions can influence the parameters and / or the process itself.

1.3 Organisation of manuscript

We start by giving an overview in chapter 2 of work which is related to our own. This chapter has been kept intentionally relatively short, as we interleave the references with the rest of the manuscript, going into detail of particular examples in the relevant sections.

In the main part of this manuscript, Part I, we examine the industrial challenge which motivated this work (chapter 4) and our response to it as a reinforcement learning problem (chapter 5), modelled as a Markov Decision Process (MDP) (chapter 6). In chapter 7, we outline the platform that we constructed, and the general experimentation framework. We detail the first experiments that we carried out with numeric, potentially sporadic, simulated feedback data (chapter 8), which show that automatically learning how to chain and parametrise services is perfectly feasible. We recognise that users do not naturally provide numeric feedback, and so in chapter 9 we present our contributions towards a novel algorithm which combines qualitative feedback with reinforcement learning, the design of which was led by Hugo Gilbert (LIP6, Université Pierre et Marie Curie, Paris) with our collaboration. We then present our experiments with two types of intuitive feedback (chapter 10), which we show are perfectly feasible solutions, while requiring little to no tuning.

In Part II, we detail two spin-offs of our research. The first (chapter 12) is Dora, an improvement to the reinforcement learning algorithm Q-Learning, which loses less information when exploring. The second is a demonstration of the flexibility of our approach, applying BIMBO and Dora to the detection of objects in images (chapter 13).

Finally, in Part III: Conclusion we lower the curtain with some thoughts on how this work might be built on in the future.

1.4 Publications

Some of the work detailed in this document has already been published or presented in the following:

Nicart, E., Zanuttini, B., Gilbert, H., Grilhères, B., and Praca, F. (2016). Building document treatment chains using reinforcement learning and intuitive feedback. In Proc. 28th International Conference on Tools with Artificial Intelligence (ICTAI), 2016.

Gilbert, H., Zanuttini, B., Weng, P., Viappiani, P., and Nicart, E. (2016). Model-free reinforcement learning with skew-symmetric bilinear utilities. In Ihler, A. and Janzing, D., editors, Proc. 32nd Conference on Uncertainty in Artificial Intelligence (UAI 2016), pages 252–261. AUAI Press.

Nicart, E., Zanuttini, B., Grilhères, B., Giroux, P., and Saval, A. (2016). Amélioration continue d'une chaîne de traitement de documents avec l'apprentissage par renforcement. Accepted in Revue d'Intelligence Artificielle (RIA). In French.

Nicart, E., Zanuttini, B., Gilbert, H., Grilhères, B., and Praca, F. (2016). Building document treatment chains using reinforcement learning and intuitive feedback. In Pellier, D., editor, Proc. 11es journées Francophones sur la Planification, la Décision et l'Apprentissage pour la conduite de systèmes (JFPDA 2016).

Nicart, E., Zanuttini, B., Grilhères, B., and Praca, F. (2016). Dora Q-learning - making better use of explorations. In Pellier, D., editor, Proc. 11es Journées Francophones sur la Planification, la Décision et l'Apprentissage pour la conduite de systèmes (JFPDA 2016).

Nicart, E., Zanuttini, B., Grilhères, B., and Giroux, P. (2015). Amélioration continue d'une chaîne de traitement de documents avec l'apprentissage par renforcement. In Abel, M.-H., editor, Proc. 26es journées francophones d'Ingénierie des Connaissances (IC 2015). In French.

Chapter 2

Related work

2.1 Introduction

Throughout this document, where relevant, we examine the existing academic work and draw parallels with our own. Nevertheless, in this chapter, we give a short, self-contained summary of work related to ours.

Even though the applications of our work are broader, as we said in chapter 1: Introduction, this thesis was initially motivated by the need to extract events from open-source documents for the *Open Source INTelligence* (OSINT) community. In section 2.2, therefore, we examine Information Extraction (IE), specifically the extraction of events.

To our knowledge, an agent which constantly learns and self-improves from natural human feedback to dynamically construct and parametrise a chain of services has not yet received academic attention. Nevertheless, several of the base elements have been studied. We summarise some of these elements here, and enter into more detail in the following chapters.

We are all familiar with and use daily the results of treatment chains, consisting of an input, a process, and a result. A production chain in a factory, for example might have pieces of metal and plastic as input, programmed robots to execute the process, and finally produce a car. Chaining together services to attain a desired result is therefore not a new concept, and we examine several approaches in section 2.3.

Neither, evidently, are we the first to model a process as a Markov Decision Process (MDP); section 2.4 gives examples of some approaches to modelling a problem using MDPs: the policy can be defined in advance by an expert, it can be learnt from existing data, or it can be learnt "on the job" from real-time observations.

Finally, using human feedback, both explicit and implicit for model-based learning has also been explored. Our objective is to enable the human users to give an intuitive, natural feedback instead of a numerical reward. In section 2.5, we examine some of the work that has already been carried out in this domain.

2.2 Information Extraction

So we examine Information Extraction (IE), but what is it, and how does it differ from Information Retrieval (IR) and Information Acquisition (IA) (which we consider out of the scope of this thesis)?

Cowie and Lehnert (1996) compare IR to a combine harvester which trawls vast fields to harvest potentially useful raw material, and IE to the refining process which transforms it. Continuing this culinary analogy, we could say that IA is like mushroom picking in the forest - you are searching for something specific. A web-crawler, for example, could be an IR engine collecting open-source documents, which are then processed by an IE treatment chain which produces a summary of the information contained in them. An IA engine then might search for further or missing information (which could be derived from several other data) on one of the elements of interest.

More specifically, Hobbs and Riloff (2010) define IE as the identification and extraction of elements of interest, such as the named entities, their relationships, and their roles in events. Event extraction from text, the "who did what to whom, how, when, where, why and with what?", is the subject of numerous studies. For instance, Hogenboom et al. (2011) give an overview of three different extraction methods: data-driven, expert knowledge-driven and hybrid, and Serrano et al. (2012) carry out a comparative study of statistical and linguistic rules for automatic event extraction.

These IE chains are expensive to build and maintain, and components are often not reusable. For instance, Cowie and Lehnert (1996) estimated that the cost of transferring an existing system to a new domain would take from three to six person-months. This transfer is long, and if it has to be carried out by an expert, would be very costly. This led them to state that domain-specific IE systems were only acceptable if they did not require a specialist to tune them. Chiticariu et al. (2013) state that some of the main costs of developing an IE system are the labour involved in understanding and defining the business problem, and writing the extraction rules. IE platforms, therefore should ideally be multi-lingual (Lejeune, 2013) and multi-media (Ji, 2010), with standard rule definition (Chiticariu et al., 2013) and should represent the knowledge acquired on several levels (Lee et al., 2010).

Not only is the chain expensive to build, it is also difficult to test. The difficulty is that the annotated corpora that currently exist (*e.g.* ESTER2, Quaero, *etc.*) tend to be domain-specific, or not broad enough, covering only named entities (Fort, 2012), and not those elusive relations between the actors and the events (Ludovic, 2011). Annotating a document by hand is

time-consuming and error-prone, and the automatic annotation of a corpus for IE is complex (Riloff, 1996). The Automatic Content Extraction (ACE) Program was a competition (Linguistic Data Corporation (LDC), 2016; Doddington et al., 2004) which aimed "to develop extraction technology to support automatic processing of source language data ..." Intitially, the ACE research objectives were the detection and characterization of named entities, but from phase 3 onwards, ACE sought to pit IE systems against each other to correctly automatically annotate the events and named entities in documents (Ji and Grishman, 2008).

The particular IE chain that we use extracts events for open-source intelligence. Serrano et al. (2013) explain the motivation behind the development of these extractions, and propose a data-fusion method to combine two events based on mutual completion of information and their proximity in an ontology. However, there is no consensus as to how to define an event, nor how to represent the entities and events so that they are really open-domain and able to be co-referenced (Bejan and Harabagiu, 2014; Serrano, 2014; Byrne, 2009; Dutkiewicz et al., 2013; Van Hage et al., 2011), nor even on how to measure the similarity between named entities, events, and their relations (Saval, 2011; Moreau et al., 2008; Wang et al., 2009; Cohen et al., 2003; Wang et al., 2011; Cohen et al., 2013; Kumaran and Allan, 2005; Lao et al., 2012; Lao and Cohen, 2010; Bellenger et al., 2011; Gardner et al., 2014). Steinberger et al. (2013) have made a step in this direction with The Europe Media Monitor platform, extracting events from world-wide news feeds, enabling them to be co-referenced, viewed on a map, translated, *etc.* Similarly New Event Detection (NED) aims to monitor a stream of news and trigger alerts when a new topic or event is detected. This problem centres around the comparison of incoming information sources with an existing knowledge base of all those seen in the past in order to detect new topics. NED techniques (see *e.g.* Kumaran and Allan (2004) for an overview) are often based on the similarity between terms used in the documents, or their temporal difference (how much older one story is than the other).

2.3 Chain configuration

Given that these IE treatment chains are so costly to build, several attempts have been made to do it automatically.

The chain can be planned and fixed in advance, as in Saïs et al. (2013) who model service composition using AI-Planning and Petri Nets for an IA system. The service compositions that they compute help the user to fill in

forms. The inputs and outputs of each e-service are known in advance, and it is a planning task to string them together.

The chain can even adapt itself depending on the input, for example, the physical characteristics of documents can be successfully used to construct adaptive, modular chains for Xerox photocopiers in real time (Fromherz et al., 2003). As with the service composition task above, each module, or operation, within the photocopier is defined in terms of its possible inputs and the time taken to produce each possible output. A system controller is given the task information or document specification, for example "five collated, stapled, double-sided copies of a 10-page document". From this specification, the controller can plan the optimal sequence of operations which produce this result, and then schedule them. The scheduling consists, for example, of leaving a gap between sheets where necessary to avoid jams because one operation takes longer than the others, or of interleaving operations where possible to save time.

The chain can also be modified "on the fly" by an expert. For instance, Doucy et al. (2008) use the platform WebLab to construct a modifiable document treatment chain. The services can be chosen from a central directory and substituted as required using BPEL (Business Process Execution Language) to change the underlying XML, and the WSDL (Web Service Definition Language) of each service to ensure the substituted service is compatible. Similarly Rodrigues et al. (2015) allow the reconfiguration of a chain of web-services through a domain specific language, ReCooPLa (developed as an Eclipse plugin). This allowed them to formalise the graph of potential interactions between the services in a Service Oriented Architecture (SOA). The software architect can then use this language through a graphical interface to try out different configurations, whilst ensuring that the whole remains coherent.

The particular chain that motivated our work is an IE document treatment chain. Open-source documents (the input) are fed into a software chain of web services which add information (or not) to the document (the process), and finally that information is presented in a user-friendly format (the output). For an overview of similar processing chains that are provided as web services, we invite you to consult Ogrodniczuk and Przepiórkowski (2010). The inputs, duration, and outputs of each web service are unknown in advance, and so the agent must learn to build the chain step by step, according to the information currently available.

2.4 Modelling

Any uncertain environment where decisions have to be taken sequentially can potentially be modelled as a Markov Decision Process (MDP).

For instance, in the financial world, So and Thomas (2011) model the credit limit of clients for banks, enabling them to estimate the profit that they might gain by increasing the credit limit. The states reflect the client's current behaviour score and their current credit limit. The actions available are to increase the credit limit by various amounts. The reward received by the agent is the profit made for the bank.

In a medical environment, a course of treatment is influenced by the patients current state of health, their response to the treatment, and the anticipation of future treatment that may be necessary. Schaefer et al. (2005) detail some of the applications which have already modelled medical treatments as MDPs. Staying in the medical domain, Ma and Kwiatkowska (2008) examine a hand-washing process for dementia patients using action policies which are pre-defined by an expert. Observations are provided by sensors, e.g. tracking user's hands, towel and soap, and the actions are the prompts given to the patient. The patient characteristics, like the action policy, are specified by an expert. They use PRISM (a stochastic model checking tool) to compare the efficiency of the policies and to identify factors that play important roles. This allows them to establish general rules for choosing policies for different users.

Closer to home, in the field of information acquisition (IA), Kanani (2012) tries to minimise the cost of filling a partially complete database where recuperating the required information is expensive. She models this Resource-bounded Information Extraction (RBIE) problem as an MDP. The probabilistic model for extracting the values from the external sources, and the relative importance of each piece of information is known in advance. The state of the database at each timestep, plus the intermediate results gained so far defines the states of the MDP, and the actions are query, download and extract. The order of these actions is fixed, and the structure of the queries defined in advance, but the queries are filled in dynamically based on the information sought. In constrast our agent must learn the probabilistic model, and the optimal order in which to perform the actions, which may vary from document to document.

2.5 Learning from users

Humans share their knowledge with each other in many ways: through speech, writing, demonstrations, *etc.* They can also transfer this knowledge to machines by training them, either to replicate the skills of the human, or to learn their preferences to aid them in performing a task.

Bratko and Šuc (2003) argue that replicating human skills through a dialogue is not a cognitively easy task. Often the human is incapable of describing precisely what he does numerically, but will be more at ease expressing himself qualitatively. They only require qualitative observations of the changes in the environment, formalised as Qualitatively Constrained Functions (QCFs) where: $Z = M^{+,-}(X, Y)$ means that Z monotonically increases in X and decreases in Y. It can be difficult, however, to tell how Z will change. For example, if both X and Y increase, then Z can increase, decrease, or stay unchanged. There must also be a large enough delta for the difference to be detectable (Saaty, 2008). Bratko and Šuc (2003) also point out that the actions of the human are often a better indicator of how to perform the task, and "cloning" the human by examining these traces enables a knowledge transfer, not only to the machine, but to other humans. These traces can also be used to evaluate user profiles. For example, Karami et al. (2014) use two algorithms for this. The first updates itself using actual user-interactions or traces. The second is faster, but with a higher risk of error, as it tries to generalise existing rules to new interaction contexts and / or user profiles. It determines the relevant attributes of the state by applying previous knowledge about the rewards gained by performing a given action, and the values of the attributes of the state at that time.

Machines can learn from qualitative feedback in a whole range of ways, from staying a silent observer to being an active participant, asking questions and trying to carry out the task itself.

Some examples of silent observers can be seen in the improvement to translation models. Users correct or evaluate the output, and the translation model is updated as a result (Formiga et al., 2015; Snover et al., 2006). Culotta et al. (2006) introduce us to the idea of using the corrections to make a permanent difference to the system, referring to the corrections or evaluations as "corrective feedback", and the updating of the model as "persistent learning". Some systems allow the user to interact more directly with the system through a specialised language (Chai et al., 2009) to allow users to give feedback by correcting the extracted data, or to change the parameters of the extraction system. Changes are then propagated through the database and the next extraction crawls.

The agent can also learn by observing the trainer's actions on the same tasks. This allows it to extract a reward function simply by observing optimal behaviour. This is known as inverse reinforcement learning (IRL) and Ng and Russell (2000) present several algorithms for recovering reward functions from the observed behaviour. The premise is that the users have already defined a reward function whether consciously or not, and the agent has to try to guess it. Loftin et al. (2015) take this idea further by allowing the agent to learn, not only from explicit user feedback where the user specifies a reward, but also by deducing the implicit or withheld feedback, where the user does not provide a reward, but where one could be inferred. Their users are non-technical, and will not necessarily provide feedback in a homogeneous manner, for instance, a user may only reward good behaviour (explicit feedback), in which case, the agent can assume that a lack of reward (withheld or implicit feedback) is a punishment, and vice versa. They show that allowing the agent to deduce these "training strategies" speeds up the learning. We could imagine a parallel with our work. In our case, the agent could be "punished" for the corrections the user makes. However, the agent could also take into account a withheld or implicit feedback, such as opening a document without correcting it, as a positive reward. We discuss the implications of this with respect to how we model the user feedback in section 6.3.

The agent can also participate actively in the improvement by presenting the user with choices, or by building up models of the user's behaviour and preferences through human-system interactions to provide personalised systems. In Veeramachaneni et al. (2016), for example, the agent analyses a large raw data set, and uses outlier detection to present the analyst with a small set of events which it thinks are potential terrorist attacks. The analyst gives feedback in the form of labels for these events. The agent then uses these labels to learn or modify the models which it applies to the data set to predict the attacks.

Task performance can also be improved by asking the human to rank outcomes. This is the subject of preference-based reinforcement learning (PBRL), which combines reinforcement learning and preference learning (see Fürnkranz et al. (2014) for a recent state of the art). The user is asked to define preferences over actions from a given state based on simulations, or rollouts (Busa-Fekete et al., 2014; Fürnkranz and Hüllermeier, 2011; Cheng et al., 2011). Fürnkranz et al. (2012) give two main approaches to representing preferences. The first is numeric, or quantitative, and is the standard approach in RL, leading to an estimation of the value of choosing an action in a state. The disadvantage of this approach is that the user must weigh

up all the criteria composing a state, and give it a numerical value. The second approach, preference learning, is qualitative and relies on the user expressing a binary preference over possible actions, rather than having to give a concrete value.

Another approach is that of Akrour et al. (2011, 2012) who ask the user to rank policies as demonstrated by the robot. They then use this qualitative feedback to optimize the policy. They point out that care must be taken to limit the complexity of the comparisons.

Akrour et al. (2013) reduce the complexity of the choice by only presenting trajectories to the user, who is asked to express pairwise preferences. They also examine the problem of trust between the agent and the human. If the agent consistently presents trajectories that are difficult to compare, then the human is likely to become inconsistent, and the agent loses trust in them. On the other hand, if the agent is well-taught at the start and presents useful information to the human, their error-rate decreases, and the agent's trust is gained. Similarly, Wilson et al. (2012) present trajectory snippets to the experts (rather than whole solutions). They calculate the distance between these trajectories by comparing the state-action pairs composing them.

The issue of trust is especially important when the human and the agent have different agendas. In Azaria et al. (2012), an agent and a human interact. The agent provides advice to the human user as to the choice he should make at each step. They show that a completely self-interested agent will lose the trust of the human, as the actions that it recommends will not accord with the human's goals. The human will then pursue his own agenda at the cost of that of the agent. By taking into account the user preferences when recommending an action, the agent can persuade the human to take a path which is beneficial to them both.

The knowledge gained by an agent can often be applied to other situations, for example, if I learn how to make carrot soup, then I can probably generalise my knowledge and make courgette soup too. This is the idea behind transfer learning; that in learning to perform one task, I gain experience which can be applied to other related, but different tasks. Taylor and Stone (2009) give an overview of transfer learning applied to reinforcement learning. Humans can guide this experience transfer, or the machine can learn it by itself.

We saw that the agent can ask the human to rank outcomes, trajectories, or trajectory snippets. Ranking can also be done on the transition probabilities and rewards (Bonet and Pearl, 2002; Sabbadin, 1999; Epshteyn and DeJong, 2006). This aids in knowledge transfer, as we can abstract over

states (Reyes et al., 2006), which enables us to make generalisations such as "a turn is easier to take at lower speed" (Epshteyn and DeJong, 2006) or "a translation introduces noise into the text".

Of course, how the agent learns might depend on the user and their training methods. For example, Knox and Stone (2015) focus on training an agent based on the polarity of the trainer ("encouraging", "punishing", *etc.*). Humans are naturally biased towards giving a positive reward, which favours myopic learning (where the agent prefers immediate gain to trying for larger, long-term rewards), but by converting episodic tasks to be continuous, the agent can successfully learn non-myopically. They differentiate between the *task objective*, which is the goal as seen by the user, and the *learning objective* which might be "to find a behavioural policy that maximises the expectation of the sum of the future human reward". They try to find the learning objective which allows the agent to perform well with respect to the task objective, in other words, they try to balance the objectives such that the agent behaves as the trainer intended.

Knox and Stone (2015) also give an interesting insight into how to adjust the long- or short-sightedness of the agent with regard to the rewards it expects. Should it be patient, hoping for larger rewards later, or should it set out to get the maximum gain it can straight away? The parameter that affects this is known as the discount factor γ. They claim that a higher value of γ renders the agent more robust to environmental changes that could block the MDP-optimal path, for example, whilst leaving the goals unchanged. In our case, this could be a run of documents that have no extractable events, for example, or that do not receive user feedback. At the other extreme, setting γ to zero "reduces reinforcement learning of a value function to supervised learning of a reward function". The disadvantage being that the reward function represents the optimal policy, but not the task goals, and is therefore not necessarily obtainable in "real-life". This forces the trainer to micro-manage, and is clearly unacceptable in our case. Their findings suggest that we should set γ to be high, as we want the agent to be task-focused.

Finally, in production, there will not just be one user of the system, but many, and we must take all of their feedback into account. Shelton (2000) discusses the difficulties involved in doing this, for example, how do we ensure that the users all use the same scale, what happens if the users do not agree on the feedback to be given for a particular situation, *etc.* In fact, our qualitative approach should help respond to some of these questions as there is no longer any need to normalise the scale used, and the definition of the high level requirements should encourage a consensus to be reached.

Part I

Learning to improve an information extraction chain from intuitive feedback

Chapter 3

Introduction to Part I

This part of the manuscript represents the bulk of the work done during the three years of the thesis.

We first present the challenge that we faced in the context of the CIFRE contract, that of extracting information from open-source documents for the *Open Source INTelligence* (OSINT) community using the platform WebLab, and explain a little more about the OSINT community.

We then give a rapid introduction to reinforcement learning (RL) and Markov Decision Processes (MDPs), in order to make our implementation choices clearer.

We go through the thought process of modelling a chain as an MDP, starting with a generic discussion of the information available, which should be applicable to any other chain. We then go into the specific choices that we made for our implementation.

We present BIMBO, the brains behind the operation. She is a fully configurable, modular platform, applicable to a variety of situations (we later apply her to image analysis in chapter 13, for example). She is responsible for interpreting the AIs' abstract choice of an action into a tangible one, and translating the current state of the document and system into a state that the AI understands. She measures the quality of the results found, keeps track of the time spent on treatment, and outputs logs and result files in order that the performance of the AI can be monitored.

We give a first suite of experiments which act as a proof of concept, showing that modelling an IE treatment chain as an MDP and then using a standard reinforcement learning technique to build the chain step by step is feasible.

We then move away from the standard approach of rewarding the agent with quantitative, or numerical feedback, and discuss how it could be rewarded with qualitative, or non-numerical feedback.

Finally, we implement a new algorithm, *SSB Q-learning* (Gilbert et al., 2016), and carry out a variety of tests that show that qualitative (non-numeric) feedback can be given with excellent results.

Chapter 4

The industrial challenge

As we said in section 1.1, our research is driven by the *Open Source INTelligence* (OSINT or OSCINT) community. The aim of OSINT is to extract structured information from unstructured open-source data. Steele (1995) defines OSINT as "[...] intelligence derived from public information – tailored intelligence which is based on information which can be obtained legally and ethically from public sources."

This information contributes to the knowledge of the OSINT analyst, and enables a full appreciation of the situation, allowing informed decision-making and actions. These decisions and actions depend on the domain's objectives, whether it is to seek an industrial advantage, to gather technological intelligence, or to enhance an online reputation. Even though the objectives may vary, every OSINT domain faces the same two main challenges (WebLab, 2016a):

Unstructured and complex data. There are a huge number of open source data readily available, especially on the Web, and in most cases, the content is very rich. However, it is a complex and fastidious task to gather usable information. The Web is dynamic, constantly growing and changing, the sources are multi-lingual, and are presented in many different media formats. Finally, there is often little or no cross-referencing or validation of the data.

Choice of tools. Not surprisingly, with the growth in data, many tools have been developed to try to extract the information. First, the best tool for each function must be chosen, whether it is the transcription of a video, or the translation of a document. Then these tools must be made to work together so that, for example, the video can be collected and transcribed, the transcription obtained can be translated, and the relevant information extracted from the translation. Like the Web, the tools are not static. The system must therefore be flexible, allowing treatments to be adapted, sources to be changed, and different information extraction requirements to be satisfied. Many data analysis systems therefore rely on a distributed architecture,

allowing a modular treatment of the multimedia (for example, Ogrodniczuk and Przepiórkowski (2010) give an overview of some processing chains that are provided as web services).

4.1 The platform WebLab

Our industrial application uses the open-source platform WebLab (2016c) for economic, strategic, and military surveillance. The WebLab platform addresses the two problems above, enabling coherent and flexible systems for OSINT to be built. WebLab integrates web services which are designed to be used in a modular fashion. They can be parametrised and chained together, interchanged or permuted to create a treatment chain for the analysis and extraction of information from web pages and other open-source multimedia. These open-source multimedia are crawled by the information retrieval (IR) tools (which are out of the scope of this thesis) and fed continuously into the chain of distributed web services which carry out modular treatments, such as transcription, translation, extraction and presentation of the results. Each service analyses the contents of the resource it receives and enriches it with annotations (see Figure 4.1). Finally, the results are stored for the analyst to consult.

Every time WebLab is used in a project, the services are tailored to suit the specific user domain, e.g. maritime, technological intelligence, *etc.* As a consequence, there are already many different proprietary versions of the services which could be used as "black boxes" in the treatment chain.

A typical WebLab document treatment chain (Figure 4.2) is based on open-source services available from WebLab (2016b).

- The document is passed to the normaliser Tika (2015) which as well as converting it to XML format, also uses the Apache Tika™ toolkit to detect and extract metadata such as its format, length, file name, date of collection, *etc.* and the structured text content.

- The resource is then passed to the language detector NGramJ (2015), which uses ngrams of characters to determine the language of a character sequence. Confidence scores between '0' (the text is definitely not written in this language) and '1' (the text is definitely written in this language) are returned for each portion of text.

- Next, an extractor encapsulating GATE (2016) extracts the named entities.

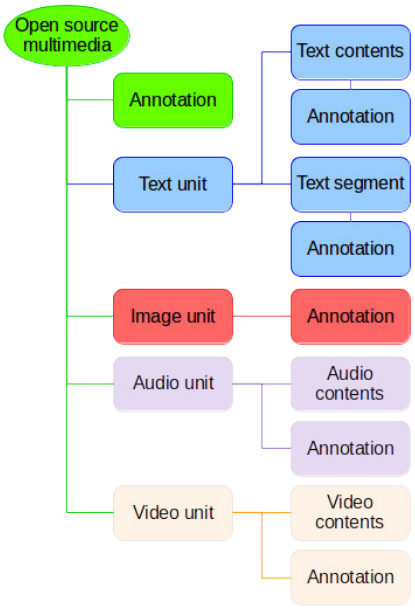


Figure 4.1 – A multimedia source is broken down by its contents - text, image, audio and video. Annotations are added to each part and sub-part, as well as to the resource as a whole.

- Finally, the results are stored in a database accessible by the analysts.

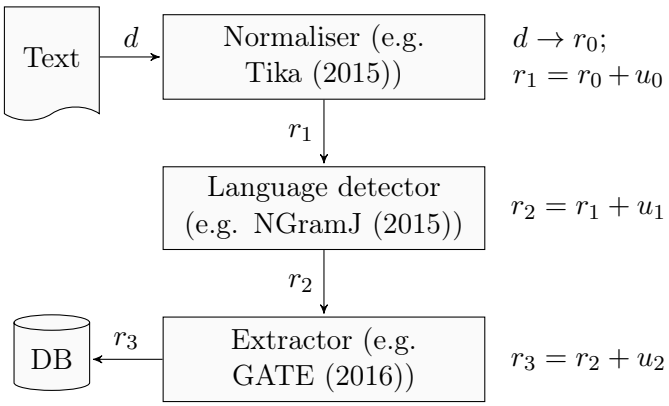


Figure 4.2 – A simple but typical Weblab chain: a document, d, is converted into an XML resource, r_0, and annotations, u_i, are added by each service. Finally the results are stored in a database for user consultation.

Example 1 *Consider the following document:*

> *4/29/1971: In a series of two incidents that might have been part of a multiple attack, suspected members of the Chicano Liberation Front bombed a Bank of America branch in Los Angeles, California, US. There were no casualties but the building sustained $1 600 in damages.*

The simplified XML resource in Figure 4.3 is produced after passing the document through a normaliser, adding the annotations "text/plain" (line 12) and original content (line 19); and a language detector, adding the language "en" (line 16).

4.2 Who are the analysts?

So far, we've written about "the analysts" as though they were a single, specific group of people using the same system. In fact, they come from many different walks of life, and use many variations of the application which have

```
<resource type="Document" uri="weblab:aaa">
   <annotation uri="weblab:aaa#a0">
       <wp:originalContent resource="file:weblab.content"/>
       <wp:originalFileSize>255</wp:originalFileSize>
       <dc:source>documents/event.txt</dc:source>
       <wp:originalFileName>event.txt</wp:originalFileName>
       <dc:modified>2015-02-14T19:52:21+0100</dc:modified>
       <wp:collected>2015-02-10T00:11:00+0200</wp:collected>
   </annotation>
   <annotation uri="weblab:aaa#a1">
       <wp:isProducedBy resource="weblab:tika"/>
       <dc:format>text/plain</dc:format>
   </annotation>
   <annotation uri="weblab:aaa#a2">
       <wp:isProducedBy resource="weblab:ngramj"/>
       <dc:language>en</dc:language>
   </annotation>
   <mediaUnit type="wl:Text" uri="weblab:aaa#0">
     <content>4/29/1971: In a series of two incidents that might have been part
      of a multiple attack, suspected members of the Chicano Liberation Front
     bombed a Bank of America branch in Los Angeles, California, US. There were no
      casualties but the building sustained $1 600 in damages.</content>
   </mediaUnit>
</resource>
```

Figure 4.3 – Simplified XML of a WebLab resource mid-chain, showing annotations added by Tika and NGramJ.

been developed specially for them. The OSINT community is large and widespread, and their requirements varied. We can, however, describe a "typical" OSINT analyst here.

Firstly, they seek information from open-source multimedia to build a picture of a current situation. They will then recommend actions based on this information. With the advent of the internet, huge volumes of open source multimedia are freely available, and the analyst has to find pertinent information within the wealth of data.

This information can be extracted in a number of different ways, but depends, obviously, on the quality of the source documents (which depends on the quality of the IR tools). Extractions can therefore be erroneous, contradictory or missing. The analysts are therefore typically forced to correct *a posteriori* the extractions using their expert knowledge and other reliable sources to fill in the gaps. This is a distracting, onerous and repetitive task, and although these corrections are stored in the database, they are currently never fed back into the extraction process. This wastes, not only the analyst's time, but also their knowledge, as the system never learns from it.

The way in which the extracted information is presented to the analyst varies depending on the user domain and their particular system. In the open-source system, it is in the context of the original document. In Fig-

ure 4.4, for example, we see the named entities highlighted, and the user adding an organisation *dublincore* (the edit and delete screens are similar, not shown). Figure 4.5 shows an example of a tooltip, allowing the user to access any extra information that is stored in the database and connected to that entity by hovering over it.

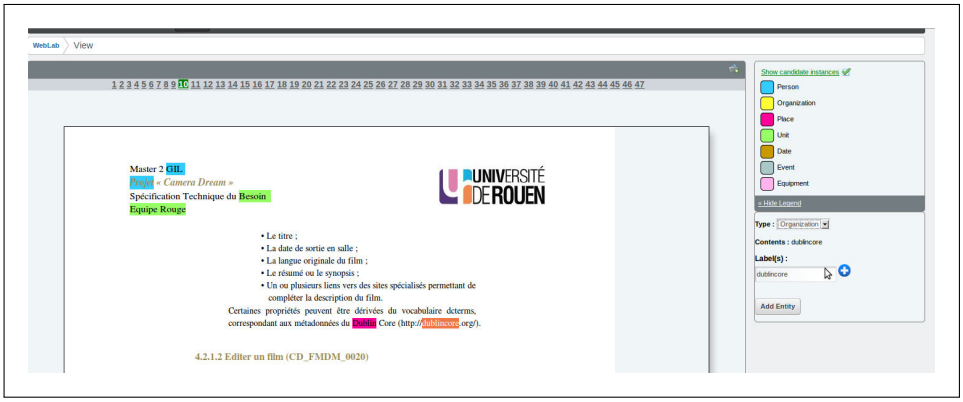


Figure 4.4 – Adding the organisation dublincore using the open-source interface.

Figure 4.5 – An example of a tooltip giving further information about an entity using the open-source interface.

Unfortunately, the current open source interface is not adapted to present the extracted events to the user. Ideally, the interface would present the information gathered as an event summary over its four dimensions - conceptual, temporal, spatial and agentive. These summaries would be indexed by

each of those dimensions, allowing the analyst to easily retrieve events that they were interested in, for example, of just one type (with the same conceptual or semantic dimension) over a certain time period, or which involve a certain agent.

Neither is an open-source fusion service available to aggregate event information gleaned from several documents. Again, ideally, if such a service were made available, each event summary should contain contextual links to the original documents from which the information was obtained, so that the analyst could easily refer back to the sources.

Given such an interface, the analyst would be able to edit the event by adding, changing or deleting part or all of the information for one or more of the dimensions. For instance, they might complete a partially extracted date, remove an incorrect geographical reference, or correct the name of one of the agents. They would also be able to correct errors made by the fusion service by merging two or more events, or by separating a single summary sheet into several events. Once they were sure that the event summary sheet, or part of it was correct, they would validate it. The analyst would also be able to see at a glance which fields had already been validated (by them, or another analyst).

Behind the interface would be an action tracker, registering, for each event, what changes were made by the analyst, which information was validated, or even if no changes were made after a summary sheet was consulted. This action tracker would link those actions (or lack of them) to the document, and hence to the treatment that the document received. This would enable feedback to be collected on the quality of that particular treatment.

The development of this ideal interface is outside the scope of this thesis, but we discuss the mechanics that we might put behind it in the following. In section 7.6, we go into more detail as to how we chose to interpret edits as a quality measure. We discuss in section 6.3 how we might use the analyst's actions to reward the AI, and in chapter 8 and chapter 10 we describe how we implemented this feedback in our tests.

4.3 Challenge summary

Although we presented a very simple chain above, in production the treatment chain is complex. It is written and calibrated by experts who choose the services making up the chain, their order, and their parameters. The service order is fixed, but can be conditional, (*e.g.*, "if the document is in English, send it to *event-extractor*, and to *translator* otherwise"). The system experts

can offer the analysts some help by fixing errors in the web services, or by developing new ones to fulfil new requirements. They can also change the chain in consultation with the analysts. Despite their joint expertise, it is difficult to create the perfect universal chain, as the treatment of open-source documents gathered from the web means that: their format and contents are not standard, the source pages themselves are not controlled, URLs change or are pirated, and there is "noise" (*e.g.*, adverts). The right web services have to be called in the correct order and supplied with the best parameters, and even then may or may not extract useful information from a document.

Undetected, and partially or falsely extracted information (*e.g.* unconnected words in the same sentence erroneously associated) provoke errors that have to be corrected. For example, imagine that the analyst wants information on aerial bombings. We saw in section 1.1 that it is impossible to say with certainty that the word "bombardment" refers to this. The analyst can see that the page is a school blog only once the event has been extracted. Even from a specialist web page, it is possible that the word refers to the incessant questions of the journalists, and not an attack. Maybe the synonym "shelling" was used, and the event is not recognised. With these uncertainties, the experts who configure the chain try to envisage the most common situations. It is inconceivable that they construct individual chains by examining each source document.

The existing chains are therefore open-loop systems. They follow a preprogrammed set of instructions based on anticipated outcomes to reach their goal. They cannot take into account the results of their actions, nor do they have the capacity to vary these instructions in unexpected or unusual situations. This can be compared to planning a drive between London and Manchester, programming how long and how hard to press the accelerator, how many degrees to turn the steering wheel, *etc.*, and then blindfolding the driver. This is fine if there's no other traffic, no diversions, or unexpected red lights...

We want to remedy this situation by making the chain a closed-loop system, where the treatments are based not only on the overall plan, but also take into account current information, and feedback on the extent to which the final goal is achieved.

We want to provide a mechanism which learns to modify its behaviour in real time, continuously improving the chain, and reducing human effort by decreasing the error rate. This mechanism receives feedback obtained as the analysts consult the system output. In other words, we allow the chain to "learn from its mistakes". For instance, that if the document is a well-written scientific article in French, a translation to English before the

extraction works well to make the most of the wealth of English extraction rules available, but for *blogs* or *tweets* containing slang, a direct extraction is preferable as the dictionaries are insufficient, and a translation only introduces more noise. The services, or building blocks for the chain, will thus remain a "black box" for the analyst, allowing them to distance themselves completely from the extraction process.

We need to collect the feedback non-intrusively, that is, in a way that is invisible to the analyst without impacting on his or her work. A user's expertise can be modelled by tracing his actions (Bratko and Šuc, 2003). In production, these action traces could be captured through the graphical interface that we spoke about in section 4.2 which highlights the extractions in the original document. We can then base the feedback on the analyst-system interactions that currently occur. For example, an event might be *corrected* (*e.g. a named entity might be added*), in which case we can deduce an explicit feedback based on the corrections that the analyst has made, that is, a distance between the event as corrected and that which was extracted. Indeed, our first set of tests in chapter 8 simulate this, in that the agent receives a numerical feedback as an automatically generated judgement on the quality of extractions.

The difficulty is that an event may also be *not extracted*, *extracted in too long a time*, *etc.* To interpret these "non-actions" as feedback, we must first answer questions such as "Is it preferable to extract something erroneously, or to miss an extraction?", "Should I sacrifice speed for quality, or vice versa?".

These questions cannot be answered naturally with a numerical response; it is not easy to determine that "A partial extraction is worth 10.5 times more than a complete extraction" or "Going 2 seconds faster is worth 3.7 times more". It is therefore essential that we progress to giving intuitive feedback (chapter 9), based on naturally expressed user preferences, such as "I prefer an extraction to no extraction" and "I prefer it to be as fast as possible". The requirements can thus be gathered very easily, and can be changed if necessary in an intuitive fashion.

We are not seeking to create a clone of the analyst, but rather to capture their hard-earned skills, knowledge and expertise. This would have a double benefit. First, it would allow the system to self-improve with each human-machine interaction, thus lightening the human analysts' workload. Secondly, analysts are expensive and increasingly rare, which makes it increasingly important not to waste their time and expertise. We could aid the transfer and perpetuation of their knowledge; when one person leaves, the next can "inherit" the knowledge of the previous one.

We faced certain operational constraints when trying to formally define

the feedback we require from the operator. For example, an analyst could reasonably be expected to spend up to an hour each day initially providing feedback, but only if he sees a real improvement in the results. Also, there is a risk of the analyst unintentionally introducing misleading feedback through his knowledge and expertise. For example, even if the extraction was perfect based on the information provided in the source document, if that information was faulty or ambiguous, the analyst will see and correct an error. The headline "Nice attack in July 2016" in a sports article probably means that the team in question played well that month, but to the system, it could refer to the 2016 terrorist attack in Nice, France, leading to an extracted event which the analyst would delete. Another example could be a mistake in the reported date due to a typographical error, which again, would be extracted correctly according to the information given, but which would need to be changed by the analyst. Next, we cannot ask the analysts to compare the extraction quality on two documents, as this is not part of their normal routine, and would be cognitively difficult. Finally, as we said above, the existing open-source interface is not adapted for the presentation and modification of event summaries, nor for the collection of user actions. Given these restrictions, and the operational impact of asking the users to correct our test output, in our experiments, we simulated our "user feedback".

Chapter 5

Reinforcement learning

Part of our contribution is to model the treatment of a document by the chain as a Markov Decision Process (MDP), and its improvement as a reinforcement learning (RL) problem. Before we go into exactly how we did that in chapter 6, it is probably a good idea to explain what RL and MDPs are, so that the connection can been made more easily. If you are already familiar with the concepts, then you are more than welcome to skip directly to chapter 6...

If you are still reading, then please note that although we try to give a self-contained overview below, we highly recommend reading Sutton and Barto (1998) for an in-depth introduction to the topic.

Although reinforcement learning forms part of the family of machine learning, it has also been considered a very natural way to model human learning, but the assumption is usually that humans will always learn faster than machines. Doshi-Velez and Ghahramani (2011) refute this theory, and make an interesting comparison of human and agent reinforcement learning in partially observable domains. They found that agents are more tenacious, and have more endurance than the human learners. This was shown to give them an advantage in learning, as they are not discouraged by early failure, and persist in refining their solutions from good enough to perfect. Shteingart and Loewenstein (2014) support this finding, claiming that human learners cannot have a precise and exact view of the states and actions comprising their environment. They are likely to over-complicate the scenario, make false assumptions, ignore instructions and even develop superstitious behaviour. This is one of the difficulties that we face in trying to define the environment for the agent - making it specific enough that the agent can infer, correctly, the causes and effects.

One of the most common scenarios in RL is that an agent tries to learn from trial and error interactions with its environment what behaviour is optimal. In other words, it is not given an explicit solution to a problem, but must use observations of its environment to establish how it should act in different situations to maximise a reward over time. The set of situations or *states* and the appropriate action to take in each forms a "strategy", or

"policy". This optimal set of one action for every state can be hard to find, not least because the effects of an action may have short-, mid- and long-term effects. For this reason, the gains are often estimated as an expected cumulative reward - "If I do this now, I don't get an immediate reward (in fact, it may cost me something), but I can see that in the future, if I follow this plan, then I should probably get a larger reward".

This word "probably" is important. The actions that the agent takes do not lead neatly from one state to another in the same way each time (otherwise it wouldn't be interesting), instead, they are stochastic. That is, from a given state A, if the agent performs a given action, then maybe he'll end up in state B, maybe state C, or maybe he'll just stay in A. He therefore has to try to learn with what probability he'll move to a given state, and use this when calculating his reward.

Given that actions may cost the agent (he might use fuel, for example, or be on a time-critical mission), he also has to weigh up the benefit of going for immediate rewards against the cost of trying for higher rewards later.

We can summarise this in Figure 5.1. The learner receives rewards, based on the results of its chosen actions. The closer the results are to the goals, the higher the reward. The learner tries to maximise these rewards, typically by *exploiting* current knowledge to continue to receive good rewards, or by *exploring* new actions with the hope of obtaining even better ones. It chooses its actions based on its observations of the environment.

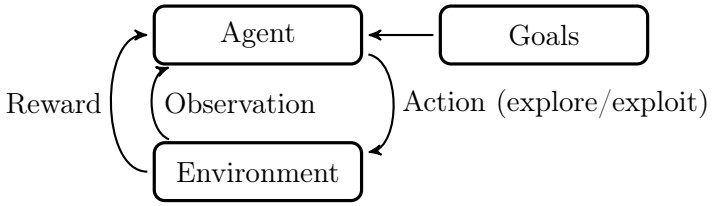


Figure 5.1 – The agent acts according to its observation of the environment, and receives a reward based on how close it is to its goals.

5.1 Markov Decision Processes

Reinforcement learning is generally formalised as a Markov Decision Process (MDP, Puterman (1994)), which models the environment in terms of *states*, in which *actions* are possible, leading to other states stochastically. The fact that the environment is in a given state at a certain instant bestows

an immediate reward on the learner (or agent). The agent's objective is to choose actions such that it maximises its expectation of cumulated rewards.

Definition 2 (MDP) *A* Markov Decision Process (MDP) *is a quintuplet* (S, A, P, R, γ) *where:*

- S *is a (here finite) set of possible states in the environment;*
- A *is a (here finite) set of actions (available to the agent);*
- P *is a set of distributions* $\{P_a(s, \cdot) \mid s \in S, a \in A\}$ *;* $P_a(s, s')$ *is the probability that the environment is in state* s' *after the agent performs action* a *in* s*;*
- R *is a reward function, defined on the states;* $R(s)$ *is the reward obtained by the agent when it reaches state* s*;*
- $\gamma \in [0, 1]$ *is a discount factor, weighting expected future rewards against those currently expected.*

In RL, the agent initially only has knowledge of the state / action space $S \times A$, as well as the factor γ. The probability distribution is unknown, which means that it has to learn to "navigate" between the states. For instance, from a given state s_0, if it takes action a_0 ten times, it might arrive in state s_1 3 times out of 10, in state s_2 6 times out of 10, and in state s_3 only once. This means that it would learn that $P_{a_0}(s_0, s_1)$ is 0.3, $P_{a_0}(s_0, s_2)$ is 0.6 and $P_{a_0}(s_0, s_3)$ is 0.1. Note that this knowledge only reflects the agent's experiences, and not necessarily the real probability function – it may just have got lucky (or not). It also has to find out what reward is given for each state, *i.e.* the reward function, and which path of actions it therefore expects will probably give the best results.

More formally, at each instant t, it knows the current state s_t of the environment, and chooses an action a_t. The environment passes into state s_{t+1} according to the probability distribution $P_{a_t}(s_t, \cdot)$, and the agent is informed of the state s_{t+1} and the reward $r_{t+1} = R(s_{t+1})$. The process continues in s_{t+1}.

The agent, as it interacts with the environment, learns a series of *policies* $\pi_0, \pi_1, \ldots, \pi_t, \ldots$, where a *policy* $\pi_t : S \to A$ gives, at instant t, the action $\pi_t(s)$ to perform if the current state s_t is s. Its goal at each moment is to maximise the expected accumulated reward, where the expectancy is taken over the trajectories generated by the policies $\pi_0, \pi_1, \ldots, \pi_t, \ldots$, that is, the expected quantity $\sum_{t'=t}^{\infty} \gamma^{t'} R(s_{t'})$. This generic framework encompasses many variations (for a recent summary, see Szepesvári (2010)).

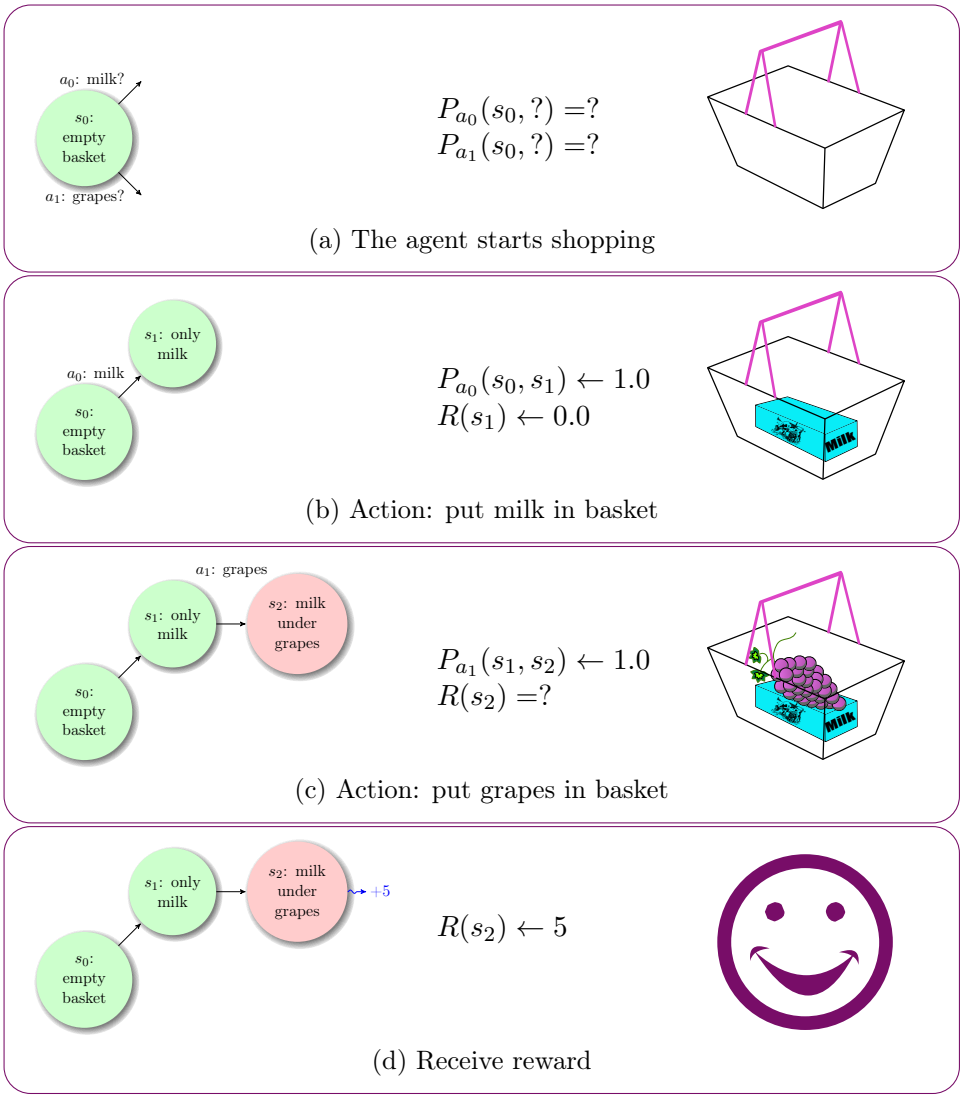
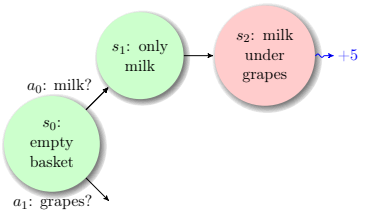
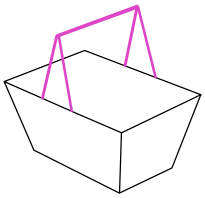


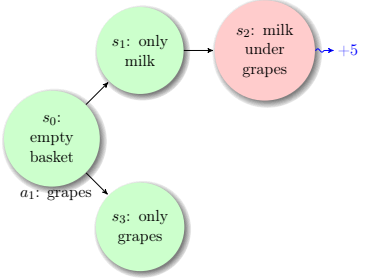
Figure 5.2 – Shopping for grapes and milk - first week.



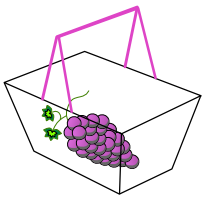
$P_{a_0}(s_0, s_1) \leftarrow 1.0$
$P_{a_1}(s_0, ?) = ?$



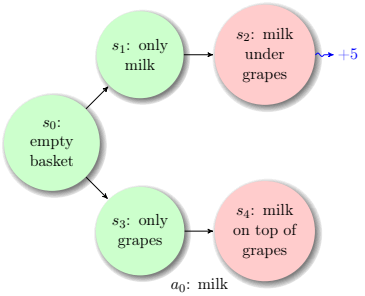
(a) The agent starts shopping again



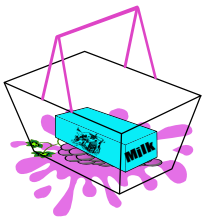
$P_{a_1}(s_0, s_3) \leftarrow 1.0$
$R(s_3) \leftarrow 0.0$



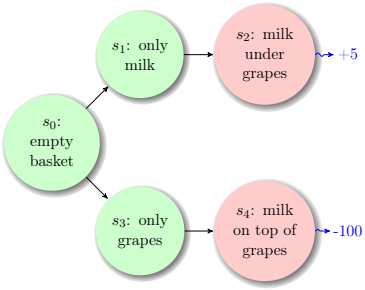
(b) Action: put grapes in basket



$P_{a_0}(s_3, s_4) \leftarrow 1.0$
$R(s_4) = ?$



(c) Action: put milk in basket



$R(s_4) \leftarrow -100$

(d) Receive punishment

Figure 5.3 – Shopping for grapes and milk - second week.

Example 3 *As a tiny example of how we might model a process as an MDP, we sent a robot shopping for grapes and milk. Figure 5.2 shows what he learnt the first time, and Figure 5.3 what he found out the second time.*

- *Figure 5.2a: He starts in state s_0 with an empty basket, and the choice of two actions - put the milk in the basket (a_0) or put the grapes in the basket (a_1). He has no knowledge yet of what the consequences might be of each action, as he has currently no idea of the probability distribution or the reward function.*
- *Figure 5.2b: He chooses a_0 randomly, and puts the milk in the basket. He's now in state s_1,* i.e. *there's just milk in the basket. He hasn't finished the shopping, so isn't rewarded yet. He now knows that the probability $P_{a_0}(s_0, s_1)$ is 1.0 and the reward $R(s_1)$ is 0.0.*
- *Figure 5.2c: From s_1, he puts some grapes in the basket (action a_1), and is now in state s_2: milk under grapes, and now $P_{a_1}(s_1, s_2)$ is 1.0.*
- *Figure 5.2d: He receives a reward of +5, and adds to his knowledge that $R(s_2)$ is 5.0 (depending on the RL algorithm used, some of this reward might be back-propagated, so that he can update $R(s_1)$ based on a proportion of 5.0).*
- *Figure 5.3a: Next week, we send him shopping again. He starts in state s_0 with an empty basket, and the same choice of a_0 or a_1.*
- *Figure 5.3b: He chooses action a_1 this time, and puts the grapes in the basket. He's now in state s_3, with just grapes in the basket. As before, he hasn't finished the shopping, so isn't rewarded yet. He now knows that the probability $P_{a_1}(s_0, s_3)$ is 1.0 and the reward $R(s_3)$ is 0.0.*
- *Figure 5.3c: From s_3, he puts some milk in the basket (action a_0), and is now in state s_4: milk on top of grapes, and now $P_{a_0}(s_3, s_4)$ is 1.0.*
- *Figure 5.3d: Bad robot! The milk has squashed the grapes, and there's grape juice running everywhere, so he receives a punishment of -100, and adds to his knowledge that $R(s_4)$ is -100.0.*

Of course, the probability $P_{a_0}(s_0, s_1)$ is not really 1.0. Maybe 10% of the time he drops the milk on the floor, which would put him in state s_5 (milky floor). In this case, after a lot of shopping expeditions to remove the effect of chance, he should learn that $P_{a_0}(s_0, s_1)$ is 0.9 and $P_{a_0}(s_0, s_5)$ is 0.1. Finally, we hope that he would learn the policy $\pi(s_0)$ is a_0 and $\pi(s_1)$ is a_1, i.e. *to always put the milk in first, followed by the grapes.*

5.2 Q-Learning

Numerous algorithms exist to solve RL problems, and one of the simplest to understand and use is called *Q-Learning* (Watkins, 1989). The few parameters that it takes can be set intuitively, and the results can be observed and interpreted easily. It was for this simplicity that we chose it for our proof of concept tests. We wanted an algorithm where any external side-effects, for example due to unfortunate parameter choices, were reduced to a minimum.

We give a brief overview of the Q-Learning algorithm (Algorithm 1) here to keep this document self-contained, but encourage you to read the original article.

Algorithm 1: Q-learning

Data: MDP $\mathcal{M}$

while *True* **do**

  Choose a_t in s_t using the EG exploration strategy

  Play a_t, observe s_{t+1}, and let $r_{t+1} = \mathcal{R}(s_{t+1})$

  $\hat{Q}_{t+1}(s_t, a_t) \leftarrow$
  $\hat{Q}_t(s_t, a_t) + \alpha_t(s_t, a_t)(r_{t+1} + \gamma \underset{b}{max}\{\hat{Q}_t(s_{t+1}, b)\} - \hat{Q}_t(s_t, a_t))$

Q-learning maintains, for each state / action pair (s, a), a value denoted $\hat{Q}(s, a)$ which represents the agent's current estimate of the expected cumulative reward if from s it executes a, then follows an optimal policy.

It is controlled using an ϵ-greedy (EG) strategy: when the agent is in state s_t at time t, it chooses a "greedy" or optimal action a_t which maximises $\hat{Q}(s_t, a_t)$ (it *exploits*), except with probability ϵ, when it chooses a random action (it *explores*). Then, upon observation of the new state s_{t+1} and immediate reward r_{t+1}, it updates $\hat{Q}(s_t, a_t)$:

$$\hat{Q}_{t+1}(s_t, a_t) \leftarrow \hat{Q}_t(s_t, a_t) + \alpha_t(s_t, a_t)(r_{t+1} + \gamma \underset{b}{max}\{\hat{Q}_t(s_{t+1}, b)\} - \hat{Q}_t(s_t, a_t))$$

where:

- the *learning rate* $\alpha \in [0, 1]$ fixes the importance of the latest experience $(r_{t+1} + \gamma\, max(\dots))$ with respect to the experience already gained (that is, the previous value of $\hat{Q}(s, a)$);
- the *discount factor* $\gamma \in [0, 1]$ is as in Definition 2 of an MDP on page 35, *i.e.* it determines the importance of long-term gains in comparison to short-term ones.

5.3 Other applications of reinforcement learning

One of the currently blossoming fields in reinforcement learning is that of "deep reinforcement learning". Popularised by Google DeepMind team (Mnih et al., 2015), they have already applied it successfully to learning how to play Atari Games (Mnih et al., 2013). They designed a variant of the Q-Learning algorithm described above, called *Deep Q-Learning*, and approximated the Q-values (which they refer to as the "action-value" or "Q-function") with a neural network.

Of course, reinforcement learning is not just applied to silly examples such as robots going shopping, or playing 1980's video games. It can be applied in much more complex situations, where the environment observed does not just depend on the actions taken, but also on external circumstances, and other agents' actions.

For instance, INVESTOPEDIA (2016) defines *algorithmic trading (automated trading, black-box trading, or algo-trading)* as "the process of using computers programmed to follow a defined set of instructions for placing a trade in order to generate profits at a speed and frequency that is impossible for a human trader."

If you buy or sell shares, then you have an impact on the market. Very simplistically, and making gross generalisations, if you buy a huge amount of a given share, then you should push its price up because people think that if you want that many, you obviously know that those shares are good. Conversely, if you sell a vast quantity, then the price will fall, as everyone will panic and start to sell their shares in a snowball effect. Assuming that the world is constant around you, then you should be able to manipulate the stock market by buying and selling to make yourself a profit. The problem is, not only are other people trying to do the same thing, but there are also external factors which influence the share price. A concern that a product causes cancer would certainly make shares in that product's manufacturer fall. The announcement of breakthrough in reducing production times will probably bounce the price up. By using algorithmic trading, companies hope to benefit from the buying / selling effect, without the annoying side-effects caused by the outside world.

Of course, the algorithms are not just fixed sets of instructions, normally there is a learning process behind them, and for those of you looking to make your fortunes by playing the stock market using Deep Q-Learning, Nydal (2016) gives a nice example of how to use reinforcement learning on existing stock market data.

Chapter 6

Modelling a chain as an MDP

Given that (as we saw in section 4.3) it is impossible to create a unique chain capable of perfectly treating every type of document, the ideal would be to have a tailor-made chain for each one.

Formalising this treatment chain and improving the extractions, however, is not a trivial problem. The only information known before starting the treatment chain consists of the available services, their parameters, and the potential characteristics of the document, XML resource and system (see chapter 4). The agent knows neither the form, nor the content of the documents in advance. The ideal treatment for each document is unknown and must be learnt. For example, it depends on the language in which the document is written, its source, *etc.*, which are only discovered during the treatment. It does not even know the proportion of documents written in each language, coming from each source, *etc.* As with the "milky floor" in Example 3, it can only learn these proportions through treating many documents. It does not know if the document contains information nor even if an extraction is possible with the given tools. Neither does it know the interfaces (inputs and outputs) of the services. Its actions are thus taken under uncertainty. As we saw in chapter 5, this sounds rather like the definition of reinforcement learning and MDPs. Part of our contribution, therefore, is to model the treatment of a document by the chain as a Markov Decision Process (MDP), and its improvement as a reinforcement learning (RL) problem. We explain first how we modelled the states and actions, and then we tackle the thornier problem of how to reward the agent.

6.1 Choice of states

Our initial ideas for state attributes were inspired by the possible annotations that are added to the WebLab resource by the services in the treatment chain (see Table 6.1). Each annotation contains information about the document, and about the extracted events and named entities. As the time taken to process a document is also important, we wanted to include system information such as time taken in the definition of a state too.

