Engineering Applications of Artificial Intelligence 30 (2014) 1–17

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Intuitive justifications of medical semantic search results

Björn Forcher [a,*], Thomas Roth-Berghofer [b], Stefan Agne [c], Andreas Dengel [c]

[a] Leibniz Institute for Psychology Information (ZPID), Germany
[b] School of Computing and Technology, University of West London, United Kingdom
[c] Knowledge Management Department, German Research Center for Artificial Intelligence (DFKI) GmbH, Germany

ARTICLE INFO

Article history:
Received 13 March 2013
Received in revised form
8 November 2013
Accepted 22 January 2014
Available online 18 February 2014

Keywords:
Explanation
Justification
Understandability
Semantic
Search

ABSTRACT

To some extent, explanations in computer science are answers to questions. Often an explanatory dialogue is necessary to satisfy needs of software users. In this paper, we introduce the concept of *intuitive explanation* representing the first explanations in an explanatory dialogue. This kind of explanation does not require a situational context to be established or that there is a user model. Depending on an abstract model of explanation generation we present the generic explanation component *Kalliope* applying Semantic Technologies to construct intuitive explanations. We illustrate our generation approach by means of the semantic search engine *KOIOS++* enabling keyword-based search on medical articles. Since semantic search results are often hard to understand *Kalliope* was integrated into *KOIOS++* in order to justify search results. In this work we describe in detail the construction of intuitive explanations for inexperienced users in the medical domain building on the concepts of *Semantic Frequency Classes* and *Semantic Cooccurrence Classes*. Various user experiments illustrate that these concepts enable the explanation component to rate the understandability of labels and of label connections. We show how *Kalliope* exploits these valuations to construct and select understandable explanations.

1. Introduction

In a certain sense, explanations in computer science can be understood as answers to questions (Roth-Berghofer and Richter, 2008) and often an explanatory dialogue (Du et al., 2006) is required to support users of a software system to make sense of its results. In this paper, we introduce the concept of *intuitive explanation*, which can be illustrated by means of an everyday scenario. Imagine, a 9-year-old child complains about dizziness and visits the doctor for an examination. After the examination the doctor concludes that the child suffers from *Mènière's disease*. The child does not know this particular disease. Thus, the doctor explains it as the child requires an explanation to understand the correlation of dizziness and Mènière's disease. The doctor will not elaborate on the Mènière's disease or use complicated terminology. On the contrary, she will roughly estimate the knowledge of the child and, based on that, give a short explanation. In other words, she relies on her intuition and experience with children of that age. *Mènière's disease is a disease of the inner ear which causes, for instance, dizziness* (probably supported by an illustration or an anatomical model of the ear) is an understandable explanation, enabling the child to process the given information and ask further questions. In contrast, the explanation that *Mènière's disease is a dysfunction of the vestibular system* is hardly understandable for a 9-year-old and therefore not helpful to the child. In the figurative sense, *intuitive explanations* represent the first explanations in an explanatory dialogue in which the explainer tries to give an understandable explanation based on a rough estimation of the knowledge of the communication partner. In addition, we assume that the situational context is unknown and hence, the explanation component has to enable follow-up questions leading to a complex explanatory dialogue. Note that intuitive explanations are not intended to be perfect, but they serve as an entrance into a dialogue. We believe that a complex explanation need of a user can only be solved in explanatory dialogue (compare to Walton, 2007).

In this work, we illustrate the construction of intuitive explanations for the semantic search engine *KOIOS++* (Forcher et al., 2010a). *KOIOS++* provides keyword-based search on graph-shaped RDF data. Among other things it searches for Wikipedia[1] articles that are annotated with medical concepts of medical structured data, namely the RadLex[2] ontology and a representation of the International Classification of Diseases, Version 10 (ICD-10) (Möller et al., 2010). For example, the Wikipedia article about the shoulder blade can be annotated with the anatomical concepts

* Corresponding author.
E-mail address: bjoern.forcher@dfki.de (B. Forcher).

[1] http://www.wikipedia.org/.
[2] http://www.radlex.org/.

0952-1976/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.engappai.2014.01.013

'radlex:acromion' or 'radlex:clavicle'. These concepts can then be used to retrieve the respective article. In this context, the search algorithm exploits the structure of the ontology to find articles that are annotated with the same or adjacent concepts.

Since semantic search results are not always selfexplanatory, the explanation facility *Kalliope* is integrated into *KOIOS++* revealing a connection between search and annotation concepts and employs the same structured data as *KOIOS++*. The constructed explanations are depicted as semantic networks containing various domain specific concepts. As said the above intuitive explanations should be brief and contain preferably only understandable information. This applies to first, labels of concepts, and second, the number of concepts. In case of RadLex and ICD10 it is possible to display concepts with more than one label. In addition, various rules can be applied to generate further connections between concepts and thus many different explanations can be constructed to justify one and the same search result. The question is only whether users can understand the connection between the concepts or concept labels or not. For instance, the statement 'The finger is part of the upper limb' is easy to understand, but the statement 'The acromion is part of the human body' may need further information (and is of questionable utility).

