Semantic Autocompletion

Eero Hyvönen and Eetu Mäkelä

Semantic Computing Research Group (SeCo)
Helsinki University of Technology (TKK), Laboratory of Media Technology
University of Helsinki, Department of Computer Science
FirstName.LastName@tkk.fi
http://www.seco.tkk.fi/

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 in multi-facet search is proposed. Four operational semantic portals on the web using the implementation are presented as application cases.

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 replaces 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 differ from each other in terms of the leading characters. These conditions hold in many applications, such as operating system shells, email programs, browsers, etc.

Autocompletion is used, e.g., in Microsoft's Intellisense feature of the Visual Studio, where the idea is applied to source code editing. Here a pop-up menu is used to show the programmer possible autocompleted forms. This is useful when 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 the full QWERTY keyboard. By associating each key with a set of letters (e.g. '1' with a, b, and c)

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

and by completing single keypresses automatically based on a dictionary, input typing can be speeded up significantly e.g. in text messaging.

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—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 only into similar words, but into matching ontological concepts whose labels may not be related to the input on the literal level. For example, the typed input 'preside...' could be autocompleted into 'George W. Bush' since George W. Bush is an instance of the class president. It is also possible to complete the input text into the different homonymous meanings (concepts) of the input, and into the different semantic roles in which the concepts are used. This possibility provides the end-user not only with a semantic matching service but can be used to disambiguate the meanings and thematic roles in which the concepts are used. To continue the example above, input 'preside...' could be autocompleted into 'George W. Bush (as an author)' or 'George W. Bush (as a document subject)'. By providing the autocompleted choices to the end-user, the right interpretation can be disambiguated and, for example, search be performed with the right meaning.

In the following, this idea to be called *semantic autocompletion* is first discussed as a means for semantic information retrieval, and some of its different forms are identified. After this, implementation of the idea in the OntoViews framework [3] is presented, and application in three semantic portals for concept-based information retrieval and in semantic indexing is exemplified.

2 From Syntactic to Semantic Completion

We consider the idea of semantic autocompletion in information retrieval, especially, in multi-facet search [4–7]. Multi-facet search is a generalized form of the traditional single-facet search paradigm. Examples of single-facet systems include Yahoo!, Open Directory Project[3], and many traditional web portals. In multi-facet search, content is organized and retrieved using multiple hierarchical structures at the same time, instead of just one like in single-facet search.

2.1 Autocompletion in multi-facet search

In multi-facet systems the data has been indexed using keywords from a set of hierarchical orthogonal facet categories. For example, in [5] the facet categories

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

of the Art and Architecture Thesaurus AAT[4] are used as subject terms. The location facet divides the earth into continents (Africa, Antarctica, Asia, ...), each continent consists of countries, and each country is divided further into counties, cities, etc. The material facet is a classification hierarchy of materials used or depicted in the collection items. The search objects are classified along facets based on the keywords used in annotating the collection items. The user selects categories from different facets and the search result is the intersection of the items belonging to the selected categories. By selecting a supercategory, all hits related to its subcategories (recursively) are returned, too. Let mapping $m : S \rightarrow C$ map each search item $s \in S$, where S is the set of search times, to the set of facet categories C. Then the hit set H corresponding to selected search categories $c_1, ..., c_n$ is $H = \{s | c_i \in m(s), i = 1, ..., n\}$.

In traditional multi-facet search, the keywords are strings as usual in keyword search. In [6] multi-facet search is extended with semantic web ontology techniques and reasoning. The idea is to replace keywords with ontological resources in indexing and then determine the mapping m between search categories and search items using logical mapping rules. In this way, multi-facet search can be generalized onto a semantic level where the mapping between facets and search items can be based on semantic relations and not only on simple keyword match. For example, in [8] the category 'Nokia' as a company in an actor facet is mapped onto different search items than 'Nokia' as a city in Finland in a location facet.

Semantic autocompletion in multi-facet search can be defined as a function $f : text \rightarrow < C, H >$ that maps an input string $t \in text$ onto a set of search categories of the facets C and the corresponding search item hits H in the data set. The hits are based on the different semantic meanings of the input. For example, if the user types in the word 'bank', this could be completed into categories 'river bank' and 'bank (financial)' and the result set includes an union of both geographical and organizational hits.