Annotation	Description and example
refersTo	Gives the type of the event or named entity and its unique reference identifier (uri), *e.g.* DeathEvent#1
label	The plain text of the event or named entity (the subject of refersTo), *e.g.* member of HAMAS
name	The geonames location of the event, *e.g.* Federation of Bosnia and Herzegovina.
involves	The people and / or places involved in the event, *e.g.* Unit#philippine_national_police,
spatial	The gps coordinates associated with the event, *e.g.* 33.0,44.0
hasSpatialThing	The geonames reference, *e.g.* http://sws.geonames.org/3351879/
peopleEmployed	People associated with the event who are employed at orgEmploys, *e.g.* Person#arthur_khoza
orgEmploys	The employer of peopleEmployed, *e.g.* Unit#google
parentFeature	The geonames parent reference, *e.g.* http://sws.geonames.org/6447142/ (could be used for inference - Marseille to France)
date	Dates associated with the event, *e.g.*1998-11-11
takesPlaceAt	Location of the event extracted from document, *e.g.* Place#ain_tagourait
isProducedBy	Name of the service which produced this annotation, *e.g.* normaliser/tika
content	The original content of the document converted to text, *e.g.* 11/12/1998: The 1 October Anti-Fascist Resistance Group was suspected of bombing an employment agency in Madrid, Spain, causing considerable damage, but no injuries.
type	The type of event or named entity, *e.g.* SurveillanceOperation
isCandidate	Whether this named entity been validated by an analyst yet, *e.g.* false
language	The identified language of the input document, *e.g.* en
hasOriginalFileSize	File size of input document, *e.g.* 339
format	Format of input document, *e.g.* text/plain

Table 6.1 – Potentially useful annotations to use when defining the state characteristics.

We therefore defined the states using the characteristics of the original document and of the resource and system at a given moment. The system thus perceives the task as the *states* of a process, and each passage through a service modifies the current state.

More precisely, the states are formed from the values of a certain number of *descriptors* of the documents and of the system. The states that the system perceives are thus combinatorial states s_i which can be defined at time t in terms of the features of the resource and the current state of the treatment chain. If we say that the document and resource properties are represented by the features X_i and the system properties by the features Y_i then we could represent some states as in Table 6.2.

Table 6.2 – A possible representation of states given document features X_i and system features Y_i, where " " indicates that the information is not (yet) available.

	X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3
s_1	en	*www.anyschool.co.uk*	true	school blog	10	3	6
s_2	en	*www.terroristweekly.com*	true	specialist	9	10	12
s_3	fr	*www.anyuniversity.fr*	false	" "	12	7	3

For instance, X_1 is the detected language that the document is written in; X_2 is the source of the document; X_3 indicates whether the type of document has been extracted; X_4 is the type of document; Y_1 is the time taken so far in seconds; Y_2 is the number of services already called, Y_3 is the number of parameters changed.

These states allow a generalisation of the learning; we have already seen that the *type* of source web page (school blog *vs.* specialist) could greatly influence the utility of the word "bombardment" for the extraction of information. We might hope that the system would learn from the user interactions that:

- if the current state has the value "true" for the descriptor "typeExtracted" and the value "school blog" for the descriptor "type" (as in s_1 in Table 6.2), then the best action to perform consists of stopping the treatment (useless to try to extract bombings from school blogs);
- if the type is extracted and is not "school blog" (as in s_2 in Table 6.2), the best action consists of passing the document to an extraction service, using the word "bombardment" among the trigger words;

- otherwise (as in s_3 in Table 6.2), the best action consists of passing the document to a type-recognition service.

Obviously, the agent will never "see" the resources nor the documents. Rather, it will see a document transformed through a treatment chain as a series of projections of resources, that is, as a series of vectors of values. If we want the agent to be able to generalise, then it is very important that the features are abstract enough for similar values to occur with a reasonable frequency, but not so abstract that they do not help the reasoning. For example, a resource feature indicating whether the document contains the name "Elizabeth II" would certainly be too precise, whereas one indicating that the document contains text would be too vague. A resource feature indicating that the document contains the name of at least one monarch, on the other hand, could be more useful.

> An astronomer, a physicist and a mathematician are on a train in Scotland. The astronomer looks out of the window, sees a black sheep standing in a field, and remarks, "How odd. All the sheep in Scotland are black!" "No, no, no!" says the physicist. "Only some Scottish sheep are black." The mathematician rolls his eyes at his companions' muddled thinking and says, "In Scotland, there is at least one sheep, at least one side of which appears to be black from here some of the time." (Stewart, 1995)

The features may be directly read from the annotations (*e.g.* the language detected), or may be derived (*e.g.* the number of royal family members identified). We also wanted to include a feature which could be defined by the analyst in order that he receives personalised extractions, for example, the type of event that they are particularly interested in.

We therefore decided to represent a state by an attribution of the following characteristics:

- the information already extracted (if any), where " " indicates not (yet) extracted:
  - *language* $\in$ {"en"," "};
  - *format* $\in$ {"*text/plain*"," "};
  - *interesting* $\in$ {*true, false*} (*true* if at least one event of the type that interests the analyst has been extracted);
  - *any* $\in$ {*true, false*} (*true* if one or more events of any type have been extracted);

- the parameters chosen (see section 6.2 below for an explanation):
  - *nounGazetteer* (the noun gazetteer currently chosen, if any);
  - *verbGazetteer* (the verb gazetteer currently chosen, if any);
- the system state:
  - $nbServices \in \{0-5, 6-20, 21+\}$ (the number of actions the AI has already taken, *i.e.* the number of services through which the current document has already passed plus the parameter changes. We explain in section 6.2 just below why these ranges were chosen);
  - *timeTakenSoFar* (number of seconds since the current document treatment started in *timeInterval* (parametrisable) steps).

Obviously, with access to the appropriate web services able to extract the information, more characteristics could be taken into account, such as

- document metadata: length, number of characters, number of paragraphs, number of words, number of pages, number of images, year, keywords, bibliography, citation, complete list of languages;
- document quality: syntax quality (*e.g.* number of spelling mistakes), complexity, subjectivity;
- source type, *e.g.* tweet or thesis, popular or specialised;
- theme, *e.g.* sport, science, philosophy.

It should be noted that the features that we detailed above were chosen in the context of the CIFRE contract, given the industrial application that we use. However, the system is completely modular, and not at all dependent on a given state definition as we show, for example, in chapter 13 where we define a state in terms of the parameters of image analysis services.

6.2 Choice of actions

The system also has a certain number of *actions* available which it can apply in the current state.

A natural choice is to offer the AI the choice of the next service to call, or to stop the treatment and offer the results to the analyst. This reinforces our choice of an MDP to model the treatment, as it takes into account probabilistic uncertainty (the results of calling a service on a resource are

not known for certain in advance), and the services take the system from one state to another. For example, if 70% of the source documents are in English (and the language extractor is completely reliable), the agent will perceive that taking the action "detect language" in a state s_t leads with probability 0.7 to a state s_{t+1} similar to s_t except that it contains the information "language detected" and the annotation "en".

However, building an efficient treatment chain depends not only on choosing the services but also on setting their parameters correctly. For instance, the extractor service encapsulating GATE (see section 4.1 on page 24) that we use relies on *gazetteers*, which are lists of nouns and verbs triggering the detection of a specific type of event (*e.g.* the *bombing* verb gazetteer contains "bomb", "explode", "detonate", *etc.*). GATE is therefore a *parametrised service*; choosing the correct gazetteers for it to use is part of building an efficient treatment chain, as without them it is ineffective.

One of our initial ideas was to dynamically construct these gazetteers using online thesaurus services. This proved technically problematic for two reasons. First, the online services restrict the number of queries made on them, which would have significantly reduced our test capabilities. Second, the gazetteers contain not only the event detection trigger words, but also the grammar rules (JAPE (2016)) applicable to each word. These rules are difficult to construct, even for a native speaker, let alone an AI (Chiticariu et al., 2013). We therefore took the gazetteers already defined in production.

The available actions are therefore to choose the next service from {*Tika, NGramJ, GATE*}, to choose a GATE gazetteer, or to *STOP* the treatment and return any extracted events. This means that in an optimal chain, the AI would choose two gazetteers (one list of verbs, and one of nouns) plus three services (*Tika, NGramJ and GATE*) giving five actions, before choosing to *STOP*. The ranges given for *nbServices* in the state therefore reflect an "optimal" number of actions (0–5), a "slightly sub-optimal" number of actions (6–20) and "far too many" actions (21+).

Note that we treat the services as "black boxes" so that knowledge of, *e.g.*, their WSDLs (Web Service Definition Language) is unnecessary. Of course, any such extra information could be used to increase the efficiency and robustness of the learning process and we discuss this in section 14.2: Future work.

The repetition of the same action on the same document is technically authorised, but will penalise the system, as the treatment time will be longer, and due to the massive volumes being treated in production (potentially all possible documents from the web) faster results are preferred.

6.3 Rewarding with user feedback

Solving an MDP with RL requires that the agent be rewarded for its actions. The closer it is to achieving its objectives, the higher the reward. In our case, only the analysts can specify the objectives, and hence the reward, using their skill and expertise to judge how close the agent came to perfection. In other words, their feedback on the treatment of a document will reward the agent, and reinforce its behaviour.

In section 2.5, we saw that Bratko and Šuc (2003) raised the question of reconstructing the skill of an operator.

They highlight the difficulty of engaging the operator in a dialogue, trying to formalise his or her skill. The operators are usually not capable of describing exactly what they do, and any description furnished is likely to be incomplete and imprecise. Dialogues are also intrusive; the operators already have a principal task to accomplish, and interruptions designed to get information about the reason for their actions are distracting or annoying. To keep interruptions to a minimum, the dialogues should therefore ideally be conducted on a qualitative basis in terms that are intuitive for the operator. We tackle the problem of formalising just such an informal description of a skill in chapter 9, where we examine rewards given in an intuitive, qualitative fashion, easily expressed and quickly specified by the users.

Bratko and Šuc (2003) then suggest identifying the skill from the traces of the user's actions. We are not seeking to create a clone of the analyst, but we are seeking to capitalise on their expertise and judgement of what the results are worth (*i.e.* their knowledge of the true reward function of the MDP) to build a coherent model. Moreover, we're seeking to build a system that self-improves with each interaction, and which encapsulates the skills of each analyst with whom it interacts. This inspired our gathering of quantitative rewards through observation of the edits that the analysts carry out on an extracted event (see chapter 8).

Bratko and Šuc (2003) remark that once a clone of the operator is trained, it can perform better and more consistently than the operator. This reinforces our idea of the continuous improvement of the AI constructing the chain; eventually, it should construct the chain better and more consistently than even the experts. They point out that another benefit is not just the reproduction of an operator's skill, but understanding it, and being able to transfer the acquired knowledge to less experienced operators. Indeed, a trained system benefits everyone.

Before we could tackle the modelling of the rewards, we had to ask

ourselves several questions. The first, and arguably the most important, is at what level we wished to invite the analysts to reward the agent. Should it be on the document as a whole, or on individual attributes within that document? Certain questions will apply to all documents (for example, the format of the document is never detected), others only to individual documents (for example, a sentence is not translated correctly). We decided that it should be a mixture. A numeric reward is given on the attributes, proportional to the number of edits that the user carries out (see chapter 8), and a higher level qualitative reward (see chapter 10) is given on the extractions within the document (and therefore the treatment) as a whole - was there an extraction, was it fast, *etc.* The advantage of this approach is that there is no need to separate the training phase and long-term usage. The system continues to learn with every document it treats.

Loftin et al. (2015) show us that agents can learn not only from explicit feedback, but also from withheld feedback. For instance, a user may only reward good behaviour, in which case, the agent can assume that a lack of reward is a punishment, and vice versa. We could imagine a parallel with our work. In our case, the agent is mostly "punished" by the user corrections, but the agent could also take into account a "non-feedback", such as opening a document without correcting it as a positive reward. The difficulty is that in our case, the analyst may have closed the document without correcting it, simply because he was too busy, or just not interested in the contents. We therefore considered asking the analyst to provide an explicit feedback every time they consulted a document. For instance, by choosing from the options "I'm correcting this because it's bad", "I'm adding information because it's incomplete", or "I'm leaving it alone, because it's perfect / because I don't know / because I don't have time to correct it". We also thought about allowing the AI to ask why a document was, or was not corrected. Given that our aim was to be as unintrusive as possible, we rejected these ideas, opting for a behind-the-scenes approach to gathering the rewards.

As we discussed in section 4.2, the event summary would allow the analysts to consult the results produced by the system for a given event by showing the information extracted for each of its dimensions, linked back to the original source documents. The analyst would be able to see at a glance which events or parts of events had already been validated (by them, or another analyst), and they would be able to edit, add or remove information. They would also be able to split or merge events. The corrections made by the analyst to the extracted information and captured by the action tracker would indirectly furnish a reward on the treatment the document received. The rewards $r(s)$ are thus given to the system only for the final states of a

treatment, and are defined in three ways: *a quantitative value* based on the corrections made to the extraction (if any) by the analyst and the document treatment time (chapter 8); *a qualitative reward* based on the user's purely ordinal preference on the results; and *a weighted ordinal preference* giving a middle-ground between the first two (chapter 10).

Chapter 7

BIMBO: a flexible platform

7.1 Introducing BIMBO

As we stated in section 1.2, our contribution consists not only of modelling a treatment chain as an MDP, but also of building a flexible platform *BIMBO: Benefiting from Intelligent and Measurable Behaviour Optimisation*. We built BIMBO in a completely modular fashion, allowing her elements to be changed easily (Figure 7.1 shows BIMBO's modules, which we will describe below.)

This means that adding a new reinforcement learning algorithm, or changing the definition of a state, for example, changes nothing in the rest of the platform. This modularity enables us to test easily a range of RL algorithms, input documents, similarity measures, rewards, user feedback, models, available services, *etc.* We later proved this flexibility by using BIMBO in a completely different context (see chapter 13), but initially we ran our experiments with our industrial application.

In Appendix A, Listing A.1, we give an example of the configuration file which enables every aspect of *BIMBO* and her AIs to be controlled, from the parameters used in the algorithm to whether progress reports should be sent by SMS.

In our tests, we used a simple, but typical chain (as described in chapter 4) as a reference, with two differences:

- Firstly, to check that the AI is capable of rejecting useless services, we added the possibility of choosing a service *Geo*. This service is a geographical inference service, which relies on the extracted spatial dimension of the event to add extra details from the geographical database Geonames (2015). For example, if "Paris, France" is extracted, we can infer "Europe". Obviously, in production, this service could be called to complete the event summary, but it contributes nothing to the extraction process itself, and so the AI should consider it useless.

- Secondly, the open-source web service which encapsulates GATE (available from WebLab (2016b)) is only capable of detecting named entities.

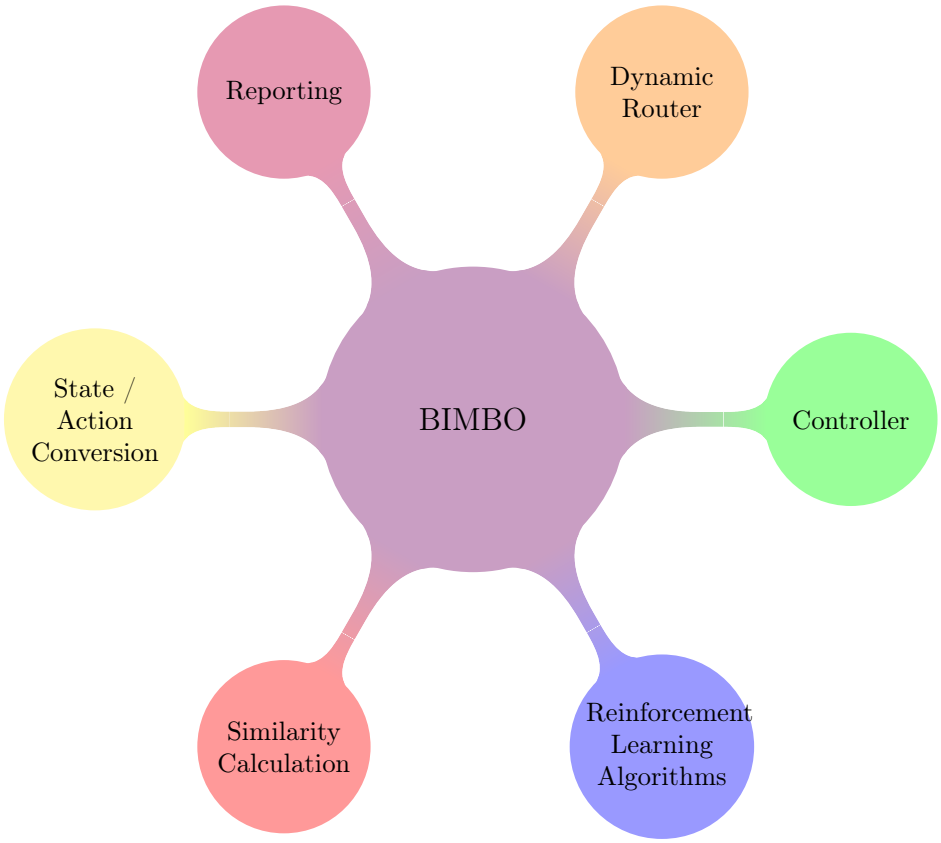


Figure 7.1 – The modules of BIMBO. The Dynamic Router defines the available services, the controller holds the stop-watch and controls the flow of input documents, *etc.*

> We therefore used a more elaborate service which encapsulates the ontology WOOKIE (developed by Serrano (2014), see Appendix B) and GATE, which is also capable of extracting OSINT events. This particular service uses gazetteers, or lists of trigger words for event detection in a text. These lists also contain the grammar rules (JAPE, 2016) applicable to each word. Unless otherwise specified, all future references to "GATE" in this document can be read as "the service encapsulating WOOKIE and GATE".

Recall that in section 6.2, we decided that the actions available to *BIMBO* were to choose the next service from {*Tika, NGramJ, GATE*} (we now also add *Geo*), to choose a GATE gazetteer, or to *STOP* the treatment and

return any extracted events. In section 6.1 we defined the states by the characteristics:

- the information already extracted (if any), where " " indicates not (yet) extracted:
  - *language* ∈ {"en"," "};
  - *format* ∈ {"*text/plain*"," "};
  - *interesting* ∈ {*true, false*} (*true* if at least one event of the type that interests the analyst has been extracted);
  - *any* ∈ {*true, false*} (*true* if one or more events of any type have been extracted);
- the parameters chosen:
  - *nounGazetteer* (the noun gazetteer currently chosen, if any);
  - *verbGazetteer* (the verb gazetteer currently chosen, if any);
- the system state:
  - *nbServices* ∈ $\{0-5, 6-20, 21+\}$ (the number of actions the AI has already taken, *i.e.* the number of services through which the current document has already passed plus the parameter changes);
  - *timeTakenSoFar* (number of seconds since the current document treatment started in *timeInterval* (parametrisable) steps).

The chain is written as a Camel (2015) route in XML (see Appendix C). We defined each of the four services as an endpoint (Figure C.1). The route "consumeFile" (Figure C.2) is triggered as soon as a file is copied into the "ai-chain-to-process" folder. After adding the header information, such as the date and file name, it sends the WebLab resource to the *Dynamic Router* in "direct:start" (Figure C.3). The *Dynamic Router* then passes control to *BIMBO*. Specifically, it calls the Java method "performActionMaster". This method is responsible for informing the AI of its current state, asking it to choose an action, and then passing the control back to the appropriate part of the camel chain. For instance, if the AI chooses "Tika" (or rather, the integer corresponding to the action "call the web service Tika"), then we call the route "direct:tika", which passes the current WebLab resource to the endpoint "weblab:analyser:service-tika". If the AI chooses to change one of the *gazetteers*, then we copy the chosen gazetteer into the GATE folder, and call the route "direct:dummy" which does nothing (except display debugging

messages if required) but pass the control back to "performActionMaster" again.

We thus give *BIMBO* and her AIs complete control over all the services called, their order and their parameters.

In section 6.3, we explained that in RL, the rewards that the agent is given are based on how close its results are to the goals it was set. In our platform, we cannot define these goals exactly, as they will vary from document to document. Instead, we have to reward the agent based on the analyst's feedback on their impression of the results of the treatment that a document received.

7.2 An example decision process for one document

Before we describe our platform in detail, and to better visualise the model we chose in chapter 6, here's an example of an actual document treatment taken from just under halfway through the learning process (during the tests with numeric rewards). Figure 7.2 gives a graphical representation of the states and actions, and Figure 7.3 shows the log produced by BIMBO.

The AI starts in state s_0, from which it has already learnt that the best action is to choose a GATE gazetteer, specifically the mixed list of verbs.

BIMBO informs it that it is now in state s_{84}, and has received zero reward. The only difference between state s_0 and state s_{84} is that the "Verbs chosen" property is no longer empty, but has the value "MixedVerbs".

Again, the AI exploits its current knowledge by taking the action that it has learnt is the best from state s_{84}, that is, to send the document to Tika to detect its format and normalise it into an XML format.

The web service successfully detects that the format is "text/plain". Therefore, the state s_{268} in which it now finds itself is the same as the previous one, but with the property "Format" equal to "text/plain".

From state s_{268}, we see that there is still some learning to do, as the AI thinks that the best action is to choose the dummy list of nouns.

This takes it to state s_{269} where, naturally, the value of the property "Nouns chosen" becomes "DummyNouns".

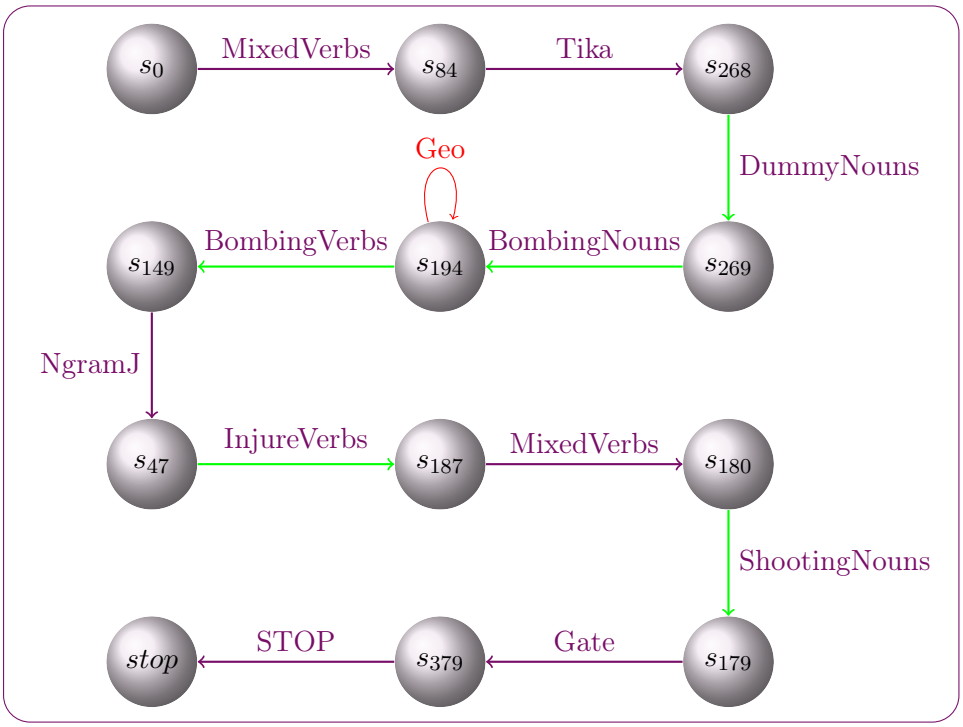


Figure 7.2 – The graphical representation of the choices made during the treatment of a single document roughly halfway through the learning process.

Similarly, from state s_{269}, it chooses to change the nouns gazetteer again, this time for the list of bombing nouns, taking it to state s_{194}.

In state s_{194}, the AI chooses a sub-optimal action on purpose to expand its knowledge (it explores). It chooses to send the document through the service "Geo". This only wastes time as it does not extract any new information, and the AI finds itself back in the same state.

The next choice it makes is unfortunately to change the verbs gazetteer again, arriving in state s_{149}. I say unfortunately, as in this particular experiment the mixed list was supposed to be the best choice. The property "Verbs chosen" becomes "BombingVerbs", and the property indicating the number of actions chosen "No. actions" is increased to the next range, i.e. to "6-20".

The AI then makes another wise choice, sending the document through the

```
16,049 **** BATCH 21 of 50, document 69 of 100 ****
16,551 Start in state = 0 [Language: - ; Format: - ; Time: 0; No actions: 0-5; Interesting event extracted: false; Any event extracted: false; Nouns chosen: - ; Verbs chosen: - ]
16,551 Exploit: change gazetteer to MixedVerbs
16,565 Informing (s, s', a, r): (0, 84, 14, 0.0)
16,565 Now in state: 84 [Language: - ; Format: - ; Time: 0; No actions: 0-5; Interesting event extracted: false; Any event extracted: false; Nouns chosen: - ; Verbs chosen: MixedVerbs]
16,565 Exploit: Chose service Tika
16,566 Starting Tika
16,573 Tika finished
16,586 Informing (s, s', a, r): (84, 268, 0, 0.0)
16,586 Now in state: 268 [Language: - ; Format: text/plain; Time: 0; No actions: 0-5; Interesting event extracted: false; Any event extracted: false; Nouns chosen: - ; Verbs chosen: MixedVerbs]
16,586 Exploit: change gazetteer to DummyNouns
16,602 Informing (s, s', a, r): (268, 269, 7, 0.0)
16,602 Now in state: 269 [Language: - ; Format: text/plain; Time: 0; No actions: 0-5; Interesting event extracted: false; Any event extracted: false; Nouns chosen: DummyNouns; Verbs chosen: MixedVerbs
16,602 Exploit: change gazetteer to BombingNouns ]
16,617 Informing (s, s', a, r): (269, 194, 5, 0.0)
16,617 Now in state: 194 [Language: - ; Format: text/plain; Time: 0; No actions: 0-5; Interesting event extracted: false; Any event extracted: false; Nouns chosen: BombingNouns; Verbs chosen: MixedVerbs]
16,617 EXPLORE ! Chose service Geo  QValue for Geo is 4600, maximum qValue is 8135 for BombingVerbs
16,618 Starting Geo
16,625 Geo finished
16,640 Informing (s, s', a, r): (194, 194, 3, 0.0)
16,640 Now in state: 194 [Language: - ; Format: text/plain; Time: 0; No actions: 0-5; Interesting event extracted: false; Any event extracted: false; Nouns chosen: BombingNouns; Verbs chosen: MixedVerbs]
16,640 Exploit: change gazetteer to BombingVerbs
16,671 Informing (s, s', a, r): (194, 149, 10, 0.0)
16,671 Now in state: 149 [Language: - ; Format: text/plain; Time: 0; No actions: 6-20; Interesting event extracted: false; Any event extracted: false; Nouns chosen: BombingNouns; Verbs chosen: BombingVerbs]
16,671 Exploit: Chose service NgramJ
16,671 Starting NgramJ
16,675 NgramJ finished
16,693 Informing (s, s', a, r): (149, 47, 1, 0.0)
16,693 Now in state: 47 [Language: en; Format: text/plain; Time: 0; No actions: 6-20; Interesting event extracted: false; Any event extracted: false; Nouns chosen: BombingNouns; Verbs chosen: BombingVerbs]
16,693 Exploit: change gazetteer to InjureVerbs
16,710 Informing (s, s', a, r): (47, 187, 11, 0.0)
16,710 Now in state: 187 [Language: en; Format: text/plain; Time: 0; No actions: 6-20; Interesting event extracted: false; Any event extracted: false; Nouns chosen: BombingNouns; Verbs chosen: InjureVerbs ]
16,710 Exploit: change gazetteer to MixedVerbs
16,728 Informing (s, s', a, r): (187, 180, 14, 0.0)
16,728 Now in state: 180 [Language: en; Format: text/plain; Time: 0; No actions: 6-20; Interesting event extracted: false; Any event extracted: false; Nouns chosen: BombingNouns; Verbs chosen: MixedVerbs]
16,728 Exploit: change gazetteer to ShootingNouns
16,745 Informing (s, s', a, r): (180, 179, 8, 0.0)
16,746 Now in state: 179 [Language: en; Format: text/plain; Time: 0; No actions: 6-20; Interesting event extracted: false; Any event extracted: false; Nouns chosen: ShootingNouns; Verbs chosen: MixedVerbs]
16,746 Exploit: Chose service Gate
16,746 Starting Gate
24,987 Gate finished
24,999 Injure event extracted
25,007 Informing (s, s', a, r): (179, 379, 2, 0.0)
25,007 Now in state: 379 [Language: en; Format: text/plain; Time: 1; No actions: 6-20; Interesting event extracted: false; Any event extracted: true; Nouns chosen: ShootingNouns; Verbs chosen: MixedVerbs]
25,007 Exploit: Chose to STOP
25,089 Time taken for document 69: 9040ms
25,089 Similarity 0.6 between extracted event and GTD; chain took 8 secs, so reward is 75.0
25,095 Informing (s, s', a, r): (379, stop, 4, 75.0)

```

Figure 7.3 – The log from the treatment of a single document roughly halfway through the learning process. The numbers on the left of each line in the log show the time in nanoseconds (we have taken out the days / hours / minutes to aid readability).

language detector. The service successfully detects that the document is written in English, and at last the property "Language" is filled in. This is good news, as now all the information about the document itself that GATE needs to start extracting events is there. We just need the AI to choose the right gazetteers now, and to send the document through GATE.

After another bad choice of gazetteer (the injure verbs), the AI finally chooses the mixed verbs gazetteer again, and arrives in state s_{180}.

It then chooses the shooting nouns gazetteer. Spoiler alert: this turned out not to be important, as we discovered later (thanks to the AI making choices like this) that this GATE service is faulty and erroneously never uses the nouns! This detection of errors in the services was an unexpected benefit of the platform, and could be interesting to explore in the future. For example, if a service exhibits a surprising behaviour, an alert could be raised with the development team, and another service substituted automatically.

At last, the AI sends the document to GATE. This is the most costly service in terms of time, and the AI has to learn rapidly to use it only once at the end of the chain, otherwise it is heavily penalised.

The AI is now in state s_{379}, and it "knows" that it has managed to extract an event ("Any event extracted" is true), even if it was not the one that the analyst had specified ("Interesting event extracted" is false). It therefore decides to STOP the treatment of this document.

At the end of the log, we can see that an event was extracted in 8 seconds, with a similarity of 0.6 to the oracle (more details on how we measured this are in section 7.6). At last BIMBO can reward the AI, so for this particular result, she gives it 75.0 (the similarity multiplied by 1000 for readability, and divided by the time taken). The AI will use this reward to update its estimates of the best actions to follow in each state.

7.3 Definition and example of an event

We've spoken about the *events* that our chain is supposed to extract, but we've not yet formally defined what they look like. As the context of our CIFRE contract means that we use the WebLab services, it was natural to

use the associated definition of an event given by Serrano (2014). In practise, the system is modular, and with the appropriate extraction service, any other definition such as those discussed in section 2.2 could also be used.

Definition 4 (Event) *An* event E *is a quadruplet* (C, T, G, A) *where:*

- $C \subseteq \mathbb{C}$ *is the conceptual, or semantic dimension, given by a set of elements taken from the domain* $\mathbb{C}$ *common to all events;*
- T *is the temporal dimension, or when the event occurred. It is potentially ambiguous,* e.g. *"last Tuesday", and to model this ambiguity, we take* $T \subseteq \mathbb{T}$*, where* $\mathbb{T}$ *is the set of all dates;*
- $G \subseteq \mathbb{G}$ *is the spatial dimension, or where the event occurred, also potentially ambiguous;*
- $A \subseteq \mathbb{A}$ *is the agentive dimension, or the participants.*

Within this general definition, we use precise domains:

- $\mathbb{C}$ *is the set of all types of event, given by a fixed and finite set of elements in the ontology* WOOKIE *(see Appendix B) used by the extractor;*
- $\mathbb{T}$ *is the set of all relative and absolute "dates", such as "tomorrow", "2001", "2001/9/11";*
- $\mathbb{G}$ *is the set of entities defined in Geonames (2015);*
- $\mathbb{A}$ *is the infinite set of all extractable participants, seen as strings.*

Example 5 (Extraction of an event) *Recall the document that we saw in Example 1 on page 26:*

> *4/29/1971: In a series of two incidents that might have been part of a multiple attack, suspected members of the Chicano Liberation Front bombed a Bank of America branch in Los Angeles, California, US. There were no casualties but the building sustained $1 600 in damages.*

Using Definition 4, an event $E = (C, T, G, A)$ *could be extracted from this document with* $C = \{AttackEvent, BombingEvent\}$, $T = \{4/29/1971\}$, $G = \{Los\ Angeles, California, US\}$, *and* $A = \{Chicano\ Liberation\ Front, Bank\ of\ America\}$.

7.4 Construction of the corpus

To test our approach in the industrial context of the CIFRE contract, we used an open-source database in which the events are already known: the *Global Terrorism Database* (*GTD*) (Global Terrorism Database, 2016b; La-Free, 2010), consisting of details of over 125 000 worldwide terrorist events from 1970 to 2014.

Each of these events lists up to three sources. We initially extracted the urls and retrieved the web pages (see Figure 7.4 for an example).

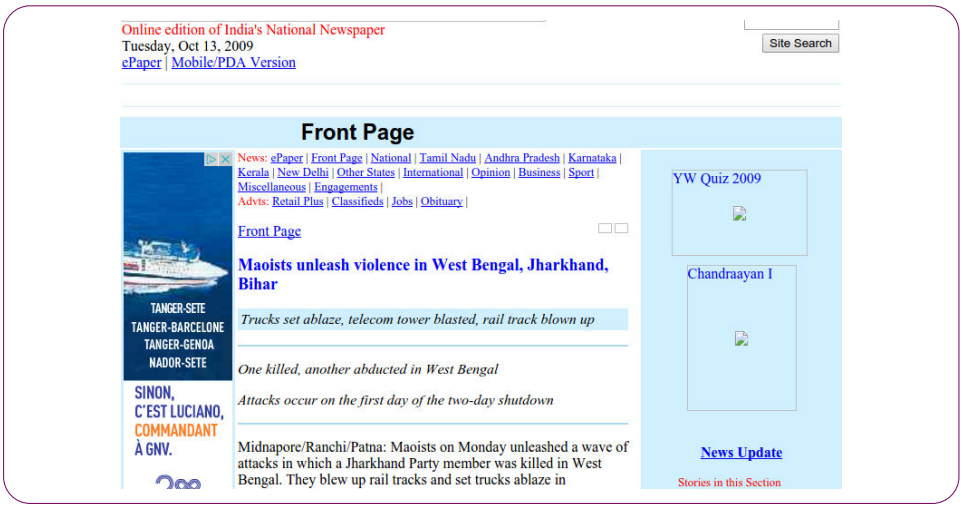


Figure 7.4 – Original document (web page) - shortened (see Appendix D for the full page. This web page is reproduced here with the kind permission of Kasturi & Sons Ltd (KSL).

From these web pages, the treatment chain was able to extract event information, such as that shown in Figure 7.5. For the brave of heart, an example of the WebLab resource corresponding to this web page at the end of the treatment is shown in Appendix E.

Unfortunately, once the duplicates and pages which were no longer valid were removed, only 51 documents remained, which was too small to construct a decent corpus. We therefore turned to the summaries of the events (for example, Figure 7.6) to create our corpus.

These summaries are more informative, and less noisy than the original web pages. Another advantage of using the summaries is that we have one document - one event, which simplifies the evaluation of the results. The

One killed, another abducted in West Bengal
Attacks occur on the first day of the two-day shutdown
Midnapore/Ranchi/Patna: Maoists on Monday unleashed a wave of attacks in which a Jharkhand Party member was killed in West Bengal.

$E = (C, T, G, A)$ with $C = \{DeathEvent\}$, $T = \{\text{Monday}\}$, $G = \{\text{West Bengal}\}$, and $A = \{\text{Jharkhand Party member}\}$.

Figure 7.5 – Part of the text from the web page in Figure 7.4 and the corresponding event dimensions extracted using WebLab.

10/12/2009: On Monday, the body of Jharkhand anti-Maoist "Maobadi Protirodh Committee" member, Kanai Murmu (48), was found. Murmu, a resident of Jorkisoli village near Lalgarh, was dragged out from his house by a group of armed men in the early hours of 10/12/2009. Later, his bullet-riddled body was found by local residents beside a pond in Binapur's Salpatra. Murmu was a former gram panchayat (village level local self-Government institution) member of Jharkhand Party-Naren, who had joined the Gana Pratirodh Committee (an anti-Maoist Committee) and mobilized local youths against Maoists. No group claimed responsibility.

$E = (C, T, G, A)$ with $C = \{DeathEvent, KidnappingEvent\}$, $T = \{10/12/2009, \text{Monday}\}$, $G = \{\text{Jorkisoli, Lalgarh, Binapur, Salpatra, West Bengal}\}$, and $A = \{\text{Jharkhand anti-Maoist "Maobadi Protirodh Committee" member, Kanai Murmu}\}$.

Figure 7.6 – The event summary in the GTD corresponding to Figure 7.4 and the corresponding formalised event.

disadvantage is that they are homogeneous, being all the same format and approximate length, and written in the same language (English). This homogeneity does, however, allow us to generalise on a random sample of documents. Ideally, with access to infinite resources, we would have constructed a corpus, annotated by hand from a variety of different documents.

This set of summaries gave us documents $\{d_1 \dots d_N\}$ from which a perfect chain could extract perfect events $E_1, \dots, E_N$, respectively (as in Figure 7.6), if it had infinite time, and access to an infinite number of perfectly parametrised web services. Obviously such a chain does not exist in real life, so we use the results from a production chain constructed by hand as a reference, referred to hereafter as the "expert" chain.

We expect our AI to learn to construct chains that approach and eventually improve on the expert chain, in learning not only the correct order of the

services and the best parameters, but also the fact that certain services (in our case, the service *geo*) and certain parameters (some GATE gazetteers) are not useful.

7.5 Conversion of the corpus to WebLab type events

In order to obtain meaningful results from the AI, large numbers of documents need to be treated by the system, and the accuracy of the resulting extracted events evaluated. As stated in section 4.3, due to operational constraints, we could not ask the analysts to participate in our tests. We therefore developed an automatic comparison of events extracted by the chain with their GTD counterparts to simulate the analysts' feedback. This comparison is also a key component of the success measure for the AI for that instance of the treatment chain (see section 7.6).

There are two sets of event information: the GTD files, and that extracted by the treatment chain. These sets need to be normalised as they are not in the same format, and then compared to give a "user feedback" on each event.

Each event in the GTD files is identified by a unique reference number. There are 133 fields / columns for each event. Some of these contain information that cannot be extracted from the summary (for example the GPS coordinates) or which are not useful in our experiments (for example links to the documents or web pages which serve as source of the information). For full details, please refer to Global Terrorism Database (2016a).

Table 7.1 gives an abridged example of an event from the GTD files (note that we have based this example on the fusion of several similar GTD events in order not to leave blanks). In Table 7.2, we show the abridged event from the GTD file, corresponding to the summary shown in Figure 7.6 on page 60.

The dimensions of the GTD event are filled using both the information given directly in the GTD files, and inferred information as shown in Table 7.3:

The agentive and spatial information contained in the GTD files is entered by hand. Often, therefore, generic words and phrases are introduced which reduce the measured similarity unnecessarily. For instance, take the following example of a summary from which we would hope to extract the spatial named entities *'Ayn al-Hulway Palestinian Refugee camp* and *Sidon*:

GTD field	Example value
event id	199311250001
year	1993
month	11
day	25
approximate date	November 25 - 26, 1993
extended resolution date	26/11/1993
country	Egypt
region	Middle East & North Africa
province / state	Al Qahirah (Governorate)
city	Heliopolis
location	This incident occurred in Heliopolis.
summary	11/25/1993: There was a failed assassination attempt on the Egyptian Prime Minister. The attack injured 20 and killed 1. The attack was claimed by both Vanguards of Victory and Islamic Jihad.
attack type	Bombing/Explosion
entity targeted	Egyptian Prime Minister
specific target	Egypt's Prime Minister Atef Sedki
perpetrator group	Al Jihad
perpetrator subgroup	Talaa'al al-Fateh
number killed	1
number wounded	20

Table 7.1 – An abridged example of the fields from several GTD events.

GTD field	Example value
event id	200910120024
year	2009
month	8
day	12
approximate date	
extended resolution date	
country	India
region	South Asia
province / state	West Bengal
city	Midnapore
location	The attack took place in Jorkisoli village near Lalgarh, India.
summary	10/12/2009: On Monday, the body of Jharkhand anti-Maoist "Maobadi Protirodh Committee" member, Kanai Murmu (48), was found. Murmu, a resident of Jorkisoli village near Lalgarh, was dragged out from his house by a group of armed men in the early hours of 10/12/2009. Later, his bullet-riddled body was found by local residents beside a pond in Binapur's Salpatra. Murmu was a former gram panchayat (village level local self-Government institution) member of Jharkhand Party-Naren, who had joined the Gana Pratirodh Committee (an anti-Maoist Committee) and mobilized local youths against Maoists. No group claimed responsibility.
attack type	Hostage Taking (Kidnapping)
entity targeted	A Jharkhand anti-Maoist "Maobadi Protirodh Committee" member was targeted.
specific target	Unknown
perpetrator group	Unknown
perpetrator subgroup	Unknown
number killed	1
number wounded	0

Table 7.2 – The GTD event corresponding to the summary shown in Figure 7.6.

Event dimension	Normalisation
Semantic	The attack types (converted as in Table 7.4) "DeathEvent" if number killed > 0 "InjureEvent" if number wounded > 0
Temporal	Year, month, day combined to form dd/MM/yyyy Extended incident resolution date in dd/MM/yyyy format
Temporal (*derived*)	Approximate date Year, month, day combined to form dd-MM, yyyy-MM, MM-dd Derived day(s) of the week
Spatial	Country Region Province / administrative region / state City Location
Agentive	Perpetrator group names Perpetrator subgroup names Names of entities targeted Specific targets / victims

Table 7.3 – Derivation of the GTD event dimensions from the information given in the GTD file.

GTD event type	WebLab event type
Assassination	DeathEvent
Hijacking	KidnappingEvent
Kidnapping	KidnappingEvent
Hostage Taking (Kidnapping)	KidnappingEvent
Barricade Incident	AttackEvent
Bombing/Explosion	BombingEvent
Unknown	Unknown
Armed Assault	AttackEvent
Unarmed Assault	AttackEvent
Facility/Infrastructure Attack	AttackEvent

Table 7.4 – Mapping from the GTD event types to the WebLab event types (as defined in the WOOKIE ontology).

06/17/1993: A bomb killed Fatah Officer 'Ali Iskandar and his wife and wounded three other family members when it exploded at their home in the 'Ayn al-Hulway Palestinian Refugee camp near Sidon. The perpetrators were unknown.

The associated location is given as

> The attack occurred in the 'Ayn al-Hulway Palestinian Refugee camp near Sidon

We therefore reduced this noise by removing the following generic words and phrases from the agentive and spatial dimensions: *the attack, the incident, occurred, took place, in front of, near, in, at, of, on, and, the, a, an, unspecified location, undisclosed, was / were targeted.*

As in standard in string manipulation, we also removed from all the dimensions, punctuation such as quotes, fullstops, commas and hyphens. We trimmed the strings, and replaced multiple whitespaces with a single space character.

7.6 Measuring the quality of the results

In order to test whether our AI is making a difference, and whether it is "intelligent enough", we need to measure its *IQ*, or how intelligent it has become. In other words, to identify at what point it becomes useful or mature. We therefore need to measure the quality of its results.

To do this, we must first define the similarity between an event extracted from the document by the chain $E_1 = (C_1, T_1, G_1, A_1)$, and the corresponding "perfect" event $E_2 = (C_2, T_2, G_2, A_2)$.

In our experiments, E_2 is the event as detailed in the *GTD* files and here we have defined a fairly simple similarity measure which is specific to the GTD – WebLab mapping. However, in production, the comparison would be taken between the event extracted and the event as corrected by the analyst. In this case, the similarity measure would certainly need to be rethought[1]. For instance, when comparing with the GTD, the geographical information is either correct or not, but in production, it may be partially correct, and, of course, the documents could contain several events instead of at most one per document.

[1]Of course, BIMBO being completely modular, slotting in a new similarity measure would have no impact on the rest of the framework.

This can be complicated by the fact that just as the analyst may add information that was not in the original document, the GTD contains event information that is not included in the summary, and thus could not possibly have been extracted. We try to avoid penalising the AI for this.

Numerous methods have been proposed for measuring similarity (Pandit et al. (2011); Tversky and Gati (1978); Cohen et al. (2013); Dutkiewicz et al. (2013) to give but a few), but we wanted a measure taking into account the events' four dimensions, and capable of differentiating between different types of event, therefore we put weights a, b, c and d on the dimensions and define:

Overall similarity between two events

$$\sigma(E_1, E_2) = \frac{a\sigma_C(C_1, C_2) + b\sigma_G(G_1, G_2) + c\sigma_T(T_1, T_2) + d\sigma_A(A_1, A_2)}{(a + b + c + d)} \tag{7.1}$$

Semantic similarity Because we mapped the values of the WebLab semantic dimension onto those of the GTD (see Table 7.4), we can define the semantic similarity $\sigma_C(C_1, C_2)$ to be 1 if there is a common element between the semantic dimensions of E_1 and E_2, and 0 otherwise. For example, for $C_1 = \{Bombing\}$ and $C_2 = \{Attack,\ Bombing\}$, we obtain $C_1 \cap C_2 = \{Bombing\}$, so $\sigma_C(C_1, C_2)$ is 1.

Geographical (spatial) similarity We define the geographical similarity $\sigma_G(G_1, G_2)$ in the same way, that is $\sigma_G(G_1, G_2)$ is 1 if there is a common element between the geographical dimensions of E_1 and E_2, and 0 otherwise. For example, for $G_1 = \{Caen\}$ and $G_2 = \{Rouen\}$ $G_1 \cap G_2 = \emptyset$, and so $\sigma_G(G_1, G_2)$ is 0

Temporal similarity The temporal similarity $\sigma_T(T_1, T_2)$ is 1 for $T_1 \cap T_2 \neq \emptyset$. Otherwise, if there is no common element, we use partial and derived information. For example, if $T_1 = \{7\ October\ 1969\}$, $T_2 = \{7/10/69\}$, $T_3 = \{October\}$ and $T_4 = \{Tuesday\}$. We can see that T_1 is the same as T_2, just in a different format, so $\sigma_T(T_1, T_2)$ is 1.

Deriving possible partial dates from $T_1 = \{7\ October\ 1969\}$ gives

$$\{October\ 1969,\ 07\text{-}10,\ 10\text{-}07,\ 1969\text{-}10,\ 1969,\ October\}$$

October is a common element with T_3, so we give $\sigma_T(T_1, T_3)$ a value of $\frac{1}{10}$. Finally, we can calculate that 7^{th} October 1969 was a Tuesday, so

$\sigma_T(T_1, T_4) = \frac{1}{7}$. Note that the values $\frac{1}{10}$ and $\frac{1}{7}$ were chosen by experiment, and highlight the difficulty of giving an exact numerical value to a comparison.

Agentive similarity For the agentive similarity, we initially tried the Jaccard similarity given by the number of elements common to each set divided by the total number of distinct elements:

$$Jacc(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

For example, if the first set A_1 is $\{$*Al-Qaeda, Army of Islam*$\}$ and the second A_2 is $\{$*Al-Qaeda, The Foundation*$\}$, then

$$Jacc(A_1, A_2) = \frac{|\textit{Al-Qaeda}|}{|\textit{Al-Qaeda, Army of Islam, The Foundation}|} = \frac{1}{3}$$

The problem is that this does not take into account spelling differences which are very common, for example in the transcription of Arabic names, and both the analysts' corrections and the GTD standardise the spellings. If A_2 contained $\{$*Al-Qaida*$\}$ instead of $\{$*Al-Qaeda*$\}$, the Jaccard similarity would give

$$Jacc(A_1, A_2) = \frac{|\emptyset|}{|\textit{Al-Qaeda, Army of Islam, Al-Qaida, The Foundation}|} = \frac{0}{4}$$

We therefore decided to use the Levenshtein distance (the minimum number of characters to delete, insert or replace to convert one string into the other) (Levenshtein, 1966) on each pair of agents a_1, a_2 in A_1, A_2.

$$\textit{Levenshtein}(\textit{Al-Qaeda, Al-Qaida}) = 1 \text{ replacement}$$

But, names are often more complex, for example, if we compare *Jharkhand anti-Maoist Maobadi Protirodh Committee member* with *Jharkhand Party member*, the standard Levenstein distance is 36, and yet, logically, they are the same entity. Also, the named entity (NE) extractors such as Gate can make partial matches, or over-match. We therefore make this Levenshtein distance "fuzzy" ($FL(a_1, a_2)$) by comparing the substrings (Ginstrom, 2007). This means that we can ignore the prefixes and suffices which are often the parts which are mismatched, whilst still accommodating slight differences in spelling.

We define the agentive similarity $\sigma_A(A_1, A_2)$ as

$$\sigma_A(A_1, A_2) = 1 - \frac{\min(FL(a_1, a_2))}{\min(|a_1|, |a_2|)} \mid \forall a_1 \in A_1, \forall a_2 \in A_2$$

if over a certain threshold θ, and 0 otherwise (in practise, $\theta = 0.45$ gave the best results).

For example, if $A_1 = \{Dr\ Dolittle\ PhD\}$ and $A_2 = \{Doolittle\}$, the standard Levenshtein distance is 7, but the fuzzy Levenshtein distance is only 1, and so the similarity $\sigma_A(A_1, A_2)$ is

$$\begin{aligned}\sigma_A(A_1, A_2) &= 1 - \frac{FL(\cancel{Dr}\ Dolittle\ \cancel{PhD}, Do\cancel{o}little)}{|Doolittle|} \\ &= 1 - \frac{1}{9} \\ &= 0.889\end{aligned}$$

Taking the problematic example above:
if $A_1 = \{Jharkhand\ anti\text{-}Maoist\ Maobadi\ Protirodh\ Committee\ member\}$ and $A_2 = \{Jharkhand\ Party\ member\}$, then the fuzzy Levenshtein calculation gives a distance of 11, and thus a similarity of $1 - \frac{11}{22} = 0.5$.

Given two lists of agents, we do a pairwise matching, and use the maximum similarity.

We do not try here to associate named entities such as *London / capital of England*, but the modularity of the system would allow this.

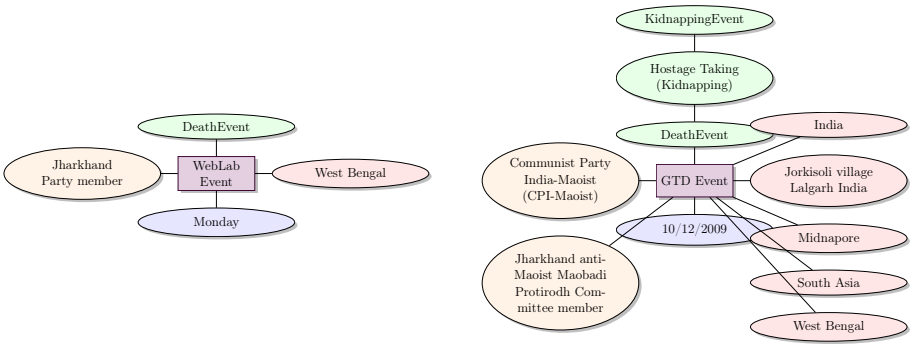
Example 6 (Full comparison) *If we take the two events described in Figure 7.7, we can see that:*

- *There is an overlap (*Death Event*) in the semantic description of the event, and so the semantic similarity is 1.*
- *The spatial similarity is 1, due to the common element* West Bengal*.*
- *We have to rely on the deduction of the day of the week, getting a temporal similarity of 0.143.*
- *Finally, the agentive similarity, as we have already seen, is 0.5.*

If we put equal weights on the dimensions ($a = b = c = d = 1$), we have an overall similarity of

$$\sigma(\textit{WebLab Event}, \textit{GTD Event}) = \frac{1.0 + 1.0 + 0.143 + 0.5}{4} = 0.66075$$

between the extracted event, and the corresponding event in the GTD.



(a) Extracted event (b) "Perfect" event from GTD

Figure 7.7 – Similarity of 0.66075 between the extracted event, and the corresponding event in the GTD: Semantic: 1; Temporal: 0.143; Spatial: 1.0; Agentive: 0.5; ($a = b = c = d = 1$).

Definition 7 (Quality) *The quality of the events extracted (if they exist) weighs this similarity against the time taken to treat the document. More precisely, if the extraction of $\hat{E}_i$ from document d_i took time t with target event E_i, we define the quality Q of the extraction to be*

$$Q = \begin{cases} \sigma(\hat{E}_i, E_i)/t & \text{if } \sigma(\hat{E}_i, E_i) > 0 \\ -t & \text{otherwise)} \end{cases}$$

Obviously, if there is no event extracted (or the extracted event has null similarity with the target event), then $Q = -t$.

We thus formalise the fact that the correct extraction of events is primordial, and must be done in a reasonable time, and that the AI should rapidly detect if there is no interesting event to extract.

As we said in section 7.4, the corpus was constructed from the GTD summaries such that the chain should only extract one event from each document. In our earliest tests, the AI learnt to "cheat", by calling the extraction service many times over to extract the same event. Therefore, now, if the AI extracts several events for one target event, we measure the quality of each, and then take their mean.

Cheng and Hüllermeier (2008) note that manually specifying a similarity measure is extremely hard, and yet the performance of the system often depends on it. We are conscious that none of the above measures are perfect, even if they do allow the AI to learn effectively (see chapter 8). Our objective

was always therefore to move away from giving a numerical reward based on this similarity measure to giving a genuinely qualitative feedback (see chapter 9).

Chapter 8

Experiments with numerical rewards

8.1 Framework

In our first experiments with numerical rewards, we use standard *Q-learning* (Watkins, 1989) as presented in chapter 5. Nonetheless, our contribution includes modelling the improvement of a treatment chain as an RL problem, and in practise, any RL algorithm could be used.

In these experiments, the AI receives numerical rewards as defined in section 7.6 on the quality Q of its results. In production, this corresponds to rewarding it based on the similarity between the event which it has extracted (if any), and the extraction as corrected by the human analyst, considered to be the "perfect" event which could be extracted from the document (if any). This is non-intrusive, as it relies on corrections which the analyst makes as part of his daily routine, but requires fine-tuning of the definition of the similarity and its parameters, a cognitively difficult task, as we've already seen. Note that the numerical reward given to the AI here is the same as that used for assessing the quality of the policy it has learnt, as is standard in RL but not natural in real life. In chapter 10, we remove these requirements and report the results with qualitative rewards.

In our tests, unless otherwise specified, the AI starts learning "from scratch" with no *a priori* knowledge of the documents or the users needs. We say that such an AI is "untrained". Obviously, this means that we cannot expect good extractions from the start. Rather, we are interested in how fast a "good" policy is learnt (after seeing how many documents), and how good this policy is. Once the AI has treated a certain number of documents, it will have learnt what it considers to be an optimal policy, and we say that it is "trained".

Obviously, here we are testing the learning capacity of an initially untrained AI, but in production we would capitalise on existing expertise by initialising *BIMBO* with a policy based on that of an existing production chain. With the Q-Learning algorithm, for example, this can be done quite

simply by importing the Q-values already learnt. We would launch the expert chain on known documents, and force the algorithm to always choose the "expert" action. The Q-values thus created can be easily stored, and then re-read to initialise the AI.

We performed three types of experiment with numerical rewards, each time comparing the performance of the AI(s) against our reference, the expert chain (as a reminder, the expert chain is a production chain constructed by hand to be optimal for this type of document).

- Firstly, in section 8.2: Trained vs untrained vs expert – we train an AI, and then pit it against an untrained AI and the expert chain on an unknown set of documents to see how effective the training was. We also emulate an analyst's interest in a certain type of event.
- Secondly in section 8.3: Few extractable events – to test the robustness of the learning, we increase the action space, and give an untrained AI a set of documents that contain a very low percentage of extractable events.
- Finally, in section 8.4: Sporadic rewards – to emulate the non-availability of the analyst, we test three untrained AIs with increasingly sporadic rewards. We observe and explain a strange behaviour, and experiment with a different way of reducing the exploration rate ϵ.

8.2 Trained vs untrained vs expert

In this section, we first train an AI repeatedly on a small set of 100 documents. We then test the quality of the policy that it learnt during this training on a larger, randomly chosen set of 1000 documents. As a comparison, we also run an untrained AI and the expert chain on the same 1000 documents.

As a reminder, the similarity calculation is

$$\sigma(E_1, E_2) = \frac{a\sigma_C(C_1, C_2) + b\sigma_G(G_1, G_2) + c\sigma_T(T_1, T_2) + d\sigma_A(A_1, A_2)}{(a + b + c + d)}$$

For both the training and the tests, when measuring the similarity between the extracted event and that of the oracle, we emphasise the semantic similarity by setting the semantic weight a to 20 and the other dimensions' weights b, c and d to 1. The aim was to reinforce the extraction of event types that the analyst finds interesting (defined here as *Bombing*), and hence

influence the choice of gazetteer. Note that the heavily weighted semantic similarity means that AI still receives a reward for an event which is not interesting, but it is much smaller than that of an interesting event.

To illustrate this, let us assume that the events that the analyst is interested in are of type *Bombing* and that we set a to 20 and b, c and d to 1. If the geographical, temporal and agentive dimensions match the expected event perfectly, then $\sigma_G(G_1, G_2)$, $\sigma_T(T_1, T_2)$ and $\sigma_A(A_1, A_2)$ are all 1.

If the event extracted is of type *Bombing*, then the semantic similarity is also 1. This means that the similarity between events E_1 and E_2:

$$\sigma(E_1, E_2) = \frac{20 \times 1 + 1 \times 1 + 1 \times 1 + 1 \times 1}{(20 + 1 + 1 + 1)} = \frac{23}{23} = 1$$

If, on the other hand, the event extracted is *not* of type *Bombing*, then the semantic similarity is 0. This means that the similarity between events E_1 and E_2:

$$\sigma(E_1, E_2) = \frac{20 \times 0 + 1 \times 1 + 1 \times 1 + 1 \times 1}{(0 + 1 + 1 + 1)} = \frac{3}{23} \simeq 0.13$$

.

8.2.1 Training the AI

We took a set of 100 *GTD* documents, 64 of which described the "interesting" *Bombing* events, 17 *Injure* events, and 19 of which contained no extractable events. We passed these documents through BIMBO 30 times in total. Note: in hindsight we could probably have trained the AI on far fewer documents, but we were young, foolish, and pessimistic.

As we said in chapter 7, the actions we made available to the AI were to choose a service from {*Tika, NGramJ, GATE, Geo*}, to choose a GATE gazetteer, or to *STOP* the treatment and return any extracted events.

We took three pairs of gazetteers, each pair consisting of a list of verbs and a list of nouns. This gave the AI six possible gazetteer choices. Two of these pairs, the *bombing* (respectively *injure*) verbs and nouns contained words likely to trigger the detection and extraction of a event of type *Bombing* (respectively *Injure*) were copied from the production gazetteers. The third pair consisted of *dummy* lists and contained verbs and nouns not present in the *GTD*. We hoped the AI would take into account the analyst's preference by favouring the *bombing* gazetteers over the *injury* gazetteers, and that it would learn to ignore the *dummy* lists as they never lead to an extraction.

