Semantic Autocompletion

Eero Hyvönen and Eetu Mäkelä
FirstName.LastName@tkk.fi

Helsinki University of Technology, Media Technology
Helsinki Institute for Information Technology (HIIT), and University of Helsinki
http://www.cs.helsinki.fi/group/seco/

Abstract. This paper generalizes the idea of traditional syntactic text autocompletion onto the semantic level. The idea is to autocomplete typed text into ontological categories instead of words in a vocabulary. The idea has been implemented and its application for semantic indexing and content-based information retrieval is proposed. Four operational semantic portals using the implementation are presented, two of which are publicly available on the web.

1 Introduction

The idea of *autocompletion*[1] is to predict what the user is typing in, and to complete the work automatically. The benefits of this simple idea are manyfold: First, the computer helps the user in memorizing the right vocabulary used. Second, typing errors in the input can be minimized. Third, autocompletion speeds up the interaction. A side effect of the idea is that it encourages the usage of long descriptive names and commands that are more understandable to the users. An idea related to autocompletion is *autoreplace*, where the idea is to use predefined abbreviations in typing and the system automatically replacing these with full-blown strings.

In order to make the prediction right and as early as possible, the underlying vocabulary must be known, be limited, and the words in the lexicon should differentiate in terms of the leading characters. These conditions hold in many applications, such as operating system shells, email programs, browsers, etc.

In Microsoft's Intellisense feature of the Visual Studio, the idea is applied to source code editing. Here a pop-up menu is used to show the programmer possible autocompleted forms, where it is difficult to remember or type in, e.g., the names of the methods of a particular class at hand. A widely used application of autocompletion is the predictive text entry system in mobile phones [1, 2] commonly known as T9, where only a limited number of keys are available instead of a full QWERTY keyboard. By associating each key with a set of letters (e.g. '1' with a, b, and c) and by completing single keypresses automatically based on a dictionary, input speeds up significantly e.g. in text messaging.

[1] See e.g. http://en.wikipedia.org/wiki/Autocompletion

Autocompletion can be done *by request* or *on-the-fly*. In Linux/Unix and DOS operating systems, for example, the command line is completed—or possible continuations are shown— only after a hit on the TAB-key. The on-the-fly-approach is used e.g. in browsers and email-systems: the text typed in is completed into matching URLs or email addresses that have been used before, or are stored in an address book. A nice recent application of autocompletion on-the-fly on the web is the beta version of Google Suggest[2] that completes input text into feasible search keywords.

Traditional autocompletion is based on matching input strings with a list of usable words in a vocabulary. This paper generalizes this approach onto the semantic level. The idea is to complete user written text not into other strings, but into matching concepts based on an ontology. In the following, this idea to be called *semantic autocompletion* is first discussed. After this, an implementation of the idea in the OntoViews framework[3] is presented, and its application in three semantic portals for concept-based information retrieval and in semantic indexing is discussed.

2 From Syntactic to Semantic Completion

Semantic autocompletion completes text on the *semantic level*, i.e., the autocompletion function tries to predict what different concepts in a vocabulary the user is trying to address. For example, if the user types in the word 'bank', then such a system could complete the input into concepts 'river bank', 'bank (financial)', 'pankki' (bank in Finnish), 'pankit' (bank in plural in Finnish) or 'Nordea' (a major Nordic bank). Semantic autocompletion does not necessary mean that the typed string appears in the label of the concept (cf. 'Nordea' above), the appearance may not occur in the beginning of the label (cf. 'river bank'), or the word may be written in a different language (cf. 'pankki' above) or be morphologically inflected (cf 'pankit' above). The user can, however, understand the semantic connection between typed text and the completion.