In this work, we present in detail the construction of understandable justifications for medical laypeople. The corresponding algorithm includes a method that is able to predict the understandability of labels and label connections by considering their usage frequency in natural language, regarding different levels of expertise (Forcher et al., 2009).

The rest of the paper is structured as follows: The next section gives an overview of relevant research on explanations and the ensuing section describes an abstract explanation generation approach. Section 4 introduces the search engine *KOIOS++* and motivates its explanation need. Section 5 describes user experiments and resulting concepts which constitute the groundwork to valuate explanations. Section 6 presents the construction algorithm of *Kalliope* in detail. The paper concludes with a summary and outlook.

2. On explanation

The notion of explanation has several aspects when used in daily life (Passmore, 1962). For instance, explanations are used to describe the causality of events or the semantics of concepts. Explanations help correct mistakes or serve as justifications. Furthermore, they are used to describe functionalities or to communicate practical knowledge. In the case of explaining semantic search results users are usually not interested in explanations of the functionalities and search algorithms. Instead, they need a justification to decide whether the search result is relevant and trustworthy.

Explanations in computer science were introduced in first generation Expert Systems (ES). They were recognised as a key feature explaining solutions and reasoning processes, especially in the domain of medical expert systems such as MYCIN (Buchanan and Shortliffe, 1984) where trust in results matters. Several approaches for providing explanations in expert systems were developed. XPLAIN (Swartout, 1983), for instance, is a tool that helps users to build expert systems containing explanation components. For this reason, it uses domain facts for explanation purposes. The second form of input is a collection of domain principles, which are the methods or algorithms that apply to the facts. This system refines the domain knowledge (preserving the knowledge) until it is at an appropriate level for the implementation of an expert system. An extension is ESS (Explainable Expert System, see Swartout and Smoliar, 1987). Here knowledge about concepts and concept classes can be formulated. In the next section we present the requirements of *Kalliope* regarding various kinds of knowledge to generate explanations.

Explanation facilities were an important component supporting the user's needs and decisions (Swartout and Smoliar, 1989; Swartout et al., 1991). In those early systems, explanations were often nothing more than (badly) paraphrased rules that lacked important aspects or too much information was given at once (Richards, 2003). For that reason, Swartout and Moore formulated five desiderata for expert system explanations (Swartout and Moore, 1993) which generally apply to knowledge-based systems (KBS). We considered especially the desiderata understandability and feedback.

1. *Fidelity*: The explanation must be an accurate representation of what the KBS really does. Hence, an explanation has to build on the same knowledge which the system uses for its reasoning.
2. *Understandability*: This comprises various factors such as *user sensitivity* and *Feedback*. User sensitivity addresses the user's goals and preferences, but also his knowledge with respect to the system and the corresponding domain. Feedback is very important because users do not necessarily understand a given explanation at once. The system should offer certain kinds of dialogue so that users can learn about the parts they do not understand.
3. *Sufficiency*: The system has to know what it is talking about. This is an important factor to enable some kind of dialogue with the user.
4. *Low construction overhead*: It should not be too costly to integrate an explanation component into a KBS.
5. *Efficiency*: The explanation component should not affect the performance of the whole system.

Wick and Thompson (1992) developed the Reconstructive Explainer (REX) and the concept of *reconstructive explanations* for ES. A trace, *i.e.,* a *line of reasoning*, is transformed into a plausible explanation story, *i.e.,* a *line of explanation*. The transformation is an active, complex problem-solving process using additional domain knowledge. The degree of coupling between the trace and the explanation is controlled by a filter which can be set to one of the four states, regulating the transparency of the filter. The more information of the trace is let through the filter, the more closely the line of explanation follows the line of reasoning. This approach enables a disengagement of an explanation component in order to reuse it in other expert systems. In a certain sense, the construction algorithm of *Kalliope*, using rating methods, represents some kind of filter that distinguishes intuitive from unintuitive explanations.

Another dimension guiding our research is the notion of explanation goals. Sørmo and Cassens (2004) describe different explanation goals for case-based reasoning systems, also applicable to knowledge based systems in general.

1. *Justification*: Explain why a system's answer is a good answer. It is used to give a simpler explanation according to the system process.
2. *Transparency*: Explaining how the system calculates a certain answer allows users to understand and (better) control the system.
3. *Relevance*: In conversational systems this goal aims at why a question asked by the software system is relevant.
4. *Learning*: In Tutoring Systems or Decision Support Systems it is important to teach users about the respective domain.

Focusing on each of these goals helps a software designer to focus on developing certain explanation capabilities. Explanation-Aware Software Design (EASD) looks at ways to guide software