We wanted to decrease the exploration and increase the exploitation with time. This is because although the exploration can sometimes be useful, especially at the start, to find better treatment combinations, usually it results in partial, faulty or missed extractions which we cannot allow to continue indefinitely. We therefore started with an exploration rate, ϵ of 0.4, then every 5×100 documents, we divided this by 2 until we reached 0.1.

Recall the Q-Learning update formula for the Q-values that we gave in Algorithm 1 in chapter 5 on page 39:

$$\hat{Q}_{t+1}(s_t, a_t) \leftarrow \hat{Q}_t(s_t, a_t) + \alpha_t(s_t, a_t)(r_{t+1} + \gamma \max_b\{\hat{Q}_t(s_{t+1}, b)\} - \hat{Q}_t(s_t, a_t))$$

The value α serves as a modifier on the latest reward gained, whether it be positive or negative. As the documents were heterogeneous, some containing extractable events, and others not, we set the learning rate α to be 0.2. This means that convergence to the exact value function cannot be guaranteed as the rewards will never be given in their entirety to the AI, but it does dampen the negative effect of large variations, for example, if the several documents in a row do not contain extractable events, resulting a long stretch of negative rewards.

8.2.2 Testing the trained AI

In order to test the policy learnt by the trained AI, we set its exploration rate ϵ and its learning rate α to zero. This meant that it followed the learnt policy with no exploration (that it, it always chooses the optimal action), and never updated the Q-values, as the update formula for the Q-values becomes:

$$\hat{Q}_{t+1}(s_t, a_t) \leftarrow \hat{Q}_t(s_t, a_t) + 0 \times (r_{t+1} + \gamma \max_b\{\hat{Q}_t(s_{t+1}, b)\} - \hat{Q}_t(s_t, a_t))$$

or in other words:

$$\hat{Q}_{t+1}(s_t, a_t) \leftarrow \hat{Q}_t(s_t, a_t)$$

To test the untrained AI, we set the learning rate α to be 0.2, and the exploration rate ϵ to be 0.4 as for the training, but we divided ϵ by 2 every 100 documents until it reached 0.1.

We gave both AIs the choice of all six gazetteers (the *bombing*, *injure* and *dummy* nouns and verbs). The expert chain was given only the *bombing* gazetteers, meaning that it could extract only the *Bombing* events. This meant that the expert chain was optimised to extract the events defined as being of interest to the analyst.

We fed BIMBO 1000 documents chosen randomly from the *GTD* that the AIs "saw" for the first time. Only 29% of these contained "interesting" *Bombing* events, 7% *Injure* events, and 64% no extractable events.

As a reminder of the quality measure that we detailed in Definition 7, page 69, BIMBO gave the AI a positive reward for an extraction (unless it was completely dissimilar to the oracle), and a negative reward based on the time it took if it extracted nothing.

The results for the untrained AI (starting from "scratch"), the trained AI and the expert chain are shown in Figure 8.1.

We can see that the untrained AI, after a shaky start, managed to start consistently extracting events after 220 documents. This shows that it generalises well, its performance improving the more different documents it encountered.

Comparing the trained AI and the expert, we can see that the trained AI demonstrated an expert-like policy, extracting 100% of the possible events. This shows that it also generalises well, applying a learnt policy successfully to unknown documents. The only problem was when there was nothing to extract, it did not stop the chain rapidly enough, which is why the negative rewards are lower for the trained AI than for the expert chain.

We also note that the AI learnt to order the chain, for example in the initial state:

language	" "	*nounGazetteer*	none
format	" "	*verbGazetteer*	none
interesting	*false*	*nbServices*	0-5
any	*false*	*timeTakenSoFar*	0

the best action learnt was to pass the document to the service Tika to extract the document metadata and to convert it to an XML format which is essential for the following services to function.

In the state corresponding to a document where at least one event is extracted, whatever the values of the other attributes:

language	en	*nounGazetteer*	any
format	text/plain	*verbGazetteer*	any
interesting	any	*nbServices*	any
any	*true*	*timeTakenSoFar*	any

the AI stopped the treatment of that document. It had learnt that once any event had been extracted, even if it was not "interesting", it was not going to do better. Indeed, with one target event per document summary, and the services that we offered it, this was the case. Even if it passed several

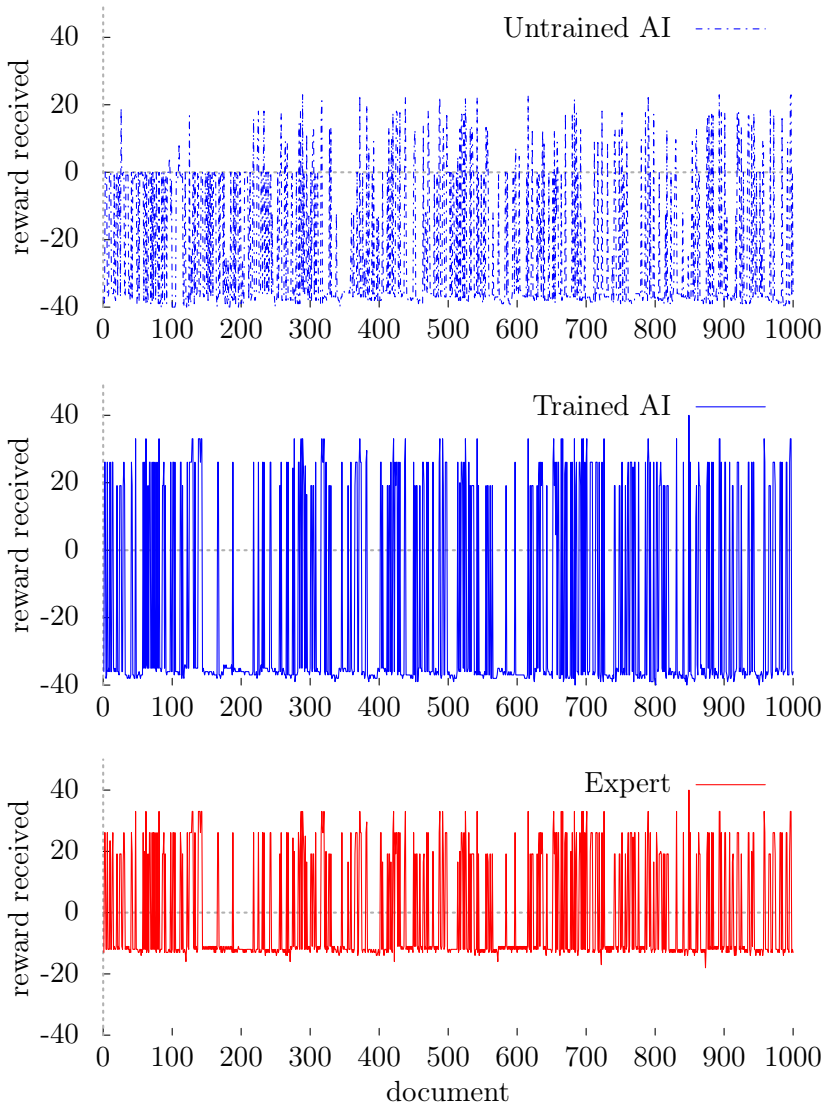


Figure 8.1 – The rewards received by the untrained AI, the trained AI and the expert chain over 1000 unknown documents.

times through the extractor, it could only extract the same event again and again. We could imagine that if we were to add a translation service, and extractors of different languages, then the AI might be able to wring more information out of the document by doing a first extraction in one language, then translating and doing another in a different language.

As hoped, the AI also learnt to optimise the chain. It did not call the service *Geo*, and only used the *bombing* verb list.

As we remarked in the example decision process in section 7.2, it also made an unexpected choice of gazetteer, finally preferring *injure* nouns to those of *bombing*. On investigation, we found that the GATE service provided erroneously only uses the verbs for event extraction, ignoring the nouns. This shows that the AI was able to discover strategies which were not clear even to the expert calibrating the chain, and could be an unexpected advantage of our approach.

In development, for example, by running the AI as we have here on a set of test documents whose events are already known, we could test new services with two objectives. Firstly, we can show that they improve the extractions, and secondly, we can ensure that they do not exhibit any unwanted behaviour. Unexpected choices by the AI point directly to the part of the service concerned (here the choice of the noun gazetteer for GATE), and could be a useful debugging tool.

In production, this test could also be carried out on a regular basis using the current pool of services. If a service exhibits a surprising behaviour, an alert could be raised with the development team, and another "safe" service substituted automatically whilst the erratic one is investigated.

8.3 Few extractable events

We saw in Figure 8.1 that the untrained AI was able to learn to extract events from a set of documents, only 36% of which contained extractable events. In this section, we push this percentage even lower, and test an untrained AI on 5000 unknown randomly chosen documents, only 22.56% of which contain extractable events. Figure 8.2 shows the percentage of extractable events averaged for readability over groups of 100 documents, and we can see that it is not at all evenly spread. For example, the average number of extractable events in documents 1701–1800 drops as low as 7%, but rises to 46% just afterwards in documents 1801–1900.

We removed the semantic bias, setting the event dimension weights a, b, c and d to all be 1, but we kept the interesting event type as *Bombing*.

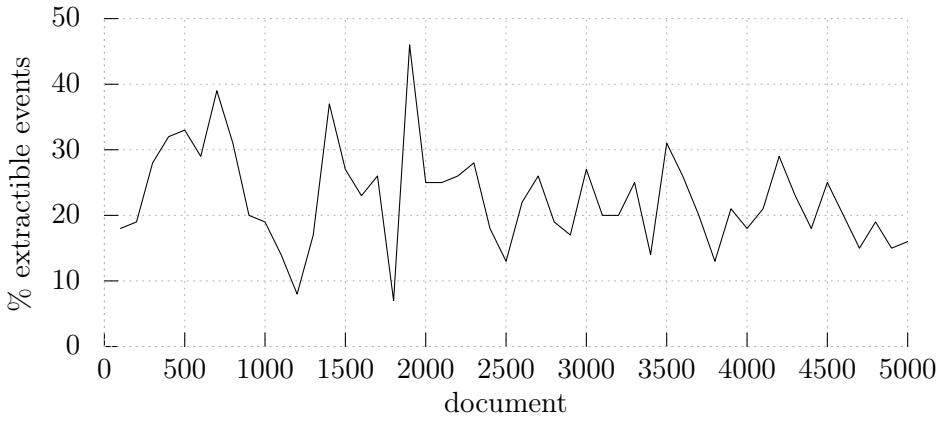


Figure 8.2 – Percentage of documents containing extractable events by 100 documents for tests with few extractable events.

To further handicap the AI, we also increased the action space by adding two more gazetteers, a pair of event specific gazetteers *shooting* (nouns and verbs), and a pair of all-purpose gazetteers *mixed* (nouns and verbs) which contain longer lists of words likely to result in the detection of several types of events. We sought to verify that the AI learnt to use the *mixed* gazetteers rather than the unique event gazetteers, that it still avoided the *dummy* gazetteers, and that it was not sensitive to a larger action space.

So why should the AI prefer the *mixed* gazetteers to the *bombing* gazetteers if the interesting event type was still *Bombing*? In fact, the *mixed* gazetteer allows all the *Bombing* events *plus* all the other types of event to be extracted. All extractions get rewarded, and this reflects the fact that the analyst is more likely to prefer to see too many extracted events, rather than risk missing one because it has been misclassified.

This means that the AI now has the choice of 15 possible actions - the services {*Tika, NGramJ, GATE, Geo*}, *STOP*, plus ten *gazetteers* (*bombing*, *shooting*, *injure*, *mixed* and *dummy* nouns and verbs).

Once again, we set the learning rate α to be 0.2, and the exploration rate ϵ to be 0.4 initially, divided ϵ by 2 every 100 documents. This time, however, we reduced the minimum value of ϵ to 0.05. As we explained for the first set of tests, we want to reduce the exploration over time so that the AI can start choosing the optimal actions more and more frequently. These tests were much longer than the previous set, and we wanted the AI to stabilise

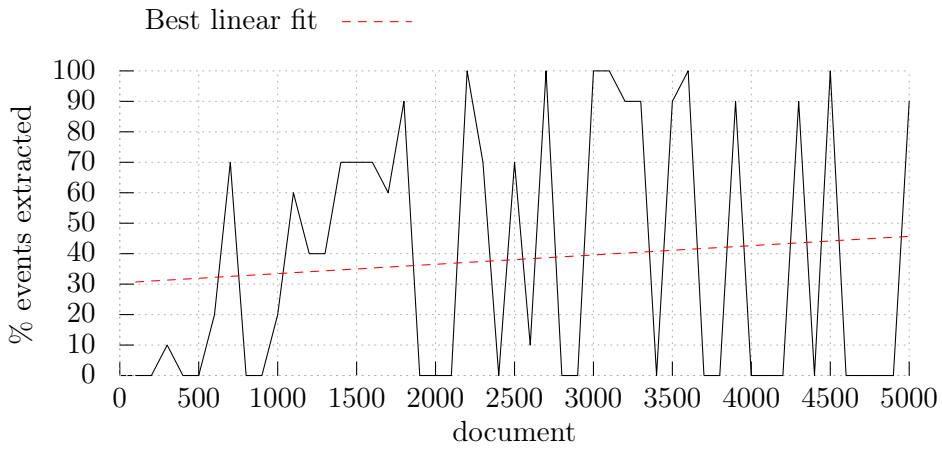


Figure 8.3 – Percentage of events extracted using the policy learnt after each 100 documents.

even further.

We show the rewards received by the AI and by the expert chain (by group of 100 documents) in Figure 8.4, Figure 8.5 and Figure 8.6.

In parallel with the learning, every 100 documents, we initialised an AI with the policy learnt at that point. We set the values of ϵ and α to 0 (recall that this means no exploration, and no learning), and tested this policy on 10 unknown documents, all of which contained extractable events. The results are shown in Figure 8.3, and our first observation is that the AI improves, even if the learnt policies are not consistently capable of extracting 100% of the events.

I'm now going to ask you to juggle three different graphs (sorry) so that we can analyse them together. Firstly the percentage of extractable events by groups of 100 documents (Figure 8.2). Secondly Figure 8.3 showing the quality of the policy learnt at the end of each group of 100 documents. Thirdly Figure 8.4 which shows the rewards received by the AI compared to those of the expert chain (again by groups of 100 documents).

In Figure 8.4, we can see that the first extraction did not happen until the document number 173, and that the few extractions that the AI manages from the first 600 documents are of lower quality than those of the expert chain. This ties in with what we observe in Figure 8.3 - the learnt policies are not capable of extracting more than 1 out of 10 of the events.

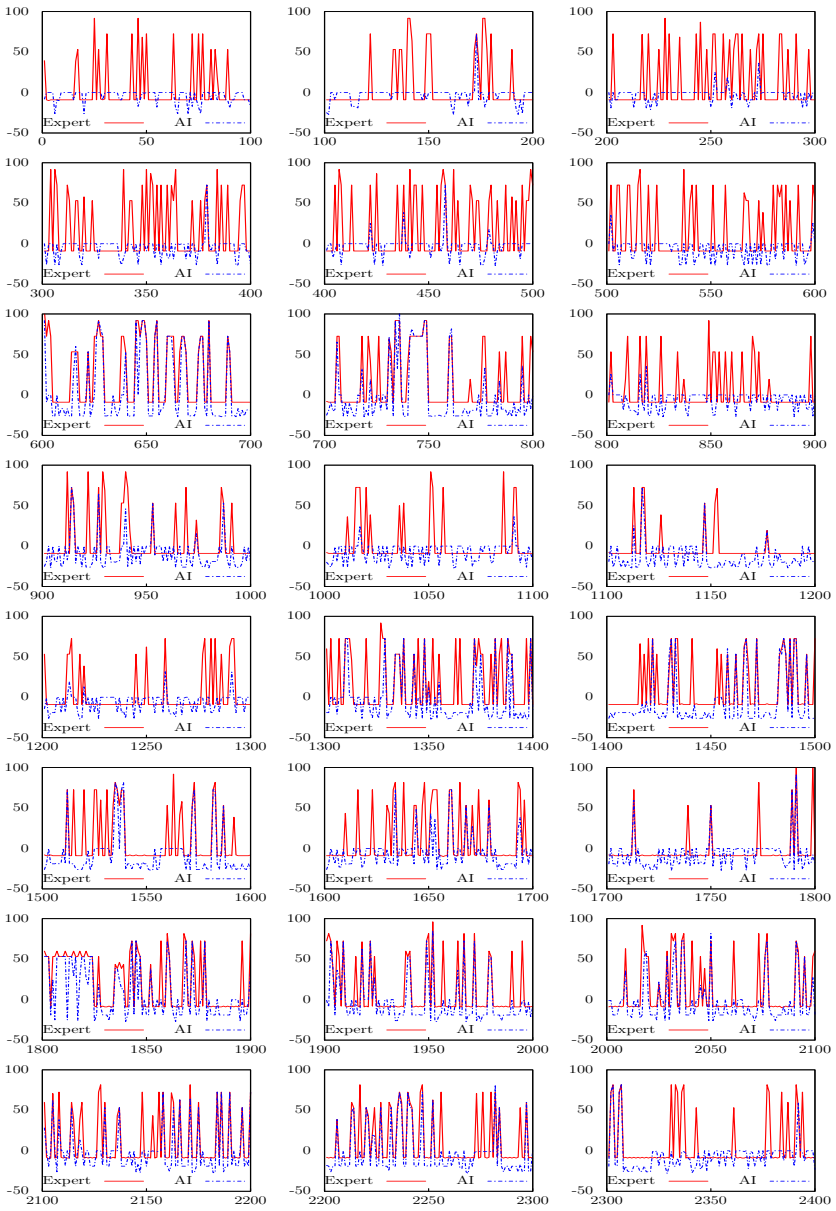


Figure 8.4 – Extraction quality for the expert chain against that of the untrained AI for documents 1 - 2 400 with a very low percentage of extractable events.

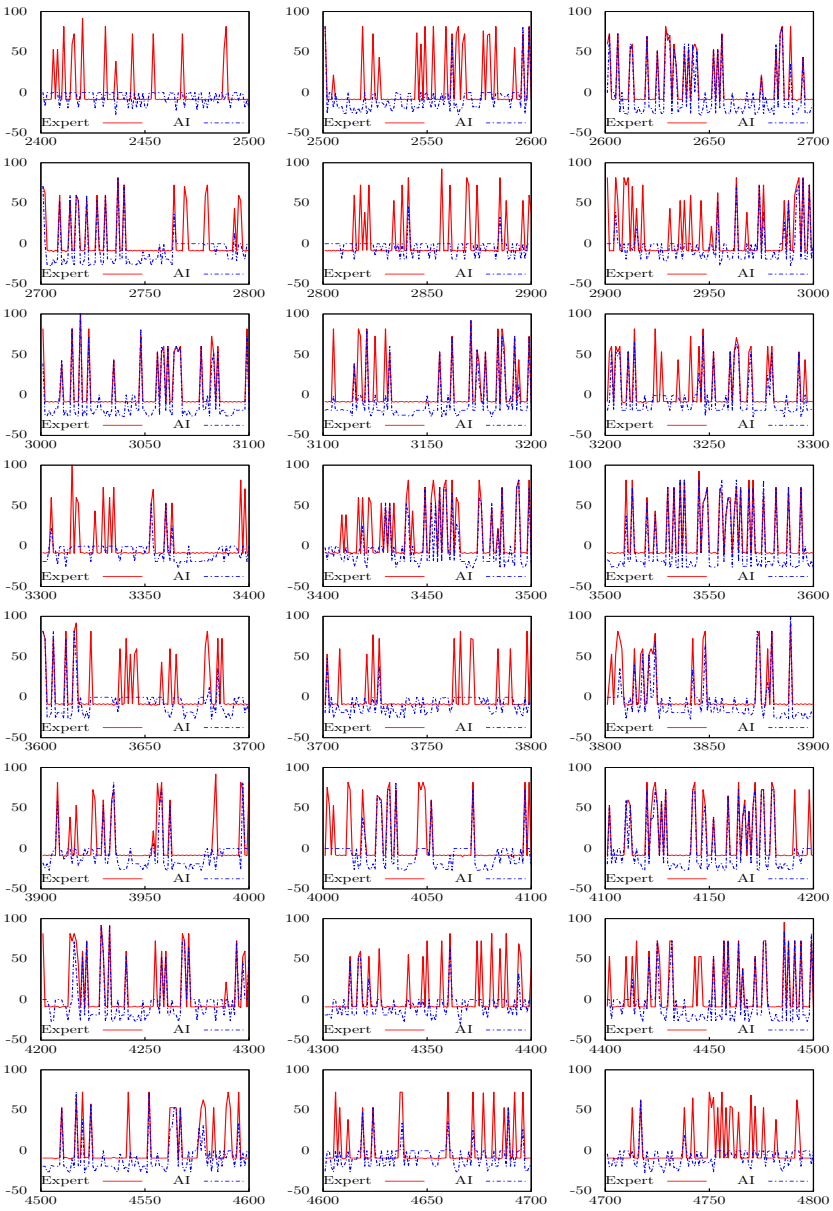


Figure 8.5 – Extraction quality for the expert chain against that of the untrained AI for documents 2 401 - 4 800 with a very low percentage of extractable events.

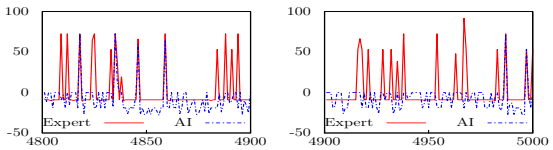


Figure 8.6 – Extraction quality for the expert chain against that of the untrained AI for documents 4 800 - 5 000 with a very low percentage of extractable events.

Between documents 601 to 700 (where Figure 8.2 shows us that 39% of the documents contain extractable events), the quality of the AI's extractions seems to improve to equal that of the expert chain, and Figure 8.3 confirms this - the policy learnt after treating document 700 can extract 7 out of 10 events.

Around document 750 (where 31% of the documents contain extractable events), the extraction quality becomes poor again, and between documents 800 to 900 (where 20% of the documents contain extractable events), the AI starts to treat almost all the documents as rapidly as possible suggesting a risk averse behaviour. It prefers to receive a reward of 0 for not doing anything to the document, rather than receiving a negative reward by trying unsuccessfully to extract an event which does not exist. Again, this is confirmed by testing the learnt policy and seeing no extractions.

Between documents 900 to 1300 (where 19%, 14%, 8% and 17% respectively of the documents contain extractable events), we can see that the percentage of events extracted is very low, and that the AI has problems both with the quality and quantity of its extractions. We can see, however, that the quality of the learnt policy is actually increasing.

The AI starts extracting events again between documents 1301 to 1700 (where 37%, 27%, 23% and 26% respectively of the documents contain extractable events) showing that the AI can still learn with relatively few rewards. Checking the learnt policies confirms this impression.

Documents 1701 to 1800 are interesting. Only 7% contain extractable events, and yet the AI still manages to extract just over half of the events. Maybe once the AI has learnt a good policy, it can keep a certain inertia even during a period of famine or maybe it was just lucky. Checking the learnt policy, it seems that it was not luck; after treating document 1800, the AI is capable of extracting 9 out of 10 events with the policy that it has learnt so far.

The extractions from documents 1801 to 2300 (where 46%, 25%, 25%, 26% and 28% respectively of the documents contain extractable events) seemed to benefit from the peak of 46% in extractable events, as after a shaky start, the quality and quantity of the extractions seems quite consistently good. However, the learnt policy tells a different story. After documents 1900, 2000 and 2100, the AI cannot extract a single event. This seems counter-intuitive until we remember the very low rate of extractable events just before. The learnt policy has obviously been damaged by the low rate of 7%, and any extractions are due to luck until after document 2200 when we see that the AI has at last managed to learn an expert-like policy, extracting 10 out of 10 possible events.

Between documents 2301 to 2400 (where 18% of the documents contain extractable events) the AI starts well, but then fails to extract most of the events. This is echoed by the learnt policy, which degrades from 7 out of 10 to zero extractions.

We can conclude that the AIs results are certainly affected by the percentage of extractable events, but that it seems to only need around 20–25% of the documents to contain extractable events to learn a good to expert-like policy.

8.4 Sporadic rewards

The previous sets of experiments aimed to assess the ability of the AI to extract events from unknown documents, not all of which contained extractable events. In this section, we tested the AI on 5000 documents again, but this time they all contained extractable events. Our aim was to see how quickly the AI could learn an expert-like policy (able to extract all events) given sporadic rewards. We gave the AI the choice of the services or *STOP* plus ten *gazetteers* as in section 8.3. In the graphs below, we only show the results for the documents for which the AI exploited, *i.e.* followed its best policy for the whole treatment (which is why the red expert line varies between plots). We also smooth the curve for readability, taking averages on sets of 100 documents in the same order as they are treated.

We performed three tests:

- First a baseline test in which BIMBO gave rewards after each document, just as we've done so far. This led us to discover a strange behaviour if the AI is allowed too much exploration, and caused us to rethink our ϵ reduction strategy.

- Naturally the analyst is not available *24×7* to consult each document as it is treated. We therefore ran a similar experiment, but BIMBO gave rewards on all extractions only once every 100 documents, hence preventing the AI from learning during those 100 documents.

- Finally, the analysts are not dedicated trainers, and will not correct all extractions. BIMBO therefore only gave a reward at the end of each 100 documents with a probability of 10% on each extraction (otherwise the AI received no reward at all for that extraction).

Initially, Q-learning was run with parameters similar to those of the first experiments: $\alpha = 0.2$ and $\epsilon = 0.4$, divided by 2 every 100 documents until $\epsilon = 0.1$. In running the first baseline test, however, we noted that although the agent learnt an expert-like policy very quickly, sometimes the performance degraded after a little while (see Figure 8.7) because the AI was choosing to STOP rather than treat the document.

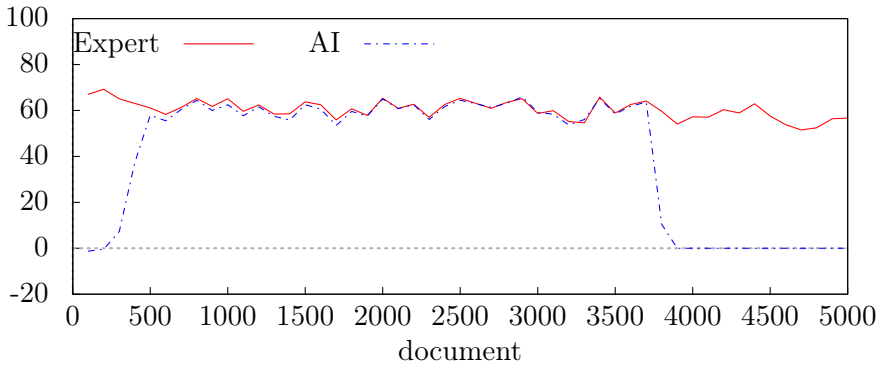


Figure 8.7 – Odd behaviour from the AI with full feedback and ϵ divided by 2 every 100 documents to a minimum value of 0.1.

On examining the logs, we saw that the AI successfully learns a path s_1 to s_n (see Figure 8.8) where s_n results in an extraction.

Then if the exploration rate is kept too high, in the state s_j, the explore actions gradually lower the Q-value of the pair (s_i, a_i) until the action a_i is a worse option than STOP (which gives a reward of 0 if done rapidly). The probability of not exploring between s_j and s_n is very low, as the chain of actions is potentially very long. For example, the time taken to change a gazetteer is negligible, and the AI might choose to do it several hundred times in one treatment. Dahl and Halck (2001, Chapter 2) noted a similar

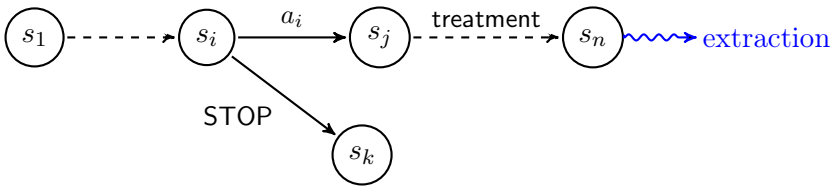


Figure 8.8 – The state s_j is pivotal. Explore actions from it can reduce its perceived value until the best action in s_i is STOP.

problem with exploring. They claim that "the feedback from the exploring actions will be biased and push the evaluations away from the correct ones".

To confirm our suspicions, we increased the exploration even more, by only dividing ϵ by 2 every 200 documents. In Figure 8.9, we can see that this does indeed have a detrimental effect.

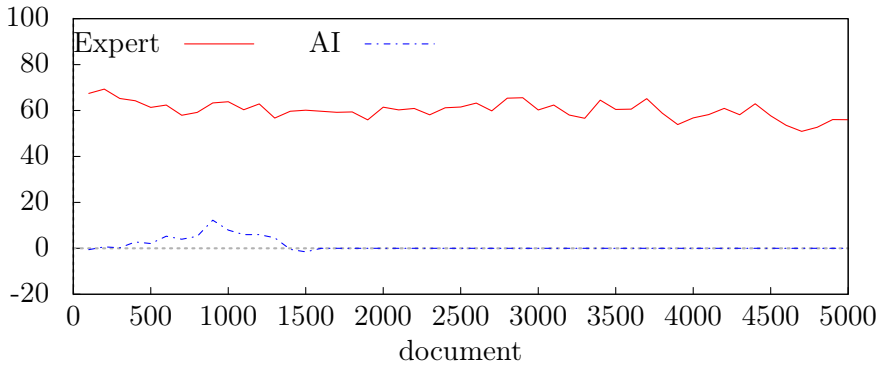


Figure 8.9 – Even worse behaviour from the AI with full feedback and increased exploration as ϵ is divided by 2 every 200 documents.

We therefore tried a more intelligent reduction strategy for epsilon. Sutton and Barto (1998, page 186, Example 6.6) state that if ϵ is gradually reduced, then Q-Learning will asymptotically converge to the optimal policy. Our problem was therefore finding out which "gradually" worked in our case.

We used

$$\epsilon(s) = \frac{1}{(1 + n(s))^{\beta}}$$

where $\epsilon(s)$ is the epsilon value to use in state s and $n(s)$ is the number of visits to state s so far. To guarantee convergence, we must choose β such that $0.5 < \beta \leqslant 1.0$ as Sutton and Barto (1998, page 53, Equation 2.8) show

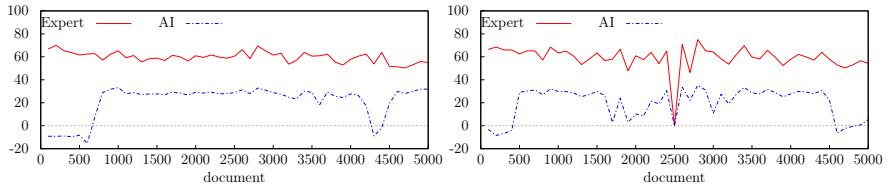
the conditions required to assure convergence with probability 1 are:

$$\sum_{k}^{\infty} \epsilon = \infty \text{ and } \sum_{k}^{\infty} \epsilon^2 < \infty$$

and we know that

$$\sum_{k}^{\infty} \frac{1}{n^r} \text{ converges if } r > 1 \text{ and diverges if } r \leqslant 1$$

We first tried the most aggressive value of $\beta = 1.0$ (see Figure 8.10a), then $\beta = 0.75$ (see Figure 8.10b), but found that these cut short the exploration too soon, and the AI never learnt an expert-like policy.



(a) $\beta = 1.0$ (b) $\beta = 0.75$

Figure 8.10 – Full feedback with aggressive non-exploration strategies based on the number of visits to each state.

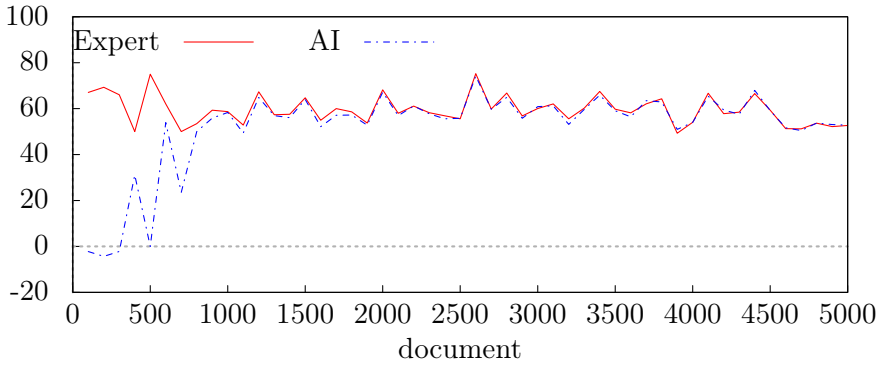


Figure 8.11 – Full feedback with an exploration strategy based on the number of visits to each state where $\beta = 0.55$.

We then set $\beta = 0.55$ with good results (see Figure 8.11). As a reminder, for this test, BIMBO gave the AI a reward for each document. After 400

documents, the average quality over 100 documents was good, and by the time the AI had treated 800 documents, it was able to extract 100% of the events. More importantly, the extraction quality with this policy was on a par with that of the expert chain. We also note that the ability to extract events and the ability to extract *correct* events increase together, as the lower quality (compared to the expert chain) that we observe in the first part of plot was due to the processing time, and not the similarity with the target event.

As we said, the analyst is not available to consult each document as it is treated. During the test where BIMBO gave a delayed reward only once 100 documents had been treated (Figure 8.12), the AI was understandably slightly slower to learn, but after 700 documents, it had learnt to extract 100% of the events, and as you can see, the extraction quality was very good.

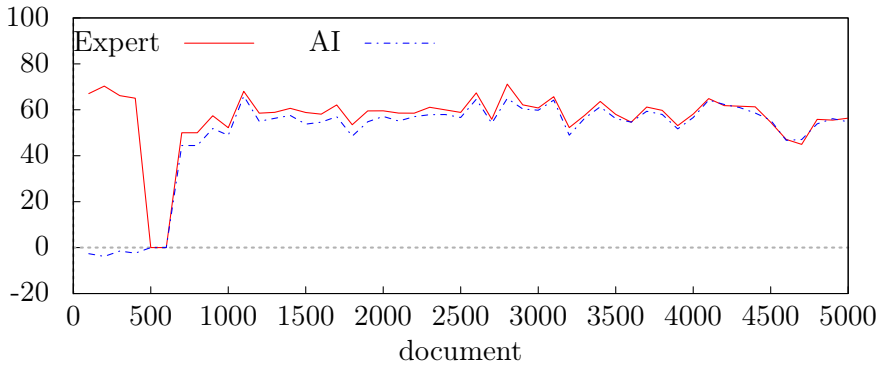


Figure 8.12 – Full feedback delayed until 100 documents have been treated.

The final test emulated the fact that analysts cannot correct all the documents. BIMBO therefore only gave a reward at the end of a group of 100 documents with a probability of 10% on each extraction (otherwise the AI received no reward at all for that document). Even with such delayed, sporadic rewards, the AI managed to extract 100% of the events after 1200 documents (see Figure 8.13).

Incidentally, you may be wondering why, if all the documents contain extractable events, you can see that both the AI and expert lines dip to zero around document 2500 in Figure 8.10b, 400–600 in Figure 8.12, and five times in Figure 8.13. Recall that we chose to only show the results for treatments where no sub-optimal action was chosen (*i.e.* the AI did not explore). For these documents it explored at least once during their treatment.

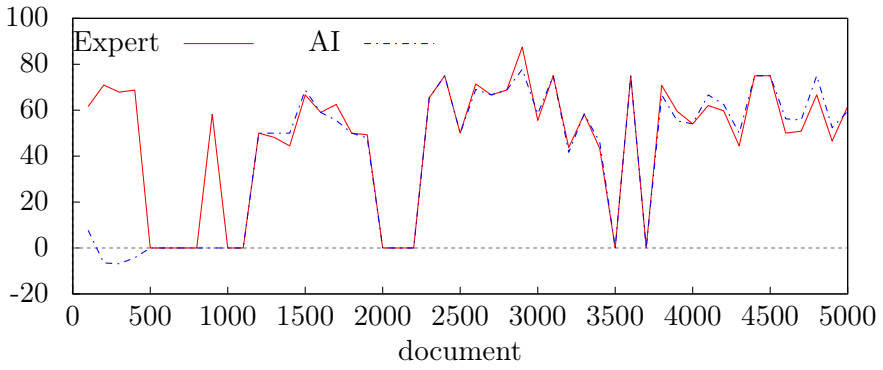


Figure 8.13 – Partial feedback delayed until 100 documents have been treated.

8.5 Summary

In this section, we performed three types of test in which we gave the AI numeric rewards, which were also used to measure the quality of its results. We compared how well untrained and trained AIs performed against an expert chain and showed that the AI consistently performs well, even with very sporadic feedback, providing that at least 20 – 25% of the documents contain extractable events. We also showed that the AI is capable of generalising on unknown documents.

The AI's unexpected choice of gazetteer to parametrise the GATE service led us to discover a previously undetected error in that service.

Finally, we observed that the AI sometimes exhibited a strange behaviour, appearing to "forget" what it had learnt. On investigation, we found that the exploration rate remained too high, and so we carried out some tests with a more sophisticated exploration reduction method.

Chapter 9

RL with intuitive feedback

9.1 Non-numerical rewards

Traditionally, reinforcement learning algorithms use numerical rewards. This has the advantage of being precise, allowing the agent to learn exactly which policies are the best, and indeed, as we showed in chapter 8, our approach works very well with this type of user feedback. The trouble is, most humans are not comfortable giving precise numeric values. Imagine that you greet your neighbour "How are you this morning?". He is highly unlikely to reply "I'm 7.5 today, 3.2 better than yesterday". He will probably respond "I'm fine thanks, much better than yesterday".

In the same way, when we ask the analyst to define his requirements, it is highly improbable that he can specify a precise numerical value for a similarity such that "an edit on a character costs 0.3", "a character deletion costs 0.6", "a missing first name of length 7 costs 10.5" *etc.* Or for the treatment time, "15.2 seconds costs 7.9", and so on. Not only is it nigh on impossible to be exhaustive in the list of possible outcomes, but it is also extremely difficult to evaluate each possibility quantitatively.

Our objective was therefore to find a way of learning from qualitative feedback, which the analyst could define easily and intuitively.

Weng et al. (2013) point out that the definition of a numerical reward function is non-intuitive, especially when this reward does not represent a physical measure. They treat this problem by offering the agent an ordinal reward from a categorical, completely ordered scale. Weng and Zanuttini (2013) treat a similar problem of non-numeric rewards which can be ordered by a tutor. Both reformulate the Ordinal Reward MDP (ORMDP) (Weng, 2011) as a Vector Reward MDP (VMDP). Similarly, in Gilbert et al. (2015a), the user is asked to provide comparisons, expressed as value vectors, rather than give a specific numerical reward.

To illustrate this idea of value vectors:
As a reminder, a policy π is a set of pairs $\{(s, a)\}$, i.e. in a given state s, a is the (best) action to perform. An MDP models the probability (as observed by the agent) for each state, for all actions from that state, of arriving in s'

and getting reward r.

Instead of using numerical values for the rewards, they are given a "label", such as w_i (for example, "an extraction in 10 seconds", or "no extraction"), which is associated automatically with a final state.

Following π then gives a vector whose indices are the probabilities of receiving each reward, e.g.

$$(p_1, p_2, p_3, p_4) \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix}$$

means that the agent receives the first reward w_1 with probability p_1, reward w_2 with probability p_2, and so on.

For Weng and Zanuttini (2013), these strictly ordinal rewards, whilst more natural than a numerical user feedback, can lead to non-intuitive questions for the user : do you prefer $2w_1 + 7w_3 + 3w_4$ to $3w_1 + w_2 + 2w_5$? Those comparisons can be complicated and non-intuitive to evaluate. For instance, could the analyst judge if it is better to have to change one letter of a name, delete a place, add a year and have an extraction two seconds faster than to have to add a name, change the month, but have an extraction one second slower? Using the technique of Weng and Zanuttini (2013) would also mean potentially asking the user questions during the treatment of the document (at the calculation of the policy), which does not fit in with their normal work-flow.

We also considered using the more relaxed landmarks of qualitative reasoning (see Travé-Massuyès et al. (2003) for an introduction). This gives us the comparisons $+, -, 0, ?$ (better than, worse than, similar to, or incomparable with the previous). Comparing "nothing extracted" with "a perfect extraction", we could assume $-$, for instance, or if something was extracted, we would rely on the user to tell us if it were $+$ or 0. This rather Orwellian approach of "double plus good" (Orwell, 1950) still requires the user to make a non-intuitive cognitive judgement (where would they draw the line between $+$ and 0, for example), and to be consistent in those judgements.

Humans are not very good at giving precise numerical values, or evaluating complex vectors, but they are usually excellent at making simple pairwise comparisons, so we turned to preference based reinforcement learning (PRBL) (Busa-Fekete et al., 2014; Akrour et al., 2012; Fürnkranz et al., 2012; Wilson et al., 2012; Wirth and Fürnkranz, 2013a,b; Wirth et al., 2016). This is the integration of two sub-fields of machine learning, namely preference learning and reinforcement learning.

For example, in Fürnkranz et al. (2012), they ask the user for their preferences over simulated roll-outs or trajectories. In Figure 9.1, we see that from a given common state s_1, the agent simulates the trajectories formed by taking each possible action from that state (the "roll-outs"), and then following a given policy until the final states. Maximizing the expectancy of cumulated rewards cannot be done directly as the numerical values of those rewards are not available, but the outcomes can be given a preference order, for instance that τ_1 is preferred to τ_2, and that τ_2 is preferred to τ_3. Knowing this preference order, we can then infer a preference order over the actions from the common state s_1: that a_1 is preferred to b_1, which is preferred to c_1. This means that we can infer that $\hat{Q}(s_1, a_1)$ is greater than $\hat{Q}(s_1, b_1)$ which is in turn greater than $\hat{Q}(s_1, c_1)$.

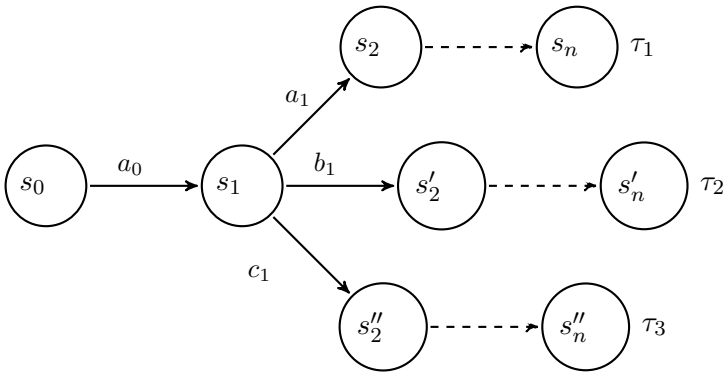


Figure 9.1 – Roll-outs are carried out from s_1, giving a preference order: τ_1, τ_2, τ_3 over the trajectories, and hence over the actions from s_1.

In our case, as we saw in chapter 6, we define our states as vectors of values, one of which is the detected language. We could therefore imagine that in Figure 9.1, s_1 could be a state where the language detected is Afrikaans. The action a_1 could be a direct extraction from the original Afrikaans, resulting in final state s_n. However, Matthew (2015) states that the extraction of named entities from Afrikaans has been neglected in favour of languages such as Dutch. Our South African friends tell us that Afrikaans is similar to Dutch, so we offer an action b_1 which is to translate from Afrikaans to Dutch, and then to use the Dutch extraction rules to finish in state s'_n. We also know that our richest named entity extraction rules are for the English language, and so we try a third action c_1, to translate the Afrikaans into English, and then extract, presenting the results in state s''_n. We then ask the user to compare the three results, and from their preferences we can infer

which action is the best in state s_1.

This is expensive in terms of processing and user time, however, as the same document would be treated three times, and the comparison between the results would still not necessarily be intuitive, nor would it enter into the analyst's normal daily routine. Also, in attempting to construct a consistent reward function from the preference orders, we risk introducing preferential information which was not given explicitly by the user, therefore giving the agent an unintended bias. For example, the user may prefer τ_1 to τ_2 until he finds out that τ_1 was produced by sheer luck, and that normally the "a_1" path would produce τ_4, which is the worst possible result.

9.2 SSB model-based approaches

An approach which adapts the preference order decision criteria to ordinal information is *probabilistic dominance* (Busa-Fekete et al., 2014). Probabilistic dominance aims to maximize the probability of yielding a preferred outcome. More formally, let π_1 and π_2 be two policies and let F_1, F_2 be two random variables on the final states $(f_1, \ldots, f_k)$ (in our case, the states where the AI chooses the action *STOP*, or we force it to stop) where $\mathbb{P}(F_i = f_j)$ is given by the probability of π_i achieving result f_j, and $F_x \succeq F_y$ indicates that the result achieved by π_x is preferred or equal to that achieved by π_y. Then according to the probabilistic dominance criterion:

$$\pi_1 \succeq \pi_2 \Leftrightarrow \mathbb{P}(F_1 \succeq F_2) \geq \mathbb{P}(F_2 \succeq F_1)$$

In words, policy π_1 is preferred to π_2 if the probability of π_1 doing better than π_2 is greater than the converse.

Probabilistic dominance is actually just a specific type of Skew Symmetric Bilinear (SSB) utility function (Fishburn, 1984). Given ϕ, an SSB utility function, $\phi(\pi, \pi') > 0$ means that the user prefers π to π' (and conversely $\phi(\pi, \pi') < 0$ means that he or she prefers π' to π). If they are similar or incomparable from the user's point of view, then $\phi(\pi, \pi') = 0$. In probabilistic dominance, we restrict the values of ϕ to 1, 0, or -1, which means that the user need only provide a purely ordinal feedback, without having to specify the degree of preference.

Gilbert et al. (2015b) showed that it is possible to learn a (potentially mixed) optimal policy with only qualitative feedback, using SSB utility functions to compare policies in MDPs. This frees us from the restriction of totally ordered rewards, allowing the reward function to be expressed in a

completely natural manner. The user can indicate preferences over final states, rather than having to give numerical or ranked rewards.

We entered into collaboration with Hugo Gilbert (LIP6, Université Pierre et Marie Curie, Paris) to expand on Gilbert et al. (2015b) with respect to our industrial challenge.[1]

Suppose that the preferences of the decision maker are represented by an SSB function φ applied on probability distributions over the set of final states $\mathcal{F}$ (the set of possible feedback). Let s_0 be a fixed initial state (in our context, this would be as an unknown document arrives for treatment, *i.e.* with no information yet extracted, no services called, zero seconds on the clock, *etc.*), π, π' be two policies and $p^{\pi|s_0}(f), p^{\pi'|s_0}(f)$ be the probability distributions induced by π and π' over final states given s_0. In other words, $p^{\pi|s_0}(f)$ is the probability that the execution of policy π from initial state s_0 finishes in final state f. The preferences of the decision maker are defined as follows:

$$\pi \succeq \pi' \Leftrightarrow \varphi(p^{\pi|s_0}(f), p^{\pi'|s_0}(f)) \geq 0$$

where

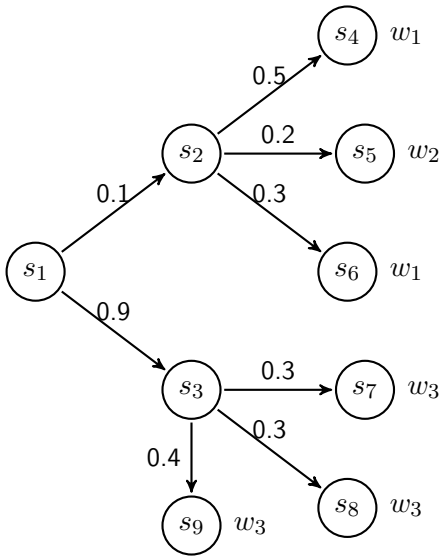
$$\begin{aligned}\varphi(p^{\pi|s_0}(f), p^{\pi'|s_0}(f)) &= \sum_{s,s' \in \mathcal{F}} \mathbb{P}(f = s|\pi, s_0)\varphi(s, s')\mathbb{P}(f = s'|\pi', s_0) \\ &= {}^{\mathrm{t}}(p^{\pi|s_0}(f))\Phi p^{\pi'|s_0}(f)\end{aligned}$$

where $\Phi[i, j] = \varphi(s_i, s_j)$.

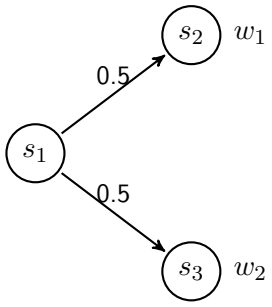
Example 8 *To take a concrete example, in Figure 9.2, we see two policies. In the first, in state s_1, if we take the best action, we have a 10% chance of going to s_2, and a 90% chance of going to s_3. From s_2, we can arrive in states s_4, s_5 and s_6. Both s_4 and s_6 offer the same reward w_1.*

If we currently use policy π_1 (Figure 9.2a) and we find the second policy π_2 (Figure 9.2b), how can we tell if π_2 is better than π_1 (is $\pi_2 \succ \pi_1$)? Imagine that the users have told us that the final state w_3 is preferred to w_2, which is in turn preferred to w_1, that is $\varphi(w_3, w_2) = 1$ and $\varphi(w_2, w_1) = 1$. Obviously $\varphi(w_i, w_i) = 0$ as the same result can be neither better nor worse than itself. Note again that we never have to ask for a numerical value of w_i, just whether w_i is preferred to w_j or not.

[1] Hugo Gilbert provided the technical proof, we provided the motivation and the experiments, and we all collaborated on the early design of the algorithm



(a) Policy π_1



(b) Policy π_2

Figure 9.2 – Two policies, where we have defined the final states in terms of wealth levels, or rewards.

First we work out the vector of probabilities that we receive each reward (accumulated wealth), w_i, for each policy:

$$\begin{aligned}
\textit{for } \pi_1 : p_w^{\pi_1} &= (p(w_1 \mid \pi_1), p(w_2 \mid \pi_1), p(w_3 \mid \pi_1)) \\
&= (0.1 \times (0.5 + 0.3), 0.1 \times 0.2, 0.9 \times 1.0) \\
&= (0.08, 0.02, 0.9) \\
\textit{for } \pi_2 : p_w^{\pi_2} &= (p(w_1 \mid \pi_2), p(w_2 \mid \pi_2), p(w_3 \mid \pi_2)) \\
&= (0.5, 0.5, 0)
\end{aligned}$$

We then compare the value of each policy by doing a pairwise comparison of the wealth (reward) vectors, and multiplying by the function $\varphi(w, w')$ which tells us if w is preferable to w':

To compare π_2 with π_1, we work out $\varphi(\pi_1, \pi_2)$

$$
\begin{aligned}
&= \sum_{w,w'} p(w \mid \pi_1) p(w' \mid \pi_2) \varphi(w, w') \\
&= (0.08 \times .5) \times \varphi(w_1, w_1) + (0.08 \times .5) \times \varphi(w_1, w_2) + (0.08 \times 0) \times \varphi(w_1, w_3) \\
&+ (0.02 \times .5) \times \varphi(w_2, w_1) + (0.02 \times .5) \times \varphi(w_2, w_2) + (0.02 \times 0) \times \varphi(w_2, w_3) \\
&+ (0.09 \times .5) \times \varphi(w_3, w_1) + (0.09 \times .5) \times \varphi(w_3, w_2) + (0.09 \times 0) \times \varphi(w_3, w_3)
\end{aligned}
$$

$$
\begin{aligned}
&= (0.04) \times \varphi(w_1, w_1) + (0.04) \times \varphi(w_1, w_2) + (0) \times \varphi(w_1, w_3) \\
&+ (0.01) \times \varphi(w_2, w_1) + (0.01) \times \varphi(w_2, w_2) + (0) \times \varphi(w_2, w_3) \\
&+ (0.045) \times \varphi(w_3, w_1) + (0.045) \times \varphi(w_3, w_2) + (0) \times \varphi(w_3, w_3)
\end{aligned}
$$

$$
\begin{aligned}
&= 0.04 \times 0 + 0.04 \times (-1) + 0 \times (-1) \\
&+ 0.01 \times 1 + 0.01 \times 0 + 0 \times (-1) \\
&+ 0.045 \times 1 + 0.045 \times 1 + 0 \times 0
\end{aligned}
$$

And so, the comparison between the two policies,

$$
\begin{aligned}
\varphi(\pi_1, \pi_2) &= 0 - 0.04 - 0 + 0.01 + 0 - 0 + 0.045 + 0.045 + 0 \\
&= 0.06
\end{aligned}
$$

In other words, $\varphi(\pi_1, \pi_2) > 0$, so π_1 is preferred to π_2 (and intuitively, we can see that this is the case).

To apply this to our treatment chain we need to define a preference order over the final states. Each final state has candidate options described on k user-defined criteria $(c_1, \ldots, c_k)$. For example, each final state could be a vector, and the criteria could be:

- extraction time t
- event dimension corrected
- event confirmed correct
- event deleted
- event not consulted

- *etc.*

This could then give final state vectors:

$\vec{v}$=(time, agentive error [Boolean], spatial error [Boolean], ...)
or $\vec{w}$=(time, should have extracted something, ...)
or $\vec{x}$=(time, was right not to extract anything, ...)
etc.

If these criteria are ranked by order of priority $c_1 \geq c_2 \geq \ldots \geq c_k$, then the preferences between two candidate options o_1 and o_2 are:

$$\begin{aligned} o_1 \succeq o_2 \Leftrightarrow & c_1(o_1) > c_1(o_2) \\ & \text{or } c_1(o_1) = c_1(o_2) \text{ and } c_2(o_1) > c_2(o_2) \\ & \text{or } c_1(o_1) = c_1(o_2) \text{ and } c_2(o_1) = c_2(o_2) \text{ and } c_3(o_1) > c_3(o_2) \\ & \text{or } \ldots \end{aligned}$$

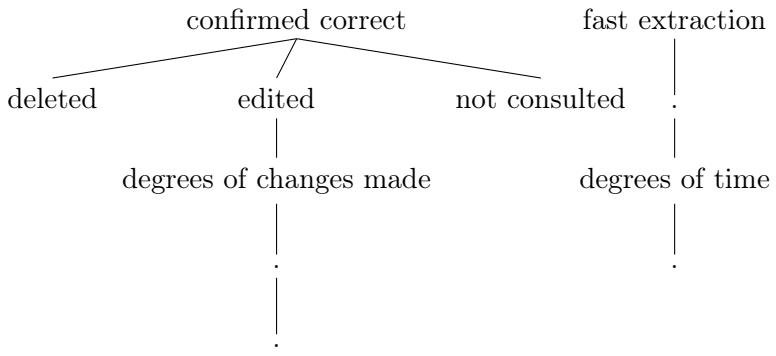
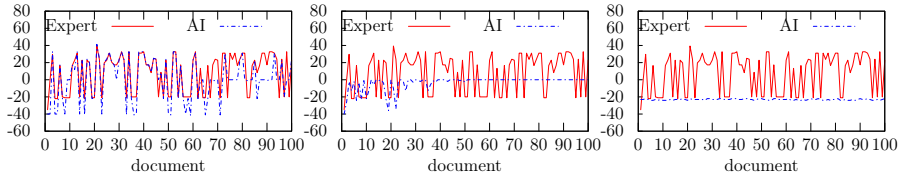


Figure 9.3 – A potential partial preference relation over final state criteria.

If the partial preference relations are as defined in Figure 9.3, for instance, then "confirmed correct" is preferred to "edited", but the user would refrain from comparing "deleted" to "not consulted" or to "fast extraction".

Then $\phi(\vec{v_1}, \vec{v_2}) > 0$ if they are equal on all dimensions but one, and $\vec{v_1} > \vec{v_2}$ on this dimension.

Hugo suggested two model-based algorithms which could be adapted to an SSB type reward: R-Max (Brafman and Tennenholtz, 2003) and V-Max (Rao and Whiteson, 2011), which learn the underlying MDP instead of a policy directly.



(a) Q-Learning (b) R-Max (c) V-Max

Figure 9.4 – Numerical rewards received by the AI after training over 30 $\times$100 documents with immediate feedback, compared with the expert chain.

R-Max builds a model of the environment in which it finds itself, initialising the rewards for unknown parts to a maximum to encourage exploration. It then uses the model to make decisions, and updates the rewards to their observed values after trying a given transition m times. In this way, once a transition is considered "learnt", the agent goes off to explore other areas which still offer the attractive maximum reward.

V-Max is similar to R-Max, but instead of updating the transitions only after trying one transaction m times, they are updated after each visit. In this way, V-Max uses the information it acquires a lot more efficiently.

We implemented both algorithms with numeric rewards initially, which we hoped would be a stepping stone to a qualitative (SSB) implementation, and compared them to the Q-learning algorithm that we saw in chapter 8.

For the tests, similarly to subsection 8.2.1, 30 iterations (batches) of 100 documents were passed through BIMBO to train the AIs. This time, however, the dimension weightings were equal ($a = b = c = d = 1$).

We can see the results in Figure 9.4. Each figure shows the quality of results of the 30^{th} iteration as a blue dotted line, compared with the expert chain as a red solid line, and includes the explorations.

As expected, the Q-Learning algorithm performs very well (see Figure 9.4a), and the results (for example, towards the end) that are sometimes lower than those of the expert were due to exploration.

The R-Max results were very disappointing (see Figure 9.4b). It explores a lot, and its theoretical guarantees of convergence unfortunately do not translate into a very good practical performance. It tends to explore in depth a tiny sub-space of the space / action couples (s, a) before passing to the next sub-space. In other words, it goes round in circles, before going elsewhere, then it turns in circles in that "elsewhere" before going somewhere else, *etc.* The good news was that it tended towards rewards of 0 from below which implies an improvement nonetheless, *i.e.* it did not learn to extract

the events, but at least it was quick to extract nothing.

We hoped that V-Max would perform better, but it never even managed to get off the ground, taking a long time to not extract anything (see Figure 9.4c).

Another problem in our particular case with both V-Max and R-Max (and we suppose any other model-based approach), was that the length of the paths followed by the agent rendered the calculation time of the policy (the Bellman backup) prohibitively long. We set a time limit for the treatment, but changing a gazetteer is practically instantaneous, and so the agent may change it several hundreds of times in a few seconds, potentially taking it to a different state each time.

9.3 SSB Q-Learning - a model-free approach

To tackle the problem of a model-based SSB approach not being viable in our case, we proposed a model-free approach based on the SSB function and Q-Learning. Gilbert et al. (2016) details a new algorithm, *SSB Q-learning* (see Algorithm 2). We invite you to read the original article for more details, but we give an overview of the general SSB Q-Learning algorithm here. Our formalisation of the user feedback for our application is detailed in chapter 10.

When we ask the analyst to define his requirements in the simplest terms, they are likely to reply something like "I prefer a false extraction to a missed extraction, as I don't want to lose information, and I'd rather it were as fast as possible". Note that there are two preferences expressed here, which although they impact each other (one might suppose that speed could result in poor extraction quality), cannot naturally be linked numerically. We therefore want to reward the agent based on the decoupled user preferences: "I prefer an extraction to no extraction" and "I prefer it to be as fast as possible". We've already seen that this preferential information can be expressed as a partial order over possible results achieved $(f_1, \ldots, f_k)$ (seen as final states) given by the human agent. For instance f_i can stand for "a good extraction in 10 seconds" or "no extraction in 5 seconds", *etc.*

Like Q-learning, *SSB Q-learning* uses epsilon greedy (EG) exploration, and it updates the Q-values similarly. However, instead of having the numerical values of the rewards given by the environment, they are defined by ϕ and the past experiences of the agent *i.e.* the frequencies $\mathbf{p}_t$ with which it has achieved each possible final result so far (the resulting numerical value is denoted by $\mathcal{R}_{\mathbf{p}_t}(s_{t+1})$). Intuitively, the algorithm continuously tries to act better, according to ϕ, than it has so far (Algorithm 2).