The benefits of the idea include the following:

1. Synonyms can be found. For example, 'cat' $\mapsto$ 'feline mammal'.
2. Word senses can be disambiguated. For example, 'Nokia' $\mapsto$ 'Nokia Corp.', 'Nokia' $\mapsto$ 'Nokia town (Finland)', 'Nokia' $\mapsto$ 'Nokia (character in F.E. Sillanpää's novel)'.
3. Multilingual autocompletion. The keywords can be expressed in different languages and be matched on the same concept. For example, 'bank (financial)' $\mapsto$ 'pankki (Finnish)'.
4. Showing related ontological choices. The ontological environment of the matches can be visualized in addition to the actual matches. For example, by showing the subsumption hierarchy leading to the matched concept, the user can complete the text into the superclass or related concepts. For example, 'bank' $\mapsto$ 'financial institution'.

[2] http://www.google.com/webhp?complete=1&hl=en

5. Integrating search with autocompletion. Autocompletion can be extended from matching ontological categories to matching the actual annotated metadata, i.e., the items searched for.

In the following we show how these ideas have been realized in practise in semantic information retrieval and indexing.

3 Semantic Category Search: Case MuseumFinland

Autocompletion can be used to disambiguate meanings in queries. This is useful especially if the content searched for has been annotated using correspondingly disambiguated concepts. An example of such a system is the semantic portal MuseumFinland [4]. We have incorporated a version of semantic autocompletion into this application.

MuseumFinland integrates semantic autocompletion with multi-facet view-based search [5, 6]. The search keywords are matched not only against the actual textual item descriptions, but also the labels and descriptions of the ontological categories by which they are annotated and organized into the view facets. As a result, a new dynamic facet is created in the user interface. This facet contains all categories whose name or other configurable property values, such as additional names, match the keyword. Intuitively, these facet categories tell the different interpretations of the keyword. By selecting one of them in the dynamic facet, a semantically disambiguated choice and query can be made.

The result of a sample keyword search is shown in figure 1. Here, a search for "nokia" has matched, for example, the categories Nokia (the telephone company as a manufacturer, view 'Valmistaja'), Nokia (the town as a place of manufacture, view 'Valmistuspaikka', and as a place of usage, view 'Käyttöpaikka') and Nokia-Mobira (an earlier incarnation of the telephone company). The categories found can be used to constrain the multi-facet search as usual, with the distinction that selections from the dynamic facet replace selections in their corresponding facets and dismiss the dynamic facet.

Fig. 1. Using the keyword search for finding categories.

In MuseumFinland, semantic autocompletion can be seen as search over a set of RDF(S) categories (although also hit lists are generated at the same time). This idea has been applied also, e.g., in the Open Directory Project[3] search engine. However, in our case the 9 categories have been projected, using a set of logical rules, from a set of 7 underlying ontologies in the system knowledge base. Matching is not straight-forward because of the projection, but more flexible. For example, in the search results of figure 1, the category Nokia (the place) appears twice. This is because the categories can appear in the content of the portal in different roles. Therefore, simply choosing e.g. the category 'Nokia (the place)' would not disambiguate the meaning sufficiently, since the same resource can have either the role 'Place of Manufacture' or 'Place of Usage', or both, in the metadata of a museum artifact. In the case of MuseumFinland, these roles can be disambiguated automatically by semantic autocompletion: the user can choose from a list of given options the correct role meaning of the keyword 'nokia' indicated by the subcategory path leading to it.

4 Semantic Autocompletion on the Fly: Case Orava

In MuseumFinland autocompletion is done on request, i.e., after pushing the search button. We have also created an on-the-fly version of the idea and applied it to another semantic portal Orava[4]. This portal provides the user with semantic search and browsing facilities similar to MuseumFinland but to a database of some 2100 video and audio clips[5] and learning object metadata (LOM)[6] related to them. In addition, the material was semantically linked to the RDF(S) metadata of MuseumFinland.

Figure 2 depicts the home page of the portal with the on-the-fly semantic autocompletion in action in the upper right corner. The user has typed in the characters 'mat', aiming perhaps at the word 'matkailu' (travel). The autocompletion function dynamically and automatically updates the category trees below as selectable links. It shows all facet categories matching the typed characters used in the multi-facet search. The facets, such as 'Oppiaine' (learning subject) and 'Teema' (theme), and their uppermost levels of subcategories are seen on the left hand side column.

