SemAcom: A System for Modeling with Semantic Autocompletion

Henning Agt

Database Systems and Information Management Group DIMA
Technische Universität Berlin
Einsteinufer 17
10587 Berlin, Germany
henning.agt@tu-berlin.de

Abstract. Autocompletion is a feature in various applications that assists users while typing into input fields. It predicts the user input based on background knowledge, rules and heuristics and is successfully used in mobile phones, web browsers, search engines and integrated development environments. We demonstrate SemAcom, a system for automatically suggesting related terms for domain-specific models. During the development of Ecore models with the Eclipse Modeling Framework SemAcom provides context-sensitive pop-up lists with related class names. The backbone of our system is a large-scale network of semantically related terms that was created automatically from a natural language dataset. The network contains probabilistic information how strong terms are related and is used to retrieve ranked lists of semantically related terms for a given set of input terms. That allows us to present the most relevant terms at the top of the list of suggestions.

Key words: Domain-Specific Language, Terminology Extraction, Autocompletion, Semantic Relatedness

1 Introduction

SemAcom is developed in the context of the research project BIZWARE[1], a collaboration of two academic and eight industrial partners to investigate the potential of domain-specific languages and model-driven engineering for small and medium enterprises. While the industrial partners develop domain-specific languages in their respective domains (healthcare, manufacturing/production, finance/insurance, publishing and facility management), the main tasks of the academic partners is the development of methodologies, guidelines and tools to support DSL development.

SemAcom aims at providing semantic modeling support for the development of domain-specific models [1, 2]. It provides a feature for language workbenches

[1] This work is partially supported by the Bundesministerium für Bildung und Forschung BMBF under grant number 03WKBU01A.

or domain-specific modeling environments that everyone knows from search engines: When entering keywords suggestions are automatically triggered that are similar to the one the user is typing. Search engine suggestions usually rely on natural language statistics, previous user searches and term popularity. In order to achieve such a support for a domain-specific modeling environment we need to automatically identify a corresponding set of semantically related terms for a given set of input terms (class names), present it to the user and filter it while typing.

2 Autocompletion Based on an Automatically Created Semantic Network

In order to provide a search engine-like behavior for creating new classes in a modeling environment the most relevant terms according to current model must appear first in the list of suggestions. Otherwise, the autocompletion would only hinder the developer. Therefor, we require a interconnected dictionary of terms, a semantic network, that is large enough to cover almost all possible terms in any domain and that contains information on the degree of relatedness between terms.

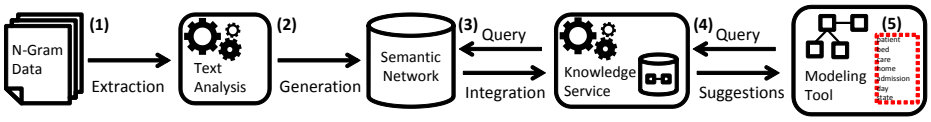
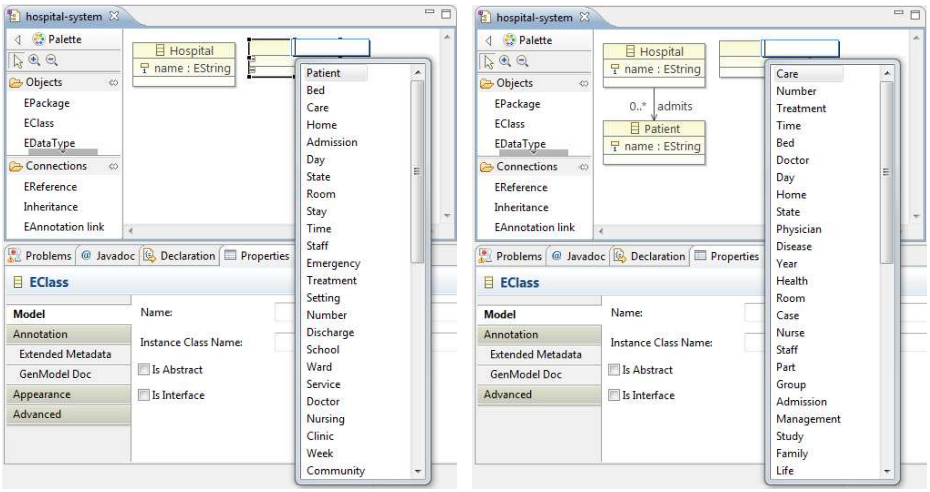


Fig. 1: Creating a semantic network of terms and using it for domain-specific modeling.

Figure 1 shows our approach of providing model autocompletion functionality. (1) We use the Google Books n-gram dataset [3] as input to extract terms and their relations. The dataset was derived from 5 million scanned books (360 billion English words) and contains CSV files of raw frequencies of word sequences (n-grams). For example, it contains the information that the sequence *"the doctor and the patient"* occurred 6765 times in the complete text corpus. (2) We apply text analysis (lexical patterns and part-of-speech tagging) on the complete dataset in order to extract terms and their degree of semantic relatedness. The analysis is based on the *Distributional Hypothesis*. It describes that "words which are similar in meaning occur in similar contexts" [4]. (3) The process is completely automated and the generated semantic network is stored in a database. It contains over 3.4M terms and 38M weighted relations between them and requires only 1.5GB of disk space. (4) The knowledge service internally queries the database for ranked lists of related terms for a given input term and computes probabilities in case multiple terms are queried. It provides a simple interface to our semantic network that any modeling tool can use to display suggestions (5).

3 SemAcom Implementation

The automated creation of the SemAcom network is implemented in Java using the Stanford POS Tagger and SQLite JDBC. The prototype for autocompletion is implemented as a set of Eclipse plug-ins. A model listener reacts to changes in Ecore models under development. We extended the Ecore Diagram Editor with a content proposal adapter to display the suggestions. Figure 2 shows autocompletion pop-up lists for two models. On the left, possible new class names for *Hospital* are displayed. On the right, the class *Patient* was added. SemAcom adjusts the suggestions based on both terms.



(a) Suggestions for a single class *Hospital*. (b) Suggestions for two classes *Hospital* and *Patient*.

Fig. 2: SemAcom showing ranked lists of possible semantically related terms.

References

1. Agt, H.: Supporting Software Language Engineering by Automated Domain Knowledge Acquisition. In: MODELS 2011 Workshops. LNCS, vol. 7167. Springer, Wellington, New Zealand (2012)
2. Agt, H., Kutsche, R.D., Wegeler, T.: Guidance for Domain Specific Modeling in Small and Medium Enterprises. In: SPLASH '11 Workshops. ACM (2011)
3. Michel, J.B., Shen, Y.K., Aiden, A.P., Veres, A., Gray, M.K., Team, T.G.B., Pickett, J.P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., Pinker, S., Nowak, M.A., Aiden, E.L.: Quantitative Analysis of Culture Using Millions of Digitized Books. Science 331(6014), 176–182 (Jan 2011)
4. Rubenstein, H., Goodenough, J.B.: Contextual correlates of synonymy. Commun. ACM 8(10), 627–633 (Oct 1965)