Algorithm 2: SSB Q-learning (Gilbert et al., 2016)

Data: MDP $\mathcal{M}$, SSB function φ

while *True* **do**
- Choose a_t using the EG exploration strategy
- Play a_t, observe s_{t+1}, and let $r_{t+1} = \mathcal{R}_{\mathbf{p}_t}(s_{t+1})$
- $\hat{Q}_{t+1}(s_t, a_t) \leftarrow$ $\hat{Q}_t(s_t, a_t) + \alpha_t(s_t, a_t)(r_{t+1} + \max_b\{\hat{Q}_t(s_{t+1}, b)\} - \hat{Q}_t(s_t, a_t))$
- **if** s_{t+1} *is a final state* f_i *and exploration is off* **then**
  - $\mathbf{p}_{t+1} = \mathbf{p}_t + \beta_p(\mathbf{1}_i - \mathbf{p}_t)$
  - # β_p is $\frac{1}{\eta+1}$.
  - # η is the number of times $\mathbf{p}$ has been updated.

Example 9 *We will expand Example 7 of the Rowett dice game given in the article (Gilbert et al., 2016) here to aid understanding. The game consists of three non-standard dice A, B and C, and the aim is to throw the highest total score. The player first chooses to throw the die A (action a_A) or not (action a_{BC}). In the latter case, he then must choose to throw either die B (action a_B) or die C (action a_C). The dice are six-sided, but the sides are not numbered 1 to 6. Instead, die A has one 1 and five 4's, die B has five 3's and one 6, and die C has three 2's and three 5's. Incidentally, these dice have the interesting property of being non-transitive, which means that A will usually beat B, B will usually beat C, and C will usually beat A. The probabilities of throwing each number for each die are shown in Table 9.1.*

Table 9.1 – The Rowett dice probabilities.

	1	2	3	4	5	6
p_A	1/6	0	0	5/6	0	0
p_B	0	0	5/6	0	0	1/6
p_C	0	1/2	0	0	1/2	0

Figure 9.5 shows the current position. We're halfway through a game at time t, and the Q-values learnt so far are:

$$Q(s_0, a_A) = 0.1;\ Q(s_0, a_{BC}) = 0.2;\ Q(s_{BC}, a_B) = 0.3;\ Q(s_{BC}, a_C) = -0.7$$

The probability vector $p_t = (\frac{1}{5}, 0, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$ shows that the agent thinks that there is a 1 in 5 chance of throwing a 1, 3, 4, 5 or 6, but zero chance of

throwing a 2. The dice have been thrown $n = 80$ times in total, 20 times die A (episode of length 1), and 30 times either B or C (episode of length 2). This means that a final state has been reached 50 times, so $\eta_p = 50$, and $\beta_p = 1/(\eta_p + 1) = 1/51 \simeq 0.02$. The state-action pair (s_0, a_{BC}) has been updated 30 times so $\eta_{s_0,a_{BC}} = 30$ and $\alpha_t(s_0, a_{BC}) = 1/31^{2/3} \simeq 0.1$. Similarly, $\eta_{s_{BC},a_B} = 20$ and $\alpha_t(s_{BC}, a_B) = 1/21^{2/3} \simeq 0.13$

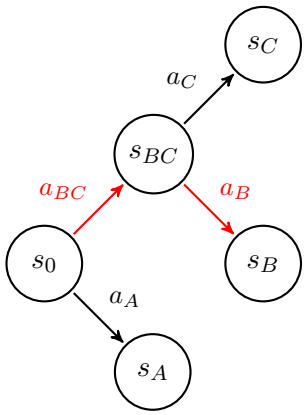


Figure 9.5 – The player first chooses not to throw die A, then chooses to throw die B.

From state s_0, the player follows the policy, and as $Q(s_0, a_A) < Q(s_0, a_{BC})$ has chosen not to throw die A. They are now in state s_{BC} and the Q-value $Q(s_0, a_{BC})$ is updated as in Algorithm 2. That is:

$$Q(s_0, a_{BC}) \leftarrow Q(s_0, a_{BC}) + \alpha_t(s_0, a_{BC}) \times (\mathcal{R}_{\mathbf{p}_t}(s_{BC}) + \max_b\{\hat{Q}(s_{BC}, b)\} - \hat{Q}(s_0, a_{BC}))$$

Because we're not yet in a final state, $\mathcal{R}_{\mathbf{p}_t}(s_{BC}) = 0$, so

$$\begin{aligned} Q(s_0, a_{BC}) &\simeq 0.2 + 0.1 \times (0 + 0.3 - 0.2) \\ &= 0.21 \end{aligned}$$

Again following the policy, the player will now throw the die B, and

$$Q(s_{BC}, a_B) \leftarrow Q(s_{BC}, a_B) + \alpha_t(s_{BC}, a_B) \times (\mathcal{R}_{\mathbf{p}_t}(s_B) + \max_b\{\hat{Q}(s_B, b)\} - \hat{Q}(s_{BC}, a_B))$$

So

$$Q(s_{BC}, a_B) \simeq 0.3 + 0.13 \times (\mathcal{R}_{\mathbf{p}_t}(s_B) + 0 - 0.3)$$

Because we're in a final state, $\mathcal{R}_{\mathbf{p}_t}(s_B) = 1_i^T \Phi p_t$. *The aim is to roll high, so a roll of 3 (*r_3*) on the die will beat a 1 (*r_1*), that is,* $\phi(r_3, r_1) = 1$, *but a 6 (*r_6*) would beat the 3. This means that if we roll a 3, then* $i = 3$ *and*

$$
\begin{aligned}
\mathcal{R}_{\mathbf{p}_t}(s_B) &= 1_3^T \Phi p_t \\
&= (0,0,1,0,0,0) \begin{array}{c} \\ r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{array} \begin{array}{c} \begin{array}{cccccc} r_1 & r_2 & r_3 & r_4 & r_5 & r_6 \end{array} \\ \begin{pmatrix} 0 & -1 & -1 & -1 & -1 & -1 \\ 1 & 0 & -1 & -1 & -1 & -1 \\ 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 \\ 1 & 1 & 1 & 1 & 0 & -1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array} \begin{pmatrix} 1/5 \\ 0 \\ 1/5 \\ 1/5 \\ 1/5 \\ 1/5 \end{pmatrix} \\
&= \frac{1}{5} - \frac{1}{5} - \frac{1}{5} - \frac{1}{5} \\
&= -\frac{2}{5}
\end{aligned}
$$

Therefore

$$
\begin{aligned}
Q(s_{BC}, a_B) &\simeq 0.3 + 0.13 \times (-0.4 - 0.3) \\
&\simeq 0.2
\end{aligned}
$$

We update the probabilities:

$$
\begin{aligned}
p_{t+1} &= p_t + \beta_t(1_i - p_n) \\
&= (\frac{1}{5}, 0, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}) + \frac{1}{51} \times (1_3 - (\frac{1}{5}, 0, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})) \\
&= (\frac{1}{5}, 0, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}) + \frac{1}{51} \times (-\frac{1}{5}, 0, \frac{4}{5}, -\frac{1}{5}, -\frac{1}{5}, -\frac{1}{5})) \\
&= (\frac{51}{255}, 0, \frac{51}{255}, \frac{51}{255}, \frac{51}{255}, \frac{51}{255}) + (-\frac{1}{255}, 0, \frac{4}{255}, -\frac{1}{255}, -\frac{1}{255}, -\frac{1}{255})) \\
&= (\frac{50}{255}, 0, \frac{55}{255}, \frac{50}{255}, \frac{50}{255}, \frac{50}{255}) \\
&= (\frac{10}{51}, 0, \frac{11}{51}, \frac{10}{51}, \frac{10}{51}, \frac{10}{51})
\end{aligned}
$$

This approach, as you will see in chapter 10 works very well. The restriction imposed that the rewards can only be given in the final states obviously is not a problem for our use-case, and even works to our advantage, because we only want feedback on the whole treatment, *i.e.* the final state representing the results.

Chapter 10

Experiments with intuitive feedback

10.1 Framework

In chapter 8, we carried out a proof of concept of our approach. We modelled the document treatment chain as an MDP, and we saw that the AI successfully learnt how to construct an efficient chain using reinforcement learning with numerical rewards.

Here, we change the experimental context and test algorithms which are only given qualitative, non-numerical rewards against a baseline which receives numerical rewards.

As we've already stated in section 9.3, to cater for qualitative, intuitive user feedback, which standard approaches cannot handle, we use *SSB Q-learning* (Gilbert et al., 2016). Because this algorithm was hot off the press, we wanted to put it through its paces. We tried it with a variety of parameters to test its robustness, comparing it with a standard Q-Learning with the same parameters.

We saw in section 2.5 that Knox and Stone (2015) suggest that where the agent should be task-focused, the discount value γ should be high. We therefore chose to use two values for γ: 0.9 and 1.0.

We've also seen (for example in section 8.4), that the amount of exploration that the agent is allowed to make has a big impact on the success of its learning. We tried two reduction strategies for the exploration rate, or EG parameter ϵ. A slow reduction (divided by 2 after 2500 documents) and a faster one (divided by 2 after 1000 documents) down to a minimum in both cases of 0.05.

With both SSB Q-Learning and Q-Learning, we used *eligibility traces*. We go into a little more detail of what these are in chapter 12, but recommend that you read Sutton and Barto (1998, Chapter 7) for an excellent explanation. When using eligibility traces, a *decay parameter* $\lambda \in [0, 1]$ controls how far new experiences are back-propagated along the trace of decisions taken thus far. At one extreme, $\lambda = 0$ corresponds to the algorithms

as described in Algorithm 1 and Algorithm 2, *i.e.* with no back-propagation along the trace. At the other extreme, $\lambda = 1$ corresponds to Monte-Carlo-like RL algorithms with full back-propagation. We wanted to find a value that was not too small, not too big, but just right (Southey, 1837), and so we tried three values for λ: 0, 0.95 and 1.0.

For each setting of these parameters, we compared the performance of the expert chain and those of three AIs learning "from scratch":

- **QL:** Q-learning(λ) with numerical feedback, as we used in chapter 8, but with eligibility traces;
- **MAG:** SSB Q-learning(λ) as presented in chapter 9, expressing preferences of different **MAG**nitudes:
  $\phi \in \{-1000, -100, -10, 0, 10, 100, 1000\}$
  (see subsection 10.1.1 for more details).
- **DOM:** SSB Q-learning(λ) as presented in chapter 9, with probabilistic **DOM**inance, that is, purely ordinal feedback:
  $\phi \in \{-1, 0, 1\}$
  (see subsection 10.1.2 for more details);

To summarise, this gave us 36 combinations of tests:

- Discount factor γ: 0.9 and 1.0.
- EG parameter ϵ divided by 2 after 2500 documents or 1000 documents.
- Decay parameter λ: 0, 0.95, 1.0.
- Three AIs: QL, MAG and DOM;

To measure the performance of all approaches, we ran them, initially untrained, on a set of 5 000 *GTD* documents (presented in the same order, and only seen once in all experiments) from which the expert chain can extract events.

The available actions were given (as in section 8.3) by choices from ten gazetteers, the services *Tika, NGramJ, GATE, Geo*, and *STOP*.

We measured the quality of the treatment of each document as described in section 7.6, and emphasise that even though the quality of the extraction was measured in the same way for all approaches (including the expert chain), the feedback given for the AIs to learn from was different.

Both Q-learning(λ) and SSB Q-learning(λ) were run with α set as in Gilbert et al. (2016), *i.e.* decreasing as the number of visits to the current state / action pair grows.

Recall that SSB Q-learning only requires feedback on the relative quality of two extractions. This means that the user no longer has to specify numeric rewards for the AI. Instead, he specifies his preferences in a simple, intuitive way, and we interpret those preferences as final states (representing the treatments finishing on those states).

10.1.1 MAG definition

The first type of SSB feedback, MAG, is a hybrid between a numeric and a purely qualitative feedback. For the MAG feedback, we emphasised the importance of extracting events ("I prefer an extraction to no extraction"), compared to the importance of extracting the exact target events ("I prefer a correct extraction to an incorrect one"), in turn compared to the importance of efficiency ("I prefer it to be as fast as possible").

Intuitively, it is therefore a middle ground between QL and DOM, and can be seen as a weighted form of probabilistic dominance. Yet such feedback remains quite natural. We formalised the MAG feedback given to the AI as follows:

- A final state f is hugely preferred under MAG to final state f', *i.e.* $\phi_{MAG}(f, f') = 1000$ if and only if:
  - f extracted an event and f' did not.
- A final state f is largely preferred under MAG to final state f', *i.e.* $\phi_{MAG}(f, f') = 100$ if and only if:
  - both f and f' extracted an event but f extracted an event of higher quality (as measured in section 7.6) than f'.
- A final state f is slightly preferred under MAG to final state f', *i.e.* $\phi_{MAG}(f, f') = 10$ if and only if:
  - both f and f' extracted an event of similar quality or neither extracted an event, but f was faster than f' by *margin* seconds (where *margin* is parametrisable).
- A final state f is considered equivalent under MAG to final state f', *i.e.* $\phi_{MAG}(f, f') = 0$ if and only if:
  - f is similar to, or incomparable with f' with respect to the previous conditions.

Note: as we initially had no idea how the algorithm SSB Q-Learning(λ) would behave, the values 10, 100 and 1000 were chosen rather arbitrarily. In section 10.4, we explore the effect of varying these magnitudes, and see that although they are not as difficult to refine as the definition of similarity, they still require a small effort to tune correctly.

10.1.2 DOM definition

For DOM, we took the simplest user preferences "I prefer an extraction to no extraction" and "I prefer it to be as fast as possible" to test just how little information we could give the learner. We therefore formalised the feedback given to DOM in our experiments as follows:

- A final state f is preferred under DOM to final state f', *i.e.* $f \succ_{DOM} f' \Leftrightarrow \phi_{DOM}(f, f') = 1$ if and only if:
  - f extracted an event and f' did not, or
  - neither treatment or both treatments resulted in an extraction, but f was faster than f' by *margin* seconds.
- A final state f is considered equivalent under DOM to final state f', *i.e.* $f \sim_{DOM} f' \Leftrightarrow \phi_{DOM}(f, f') = 0$ if and only if:
  - f is similar to, or incomparable with f' with respect to the previous conditions.

We thus encouraged the AI, in an arguably very natural manner, to extract events first, and to do so fast, or to recognise quickly that there are no events to extract. This relies on the correlation, demonstrated in chapter 8, between the ability of the treatment chain to extract any event, and its ability to extract the correct event.

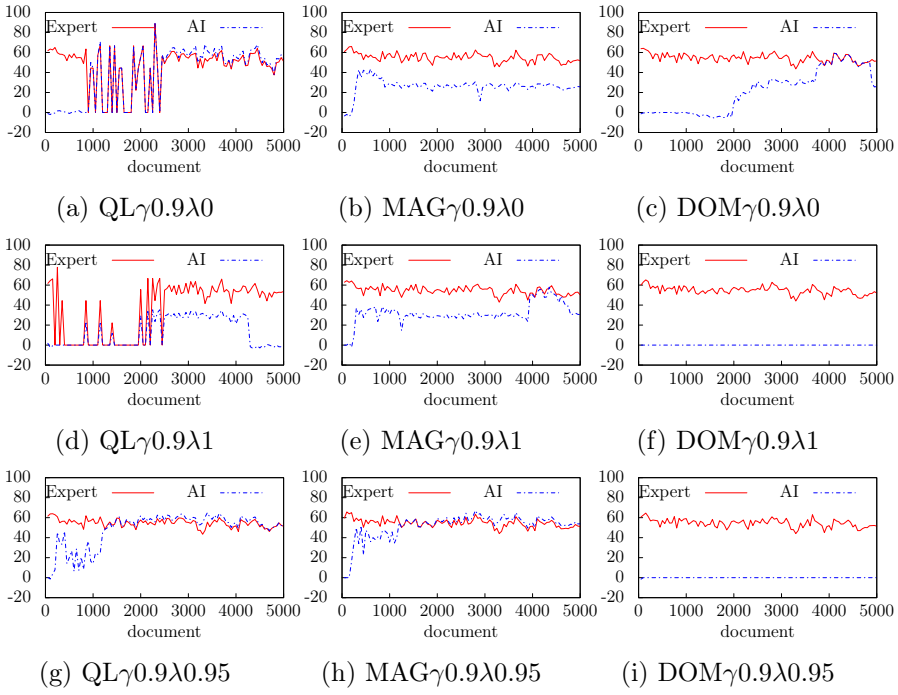
As we stated just above, we took the simplest user preferences for DOM, but obviously, this preference relation could be completed with additional information over the obtained results, such as the perceived quality of the extraction, *etc.*

In these tests, we set *margin* to be 5 seconds which is between $\frac{1}{2}$ and $\frac{1}{4}$ of a typical treatment time (this obviously depends on the speed of the machine used to run the tests, and *margin* should be adjusted accordingly).

10.2 Results

Our intuition was that QL should be more effective than MAG, and MAG more effective than DOM, as QL receives precise numeric rewards which reflect the evaluation criteria, giving it much more information than MAG, and MAG receives more than DOM as it takes into account the quality of the extraction. However, we demonstrate in this section that MAG and DOM are perfectly realistic approaches in an industrial setting, and may even be preferable to QL.

The results are shown in Figure 10.1–Figure 10.4. On each plot, we show the extraction quality of a particular feedback approach (blue dotted line) against that of the expert chain (red solid line). We only consider the results for the documents for which the AI "exploited", *i.e.*, followed its "best" policy for the whole treatment (which is why the red line varies between plots). For readability, we smooth the curve, taking averages on sets of 50 documents in the same order as they are treated.

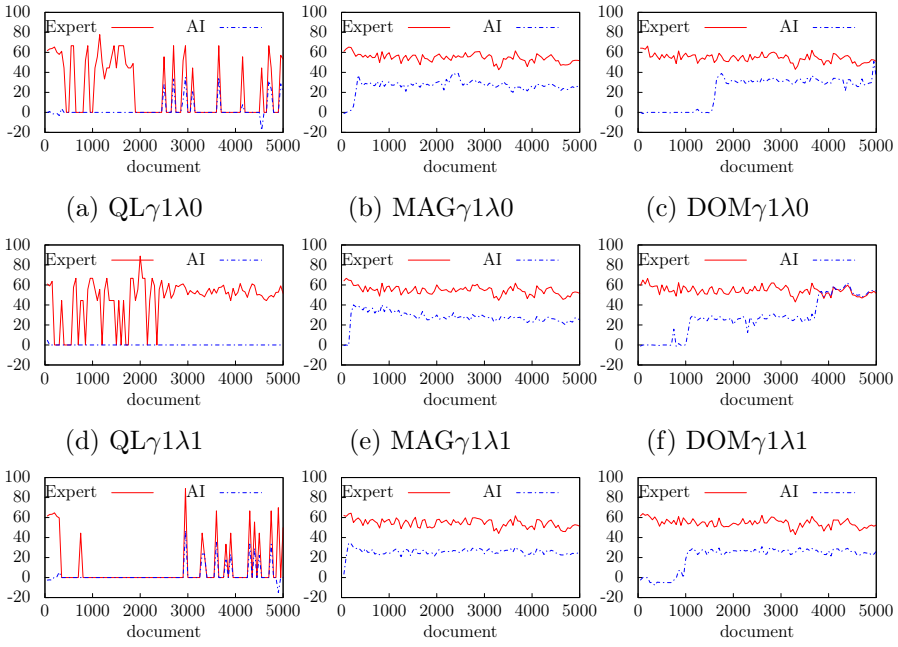


(a) QLγ0.9λ0 (b) MAGγ0.9λ0 (c) DOMγ0.9λ0

(d) QLγ0.9λ1 (e) MAGγ0.9λ1 (f) DOMγ0.9λ1

(g) QLγ0.9λ0.95 (h) MAGγ0.9λ0.95 (i) DOMγ0.9λ0.95

Figure 10.1 – The quality of the results by document for $\gamma = 0.9$; varying λ; slow reduction of $\epsilon = 0.4/2$ every 2 500 documents, minimum 0.05.

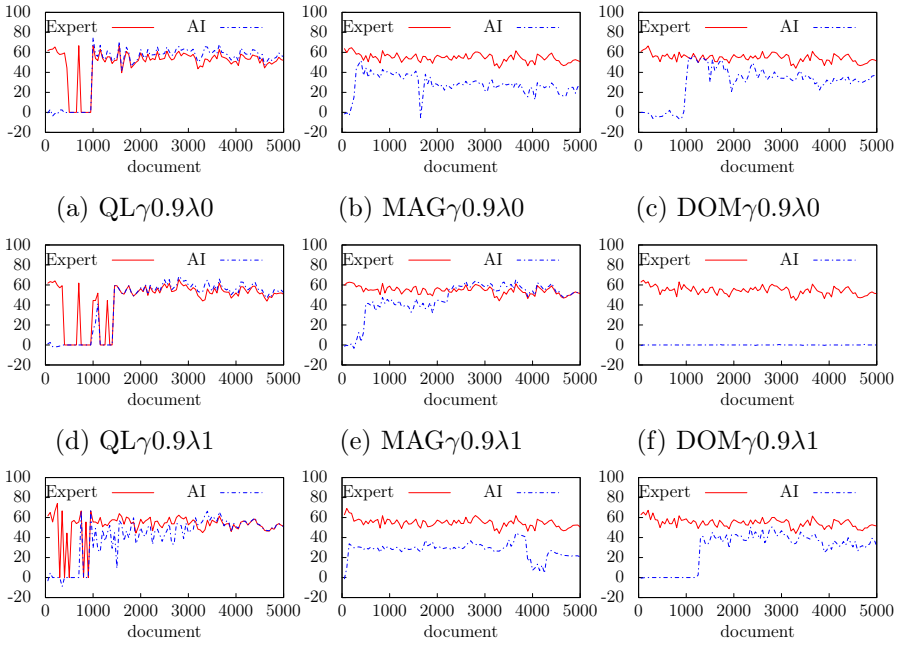


(a) QLγ1λ0 (b) MAGγ1λ0 (c) DOMγ1λ0

(d) QLγ1λ1 (e) MAGγ1λ1 (f) DOMγ1λ1

(g) QLγ1λ0.95 (h) MAGγ1λ0.95 (i) DOMγ1λ0.95

Figure 10.2 – The quality of the results by document for $\gamma = 1$; varying λ; slow reduction of $\epsilon = 0.4/2$ every 2 500 documents, minimum 0.05.

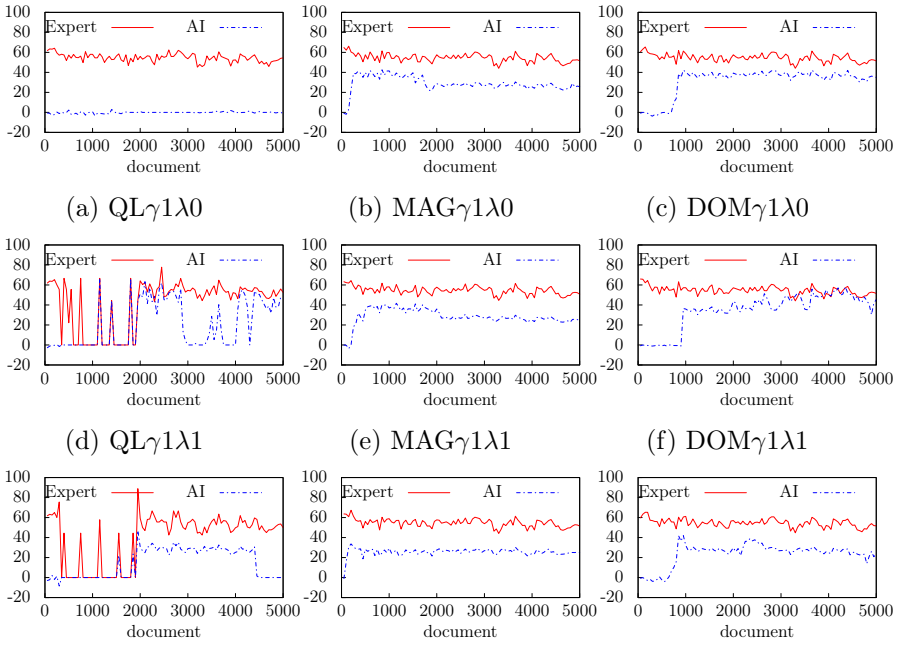


(a) QLγ0.9λ0 (b) MAGγ0.9λ0 (c) DOMγ0.9λ0

(d) QLγ0.9λ1 (e) MAGγ0.9λ1 (f) DOMγ0.9λ1

(g) QLγ0.9λ0.95 (h) MAGγ0.9λ0.95 (i) DOMγ0.9λ0.95

Figure 10.3 – The quality of the results by document for $\gamma = 0.9$; varying λ; faster reduction of $\epsilon = 0.4/2$ every 1 000 documents, minimum 0.05.



(a) QLγ1λ0 (b) MAGγ1λ0 (c) DOMγ1λ0

(d) QLγ1λ1 (e) MAGγ1λ1 (f) DOMγ1λ1

(g) QLγ1λ0.95 (h) MAGγ1λ0.95 (i) DOMγ1λ0.95

Figure 10.4 – The quality of the results by document for $\gamma = 1$; varying λ; faster reduction of $\epsilon = 0.4/2$ every 1 000 documents, minimum 0.05.

Table 10.1 gives a cross-reference of the varying parameters and the resulting graphs with a subjective colour code indicating how good to excellent (light to dark green) or bad to awful (light to dark red) the results were. The first column shows the rate (in documents) at which ϵ is reduced, the second the values of γ tested, the third the values of λ. The fourth column shows the results for Q-Learning(λ) and the last two, the results for SSB Q-Learning(λ) with the definition of ϕ used.

Table 10.1 – A colour-coded easy reference showing the quality of the results for Q-Learning(λ) and SSB Q-Learning(λ) varying the number of documents after which ϵ is divided, and the coefficients γ and λ. Dark red indicates a very poor result, whilst dark green indicates an excellent result.

				SSB QL(λ): Type of ϕ	
divide ϵ every	γ	λ	QL	MAG	DOM
2500 documents	0.9	0	Figure 10.1a	Figure 10.1b	Figure 10.1c
		1	Figure 10.1d	Figure 10.1e	Figure 10.1f
		0.95	Figure 10.1g	Figure 10.1h	Figure 10.1i
	1	0	Figure 10.2a	Figure 10.2b	Figure 10.2c
		1	Figure 10.2d	Figure 10.2e	Figure 10.2f
		0.95	Figure 10.2g	Figure 10.2h	Figure 10.2i
1000 documents	0.9	0	Figure 10.3a	Figure 10.3b	Figure 10.3c
		1	Figure 10.3d	Figure 10.3e	Figure 10.3f
		0.95	Figure 10.3g	Figure 10.3h	Figure 10.3i
	1	0	Figure 10.4a	Figure 10.4b	Figure 10.4c
		1	Figure 10.4d	Figure 10.4e	Figure 10.4f
		0.95	Figure 10.4g	Figure 10.4h	Figure 10.4i

10.2.1 QL results

Glancing at Table 10.1, it is evident that QL is split down the middle with 50% of good to very good results, and two rather inconclusive results for $\gamma = 1$ and $\lambda = 1$ and 0.95.

Examining the more detailed graphs in Figure 10.1–Figure 10.4, we can see that QL can indeed give excellent results, but that it is quite sensitive to changing parameters.

- For both ϵ reduction strategies, the combinations of $\gamma = 0.9$ and $\lambda = 0$ (Figure 10.1a and Figure 10.3a) or $\lambda = 0.95$ (Figure 10.1g and Figure 10.3g) gave excellent results. For example in Figure 10.1g the AI learnt a "good enough" policy after only 250 documents, and a expert-like one after 1200 documents.

- Testing $\gamma = 0.9$ with the faster ϵ reduction strategy (Figure 10.3a, Figure 10.3d and Figure 10.3g) gave excellent results whatever the value of λ.

- With $\gamma = 0.9$ and $\lambda = 1$ and the slower ϵ reduction strategy (Figure 10.1d), however, the results were mediocre. The AI started to extract events around the 2000 document mark, but then after around 4100 documents chose to STOP rather than treat the document. This is likely to be a similar problem to the one we noted in section 8.4, where the explorations were eroding the Q-Value of a key state / action pair.

- With $\gamma = 1$, regardless of the ϵ reduction strategy, and regardless of λ (Figure 10.2a, Figure 10.2d, Figure 10.2g, Figure 10.4a, Figure 10.4d and Figure 10.4g), the results were very bad: the AI learnt to STOP very early, suggesting a risk-adverse behaviour as $\gamma = 1$ puts a heavy emphasis on the future possible rewards.

10.2.2 MAG results

Table 10.1 shows us that MAG's results are consistently good to excellent.

Specifically, from Figure 10.1–Figure 10.4, we can see that the AI with MAG feedback appeared very robust to changes in parameters, quickly learning a "good enough" policy in every case. That is, it learnt more or less rapidly to extract the events, even if it was not always as rapid as the expert chain. In fact, in section 10.4, we show that although it is not sensitive to the changes in $\gamma =$ and λ, it is sensitive the scale of feedback offered.

- Setting $\gamma = 0.9$ and $\lambda = 1.0$ for the faster ϵ reduction strategy (Figure 10.3e) also allowed the AI to learn an expert-like policy.

- The combination $\gamma = 0.9, \lambda = 0.95$ for the slower ϵ reduction strategy (Figure 10.1h) is particularly good, learning a "good enough" policy almost straight away, and improving rapidly by speeding up to achieve an expert-like policy after 3 000 documents.

- When $\lambda = 1$, there is a back-propagation of the rewards along all of the trace (although it may be cut at a certain length depending on the implementation) and the AI has no need to revisit the state / action pairs on the trace to feel the effect of the rewards at state s_0. This means that the exploration can be decreased faster. For $\lambda = 0.95$, however, this back-propagation is shorter, which means that the AI needs to revisit the pairs along the trace more times before the effect of the feedback reaches state s_0. This explains why the slower ϵ reduction strategy was necessary in this case.

10.2.3 DOM results

We can see in Table 10.1 that DOM's results are usually good to excellent, but that when they are bad, they are very, very bad.

Indeed, in Figure 10.1–Figure 10.4, we can see that DOM was fairly sensitive to the choice of γ and λ (but still not quite as much as QL was).

- With $\gamma = 0.9$ and the slower ϵ reduction strategy the results were awful for both $\lambda = 1$ and $\lambda = 0.95$ (Figure 10.1f and Figure 10.1i). Similarly for $\gamma = 0.9$ and $\lambda = 1$ and the faster ϵ reduction strategy (Figure 10.3f). The AI never managed to extract even a single event.

- However, reasonably good results were achieved with $\gamma = 0.9, \lambda = 0$ for the slower ϵ reduction strategy (Figure 10.1c), and with $\gamma = 0.9, \lambda = 0$ and $\gamma = 0.9, \lambda = 0.95$ for the faster strategy (Figure 10.3c and Figure 10.3i).

- With $\gamma = 1$, the results were good to excellent regardless of the ϵ reduction strategy (Figure 10.2c, Figure 10.2f, Figure 10.2i, Figure 10.4c, Figure 10.4f and Figure 10.4i). The best results amongst these are for $\gamma = 1, \lambda = 1$ (Figure 10.2f), which learnt a "good enough" policy after 1000 documents and stabilises with an expert-like policy after 3750. Note that we continued this run on a further 5000 documents (not shown), and it stayed expert-like. This confirms the suggestion of Knox and Stone (2015) that γ should be as high as possible for a task-focused performance.

10.3 Summary

We noticed that the AI was capable of improving on a chain which was already good (where the events were extracted correctly but which took a

few seconds longer than the expert chain). It learnt to speed up, eventually matching the expert chain, for example in Figure 10.3e. This increase in speed is very important, as in the worst cases, it was taking up to 24 hours to run 5000 documents when it should take around half that time.

We also noted that SSB Q-Learning sometimes learnt an expert-like policy but then the performance degrades. For example, in Figure 10.1e and Figure 10.3c, even though the events are still well extracted, it starts taking too long. The logs show that the AI learnt that passing through GATE from a given state gives a good reward, and if γ and λ are not correctly set, although it takes longer, it starts to prefer this action to stopping, and has to be stopped forcibly by BIMBO.

We noted a slight improvement in the results with a faster reduction in ϵ (every 1 000 documents vs every 2 500), that is, with less exploration overall.

QL can give excellent results, but is (very) sensitive to parameter variation and depends on numerical feedback, which is non-intuitive for the user to provide.

MAG seems to offer a good middle ground between a purely numeric and a purely qualitative feedback. It is quite robust to parameter choice (see also section 10.4), uses mostly intuitive feedback, and still learns a good to expert-like policy very quickly.

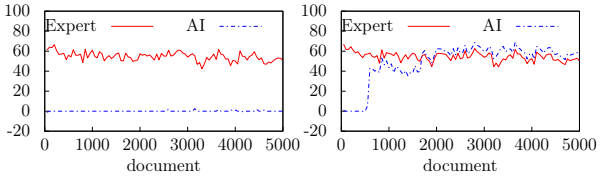
DOM only requires purely ordinal feedback, and yet with the correct parameters is able to learn expert-like policies, proving to be a very interesting approach for situations where it is impossible to gather a precise numerical feedback.

All in all, the experiments demonstrate that it is feasible to automatically improve a document treatment chain, and even to learn one from scratch, in settings where very little or no numerical information is given.

10.4 More MAG results

As we mentioned above, the set of values for ϕ of -1000, -100, -10, 0, 10, 100 and 1000 were chosen by instinct, and we wondered what would happen if we varied them. We took one of the most successful settings for MAG, that is $\gamma = 0.9$, $\lambda = 0.95$ and ϵ divided at the slower rate every 2500 documents.

First we tried a linear progression, setting the possible values of ϕ to be -30, -20, -10, 0, 10, 20 and 30. We ended up with the results shown in Figure 10.5. As you can see, the first try (Figure 10.5a) was a complete failure. In the second try (Figure 10.5b), the AI started extracting the events slightly late, but the results were good.



(a) First try. (b) Second try.

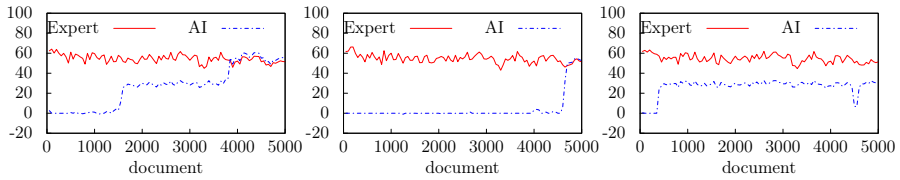
Figure 10.5 – Results for MAG with ϕ in -30, -20, -10, 0, 10, 20 and 30.

We then tried a linear increase of 100, which left us puzzled, as we expected it to be similar to the linear increase by 10. In Figure 10.6, you can see the results of putting ϕ in -300, -200, -100, 0, 100, 200 and 300. We tried several times, but either the algorithm was unlucky every time, or this range of ϕ with these parameters is not suitable for some reason.

We then decided to go to the other extreme, and increase the distance between the feedback steps by multiplying by 100 instead of by 10. In Figure 10.7, you can see the results of putting ϕ in -100 000, -1000, -10, 0, 10, 1000 and 100 000. We were tantalised to see that these results are not great either, the first time, but reasonable the second time.

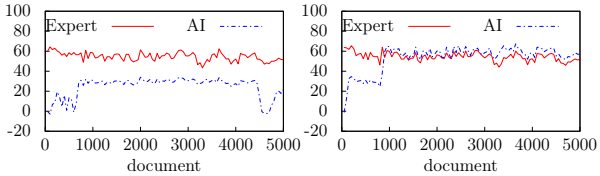
We wondered if these differences were because we had changed the code since the previous tests were run, so we re-ran the control with ϕ in -1000, -100, -10, 0, 10, 100 and 1000 (Figure 10.8). We were reassured to get similar results to those in Figure 10.1h on page 107.

We can draw two lessons from this little experiment. The first is that even the best AIs sometimes get bad luck, and for the results to be conclusive, we should have run them several times and taken the average. Unfortunately, each trial takes around 12-24 hours to complete, and we could not carry on the thesis indefinitely, so we had to stop. Secondly, you can see that the tuning of the magnitude for ϕ is not a completely trivial task after all, even if it is easier than determining an exact numerical value for a similarity. This renders the results of DOM, the purely qualitative solution even more attractive.



(a) First try. (b) Second try. (c) Third try.

Figure 10.6 – Results for MAG with ϕ in -300, -200, -100, 0, 100, 200 and 300.



(a) First try. (b) Second try.

Figure 10.7 – Results for MAG with ϕ in -100 000, -1000, -10, 0, 10, 1000 and 100 000.

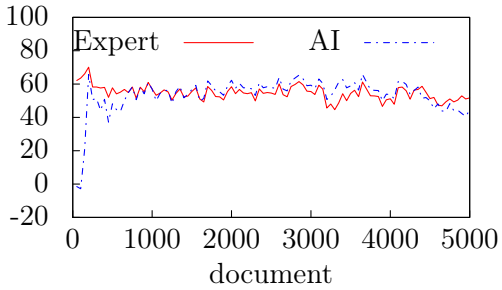


Figure 10.8 – Results for MAG with ϕ in -1000, -100, -10, 0, 10, 100 and 1000.

Chapter 11

Conclusion to Part I

In this, the main part of the manuscript, we detailed how we responded to the challenge laid down by our CIFRE contract. We modelled the IE treatment chain used to extract OSINT events from open-source documents as an MDP, and its improvement as a reinforcement learning problem.

We presented a proof-of-concept in which we used a modular framework, BIMBO, to test untrained AIs in a variety of situations given numerical rewards. We showed that the AIs are capable of constructing a chain step by step, that they generalise well on unknown documents, and that they learn even when given meagre source information, or sporadic feedback.

Encouraged by this success, we wanted to move away from the standard approach of rewarding the agent with quantitative, or numerical feedback. It is non-intuitive for the analyst, and cognitively difficult to define. We therefore investigated various qualitative feedback strategies, and we presented a novel algorithm, SSB Q-Learning (Gilbert et al., 2016).

We applied SSB Q-Learning to our treatment chain, and we showed that even with purely qualitative feedback, we still get excellent results.

In the following Part II, we describe two additional adventures which were born from the main part of the thesis. The first is Dora, a variation on the Q-Learning algorithm which aims to lose less information when exploring. The second is the application of Bimbo to an image analysis project.

Part II

Further contributions

Chapter 12

Dora - exploiting QL explorations

12.1 Introduction

In chapter 10, we said that we were using eligibility traces, but we did not really explain what they were. We go into more detail here, but still recommend that you read Sutton and Barto (1998, Chapter 7).

They explain that there are two ways to view a followed policy: the first is "a theoretical forward view", the second "a mechanistic backward view".

In their forward view, at time t, we stand at state s_t and see the policy followed stretching out into the future:

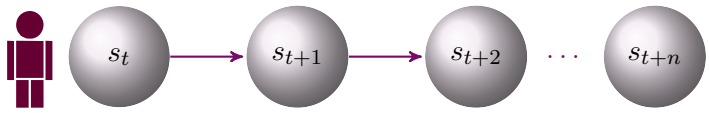


Figure 12.1 – Sutton and Barto (1998)'s forward (theoretical) view: looking forward in time to update the current state.

We update the current state based on what we can see in the state s_{t+1}, that is, at time $t+1$, which is updated based on s_{t+2} and so on.

Unfortunately, not many people possess a time machine to be able to go forward in time to see the value of state s_{t+n} in order to update state s_{t+n-1} and so on, all the way back to state s_t, so Sutton and Barto (1998) introduce a "mechanistic backward" view:

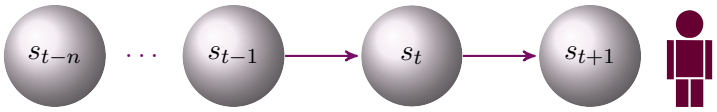


Figure 12.2 – Sutton and Barto (1998)'s backward (mechanistic) view: looking backward in time to update the previous states.

In this view, we've already arrived at state s_{t+1} by following the policy (taking optimal actions in each state). From here, we can see by how much our previous estimates for each state were out, and we can back-propagate the changes back through the states in the path to state s_{t-n}. For example, the expected value of each state might increase if the agent has just received a better reward than it thought it would.

In order to keep a record of the path the agent followed (s_{t-n}, ..., s_{t-1}, s_t, s_{t+1}), Sutton and Barto (1998) use an *eligibility trace*. Each time the agent visits a state, we add it to the trace.

For Q-Learning(λ) (hereafter referred to as *QL(λ)*) [Watkins (1989); Peng and Williams (1996); Baird (1995); Cichosz (1995); Sutton and Barto (1998); Wang et al. (2013); ...], the eligibility traces record the stack of state / action pairs enacted during a learning episode. In our case, a learning episode is the treatment of a single document for which the reward is only given at the very end.

To see why we would want to use eligibility traces, recall the Q-Learning algorithm (Algorithm 3):

Algorithm 3: Q-learning

Data: MDP $\mathcal{M}$

while *True* **do**

  Choose a_t in s_t using the EG exploration strategy

  Play a_t, observe s_{t+1}, and let $r_{t+1} = \mathcal{R}(s_{t+1})$

  $\hat{Q}_{t+1}(s_t, a_t) \leftarrow$

  $\hat{Q}_t(s_t, a_t) + \alpha_t(s_t, a_t)(r_{t+1} + \gamma \underset{b}{max}\{\hat{Q}_t(s_{t+1}, b)\} - \hat{Q}_t(s_t, a_t))$

With no eligibility traces, the potential value $\underset{b}{max}\{\hat{Q}_t(s_{t+1}, b)\}$ of state s_{t+1} is only back-propagated one time-step to the state / action pair (s_t, a_t). Given that in our case, the agent only gets a reward in the final states, and that it may try several hundreds of actions in one episode, it can take a very long time before the effect of these rewards is seen in the initial state. Using eligibility traces, on the other hand, enables the rewards observed to be back-propagated down the stack (or trace), thus speeding up learning by percolating the values back towards the initial state more quickly.

We assumed above that the agent follows the greedy policy (the one which gives the largest expected rewards), but we did not explain what happens if it does not. In fact, Sutton and Barto (1998) state that "we can use subsequent experience only as long the greedy policy is being followed". As soon as the

agent selects an exploratory, non-greedy (sub-optimal) action, the returns observed "no longer have any necessary relationship to the greedy policy".

In standard QL(λ), therefore, after an exploratory action, the eligibility trace is cut (reset to an empty stack), meaning that any good results found further on can take a long time to percolate back to the initial state (Dahl and Halck, 2001). In the worst case, where most of the actions are explorations, or where the reward is given only at the end of a (very) long path, the benefits of the eligibility traces can be practically eliminated (Figure 12.3).

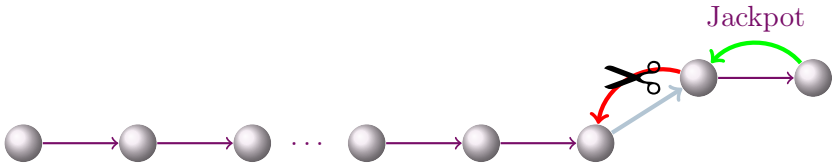
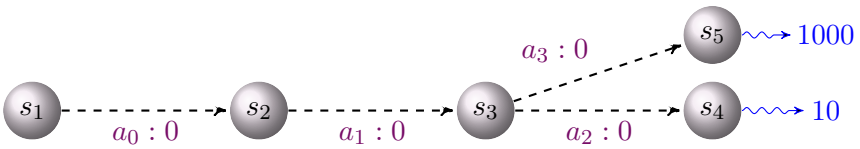


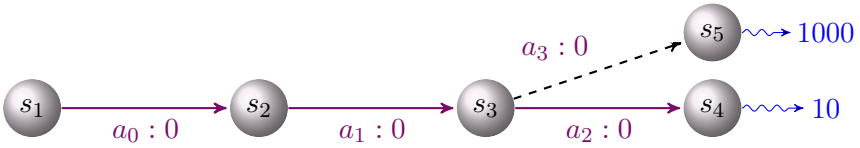
Figure 12.3 – A frustrating exploration gap in a very long chain, just before a large reward, where standard QL(λ) cuts the trace, preventing the back-propagation of the information.

Example 10 *Let us look in Figure 12.4 at a small example of how this trace cutting by QL(λ) affects the back-propagation (for simplicity, we focus only on the final back-propagations).*

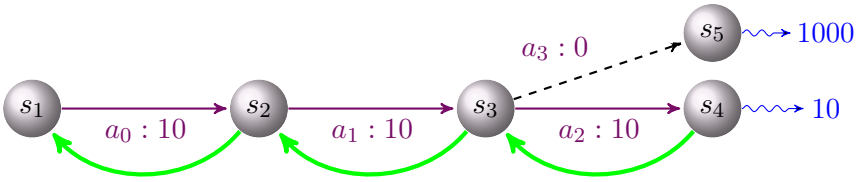
- *Figure 12.4a: We see that there are two rewarding states: s_4 which gives a small reward of 10, and s_5 which gives the jackpot of 1000.*
- *Figure 12.4b: From state s_1, the agent takes the optimal action a_0 (it exploits) to state s_2 [trace=$\{(s_1, a_0)\}$]. It then exploits a_1 to state s_3 [trace=$\{(s_1, a_0), (s_2, a_1)\}$], and finally exploits (as the two actions a_2 and a_3 apparently have equal value) action a_2 to state s_4, so now [trace=$\{(s_1, a_0), (s_2, a_1), (s_3, a_2)\}$].*
- *Figure 12.4c: It receives a reward of 10, which it can back-propagate down the trace to pairs (s_3, a_2), (s_2, a_1) and (s_1, a_0).*
- *Figure 12.4d: We start a new episode, and the agent as before exploits action a_0 to state s_2 [trace=$\{(s_1, a_0)\}$], and exploits a_1 to state s_3 which gives [trace=$\{(s_1, a_0), (s_2, a_1)\}$], but then it chooses to explore action a_3 to state s_5. This cuts the trace [trace=$\{\}$].*
- *Figure 12.4e: It receives the jackpot of 1000, which cannot now be back-propagated. The action a_0 in state s_1 is still apparently worth 10.*



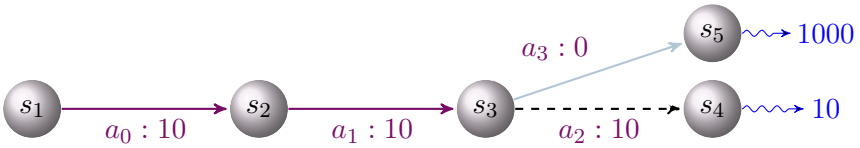
(a) The world - state s_5 rewards much more than state s_4



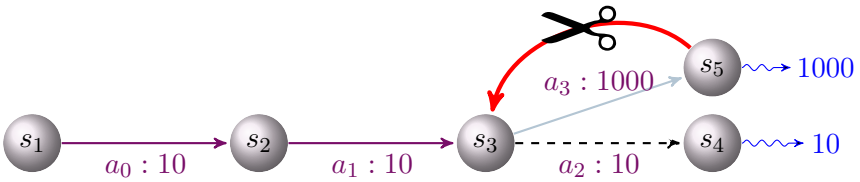
(b) From state s_1, the agent exploits action a_0, then from state $s2$ it exploits action a_1, and finally, from state s_3, it exploits action a_2.



(c) The agent receives reward of 10 and back-propagates it along the trace.



(d) A new episode: from state s_1, the agent exploits action a_0, then from state $s2$ exploits action a_1, but, in state s_3, it chooses the sub-optimal action a_3 (it explores) which cuts the trace.



(e) The agent receives the reward of 1000 but cannot back-propagate it.

Figure 12.4 – An example of how QL(λ) cuts the trace on an explore, preventing the back-propagation of useful learnt values.

12.2 Introducing Dora

Obviously these exploration gaps are sometimes inappropriate. With the benefit of hindsight, in Example 10, the action a_3 which hit the jackpot actually reflected the optimal policy better than the action a_2 which led to the smaller reward, and so cutting the trace was unnecessary.

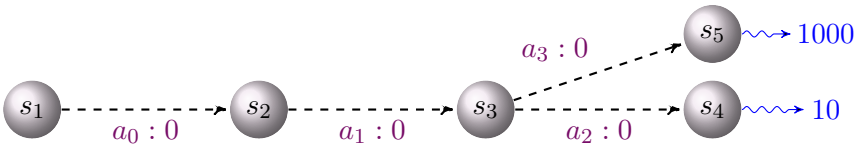
We wondered if we could use this hindsight to "exploit the explores", and we came up with Dora, an adaptation of QL(λ), which does just that.

The principle behind Dora is simple. Our aim is to avoid cutting the trace on an explore if possible by testing whether the new experience gained concerning that explore-action now makes it an exploit-action. If it does, then we treat it as if it had actually been an exploit (which in retrospect is the case), and we keep the trace intact enabling the propagation of results from further in the run back across this "explore" join.

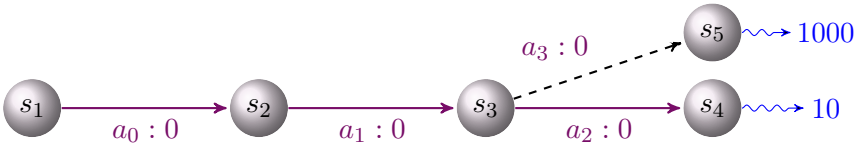
We will formalise this in section 12.3, but first:

Example 11 *Let us see in Figure 12.5 how Dora would change the situation given in Figure 12.4.*

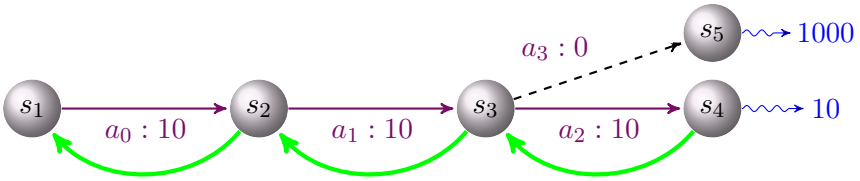
- *Figure 12.5a: As before, there are two rewarding states: s_4 which gives a small reward of 10, and s_5 which gives the jackpot of 1000.*
- *Figure 12.5b: As before, we build up the trace [trace={(s_1, a_0), (s_2, a_1), (s_3, a_2)}] by exploiting from states s_1, s_2 and s_3.*
- *Figure 12.5c: As before, the reward of 10 is back-propagated down the trace to pairs (s_3, a_2), (s_2, a_1) and (s_1, a_0).*
- *Figure 12.5d: We start a new episode. As before, Dora exploits action a_0 to state s_2 [trace={(s_1, a_0)}], and exploits a_1 to state s_3 which gives [trace={(s_1, a_0), (s_2, a_1)}], BUT when she chooses to explore action a_3 to state s_5, something different happens. Dora waits before clearing the trace to see what the outcome will be.*
- *Figure 12.5e: She receives the jackpot of 1000. This means that she now knows that action a_3 in state s_3 gives an expected value of 1000,* i.e. $\hat{Q}_{t+1}(s_3, a_3) = 1000$. *In other words, the action a_3 that she has just taken is now the best action in state s_3,* i.e. *the new value $\hat{Q}_{t+1}(s_3, a_3)$ of 1000 beats the old maximum value $\max_b\{\hat{Q}_t(s_3, b)\}$ of 10, so she treats it as if it had actually been an exploit. She keeps the trace intact and adds the new pair [trace={(s_1, a_0), (s_2, a_1), (s_3, a_3)}]. The jackpot can now be back-propagated all the way back to the state s_1.*



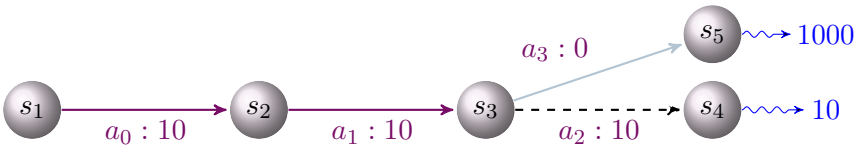
(a) The world - state s_5 rewards much more than state s_4



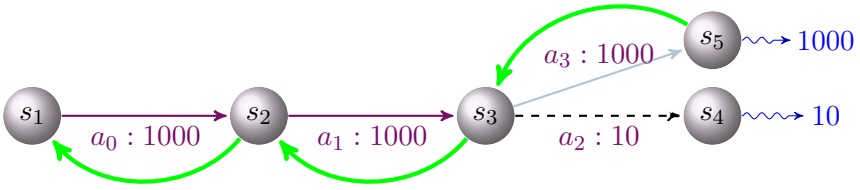
(b) From state s_1, Dora exploits action a_0, then from state $s2$ she exploits action a_1, and finally, from state s_3, she exploits action a_2.



(c) Dora receives reward of 10 and back-propagates it along the trace.



(d) A new episode: from state s_1, Dora exploits action a_0, then from state $s2$ exploits action a_1. In state s_3, she chooses the sub-optimal action a_3 (she explores), but waits before cutting the trace.



(e) Dora receives the reward of 1000. The best action in state s_3 is now a_3, which means that the trace can be kept intact, and the jackpot can now be back-propagated all the way to state s_1.

Figure 12.5 – An example of how Dora keeps the trace intact on an explore which turns out with hindsight to be a "best action", thus allowing the back-propagation of useful learnt values.

12.3 Formalisation

12.3.1 Q-Learning(λ)

Before we look at the mechanics behind Dora, let us examine QL(λ).

We can see in Algorithm 4 that as in Q-Learning with no eligibility traces (Algorithm 3), the agent chooses an action a_t in state s_t and observes in which state s_{t+1} it lands and what reward r_{t+1} it receives. However, instead of back-propagating just one time-step, as in Q-Learning with no eligibility traces:

$$\hat{Q}_{t+1}(s_t, a_t) \leftarrow \hat{Q}_t(s_t, a_t) + \alpha_t(s_t, a_t)(r_{t+1} + \gamma \max_b\{\hat{Q}_t(s_{t+1}, b)\} - \hat{Q}_t(s_t, a_t))$$

on Algorithm 4: line 4, we update the Q-Values using Algorithm 5.

Algorithm 4: Q-learning(λ)

Data: MDP $\mathcal{M}$
1 **while** *True* **do**
2 Choose a_t in s_t using the EG exploration strategy
3 Play a_t, observe s_{t+1}, and let $r_{t+1} = \mathcal{R}(s_{t+1})$
4 Update Q-Values using Algorithm 5

Algorithm 5: Q-Learning(λ): update Q-Values.

Data: Given: s_t, $\hat{Q}_t(s_t, a_t)$; Played: a_t ; Observed: s_{t+1}, r_{t+1}
1 $\delta_t \leftarrow r_{t+1} + \gamma \max_b\{\hat{Q}_t(s_{t+1}, b)\} - \hat{Q}_t(s_t, a_t)$
2 **if** a_t *was* $\operatorname{argmax}_b\{\hat{Q}_t(s_t, b)\}$ **then**
  `//a was an exploit, add to trace, back-propagate temporal difference`
3 *trace*.add(s_t, a_t)
4 QL-Back-propagate($trace, \delta_t$) using Algorithm 6
5 **else**
6 `//a was an explore, clear trace, update one time-step, add to trace`
7 *trace*.clear()
8 $\hat{Q}_{t+1}(s_t, a_t) \leftarrow \hat{Q}_t(s_t, a_t) + \alpha_t(s_t, a_t)\delta_t$
9 *trace*.add(s_t, a_t)

Algorithm 6: Function QL-Back-propagate($trace, \delta_t$)

Data: $trace, \delta_t$

1 $exponent \leftarrow 0$

2 **for** (s_i, a_i) *in reverse(trace)* **do**

3 $\quad \hat{Q}_{t+1}(s_i, a_i) \leftarrow \hat{Q}_t(s_i, a_i) + \lambda^{exponent} \alpha_t(s_i, a_i)\delta_t$

4 $\quad exponent++$

Let us look at Algorithm 5 in more depth:

- On Algorithm 5: line 1, we calculate the *temporal difference* δ_t to be

$$r_{t+1} + \gamma \underset{b}{max}\{\hat{Q}_t(s_{t+1}, b)\} - \hat{Q}_t(s_t, a_t)$$

  This temporal difference is the change over one time-step in the predicted value of $\hat{Q}(s_t, a_t)$, and should look familiar. If we multiply it by the learning rate $\alpha_t(s_t, a_t)$, and add $\hat{Q}_t(s_t, a_t)$, we get $\hat{Q}_{t+1}(s_t, a_t)$:

$$\hat{Q}_{t+1}(s_t, a_t) \leftarrow \hat{Q}_t(s_t, a_t) + \alpha_t(s_t, a_t)\delta_t$$

- On Algorithm 5: line 2, we test whether or not the action a_t was an optimal action in state s_t, that is, whether it was an exploit or an explore.

- If action a_t was an exploit, then we add the state / action pair (s_t, a_t) to the trace, and back-propagate the temporal difference down through the trace (we will look at the back-propagation below).

- If action a_t was an explore, then we clear the trace, and do the one time-step update that we see in the standard Q-Learning without eligibility traces. We then start the trace again by adding the state / action pair (s_t, a_t), as if we follow the policy from this point, we can safely back-propagate to here.

Let us look at the back-propagation Algorithm 6 now:

- The trace is treated in reverse order so that the most recently visited states are updated first. This is because a *decay factor* λ (which is why the algorithm is called QL(λ)) is used to make sure that the states closest to state s_t in the trace are more affected by the change than those further away. Sutton and Barto (1998) liken this to shouting the

results back down the trace – the closer you are, the louder and clearer you hear the message, but as you get further away, the message gets quieter. This is because the actions are stochastic; the further back you go in the trace, the less likely you would be to come back to exactly the same state, even by following the policy.

- λ is decayed using the *exponent*, which is initialised to 0, and augmented by 1 for each state / action pair in the trace.

12.3.2 Dora

As we did for QL(λ), we will show you the algorithms behind Dora, and walk you through them step by step.

Algorithm 7: Dora

Data: MDP $\mathcal{M}$
while *True* **do**
    Choose a_t in s_t using the EG exploration strategy
    Play a_t, observe s_{t+1}, and let $r_{t+1} = \mathcal{R}(s_{t+1})$
    Update Q-Values using Algorithm 8

As you can see, Algorithm 7 and Algorithm 4 are identical, and the only change in Algorithm 8 is in the addition of line 7–line 9, which we explain here.

We saw above that

$$\hat{Q}_{t+1}(s_t, a_t) \leftarrow \hat{Q}_t(s_t, a_t) + \alpha_t(s_t, a_t)\delta_t$$

We can test this new Q-value for (s_t, a_t) against the previous maximum Q-value for state s_t taken over all possible actions. If it is still less than this maximum, then the action was and remains an explore.

If, on the other hand, it is now greater than or equal to the previous maximum, it means that action a_t becomes an optimal, or exploit action in state s_t. We calculate the temporal difference Δ_t where:

$$\Delta_t \leftarrow \hat{Q}_t(s_t, a_t) + \alpha_t(s_t, a_t)\delta_t - \max_b\{\hat{Q}_t(s_t, b)\}$$

and back-propagate it down the trace.