Continuing by typing the letter 'k' would eliminate the category 'matematiikka' (mathematics) as no longer matching, updating the trees accordingly. Alternately, at any point the user can select a link in the dynamic facet, and the system retrieves all material related to the selected category or any of its subcategories. The presentation of the retrieved categories as trees gives the user the context necessary to make informed selections, as well as makes it possible to make a broader search by selecting some supercategory of the ones matched.

---

[3] http://dmoz.org/

[4] http://www.museosuomi.fi/orava/

[5] The material is from the Klaffi portal (http://www.yle.fi/klaffi/) of the Finnish Broadcasting Company YLE.

[6] http://ltsc.ieee.org/wg12/

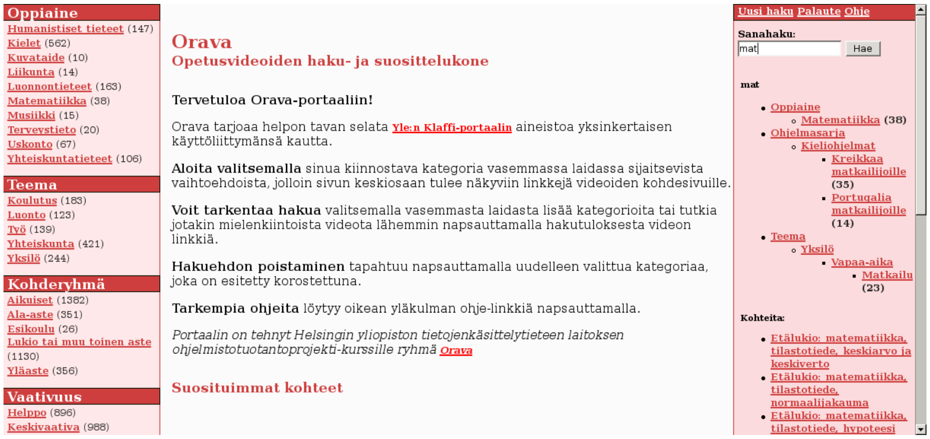

Fig. 2. Semantic autocompletion on-the-fly in Orava.

Below the dynamic category tree, a dynamic hit list that consists of the union of all video and audio clips matching 'mat' is also shown for the direct selection of a particular item. As in MuseumFinland, autocompletion is here extended to actually searching the contents, but on the fly this time.

5 Semantic Autocompletion in Multi-Facet Search: Case Veturi

In the semantic yellow page portal Veturi [7] created in the Intelligent Web Services (IWebS) project[7], the integration between view hierarchy -based search and on-the-fly semantic autocompletion is taken even further. For this portal, on-the-fly semantic autocompletion was chosen as the central user interface element. The portal makes ample use of otherwise invisible metadata to match typed-in keywords to categories, as will be shown below.

Figure 3 depicts the search interface of the Veturi portal. The five view-facets used in the portal (Consumer, Producer, Target, Process and Location of the Service provided) are located at the top, initially marked only by their name and an empty keyword field. Typing search terms in the fields immediately opens the corresponding facet to show matching categories available for selection. After such a selection, the facet closes again, showing only what was selected, while the results view below the facets dynamically updates to show relevant hits. For quick searches, a globally effective keyword search box is provided in the upper left corner of the interface.