The input may consist of several partly written keywords that correspond to category selections. For example, 'Finl presid' could mean that the user searches information about categories 'Finland' and 'president', e.g., about the presidents of Finland. The categories C and hits H matching the input should, in the user's view, match in meaning with the intended meanings of *text*. For example, input 'Scandin...' may match the category 'Nordic countries'. Notice that here 'Scandinavia' and 'Nordic countries' do not share substrings as required in traditional autocompletion. In our case, autocompletion is occuring on the semantic level in the user's mind, and is implemented using the underlying ontological structures.

Autocompleting an input string into facet categories can be based on several principles. In below, some forms of autocompletion are discussed.

2.2 Autocompletion based on equivalence relations

This form of autocompletion deals with the problems of lexical variants, synonymy, polysemy, and homonymy. Lexical variants and synonyms are alterna-

[4] http://www.getty.edu/research/conducting_research/vocabularies/aat/about.html

tive terms that correspond to the same ontological concept. For example, 'NYC', 'New York City', and 'Big Apple' refer to the same city. Semantic autocompletion can provide a service, where typing in any of the terms is completed into the same concept, denoted by its preferred term, here 'New York City'.

This kind of autocompletion can be enabled to some extent by listing alternative and preferred labels for concepts. If the input matches any of these, the corresponding concept is selected, and the preferred label is shown to the user. However, in morphologically rich languages, such as a Finnish, listing all morphological variants as explicit alternatives may not be feasible, and dynamic morphological analysis may be needed as a part of autocompletion before ontological matching. For example, the genitive plural for of the Finnish word 'yö' (night) is 'öiden', a literal quite different from the nominative form.

In polysemy, a single term has different but related meanings (e.g., 'arrow head' and 'human head'); in homonymy the meanings are totally different (e.g., 'river bank' and 'blood bank'). In both cases, the meaning cannot be disambiguated based on the user's shorthand input ('head' or 'bank'). The same happens when the user's partial input can be completed in different ways (e.g., 'New' $\mapsto$ 'New York' or 'New' $\mapsto$ 'New Year'. In these cases the autocompletion function can provide the user with a list of possible choices from which to disambiguate.

One problem in determining the equivalence between input text and categories is how to deal with phrasal concept labels, such as 'broadband integrated services digital network'. Here, the categories can be matched against all permutations, and only the combinations leading to actual hits returned, so that for example the search can return the two-category combination 'Integrated Services + Digital Network (11 hits)' as a reasonable autocompletion, while the two-category combination 'Broadband Integration + Digital Services (0 hits)' is left out. In such complex multi-word labels, words may also appear in morphologically conjugated forms, which makes pattern matching more difficult, again possibly requiring morphological analysis as a pre-step. On the user interface level, one must also remember that particularly for compound words the matching part may not necessarily begin the input string, so that the prefix matching is not sufficient, but the whole string needs to be scanned for matches.

In multilingual autocompletion the keywords can be expressed in different languages and be matched on the same concept. This facilitates multilingual search even when the actual data is available or has been indexed in one only language. For example, 'bank (financial)' $\mapsto$ 'pankki (Finnish)'.

A benefit of semantic autocompletion is that the ontological environment of the matched categories can be visualized in addition to the actual matches. By showing the category hierarchy leading to the matched concept, the user can easily understand the meaning of the different completions. Furthermore, she can complete the text into the superclass or related concepts. For example, 'bank' $\mapsto$ 'financial institution > bank', where '>' indicates the subclass relation in the hierarchy.

2.3 Indirect semantic autocompletion

Semantic completion can be extended beyond equality to other semantic relations. The input string can be matched with not only the corresponding equivalent category, but with other related categories, too. For example, assume that you are looking for information about countries. By typing in 'EU' or 'US' semantic autocompletion could complete the text into a choice list of member countries of EU or states of the US, respectively, saving the effort of memorizing their names. Here the isPartOf-relation to is used for completing the text into neighboring ontological resources. However, in principle any arbitrarily complex relation could be used here, as long as its interpretation is intuitive and of use to the end-user.