Algorithm 8: Dora: update Q-Values

Data: Given: s_t, $\hat{Q}_t(s_t, a_t)$; Played: a_t ; Observed: s_{t+1}, r_{t+1}

$\delta_t \leftarrow r_{t+1} + \gamma \underset{b}{max}\{Q_t(s_{t+1}, b)\} - \hat{Q}_t(s_t, a_t)$

if a_t *was* $\underset{b}{argmax}\{\hat{Q}_t(s_t, b)\}$ **then**

`//a was an exploit, treat as in QL(lambda)`

$trace$.add(s_t, a_t)

Dora-Back-propagate$(trace, \delta_t, 0)$ using Algorithm 9

else

if $\overbrace{\hat{Q}_t(s_t, a_t) + \alpha_t(s_t, a_t)\delta_t}^{\hat{Q}_{t+1}} \geqslant \underset{b}{max}\{\hat{Q}_t(s_t, b)\}$ **then**

`//a is now the new best action in s_t`

$\Delta_t \leftarrow \hat{Q}_t(s_t, a_t) + \alpha_t(s_t, a_t)\delta_t - \underset{b}{max}\{\hat{Q}_t(s_t, b)\}$

Dora-Back-propagate$(trace, \Delta_t, 1)$ using Algorithm 9

$\hat{Q}_{t+1}(s, a) \leftarrow \hat{Q}_t(s_t, a_t) + \alpha_t(s_t, a_t)\delta_t$

$trace$.add(s_t, a_t)

else

`//a was an explore and is still worse than the best action, treat as in QL(lambda)`

$trace$.clear()

$\hat{Q}_{t+1}(s_t, a_t) \leftarrow \hat{Q}_t(s_t, a_t) + \alpha_t(s_t, a_t)\delta_t$

$trace$.add(s_t, a_t)

Algorithm 9: Function Dora-Back-propagate$(trace, \delta, exponent_0)$

Data: $trace, \delta_t, exponent_0$

$exponent \leftarrow exponent_0$

for (s_i, a_i) *in reverse(trace)* **do**

$\hat{Q}_{t+1}(s_i, a_i) \leftarrow \hat{Q}_t(s_i, a_i) + \lambda^{exponent}\alpha_t(s_i, a_i)\delta_t$

$exponent++$

There are two important things to note here. The first is that we have not yet added the state / action pair (s_t, a_t) to the trace, and the second is that we back-propagate with the *exponent* of λ initialised to 1 instead of 0. This is because, in examining state s_{t+1} (the result of taking action a_t in state s_t), we have actually used a tiny time-machine to jump two time-

steps with respect to the last element in the trace, the state / action pair (s_{t-1}, a_{t-1}).

We need to imagine that we can go back in time to when we took action a_t in state s_t. If we had known then what we know now, we would have added (s_t, a_t) to the trace and back-propagated the temporal difference Δ_t down the trace. This means that (s_{t-1}, a_{t-1}) would have been the second pair in the trace, and the corresponding *exponent* would have been 1.

However, the current Q-value $\hat{Q}_t(s_t, a_t)$ for the pair (s_t, a_t) is *not* based on our new knowledge, so we cannot use the temporal difference Δ_t to update $\hat{Q}_t(s_t, a_t)$. Instead, it needs to be updated with respect to the temporal difference δ_t which really applies at time t. We therefore perform a one-step back-propagation using the now familiar update

$$\hat{Q}_{t+1}(s_t, a_t) \leftarrow \hat{Q}_t(s_t, a_t) + \alpha_t(s_t, a_t)\delta_t$$

and only add (s_t, a_t) to the trace once we have safely parked the time-machine.

You can see that in this way, if a_t becomes an optimal action in s_t, we gain two advantages. Firstly, we back-propagate the temporal differences δ_t and Δ_t. Secondly, we keep the trace intact, enabling the propagation of results from further in the run back across this "*explore*" join. This is in sharp contrast with QL(λ), which does neither.

12.4 Experiments

We tested Dora against QL(λ) taking the average results over at least 50 runs each time. As the paths in the treatment are potentially very long, the discount value γ and the decay rate λ were both set to 0.9. This ensures that a significant proportion of the temporal difference is back-propagated further down the trace than it would have been with a lower value of λ, and that the agent seeks long-term gain rather than immediate rewards. The learning rate α was set to 0.2. To avoid wasting time calculating negligible changes, and the risk of memory explosion, we also limited the trace to a maximum length of 100 pairs.

We tested both with a constant ϵ of 0.2, and a decreasing $\epsilon(s)$ as in section 8.4, *i.e.*

$$\epsilon(s) = \frac{1}{(1 + n(s))^\beta}$$

where $\epsilon(s)$ is the epsilon value to use in state s, $n(s)$ is the number of visits to state s so far, and β is 0.55.

We measured the quality of their learning by recording the evolution of the distance of their current *value function* (which measures the best possible value that the agent could achieve from a given state) V^t from the optimal V^* at each time-step (where s_0 is the starting state):

$$\text{episodic-distance}(V^*, V^t) = |V^*(s_0) - V^t(s_0)|$$

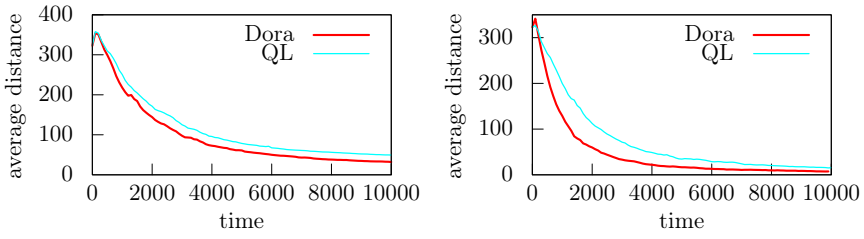
Note, we also measured the L_2 and infinite distances, but have not shown them as they gave the same curves:

$$\text{L}_2\text{-distance}(V^*, V^t) = (\sum_s (V^*(s) - V^t(s))^2)^{1/2}$$
$$\text{infinite-distance}(V^*, V^t) = \max_s \{|V^*(s) - V^t(s)|\}$$

12.4.1 Random MDP tests

We first tested the comparative performance of Dora and QL(λ) on a series of randomly generated MDPs from 10 to 100 states, and 5 to 10 actions. Each state / action pair (s, a) had a randomly chosen, unrestricted number of next states, and were allocated integer rewards generated randomly between 0 and 100.

Figure 12.6a and Figure 12.6b are representative of the results we obtained. We observed that Dora consistently learnt faster than QL(λ), and the larger the MDP, the more significant her advantage.



(a) constant $\epsilon = 0.2$ (b) decreasing ϵ

Figure 12.6 – Dora and QL(λ) were run on the same randomly generated MDP of 80 states and 6 actions. Results were measured using the episodic distance averaged over 50 runs.

12.4.2 Cliff-walk MDP tests

Intuitively, we thought that Dora should perform even better on "long thin" or "cliff-walk" MDPs with long trajectories, and lots of exploration potentially necessary to reach the goal. We constructed such an MDP (see Figure 12.7), where every action except one leads back to the initial state with probability 0.5, or stays on the same state with probability 0.5, obtaining a zero reward. Only one action advances the agent towards the goal with probability 1.0. The reward given for reaching the final state was 100 times the number of states. We ran tests on a series of these MDPs from 5 to 15 states and 2 to 10 actions.

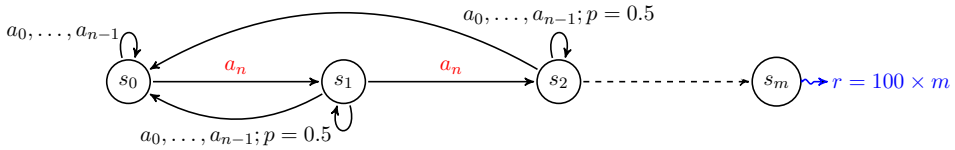
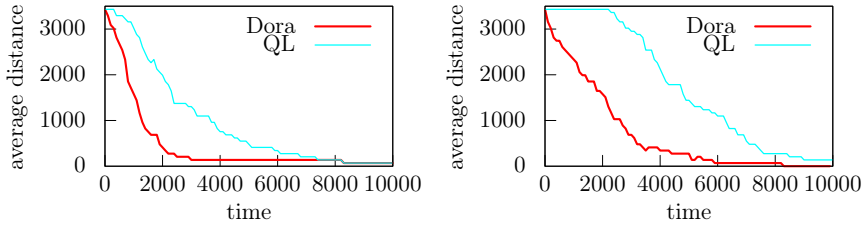


Figure 12.7 – "Long thin" MDP (n actions, m states) with $p(s_i, a_1 \ldots a_{n-1}, s_0) = 0.5$ and $p(s_i, a_1 \ldots a_{n-1}, s_i) = 0.5$ (zero reward); $p(s_i, a_n, s_{i+1}) = 1.0$. State s_m is the only one to offer a reward of $100 \times m$.

We found that on this particular MDP, Dora consistently significantly outperforms QL(λ) (Figure 12.8a), especially with a decreasing ϵ (Figure 12.8b).



(a) constant $\epsilon = 0.2$ (b) decreasing ϵ

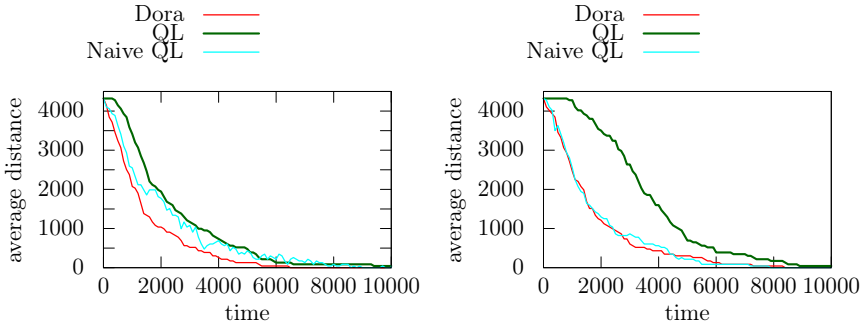
Figure 12.8 – Dora and QL(λ) were run on the same Long Thin MDP (Figure 12.7) of 15 states and 2 actions. Results were measured using the episodic distance averaged over 50 runs.

12.4.3 Naive QL(λ)

We thought that an interesting baseline for assessing the efficiency of Dora could be a naive version of QL(λ) with no trace-clearing at all as mentioned at the end of Sutton and Barto (1998, Chapter 7, Section 7.6):

> One could imagine yet a third version of QL(λ), let us call it naive QL(λ), that is just like Watkins's QL(λ) except that the traces are not set to zero on exploratory actions ... We know of no experience with this method, but perhaps it is not as naive as one might at first suppose.

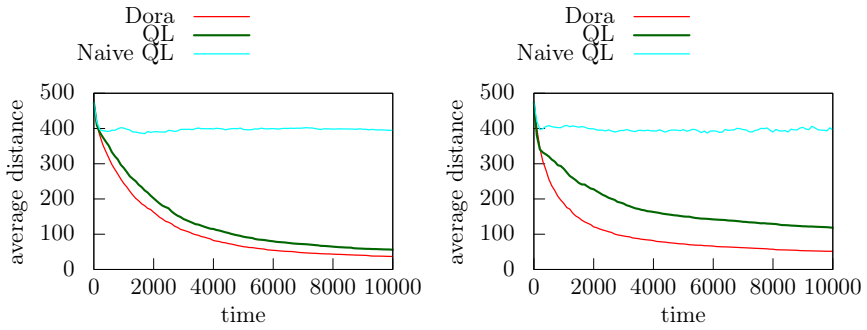
We ran some preliminary experiments which suggest that on long thin MDPs, especially with a decreasing ϵ, there is little difference between Dora and Naive QL(λ) (see Figure 12.9).



(a) constant $\epsilon = 0.2$ (b) decreasing ϵ

Figure 12.9 – Dora, QL(λ) and Naive QL(λ) were run on the same Long Thin MDP (see Figure 12.7) of 15 states and 2 actions. Results were measured using the episodic distance averaged over 100 runs.

On generic MDPs, however, Naive QL(λ) gives much worse results than both QL(λ) and Dora (see Figure 12.10). In fact, according to Wyatt (1998) Naive QL(λ) is not guaranteed to converge.



(a) constant $\epsilon = 0.2$ (b) decreasing ϵ

Figure 12.10 – Dora, QL(λ) and Naive QL(λ) were run on the same randomly generated MDP of 80 states and 6 actions. Results were measured using the episodic distance averaged over 100 runs.

12.5 Experience replay

The idea behind Dora is quite simple and natural, but to the best of our knowledge, it has not been developed like this before. Initially, however, we wondered if the principle of Dora could be argued to resemble that of *experience replay* (Lin, 1991) which Schaul et al. (2015) define as follows:

> Experience replay lets online reinforcement learning agents remember and reuse experiences from the past.

Lin (1991) was motivated to develop experience replay by two perceived inefficiencies in the Q-Learning proposed by Watkins (1989) in his thesis. The first was the waste of the agent's experiences, which are only used once to update the Q-value. Lin (1991) argues that some of these experiences are costly to acquire, or rare, and should be used more effectively. The second inefficiency was the order in which the experiences are presented to the agent. As the agent is exploring the world, it experiences the effects of its actions in chronological order. According to Lin (1991), this is less effective than presenting the experiences in reverse-chronological order.

In standard experience replay, the experiences $(s_i, a_i, s_{i+1}, r_{i+1})$ of the agent are collected and stored for an entire lesson:

$$\{(s_0, a_0, s_1, r_1), (s_1, a_1, s_2, r_2), (s_2, a_2, s_3, r_3) \dots (s_n, a_n, s_{n+1}, r_{n+1})\}$$

where s_0 is the initial state, and s_n is the final state. Each lesson is chosen either randomly or prioritized (see below) and replayed at the end of one or

more episodes. This enables the agent to learn and self-improve even when a teacher is not available.

In fact, there are several important differences between Dora and experience replay. Firstly, Dora is not model-based, she has fewer parameters to tune (as we're not testing for replays), and consumes less memory (as no storage is required for the lessons), whilst still giving excellent results.

Of course, with experience replay, there is also a danger of introducing statistical bias if the lessons to be replayed are not representative, which is not a problem with Dora. Several methods have been suggested to overcome this weakness by prioritising the lessons selected, for example Schaul et al. (2015), Sander et al. (2012), Moore and Atkeson (1993) and Peng and Williams (1993). For a more complete overview, see Sigaud and Buffet (2010, Chapter 2).

Also, Dora exclusively uses online learning where the updates occur after each state visit, which is more efficient and faster than "offline" learning (Wiering and Schmidhuber, 1998), which experience replay relies on heavily, where the updates occur only after the end of a trial.

12.6 Related and future work

We are obviously not the first to propose improvements to QL(λ), and we give an overview of some of the techniques that we found most interesting below. In many cases, Dora could be combined with them simply by adding the check in Algorithm 8, line 7–line 9 and modifying the back-propagation to take into account the *exponent* parameter.

Wyatt (1998) gives us some possible ideas for improving the performance of Dora. He compares the effect of various values of α and λ on replacing and accumulating traces. Sutton and Barto (1998) note that if a state is already in the trace, and it is revisited, then performance can be improved by resetting the decay of the trace (replacing), rather than allowing the decay to increase (accumulating). To explore the sensitivity of the trace to exploration, Wyatt (1998) carries out two types of test. During the first, explores are left in the trace (which he calls uncorrected). In the second, he resets the trace on exploration (which he calls corrected). To use his terminology, we therefore used accumulating corrected traces for QL(λ) and Dora, and accumulating uncorrected traces for Naive QL(λ). He shows that accumulating and reducing traces have similar performance, except at $\lambda = 1$ when accumulating becomes unstable and replacing continues to improve. He also finds that the corrected trace performs better than the uncorrected

trace, and that the uncorrected trace does not converge, which confirms what we saw in Figure 12.10.

Another interesting idea for improving QL comes from Strehl et al. (2006) who introduce Delayed QL, proving that it is PAC (Probably Approximately Correct) and comparing it with R-Max. Their algorithm uses "learn" flags to determine if an update to the pair (s, a) is allowed. After m sample updates of (s, a), if the "learn" flag is true, an update is attempted. This minimises the effect of randomness. If this update fails, then the flag is set to false, and can only be reset to true if another Q-value estimate is updated.

Wiering and Schmidhuber (1998) also try to reduce the number of updates with their algorithm. Fast QL is online, and uses "lazy learning" to only update when needed. They keep track of the changes in a global variable, and then make a local update to the Q-value before an action is selected. After the execution of an action, they perform a global update: update $V(s_{t+1})$ by calling a local update on the next possible state / action pairs, update global variables, then update the current Q-value and trace value and replace the current local value by its global value. Their main advantage over the standard QL(λ) is the reduction in CPU time (but it still takes twice as long as QL with no traces). The learning rate is unchanged.

Like Dahl and Halck (2001), we recognise that exploration can distort the learning. They refer in their Section 2 to the fact that the exploration in standard QL(λ) breaks the trace, and therefore the propagation of feedback to earlier states is delayed. They estimate the cost of not following the greedy policy, *i.e.* the difference between the exploratory action and the ideal action, and then give these estimates as rewards to compensate the agent for following what it perceives as inferior actions. Because this led to instability, they limit this reward to a maximum value.

Van Hasselt (2010) postulates that QL(λ) is over-optimistic, and to mitigate this, proposes the Double Q-Learning algorithm. Double QL maintains two sets of Q-Values based on the same problem but different experience samples. It then uses one to update the other, and both to choose the next action. He shows (on roulette and grid world experiments) that Double QL converges faster than QL. In Van Hasselt et al. (2015), they apply Double QL to deep reinforcement learning, giving the results over several Atari games. This gives us another potential avenue to explore - the application of Dora to deep RL, or machine learning. In chapter 13, we use Dora to parametrise two image treatment services, and we would like to expand this to parametrising a deep learning neural network.

In this case, we might imagine combining Dora and experience replay. As Tambet (2016) points out, the Q-function can be approximated using a con-

volutional neural network, but an approximation of Q-values using non-linear functions is unstable. He increases the stability when training the network, by randomly giving stored experiences instead of the most recent transition. This helps to keep the network from getting distracted by similarities in successive training samples. Mnih et al. (2013) also use experience replay with a neural network, but they select an experience to replay randomly from the stored transition set at every step, rather than sporadically.

It would be interesting to run experiments in a wider variety of settings, for example, with rewards which reduce the optimism rather than increase it, and with several optimal paths in a "long thin" MDP. We might also imagine testing a case similar to the long thin MDP in Figure 12.7, except that the non-advancing actions a_1 to a_{n-1} actually result in a small reward. This could happen, for example if we want the AI to concentrate on extracting "interesting" events as in section 8.2, when we offer a smaller reward for any extraction deemed not "interesting". In this case, as Moore and Atkeson (1993) found, insufficient exploration would be dangerous as the small reward would tempt the agent away from the longer path to the large reward.

We also conjecture that there are families of "long thin" MDPs for which we can *prove* that Dora learns exponentially faster than QL(λ) or other algorithms. Apart from the improved algorithm, we believe that such formal results would help gain insight into the interplay between exploration and back-propagation.

12.7 Summary

We presented Dora, a variation on Q-Learning(λ), which exploits explores whenever they turn out to be the best action with hindsight. This allows us not only to back-propagate the temporal difference δ_t at time t, but also to keep the trace intact, enabling the propagation of results from further in the run back across this "explore" join. Dora therefore learns consistently faster than the standard QL(λ), which does neither.

We gave test results based on two types of MDP – randomly constructed and cliff-walk, and showed that Dora consistently outperforms QL(λ).

We also tried Naive QL(λ), and found that in a cliff-walk MDP it gives similar results to Dora, but that in a random MDP it failed to converge.

Finally, we discussed other methods, including experience replay and how it differs from Dora, and possible future work.

Chapter 13

Application to image analysis

13.1 Motivation

One of the business areas of *Cordon Electronics DS2i* is the maintenance of hardware and systems. Some of these date back decades, and their documentation has been scanned as pdfs and archived. These pdfs cover everything from the technical details of a power supply to the specification of a motherboard, to templates for other documents (see Figure 13.1). Unfortunately, when the documents were scanned, they were only given a number with no indication of their contents. Given that there can be several thousands of documents for one project, searching for the dimensions of a particular obsolete component, for example, can be a long and painstaking task.

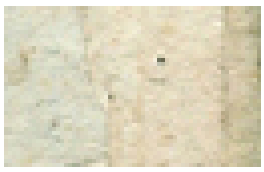
Figure 13.1 – An example of a scanned front page, in which the title needs to be detected. Reproduced with the kind permission of Valentin Laforge.

13.2 Title detection

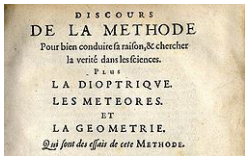
Valentin Laforge, an internship student from the University of Caen, Normandy, specialising in signal-treatment, was given the objective of automatically detecting the titles of the documents through image analysis. The premise was that any suggestion of a title, even if it was not completely correct was better than nothing, as it would aid in the indexing of the document, and in the search for the contents.

He created two web services, an image pretreatment service and an optical character recognition (OCR) service to identify the most likely titles for a document and present them to the user. We give an overview of the techniques that he implemented below, but invite you to read his internship report *"Vision par Ordinateur: Segmentation, Extraction et Reconnaissance Visuelle d'Élements dans des Contextes Variants"* ("Computer Vision: Segmentation, Extraction and Visual Recognition of Elements in Various Contexts"), available from the University of Caen, France.

First, the pretreatment service cleaned the image to remove noise (see Figure 13.2). The method used was *non-local means denoising* (Buades et al., 2011). This differs from *local means denoising* which considers each pixel one by one, and takes the mean colour of the pixels in close proximity. The problem is that the closest pixels along the side of a character on a page, for example, are not at all homogeneous, and their average only adds more noise. In *non-local means denoising*, the mean is taken over larger areas, which leads to a cleaner, more uniform effect than the localised methods.



(a) An example of noise in a document



(b) The document before treatment

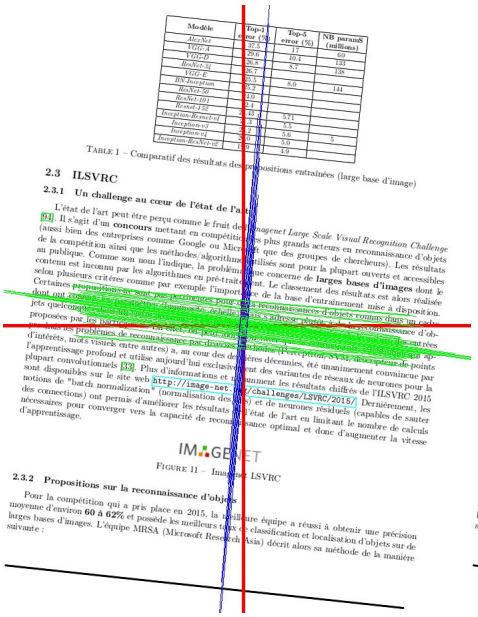
(c) The document with noise removed

(d) A character before treatment

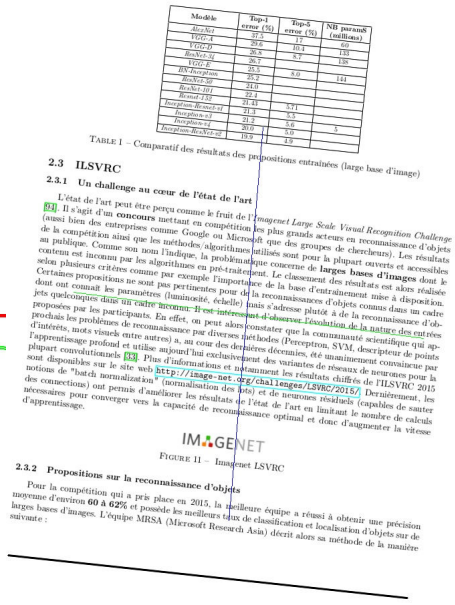
(e) The character with noise removed

Figure 13.2 – Documents contained noise which could hinder the text recognition software, and so had to be cleaned. Images reproduced with the kind permission of Valentin Laforge.

Second, as the documents had been scanned by hand, many were slightly lopsided (see Figure 13.3), and so it was necessary to reorient them. The pre-treatment service calculated the dominant lines in the document, and then took the average of their perpendiculars to establish the degree of rotation and automatically straighten them. For the most part, the degree of rotation was quite low (between 5 and 20°), but sometimes the document was upside down and therefore impossible to rectify correctly.



(a) The dominant lines in the document were established.



(b) The average of the perpendiculars gave the degree of rotation.

(c) If the document was upside down, it was impossible to rectify.

Figure 13.3 – The document was scanned at a slant and had to be reoriented. Images reproduced with the kind permission of Valentin Laforge.

To finish the image pre-treatment, the image was dilated and eroded. Dilation is a little like blowing up a balloon. It removes the fine details, for instance the hole in the letter "p". Erosion is like peeling a potato – it pares away the edges to remove extrusions. The combination of the dilation and erosion removed the details of the text, and a binary mask was then applied, which enabled the detection of rectangular regions (see Figure 13.4).

(a) Original image.

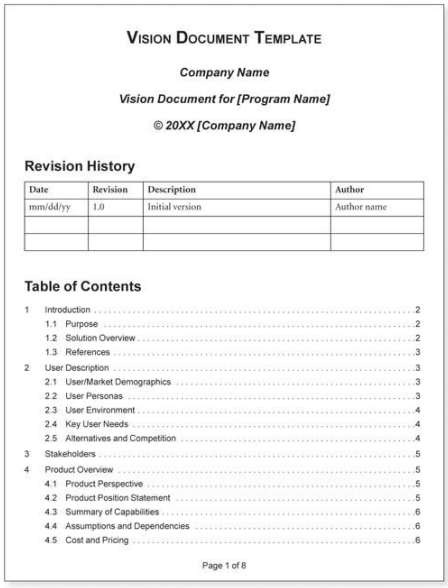
(b) The image was eroded and dilated, and a binary mask was constructed.

(c) Delineating rectangles were identified.

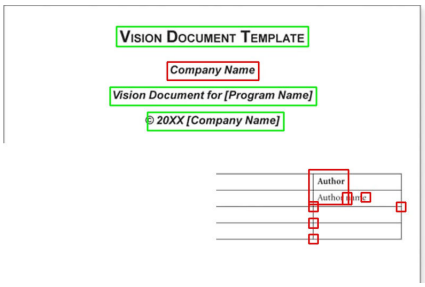
Figure 13.4 – The document was eroded and dilated until a binary mask showed rectangular regions. Images reproduced with the kind permission of Valentin Laforge.

The second service applied OCR to decode the text in those regions. Certain characteristics tend to define a title. For instance, they might be in a larger font than the rest of the text, bold, italic, or may be outlined in some way. In order to detect and recognise an object in an input image, a fixed size window is used to examine the pixels group by group, and it may be necessary to scale the given input image so that the object fits into the window. A "pyramid" of images in various sizes is therefore created and the window applied to each.

Finally, for each document, the OCR service suggested several titles, with a confidence rating for each (see Figure 13.5).



(a) An example of the front page of a scanned document, cleaned and reoriented, in which the title needs to be detected.



(b) Candidates for the title have been detected. The most likely are outlined in green, the least likely in red.

Figure 13.5 – Detection of titles by the OCR service. Images reproduced with the kind permission of Valentin Laforge.

13.3 Application of BIMBO

The parameters for the image pre-treatment and OCR were established by expertise and observation of test results, and were set up to cater for the widest possible range of cases. These parameters, however, were insufficient for certain specific cases, for example, the document shown in Figure 13.3b, where the title contains a logo in place of the "A". We therefore decided to apply BIMBO to continuously improve the title suggestions by establishing the best parameters to use for each image on a case by case basis. Guillaume Leroy, internship student from the University of Rouen, Normandy, integrated BIMBO with the two web-services.

Recall the modules of BIMBO (Figure 7.1 on page 52). In Figure 13.6 we show how we adapted her to image detection with only a few changes. Instead of constructing a treatment chain for a document, the AI now has to decide on a parameter configuration. Only when it chooses *STOP* is the

document sent to the image processing services. The "Dynamic Router" has therefore been replaced by a simple definition of the end-point of the first of the services (from which it is passed automatically to the second). The "State / Action Conversion" has obviously changed too (the possible states and actions are detailed in section 13.4). The "Similarity Calculation" has also changed, as detailed in section 13.5. The "Controller" is still responsible for lining up the documents, and handling the stopwatch. The "Reinforcement Learning Algorithms" module remains the same (although we only used Dora), and we still use the "Reporting" module to produce logs *etc.*.

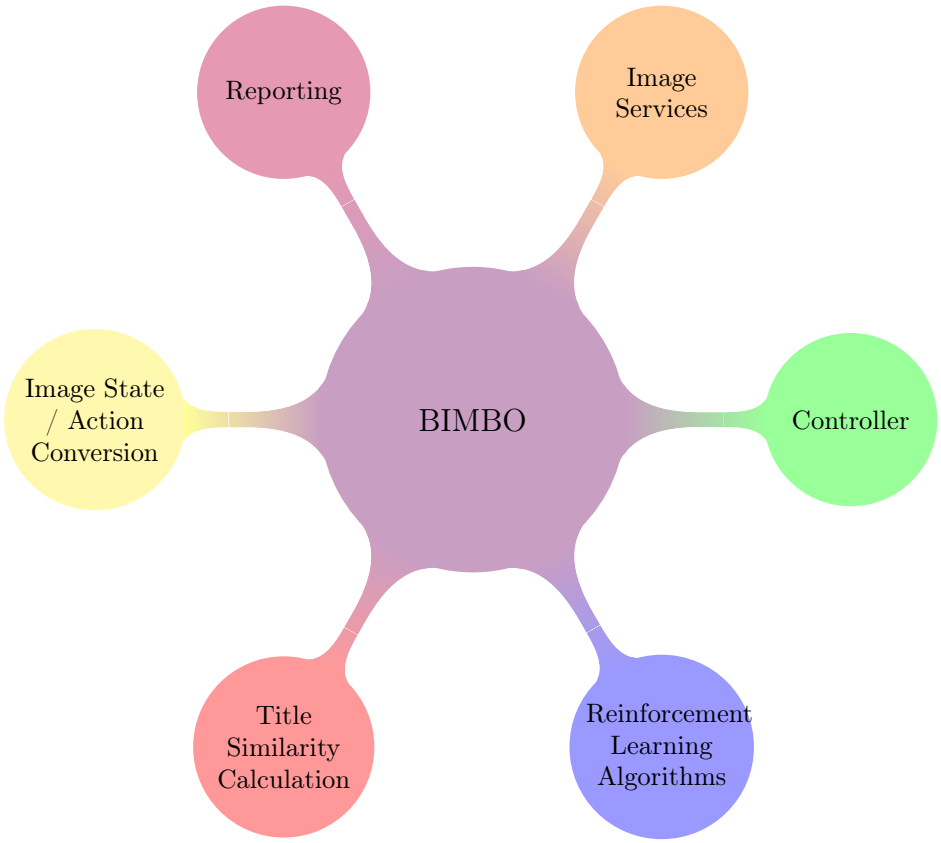


Figure 13.6 – The modules of BIMBO applied to the detection of objects in an image.

13.4 Choice of states and actions

We gave Dora a range of values to chose from for each parameter (Figure 13.7), initialising them by default with the parameters as determined by the expert (Figure 13.8).

Each choice of value for each parameter was one action, and the combinations of values for each of these parameters defined the states:

- The first part of the state were the document characteristics:
  - a calculated indicator of the page's complexity, for example, whether the page was typeset, or contains an image such as a company logo;
  - image dimensions - width and height, which are linked and depend on whether the document is portrait or landscape;
  - detected language of the document.
- The second part of the state depends on the variable parameters used by the image pretreatment service and changeable by the AI:
  - the erosion and dilation parameters (which control the amount of detail that is added or removed as in Figure 13.4);
- The third part of the state depends on the variable parameters used by the OCR service and changeable by the AI:
  - the reduction factors - portrait, landscape, width and height (used to create the pyramid of images);
  - the number of rectangular selections to analyse;
  - the kernel parameters (the kernel function is used in the classification of the object detected in the image).

This gives 1 944 000 states and 43 actions in total. Note that this is huge compared to the experiments that we carried out in Part I, and reassures us that the performance of our approach does not seem to be affected by a large state-action space.

```
<bean id="bimboStatesConfig" class="config.ConfigStates">
  <!-- The document characteristics -->
  <property name="lowIndicatorLimits" value="clean, average, complex" />
  <property name="imageWidth" value ="500, 1000" />
  <property name="imageHeight" value = "500, 1000" />
  <property name="lang" value="eng" />
  <!-- The variable parameters for the pretreatement service -->
  <property name="morphErodeN" value="1, 2, 3, 4" />
  <property name="morphDilateN" value="4, 6, 8, 9, 10" />
  <property name="morphErodeType" value="MORPH_ELLIPSE" />
  <property name="morphDilateType" value="MORPH_RECT" />
  <property name="kernelSize" value="0, 3, 5" />
  <property name="widthKernelDilatationRatio" value="1.0, 1.5, 2.0" />
  <property name="heightKernelErosionRatio" value="1.0, 1.33, 1.35, 1.75" />
  <!-- The variable parameters for the OCR service -->
  <property name="portraitReduction" value="1, 2, 3" />
  <property name="landscapeReduction" value="1, 2, 3" />
  <property name="widthReductionFactor" value="1.5, 2.0, 2.5" />
  <property name="heightReductionFactor" value="1.5, 2.0, 2.5" />
  <property name="reductionFactor" value="45" />
  <property name="numberOfSelections" value="3, 6" />
  <property name="arbitraryZoomforKernelSize" value="150000" />
</bean>
```

Figure 13.7 – The state offered to the AI, consisting of the document characteristics and the variable parameters with their possible values.

```
<!-- The variable parameters for the pretreatement service -->
"morphErodeN" = "2";
"morphDilateN" = "9";
"morphErodeType" = "MORPH_ELLIPSE";
"morphDilateType" = "MORPH_RECT";
"kernelSize" = "0"
"widthKernelDilatationRatio" = "1.5";
"heightKernelErosionRatio" = "1.33";
<!-- The variable parameters for the OCR service -->
"portraitReduction" = "2";
"landscapeReduction" = "3";
"widthReductionFactor" = "1.5";
"heightReductionFactor" = "2.5";
"reductionFactor" = "45";
"numberOfSelections" = "6";
"arbitraryZoomForKernelSize" = "150000";
```

Figure 13.8 – The default parameters as determined by the expert.

13.5 Similarity measure

Valentin constructed a corpus of 42 documents, each of which was associated with the title that a user would expect to see in an oracle.

An individual score $IS(title)$ was calculated for each suggested title based on the Fuzzy Levenshtein similarity between it and the oracle $FLS(title, oracle)$ combined with the OCR's confidence $C(title)$ in its detection:

$$IS(title) = \frac{a * FLS(title, oracle) + b * C(title)}{a + b}$$

In order to take the confidence of the OCR into account whilst still relying mostly on the similarity, a was set to 75, and b to 15.

13.6 Tests

First, we ran the corpus of 42 documents with no AI, and obtained a baseline expert score of 0.81, which is shown as the red line in the following graphs.

To calculate the reward, we took the average of the individual scores for all the suggested titles. This gave us an overall score for the document. The AI was rewarded with this overall score as feedback on the parameters chosen for that document treatment.

We ran three tests using Dora, each with an exploration rate ϵ starting at 0.2, and reduced every 84 documents (twice through the corpus):

- During the first tests, this division was by 2.0, and we noted (Figure 13.9) that Dora stabilised very rapidly at 0.85.

- We then divided ϵ by 1.05, allowing Dora to explore for much longer. By the time it had treated the corpus 80 times, it still had not stabilised (Figure 13.10).

- Finally, we took a middle ground (Figure 13.11), dividing ϵ by 1.5, and Dora stabilised at a result of 0.86, beating the expert parameters of 0.81 by 0.05 or 6%.

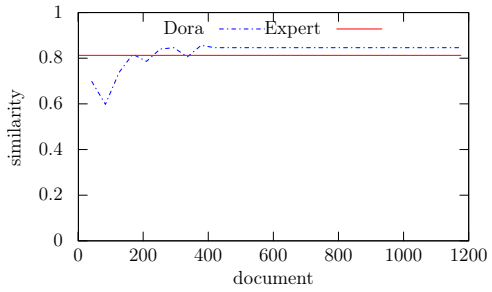


Figure 13.9 – The quality of the image analysis given ϵ starting at 0.2, divided by 2 every two times through the corpus, and stabilising at 0.85.

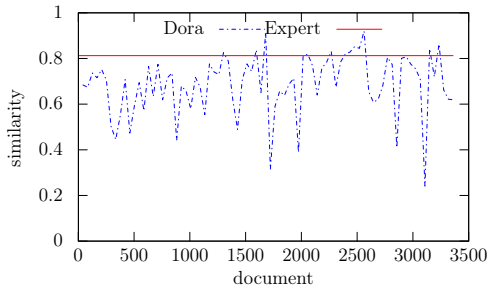


Figure 13.10 – The quality of the image analysis given ϵ starting at 0.2, divided by 1.05 every two times through the corpus, but not stabilising even after 80 times through the corpus.

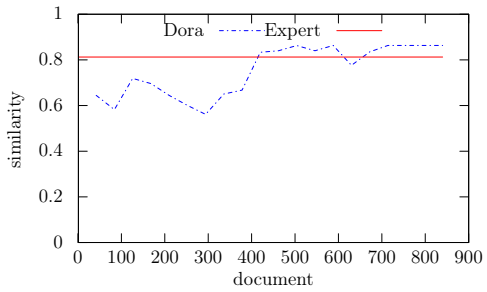


Figure 13.11 – The quality of the image analysis given ϵ starting at 0.2, divided by 1.5 every two times through the corpus, and stabilising at 0.86, beating the expert parameters by 0.05 or 6%.

13.7 Summary

The corpus, although tiny, was constructed based on as many different types of document and title as possible. Dora learnt, not only to parametrise the services correctly, but to do so on a case by case basis, beating the expert parameters convincingly, even on the "problem cases".

This validates the flexibility of our approach, showing that it can be easily and successfully applied to other settings than an IE document treatment chain.

Part III

Conclusion

Chapter 14

Discussion and perspectives

14.1 Conclusion

14.1.1 Summary

Let's go back to our bizarre car factory, and see how they are doing.

They hired a BIMBO to head up their customer feedback department, and she, in turn, hired a team of crack AIs. Each AI was equipped with the blueprints of all possible car models, and put in charge of organising the factory.

The AIs were on performance-related pay, with bonuses for happy customers. To measure the customer satisfaction, they issued a questionnaire every time a car was sold. Some customers didn't bring the questionnaires back immediately, and the vast majority were not even filled in, but the AIs still managed to earn a very good living, and in general customer satisfaction increased.

Then a bright spark at the factory suggested the monitoring device, designed to work alongside the questionnaires. The AIs were delighted, as this meant that they were still paid, even if customers did not want to fill in a questionnaire. Customers were thrilled too. They no longer had to score mint against olive, or asparagus against avocado, they could just say "I like green, preferably dark".

BIMBO noticed that the AIs were sometimes rather slow to file their paperwork, and so she hired a personal assistant, Dora to help them out.

The factory became such a success that they branched out, and bought up a tiny image analysis company.

14.1.2 Implications

Although we applied this approach to an IE document treatment chain, there is no reason why it should not be applied to other processes, from the design of factory floors to route-planning. The ability to mix and match the qualitative and numeric feedback means that the system can be finely tuned to meet the users' requirements.

The ability to effectively store the knowledge of the users in the "heads" of the AIs means that the longer the system is used, the more it should be able to deal competently with even the most esoteric of situations. The system becomes a knowledge bank, formalising the users' skills and experience. As people leave, or join a company, the system remains the connecting thread between them.

We can also use the unexpected choices that the AI makes, such as its surprising choice of the gazetteers in section 7.2, to perform "sanity checks" on a system. In the case of error in one of the services, it reduces developer labour cost by pinpointing the service and the functional behaviour at fault. In the case where it is not due to a code error, it could point to a misunderstanding of the user requirements.

14.1.3 Limits

Complexity of modelling: One of the limits of our approach is that the modelling task is not a simple one. Firstly, the measurable characteristics of the environment must be identified, along with all their possible values. A decision must then be taken whether to group those values together to avoid a state / action space explosion and speed up the learning, and if so, what size the groups should be, or whether to keep them separate to increase the accuracy of the learning. Secondly, all possible actions that might influence the environment must be listed.

We suspect that it should be possible to mix qualitative model learning (Pang and Coghill (2010) gives a recent overview) with the ability to learn the attributes of a state. For example, Bratko and Šuc (2003) learn qualitative models from the operators control strategy, and Karami et al. (2014) learn which attributes of a state are important.

Trampus and Mladenic (2014) examine the extraction of structured information from natural language text. Given a document, they try to detect which properties are the most relevant to extract, giving as much information, concisely about the document, as possible. Their "properties / slots" are equivalent to our (event) dimensions, and we suspect that this approach could be used to help construct the states of our MDP. Learning the structure of the problem is demonstrated by Strehl et al. (2007) whose structure-learning algorithm is a variation on R-Max and Degris et al. (2006) who develop SDYNA - a general framework for addressing large reinforcement learning problems by trial-and-error and no initial knowledge of their structure, and SPITI - an instantiation of SDYNA that uses incremental decision tree induction to learn the structure of a problem combined with an incre-

mental version of the Structured Value Iteration algorithm. Decision tree induction allows the agent to generalize from its history (c.f. Bratko and Šuc (2003) and their qualitative controller). Features are derived by Guan and Qiu (2007) who extract objects and regions from images that the user is interested in to improve image retrieval.

Limited tests: We would have liked to have run more combinations of parameters and tested more algorithms with BIMBO. Unfortunately, a thesis has to end some time, and we therefore had to make representative choices. Coggan (2004) made a study of different parameters for Q-Learning and SARSA, and we were particularly interested by the idea of "disaster situation" learning. In the IE chain, this could be used if the chain goes over time with no extraction, for instance.

Due to the problem of finding a sufficiently large annotated corpus, the documents were homogeneous. They were all approximately the same length and all in the same language, English. We would have liked to have introduced more variety, but we were also lacking the services to translate and extract events in different languages. Another limitation with the corpus was that each document only contained one ideal event. We therefore did not attempt to fusion events, or to work on how to allocate feedback over multiple events. We tried using the 51 web-pages that we had extracted in section 7.4, hoping to at least test the trained AI on them, but they were so noisy (for example with html formatting, adverts, and multiple stories) that even the expert chain was unable to correctly extract the target events from them. To make use of these pages, therefore, we would need to use additional services to clean them.

Unfortunately, the lack of choice of services also impacted the possible actions that we could offer to the AI, and the observations we could make on the documents, and hence the features that we could build into our states. This meant that our state / action space was smaller than we would have liked. One of our original goals was to show the substitution of one service for another if it were better for a given type of document. We considered (but decided against) artificially degrading one of the existing services.

We currently "spoon feed" the AI the feedback at the end of the treatment. What if the AI must also learn when to ask for rewards, as well as choose its actions? Obviously, it would need to minimise the number of demands. Intuitively, it could be interesting to ask for feedback when it chooses to explore, or when it is faced with two approximately equal choices. A study would need to be made into the convergence properties.

State / action space explosion: One of the known problems with MDPs is that in a complex system, there can be a rapid explosion of the state / action space (Sutton and Barto, 1998, Chapter 8). We designed our states to have nice, neat boundaries, but in real life, often the information about the environment cannot be compartmentalised so easily. For example, if we wanted to have a feature of the state which gave the source of the document, depending on the granularity, we might need an endless list, which would be impossible to model with our current approach.

This explosion can also result in system failure, as the memory becomes insufficient. Several attempts have been made to combat this problem. Dean and Givan (1997) reduce the model by aggregating the states, and Boutilier et al. (2000) cluster states that have the same estimated value or the same optimal choice of action by representing the states using decision trees that test the values of specific variables. Hoey et al. (1999) developed SPUDD (Stochastic Planning Using Decision Diagrams) which uses algebraic decision diagrams to represent value functions and policies. These decision graphs are compact, and group states by their values, allowing a generalisation to be made over the structure of the problem. The use of neural networks is another avenue to explore. Dini and Serrano (2012) attempt to solve the problem of scaling due to lack of space for the standard table of Q-values by using an artificial neural network as a function approximator instead (see Sutton and Barto (1998, Chapter 8) for an introduction). Their results show that for simple implementations, the standard table approach works better, but for more complex scenarios, the neural network appears to be more useful. Finally, potentially slow convergence can be tackled by using approximations of the policy iterations (for example (Ma and Powell, 2009)).

Discerning between an erroneous treatment, and faulty source information: A question that remains unanswered is how to differentiate between different types of user corrections. It may be that the treatment was flawed (which is our assumption in this manuscript), but it could also be that the original source was false, or even that the users are adding information gleaned from an external source and therefore not in the original document. Care needs to be taken that user corrections which cannot be inferred from the original document itself do not penalise the AI.

We considered allowing the system to ask for explicit feedback when the user corrects an entity, but this becomes intrusive. We can maybe draw a parallel with systems capable of filtering error-prone feedback from untrained users, for example Formiga et al. (2015). They examine the use of user cor-

rections to improve a statistical machine translation (SMT) system. Firstly they filter the user feedback to discard useless user translations. Secondly they align the system output and the user-edited sentence (using a variation of the Levenshtein distance on the words). Thirdly, they produce a new translation model and combine it with the one from the original system.

14.2 Future work

The limits listed above all provide interesting work for the future. We've identified several other areas that we would love to work on if we were allowed to extend this thesis indefinitely.

POMDPs and MOMDPs: We modelled our IE chain as a "basic" MDP. The services, however, will not necessarily produce a 100% accurate result. For instance, the language is detected with a certain percentage of confidence. We could therefore see the problem as learning in a Partially Observable MDP (POMDP) where the states (*e.g.* "language=en_gb") are only partially observable (in fact, we only see them via the results of the extractions) and not 100% reliable or accurate (the document may contain several languages, for example). We could therefore also have the observation "language=en_us" for the same document, or even "language=fr_fr". This is particuarly pertinant, not only for the extraction of the language but for any potentially subjective or ambiguous value (*e.g.* the domain - politics, economics, ...). In this case, we might imagine using a Mixed Observability MDP (MOMDP) (Ong et al., 2010) where we separate states into fully observable variables (*e.g.* treatment time or extraction made) and partially observable variables (*e.g.* language detected or domain).

Effective initialisation of the AI: The important Q-values to be initialised from the "expert chain" concern the first (and maybe second) service(s) called - the document has to pass through a Normaliser first (to convert to an XML resource). This Normaliser could also be capable of detecting the language, but if this is not used, then the resource must pass through a Language Detector. Once we have "format" and "language" filled, we can start playing with the translation / extraction services. It's possible to create libraries of services, so we could call "a Normaliser", and pick from a list, or give them different parameters. We could therefore imagine a more general initialisation, such as : Each state where format is not detected and time / number of services under threshold gives action Normaliser; language not yet

detected and time under threshold gives Language Detector, etc. This would still give a head-start to the AI, but not constrain the choice of services so much that it cannot find the optimum policy.

There's a hierarchy between the actions "choose the next service to which the resource should be sent", and the actions "choose the parameters for the service". We can treat them uniformly initially (combinatorially), but eventually, we could learn, for example, an HTN (Hierarchical Task Network), of the form : "extract named entities" decomposes into "pick the extraction service" then "choose the configuration of this extraction service" then "send resource to this extraction service". The instantiation of these sub-tasks would vary, for example which parameters to set would depend on the service, and which service to use would depend on the directories of services available. Kanani (2012), for instance, treats the global actions query, download and extract. The order of these actions is fixed, but the specific contents of the action are generated dynamically. We could draw a parallel with the IE chain, where language extraction has to happen before translation, and so the translation actions are only instantiated once the language has been identified.

As we saw in section 2.3, Doucy et al. (2008) construct a modifiable document treatment chain using the WSDL of each service to ensure that the substituted service is compatible, and we could, of course use the same technique. This would also allow the AI to generalise over actions, for example, it could learn that an extraction service should follow a language detection service, as well as learning that the *best* language detection for a particular case is X, and the *best* extraction service for that case is Y. We could also use the WSDLs with the hierarchical approach to accelerate the AI's learning process by giving it more information about how the services might fit together, rather than making it find out by trial and error, thus allowing it to build prior models of the actions and states. For example, the AI would no longer try the action "extract events" when it is in a state where the language is unknown if the WSDL of the extraction module specifies that the language is required.

We could even get rid of the need to interrogate the WSDLs by using *Microservice Architecture*. Microservices are independently and automatically deployable, they can be built into hierarchies, and organised around the business capabilities. This allows a decentralised control, as the services no longer have to be orchestrated according to their interfaces, and is effectively what the AIs learnt to do with BIMBO's dynamic router.

We might also imagine that if we had several AIs running on different chains, that we could transfer knowledge between them. Singh (1992) invest-

igates the sharing of knowledge between tasks with common sub-tasks. In our case, for instance, if the highest task was "extract events from document", we could use the knowledge gained for the sequence "collect document, clean it, detect format, detect language" which would be the same for all documents, and generalise over the states. In this way, we could have "trained" sequences or blocks of services which could be plugged into the algorithms, instead of individual services. This would also help to generalize the state space.

Reliance on the system: We saw that Akrour et al. (2013) examine the problem of trust between the agent and the human. If the agent consistently presents trajectories that are difficult to compare, then the human is likely to become inconsistent, and the agent loses trust in them. On the other hand, if the agent is well-taught at the start and presents useful information to the human, their error-rate decreases, and the agent's trust is gained. We can imagine that if the AI is not correctly initialised, or if it is allowed to explore too much, then it might produce very poor results, such as no extractions, or ridiculous summaries. This would be counter-productive to our objective of reducing the analysts' workload, and could result in discouraging them from using the system. It would be interesting to explore the implications of providing a safety net, not only to prevent operational problems for the analyst, and possible loss of trust on both sides, but also to refine the reward process. For example, the expert chain could be run alongside the AI, and the results used to complement those of the AI. If the AI did not extract an event, but the expert chain did, then the AI could be punished, unless the analyst deletes the event extracted by the expert chain because it was false, in which case we should reward the AI.

We can, however, imagine some disadvantages to building up mutual trust.

Firstly, in seeking feedback from the analyst, we are not just capturing their knowledge, but also their preconceptions, and their subjective views. We could seek to dampen this effect by taking the views of a group of analysts, but we would obtain the preconceptions of the group as a whole instead of the individuals.

Secondly, the trust could lead to complacency. This could be the system which loses its adaptability to new situations (for example, if the exploration is reduced too much), or the analysts who rely too much on the system without using their experience, intuition and even common sense. This complacency could be a weak point, attackable by disinformation (intentional)

or misinformation (unintentional). For example, the national media might be swamped with false reports of a certain candidate's popularity in order to sway voters. Another example could be the false press communiqué issued on the 22^{nd} November 2016 purporting to come from the group Vinci, announcing that their financial director was to be fired for fraud, and their accounts audited. Vinci suffered massive losses on the stock market that day until the fraud was discovered and denounced. Yet another example could be that of an adversary who knows that the analysts are monitoring a given situation and so creates false websites, news reports, tweets, *etc.* to make it seem that an attack will take place in a certain area. Once he is sure that the defensive troops are diverted there, he is free to strike elsewhere. Note that the adversary does not even need to know the model that we are using, nor how it is parametrised or trained, attacks have been successfully made which "steal" the model, enabling the adversary to duplicate its functionality (Tramèr et al., 2016).

Nor are these attacks limited to text; image recognition systems can be fooled into thinking that they see something that is not there, either by altering an image in a way that is imperceptible to the human eye (Szegedy et al., 2013; Kurakin et al., 2016), or through what seems like white noise or random patterns (Nguyen et al., 2014).

Quality improvement: We currently treat the services as "black boxes", aiming to improve the way that the agent combines and parametrises them. We could also aim to improve the services themselves. For instance, Formiga et al. (2015); Culotta et al. (2006) show that individual services can be automatically updated based on user edits. Even closer to home, Caron et al. (2014) and Amann et al. (2013) improve the quality in a treatment chain by tracing the provenance of each piece of data, annotating the XML, and inferring new values. The premise is that services can propagate errors from their input data to the rest of the chain. The cumulation of errors with each service can mean a rapid degradation of quality of the end results. We can use the feedback that we gather as a quality value. As well as providing a feedback for the treatment as a whole, the feedback could also be applied to specific services in that chain, depending on the errors propagated. For instance, the quality or consistency of automatically translated text can never be higher than that of the original text, and if the detected language is false, then the consistency is bound to be low. Knowing this, we could imagine that the negative feedback is weighted to be applied to the translation service, and not to the order in which the services were chained together or

parametrised. This method has the advantage of being non-intrusive, requiring no code changes to the services themselves, and can be applied to any XML based data processing workflow.

We've also mentioned that the AI's decisions can lead to interesting discoveries. Bratko and Šuc (2003) cite the importance of understanding the user's strategy, and defining it in terms that are easily accessible to the user, for example visually. It could be interesting to develop a visualisation of the thought-process behind the AI, allowing the developers, or even an interested user, to see at a glance why a certain chain configuration was chosen.

