Explanations in the information extraction system iDocument

Björn Forcher, Benjamin Adrian, Thomas Roth-Berghofer

The information extraction system iDocument interactively extracts information from texts such as instances and relations with respect to existing background knowledge. An extraction process creates weighted hypotheses describing indications of relevant information. During execution, each process step records its output into an instantiated process model. We reused these bits of information for generating conceptual, functional as well as causal explanations. In order to visualise explanations, our component utilises different mechanisms for textual, explorative, and pictorial rendering styles.

1 Introduction

The need for explanations in computer science has been recognised since the first generation of expert systems. The main goal of explanations in expert systems was to justify the results of reasoning processes supporting the user's decision processes and increasing trust in results. The goal concerns not only expert systems but also knowledge based systems in general.

The notion of explanation is used in different situations and has several aspects, building a complete notion family [3]. For instance, explanations are used to describe the causality of events or to describe the semantics of concepts. Explanations help correcting mistakes or they serve as moral justifications. Finally, they are used to describe functionalities.

Expert systems are designed to solve problems similar to human experts in certain domains. In the first generation of expert systems explanations are recognised as key feature explaining solutions and reasoning processes. Hence, explanation facilities were an important component to support the user's needs and decisions [4]. Often explanations were nothing more than (badly) paraphrased rules because important aspects were missing or too much information was given [1]. In order to improve on dialogues, second generation systems focused on context, goals and actions, methods and justifications to support explanations, together with an even richer knowledge representation.

Spieker [2] distinguishes several kinds of explanation, which we adopt in this paper. *Conceptual explanations* concern questions such as "What is...?", or "What is the semantics of...?". The purpose of these explanations is to establish cognitive links between unknown and known concepts. The goal of *why-explanations* is to explain the cause or the justification for a fact or a current situation whereas *how-explanations* are a special case of why-explanations. They describe processes leading to an event using a causal chain.

In this paper we describe explanations used in the knowledge-based information extraction system iDocument[1]. Based on a knowledge base (KB), iDocument generates recommendations for annotating documents with relevant information and finally for populating the KB with extracted information, namely instances and relations. As recommendations as such are not self-explanatory, the explanation facility of iDocument offers different styles of explanations: natural language, graphics and tabled visualisations.

The rest of this paper is organised as follows. In the next section, we introduce iDocument before we describe its explanation facilities in Section 3. Finally, we briefly summarise the results of our work.

2 iDocument

iDocument extracts relevant information from text with the help of a knowledge base. The extraction process is based on a complex, cascading of extraction tasks, of which the two steps *instantiation* and *contextualisation* are especially interesting for the user. The extraction process comprises the following steps:

Normalisation: Extracts document metadata and the plain text data from textual or binary file formats.

Segmentation: Partitions plain text into segments: paragraphs, sentences, and tokens.

Symbolisation: Resolves known tokens or token sequences as symbols and unknown noun phrases as named entities.

Instantiation: Recognises symbols as occurrences of single instances, which are resolved, unified, and disambiguated. If an entity cannot be mapped to an existing instance, it is classified as new instance.

Contextualisation: Resolves recognised relations between instances inside the document with respect to existing contexts, namely the user and his question.

Population: Evaluates extracted instances, attributes, and relations. Valid information is populated into the knowledge base.

iDocument aims at assisting users with valuable recommendations for a given text. Each extraction task creates recommendations in form of hypotheses with weights representing belief values. The hypotheses are conceptualised using an application ontology. Existing hypotheses may lead to new hypotheses in proceeding tasks. As a result, a complex mesh of hypotheses describes final recommendations concerning document and knowledge base. In spite of this complexity, the user interface of iDocument is kept slender and provides a quick overview of extracted recommendations.

[1] http://idocument.opendfki.de/

Auszug aus: Künstliche Intelligenz, Heft 2/2008, ISSN 0933-1875, BöttcherIT Verlag, Bremen, **www.kuenstliche-intelligenz.de/order**