The example search depicted in figure 3 shows a fictive user trying to find out where he can buy rye bread in Helsinki. He has already selected Helsinki as

[7] http://www.cs.helsinki.fi/group/iwebs/

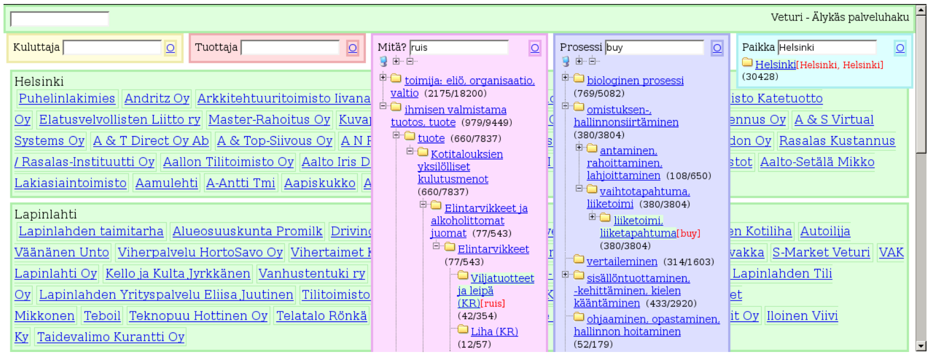

Fig. 3. Semantic autocompletion on-the-fly in Veturi.

the locale for the services he requires. Now, he is in the process of describing the actual service.

In the view 'Mitä?' (service target), the user has typed in the word 'ruis' (rye). While the annotation ontology used does not contain different grains, the concept 'Viljatuotteet ja Leipä (KR)' (grain products and bread) contains a textual reference to rye, resulting in a category match. In this way, existing textual material can be used to augment incomplete ontologies to at least return some hits for concepts that have not yet been added to the ontology. Showing such hits in their ontological context allows for easy spotting of irrelevant hits and close misses, where for example the keyword matches a subcategory of a more appropriate one.

The search query entered in the view 'Prosessi' (Process) divulges another feature of semantic autocompletion: multilanguage support. Typing in the word 'buy' matches the appropriate business transaction, even though the word for 'buy' in Finnish would be 'ostaa'.

6 Semantic Indexing: Case ONKI Ontology Server

ONKI [8] is a part of the "Finnish National Ontologies on the Semantic Web" (FinnONTO)[8] framework project. Its goal is to support the development and use of nationally shared ontologies in order to enhance semantic interoperability on the Finnish semantic web. A central part of FinnONTO research deals with providing ontology services through public web services. For a content indexer, the ONKI ontology server provides a web-based browser for finding desired concepts. Semantic autocompletion has been implemented as a part of a demonstrational ONKI service. The interface is analogous to the one in the Orava portal. After locating a concept, it can be selected by clicking on it and the corresponding URI can be used for accurate semantic indexing, and automatically transported from ONKI into external applications via a web service interface.

[8] http://www.cs.helsinki.fi/group/seco/ontologies

7 Implementation

All of the portals covered are based on the portal tool OntoViews [3] , and therefore also share the same implementation of semantic autocompletion. This implementation consists of two clearly separated components: the actual search engine and the user interface elements. This way, it is possible to switch the implementation of one component without affecting the other, if desired.

In the current implementation, the actual keyword matching is done on the server by Ontogator, the main view-based search engine of OntoViews. This gives the benefit of direct access to the view hierarchies to be used in visualization, as well as tight integration with the main multi-facet search facilities of the engine. The implementation was also extended to support T9-type ambiguous numerical queries [1, 2]. This was done to demonstrate how semantic autocompletion can easily be combined with other advances in predictive text autocompletion, because the ontological navigation happens separately after string matching, similarly to the approach described in [9].

In MuseumFinland the user interface elements are static HTML, but all the newer on-the-fly implementations make use of AJAX[9] techniques. Depending on the complexity of the user interface, the returned content is either simple HTML to be added to the page, or JavaScript code to be executed in the context of the page.

8 Discussion

This paper introduced the idea of semantic autocompletion based on ontologies as a natural extension to traditional autocompletion based on string matching. While in the end also semantic autocompletion often boils down to string matching, from the end-users viewpoint the matching occurs rather on the semantic level. In many cases the matched concept could not be found using straightforward string matching, but was found by utilizing the logic of the metadata in the underlying knowledge base.