Appendices

Appendix A

An example configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:jaxws="http://cxf.apache.org/jaxws"
    xmlns:soap="http://cxf.apache.org/bindings/soap"
    xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

    <description>
        Contains the configuration parameters for BIMBO: Benefiting from Intelligent and Measurable Behaviour Optimisation
    </description>

    <bean id="properties"
        class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
        <property name="ignoreResourceNotFound" value="true" />
        <property name="properties">
            <props>
                <!-- Your home here -->
                <prop key="homeDirectory">/home/nicart/</prop>
                <!-- The folder containing all the stuff necessary to run BIMBO -->
                <prop key="bundleTesting">${homeDirectory}BundleTesting/</prop>
                <!-- The bundle instance should be in the directory ${homeDirectory}/${bundleTesting}/ -->
                <prop key="bundleInstance">Thesis-WebLab</prop>
                <!-- If using the bundle provided, this should not change -->
                <prop key="sparqlEndpoint">http://localhost:3030</prop>
                <!-- Used to find the routes file, and the gazetteers which the AI uses
                    to replace those in GATE -->
                <prop key="thesis-aiResources">${bundleTesting}resources/
                </prop>
            </props>
        </property>
    </bean>

    <bean id="ConfigAI" class="config.ConfigAI">
        <description>
            Configuration parameters for the artificial intelligence
        </description>
        <!-- The number of documents to be processed per batch -->
        <property name="noDocs" value="1" />
        <!-- The number of batches, e.g. set to 5 if you want to run the same documents through 5 times -->
        <property name="noBatches" value="10" />
        <!-- Whether the same documents should be used each batch -->
        <property name="sameDocs" value="FALSE" />
        <!-- Whether values have been stored that can be used to continue a run already started. -->
        <property name="sameRun" value="FALSE" />
        <!-- Should we use the AI (true), or the standard chain (false) -->
        <property name="useAIForRun" value="TRUE" />
        <!-- Algorithm to use : QL = QLearning, RM = RMax, RMR = RMaxRandom, SSBRM = SSBRMax, VM = VMax VMR = VMaxRandom, SSBQL = SSBQLearning, DORA = Dora -->
        <property name="algoToUse" value="QL" />
    </bean>

    <bean id="ConfigCommonQLParams" class="config.ConfigCommonQLParams">
        <description>
            Configuration parameters for the QLearning and SSB QLearning algorithms
```

```
53        </description>
54        <!-- The discount rate to be used for solving the problem -->
55        <property name="gamma" value="0.9" />
56        <!-- The decay parameter -->
57        <property name="lambda" value="0.95" />
58        <!-- Explore if random value less than epsilon -->
59        <property name="epsilon" value="0.4" />
60        <!-- The minimum value of epsilon, below which we do not divide by 2 any
          more -->
61        <property name="epsilonMinimum" value="0.05" />
62        <!-- Divide epsilon by 2 every changeEpsilon batches -->
63        <property name="changeEpsilon" value="50" />
64        <!-- Do we use traces or not (always uses newAlpha)-->
65        <property name="useLambda" value="FALSE" />
66        <!-- If using traces, do we profit from the exploration results?-->
67        <property name="profitExploration" value="TRUE" />
68        <!-- The learning rate (flat 0.2, alpha 0.55) -->
69        <property name="learningRate" value="0.55" />
70        <!-- Are we decreasing epsilon using the number of visits to the state? -->
71        <property name="decreaseEpsilonByVisits" value="TRUE" />
72        <!-- If so, then what exponent should we use 0.5 < beta <= 1 -->
73        <property name="exponent" value="0.55" />
74    </bean>
75
76    <bean id="ConfigQLearning" class="config.ConfigQLearning">
77        <description>
78            Configuration parameters for the QLearning algorithm
79        </description>
80        <!-- Do we use the "normal" version of QLearning, but with alpha modified
          to take into account the number of visits or not -->
81        <property name="useNewAlpha" value="TRUE" />
82        <!-- Do we have whole trajectories as explore / exploit? (not just
          individual actions) -->
83        <property name="wholeTrajectoryExploreExploit" value="FALSE" />
84    </bean>
85
86    <bean id="ConfigSSBQLearning" class="config.ConfigSSBQLearning">
87        <description>
88            Configuration parameters for the SSBQLearning algorithm
89        </description>
90        <!-- Which version of phi(WL1,WL2) to use : EXP = expectation R1-R2; MAG =
          magnitude in 1000, 100, 1, 0, etc.; STR = strict in 1 to -1; STR1000 = strict
          in 1000 to -1000 -->
91        <property name="phiVersion" value="STR1000" />
92        <!-- What parameters to use for the dummy wealth level -->
93        <property name="dummySimilarity" value="0" />
94        <property name="dummyTimeTaken" value="9999" />
95        <property name="magTop" value="30" />
96        <property name="magMid" value="20" />
97        <property name="magLow" value="10" />
98    </bean>
99
100   <bean id="ConfigRMax" class="config.ConfigRMax">
101       <description>
102           Configuration parameters for the RMax algorithm, and the RMaxRandom
          algorithm
103       </description>
104       <!-- The discount rate to be used for solving the problem -->
105       <property name="gamma" value="0.9" />
106       <!-- The precision to which the problem must be solved -->
107       <property name="epsilon" value="1.0" />
108       <!-- The number of times a couple (state, action) will be visited before
          the AI is sure of its value (normally 20) -->
109       <property name="experience" value="2.0" />
110       <!-- The maximum possible reward -->
111       <property name="rMax" value="1000.0" />
112       <!-- The horizon (-1 for infinity) -->
113       <property name="horizon" value="-1" />
114   </bean>
115
116   <bean id="ConfigVMax" class="config.ConfigVMax">
117       <description>
118           Configuration parameters for the VMax algorithm
119       </description>
120       <!-- The discount rate to be used for solving the problem -->
121       <property name="gamma" value="0.95" />
```

```
122         <!-- The precision to which the problem must be solved -->
123         <property name="epsilon" value="0.01" />
124         <!-- The number of times a couple (state, action) will be visited before
          the AI is sure of its value (normally 20) -->
125         <property name="experience" value="100" />
126         <!-- The maximum possible reward -->
127         <property name="rMax" value="1000.0" />
128         <!-- The horizon (-1 for infinity) -->
129         <property name="horizon" value="-1" />
130     </bean>
131
132     <bean id="ConfigStates" class="config.ConfigStates">
133         <description>
134             Configuration parameters for the state vector
135         </description>
136         <!-- The possible languages, including empty -->
137         <property name="languages" value="en, " />
138         <!-- The possible document formats, including empty -->
139         <property name="formats" value="text/plain, " />
140         <!-- The time intervals that are used to create the states, for example, if
          the threshold is 18, and timeInterval is 10, there will be two states. -->
141         <!-- Also used as the margin when determining if two times are similar or
          not -->
142         <property name="timeInterval" value="5" />
143         <!-- The intervals that will be used to group the number of services passed
                through into states -->
144         <property name="serviceIntervals" value="5, 20, 100" />
145         <!-- Whether an interesting event has been extracted or not -->
146         <property name="interestingEventExtracted" value="true, false" />
147         <!-- Whether any event has been extracted or not -->
148         <property name="anyEventExtracted" value="true, false" />
149     </bean>
150
151     <bean id="ConfigSimilarityWeights" class="config.ConfigSimilarityWeights">
152         <description>
153             Configuration parameters for the weightings given to the
154             four event dimensions and the context in the calculations of similarity
155         </description>
156         <!-- semantic dimension -->
157         <property name="semWeight" value="1" />
158         <!-- temporal dimension -->
159         <property name="temWeight" value="1" />
160         <!-- spatial dimension -->
161         <property name="spaWeight" value="1" />
162         <!-- agentive dimension -->
163         <property name="agWeight" value="1" />
164         <!-- contextual -->
165         <property name="conWeight" value="1" />
166     </bean>
167
168     <bean id="ConfigGeneral" class="config.ConfigGeneral">
169         <description>
170             Configuration parameters for the program surrounding the
171             AI
172         </description>
173         <!-- The length of time in seconds that will be used as a parameter in the
          state vectors, and to determine how long the AI can take before being
          forcibly sent back to the start and punished (this depends on the machine,
          and should be calibrated using the expert chain) -->
174         <property name="threshold" value="10" />
175         <!-- The number of documents that will be processed before the triplestore
          is cleared down -->
176         <property name="clearTriplestore" value="100" />
177         <!-- Gives the sleep time between checks for the end of the chain -->
178         <property name="sleep" value="40" />
179         <!-- The services plus stop to allow the AI to choose to stop processing
          the resource. This is used to define the action choices for the AI (0 = tika,
          1 = ngramj, 2 = geo, 3 = GATE), etc. -->
180         <property name="servicesAndStop"
181             value="direct:tika, direct:ngramj, direct:gate, direct:geo, stop" />
182         <!-- Whether the WebLab resource should be indexed or not -->
183         <property name="indexResource" value="FALSE" />
184         <!-- Whether the triplestore should be used or not. If FALSE, then the
          internal Jena InfModel is used (for better performance) -->
185         <property name="useTriplestore" value="FALSE" />
186         <!-- Whether the similarity results should be stored or not -->
187
```

```
188         <property name="writeSimResults" value="FALSE" />
189         <!-- Whether the bodies should be stored or not -->
190         <property name="writeBody" value="FALSE" />
191         <!-- Whether the rewards should be delayed until the end of batch (TRUE) or
            given every document (FALSE) -->
192         <property name="delayRewards" value="FALSE" />
193         <!-- If the rewards are delayed until the end of batch, what percentage of
          the rewards should be given -->
194         <property name="percentageToGive" value="100" />
195         <!-- Whether BIMBO should send Esther progress reports by SMS -->
196         <property name="sendSMS" value="FALSE" />
197         <!-- If so, every how many batches? -->
198         <property name="sendSMSBatches" value="10" />
199     </bean>
200
201     <bean id="ConfigPaths" class="config.ConfigPaths">
202         <description>
203             Configuration parameters for the paths and filenames needed by the AI.
          If the bundle testing folder has been copied  correctly, these should never
          change
204         </description>
205         <!-- SPARQL RESOURCES -->
206         <property name="sparqlQueryEndpoint" value="${sparqlEndpoint}/KDB-infer/
          query" />
207         <property name="sparqlUpdateEndpoint" value="${sparqlEndpoint}/KDB/update"
          />
208         <property name="sparqlRequestsPath" value="${bundleTesting}/requests/" />
209
210         <!-- BUNDLE RESOURCES -->
211         <property name="bundleTesting" value="${bundleTesting}" />
212         <property name="bundleFolder" value="${bundleTesting}${bundleInstance}/" />
213         <property name="gateConf"
214             value="${bundleTesting}${bundleInstance}/conf/services/wookie-gate-
          extraction.cxf-servlet.xml" />
215         <property name="gazetteerPath"
216             value="${bundleTesting}${bundleInstance}/conf/gate/plugins/EADS/ENG/
          Gazetteers/Events/OsintEvents/" />
217
218         <!-- AI RESOURCES -->
219         <!-- Location of toIndex folder in BIMBOs treatment chain -->
220         <property name="aiChainToIndex" value="${bundleTesting}ai-chain-to-process/
          " />
221         <!-- Location of the copied GATE master lists manipulated by the AI -->
222         <property name="gazetteerListPath" value="${thesis-aiResources}gazetteers/"
          />
223         <!-- Location of the resources used by the AI -->
224         <property name="aiResourcesPath" value="${thesis-aiResources}" />
225         <!-- Where the Camel Routes are -->
226         <property name="routesFile" value="file:${thesis-aiResources}DynamicRoutes.
          xml" />
227         <!-- Where the GTD csv files and source pages are stored -->
228         <property name="gtdCsvLocations" value="${bundleTesting}
          pages_all_extractables" />
229
230         <!-- TEST RESULTS -->
231         <!-- Where the results are stored - the learnt values, rewards, console
          summary and explore / exploit data, plus the events extracted -->
232         <property name="resultFiles" value="${bundleTesting}Results/" />
233         <!-- Where the bodies are stored - the WebLab resource -->
234         <property name="bodyFiles" value="${bundleTesting}Morgue/" />
235         <!-- The base name of the action / state / reward csv file written at each
236             iteration, and at the end the filename of the learnt values -->
237         <property name="learntValuesFilename" value="learntValues" />
238         <!-- Used to distinguish the final action / state / reward csv file which
239             is used to restart the run, from those written at each iteration -->
240         <property name="restartString" value="Restart" />
241         <!-- The filename of the QL algorithm, stored for a repeat run -->
242         <property name="QLSaveFile" value="QL.ser" />
243         <!-- The filename of the rMax algorithm, stored for a repeat run -->
244         <property name="rMaxSaveFile" value="rMaxSave.ser" />
245         <!-- The filename of the vMax algorithm, stored for a repeat run -->
246         <property name="vMaxSaveFile" value="vMaxSave.ser" />
247         <!-- The filename of the SSBQL algorithm, stored for a repeat run -->
248         <property name="SSBQLSaveFile" value="SSBQLSave.ser" />
249         <!-- The filename of the Dora algorithm, stored for a repeat run -->
250         <property name="DoraSaveFile" value="DoraSave.ser" />
```

```
251         <!-- The base name of the console dat file written at each iteration, and
252             at the end -->
253         <property name="consoleFilename" value="console_summary" />
254         <!-- The base name of the reward dat file written at each iteration, and at
            the end -->
255         <property name="rewardFilename" value="rewards" />
256         <!-- The base name of the explore/ exploit dat file written at each
            iteration, and at the end -->
257         <property name="exploreExploitFilename" value="exploreExploit" />
258         <!-- The file containing the serialized GTD events -->
259         <property name="gtdEventsSerFilename" value="gtdEventsModified.ser" />
260         <!-- The file containing the events extracted by BIMBO -->
261         <property name="eventsExtractedFilename" value="eventsExtracted" />
262         <!-- The file containing (s, a, s prime, r), used to construct dot file for
            graphviz -->
263         <property name="transitionRewardsFilename" value="transitionRewards" />
264     </bean>
265
266     <bean id="ConfigGazetteers" class="config.ConfigGazetteers">
267         <description>
268             Contains the configuration parameters for the GATE gazetteers that the
            AI controls
269         </description>
270         <!-- File names of the nouns and verbs gazetteers -->
271         <property name="gazetteerNounsFilenames"
272             value="listEventNounsBombingOnly.def, listEventNounsInjureOnly.def,
            listEventNounsDummyOnly.def, listEventNounsShootingOnly.def,
            listEventNounsMixed.def" />
273         <property name="gazetteerVerbsFilenames"
274             value="listEventVerbsBombingOnly.def, listEventVerbsInjureOnly.def,
            listEventVerbsDummyOnly.def, listEventVerbsShootingOnly.def,
            listEventVerbsMixed.def" />
275         <!-- File names of the nouns and verbs master lists -->
276         <property name="gazetteerNounsMasterFiles" value="listEventNouns.def,
            listEventNounsEmpty.def" />
277         <property name="gazetteerVerbsMasterFiles" value="listEventVerbs.def,
            listEventVerbsEmpty.def" />
278         <!-- Location of the nouns and verbs lists -->
279         <property name="nounsListLocation" value="EventNouns/listEventNouns.def" />
280         <property name="verbsListLocation" value="EventVerbs/listEventVerbs.def" />
281         <!-- Which gazetteers should be used when resetting GATE -->
282         <property name="resetGazetteerNouns" value="listEventNounsEmpty.def" />
283         <property name="resetGazetteerVerbs" value="listEventVerbsEmpty.def" />
284         <!-- Which event is the interesting one, getting extra rewards -->
285         <property name="interestingEvent" value="http://weblab.ow2.org/wookie#
            BombingEvent" />
286     </bean>
287
288 </beans>
```

Listing A.1 – An example configuration file.

Appendix B

Wookie Ontology

Figure B.1 – The WOOKIE ontology Copyright ©2016 Airbus Defence and Space.

Appendix C

Chain and dynamic router

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:jaxws="http://cxf.apache.org/jaxws"
    xmlns:soap="http://cxf.apache.org/bindings/soap"
    xmlns:camel="http://camel.apache.org/schema/spring"
    xmlns:cxf="http://camel.apache.org/schema/cxf"
    xmlns:osgi="http://www.springframework.org/schema/osgi"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
      http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
      http://camel.apache.org/schema/spring http://camel.apache.org/schema/spring/
     camel-spring.xsd
      http://www.springframework.org/schema/osgi http://www.springframework.org/
     schema/osgi/spring-osgi.xsd
      http://cxf.apache.org/jaxws http://cxf.apache.org/schemas/jaxws.xsd
      http://camel.apache.org/schema/cxf http://camel.apache.org/schema/cxf/camel-
     cxf.xsd">

    <bean id="bimbo" class="bimbo.Bimbo" />
    <bean id="filereader" class="processor.FileReaderProcessor" />

    <!--============================================================== -->
    <!--======================== Web services ======================== -->
    <!--============================================================== -->

    <osgi:service id="osgi-service-tika" ref="tika"
        interface="org.apache.camel.Endpoint">
        <osgi:service-properties>
            <entry key="name" value="service-tika" />
        </osgi:service-properties>
    </osgi:service>

    <osgi:service id="osgi-service-ngramj" ref="ngramj"
        interface="org.apache.camel.Endpoint">
        <osgi:service-properties>
            <entry key="name" value="service-ngramj" />
        </osgi:service-properties>
    </osgi:service>

    <osgi:service id="osgi-service-gate" ref="wookie-gate"
        interface="org.apache.camel.Endpoint">
        <osgi:service-properties>
            <entry key="name" value="service-gate" />
        </osgi:service-properties>
    </osgi:service>

    <osgi:service id="osgi-service-geo" ref="geo"
        interface="org.apache.camel.Endpoint">
        <osgi:service-properties>
            <entry key="name" value="service-geo" />
        </osgi:service-properties>
    </osgi:service>

```

Figure C.1 – Chain definition in XML (part 1) - declaration of web services.

```
<!-- ============================================================ -->
<!-- ======== Declaration of Camel context and route ========== -->
<!-- ============================================================ -->
<camelContext id="rest" errorHandlerRef="myErrorHandler"
    xmlns="http://camel.apache.org/schema/spring">
    <errorHandler id="myErrorHandler" type="DefaultErrorHandler">
        <redeliveryPolicy maximumRedeliveries="0"
            redeliveryDelay="0" allowRedeliveryWhileStopping="false"
            retryAttemptedLogLevel="INFO" />
    </errorHandler>
    <route id="consumeFile" streamCache="true" autoStartup="true">
        <from uri="pathTo/ai-chain-to-process/" />
        <log message="new file to process: ${file:name}" />
        <convertBodyTo type="java.io.File" />
        <setHeader headerName="weblab:dc:source">
            <simple>${headers.CamelFileAbsolutePath}</simple>
        </setHeader>
        <setHeader headerName="weblab:dct:modified">
            <simple resultType="java.util.Date">${header.CamelFileLastModified}</
    simple>
        </setHeader>
        <setHeader headerName="weblab:wlp:hasGatheringDate">
            <simple resultType="java.util.Date">${date:now:yyyy-MM-dd'T'HH:mm:ss.
    SSSZ}
            </simple>
        </setHeader>
        <setHeader headerName="weblab:wlp:hasOriginalFileName">
            <simple>${header.CamelFileNameOnly}</simple>
        </setHeader>
        <setHeader headerName="weblab:wlp:hasOriginalFileSize">
            <simple>${header.CamelFileLength}</simple>
        </setHeader>
        <setHeader headerName="weblab:dc:identifier">
            <simple resultType="java.util.Date">${date:now:yyyy-MM-dd'T'HH:mm:ss.
    SSSZ}</simple>
        </setHeader>
        <filter>
            <simple>${file:ext} == 'pdf'</simple>
            <setHeader headerName="weblab:dc:format">
                <constant>application/pdf</constant>
            </setHeader>
        </filter>
        <to uri="weblab://create?type=Document&amp;amp;outputMethod=xml" />
        <convertBodyTo type="java.lang.String" />
        <to uri="direct:start" />
    </route>

```

Figure C.2 – Chain definition in XML (part 2) - declaration of Camel context and file consumer route.

```
1       <route>
2          <from uri="direct:dummy" />
3          <log message="Dummy" />
4       </route>
5       <route>
6          <from uri="direct:tika" />
7          <to uri="weblab:analyser:service-tika" />
8       </route>
9       <route>
10         <from uri="direct:ngramj" />
11         <to uri="weblab:analyser:service-ngramj" />
12      </route>
13      <route>
14         <from uri="direct:gate" />
15         <to uri="weblab:analyser:service-gate" />
16      </route>
17      <route>
18         <from uri="direct:geo" />
19         <to uri="weblab:analyser:service-geo" />
20      </route>
21      <route>
22         <from uri="direct:filereader" />
23         <process ref="filereader" />
24         <to uri="direct:start" />
25      </route>
26      <route id="start">
27         <from uri="direct:start" />
28         <dynamicRouter>
29            <method ref="bimbo" method="performActionMaster" />
30         </dynamicRouter>
31      </route>
32   </camelContext>
33</beans>
34
```

Figure C.3 – Chain definition in XML (part 3) - declaration of routes, including Chain definition.

Appendix D

Full web page

This web page is reproduced here with the kind permission of Kasturi & Sons Ltd (KSL).

Front Page

YW Quiz 2009

Chandraayan I

Front Page

Maoists unleash violence in West Bengal, Jharkhand, Bihar

Trucks set ablaze, telecom tower blasted, rail track blown up

One killed, another abducted in West Bengal

Attacks occur on the first day of the two-day shutdown

Midnapore/Ranchi/Patna: Maoists on Monday unleashed a wave of attacks in which a Jharkhand Party member was killed in West Bengal. They blew up rail tracks and set trucks ablaze in Jharkhand and blasted a telecommunications tower in Bihar.

The attacks come even as the Union government is all set to launch a massive offensive against Left wing extremism in the naxalite-hit States

The violence occurred on the first day of a two-day shutdown called by the Communist Party of India (Maoist).

The body of Jharkhand Party member Kanai Murmu, who was kidnapped from his Ergada village in the early hours of Monday, was found later, the police said.

Informed sources said another member of the party, Ananda Mahato, was abducted by Maoists from Kulabheda village in the same Midnapore district. Both Murmu and Mahato were members of the anti-Maoist "Maobadi Protirodh Committee." At Ghatbera in Purulia district of West Bengal, Maoists ransacked the house of a local CPI(Marxist) leader and shot and injured a member of the village resistance committee.

News Update

Stories in this Section

- Maoists unleash violence in West Bengal, Jharkhand, Bihar
- Joint operations to continue: Buddhadeb
- Industrial growth soared to a high in August
- Ostrom, Williamson win Economics Nobel
- They showed that economic analysis can shed light on social organisation
- Lashkar chief's release cause of global concern
- Lahore court quashes two cases against Saeed
- MPs team visits hill region in Sri Lanka, interacts with Indian-origin workers
- India to take up Saeed's case with Pakistan
- Prithvi-II missiles successfully test-fired
- Villagers want Dinakaran's surplus land holdings redistributed to them
- "No channel opened with Vijayakant"
- Ajmal wants trial held in international court
- Rs. 2,000-cr. taxes to fund flood relief
- Candidates
- Directive to follow panel

As the attackers failed to find Chandan Singh Laya, CPI(Marxist) Ghatbera local committee secretary, they set his house ablaze.

In Jharkhand, Maoists blew up rail tracks at Jharandih in Dhanbad, resulting in the Shaktipunj Express and some local trains being held up at various points, Senior Public Relations Officer of Dhanbad Rail Division Amrendra Das said adding a light engine derailed after a one-and-half metre section of the track was blown up.

A group of 12 Maoists set three trucks ablaze in Giridih district's Isri area and blocked the Dumri-Giridih road with felled trees. Maoists also partially blasted a road bridge connecting Dumri to the Grand Trunk Road and gunshots were heard, Giridih Superintendent of Police Ravi Kant Dhan said.

Hazaribagh Superintendent of Police Pankaj Kamboj said Maoists partially damaged a road with explosives at Sardalo, bordering Bokaro district.

In Bihar, Maoists blasted the telecom tower at Salaiya village. Maoists also dug up a 15-metre stretch of a road at Chanda village, disrupting traffic between Deo and Dhibra. They left behind pamphlets, claiming responsibility for the action, official sources said.

Bihar Additional Director-General of Police (Headquarters) Neelmani said tight security arrangements were in place to counter the Maoist bandh. — PTI

Printer friendly page
Send this article to Friends by E-Mail

Front Page

News: ePaper | Front Page | National | Tamil Nadu | Andhra Pradesh | Karnataka | Kerala | New Delhi | Other States | International | Opinion | Business | Sport | Miscellaneous | Engagements |
Advts: Retail Plus | Classifieds | Jobs | Obituary | Updates: Breaking News |

guidelines
- Pakistani diplomat electrocuted
- Kidnapped Delhi child found dead in relative's car
- Petition filed against Nitish
- Police deployed in Kaithal
- Delhi can deliver great Games: Fennell
- Pakistan diplomat dies of electric shock
- Worker dies after inhaling poisonous gases
- 10 killed in UP floods
- Fennell confident
- Cases withdrawn
- One-way soon at Lakdikapul

Archives
Yesterday's Issue
Datewise

Features:
Magazine
Literary Review
Metro Plus
Open Page
Education Plus
Book Review
Business
SciTech
NXg
Friday Review
Cinema Plus
Young World
Property Plus
Quest

The Hindu Group: Home | About Us | Copyright | Archives | Contacts | Subscription
Group Sites: The Hindu | The Hindu ePaper | Business Line | Business Line ePaper | Sportstar | Frontline | Publications | eBooks | Images | Ergo | Home |

Appendix E

Full WebLab resource

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<resource xsi:type="ns3:Document" uri="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39" xmlns:ns3="http://weblab.ow2.org/core/1.2/model#" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <annotation uri="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39#a0">
        <data xmlns:wl="http://weblab.ow2.org/core/1.2/model#" xmlns:resourceContainer="http://weblab.ow2.org/core/1.2/services/resourcecontainer">
            <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
                <rdf:Description rdf:about="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39">
                    <wlp:hasNativeContent rdf:resource="file:data/content/weblab.2390572096489910238.content" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/processing#"/>
                    <wlp:hasOriginalFileSize rdf:datatype="http://www.w3.org/2001/XMLSchema#long" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/processing#">23023</wlp:hasOriginalFileSize>
                    <wlp:hasOriginalFileName xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/processing#">200910120024_1.html</wlp:hasOriginalFileName>
                    <dc:source xmlns:dc="http://purl.org/dc/elements/1.1/">data/toIndex/200910120024_1.html</dc:source>
                    <dct:modified rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime" xmlns:dct="http://purl.org/dc/terms/">2014-07-08T15:01:22+02:00</dct:modified>
                    <wlp:hasGatheringDate rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/processing#">2014-07-08T15:14:54.719+02:00</wlp:hasGatheringDate>
                </rdf:Description>
            </rdf:RDF>
        </data>
    </annotation>
    <annotation uri="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39#a2">
        <data xmlns:wl="http://weblab.ow2.org/core/1.2/model#" xmlns:resourceContainer="http://weblab.ow2.org/core/1.2/services/resourcecontainer">
            <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
                <rdf:Description rdf:about="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39#a2">
                    <dct:created xmlns:dct="http://purl.org/dc/terms/">2014-07-08T15:15:43.840+02:00</dct:created>
                    <wlp:isProducedBy rdf:resource="http://weblab.ow2.org/service/normaliser/tika" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/processing#"/>
                </rdf:Description>
                <rdf:Description rdf:about="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39">
                    <dc:format xmlns:dc="http://purl.org/dc/elements/1.1/">text/html</dc:format>
                    <dc:title xmlns:dc="http://purl.org/dc/elements/1.1/">The Hindu : Front Page : Maoists unleash violence in West Bengal, Jharkhand, Bihar</dc:title>
                </rdf:Description>
            </rdf:RDF>
        </data>
    </annotation>
    <annotation uri="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39#a4">
        <data xmlns:wl="http://weblab.ow2.org/core/1.2/model#" xmlns:resourceContainer="http://weblab.ow2.org/core/1.2/services/resourcecontainer">
            <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
                <rdf:Description rdf:about="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39">
```

```
35                      <dc:language xmlns:dc="http://purl.org/dc/elements/1.1/">en</
    dc:language>
36                  </rdf:Description>
37                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427f-
    a697-0ebb0c238e39#a4">
38                      <dcterms:created xmlns:dcterms="http://purl.org/dc/terms/">
    2014-07-08T15:15:43+0200</dcterms:created>
39                      <wlp:isProducedBy rdf:resource="http://weblab.ow2.org/
    services#LanguageExtraction" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
40                  </rdf:Description>
41              </rdf:RDF>
42          </data>
43      </annotation>
44      <annotation uri="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39#a66">
45          <data xmlns:wl="http://weblab.ow2.org/core/1.2/model#"
    xmlns:resourceContainer="http://weblab.ow2.org/core/1.2/services/
    resourcecontainer">
46              <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
47                  <rdf:Description rdf:about="http://sws.geonames.org/1227603/">
48                      <rdf:type rdf:resource="http://www.geonames.org/ontology#
    Feature"/>
49                      <geo:parentFeature rdf:resource="http://sws.geonames.org
    /6295630/" xmlns:geo="http://www.geonames.org/ontology#"/>
50                      <geo:parentFeature rdf:resource="http://sws.geonames.org
    /6255147/" xmlns:geo="http://www.geonames.org/ontology#"/>
51                      <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#
    ">Democratic Socialist Republic of Sri Lanka</rdfs:label>
52                      <geo:name xmlns:geo="http://www.geonames.org/ontology#">
    Democratic Socialist Republic of Sri Lanka</geo:name>
53                  </rdf:Description>
54                  <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#kerala">
55                      <geo:hasSpatialThing rdf:resource="http://sws.geonames.org
    /1267254/" xmlns:geo="http://www.geonames.org/ontology#"/>
56                  </rdf:Description>
57                  <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#karnataka">
58                      <geo:hasSpatialThing rdf:resource="http://sws.geonames.org
    /1267701/" xmlns:geo="http://www.geonames.org/ontology#"/>
59                  </rdf:Description>
60                  <rdf:Description rdf:about="http://sws.geonames.org/1273293/">
61                      <rdf:type rdf:resource="http://www.geonames.org/ontology#
    Feature"/>
62                      <geo:parentFeature rdf:resource="http://sws.geonames.org
    /6295630/" xmlns:geo="http://www.geonames.org/ontology#"/>
63                      <geo:parentFeature rdf:resource="http://sws.geonames.org
    /6255147/" xmlns:geo="http://www.geonames.org/ontology#"/>
64                      <geo:parentFeature rdf:resource="http://sws.geonames.org
    /1269750/" xmlns:geo="http://www.geonames.org/ontology#"/>
65                      <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#
    ">National Capital Territory of Delhi</rdfs:label>
66                      <geo:name xmlns:geo="http://www.geonames.org/ontology#">
    National Capital Territory of Delhi</geo:name>
67                  </rdf:Description>
68                  <rdf:Description rdf:about="http://sws.geonames.org/6295630/">
69                      <rdf:type rdf:resource="http://www.geonames.org/ontology#
    Feature"/>
70                      <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#
    ">Earth</rdfs:label>
71                      <geo:name xmlns:geo="http://www.geonames.org/ontology#">Earth
    </geo:name>
72                  </rdf:Description>
73                  <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#west_bengal">
74                      <geo:hasSpatialThing rdf:resource="http://sws.geonames.org
    /1252881/" xmlns:geo="http://www.geonames.org/ontology#"/>
75                  </rdf:Description>
76                  <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#delhi">
77                      <geo:hasSpatialThing rdf:resource="http://sws.geonames.org
    /1273293/" xmlns:geo="http://www.geonames.org/ontology#"/>
78                  </rdf:Description>
79                  <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#india">
```

```
80                      <geo:hasSpatialThing rdf:resource="http://sws.geonames.org
    /1269750/" xmlns:geo="http://www.geonames.org/ontology#"/>
81                  </rdf:Description>
82                   <rdf:Description rdf:about="http://sws.geonames.org/1267254/">
83                      <rdf:type rdf:resource="http://www.geonames.org/ontology#
    Feature"/>
84                      <geo:parentFeature rdf:resource="http://sws.geonames.org
    /6295630/" xmlns:geo="http://www.geonames.org/ontology#"/>
85                      <geo:parentFeature rdf:resource="http://sws.geonames.org
    /6255147/" xmlns:geo="http://www.geonames.org/ontology#"/>
86                      <geo:parentFeature rdf:resource="http://sws.geonames.org
    /1269750/" xmlns:geo="http://www.geonames.org/ontology#"/>
87                      <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#
    ">State of Kerala</rdfs:label>
88                      <geo:name xmlns:geo="http://www.geonames.org/ontology#">State
     of Kerala</geo:name>
89                  </rdf:Description>
90                   <rdf:Description rdf:about="http://sws.geonames.org/1267701/">
91                      <rdf:type rdf:resource="http://www.geonames.org/ontology#
    Feature"/>
92                      <geo:parentFeature rdf:resource="http://sws.geonames.org
    /6295630/" xmlns:geo="http://www.geonames.org/ontology#"/>
93                      <geo:parentFeature rdf:resource="http://sws.geonames.org
    /6255147/" xmlns:geo="http://www.geonames.org/ontology#"/>
94                      <geo:parentFeature rdf:resource="http://sws.geonames.org
    /1269750/" xmlns:geo="http://www.geonames.org/ontology#"/>
95                      <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#
    ">State of KarnÄĄtaka</rdfs:label>
96                      <geo:name xmlns:geo="http://www.geonames.org/ontology#">State
     of KarnÄĄtaka</geo:name>
97                  </rdf:Description>
98                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#sri_lanka">
99                      <geo:hasSpatialThing rdf:resource="http://sws.geonames.org
    /1227603/" xmlns:geo="http://www.geonames.org/ontology#"/>
100                 </rdf:Description>
101                  <rdf:Description rdf:about="http://sws.geonames.org/6255147/">
102                     <rdf:type rdf:resource="http://www.geonames.org/ontology#
    Feature"/>
103                     <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#
    ">Asia</rdfs:label>
104                     <geo:name xmlns:geo="http://www.geonames.org/ontology#">Asia<
    /geo:name>
105                 </rdf:Description>
106                  <rdf:Description rdf:about="http://sws.geonames.org/1252881/">
107                     <rdf:type rdf:resource="http://www.geonames.org/ontology#
    Feature"/>
108                     <geo:parentFeature rdf:resource="http://sws.geonames.org
    /6295630/" xmlns:geo="http://www.geonames.org/ontology#"/>
109                     <geo:parentFeature rdf:resource="http://sws.geonames.org
    /6255147/" xmlns:geo="http://www.geonames.org/ontology#"/>
110                     <geo:parentFeature rdf:resource="http://sws.geonames.org
    /1269750/" xmlns:geo="http://www.geonames.org/ontology#"/>
111                     <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#
    ">State of West Bengal</rdfs:label>
112                     <geo:name xmlns:geo="http://www.geonames.org/ontology#">State
     of West Bengal</geo:name>
113                 </rdf:Description>
114                  <rdf:Description rdf:about="http://sws.geonames.org/1269750/">
115                     <rdf:type rdf:resource="http://www.geonames.org/ontology#
    Feature"/>
116                     <geo:parentFeature rdf:resource="http://sws.geonames.org
    /6295630/" xmlns:geo="http://www.geonames.org/ontology#"/>
117                     <geo:parentFeature rdf:resource="http://sws.geonames.org
    /6255147/" xmlns:geo="http://www.geonames.org/ontology#"/>
118                     <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#
    ">Republic of India</rdfs:label>
119                     <geo:name xmlns:geo="http://www.geonames.org/ontology#">
    Republic of India</geo:name>
120                 </rdf:Description>
121             </rdf:RDF>
122         </data>
123     </annotation>
124     <mediaUnit xsi:type="ns3:Text" uri="http://resource_aeca4252-74fd-427f-a697-0
    ebb0c238e39#1">
```

```
125         <annotation uri="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-a0">
126             <data xmlns:wl="http://weblab.ow2.org/core/1.2/model#" xmlns:resourceContainer="http://weblab.ow2.org/core/1.2/services/resourcecontainer">
127                 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
128                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1">
129                         <dc:language xmlns:dc="http://purl.org/dc/elements/1.1/">en</dc:language>
130                     </rdf:Description>
131                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-a0">
132                         <dcterms:created xmlns:dcterms="http://purl.org/dc/terms/">2014-07-08T15:15:43+0200</dcterms:created>
133                         <wlp:isProducedBy rdf:resource="http://weblab.ow2.org/services#LanguageExtraction" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/processing#"/>
134                     </rdf:Description>
135                 </rdf:RDF>
136             </data>
137         </annotation>
138         <annotation uri="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-a1">
139             <data xmlns:wl="http://weblab.ow2.org/core/1.2/model#" xmlns:resourceContainer="http://weblab.ow2.org/core/1.2/services/resourcecontainer">
140                 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
141                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-2">
142                         <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/instances/Place#midnapore_district" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/processing#"/>
143                     </rdf:Description>
144                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/instances/Place#purulia_district">
145                         <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/processing#">true</wlp:isCandidate>
146                         <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">Purulia district</rdfs:label>
147                         <rdf:type rdf:resource="http://weblab.ow2.org/wookie#Place"/>
148                     </rdf:Description>
149                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/instances/Place#kulabheda_village">
150                         <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/processing#">true</wlp:isCandidate>
151                         <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">Kulabheda village</rdfs:label>
152                         <rdf:type rdf:resource="http://weblab.ow2.org/wookie#Place"/>
153                     </rdf:Description>
154                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/instances/Unit#plus_book_review_business_scitech_nxg_friday_review">
155                         <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/processing#">true</wlp:isCandidate>
156                         <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">Plus Book Review Business SciTech NXg Friday Review</rdfs:label>
157                         <rdf:type rdf:resource="http://weblab.ow2.org/wookie#Unit"/>
158                     </rdf:Description>
159                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-57">
160                         <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/instances/TroopMovementOperation#deployed" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/processing#"/>
161                     </rdf:Description>
162                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/instances/DeathEvent#de7948b6-f4b7-43c1-910c-98ca23f90f73">
163                         <wookie:involves rdf:resource="http://weblab.ow2.org/wookie/instances/Place#west_bengal" xmlns:wookie="http://weblab.ow2.org/wookie#"/>
164                         <wookie:takesPlaceAt rdf:resource="http://weblab.ow2.org/wookie/instances/Place#west_bengal" xmlns:wookie="http://weblab.ow2.org/wookie#"/>
```

```
165                 </rdf:Description>
166                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-43">
167                     <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#bihar" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/
    processing#"/>
168                   </rdf:Description>
169                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#west_bengal">
170                     <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
171                     <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">West Bengal</rdfs:label>
172                     <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    Place"/>
173                   </rdf:Description>
174                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#india">
175                     <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
176                     <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">India</rdfs:label>
177                     <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    Place"/>
178                   </rdf:Description>
179                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-37">
180                     <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#bihar" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/
    processing#"/>
181                   </rdf:Description>
182                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Meeting#visits_hill_region_in_sri_lanka">
183                     <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">visits hill region in Sri Lanka</rdfs:label>
184                     <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    Meeting"/>
185                   </rdf:Description>
186                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/TroopMovementOperation#deployed">
187                     <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">deployed</rdfs:label>
188                     <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    TroopMovementOperation"/>
189                   </rdf:Description>
190                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-15">
191                     <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#bihar" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/
    processing#"/>
192                   </rdf:Description>
193                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-23">
194                     <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Unit#magazine_literary_review_metro_plus_open_page_education"
    xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/processing#"/>
195                   </rdf:Description>
196                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-56">
197                     <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/DamageEvent#damaged" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
198                   </rdf:Description>
199                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-42">
200                     <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#bihar" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/
    processing#"/>
201                   </rdf:Description>
202                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-22">
203                     <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Unit#plus_book_review_business_scitech_nxg_friday_review" xmlns:wlp
    ="http://weblab.ow2.org/core/1.2/ontology/processing#"/>
204                 </rdf:Description>
```

```
205                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-36">
206                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#new_delhi" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology
    /processing#"/>
207                     </rdf:Description>
208                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Unit#mps_team">
209                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
210                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">MPs team</rdfs:label>
211                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#Unit
    "/>
212                     </rdf:Description>
213                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Unit#hindu_group">
214                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
215                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Hindu Group</rdfs:label>
216                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#Unit
    "/>
217                     </rdf:Description>
218                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#sri_lanka">
219                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
220                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Sri Lanka</rdfs:label>
221                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    Place"/>
222                     </rdf:Description>
223                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-14">
224                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#west_bengal" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
225                     </rdf:Description>
226                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-45">
227                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Unit#police" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/
    processing#"/>
228                     </rdf:Description>
229                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-55">
230                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Meeting#visits_hill_region_in_sri_lanka" xmlns:wlp="http://weblab.
    ow2.org/core/1.2/ontology/processing#"/>
231                     </rdf:Description>
232                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#karnataka">
233                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
234                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Karnataka</rdfs:label>
235                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    Place"/>
236                     </rdf:Description>
237                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-4">
238                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#ergada_village" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
239                     </rdf:Description>
240                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-9">
241                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#giridih_district" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
242                     </rdf:Description>
243                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#bihar">
244                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
```

```
245                         <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Bihar</rdfs:label>
246                         <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    Place"/>
247                 </rdf:Description>
248                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-35">
249                         <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#pakistan" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/
    processing#"/>
250                 </rdf:Description>
251                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#ergada_village">
252                         <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
253                         <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Ergada village</rdfs:label>
254                         <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    Place"/>
255                 </rdf:Description>
256                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/TroopMovementOperation#86ac1ebe-6ef3-4a88-a877-de8fe489e2c7">
257                         <wookie:involves rdf:resource="http://weblab.ow2.org/
    wookie/instances/Unit#police" xmlns:wookie="http://weblab.ow2.org/wookie#"/>
258                 </rdf:Description>
259                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-17">
260                         <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#west_bengal" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
261                 </rdf:Description>
262                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-25">
263                         <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Unit#lahore_court" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
264                 </rdf:Description>
265                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Unit#quest">
266                         <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
267                         <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Quest</rdfs:label>
268                         <rdf:type rdf:resource="http://weblab.ow2.org/wookie#Unit
    "/>
269                 </rdf:Description>
270                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-54">
271                         <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#west_bengal" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
272                 </rdf:Description>
273                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#kerala">
274                         <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
275                         <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Kerala</rdfs:label>
276                         <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    Place"/>
277                 </rdf:Description>
278                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Unit#jharkhand_party_member">
279                         <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
280                         <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Jharkhand Party member</rdfs:label>
281                         <rdf:type rdf:resource="http://weblab.ow2.org/wookie#Unit
    "/>
282                 </rdf:Description>
283                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-3">
284                         <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#purulia_district" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
285                 </rdf:Description>
```

```
286                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-44">
287                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Person#pankaj_kamboj" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
288                     </rdf:Description>
289                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/KidnappingEvent#cb6d501a-1cb7-4223-ae30-7a9d8c0f77ef">
290                       <wookie:involves rdf:resource="http://weblab.ow2.org/
    wookie/instances/Place#ergada_village" xmlns:wookie="http://weblab.ow2.org/
    wookie#"/>
291                       <wookie:involves rdf:resource="http://weblab.ow2.org/
    wookie/instances/Unit#jharkhand_party_member_kanai_murmu" xmlns:wookie="http:
    //weblab.ow2.org/wookie#"/>
292                       <wookie:takesPlaceAt rdf:resource="http://weblab.ow2.org/
    wookie/instances/Place#ergada_village" xmlns:wookie="http://weblab.ow2.org/
    wookie#"/>
293                       <wookie:date xmlns:wookie="http://weblab.ow2.org/wookie#"
    >Monday</wookie:date>
294                     </rdf:Description>
295                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-34">
296                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#delhi" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/
    processing#"/>
297                     </rdf:Description>
298                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-24">
299                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Unit#mps_team" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/
    processing#"/>
300                     </rdf:Description>
301                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-16">
302                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#west_bengal" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
303                     </rdf:Description>
304                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Unit#communist_party_of_india">
305                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
306                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Communist Party of India</rdfs:label>
307                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#Unit
    "/>
308                     </rdf:Description>
309                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Unit#magazine_literary_review_metro_plus_open_page_education">
310                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
311                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Magazine Literary Review Metro Plus Open Page Education</
    rdfs:label>
312                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#Unit
    "/>
313                     </rdf:Description>
314                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-26">
315                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Unit#union_government" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
316                     </rdf:Description>
317                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/DeathEvent#killed">
318                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">killed</rdfs:label>
319                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    DeathEvent"/>
320                     </rdf:Description>
321                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-10">
322                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Unit#business_line" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
323                     </rdf:Description>
```

```
324                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Unit#jharkhand_party_member_kanai_murmu">
325                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
326                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Jharkhand Party member Kanai Murmu</rdfs:label>
327                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#Unit
    "/>
328                   </rdf:Description>
329                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-6">
330                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#bokaro_district" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
331                   </rdf:Description>
332                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Unit#union_government">
333                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
334                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Union government</rdfs:label>
335                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#Unit
    "/>
336                   </rdf:Description>
337                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-47">
338                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Person#ravi_kant_dhan" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
339                   </rdf:Description>
340                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#giridih_district">
341                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
342                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Giridih district</rdfs:label>
343                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    Place"/>
344                   </rdf:Description>
345                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-33">
346                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#delhi" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/
    processing#"/>
347                   </rdf:Description>
348                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#new_delhi">
349                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
350                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">New Delhi</rdfs:label>
351                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    Place"/>
352                   </rdf:Description>
353                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-19">
354                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Unit#maobadi_protirodh_committee" xmlns:wlp="http://weblab.ow2.org/
    core/1.2/ontology/processing#"/>
355                   </rdf:Description>
356                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Person#pankaj_kamboj">
357                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
358                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Pankaj Kamboj</rdfs:label>
359                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    Person"/>
360                   </rdf:Description>
361                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#chanda_village">
362                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
363                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Chanda village</rdfs:label>
```

```
364                         <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    Place"/>
365                 </rdf:Description>
366                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-53">
367                         <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#chanda_village" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
368                 </rdf:Description>
369                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Unit#subscription_group_sites">
370                         <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
371                         <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Subscription    Group Sites</rdfs:label>
372                         <rdf:type rdf:resource="http://weblab.ow2.org/wookie#Unit
    "/>
373                 </rdf:Description>
374                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-27">
375                         <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Unit#jharkhand_party_member" xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#"/>
376                 </rdf:Description>
377                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-11">
378                         <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Unit#business_line_epaper" xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#"/>
379                 </rdf:Description>
380                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Unit#dhanbad_rail_division_amrendra_das">
381                         <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
382                         <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Dhanbad Rail Division Amrendra Das</rdfs:label>
383                         <rdf:type rdf:resource="http://weblab.ow2.org/wookie#Unit
    "/>
384                 </rdf:Description>
385                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Meeting#4003bba8-f3a7-4287-9471-55704359e93a">
386                         <wookie:takesPlaceAt rdf:resource="http://weblab.ow2.org/
    wookie/instances/Place#sri_lanka" xmlns:wookie="http://weblab.ow2.org/wookie#
    "/>
387                 </rdf:Description>
388                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-5">
389                         <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#kulabheda_village" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
390                 </rdf:Description>
391                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-46">
392                         <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Person#singh_laya" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
393                 </rdf:Description>
394                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Unit#india_s_national_newspaper_tuesday">
395                         <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
396                         <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">India's National Newspaper Tuesday</rdfs:label>
397                         <rdf:type rdf:resource="http://weblab.ow2.org/wookie#Unit
    "/>
398                 </rdf:Description>
399                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-18">
400                         <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Unit#dhanbad_rail_division_amrendra_das" xmlns:wlp="http://weblab.
    ow2.org/core/1.2/ontology/processing#"/>
401                 </rdf:Description>
402                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-32">
403                         <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#india" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/
```

```
processing#"/>
404                 </rdf:Description>
405                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
instances/Unit#lahore_court">
406                     <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
/1.2/ontology/processing#">true</wlp:isCandidate>
407                     <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
schema#">Lahore court</rdfs:label>
408                     <rdf:type rdf:resource="http://weblab.ow2.org/wookie#Unit
"/>
409                 </rdf:Description>
410                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
instances/Person#singh_laya">
411                     <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
/1.2/ontology/processing#">true</wlp:isCandidate>
412                     <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
schema#">Singh Laya</rdfs:label>
413                     <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
Person"/>
414                 </rdf:Description>
415                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
instances/Place#bokaro_district">
416                     <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
/1.2/ontology/processing#">true</wlp:isCandidate>
417                     <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
schema#">Bokaro district</rdfs:label>
418                     <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
Place"/>
419                 </rdf:Description>
420                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
instances/Unit#police">
421                     <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
/1.2/ontology/processing#">true</wlp:isCandidate>
422                     <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
schema#">Police</rdfs:label>
423                     <rdf:type rdf:resource="http://weblab.ow2.org/wookie#Unit
"/>
424                 </rdf:Description>
425                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
f-a697-0ebb0c238e39#1-52">
426                     <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
instances/Place#sri_lanka" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology
/processing#"/>
427                 </rdf:Description>
428                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
f-a697-0ebb0c238e39#1-12">
429                     <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
instances/Unit#hindu_group" xmlns:wlp="http://weblab.ow2.org/core/1.2/
ontology/processing#"/>
430                 </rdf:Description>
431                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
f-a697-0ebb0c238e39#1-28">
432                     <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
instances/Unit#india_s_national_newspaper_tuesday" xmlns:wlp="http://weblab.
ow2.org/core/1.2/ontology/processing#"/>
433                 </rdf:Description>
434                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
f-a697-0ebb0c238e39#1-59">
435                     <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
instances/DeathEvent#killed" xmlns:wlp="http://weblab.ow2.org/core/1.2/
ontology/processing#"/>
436                 </rdf:Description>
437                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
instances/KidnappingEvent#kidnapped">
438                     <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
schema#">kidnapped</rdfs:label>
439                     <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
KidnappingEvent"/>
440                 </rdf:Description>
441                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
f-a697-0ebb0c238e39#1-60">
442                     <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
instances/DeathEvent#killed" xmlns:wlp="http://weblab.ow2.org/core/1.2/
ontology/processing#"/>
443                 </rdf:Description>
```

```
444                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-8">
445                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#west_bengal" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
446                     </rdf:Description>
447                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-38">
448                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#karnataka" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology
    /processing#"/>
449                     </rdf:Description>
450                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/DamageEvent#d845e46f-37c7-4e28-aff8-e7221d16acdc">
451                       <wookie:takesPlaceAt rdf:resource="http://weblab.ow2.org/
    wookie/instances/Place#bokaro_district" xmlns:wookie="http://weblab.ow2.org/
    wookie#"/>
452                     </rdf:Description>
453                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-49">
454                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Unit#quest" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/
    processing#"/>
455                     </rdf:Description>
456                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#delhi">
457                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
458                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Delhi</rdfs:label>
459                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    Place"/>
460                     </rdf:Description>
461                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-31">
462                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#pakistan" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/
    processing#"/>
463                     </rdf:Description>
464                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#pakistan">
465                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
466                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Pakistan</rdfs:label>
467                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    Place"/>
468                     </rdf:Description>
469                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Person#ravi_kant_dhan">
470                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
471                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Ravi Kant Dhan</rdfs:label>
472                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    Person"/>
473                     </rdf:Description>
474                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-41">
475                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#kerala" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/
    processing#"/>
476                     </rdf:Description>
477                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/DamageEvent#damaged">
478                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">damaged</rdfs:label>
479                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    DamageEvent"/>
480                     </rdf:Description>
481                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-51">
482                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Unit#police" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/
    processing#"/>
483                     </rdf:Description>
```

```
484                     <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Unit#business_line">
485                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
486                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Business Line</rdfs:label>
487                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#Unit
    "/>
488                     </rdf:Description>
489                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-21">
490                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Unit#communist_party_of_india" xmlns:wlp="http://weblab.ow2.org/
    core/1.2/ontology/processing#"/>
491                     </rdf:Description>
492                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-29">
493                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#bihar" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/
    processing#"/>
494                     </rdf:Description>
495                       <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#salaiya_village">
496                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
497                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Salaiya village</rdfs:label>
498                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    Place"/>
499                     </rdf:Description>
500                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-13">
501                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Unit#subscription_group_sites" xmlns:wlp="http://weblab.ow2.org/
    core/1.2/ontology/processing#"/>
502                     </rdf:Description>
503                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-58">
504                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/KidnappingEvent#kidnapped" xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#"/>
505                     </rdf:Description>
506                       <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Place#midnapore_district">
507                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
508                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Midnapore district</rdfs:label>
509                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#
    Place"/>
510                     </rdf:Description>
511                     <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-48">
512                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Unit#police" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/
    processing#"/>
513                     </rdf:Description>
514                       <rdf:Description rdf:about="http://weblab.ow2.org/wookie/
    instances/Unit#maobadi_protirodh_committee">
515                       <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core
    /1.2/ontology/processing#">true</wlp:isCandidate>
516                       <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-
    schema#">Maobadi Protirodh Committee</rdfs:label>
517                       <rdf:type rdf:resource="http://weblab.ow2.org/wookie#Unit
    "/>
518                     </rdf:Description>
519                       <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-7">
520                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#salaiya_village" xmlns:wlp="http://weblab.ow2.org/core/1.2/
    ontology/processing#"/>
521                     </rdf:Description>
522                       <rdf:Description rdf:about="http://resource_aeca4252-74fd-427
    f-a697-0ebb0c238e39#1-39">
523                       <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/
    instances/Place#kerala" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/
```

```
524         processing#"/>
525                 </rdf:Description>
526                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/instances/DeathEvent#25ea8fc1-9848-46dd-a7b5-cc2d75bf7852">
527                     <wookie:involves rdf:resource="http://weblab.ow2.org/wookie/instances/Place#west_bengal" xmlns:wookie="http://weblab.ow2.org/wookie#"/>
528                     <wookie:involves rdf:resource="http://weblab.ow2.org/wookie/instances/Unit#jharkhand_party_member" xmlns:wookie="http://weblab.ow2.org/wookie#"/>
529                     <wookie:takesPlaceAt rdf:resource="http://weblab.ow2.org/wookie/instances/Place#west_bengal" xmlns:wookie="http://weblab.ow2.org/wookie#"/>
530                     <wookie:date xmlns:wookie="http://weblab.ow2.org/wookie#">Monday</wookie:date>
531                   </rdf:Description>
532                   <rdf:Description rdf:about="http://weblab.ow2.org/wookie/instances/Unit#business_line_epaper">
533                     <wlp:isCandidate xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/processing#">true</wlp:isCandidate>
534                     <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">Business Line ePaper</rdfs:label>
535                     <rdf:type rdf:resource="http://weblab.ow2.org/wookie#Unit"/>
536                   </rdf:Description>
537                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-40">
538                     <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/instances/Place#karnataka" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/processing#"/>
539                   </rdf:Description>
540                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-30">
541                     <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/instances/Place#new_delhi" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/processing#"/>
542                   </rdf:Description>
543                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-20">
544                     <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/instances/Unit#jharkhand_party_member_kanai_murmu" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/processing#"/>
545                   </rdf:Description>
546                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-50">
547                     <wlp:refersTo rdf:resource="http://weblab.ow2.org/wookie/instances/Unit#police" xmlns:wlp="http://weblab.ow2.org/core/1.2/ontology/processing#"/>
548                   </rdf:Description>
549               </rdf:RDF>
550           </data>
551       </annotation>
552       <annotation uri="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-a61">
553           <data xmlns:wl="http://weblab.ow2.org/core/1.2/model#" xmlns:resourceContainer="http://weblab.ow2.org/core/1.2/services/resourcecontainer">
554               <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
555                   <rdf:Description rdf:about="http://resource_aeca4252-74fd-427f-a697-0ebb0c238e39">
556                     <dct:spatial xmlns:dct="http://purl.org/dc/terms/">20.0,77.0</dct:spatial>
557                     <dct:spatial xmlns:dct="http://purl.org/dc/terms/">10.0,76.5</dct:spatial>
558                     <dct:spatial xmlns:dct="http://purl.org/dc/terms/">7.0,81.0</dct:spatial>
559                     <dct:spatial xmlns:dct="http://purl.org/dc/terms/">24.0,88.0</dct:spatial>
560                     <dct:spatial xmlns:dct="http://purl.org/dc/terms/">13.5,76.0</dct:spatial>
561                     <dct:spatial xmlns:dct="http://purl.org/dc/terms/">28.6667,77.1</dct:spatial>
562                   </rdf:Description>
563               </rdf:RDF>
564           </data>
       </annotation>
```

```
565         <segment xsi:type="ns3:LinearSegment" start="18" end="52" uri="http://
        resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-28"/>
566         <segment xsi:type="ns3:LinearSegment" start="248" end="257" uri="http://
        resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-38"/>
567         <segment xsi:type="ns3:LinearSegment" start="261" end="267" uri="http://
        resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-39"/>
568         <segment xsi:type="ns3:LinearSegment" start="271" end="280" uri="http://
        resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-36"/>
569         <segment xsi:type="ns3:LinearSegment" start="502" end="513" uri="http://
        resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-16"/>
570         <segment xsi:type="ns3:LinearSegment" start="526" end="531" uri="http://
        resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-37"/>
571         <segment xsi:type="ns3:LinearSegment" start="626" end="632" uri="http://
        resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-60"/>
572         <segment xsi:type="ns3:LinearSegment" start="654" end="665" uri="http://
        resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-17"/>
573         <segment xsi:type="ns3:LinearSegment" start="813" end="835" uri="http://
        resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-27"/>
574         <segment xsi:type="ns3:LinearSegment" start="840" end="846" uri="http://
        resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-59"/>
575         <segment xsi:type="ns3:LinearSegment" start="850" end="861" uri="http://
        resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-14"/>
576         <segment xsi:type="ns3:LinearSegment" start="965" end="970" uri="http://
        resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-15"/>
577         <segment xsi:type="ns3:LinearSegment" start="1004" end="1020" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-26"/>
578         <segment xsi:type="ns3:LinearSegment" start="1196" end="1220" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-21"/>
579         <segment xsi:type="ns3:LinearSegment" start="1246" end="1280" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-20"/>
580         <segment xsi:type="ns3:LinearSegment" start="1290" end="1299" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-58"/>
581         <segment xsi:type="ns3:LinearSegment" start="1309" end="1323" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-4"/>
582         <segment xsi:type="ns3:LinearSegment" start="1486" end="1503" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-5"/>
583         <segment xsi:type="ns3:LinearSegment" start="1516" end="1534" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-2"/>
584         <segment xsi:type="ns3:LinearSegment" start="1591" end="1618" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-19"/>
585         <segment xsi:type="ns3:LinearSegment" start="1636" end="1652" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-3"/>
586         <segment xsi:type="ns3:LinearSegment" start="1656" end="1667" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-8"/>
587         <segment xsi:type="ns3:LinearSegment" start="1838" end="1848" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-46"/>
588         <segment xsi:type="ns3:LinearSegment" start="2122" end="2156" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-18"/>
589         <segment xsi:type="ns3:LinearSegment" start="2307" end="2323" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-9"/>
590         <segment xsi:type="ns3:LinearSegment" start="2527" end="2533" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-45"/>
591         <segment xsi:type="ns3:LinearSegment" start="2534" end="2548" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-47"/>
592         <segment xsi:type="ns3:LinearSegment" start="2586" end="2592" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-50"/>
593         <segment xsi:type="ns3:LinearSegment" start="2593" end="2606" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-44"/>
594         <segment xsi:type="ns3:LinearSegment" start="2630" end="2637" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-56"/>
595         <segment xsi:type="ns3:LinearSegment" start="2683" end="2698" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-6"/>
596         <segment xsi:type="ns3:LinearSegment" start="2705" end="2710" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-42"/>
597         <segment xsi:type="ns3:LinearSegment" start="2749" end="2764" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-7"/>
598         <segment xsi:type="ns3:LinearSegment" start="2818" end="2832" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-53"/>
599         <segment xsi:type="ns3:LinearSegment" start="2971" end="2976" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-43"/>
600         <segment xsi:type="ns3:LinearSegment" start="3008" end="3014" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-51"/>
601         <segment xsi:type="ns3:LinearSegment" start="3310" end="3319" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-40"/>
602         <segment xsi:type="ns3:LinearSegment" start="3323" end="3329" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-41"/>
```

```
603         <segment xsi:type="ns3:LinearSegment" start="3333" end="3342" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-30"/>
604         <segment xsi:type="ns3:LinearSegment" start="3634" end="3645" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-54"/>
605         <segment xsi:type="ns3:LinearSegment" start="3658" end="3663" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-29"/>
606         <segment xsi:type="ns3:LinearSegment" start="3927" end="3939" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-25"/>
607         <segment xsi:type="ns3:LinearSegment" start="3975" end="3983" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-24"/>
608         <segment xsi:type="ns3:LinearSegment" start="3984" end="4015" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-55"/>
609         <segment xsi:type="ns3:LinearSegment" start="4006" end="4015" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-52"/>
610         <segment xsi:type="ns3:LinearSegment" start="4057" end="4062" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-32"/>
611         <segment xsi:type="ns3:LinearSegment" start="4092" end="4100" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-31"/>
612         <segment xsi:type="ns3:LinearSegment" start="4456" end="4461" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-34"/>
613         <segment xsi:type="ns3:LinearSegment" start="4533" end="4539" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-48"/>
614         <segment xsi:type="ns3:LinearSegment" start="4540" end="4548" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-57"/>
615         <segment xsi:type="ns3:LinearSegment" start="4563" end="4568" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-33"/>
616         <segment xsi:type="ns3:LinearSegment" start="4605" end="4613" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-35"/>
617         <segment xsi:type="ns3:LinearSegment" start="4851" end="4910" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-23"/>
618         <segment xsi:type="ns3:LinearSegment" start="4911" end="4967" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-22"/>
619         <segment xsi:type="ns3:LinearSegment" start="5010" end="5015" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-49"/>
620         <segment xsi:type="ns3:LinearSegment" start="5052" end="5063" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-12"/>
621         <segment xsi:type="ns3:LinearSegment" start="5124" end="5151" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-13"/>
622         <segment xsi:type="ns3:LinearSegment" start="5188" end="5201" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-10"/>
623         <segment xsi:type="ns3:LinearSegment" start="5205" end="5225" uri="http:
        //resource_aeca4252-74fd-427f-a697-0ebb0c238e39#1-11"/>
624         <content>Online edition of India's National Newspaper Tuesday, Oct 13,
        2009 ePaper  | Mobile/PDA Version
625
626  Front Page
627
628
629
630         News:  ePaper  | Front Page  | National  | Tamil Nadu  | Andhra Pradesh
        | Karnataka  | Kerala  | New Delhi  | Other States  | International  |
        Opinion  | Business  | Sport  | Miscellaneous  | Engagements  | Advts:
        Retail Plus  | Classifieds  | Jobs  | Obituary  |
631
632
633 Front Page      Maoists unleash violence in West Bengal, Jharkhand, Bihar
634
635     Trucks set ablaze, telecom tower blasted, rail track blown up
636
637 One killed, another abducted in West Bengal
638
639 Attacks occur on the first day of the two-day shutdown
640
641  Midnapore/Ranchi/Patna: Maoists on Monday unleashed a wave of attacks in which a
         Jharkhand Party member was killed in West Bengal. They blew up rail tracks
        and set trucks ablaze in Jharkhand and blasted a telecommunications tower in
        Bihar.
642
643  The attacks come even as the Union government is all set to launch a massive
        offensive against Left wing extremism in the naxalite-hit States
644
645  The violence occurred on the first day of a two-day shutdown called by the
        Communist Party of India (Maoist).
646
647  The body of Jharkhand Party member Kanai Murmu, who was kidnapped from his
        Ergada village in the early hours of Monday, was found later, the police said
```

.

Informed sources said another member of the party, Ananda Mahato, was abducted by Maoists from Kulabheda village in the same Midnapore district. Both Murmu and Mahato were members of the anti-Maoist "Maobadi Protirodh Committee." At Ghatbera in Purulia district of West Bengal, Maoists ransacked the house of a local CPI(Marxist) leader and shot and injured a member of the village resistance committee.

As the attackers failed to find Chandan Singh Laya, CPI(Marxist) Ghatbera local committee secretary, they set his house ablaze.

In Jharkhand, Maoists blew up rail tracks at Jharandih in Dhanbad, resulting in the Shaktipunj Express and some local trains being held up at various points, Senior Public Relations Officer of Dhanbad Rail Division Amrendra Das said adding a light engine derailed after a one-and-half metre section of the track was blown up.

A group of 12 Maoists set three trucks ablaze in Giridih district's Isri area and blocked the Dumri-Giridih road with felled trees. Maoists also partially blasted a road bridge connecting Dumri to the Grand Trunk Road and gunshots were heard, Giridih Superintendent of Police Ravi Kant Dhan said.

Hazaribagh Superintendent of Police Pankaj Kamboj said Maoists partially damaged a road with explosives at Sardalo, bordering Bokaro district.

In Bihar, Maoists blasted the telecom tower at Salaiya village. Maoists also dug up a 15-metre stretch of a road at Chanda village, disrupting traffic between Deo and Dhibra. They left behind pamphlets, claiming responsibility for the action, official sources said.

Bihar Additional Director-General of Police (Headquarters) Neelmani said tight security arrangements were in place to counter the Maoist bandh. – PTI

```
702
703 Directive to follow panel guidelines
704
705 Pakistani diplomat electrocuted
706
707 Kidnapped Delhi child found dead in relative's car
708
709 Petition filed against Nitish
710
711 Police deployed in Kaithal
712
713 Delhi can deliver great Games: Fennell
714
715 Pakistan diplomat dies of electric shock
716
717 Worker dies after inhaling poisonous gases
718
719 10 killed in UP floods
720
721 Fennell confident
722
723 Cases withdrawn
724
725 One-way soon at Lakdikapul
726
727 Archives Yesterday's Issue  Datewise
728
729  Features:  Magazine  Literary Review  Metro Plus  Open Page  Education Plus
       Book Review  Business  SciTech  NXg  Friday Review  Cinema Plus  Young World
       Property Plus  Quest
730
731  The Hindu Group:  Home  | About Us  | Copyright  | Archives  | Contacts  |
       Subscription  Group Sites:  The Hindu  | The Hindu ePaper  | Business Line
       | Business Line ePaper  | Sportstar  | Frontline  | Publications  | eBooks
       | Images  | Ergo  | Home  |
732   Copyright Â© 2009, The Hindu. Republication or redissemination of the contents
       of this screen are expressly prohibited without the written consent of The
       Hindu</content>
733     </mediaUnit>
734 </resource>
```

Listing E.1 – An example of a full WebLab resource.

Annexe F

Résumé en français

F.1 Introduction

Imaginons une usine qui produit des voitures avec une roue sur le toit, une sur le côté, et deux qui touchent le sol. Les clients de cette usine étrange doivent changer les roues à chaque fois qu'ils achètent une voiture. Cela peut sembler improbable, mais c'est le cas avec certains systèmes d'Extraction d'Information (EI). D'énormes quantités de données sont traitées, et le système tente de fournir une synthèse des informations utiles. Des failles dans l'extraction de cette information rendent la synthèse inexacte, et c'est à l'utilisateur de la corriger.

L'usine se rend compte que les clients ne sont pas contents, donc elle met à leur disposition un expert qui peut produire des voitures sur demande. Le client n'a qu'à trouver le temps de venir dans l'usine pour discuter de ses besoins précis. D'une façon similaire, le système EI peut être configuré par un expert en consultation avec l'utilisateur (en espérant que l'utilisateur ne change pas d'avis une fois qu'il voit les résultats).

L'usine décide d'améliorer la satisfaction des utilisateurs, qui ont maintenant des outils leur permettant de changer la configuration de l'usine eux-mêmes après une formation en assemblage de voitures. Les utilisateurs des systèmes EI ont des langages développés spécifiquement qui leur permettent d'interagir directement avec la chaîne, mais il peut leur manquer l'expertise pour apprécier les conséquences de leurs actions.

L'usine prend une décision radicale. Elle crée un département dédié aux retours utilisateur. Elle fait des sondages, demandant aux utilisateurs de noter les caractéristiques de leurs voitures de 1 à 1000, ou de les trier en ordre de préférence. Elle utilise ensuite ces questionnaires pour construire des voitures fabriquées individuellement pour chaque client. Nous pouvons voir que l'obtention des retours utilisateur pourrait aider à améliorer les services EI, mais est-ce vraiment naturel de demander qu'ils fournissent ces retours explicitement et de façon numérique ?

Enfin, l'usine installe un dispositif de contrôle continu dans la voiture. Cet outil fait automatiquement des retours au département sur l'utilisation

exacte de la voiture, les modifications effectuées, la couleur dans laquelle le client choisit de repeindre sa voiture, quelles caractéristiques lui plaisent le plus, *etc.* Enfin, elle peut produire des voitures uniques qui plaisent aux clients, sans avoir besoin de les interroger, les former, ou demander qu'ils remplissent de longs formulaires.

Notre objectif est d'installer un tel dispositif de contrôle continu dans la chaîne EI. Il observera les actions de l'utilisateur, et n'aura besoin que de retours intuitifs et naturels pour améliorer la chaîne (mais bien sûr, il pourra aussi prendre en compte des retours numériques).

F.1.1 Contexte

Notre travail est motivé par les besoins réels de la communauté *Renseignement d'Origine Source Ouverte* (ROSO). Par le passé, la tâche des analystes des services de renseignement a été de chercher des informations cachées. Maintenant, ils font face à un volume toujours croissant de pages *web*, ainsi que d'autres documents d'origine source ouverte, et ils doivent extraire de l'information utile et pertinente à partir de cette profusion de données. Des chaînes de traitement spécialisées ont été développées pour faciliter cette tâche (par exemple, Ogrodniczuk and Przepiórkowski (2010) donnent une synthèse de quelques chaînes fournies en tant que services *Web*).

Nous nous intéressons au problème générique de l'amélioration continue d'une telle chaîne de traitement de documents, plus précisément d'extraction d'événements d'intérêt pour la veille et le renseignement. Dans cette application, des documents provenant essentiellement du *Web* sont fournis en continu à une chaîne de traitement, qui vise à extraire des événements (par exemple, une attaque terroriste) et leurs caractéristiques (date, lieu, acteurs, *etc.*) pour les intégrer dans une base de données.

Dans de telles applications, il est clair que l'extraction ne peut pas être parfaite. Tout d'abord, du fait de la grande diversité de documents sur le *Web*, il ne peut pas exister une unique chaîne optimale pour tous les documents. Ensuite, on peut imaginer, par exemple, qu'une dépêche relatant « le *bombardement*, par des ions, d'une *cible* d'or dans une animation autour de la physique *atomique* lors d'une *manifestation* pour la fête de la science », puisse induire une chaîne de traitement en erreur, et lui fasse insérer dans la base un attentat à l'arme atomique. Par ailleurs, la volonté de traiter des documents provenant du monde entier entraîne le besoin de traiter des documents dans des langues très diverses, pour lesquelles des dictionnaires peuvent être de qualité très variable. Pour toutes ces raisons, les analystes perdent du temps dans la tâche lourde et répétitive de la correction *a poste-*

riori des événements.

L'application qui nous intéresse utilise une chaîne de traitement, définie par des experts, consistant en un enchaînement figé (mais potentiellement conditionnel) de traitements atomiques, tels que la détection de la langue ou du format, la traduction, la détection d'événements en utilisant des mots ou verbes déclencheurs, *etc.* Actuellement, aucun mécanisme ne permet d'améliorer la chaîne au fil du temps sans l'intervention d'un expert, en consultation avec les analystes, pour recalibrer la chaîne.

Notre objectif est de combler ce manque, en fournissant un mécanisme automatique d'amélioration continue de la chaîne de traitement, tirant parti des retours (ou *feedback*) exprimés implicitement par les analystes lorsqu'ils consultent la sortie du système. Il s'agit de faire en sorte que la chaîne « apprenne de ses erreurs ». Par exemple, si le document est un article scientifique en français, il peut être préférable de le traduire d'abord en anglais, pour profiter du plus grand choix de règles d'extraction pour l'anglais, puis d'extraire les événements. En revanche, pour un *blog* sur lequel l'argot est abondamment utilisé, une extraction directe peut être préférable, car les dictionnaires sont insuffisants pour l'argot, et une traduction ne fait qu'ajouter du bruit. Ainsi, les services, ou briques pour la construction de la chaîne, restent des « boîtes noires » permettant à l'analyste de se distancier du processus d'extraction.

Nous devons récupérer le *feedback* de manière invisible pour l'analyste, sans impact sur son travail. Bratko and Šuc (2003) montrent que l'expertise de l'utilisateur (son *feedback*) est manifestée par les traces de ses actions. Ces traces pourraient être captées à travers l'interface graphique qui donne le détail des événements en synthèse, et le *feedback* basé sur les interactions analyste-système actuelles. Par exemple, d'un événement *corrigé*, nous pouvons tirer un *feedback* explicite, via une distance entre l'événement idéal (corrigé) et celui qui avait été extrait. Nos premières expériences simulent ceci : l'agent reçoit un jugement sur la qualité de ses extractions sous la forme d'une distance, et donc d'un *feedback* numérique.

Mais l'événement peut aussi être *non extrait*, *correct mais extrait par un traitement trop lent*, *etc.* Prendre en compte ces différents aspects donne lieu à des questions, telles que « Est-il préférable d'extraire un événement incorrect, ou de manquer une extraction ? La qualité est-elle plus importante que la vitesse, ou l'inverse ? ». Il n'est pas évident de répondre à de telles questions avec une réponse numérique, par exemple, de dire qu'une extraction partielle mais rapide vaut 10.5 fois moins qu'une extraction complète mais lente. Il est essentiel, alors, que nous progressions vers un *feedback* intuitif, basé sur des préférences que les utilisateurs puissent exprimer naturellement,

telles que « Je préfère une extraction incorrecte à une extraction manquée », ou encore « Je préfère que cela soit aussi rapide que possible ». Les besoins peuvent alors être spécifiés et changés très facilement, de façon intuitive.

F.1.2 Contribution

Nous formalisons notre problème comme un problème d'apprentissage par renforcement, et la chaîne de traitement elle-même comme un processus de décision markovien (MDP, Définition 2). Nous étudions trois cadres différents. Dans le premier, l'agent reçoit des retours *numériques* sur la qualité du traitement, ce qui est très informatif, mais difficile à calibrer. Les deux autres cadres utilisent des retours *intuitifs* (pour l'utilisateur), avec une formalisation partiellement qualitative pour le deuxième, et totalement qualitative pour le troisième. De tels retours sont moins informatifs pour l'agent, mais requièrent beaucoup moins, voire pas du tout de calibrage.

Nous rendons compte de la plateforme que nous avons construite, appelée *BIMBO (Benefiting from Intelligent and Measurable Behaviour Optimisation)*. Dans cette plateforme, on peut « brancher » différents algorithmes d'apprentissage par renforcement, différents modèles de la tâche, différents services *web*, différents types de retours de l'utilisateur, *etc.*, ce qui nous permet de mesurer l'impact de ces éléments sur l'apprentissage. Nous rendons ensuite compte de l'utilisation de cette plateforme pour l'application industrielle qui nous intéresse, c'est-à-dire pour l'amélioration continue d'une chaîne de traitement qui extrait des événements à partir de pages *web* et d'autres documents de sources ouvertes. La plateforme sous-tend notre application, mais peut également être utilisée pour d'autres applications, et/ou pour l'étude de différentes méthodes d'amélioration continue de chaînes de traitement. Nous décrivons brièvement, en conclusion, son utilisation pour des tâches de segmentation d'images et de reconnaissance optique de caractères.

Nous présentons ensuite des résultats expérimentaux, tout d'abord avec des retours numériques (simulés) des utilisateurs (section F.7). Les résultats montrent qu'apprendre automatiquement à enchaîner et paramétrer les services est totalement faisable en pratique. Nous présentons enfin des résultats expérimentaux avec des retours *intuitifs* (section F.8). Là encore, et malgré l'information bien moins précise véhiculée par de tels retours, les résultats montrent que l'apprentissage reste totalement faisable en pratique, et ce, en demandant très peu d'effort de calibrage de la part des analystes.

Notre approche, aussi, peut être utilisée pour d'autres traitements, par exemple nous l'avons appliqué, en utilisant la plateforme *BIMBO* et une va-

riation de Q-Learning(λ), *Dora* (Nicart et al., 2016), aux tâches de segmentation (le pré-traitement de l'image) et de la *ROC* (reconnaissance optique de caractères). Notre but est la détection des éléments d'intérêt, pour suggérer automatiquement des titres depuis les pages de garde des documents.

F.1.3 Travaux connexes

À notre connaissance, il n'y a pas eu de travaux académiques s'intéressant à des agents capables de construire dynamiquement une chaîne de services, et d'apprendre et de s'améliorer en continu en utilisant des retours intuitifs des utilisateurs. Toutefois, de nombreux ingrédients ont été étudiés et proposés pour de telles applications.

L'enchaînement de services pour réaliser un but précis, par exemple, ne constitue pas une idée nouvelle. Des services peuvent être composés, par exemple, pour aider l'utilisateur à remplir des formulaires (Saïs et al., 2013). Dans une autre application, les caractéristiques physiques de documents sont utilisées avec succès pour construire, en temps réel, des chaînes adaptatives et modulaires pour des photocopieurs Xerox (Fromherz et al., 2003). Toutefois, les entrées et sorties de chaque service sont connues à l'avance, et enchaîner ces services constitue donc un problème de planification, sans apprentissage. Par ailleurs, les préférences de l'utilisateur ne sont pas prises en compte, et aucun retour n'est utilisé pour améliorer le processus d'une fois sur l'autre.

De façon générale, un utilisateur peut reconfigurer une chaîne en utilisant un langage dédié, comme ReCooPLa (Rodrigues et al., 2015), ou en choisissant un service dans un répertoire de services du même type (Doucy et al., 2008). Néanmoins, une chaîne ainsi reconfigurée ne s'adapte pas dynamiquement sans intervention de l'utilisateur, et ce dernier doit avoir une certaine expertise sur le système pour apprécier les conséquences de ses choix.

L'utilisation de retours de l'utilisateur, implicites ou explicites, pour l'apprentissage par renforcement, a également été explorée. C'est notamment l'objet d'un thème très actuel en intelligence artificielle : l'*apprentissage par renforcement inverse*. Par exemple, Knox and Stone (2015) étudient, sur des tâches synthétiques mais avec des utilisateurs réels, l'influence de la polarité (« encourageant », « punisseur », *etc.*) d'un utilisateur entraînant un agent, via des retours continuels, pour des tâches séquentielles. Loftin et al. (2015) étudient l'inférence de *feedback* implicite, non numérique, par des agents à partir de l'observation des actions prises par des utilisateurs sur les mêmes tâches. Un autre exemple de retour implicite consiste, pour l'utilisateur, à indiquer sa préférence parmi deux politiques montrées par l'agent (Akrour et al., 2011). Par ailleurs, des modèles du comportement (Azaria

et al., 2012) et des préférences (Karami et al., 2014) d'un utilisateur peuvent être construits en utilisant des interactions utilisateur-système, pour fournir des systèmes personnalisés. Toutefois, bien que ces systèmes soient capables de bien réagir, en particulier à des changements d'utilisateurs et de préférences, ils ne sont pas adaptés à notre cas d'utilisation, car ils requièrent une interaction continue avec l'utilisateur, qui doit constamment critiquer les choix de l'agent. Ceci aurait un coût prohibitif dans notre application, car les utilisateurs principaux du système sont les analystes, dont l'amélioration du système d'extraction n'est pas la tâche première, et car la chaîne est appelée à traiter des volumes extrêmement importants de documents (de l'ordre d'un document toutes les 20 secondes, en continu, pour une instance de la chaîne).

Nous considérons actuellement les services individuels constituant la chaîne comme des « boîtes noires », avec pour objectif que l'agent améliore la façon de les enchaîner et de les paramétrer. Nous notons toutefois que des travaux s'intéressent à l'estimation de la qualité de tels services (Caron et al., 2014) et aux dégradations potentielles dans la chaîne (Amann et al., 2013), ainsi qu'à leur mise à jour automatique en utilisant les retours de l'utilisateur (Formiga et al., 2015). On pourrait tout à fait utiliser une combinaison de ces techniques en parallèle de notre approche.

F.2 La plateforme WebLab

La chaîne de traitement qui motive notre travail s'appuie sur la plateforme *open-source* (WebLab, 2016c). WebLab intègre des services *web* qui encapsulent les processus unitaires, et qui peuvent être interchangés ou permutés afin de créer une chaîne de traitement. Cette chaîne peut ensuite être utilisée pour analyser des documents multimédia *open-source*, et en extraire l'information.

Une chaîne de traitement typique de WebLab (Figure F.1), quand elle traite un nouveau document, commence par convertir le document source en une ressource XML. Cette ressource est ensuite transmise de service en service. Chaque service analyse le contenu de la ressource telle qu'il la reçoit, et l'enrichit avec des annotations. Enfin, les résultats sont stockés pour consultation par l'analyste.

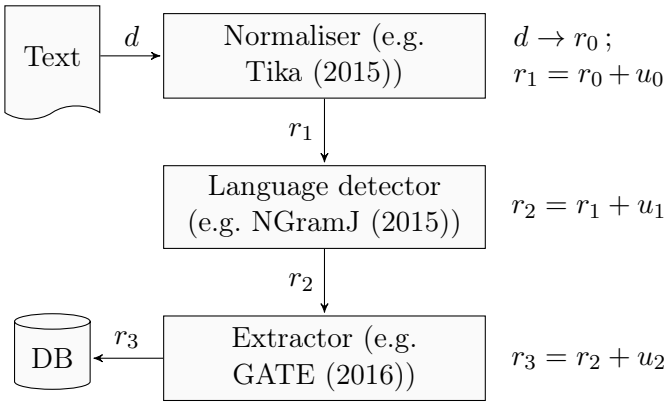


FIGURE F.1 – Une chaîne WebLab minimale : un document d est converti en une ressource XML r_0 et les ensembles d'annotations u_i sont ajoutés par chaque service.

Exemple 12 (annotations) *Considérons le document textuel suivant :*

> *4/29/1971: In a series of two incidents that might have been part of a multiple attack, suspected members of the Chicano Liberation Front bombed a Bank of America branch in Los Angeles, California, US. There were no casualties but the building sustained $1 600 in damages.*

La ressource XML (simplifiée) de la Figure F.2 est produite après le passage du document par deux services :

- *le* normaliser *a converti le document en ressource XML et a ajouté l'annotation « text/plain » (ligne 12) et le contenu original (ligne 19) ;*
- *le détecteur de la langue a ajouté la langue « en » (ligne 16).*

Nous considérons l'utilisation d'une telle chaîne pour l'extraction d'événements d'intérêt pour la veille économique, stratégique, ou militaire. En travaillant avec WebLab, nous nous situons dans la continuité du travail de Serrano (2014), qui propose une définition d'un événement (Definition 13). Cependant, notre travail est indépendant de WebLab, et nous aurions pu utiliser une autre définition (*e.g.* Van Hage et al. (2011)).