Our implementation and practical application of the idea to multi-facet search in semantic portals suggest that such semantic autocompletion should be of practical value on the semantic web. The facet hierarchies give the user an overview of what kind of information there is in the vocabulary of the data repository and guide the user in formulating the query in terms of appropriate concepts. Furthermore, the hierarchies do not suffer from the problems of homonymous query terms: the right meaning is determined by the context.

Dealing with large and deep hierarchies is major bottleneck of the multi-facet search paradigm. According to user tests [10], keyword search is usually preferred over multi-facet search if the user is capable of expressing her information need terms of accurate keywords. Semantic autocompletion makes it easier to the

[9] Asynchronous JavaScript and the XMLHttpRequest-object, which allow for making HTTP calls to the server in the background while viewing a page. See e.g. http://en.wikipedia.org/wiki/AJAX

end-user to deal the wealth of categories used in facets. The value of the idea comes from the integration of benefits from the keyword-based, single-facet, and multi-facet search paradigms.

Acnowledgements

This work was mainly supported by the National Technology Agency Tekes. Samppa Saarela designed and implemented the server-side search engine used in the OntoViews framework, and Teppo Känsälä integrated autocompletion on-the-fly with Orava.

References

1. Dunlop, M.D., Crossan, A.: Predictive text entry methods for mobile phones. Personal Technologies **4** (2000)
2. Hasselgren, J., Montnemery, E., Nugues, P., Svensson, M.: Hms: A predictive text entry method using bigrams. In: Proceedings of the Workshop on Language Modeling for Text Entry Methods, 10th Conference of the European Chapter of the Association of Computational Linguistics, Budapest, Hungary, Association for Computational Linguistics (2003) 43–49
3. Mäkelä, E., Hyvönen, E., Saarela, S., Viljanen, K.: Ontoviews—a tool for creating semantic web portals. In: Proceedings of the 3rd International Semantic Web Conference (ISWC 2004), Hiroshima, Japan, Springer–Verlag, Berlin (2004) 797–811
4. Hyvönen, E., Junnila, M., Kettula, S., Mäkelä, E., Saarela, S., Salminen, M., Syreeni, A., Valo, A., Viljanen, K.: Finnish Museums on the Semantic Web. User's perspective on MuseumFinland. In: Proceedings of Museums and the Web 2004 (MW2004), Seleted Papers, Arlington, Virginia, USA. (2004) http://www.archimuse.com/mw2004/papers/hyvonen/hyvonen.html.
5. Pollitt, A.S.: The key role of classification and indexing in view-based searching. Technical report, University of Huddersfield, UK (1998) http://www.ifla.org/IV/ifla63/63polst.pdf.
6. Hearst, M., Elliott, A., English, J., Sinha, R., Swearingen, K., Lee, K.P.: Finding the flow in web site search. CACM **45** (2002) 42–49
7. Laukkanen, M., Viljanen, K., Apiola, M., Lindgren, P., Hyvönen, E.: Towards ontology-based yellow page services. In: Proceedings of WWW2004 Workshop on Application Design, Development, and Implementation Issues in the Semantic Web, New York, USA, CEUR Workshop Proceedings, Vol-105 (2004) http://ceur-ws.org.
8. Valo, A., Komulainen, V., Hyvönen, E.: A collaborative ontology development and service framework onki. In: Proceedings of Int. Conf. on Dublin Core and Metadata Application (DC-2005), short papers, Madrid. (2005)
9. Legrand, S., Tyrväinen, P., Saarikoski, H.: Bridging the word disambiguation gap with the help of OWL and semantic web ontologies. In: Proceedings of the Workshop on Ontologies and Information Extraction, Eurolan 2003. (2003) 29–35
10. English, J., Hearst, M., Sinha, R., Swearingen, K., Lee, K.P.: Flexible search and navigation using faceted metadata. Technical report, University of Berkeley, School of Information Management and Systems (2003)