```
<resource type="Document" uri="weblab:aaa">
   <annotation uri="weblab:aaa#a0">
   <wp:originalContent resource="file:weblab.content"/>
   <wp:originalFileSize>255</wp:originalFileSize>
   <dc:source>documents/event.txt</dc:source>
   <wp:originalFileName>event.txt</wp:originalFileName>
   <dc:modified>2015-02-14T19:52:21+0100</dc:modified>
   <wp:collected>2015-02-10T00:11:00+0200</wp:collected>
   </annotation>
   <annotation uri="weblab:aaa#a1">
   <wp:isProducedBy resource="weblab:tika"/>
   <dc:format>text/plain</dc:format>
   </annotation>
   <annotation uri="weblab:aaa#a2">
   <wp:isProducedBy resource="weblab:ngramj"/>
   <dc:language>en</dc:language>
   </annotation>
   <mediaUnit type="weblab:Text" uri="weblab:aaa#0">
   <content>4/29/1971: In a series of two incidents that might have been part
   of a multiple attack, suspected members of the Chicano Liberation Front
   bombed a Bank of America branch in Los Angeles, California, US. There were no
    casualties but the building sustained $1 600 in damages.</content>
   </mediaUnit>
</resource>

```

FIGURE F.2 – Flux XML simplifié d'une ressource WebLab au milieu de la chaîne.

Definition 13 (Un événement) *Un* événement E *est un quadruplet* $E = (C, T, G, A)$*, où :*

- $C \subseteq \mathbb{C}$ *est la dimension conceptuelle ou sémantique de* E*, donnée par un ensemble d'atomes pris dans un domaine* $\mathbb{C}$ *commun à tous les événements ;*
- $T \subseteq \mathbb{T}$ *est la dimension temporelle de* E*, c'est-à-dire quand* E *est survenu (potentiellement ambiguë, telle que « mardi dernier ») ;*
- $G \subseteq \mathbb{G}$ *est la dimension spatiale de* E*, c'est-à-dire le lieu où* E *est survenu (potentiellement ambiguë également) ;*
- $A \subseteq \mathbb{A}$ *est la dimension agentive de* E*, c'est-à-dire l'ensemble des participants impliqués.*

Plus précisément :

- $\mathbb{C}$ est un ensemble fixé et fini d'atomes formalisés dans l'ontologie *WOOKIE* (Serrano (2014)) ;
- $\mathbb{T}$ est l'ensemble de toutes les dates relatives et absolues, par exemple « mardi dernier », « 2001 », « 2001/9/11 » ;

- $\mathbb{G}$ est l'ensemble des entités utilisées par Geonames (2015) ;
- $\mathbb{A}$ est l'ensemble (infini) de tous les participants pouvant être extraits, par exemple, les entités nommées, vus comme des chaînes de caractères.

Exemple 14 (extraction) *Le document précédent (Exemple 12) donnera lieu à l'extraction d'un événement* $E = (C, T, G, A)$ *avec :*

- $C = \{$*AttackEvent*, *BombingEvent*$\}$;
- $T = \{$*4/29/1971*$\}$;
- $G = \{$*Los Angeles*, *California*, *United States*, *US*, *North America*$\}$;
- $A = \{$*Chicano Liberation Front*, *Bank of America*$\}$.

L'efficience d'une chaîne de traitement ne dépend pas seulement des services utilisés et de l'ordre dans lequel ils sont appelés, mais elle dépend aussi des paramètres utilisés par ces services. Par exemple, l'extracteur GATE est paramétré par des dictionnaires *(gazetteers)*, qui sont des listes de noms et de verbes déclenchant la détection d'un type spécifique d'événement. Son dictionnaire de verbes pour l'extraction d'événements de type *bombing* contient typiquement les verbes *explode*, *detonate*, etc. Nous voyons donc GATE comme un *service paramétré* : choisir les bons dictionnaires à lui donner en paramètre fait partie des actions à (apprendre à) réaliser pour l'agent.

F.3 Apprentissage par renforcement

Dans la pratique, la chaîne de traitement utilisée est complexe. Elle est écrite et calibrée par des experts qui choisissent les services constituant la chaîne, leur ordonnancement et leurs paramètres (par exemple, les *gazetteers* de mots déclencheurs pour les services de détection d'événements et d'entités nommées). L'ordonnancement est figé, tout en pouvant être conditionnel (par exemple, si le document est en format PDF, passer au service 1 pour le convertir en XML, et passer au service 2 sinon).

Même avec de l'expertise, il est très difficile d'obtenir une chaîne parfaite. Ceci provient du fait que l'utilisation des documents *open source* du *Web* apporte des difficultés : leurs formats et contenus ne sont pas standardisés, les pages sources elles-mêmes ne sont pas contrôlables, elles peuvent comporter du « bruit » (comme de la publicité), être piratées, et leurs URL peuvent changer. On observe des erreurs d'extraction pouvant consister en des événements d'intérêt entièrement ou partiellement manqués, ou des événements

mal extraits (des informations non connexes dans la même phrase associées faussement, par exemple).

Par exemple, imaginons que l'analyste veuille de l'information sur les accords entre pays. Il est impossible de dire avec certitude si le mot « alliance » dans un document y fait référence ou non. Ce n'est qu'après l'extraction de l'événement, déclenchée par le mot « alliance », que l'analyste peut se rendre compte que la page parle de mariages, par exemple. Même si le document provient d'un journal politique, il se peut que l'on parle d'une coalition entre partis politiques, ou que les filtres de publicité n'aient pas réussi à attraper une vente de bagues. Un synonyme tel qu'« union » a pu être utilisé au lieu d'« alliance », et l'événement n'a pas été reconnu. Avec ces incertitudes, les experts qui paramètrent la chaîne essaient d'envisager les situations les plus communes. Il est inconcevable qu'ils puissent construire des chaînes à la main en examinant chaque document source.

Les seules informations connues avant de commencer une chaîne de traitement sont les services disponibles, leurs paramètres, et les états potentiels des documents, des ressources XML et du système (*cf.* section F.2). L'apprenant ne connaît ni la forme, ni le contenu des documents à l'avance, et il ne sait pas si une extraction sera possible. Ses décisions sont prises dans l'incertain. Pour atteindre notre objectif, nous appliquons donc les techniques de l'*apprentissage par renforcement* (*Reinforcement Learning*, RL) pour les processus de décision markoviens (*Markov Decision Processes*, MDP (Puterman, 1994)). Nous en rappelons les grands principes dans la suite ; pour une introduction détaillée, nous renvoyons le lecteur à Sutton and Barto (1998).

En RL, l'apprenant reçoit une récompense, basée sur les résultats des actions qu'il a choisies. Plus les résultats sont proches des objectifs, plus la récompense est élevée. Le système essaie de maximiser ces récompenses, typiquement en *exploitant* ce qu'il connaît déjà pour continuer à recevoir de bonnes récompenses, et en *explorant* de nouvelles actions avec l'espoir d'obtenir des récompenses encore plus importantes.

Le RL est généralement formalisé pour des MDP. Un tel processus modélise l'environnement en termes d'*états*, dans lesquels des *actions*, qui mènent à d'autres états du système de manière stochastique, sont possibles. Le fait que l'environnement soit dans un état donné à un certain instant apporte une récompense immédiate à l'agent. L'objectif d'un apprenant est de choisir ses actions de façon à maximiser son espérance de récompenses cumulées, sans connaître, initialement, ni les distributions sur les états résultant d'une action, ni les récompenses associées aux états.

Definition 15 (MDP) *Un* processus de décision markovien *est un 5-uplet* (S, A, P, R, γ)*, avec*

- S *un ensemble (ici fini) d'états possibles de l'environnement,*
- A *un ensemble (ici fini) d'actions (que l'agent peut effectuer),*
- P *un ensemble de distributions* $\{P_a(s, \cdot) \mid s \in S, a \in A\}$ *;* $P_a(s, s')$ *est la probabilité que l'environnement soit dans l'état* s' *après que l'agent a effectué l'action* a *dans l'état* s*,*
- R *une fonction de récompense, que nous supposons définie sur les états ;* $R(s)$ *est la récompense obtenue par l'agent pour se trouver dans l'état* s*,*
- γ *est un facteur d'atténuation dans* $[0, 1]$*, qui contrôle l'importance des récompenses espérées dans le futur, relativement aux récompenses espérées dans l'immédiat.*

Dans le cadre du RL, l'agent (apprenant) ne connaît initialement que les espaces d'états S et d'actions A, ainsi que le facteur γ. À tout instant t, il connaît l'état courant s_t de l'environnement, et choisit une action a_t. L'environnement passe dans un état s_{t+1} tiré selon la distribution $P_{a_t}(s_t, \cdot)$, et l'agent est informé de l'état s_{t+1} et de la récompense $r_{t+1} = R(s_{t+1})$. Le processus continue en s_{t+1}. L'agent doit, au fil de ses interactions avec l'environnement, apprendre une série de *politiques* $\pi_0, \pi_1, \ldots, \pi_t, \ldots$, une *politique* $\pi_t : S \to A$ donnant, pour l'instant t, l'action $\pi_t(s)$ à effectuer si l'état courant est s_t. Son objectif est à tout instant de maximiser l'espérance de récompense cumulée, c'est-à-dire l'espérance de la quantité $\sum_{t'=t}^{\infty} \gamma^{t'} R(s_{t'})$. Ce cadre générique admet de nombreuses variantes (pour un aperçu récent, voir Szepesvári (2010)).

De nombreux algorithmes ont été proposés dans la littérature pour les problèmes de RL. Dans ce document, nous utilisons d'abord une approche standard, le *Q-learning* (Watkins, 1989), avec des récompenses numériques, puis, pour le cadre des récompenses qualitatives, que les approches standards ne traitent pas, nous utilisons l'approche *SSB Q-learning* (Gilbert et al., 2016). Notons que notre contribution consiste d'abord à modéliser le problème de l'amélioration continue d'une chaîne de traitement comme un problème de RL, et que d'autres algorithmes pourraient être utilisés.

Pour que ce document soit autosuffisant, nous présentons dans la suite les deux algorithmes de façon générale, mais nous encourageons le lecteur

à consulter les articles originaux. Nous avons aussi lancé des tests en utilisant RMax (Brafman and Tennenholtz, 2003) et VMax (Rao and Whiteson, 2011), qui sont conceptuellement différents (ils apprennent le MDP sous-jacent, au lieu d'apprendre directement la politique), mais leurs temps de convergence et de calcul les rendent prohibitifs pour notre problématique.

F.3.1 Q-learning

Le *Q-learning* est un algorithme simple, facile à mettre en œuvre et à paramétrer. Il maintient, pour chaque couple état/action (s, a), une valeur notée $\hat{Q}(s, a)$, qui représente l'estimation courante, par l'agent, de l'espérance de récompense s'il se trouve dans s, exécute a, puis suit une politique optimale. Lorsque l'agent est dans l'état s_t, choisit l'action a_t, se retrouve en s_{t+1} et reçoit une récompense r_{t+1}, il met à jour son estimation de la valeur $\hat{Q}(s_t, a_t)$ comme indiqué dans l'algorithme 10, où α, le *taux d'apprentissage*, est un coefficient dans $[0, 1]$ qui fixe l'importance de la dernière expérience $(r_{t+1} + \gamma\, max(\dots))$ par rapport à l'expérience déjà accumulée (l'ancienne valeur de $\hat{Q}(s_t, a_t)$).

Pour l'exploration, nous utilisons une stratégie ϵ-gloutonne (EG). Le *taux d'exploration* $\epsilon \in [0, 1]$ règle le dilemme exploitation/exploration de la manière suivante. À chaque pas de temps t, avec une probabilité ϵ, une action est choisie aléatoirement (l'agent *explore*) ; sinon, l'agent *exploite* et choisit simplement l'action a_t qui maximise $\hat{Q}(s_t, a_t)$.

Algorithm 10: Q-learning

Data: MDP $\mathcal{M}$

while *vrai* **do**

- Choisir a_t en utilisation la stratégie d'exploration EG
- Exécuter a_t, observer s_{t+1}
- Soit $r_{t+1} = \mathcal{R}(s_{t+1})$
- $\hat{Q}_{t+1}(s_t, a_t) \leftarrow$ $\hat{Q}_t(s_t, a_t) + \alpha_t(s_t, a_t)(r_{t+1} + \gamma \max_b\{\hat{Q}_t(s_{t+1}, b)\} - \hat{Q}_t(s_t, a_t))$

F.3.2 SSB Q-learning

Notre approche fonctionne très bien avec du *feedback* numérique (section F.7), mais très peu d'utilisateurs pourraient dire « l'extraction d'un événement incorrect vaut 10,5 fois mieux qu'une extraction manquée, qui elle-même

vaut 7,9 fois mieux qu'une extraction prenant 90 secondes ». Plus vraisemblablement, les utilisateurs peuvent affirmer « je préfère l'extraction d'un événement incorrect à une extraction manquée (car je ne veux pas perdre d'information), et je préférerais que l'extraction soit aussi rapide que possible ». Notons ici l'expression de deux préférences, qui, bien qu'elles ne soient pas indépendantes (on peut supposer qu'une extraction plus rapide soit de moins bonne qualité), ne peuvent pas facilement être reliées numériquement. Nous voulons donc permettre aux utilisateurs de donner des retours à l'agent de façon indépendante sur les deux dimensions : « je préfère qu'un événement soit extrait », et « je préfère que l'extraction soit aussi rapide que possible ». De telles préférences peuvent être représentées par un ordre partiel sur les résultats possibles de l'extraction tels que perçus par l'utilisateur, notés $f_1, \ldots, f_k$ et vus comme des états finaux du MDP. Par exemple, f_i peut être « extraction correcte en 10 secondes », ou encore « traitement stoppé après 5 secondes, sans événement extrait ».

La résolution de problèmes de RL avec ce type d'information préférentielle/ordinale est le problème étudié par les travaux autour du *preference-based RL* (Akrour et al., 2012; Fürnkranz et al., 2012; Wilson et al., 2012; Wirth and Fürnkranz, 2013a; Wirth et al., 2016). Dans ce contexte, on ne peut pas maximiser directement l'espérance de récompense cumulée, car les valeurs numériques des récompenses ne sont pas accessibles (tout du moins directement).

Une approche consiste à calculer une fonction de récompense numérique à partir des retours qualitatifs reçus, mais ceci a typiquement l'inconvénient d'introduire des informations additionnelles, non exprimées par l'utilisateur. Une approche alternative consiste à utiliser des critères de décision adaptés à l'information ordinale. Un critère de ce type est la *dominance probabiliste* (Busa-Fekete et al., 2014), qui cherche à maximiser la probabilité d'obtenir un résultat (un état final f_i) préféré. Plus formellement, soient π_1, π_2 deux politiques, et soient F_1, F_2 deux variables aléatoires sur $(f_1, \ldots, f_k)$, où $\mathbb{P}(F_i = f_j)$ est donné par la probabilité que π_i aboutisse au résultat f_j ; $F_x \succeq F_y$ indique donc que le résultat obtenu par π_x est préféré (ou équivalent) à celui obtenu par π_y. Alors le critère de dominance probabiliste s'écrit :

$$\pi_1 \succeq \pi_2 \Leftrightarrow \mathbb{P}(F_1 \succeq F_2) \geq \mathbb{P}(F_2 \succeq F_1)$$

En français, une politique π_1 est préférée à une politique π_2 si la probabilité que π_1 obtienne un meilleur résultat que π_2 est plus grande que l'inverse.

En fait, la dominance probabiliste est un cas particulier d'une large classe de critères de décision, appelés *Skew Symmetric Bilinear — SSB —*

utility functions (Fishburn, 1984). Étant donné ϕ une fonction d'utilité SSB, $\phi(\pi, \pi') > 0$ (resp. $\phi(\pi, \pi') < 0$) exprime la préférence de l'utilisateur pour π sur π' (resp. pour π' sur π), tandis que $\phi(\pi, \pi') = 0$ exprime son indifférence. La dominance probabiliste est le cas où ϕ est toujours 1, 0 ou -1, et ne requiert donc de l'utilisateur que des retours purement ordinaux (« préféré », « indifférent ») sur les états finaux.

Pour traiter des problèmes de RL avec ce critère d'optimisation, nous utilisons l'algorithme *SSB Q-learning* récemment proposé (Gilbert et al., 2016). Cet algorithme traite le cas général d'un critère SSB. Comme le Q-learning, il utilise l'exploration EG, et il met à jour les Q-valeurs d'une manière similaire. Toutefois, au lieu d'utiliser du *feedback* numérique, que l'utilisateur n'a pas à lui fournir, il utilise ϕ et ses traitements passés, précisément les fréquences $\mathbf{p}_t$ avec lesquelles il a obtenu chaque résultat possible jusqu'alors (la valeur numérique ainsi définie est notée $\mathcal{R}_{\mathbf{p}_t}(s_{t+1})$). Intuitivement, l'algorithme cherche continuellement à faire mieux, au sens de ϕ, qu'il n'a fait jusqu'alors. Le pseudo-code de *SSB Q-learning* est donné par l'algorithme 11.

Algorithm 11: SSB Q-learning

Data: MDP $\mathcal{M}$, fonction SSB ϕ

while *vrai* **do**
- Choisir a_t en utilisant la stratégie d'exploration EG
- Exécuter a_t, observer s_{t+1}
- Soit $r_{t+1} = \mathcal{R}_{\mathbf{p}_t}(s_{t+1})$
- $\hat{Q}_{t+1}(s_t, a_t) \leftarrow$ $\hat{Q}_t(s_t, a_t) + \alpha_t(s_t, a_t)(r_{t+1} + \max_b\{\hat{Q}_t(s_{t+1}, b)\} - \hat{Q}_t(s_t, a_t))$
- **if** *s_{t+1} est un état final f_i et que l'agent n'a pas exploré* **then**
  - $\mathbf{p}_{t+1} = \mathbf{p}_t + \frac{1}{\eta+1}(\mathbf{1}_i - \mathbf{p}_t)$
  - $\#$ η est le nombre de fois où $\mathbf{p}$ a été mis à jour

Enfin, pour les deux algorithmes, nous avons utilisé des versions avec *traces d'éligibilité* (Sutton and Barto, 1998, section 7). Dans ces versions, un paramètre $\lambda \in [0, 1]$ permet et contrôle la rétropropagation des nouvelles expériences le long de la trace de décisions prises : $\lambda = 0$ correspond aux algorithmes tels que décrits ci-dessus, tandis qu'à l'autre extrême, $\lambda = 1$ correspond aux algorithmes de type Monte-Carlo.

F.4 Amélioration continue via l'apprentissage par renforcement

Puisque une chaîne unique pour traiter parfaitement chaque type de document est impossible à construire, l'idéal serait une chaîne faite sur mesure pour chaque document. Notre objectif est donc l'amélioration continue de la chaîne de traitement, de sorte que le système apprenne de ses erreurs. Une partie de notre contribution est de modéliser le traitement d'un document comme un processus décisionnel markovien (MDP), et son amélioration comme un problème d'apprentissage par renforcement. Une telle formalisation n'est pas directe, car le processus de traitement et ses entrées sont hétérogènes, et malgré tout ils doivent être formalisés en états et actions d'un problème de décision. De plus, les récompenses ne peuvent pas être spécifiées aisément, en particulier parce qu'elles doivent être récoltées de façon invisible pour les utilisateurs, sans impact sur leur travail. Finalement, la distribution des documents et les préférences des utilisateurs sont inconnues, et peuvent changer à tout moment.

Nous définissons les états en utilisant les métadonnées du document source, de la ressource et du système à un instant donné, par exemple, les informations déjà extraites, le temps déjà passé sur ce document, etc. (section F.5). Le système a ainsi une perception de la tâche sous la forme d'*états* du processus, et chaque passage par un service modifie l'état courant.

Par ailleurs, le système dispose d'un certain nombre d'*actions* qu'il peut appliquer dans l'état courant : ces actions correspondent au service suivant à lancer, à l'arrêt du traitement et l'enregistrement des événements extraits en base de données, ou à la configuration d'un service. La répétition d'une même action sur un même document est techniquement autorisée, mais sera pénalisée par le système de récompense si elle induit un temps de traitement trop long sans amélioration de l'extraction. La chaîne de traitement n'est plus figée, mais contrôlée par un algorithme de RL.

Les actions font transiter le système d'un état à l'autre. La stochasticité nous permet de prendre en compte le fait que des actions, menées dans un contexte apparemment identique, peuvent ne pas produire le même résultat. À titre d'exemple, choisir de détecter la langue peut résulter en l'extraction de langues différentes, ce qui est pris en compte directement par les actions stochastiques des MDP. Par exemple, supposons que 70 % des documents sources déjà traités étaient en français. L'agent percevra alors une probabilité 0.7 que, lorsqu'il applique l'action consistant à détecter la langue dans un état s_t, il arrive dans l'état s_{t+1}^{fr}, avec s_{t+1}^{fr} égal à s_t mais augmenté de

l'information « langue extraite » et de l'annotation « fr ».

Plus précisément, les états que perçoit le système sont des états combinatoires, formés par les valeurs d'un certain nombre de *descripteurs* des documents. Ces états permettent une généralisation en apprentissage, par exemple, nous avons déjà vu que le *sujet* du document (politique vs. mariage) influence de manière forte l'utilité du mot « alliance » pour l'extraction de l'information. Imaginons qu'un utilisateur cherche des accords entre des pays. On peut espérer que le système apprenne à modifier la chaîne ainsi au fil des interactions :

- si l'état courant a la valeur « vrai » pour le descripteur « sujetExtrait » et la valeur « mariage » pour le descripteur « sujetDuDocument », alors la meilleure action à effectuer consiste à arrêter le traitement (inutile de continuer) ;

- si le sujet est extrait, mais n'est pas « mariage », la meilleure action consiste à lancer un service d'extraction qui utilise « alliance » parmi les mots déclencheurs ;

- sinon, la meilleure action consiste à lancer un service de reconnaissance du sujet du document.

Enfin, les récompenses sont construites à partir du *feedback* de l'utilisateur de trois manières. La première consiste en une valeur numérique, basée sur les corrections apportées à l'extraction par l'analyste (le cas échéant) et sur le temps passé à traiter le document (section F.7). La seconde consiste en une valeur qualitative, basée sur les préférences purement ordinales de l'utilisateur sur les résultats finaux de l'extraction, et la troisième, qui forme un entre-deux, consiste en une préférence ordinale pondérée (section F.8). Les récompenses $r(s)$ sont donc données à l'agent uniquement pour les états finaux des traitements.

En pratique, il s'agit d'utiliser le fait que les analystes consultent les synthèses produites par le système, synthèses qui présentent les événements extraits, avec des liens vers les documents sources. Les actions de l'analyste, consistant par exemple à corriger l'information extraite, ou encore à consulter une synthèse sans rien corriger, fournissent indirectement un retour sur la qualité du traitement opéré, tout en utilisant, de manière non-intrusive, un travail effectué par les analystes indépendamment de la problématique d'amélioration du processus de traitement.

F.5 Cadre expérimental

Nous nous basons sur une chaîne simple mais typique (*cf.* section F.2), à laquelle nous avons ajouté la possibilité d'utiliser un service en réalité inutile, *Geo*. La chaîne est écrite comme une route Camel (2015) en XML, où chaque service est défini comme un *endpoint*. Nous utilisons le *Dynamic Router* pour donner à *BIMBO* le contrôle sur les services appelés, leur ordonnancement, et leurs paramètres (ces derniers sont plus spécifiquement le choix des *gazetteers* de GATE (Cunningham et al., 2014) pour la détection des événements dans un texte). Notons que les services sont des « boîtes noires » pour *BIMBO*, et que la connaissance de leurs WSDL, par exemple, n'est pas nécessaire [1].

Les actions disponibles consistent donc à :

- choisir le prochain service parmi {*Tika, NGramJ, GATE, Geo*} ;
- arrêter le traitement du document (« *STOP* ») ;
- choisir la configuration du service GATE, c'est-à-dire, ses *gazetteers*.

Un état est représenté par une affectation de caractéristiques :

- la langue du document, *language* ∈ {"en"," "} (où " " code « non-extraite ») ;
- le format du document, *format* ∈ {"*text/plain*"," "} (où " " code « non-extrait ») ;
- le nombre de secondes écoulées depuis le début du traitement du document courant (arrondi à la dizaine de secondes), *durée* ∈ $\mathbb{R}$;
- le nombre de services déjà appliqués sur ce document (groupés pour éviter l'explosion du nombre des états), *nbServices* ∈ $\{0-5, 6-20, 21+\}$;
- si un événement « intéressant » a déjà été extrait, *interesting* ∈ {*true, false*} ;
- si un événement quelconque a déjà été extrait, *any* ∈ {*true, false*} ;
- le *gazetteer* de noms choisi (si un tel choix a déjà été fait), *nounGazetteer* ;

[1] Bien sûr, une telle information pourrait être utilisée afin de généraliser sur les actions par exemple, ou de construire un modèle *a priori* des actions et des états, mais nous laissons ceci pour un travail futur.

- le *gazetteer* de verbes choisi (si un tel choix a déjà été fait), *verbGazetteer*.

Dans nos expériences, ces ensembles donnent un espace de 8 000 états pour 15 actions, et sont choisis à titre illustratif pour la preuve de notre approche. Un système opérationnel prendrait évidemment en compte de nombreuses autres caractéristiques, services et paramètres (type de documents, liste complète de langues, service de traduction, etc.). Notons que même avec un espace d'états/d'actions ainsi enrichi, le temps de calcul ne serait pas un obstacle avec un algorithme de type *(SSB) Q-learning*, pour lequel les calculs sont instantanés à chaque pas de décision.

Nous avons considéré un corpus de textes, dont les événements d'intérêt sont déjà connus. Le *Global Terrorism Database* (GTD) est une base de données *open source* composée des détails de plus de 125 000 événements terroristes mondiaux de 1970 à 2014 (LaFree (2010)). Nous avons mis les types d'événements du GTD en correspondance avec ceux de WOOKIE ; une chaîne d'extraction parfaite extrairait des documents $d_1 \dots d_N$ de notre corpus les événements $E_1, \dots, E_N \in GTD$, respectivement (*cf.* Exemple 12, Exemple 14). Une telle chaîne n'existe pas en réalité. Nous utilisons donc les résultats d'une chaîne « experte » (construite à la main) comme référence. Nous attendons de notre système qu'il apprenne une chaîne qui s'approche de ce but, en apprenant le bon ordonnancement des services, le bon paramétrage des services, et le fait que certains services (dans notre cas, *Geo*) et certains *gazetteers* de GATE ne sont pas utiles.

Comme expliqué dans la section F.3, l'agent (l'IA) n'a initialement connaissance que de l'espace des états et des actions, et donc, dans notre cas, commencera à apprendre sans connaissance *a priori* des documents ou des besoins de l'analyste. Nous dirons qu'une telle IA est « non-formée ». Une fois que l'IA a traité un certain nombre de documents, elle aura appris ce qu'elle considère être une politique optimale. Nous dirons qu'elle est « formée », et nous pourrons vérifier l'efficacité de la politique apprise en obligeant l'IA à la suivre, c'est-à-dire, en mettant $\epsilon = \alpha = 0$, arrêtant ainsi son exploration et son apprentissage.

Ici, nous testons la capacité d'apprentissage à partir de rien d'une IA sur des documents complètement inconnus. Bien sûr, en production, nous pourrions capitaliser sur l'expertise existante en initialisant l'IA avec une politique basée sur celle d'une chaîne standard.

F.6 Mesure de la qualité des résultats

Pour mesurer la qualité des résultats de nos tests, il faut d'abord définir la similarité entre un événement extrait par la chaîne sur le document $E_1 = (C_1, T_1, G_1, A_1)$ et l'événement « parfait » du *GTD* (l'oracle) $E_2 = (C_2, T_2, G_2, A_2)$ qui correspond à ce document. De nombreuses méthodes ont été proposées pour mesurer la similarité (Pandit et al. (2011); Tversky and Gati (1978); Cohen et al. (2013); Dutkiewicz et al. (2013), pour n'en citer que quelques-unes), mais nous avons besoin d'une mesure qui prenne en compte les quatre dimensions des événements, et sensible par ailleurs à la différence entre différents types d'événements. Nous définissons donc :

$$\sigma(E_1, E_2) = \frac{a\sigma_C(C_1, C_2) + b\sigma_T(T_1, T_2) + c\sigma_G(G_1, G_2) + d\sigma_A(A_1, A_2)}{(a + b + c + d)}$$

La similarité conceptuelle $\sigma_C(C_1, C_2)$ est de 1 s'il y a un atome commun entre les dimensions conceptuelles de E_1 et E_2, et de 0 sinon. Par exemple, pour $C_1 = \{BombingEvent, AttackEvent\}$ et $C_2 = \{BombingEvent\}$, on obtient $C_1 \cap C_2 = \{BombingEvent\}$, et donc $\sigma_C(C_1, C_2) = 1$. Nous procédons de même pour la similarité géographique $\sigma_G(G_1, G_2)$.

La similarité temporelle $\sigma_T(T_1, T_2)$ est de 1 s'il y a au moins un atome en commun entre les dimensions temporelles de E_1 et E_2. Sinon, nous utilisons l'information dérivée (jour, mois, année, jour de la semaine). Par exemple, pour $T_1 = \{7\ October\ 1969\}$ et $T_2 = \{October\}$, $\sigma_T(T_1, T_2) = \frac{1}{10}$. Pour $T_3 = \{Tuesday\}$, l'intersection avec T_1 est vide, mais en notant que le 7 octobre 1969 a été un mardi, nous obtenons $\sigma_T(T_1, T_2) = \frac{1}{7}$. Ces valeurs ont été choisies par expérience, et soulignent la difficulté d'associer une valeur numérique à une comparaison. Cette difficulté sera levée par l'approche basée sur du *feedback* qualitatif.

Enfin, pour la similarité entre les dimensions agentives de E_1 et E_2, nous utilisons la distance de Levenshtein sur chaque paire d'agents a_1, a_2 pris dans A_1, A_2 (nombre minimal de caractères à supprimer, insérer ou remplacer pour passer d'une chaîne à l'autre), distance rendue « floue » en considérant les sous-séquences de la chaîne principale (Ginstrom, 2007), et notée $FSLD(a_i, a_j)$. Nous définissons $\sigma_A(A_1, A_2)$ comme $(1 - \max\{FSLD(a_1, a_2) \mid a_1 \in A_1, a_2 \in A_2\})$, si elle est au-dessus d'un certain seuil θ, et 0 sinon (après expérimentation, $\theta = 0.45$ donne de bons résultats). Par exemple, pour $A_1 = \{Jharkhand\ \cancel{anti\text{-}Maoist}\ \cancel{Maobadi}\ \cancel{Protirodh}\ \cancel{Committee}\ member\}$ et $A_2 = \{Jharkhand\ \cancel{Party}\ member\}$, on obtient une distance Levenshtein floue de 11, donc $\sigma_A(A_1, A_2) = 1 - \frac{11}{22} = 0.5$.

Nous ne nous intéressons pas ici à l'association d'entités telles que *François Hollande / le président*, mais la modularité du système permettrait de prendre en compte cette similarité facilement, en s'appuyant sur des ressources adéquates.

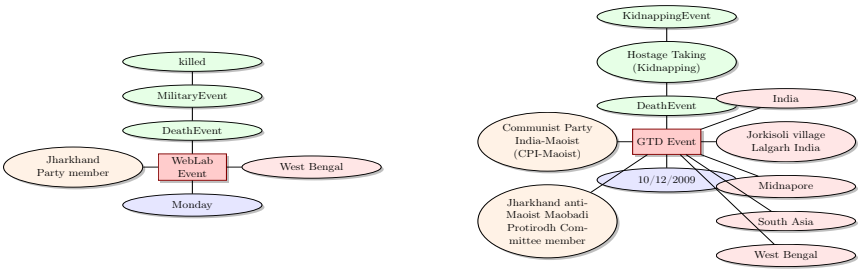
Les volumes de documents traités en production (idéalement, tous les documents du *web*) étant très importants, le temps passé à traiter un document doit être minimisé. La qualité des extractions prend donc en compte la similarité entre l'événement cible et l'événement extrait (le cas échéant), mais aussi le temps de traitement. Précisément, en notant $\tilde{\sigma}(E_e, E_g)$ la similarité moyenne entre les événements extraits (s'ils existent) et l'événement du *GTD*, et t le temps passé par la chaîne sur le document, nous définissons la qualité de l'extraction Q par $\tilde{\sigma}(E_e, E_g)/t$ pour $\tilde{\sigma}(E_e, E_g) \neq 0$, et par $Q = -t$ sinon (c'est-à-dire s'il n'y pas d'extraction, ou une similarité nulle avec l'événement cible). Nous formalisons ainsi le fait que l'extraction d'événements corrects est primordiale, doit être effectuée en un temps raisonnable, et que l'agent doit détecter rapidement, le cas échéant, qu'il n'y a pas d'événement intéressant à extraire.

Example 16 (Similarité) *Prenons les deux événements qui sont décrits dans la Figure F.3. Nous voyons un atome commun (*Death Event*) dans la dimension conceptuelle, qui donne une similarité de 1. La similarité spatiale est de 1 grâce à l'élément* West Bengal. *Nous déduisons le jour de la semaine, pour obtenir une similarité temporelle de 0.143. Enfin, la similarité agentive, comme nous avons déjà vu, est de 0.5. Avec des poids égaux sur les dimensions ($a = b = c = d = 1$), nous avons une similarité globale entre l'événement extrait, et celui du GTD de*

$$\sigma(\textit{WebLab Event}, \textit{GTD Event}) = \frac{1.0 + 0.143 + 1.0 + 0.5}{4} = 0.66075$$

F.7 Tests avec un *feedback* numérique

Durant ces tests, l'IA reçoit une récompense numérique sur la qualité Q de ses résultats, telle que définie dans la section F.6. En production, ceci équivaut à la récompenser en fonction de l'événement qu'elle a extrait (si elle en a extrait un) et de l'événement tel que corrigé par l'analyste humain, ce dernier étant considéré comme l'événement « parfait » pouvant être extrait du document (si un événement peut être extrait). Ceci est réalisé de manière non-intrusive, dans la mesure où cela repose sur des corrections que l'analyste a besoin d'effectuer de toute façon, mais nécessite un réglage fin



(a) L'événement extrait

(b) L'événement extrait « parfait » du GTD

FIGURE F.3 – Comparaison de deux événements.

de la définition de la similarité et de ses paramètres, une tâche cognitive difficile. Notons que l'information donnée à l'IA, ici, est la même que celle utilisée pour l'évaluation de la qualité de la politique qu'elle a apprise. Cette méthode est classique en RL, mais pas naturelle dans la vie réelle. Dans la section F.8, nous retirons cette contrainte et produisons les résultats obtenus avec du *feedback* qualitatif.

Nous avons constitué trois groupes de tests avec une récompense numérique :

1. Une IA a été formée et sa performance a ensuite été comparée avec celle d'une IA non-formée, et d'une chaîne « experte » sur 1 000 documents inconnus, biaisés vers un type d'événement « intéressant ».

2. La performance d'une IA non-formée a été testée sur 5 000 documents inconnus, choisis aléatoirement, avec un très faible pourcentage d'événements extractibles.

3. Finalement, la capacité d'apprentissage d'une IA non-formée a été examinée avec une récompense sporadique.

F.7.1 Formation et test sur des documents biaisés vers un type d'événement « intéressant »

Nous avons d'abord testé la qualité de la politique apprise (1) après un entraînement répété sur un petit ensemble de documents, et (2) à partir de zéro sur un ensemble plus large de documents choisis au hasard.

(1) Nous avons entrainé notre AI sur 100 documents *GTD*, traités dans le même ordre 30 fois. L'IA a eu le choix entre les services et *STOP* (comme

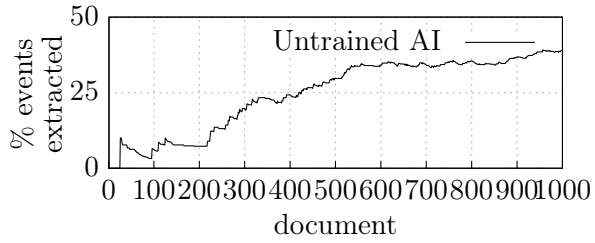


FIGURE F.4 – Pourcentage d'événements extraits par l'IA non-formée.

dans la section F.5), et six *gazetteers* (trois listes de verbes et trois de noms). Deux paires de *gazetteers* (les noms et verbes *bombing* et *injure*) contiennent des listes de mots susceptibles de déclencher l'extraction d'événements de ce type. Une autre paire de *gazetteers* factices contenait des mots non présents dans la *GTD*. Nous avons représenté la préférence d'un analyste en définissant les événements *Bombing* comme des événements « intéressants » en accentuant la similarité sémantique (spécifiquement $a = 20, b = c = d = 1$). Il s'agissait de vérifier que l'IA tiendrait compte de la préférence de l'analyste en favorisant les *gazetteers bombing* par rapport aux *gazetteers injury*, et en ignorant les factices, qui ne doivent donner lieu à aucune extraction. 64% des documents décrivaient ces événements *Bombing* « intéressants », 17% d'autres événements, et 19% ne contenaient aucun événement extractible.

Le Q-learning a été lancé avec des paramètres standards : $\epsilon = 0.4$, qui sera divisé par 2 tous les 500 documents jusqu'à $\epsilon = 0.1$, $\alpha = 0.2$. Pour ces tests λ était fixé à 0 (pas de traces d'éligibilité).

(2) Nous avons testé à la fois la politique apprise par cette IA formée, et une IA non-formée sur 1 000 nouveaux documents choisis aléatoirement depuis la *GTD*. Seulement 29% d'entre eux contenait des événements *Bombing* « intéressants », 7% d'autres événements, et 64% aucun événement extractible. Nous avons réglé la chaîne « experte » (*c.f.* la section F.5) pour n'extraire que des événements *Bombing*.

L'IA non-formée (partant de zéro) est parvenue à extraire 39% des événements possible des 1 000 documents inconnus, et nous voyons sur la Figure F.4 qu'elle généralise bien, sa performance s'améliorant au fur et à mesure des documents rencontrés.

L'IA formée a démontré une politique similaire à celle de la chaîne experte, en extrayant 100% des événements. Ainsi, elle aussi généralise bien, appliquant avec succès la politique apprise à des documents inconnus. Notons que les IAs ont également appris à ordonner la chaîne, *e.g.*, dans l'état

correspondant à un document dont ni le format ni la langue ont été détectées, et dont le temps de traitement est quasi-nul, la meilleure action apprise était de passer le document au service Tika. Dans l'état correspondant à un document où au moins un événement est extrait, quelle que soit la valeur des autres attributs, l'IA arrête le traitement de ce document.

Comme espéré, l'IA a aussi appris à optimiser la chaîne. Elle n'a pas utilisé le service inutile (*Geo*), et a seulement utilisé la liste de verbes *bombing*. Cependant, elle a de manière surprenante préféré les noms *injure* à ceux de *bombing*. Après enquête, nous avons trouvé que le service GATE fourni n'utilise que les verbes pour l'extraction d'événements, ignorant les noms. Ceci montre que l'IA est capable de découvrir des stratégies qui ne sont pas évidentes pour l'expert ayant calibré la chaîne.

F.7.2 Tests de performance sur des documents avec un très faible pourcentage d'événements extractibles.

Pour ces tests, nous avons augmenté l'espace des actions en donnant à l'IA le choix parmi dix *gazetteers* (cinq listes de verbes et cinq de noms) : six (les verbes et noms de *bombing*, *shooting* et *injure*) contiennent des mots susceptibles de déclencher la détection d'un événement de ce type, deux (les verbes et noms de *mixte*) contiennent un mélange de mots susceptibles de déclencher la détection de plusieurs types d'événements, et deux sont factices. Nous nous attendions à ce que l'IA apprenne à utiliser de préférence les *gazetteers mixte*, ensuite les *gazetteers* qui déclenchent la détection d'un événement unique, et qu'elle évite les *gazetteers factices*. Nous nous attendions également à ce que l'IA ne soit pas sensible à un espace d'actions plus grand.

Nous avons testé une IA non-formée cette fois-ci sur 5 000 documents inconnus, aléatoirement choisis avec en moyenne 22.56% d'événements extractibles (Figure F.5).

Parallèlement, après le traitement de chaque 100 documents, nous avons testé une IA initialisée avec la politique apprise à ce point, et les valeurs de ϵ et α à 0, sur 10 documents inconnus contenant tous des événements extractibles.

Nous voyons dans les Figures F.7– F.9 que la première extraction n'a lieu qu'à 173 documents, et que les extractions sur les premiers 700 documents ont une qualité inférieure à celle de la chaîne « experte ». Au document 701, en revanche, l'IA commence à recevoir des récompenses similaires à celles de la chaîne « experte », ce qui indique que la qualité de ses extractions s'améliore.

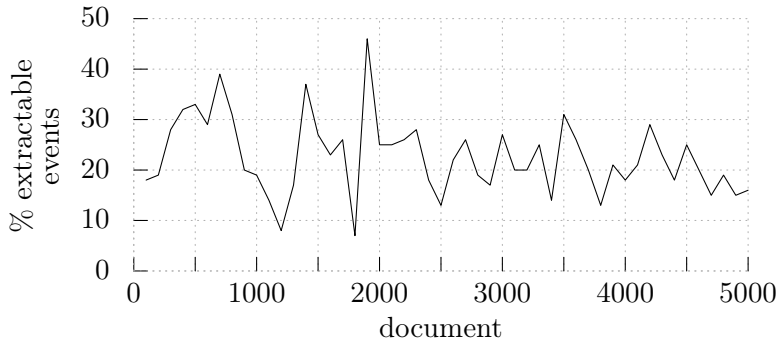


FIGURE F.5 – Le pourcentage de documents contenant des événements extractibles par 100 documents.

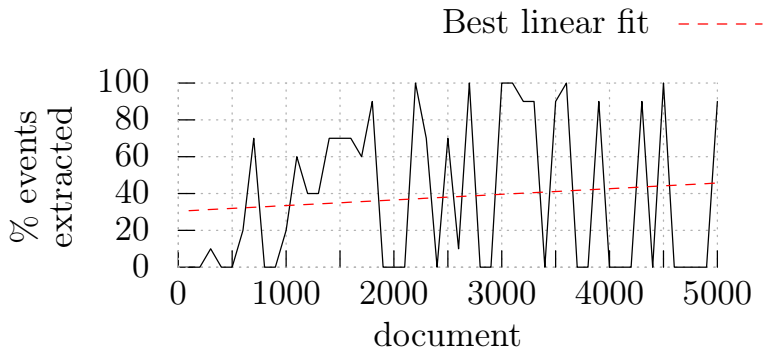


FIGURE F.6 – Le pourcentage des événements extraits en utilisant les valeurs apprises à la fin de chaque 100 documents.

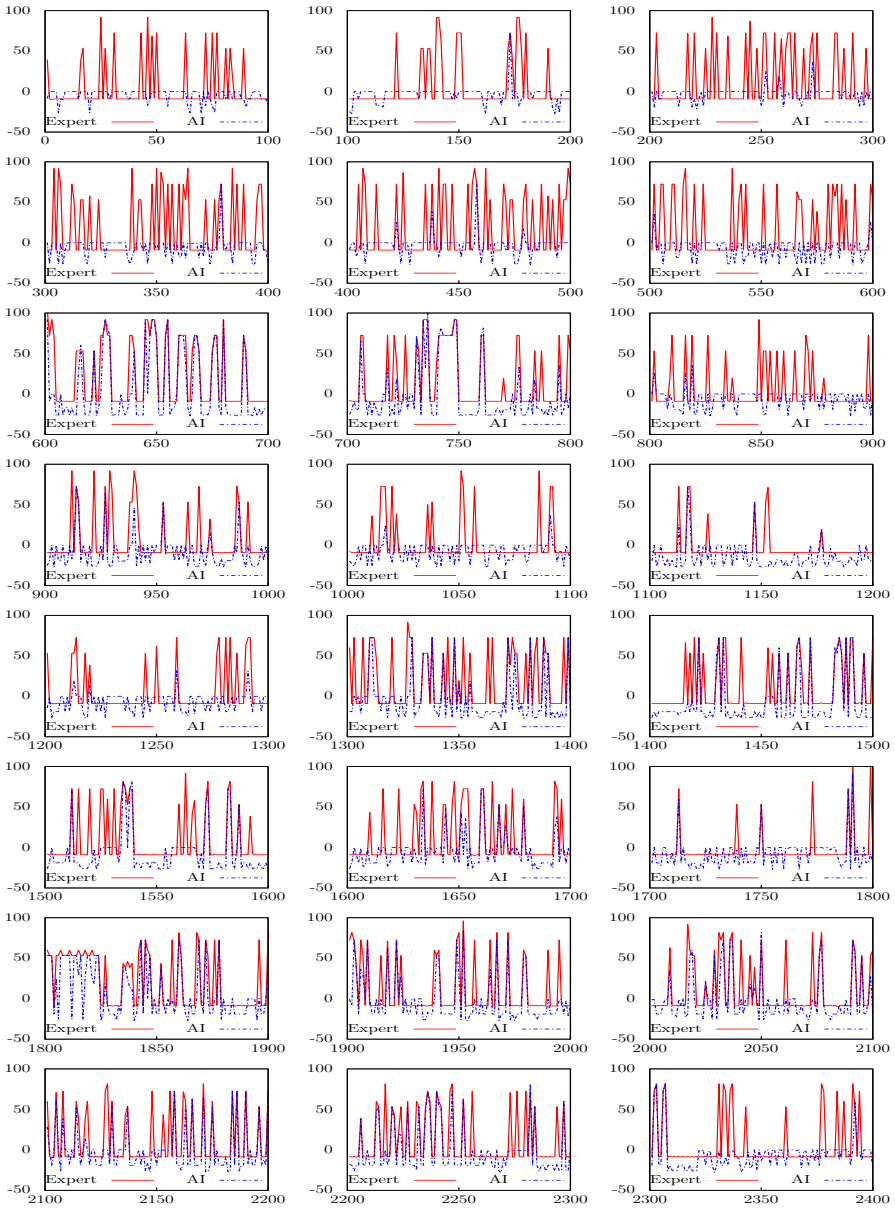


FIGURE F.7 – La qualité des extractions de la chaîne « experte », comparées à celles de l'IA non-formée pour les documents 1 - 2 400.

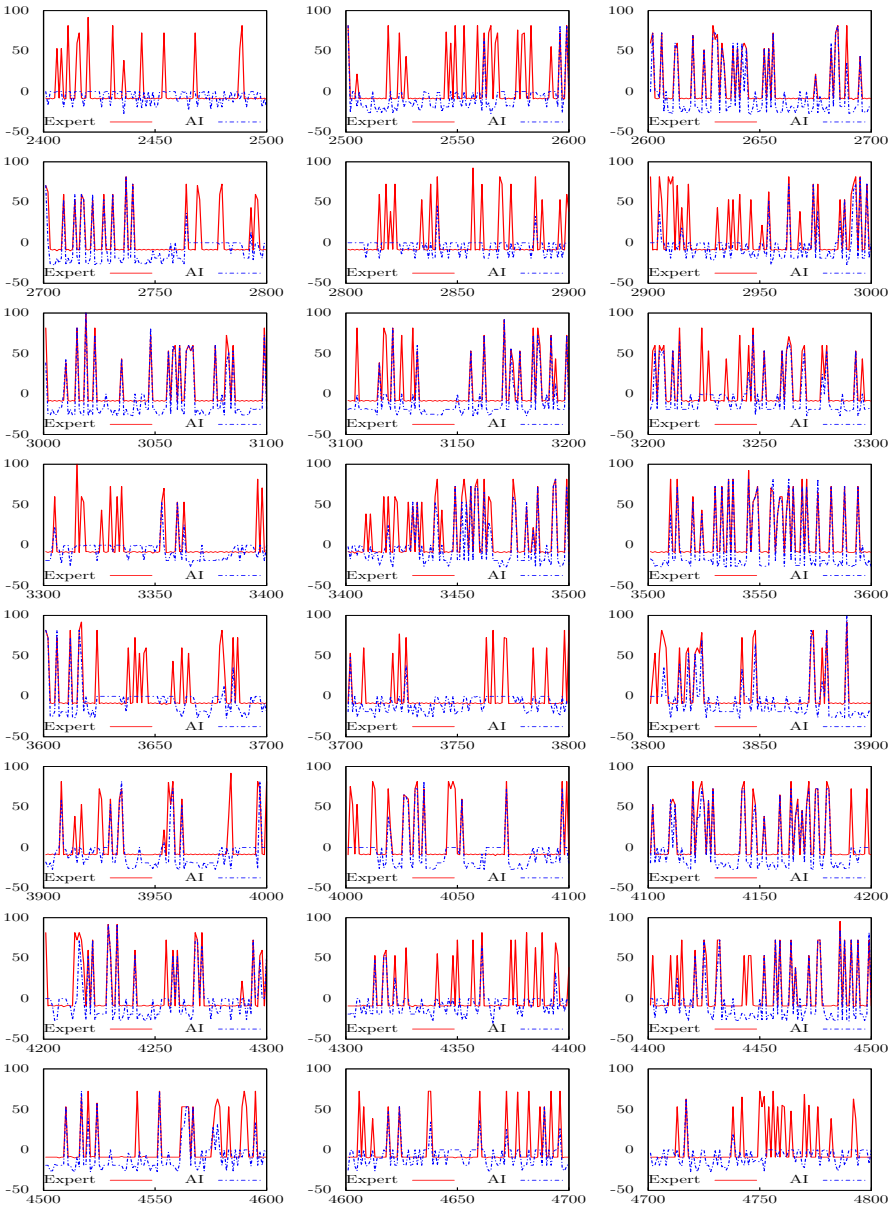


FIGURE F.8 – La qualité des extractions de la chaîne « experte », comparées à celles de l'IA non-formée pour les documents 2 401 - 4 800.

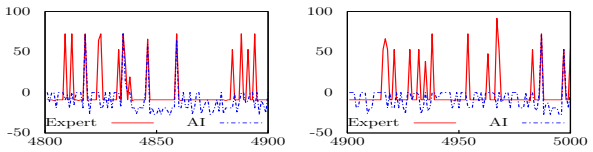


FIGURE F.9 – La qualité des extractions de la chaîne « experte », comparées à celles de l'IA non-formée pour les documents 4 800 - 5 000.

En testant les politiques apprises, nous constatons à la Figure F.6 une amélioration de performance de l'IA, même si les politiques ne sont pas constamment capables d'extraire 100 % des événements. Par exemple, au document 900, l'IA traite la quasi-totalité des documents le plus rapidement possible. Ceci peut être dû à la baisse en pourcentage d'événements extractibles dans les 100 documents précédents. Elle préfère recevoir une récompense de zéro pour passer un document tout de suite, au lieu de recevoir une récompense négative en essayant d'extraire un événement qui n'existe pas. En comparant les documents 1100–1300 de la Figure F.7 avec la Figure F.5, nous constatons que le pourcentage d'événements extractibles est très faible, et que l'IA a des problèmes de qualité et de quantité de ses extractions. Nous pouvons déduire de ces résultats que l'IA est sensible au pourcentage d'événements extractibles, mais qu'elle n'a besoin que de documents ayant environ 20 % d'événements extractibles pour réussir les extractions.

F.7.3 Tests de performance avec un *feedback* sporadique

Les deux jeux de tests précédents avaient pour but d'évaluer la capacité de l'IA à extraire des événements depuis des documents inconnus. Nous avons réalisé une troisième série de tests afin d'évaluer la vitesse à laquelle une IA non-formée pourrait apprendre la politique d'une chaîne experte (capable d'extraire 100 % des événements), et la qualité de cette politique, pour vérifier que les événements extraits sont bien similaires aux résultats attendus. Pour cela, nous avons pris un jeu de 63 documents de la *GTD*, chacun d'eux contenant un événement extractible. Nous avons traité ces documents dans le même ordre, de manière répétée, par lots. Nous avons utilisé le même espace d'actions qu'en sous-section F.7.2, c'est-à-dire, un choix entre les services, *STOP* et dix *gazetteers* (cinq listes de verbes, et cinq listes de noms). Le Q-learning a été lancé avec des paramètres similaires à ceux des premiers tests : $\epsilon = 0.4$, divisé par 2 tous les 10 lots jusqu'à $\epsilon = 0.1$, $\alpha = 0.2$, et $\lambda = 0$.

Nous avons d'abord entraîné l'IA avec une récompense donnée pour

chaque document. Il s'est avéré qu'après seulement 1 008 documents (c'est-à-dire 16 lots), la politique apprise était capable d'extraire 100 % des événements. De manière plus importante, cette politique était optimale, c'est-à-dire que la qualité de l'extraction était à égalité avec celle de la chaîne experte, comme l'illustre la Figure F.10a.

L'analyste n'est pas disponible en permanence pour consulter chaque document dès qu'il est traité. Nous avons alors lancé un test similaire, mais la récompense n'était donnée qu'à la fin du traitement de chaque lot, empêchant l'IA d'apprendre durant un lot. L'IA était légèrement plus lente à apprendre une politique optimale, ayant besoin de 1 260 documents (20 lots, Figure F.10b).

Enfin, les analystes ne sont pas des entraîneurs dédiés, et ne corrigeront pas toutes les extractions. Nous avons donc testé une récompense donnée à la fin de chaque lot, avec une probabilité de 10 % sur chaque extraction (sinon, l'IA ne recevait aucune récompense pour cette extraction).

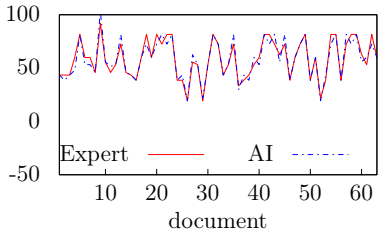
Même avec une récompense aussi sporadique, l'IA est parvenue à extraire 50 % des événements possibles après seulement 1 890 documents (30 lots), et 100 % après 5 103 documents (81 lots). La qualité de ces deux politiques est représentée sur les Figures F.10c,F.10d, et suggère que la capacité à extraire des événements et à extraire des événements corrects croissent conjointement. Notons que la qualité la plus basse (en comparaison avec la chaîne experte) que l'on observe dans la dernière courbe est due au temps de traitement, et non à la similarité avec les événements attendus.

F.8 Tests avec un *feedback* intuitif

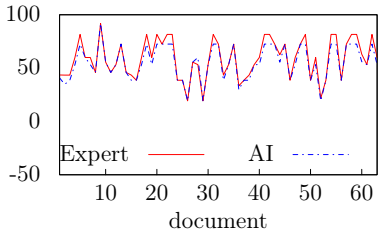
Les tests présentés en section F.7 supposent que l'utilisateur est capable de donner un *feedback* numérique à l'agent, ce qui n'est pas réaliste. Nous présentons ici donc une évaluation expérimentale de notre méthode en utilisant un *feedback* qualitatif. Rappelons que l'approche n'a besoin que d'un retour sur les qualités relatives de deux extractions ; la comparaison entre politiques se fait alors sur le critère de dominance pondérée, relativement à ces comparaisons.

Nous explorons ici deux types de comparaison entre les résultats de deux traitements f, f'. Le protocole expétimental est similaire à celui de la section F.7, et nous prenons comme références la chaîne experte, ainsi que les résultats obtenus par le Q-learning avec une récompense numérique (noté QL dans cette section).

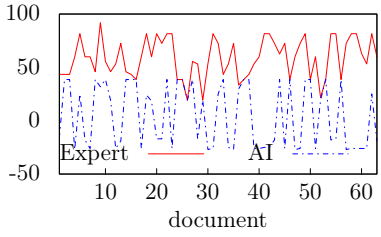
Premièrement nous encourageons l'IA à d'abord extraire les événements,



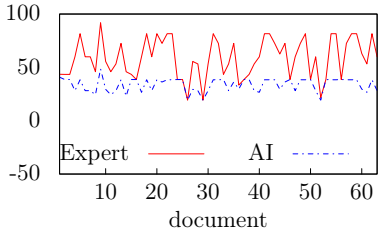
(a) Complète instantanée.



(b) Complète différée.



(c) Sporadique 50%.

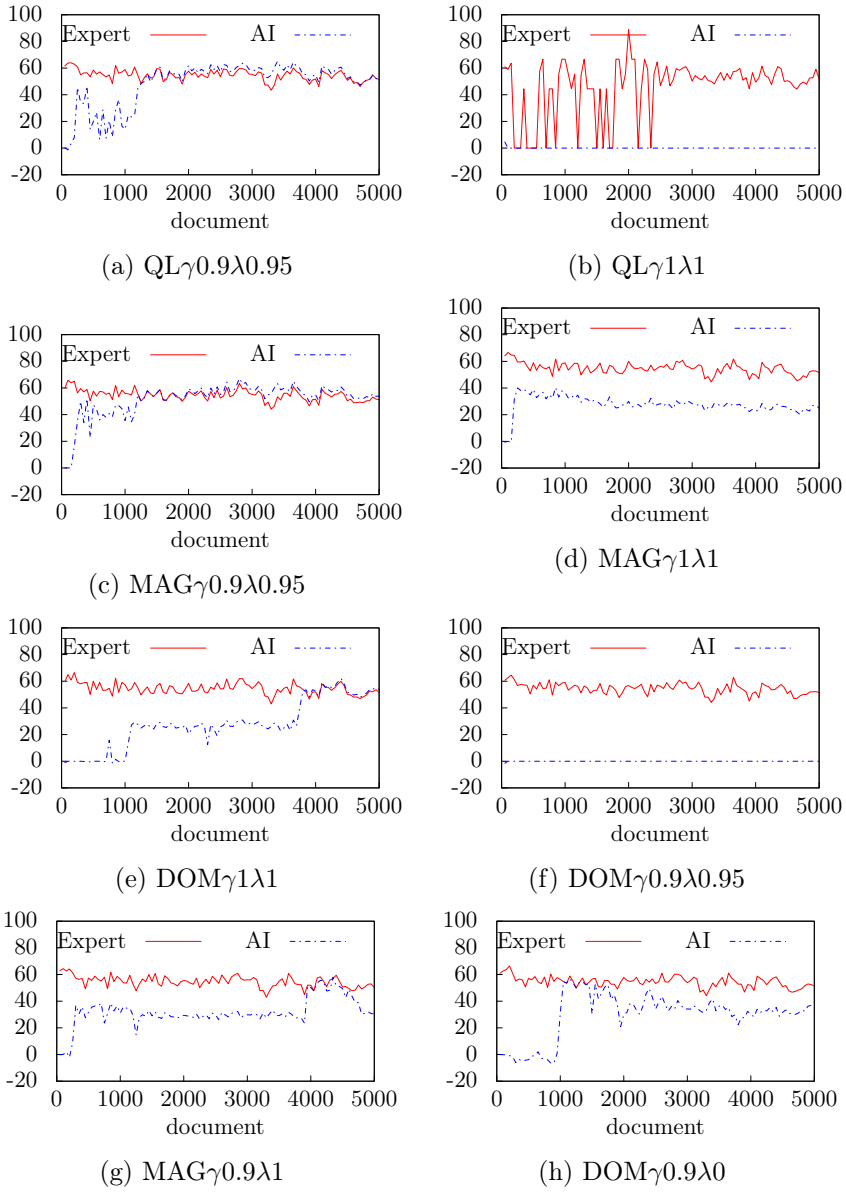


(d) Sporadique 100%

FIGURE F.10 – La qualité de la politique apprise avec une récompense (a) complète instantanée ; (b) complète différée ; (c) sporadique à 50% extraction ; (d) sporadique à 100% extraction.

et à le faire rapidement, ou à vite reconnaître qu'il n'y a aucun événement à extraire. Ceci repose sur la corrélation entre la capacité à extraire n'importe quel événement et celle à extraire des événements corrects, tel que démontré à la section F.7 pour le Q-learning. En l'état, notons que cette récompense peut même être calculé automatiquement, sans intervention de l'analyste. Évidemment, cette relation de préférence pourrait être complétée avec des informations supplémentaires sur les résultats obtenus, tels que la qualité perçue de l'extraction, *etc.* Cette récompense noté DOM est défini comme :
– $f \succ_{\text{DOM}} f' \Leftrightarrow \phi_{\text{DOM}}(f, f') = 1$ ssi (i) le traitement f a extrait un événement mais pas f', ou (ii) aucun ou bien les deux ont extrait un événement, et f a été plus rapide ;
– $f \sim_{\text{DOM}} f' \Leftrightarrow \phi_{\text{DOM}}(f, f') = 0$ ssi les deux ont pris à peu près le même temps, *i.e.*, l'écart entre les deux temps de traitement est inférieur à $margin$ (réglé à 5) secondes.

Nous notons MAG le deuxième type de récompense. Intuitivement, MAG est un intermédiaire entre QL et DOM, qui peut être vu comme une forme pondérée de dominance probabiliste, demeurant toutefois naturel. Nous ac-



(a) QLγ0.9λ0.95 (b) QLγ1λ1

(c) MAGγ0.9λ0.95 (d) MAGγ1λ1

(e) DOMγ1λ1 (f) DOMγ0.9λ0.95

(g) MAGγ0.9λ1 (h) DOMγ0.9λ0

FIGURE F.11 – Résultats : (a) QL meilleur ; (b) QL pire ; (c) MAG meilleur ; (d) MAG pire ; (e) DOM meilleur ; (f) DOM pire ; (g) et (h) MAG et DOM dégradation.

cordons plus d'importance dans l'ordre à : l'extraction des événements, puis l'extraction des événements exactes, et enfin la vitesse de traitement :
– $\phi_{\text{MAG}}(f, f') = 1000$ ssi f a extrait un événement et pas f' ;
– $\phi_{\text{MAG}}(f, f') = 100$ ssi les deux ont extrait un événement, mais l'extraction de f est de meilleure qualité (section F.6) que celle de f' ;
– $\phi_{\text{MAG}}(f, f') = 10$ ssi les extractions étaient de qualité similaire, ou aucune extraction a eu lieu, mais f a été plus rapide par au moins *margin* secondes ;
– $\phi_{\text{MAG}}(f, f') = 0$ ssi f et f' ne tombent dans aucun des cas précédents. Les valeurs $0, 10, 100, 1000$ reflètent une différence de magnitude entre l'importance des différents critères, sans besoin de les régler plus précisément.

Nous avons réalisé un jeu de tests étendu avec QL, DOM et MAG, en faisant varier les paramètres γ, ϵ, et λ (voir la section F.3). Nous avons exécuté les IA non-formées sur un ensemble de 5 000 documents *GTD* (présentés dans le m̀ême ordre, et rencontrés une seule fois dans tous les tests). Les actions étaient choisies parmi dix *gazetteers*, les services *Tika, NGramJ, GATE, Geo*, et *STOP*. La qualité du traitement de chaque document a été mesurée comme décrit en section F.5 pour toutes les approches. Nous insistons sur le fait que la qualité est donc mesurée avec le même critère (numérique) que celui utilisé pour donner un récompense à QL, et ceci pour les trois approches ; DOM et MAG reçoivent donc un feedback à première vue peu informatif vis-à-vis du critère à optimiser, demandant donc un effort cognitif faible aux experts.

Évidemment, comme les IA apprenaient de zéro, la politique initiale était de mauvaise qualité. Nous nous sommes intéressés au temps requis (en termes de nombre de documents traités) pour atteindre une bonne politique.

Le paramètre α a été réglé comme dans Gilbert et al. (2016), *i.e.* diminuant avec le nombre de visites de la paire état / action courante. Nous avons exécuté 36 combinaisons de tests en faisant varier les autres paramètres, afin d'obtenir un ensemble de résultats étendu, et de mesurer la robustesse vis-à-vis du choix des paramètres :
– les algorithmes QL, DOM, et MAG,
– le paramètre EG ϵ divisé par 2 après 2500 ou 1000 documents,
– $\gamma = 0.9$ et $\gamma = 1$,
– $\lambda = 0$, $\lambda = 0.95$, et $\lambda = 1.0$.
La Figure F.11 montre les courbes les plus intéressantes et représentatives. Nous avons tracé uniquement la qualité d'extraction lorsque l'IA a suivi sa meilleure politique pendant tout le traitement (elle a exploité) avec la ligne bleu en pointillés, et celle de la chaîne « experte » (ligne continue

rouge)[2]. Pour une meilleure lisibilité, nous avons lissé les courbes en prenant les moyennes sur les ensembles de 50 documents avec le même ordre de traitement.

QL a donné d'excellents résultats, mais a montré une certain sensibilité au changement de paramètres. $\gamma = 0.9$ donne d'excellents résultats avec $\lambda = 0.95$ (Figure F.11a), où l'IA apprend une politique « assez bonne » après seulement 250 documents (les événements sont correctement extraits, mais elle prend quelques secondes de plus que la chaîne « experte »), et une politique optimale après 1 200 documents. $\gamma = 0.9$ donne aussi de très bons résultats avec $\lambda = 0$ (non montré). Cependant, avec $\gamma = 0.9$ et $\lambda = 1$, les résultats sont médiocres, et $\gamma = 1$ (Figure F.11b) donne de très mauvais résultats indépendamment de λ : l'IA a appris à s'arrêter très tôt, qui suggère une aversion au risque.

DOM, comme QL, a été sensible au choix de γ et λ. Avec $\gamma = 0.9$ et $\lambda = 1$ (non montré) et $\lambda = 0.95$ (Figure F.11f) les résultats sont très mauvais. Toutefois des résultats acceptables (non montré) ont été obtenus avec $\gamma = 0.9$ et $\lambda = 0$ pour la stratégie de réduction de ϵ la plus longue, et avec $\gamma = 0.9$ et $\lambda = 0$ ou $\lambda = 0.95$ pour la stratégie la plus courte. Avec $\gamma = 1$, les résultats allaient de bons à excellents, et la Figure F.11e montre les meilleurs résultats pour $\gamma = 1, \lambda = 1$, où une politique « assez bonne » est apprise après 1 000 documents et se stabilise à l'optimum après 3 750 documents (vérifié jusqu'à 10 000 documents — non montré).

MAG s'est révélé robuste au changement de paramètres, apprenant rapidement une politique au moins « assez bonne » dans tout les cas (*e.g.* la Figure F.11c).

Nous avons remarqué que le SSB Q-Learning (DOM ou MAG) apprend parfois une politique optimale mais que la performance se dégrade. Même si les événements sont toujours bien extraits, le traitement devient trop lent (voir les Figures F.11g, F.11h) : l'IA apprend qu'un appel à GATE depuis un état donné produit une bonne récompense, et si γ et λ ne sont pas correctement réglés, bien que cela prenne plus de temps, elle commence à préférer cette action à l'arrêt.

Enfin, nous avons constaté une légère amélioration des résultats avec une réduction plus rapide de ϵ (tous les 1 000 documents vs tout les 2 500), c'est-à-dire, avec globalement moins d'exploration.

En résumé, nous nous attendions à ce que QL soit plus efficace que MAG,

[2]Ainsi, bien que les courbes rouges représentent toujours la même approche appliquée sur les mêmes documents, elles n'ont pas toujours la même allure puisque seuls des extraits en sont présentés.

et MAG plus efficace que DOM, étant donné la quantité d'information qu'ils reçoivent. Nous constatons, cependant, que QL peut donner d'excellents résultats, mais est sensible à la variation de paramètres et dépend du *feedback* numérique. DOM ne nécessite qu'un *feedback* purement ordinal, et pourtant, avec les bons paramètres, est capable d'apprendre des politiques optimales, montrant la faisabilité de cette approche. MAG se révèle la meilleure approche dans un contexte industriel, combinant les avantages des deux méthodes : robustesse au choix des paramètres, utilisation du *feedback* essentiellement intuitif, et capacité à apprendre très rapidement une politique optimale.

F.9 Discussion et perspectives

Revenons à notre usine de voitures étrange, et observons comment ils ont réussi.

Elle a recruté une BIMBO comme chef de département, chargée de s'occuper des retours utilisateur. À son tour, celle-ci a recruté une équipe de choc d'IA. Chaque IA a été munie des plans de tous les modèles de voiture possibles. Ensuite, elle a commencé à organiser l'usine.

Les IA étaient rémunérés aux résultats, avec des primes pour des clients satisfaits. Afin de mesurer cette satisfaction client, ils ont émis un sondage à chaque fois qu'une voiture a été achetée. Quelques clients ne rendaient pas immédiatement leur formulaire, et la vaste majorité de clients ne l'ont jamais rempli. Cependant, les IA ont réussi à bien gagner leur vie, et en général, la satisfaction client a augmenté.

Puis, un petit futé à l'usine a suggéré le dispositif de contrôle continu, qui pourrait fonctionner en parallèle des sondages. Les IA étaient ravies parce qu'elles étaient toujours payées, même si les clients ne voulaient pas remplir un formulaire. Les clients étaient enthousiastes également. Ils n'avaient plus besoin de comparer la menthe et l'olive, ou l'asperge et l'avocat, ils pouvaient tout simplement dire « j'aime bien le vert, de préférence foncé ».

BIMBO s'est rendu compte que les IA, parfois, étaient un peu lentes à remplir la paperasse. Elle a donc recruté une assistante personnelle, Dora, qui les a aidées.

L'usine a connu un tel succès, qu'elle a diversifié ses activités en achetant une petite société d'analyse d'images.

F.9.1 Conclusion

Nous avons modélisé une chaîne de traitement de documents sous forme de processus de décision markovien, que nous avons résolu en utilisant l'apprentissage par renforcement. Pour implémenter cela, nous avons développé l'application modulaire *BIMBO (Benefiting from Intelligent and Measurable Behaviour Optimisation)* dans laquelle nous pouvons « brancher » différents algorithmes d'apprentissage par renforcement, services *web* et modèles afin de mesurer leur impact sur l'apprentissage.

Nous avons établi que notre approche donne de bons résultats avec un *feedback* numérique sporadique. Nous avons ensuite intégré une fonction de récompense formalisant des préférences utilisateur exprimées naturellement, permettant d'obtenir d'aussi bons résultats, tout en exigeant moins d'efforts cognitifs pour définir le *feedback*. Nous avons ainsi montré qu'il est possible d'aboutir à une chaîne de traitement capable de s'améliorer sans intervention ou réglage par un utilisateur humain, et qui obtient son *feedback* de façon non-intrusive.

Notre approche peut être utilisée pour d'autres traitements, par exemple la reconnaissance des objets dans une image. En effet, plusieurs types de capteurs et de nombreux algorithmes existent pour cela, et il n'est pas encore évident de les combiner optimalement. Nous avons appliqué notre approche, en utilisant la plateforme *BIMBO* et une variation de Q-Learning(λ), *Dora* (Nicart et al., 2016), aux tâches de segmentation (le pré-traitement de l'image) et de la *ROC* (reconnaissance optique de caractères). Notre but est la détection des éléments d'intérêt, pour suggérer automatiquement des titres depuis les pages de garde des documents. Spécifiquement, nous avons utilisé les caractéristiques des pages, leurs images, et les paramètres utilisés par les services pour définir les états du MDP. Les actions consistent à choisir les valeurs de ces paramètres, ou à envoyer l'image au traitement. Nous avons constaté des résultats très encourageants avec un espace de plus d'un million d'états.

F.9.2 Perspectives

Dans ce document, les ensembles d'états et d'actions ont été choisis pour démontrer la validité de notre approche. La prochaine étape consiste à produire des chaînes plus complexes, en ajoutant des services alternatifs (par exemple, la traduction), en étendant l'ensemble d'états (par exemple avec la liste complète de langues), et en introduisant une gamme de documents d'entrée plus large. Notre objectif est de montrer que le système est capable de construire

différentes chaînes pour différents types de documents. Même avec cet espace d'états / actions plus large, le temps de calcul ne sera pas un obstacle avec un algorithme de type Q-Learning, où les calculs sont instantanés à chaque pas de temps.

La plateforme *BIMBO* sous-tend notre application, mais peut également être utilisée pour l'étude de différentes méthodes d'amélioration continue, et/ou pour l'évaluation des approches principales de RL, de l'efficacité de différents algorithmes, et de l'impact de la variation des paramètres, afin de démontrer l'utilité de ces méthodes en milieu industriel.

Bibliography

Akrour, R., Schoenauer, M., and Sebag, M. (2011). Preference-based policy learning. In *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. Springer. [Cited on pages 16 and 203.]

Akrour, R., Schoenauer, M., and Sebag, M. (2012). APRIL: Active preference learning-based reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases*, volume 7524 of *Lecture Notes in Computer Science*, pages 116–131. Springer Berlin Heidelberg. [Cited on pages 16, 90, and 211.]

Akrour, R., Schoenauer, M., and Sebag, M. (2013). Interactive robot education. In *ECML/PKDD Workshop on Reinforcement Learning with Generalized Feedback: Beyond Numeric Rewards*. [Cited on pages 16 and 159.]

Amann, B., Constantin, C., Caron, C., and Giroux, P. (2013). WebLab PROV: Computing fine-grained provenance links for XML artifacts. In *BIGProv'13 Workshop (in conjunction with EDBT/ICDT)*, pages 298–306, Gênes, Italy. ACM. [Cited on pages 160 and 204.]

Azaria, A., Rabinovich, Z., Kraus, S., Goldman, C. V., and Gal, Y. (2012). Strategic advice provision in repeated human-agent interactions. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*. [Cited on pages 16 and 203.]

Baird, L. (1995). Residual algorithms: Reinforcement learning with function approximation. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 30–37. Morgan Kaufmann. [Cited on page 122.]

Bejan, C. and Harabagiu, S. (2014). Unsupervised event coreference resolution. *Comput. Linguist.*, 40(2):311–347. [Cited on page 11.]

Bellenger, A., Gatepaille, S., Abdulrab, H., and Kotowicz, J.-P. (2011). An Evidential Approach for Modeling and Reasoning on Uncertainty in Semantic Applications. In *URSW*, pages 27–38. [Cited on page 11.]

Bonet, B. and Pearl, J. (2002). Qualitative MDPs and POMDPs: An order-of-magnitude approximation. In *UAI '02, Proceedings of the 18th*

Conference in Uncertainty in Artificial Intelligence, University of Alberta, Edmonton, Alberta, Canada, August 1-4, 2002, pages 61–68. [Cited on page 16.]

Boutilier, C., Dearden, R., and Goldszmidt, M. (2000). Stochastic dynamic programming with factored representations. *Artificial Intelligence*, 121(1-2):49–107. [Cited on page 156.]

Brafman, R. I. and Tennenholtz, M. (2003). R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *The Journal of Machine Learning Research*, 3:213–231. [Cited on pages 96 and 210.]

Bratko, I. and Šuc, D. (2003). Learning qualitative models. *Artificial Intelligence*, 24(4):107. [Cited on pages 14, 31, 47, 154, 155, 161, and 201.]

Buades, A., Coll, B., and Morel, J.-M. (2011). Non-Local Means Denoising. *Image Processing On Line*, 1. [Cited on page 140.]

Busa-Fekete, R., Szörényi, B., Weng, P., Cheng, W., and Hüllermeier, E. (2014). Preference-based reinforcement learning: evolutionary direct policy search using a preference-based racing algorithm. *Machine Learning*, 97(3):327–351. [Cited on pages 15, 90, 92, and 211.]

Byrne, K. (2009). Populating the semantic web: combining text and relational databases as RDF graphs. [Cited on page 11.]

Camel (2015). Apache Camel. `http://camel.apache.org/`. Accessed: 2015-03-17. [Cited on pages 53 and 215.]

Caron, C., Amann, B., Constantin, C., Giroux, P., and Santanchè, A. (2014). Provenance-based quality assessment and inference in data-centric workflow executions. In *On the Move to Meaningful Internet Systems: OTM 2014 Conferences*, pages 130–147. Springer. [Cited on pages 160 and 204.]

Chai, X., Vuong, B.-Q., Doan, A., and Naughton, J. F. (2009). Efficiently incorporating user feedback into information extraction and integration programs. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 87–100. ACM. [Cited on page 14.]

Cheng, W., Fürnkranz, J., Hüllermeier, E., and Park, S.-H. (2011). Preference-based policy iteration: Leveraging preference learning for reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases*, pages 312–327. Springer. [Cited on page 15.]

Cheng, W. and Hüllermeier, E. (2008). Learning similarity functions from qualitative feedback. In *Advances in Case-Based Reasoning*, pages 120–134. Springer. [Cited on page 69.]

Chiticariu, L., Li, Y., and Reiss, F. R. (2013). Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 827–832. [Cited on pages 10 and 46.]

Cichosz, P. (1995). Truncating temporal differences: On the efficient implementation of TD (lambda) for reinforcement learning. *CoRR*, cs.AI/9501103. [Cited on page 122.]

Coggan, M. (2004). Exploration and exploitation in reinforcement learning. Technical report, CRA-W DMP Project at McGill University, Canada. [Cited on page 155.]

Cohen, W., Ravikumar, P., and Fienberg, S. (2003). A comparison of string metrics for matching names and records. In *Kdd workshop on data cleaning and object consolidation*, volume 3, pages 73–78. [Cited on page 11.]

Cohen, W. W., Dalvi, B. B., and Cohen, B. D. W. W. (2013). Very Fast Similarity Queries on Semi-Structured Data from the Web. In *SDM*, pages 512–520. [Cited on pages 11, 66, and 217.]

Cowie, J. and Lehnert, W. (1996). Information extraction. *Commun. ACM*, 39(1):80–91. [Cited on page 10.]

Culotta, A., Kristjansson, T., Mccallum, A., and Viola, P. (2006). Corrective feedback and persistent learning for information extraction. *Artificial Intelligence*, 170(14-15):1101–1122. [Cited on pages 14 and 160.]

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Thomas Heitz, Mark A. Greenwood, Horacio Saggion, Johann Petrak, Yaoyong Li, and Wim Peters (2014). Developing Language Processing Components with GATE Version 8 (a User Guide). `https://gate.ac.uk/sale/tao/tao.pdf`. Accessed: 2014-12-17. [Cited on page 215.]

Dahl, F. A. and Halck, O. M. (2001). *Learning While Exploring: Bridging the Gaps in the Eligibility Traces*, pages 73–84. Springer Berlin Heidelberg, Berlin, Heidelberg. [Cited on pages 84, 123, and 137.]

Dean, T. and Givan, R. (1997). Model minimization in Markov decision processes. In *AAAI/IAAI*, pages 106–111. [Cited on page 156.]

Degris, T., Sigaud, O., and Wuillemin, P.-H. (2006). Learning the structure of factored markov decision processes in reinforcement learning problems. In *Proceedings of the 23rd international conference on Machine learning*, pages 257–264. ACM. [Cited on page 154.]

Dini, S. and Serrano, M. (2012). Combining q-learning with artificial neural networks in an adaptive light seeking robot. Technical report, Computer Science Department, Swarthmore College, USA. [Cited on page 156.]

Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., and Weischedel, R. (2004). The automatic content extraction (ACE) program-tasks, data, and evaluation. [Cited on page 11.]

Doshi-Velez, F. and Ghahramani, Z. (2011). A comparison of human and agent reinforcement learning in partially observable domains. *CogSci*. [Cited on page 33.]

Doucy, J., Abdulrab, H., Giroux, P., and Kotowicz, J.-P. (2008). Méthodologie pour l'orchestration sémantique de services dans le domaine de la fouille de documents multimédia. In *Proceedings of MajecSTIC 2009*. [Cited on pages 12, 158, and 203.]

Dutkiewicz, J., Jędrzejek, C., Cybulka, J., and Falkowski, M. (2013). Knowledge-based highly-specialized terrorist event extraction. *RuleML2013 Challenge, Human Language Technology and Doctoral Consortium*, page 1. [Cited on pages 11, 66, and 217.]

Epshteyn, A. and DeJong, G. (2006). *Qualitative reinforcement learning*, volume 2006, pages 305–312. [Cited on pages 16 and 17.]

Fishburn, P. C. (1984). SSB utility theory: an economic perspective. *Mathematical Social Sciences*, 8(1):63 – 94. [Cited on pages 92 and 212.]

Formiga, L., Barrón-Cedeño, A., Màrquez, L., Henríquez, C. A., and Mariño, J. B. (2015). Leveraging online user feedback to improve statistical machine translation. *Journal of Artificial Intelligence Research*, 54:159–192. [Cited on pages 14, 156, 160, and 204.]

Fort, K. (2012). *Les ressources annotées, un enjeu pour l'analyse de contenu: vers une méthodologie de l'annotation manuelle de corpus*. PhD thesis, Université Paris-Nord-Paris XIII. [Cited on page 10.]

Fromherz, M. P., Bobrow, D. G., and De Kleer, J. (2003). Model-based computing for design and control of reconfigurable systems. *AI magazine*, 24(4):120. [Cited on pages 12 and 203.]

Fürnkranz, J. and Hüllermeier, E. (2011). *Preference Learning*. Springer-Verlag Berlin Heidelberg. [Cited on page 15.]

Fürnkranz, J., Hüllermeier, E., Cheng, W., and Park, S.-H. (2012). Preference-based reinforcement learning: a formal framework and a policy iteration algorithm. *Machine Learning*, 89(1-2):123–156. [Cited on pages 15, 90, 91, and 211.]

Fürnkranz, J., Hüllermeier, E., Rudin, C., Slowinski, R., and Sanner, S. (2014). Preference learning (dagstuhl seminar 14101). *Dagstuhl Reports*, 4(3):1–27. [Cited on page 15.]

Gardner, M., Talukdar, P., Krishnamurthy, J., and Mitchell, T. (2014). Incorporating vector space similarity in random walk inference over knowledge bases. In *Proceedings of EMNLP*. [Cited on page 11.]

GATE (2016). GATE Information Extraction. `https://gate.ac.uk/ie/`. Accessed: 2016-06-20. [Cited on pages 24, 26, and 205.]

Geonames (2015). Geonames. `http://www.geonames.org/`. Accessed: 2015-03-17. [Cited on pages 51, 58, and 207.]

Gilbert, H., Spanjaard, O., Viappiani, P., and Weng, P. (2015a). *Reducing the Number of Queries in Interactive Value Iteration*, pages 139–152. Springer International Publishing, Cham. [Cited on page 89.]

Gilbert, H., Spanjaard, O., Viappiani, P., and Weng, P. (2015b). Solving MDPs with Skew Symmetric Bilinear Utility Functions. In *24th International Joint Conference on Artificial Intelligence (IJCAI-15)*, pages 1989–1995, Buenos Aires, Argentina. [Cited on pages 92 and 93.]

Gilbert, H., Zanuttini, B., Weng, P., Viappiani, P., and Nicart, E. (2016). Model-free reinforcement learning with skew-symmetric bilinear utilities. In Ihler, A. and Janzing, D., editors, *Proc. 32nd Conference on Uncertainty in Artificial Intelligence (UAI 2016)*, pages 252–261. AUAI Press. [Cited on pages 21, 98, 99, 103, 104, 117, 209, 212, and 229.]

Ginstrom, R. (2007). The GITS Blog: Fuzzy substring matching with Levenshtein distance in Python. `http://ginstrom.com/scribbles/2007/12/01/`

fuzzy-substring-matching-with-levenshtein-distance-in-python/. Accessed: 2014-08-19. [Cited on pages 67 and 217.]

Global Terrorism Database (2016a). CODEBOOK: INCLUSION CRITERIA AND VARIABLES. https://www.start.umd.edu/gtd/downloads/Codebook.pdf. Accessed: 2016-10-07. [Cited on page 61.]

Global Terrorism Database (2016b). National Consortium for the Study of Terrorism and Responses to Terrorism (START). https://www.start.umd.edu/gtd, note = Accessed: 2016-10-07. [Cited on page 59.]

Guan, J. and Qiu, G. (2007). Modeling user feedback using a hierarchical graphical model for interactive image retrieval. In *Advances in Multimedia Information Processing-PCM 2007*, pages 18–29. Springer. [Cited on page 155.]

Hobbs, J. R. and Riloff, E. (2010). Information extraction. In Indurkhya, N. and Damerau, F. J., editors, *Handbook of Natural Language Processing, Second Edition*. CRC Press, Taylor and Francis Group, Boca Raton, FL. ISBN 978-1420085921. [Cited on page 10.]

Hoey, J., St-Aubin, R., Hu, A., and Boutilier, C. (1999). SPUDD: Stochastic planning using decision diagrams. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 279–288. Morgan Kaufmann Publishers Inc. [Cited on page 156.]

Hogenboom, F., Frasincar, F., Kaymak, U., and De Jong, F. (2011). An Overview of Event Extraction from Text. *Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011) at Tenth International Semantic Web Conference (ISWC 2011)*. [Cited on page 10.]

INVESTOPEDIA (2016). INVESTOPEDIA. www.investopedia.com. Accessed: 2016-10-07. [Cited on page 40.]

JAPE (2016). JAPE: Regular Expressions over Annotations. https://gate.ac.uk/sale/tao/splitch8.html. Accessed: 2016-10-07. [Cited on pages 46 and 52.]

Ji, H. (2010). Challenges from information extraction to information fusion. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 507–515. Association for Computational Linguistics. [Cited on page 10.]

Ji, H. and Grishman, R. (2008). Refining Event Extraction through Cross-Document Inference. In *ACL*, pages 254–262. Citeseer. [Cited on page 11.]

Kanani, P. H. (2012). *Resource-bounded Information Acquisition and Learning*. PhD thesis, University of Massachusetts Amherst. [Cited on pages 13 and 158.]

Karami, A. B., Sehaba, K., and Encelle, B. (2014). Apprentissage de connaissances d'adaptation à partir des feedbacks des utilisateurs. In *25es Journées francophones d'Ingénierie des Connaissances*, pages 125–136. [Cited on pages 14, 154, and 204.]

Knox, W. B. and Stone, P. (2015). Framing reinforcement learning from human reward: Reward positivity, temporal discounting, episodicity, and performance. *Artificial Intelligence*, 225:24–50. [Cited on pages 17, 103, 113, and 203.]

Kumaran, G. and Allan, J. (2004). Text classification and named entities for new event detection. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 297–304. ACM. [Cited on page 11.]

Kumaran, G. and Allan, J. (2005). Using names and topics for new event detection. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 121–128. Association for Computational Linguistics. [Cited on page 11.]

Kurakin, A., Goodfellow, I. J., and Bengio, S. (2016). Adversarial examples in the physical world. *CoRR*, abs/1607.02533. [Cited on page 160.]

LaFree, G. (2010). The Global Terrorism Database: Accomplishments and Challenges | LaFree | Perspectives on Terrorism. *Perspectives on Terror*, 4(1). [Cited on pages 59 and 216.]

Lao, N. and Cohen, W. W. (2010). Relational retrieval using a combination of path-constrained random walks. *Machine learning*, 81(1):53–67. [Cited on page 11.]

Lao, N., Subramanya, A., Pereira, F., and Cohen, W. W. (2012). Reading the web with learned syntactic-semantic inference rules. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1017–1026. Association for Computational Linguistics. [Cited on page 11.]

Lee, A., Passantino, M., Ji, H., Qi, G., and Huang, T. (2010). Enhancing multi-lingual information extraction via cross-media inference and fusion. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 630–638. Association for Computational Linguistics. [Cited on page 10.]

Lejeune, G. (2013). *Veille épidémiologique multilingue: une approche parcimonieuse au grain caractere fondée sur le genre textuel*. PhD thesis, Université de Caen. [Cited on page 10.]

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965. [Cited on page 67.]

Lin, L.-J. (1991). Programming robots using reinforcement learning and teaching. In *AAAI*, pages 781–786. [Cited on page 135.]

Linguistic Data Corporation (LDC) (2016). ACE. `https://www.ldc.upenn.edu/collaborations/past-projects/ace`. Accessed: 2016-10-07. [Cited on page 11.]

Loftin, R., Peng, B., MacGlashan, J., Littman, M. L., Taylor, M. E., Huang, J., and Roberts, D. L. (2015). Learning behaviors via human-delivered discrete feedback: modeling implicit feedback strategies to speed up learning. *Autonomous Agents and Multi-Agent Systems*, 30(1):30–59. [Cited on pages 15, 48, and 203.]

Ludovic, J.-L. (2011). *Approches supervisées et faiblement supervisées pour l'extraction d'événements complexes et le peuplement de bases de connaissances*. PhD thesis. [Cited on page 10.]

Ma, J. and Powell, W. B. (2009). A Convergent Recursive Least Squares Approximate Policy Iteration Algorithm for Multi-Dimensional Markov Decision Process with Continuous State and Action Spaces. In *2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 66–73. IEEE. [Cited on page 156.]

Ma, Z. and Kwiatkowska, M. (2008). *Modelling with PRISM of intelligent system*. phdthesis. [Cited on page 13.]

Matthew, G. (2015). Using technology recycling to develop a named entity recogniser for afrikaans. *Southern African Linguistics and Applied Language Studies*, 33(2):199–216. [Cited on page 91.]

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*. [Cited on pages 40 and 138.]

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533. [Cited on page 40.]

Moore, A. W. and Atkeson, C. G. (1993). Prioritized sweeping: Reinforcement learning with less data and less time. *Machine Learning*, 13(1):103–130. [Cited on pages 136 and 138.]

Moreau, E., Yvon, F., and Cappé, O. (2008). Robust similarity measures for named entities matching. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 593–600. Association for Computational Linguistics. [Cited on page 11.]

Ng, A. Y. and Russell, S. J. (2000). Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 663–670, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. [Cited on page 15.]

NGramJ (2015). NGramJ, smart scanning for document properties. `http://ngramj.sourceforge.net/`. Accessed: 2015-02-18. [Cited on pages 24, 26, and 205.]

Nguyen, A. M., Yosinski, J., and Clune, J. (2014). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *CoRR*, abs/1412.1897. [Cited on page 160.]

Nicart, E., Zanuttini, B., Grilhères, B., and Praca, F. (2016). Dora Q-learning - making better use of explorations. In Pellier, D., editor, *Proc. 11es Journées Francophones sur la Planification, la Décision et l'Apprentissage pour la conduite de systèmes (JFPDA 2016)*. [Cited on pages 203 and 232.]

Nydal, H. M. (2016). Deep q-learning for stock trading. `http://hallvardnydal.github.io/new_posts/2015-07-21-deep_q/`. Accessed: 2016-10-07. [Cited on page 40.]

Ogrodniczuk, M. and Przepiórkowski, A. (2010). Linguistic Processing Chains as Web Services: Initial Linguistic Considerations. In *Proceedings of the Workshop on Web Services and Processing Pipelines in HLT: Tool Evaluation, LR Production and Validation (WSPP 2010) at the Language Resources and Evaluation Conference (LREC 2010)*, pages 1–7. [Cited on pages 12, 24, and 200.]

Ong, S. C. W., Png, S. W., Hsu, D., and Lee, W. S. (2010). Planning under uncertainty for robotic tasks with mixed observability. *International Journal of Robotics Research*, 29(8):1053–1068. [Cited on page 157.]

Orwell, G. (1950). *1984*. Tandem Library, centennial. edition. [Cited on page 90.]

Pandit, S., Gupta, S., and others (2011). A comparative study on distance measuring approaches for clustering. *International Journal of Research in Computer Science*, 2(1):29–31. [Cited on pages 66 and 217.]

Pang, W. and Coghill, G. M. (2010). Learning qualitative differential equation models: a survey of algorithms and applications. *Knowledge Eng. Review*, 25:69–107. [Cited on page 154.]

Peng, J. and Williams, R. J. (1993). Efficient learning and planning within the dyna framework. *Adaptive Behavior*, 1(4):437–454. [Cited on page 136.]

Peng, J. and Williams, R. J. (1996). Incremental multi-step q-learning. *Machine Learning*, 22(1-3):283–290. [Cited on page 122.]

Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc. New York, NY, USA. [Cited on pages 34 and 208.]

Rao, K. and Whiteson, S. (2011). *V-MAX: A General Polynomial Time Algorithm for Probably Approximately Correct Reinforcement Learning*. PhD thesis, Amsterdam. [Cited on pages 96 and 210.]

Reyes, A., Ibargüengoytia, P. H., Sucar, L. E., and Morales, E. F. (2006). Abstraction and refinement for solving continuous markov decision processes. In Studený, M. and Vomlel, J., editors, *Probabilistic Graphical Models*, pages 263–270. [Cited on page 17.]

Riloff, E. (1996). Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, pages 1044–1049. AAAI Press. [Cited on page 11.]

Rodrigues, F., Oliveira, N., and Barbosa, L. (2015). Towards an engine for coordination-based architectural reconfigurations. *Computer Science and Information Systems*, 12(2):607–634. [Cited on pages 12 and 203.]

Saaty, T. L. (2008). Relative measurement and its generalization in decision making why pairwise comparisons are central in mathematics for the measurement of intangible factors the analytic hierarchy/network process. *RACSAM-Revista de la Real Academia de Ciencias Exactas, Fisicas y Naturales. Serie A. Matematicas*, 102(2):251–318. [Cited on page 14.]

Sabbadin, R. (1999). A possibilistic model for qualitative sequential decision problems under uncertainty in partially observable environments. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pages 567–574, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. [Cited on page 16.]

Saïs, F., Serrano, L., Khefifi, R., and Scharffe, F., editors (2013). *Proceedings of SOS-DLWD 2013*. [Cited on pages 11 and 203.]

Sander, A., Buşoniu, L., and Babuška, R. (2012). Experience replay for real-time reinforcement learning control. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(2):201–212. [Cited on page 136.]

Saval, A. (2011). *Temporal pattern, spatial and semantic for the discovery of relationships between events*. Theses, Université de Caen. [Cited on page 11.]

Schaefer, A. J., Bailey, M. D., Shechter, S. M., and Roberts, M. S. (2005). Modeling medical treatment using markov decision processes. In *Operations research and health care*, pages 593–612. Springer. [Cited on page 13.]

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. (2015). Prioritized experience replay. *CoRR*, abs/1511.05952. [Cited on pages 135 and 136.]

Serrano, L. (2014). *Vers une capitalisation des connaissances orientée utilisateur: extraction et structuration automatiques de l'information issue de sources ouvertes*. PhD thesis, Université de Caen. [Cited on pages v, 11, 52, 58, 205, and 206.]

Serrano, L., Bouzid, M., Charnois, T., Brunessaux, S., and Grilheres, B. (2013). Events extraction and aggregation for open source intelligence: from text to knowledge. *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, pages 518–523. [Cited on page 11.]

Serrano, L., Charnois, T., Brunessaux, S., Grilheres, B., Bouzid, M., and others (2012). Combinaison d'approches pour l'extraction automatique d'événements. In *19e conférence sur le Traitement Automatique des Langues Naturelles (TALN 2012)*, pages 423–430. [Cited on page 10.]

Shelton, C. R. (2000). Balancing multiple sources of reward in reinforcement learning. In *Neural Information Processing Systems-2000*, pages 1082–1088. [Cited on page 17.]

Shteingart, H. and Loewenstein, Y. (2014). Reinforcement learning and human behavior. *Current Opinion in Neurobiology*, 25:93–98. [Cited on page 33.]

Sigaud, O. and Buffet, O. (2010). *Markov Decision Processes in Artificial Intelligence*. Wiley-IEEE Press. [Cited on page 136.]

Singh, S. P. (1992). Transfer of learning by composing solutions of elemental sequential tasks. *Machine Learning*, 8(3-4):323–339. [Cited on page 158.]

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231. [Cited on page 14.]

So, M. M. and Thomas, L. C. (2011). Modelling the profitability of credit cards by markov decision processes. *European Journal of Operational Research*, 212(1):123 – 130. [Cited on page 13.]

Southey, R. (1837). *The story of the three bears*. The Doctor. [Cited on page 104.]

Steele, R. D. (1995). The importance of open source intelligence to the military. *International Journal of Intelligence and Counter Intelligence*, 8(4):457–470. [Cited on page 23.]

Steinberger, R., Pouliquen, B., and Van der Goot, E. (2013). An introduction to the europe media monitor family of applications. *CoRR*, abs/1309.5290. [Cited on page 11.]

Stewart, I. (1995). *Concepts of Modern Mathematics*. Dover Books on Mathematics. Dover Publications. [Cited on page 44.]

Strehl, A. L., Diuk, C., and Littman, M. L. (2007). Efficient structure learning in factored-state MDPs. In *AAAI*, volume 7, pages 645–650. [Cited on page 154.]

Strehl, A. L., Li, L., Wiewiora, E., Langford, J., and Littman, M. L. (2006). PAC model-free reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 881–888. ACM. [Cited on page 137.]

Sutton, R. J. and Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT press. [Cited on pages xiii, 33, 85, 103, 121, 122, 128, 134, 136, 156, 208, and 212.]

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. (2013). Intriguing properties of neural networks. *CoRR*, abs/1312.6199. [Cited on page 160.]

Szepesvári, C. (2010). Algorithms for reinforcement learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 4(1):1–103. [Cited on pages 35 and 209.]

Tambet, M. (2016). Guest post: Demystifying deep reinforcement learning - nervana. `https://www.nervanasys.com/demystifying-deep-reinforcement-learning/`. Accessed: 2016-06-08. [Cited on page 137.]

Taylor, M. E. and Stone, P. (2009). Transfer learning for reinforcement learning domains: A survey. *The Journal of Machine Learning Research*, 10:1633–1685. [Cited on page 16.]

Tika (2015). Apache Tika - a content analysis toolkit. `http://tika.apache.org/`. Accessed: 2015-02-18. [Cited on pages 24, 26, and 205.]

Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., and Ristenpart, T. (2016). Stealing machine learning models via prediction apis. *CoRR*, abs/1609.02943. [Cited on page 160.]

Trampus, M. and Mladenic, D. (2014). Constructing domain templates with concept hierarchy as background knowledge. *Information Technology And Control (ITC)*, 43(4):414–432. [Cited on page 154.]

Travé-Massuyès, L., Ironi, L., and Dague, P. (2003). Mathematical foundations of qualitative reasoning. *AI magazine*, 24(4):91. [Cited on page 90.]

Tversky, A. and Gati, I. (1978). Studies of similarity. *Cognition and categorization*, 1(1978):79–98. [Cited on pages 66 and 217.]

Van Hage, W. R., Malaisé, V., Segers, R., Hollink, L., and Schreiber, G. (2011). Design and use of the Simple Event Model (SEM). *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(2):128–136. [Cited on pages 11 and 205.]

Van Hasselt, H. (2010). Double q-learning. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *NIPS*, pages 2613–2621. Curran Associates, Inc. [Cited on page 137.]

Van Hasselt, H., Guez, A., and Silver, D. (2015). Deep reinforcement learning with double q-learning. *CoRR*, abs/1509.06461. [Cited on page 137.]

Veeramachaneni, K., Arnaldo, I., Cuesta-Infante, A., Korrapati, V., Bassia, C., and Li, K. (2016). AI2: Training a big data machine to defend. In *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*, pages 49–54. [Cited on page 15.]

Wang, J., Li, G., Yu, J. X., and Feng, J. (2011). Entity matching: How similar is similar. *Proceedings of the VLDB Endowment*, 4(10):622–633. [Cited on page 11.]

Wang, W., Xiao, C., Lin, X., and Zhang, C. (2009). Efficient approximate entity extraction with edit distance constraints. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 759–770. ACM. [Cited on page 11.]

Wang, Y.-H., Li, T.-H. S., and Lin, C.-J. (2013). Backward Q-learning: The Combination of Sarsa Algorithm and Q-learning. *Eng. Appl. Artif. Intell.*, 26(9):2184–2193. [Cited on page 122.]

Watkins, C. J. C. H. (1989). *Learning From Delayed Rewards*. PhD thesis, Kings College. [Cited on pages 39, 71, 122, 135, and 209.]

WebLab (2016a). OSINT challenges. `http://weblab-project.org/index.php?title=Features`. Accessed: 2016-10-07. [Cited on page 23.]

WebLab (2016b). WebLab downloads. `http://weblab-project.org/index.php?title=WebLab_1.2.5/Released_Elements`. Accessed: 2016-10-07. [Cited on pages 24 and 51.]

WebLab (2016c). WebLab wiki. `http://weblab-project.org/`. Accessed: 2016-10-07. [Cited on pages 24 and 204.]

Weng, P. (2011). Markov Decision Processes with Ordinal Rewards: Reference Point-Based Preferences. In *ICAPS*. [Cited on page 89.]

Weng, P., Busa-Fekete, R., and Hüllermeier, E. (2013). Interactive Q-Learning with Ordinal Rewards and Unreliable Tutor. *ECML/PKDD Workshop Reinforcement Learning with Generalized Feedback*. [Cited on page 89.]

Weng, P. and Zanuttini, B. (2013). Interactive value iteration for markov decision processes with unknown rewards. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2415–2421. AAAI Press. [Cited on pages 89 and 90.]

Wiering, M. and Schmidhuber, J. (1998). Speeding up Q(λ)-learning. In *Machine Learning: ECML-98*, pages 352–363. Springer. [Cited on pages 136 and 137.]

Wilson, A., Fern, A., and Tadepalli, P. (2012). A bayesian approach for policy learning from trajectory preference queries. In *Advances in neural information processing systems*, pages 1133–1141. [Cited on pages 16, 90, and 211.]

Wirth, C. and Fürnkranz, J. (2013a). EPMC: Every visit preference monte carlo for reinforcement learning. In *Asian Conference on Machine Learning, ACML 2013, Canberra, ACT, Australia, November 13-15, 2013*, pages 483–497. [Cited on pages 90 and 211.]

Wirth, C. and Fürnkranz, J. (2013b). Preference-based reinforcement learning: A preliminary survey. In *Proceedings of the ECML/PKDD-13 Workshop on Reinforcement Learning from Generalized Feedback: Beyond Numeric Rewards*. [Cited on page 90.]

Wirth, C., Fürnkranz, J., and Neumann, G. (2016). Model-free preference-based reinforcement learning. In *Proceedings of the 30-th AAAI Conference on Artificial Intelligence (AAAI-16)*, pages 2222–2228. [Cited on pages 90 and 211.]

Wyatt, J. (1998). *Exploration and inference in learning from reinforcement.* PhD thesis, University of Edinburgh. College of Science and Engineering. School of Informatics. [Cited on pages 134 and 136.]