2.4 Semantic role completion

An application of semantic autocompletion is *semantic role completion.* Here we not only match the input text with categories but also take into account the roles in which the categories are used. For example, the same city may be related with a museum collection artifact either as the place of manufacture or as the place of usage in the metadata. Depending on the choice, different result sets are obtained (unless all relevant items are both manufactured and used in the same place). Semantic autocompletion can provide the user with the possible choices to disambiguate.

2.5 Semantic autocompletion search

Semantic autocompletion can be combined seamlessly with semantic search. By completing the input string not only in related categories but also into the actual hits in the underlying data set, the user can actually see the hit list to narrow down as she types in text.

3 Application of Semantic Autocompletion

In the following we show by examples from various case studies, how the different forms of semantic autocompletion can be realized in practise in semantic information retrieval and indexing.

3.1 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[5] [7]. We have incorporated a version of semantic autocompletion into this application.

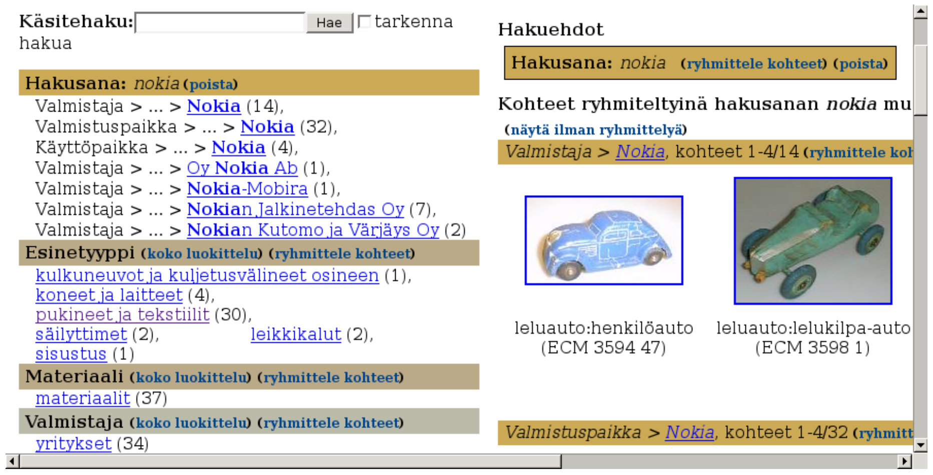
[5] http://www.museosuomi.fi

MuseumFinland integrates semantic autocompletion with multi-facet search. 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 of semantic autocompletion, a new "dynamic facet" is created in the user interface. This facet contains all categories whose name or other configurable property value, such as alternative labels, match the keyword. Intuitively, the dynamic facet categories tell 1) the different interpretations of the keyword and 2) their roles with respect to the search items (here museum collection artifacts) in the metadata.

The result of a sample keyword search is shown in figure 1. Here, a search for input "nokia" has matched, for example, the following view categories:

– 'Nokia' as the telephone company and a manufacturer in the view Manufacturer ('Valmistaja' in the screenshot),
– 'Nokia' as a town in the view Place of Manufacture ('Valmistuspaikka'),
– 'Nokia' as a town in the view Place of Usage ('Käyttöpaikka'), and
– 'Nokia-Mobira', a predecessor of the telephone company, in the view Manufacturer.

By default, search is done by using the union of all possible interpretations. Search results are shown and classified according the possible choices on the right in the figure. However, the categories found can be used to constrain the multi-facet search further, with the distinction that selections from the dynamic facet replace selections in their corresponding facets and dismiss the dynamic facet. The right interpretation is selected by clicking on the corresponding link in the dynamic facet.



Käsitehaku: Hae ☐ tarkenna hakua

Hakusana: *nokia* (poista)
Valmistaja > ... > Nokia (14),
Valmistuspaikka > ... > Nokia (32),
Käyttöpaikka > ... > Nokia (4),
Valmistaja > ... > Oy Nokia Ab (1),
Valmistaja > ... > Nokia-Mobira (1),
Valmistaja > ... > Nokian Jalkinetehdas Oy (7),
Valmistaja > ... > Nokian Kutomo ja Värjäys Oy (2)

Esinetyyppi (koko luokittelu) (ryhmittele kohteet)
kulkuneuvot ja kuljetusvälineet osineen (1),
koneet ja laitteet (4),
pukineet ja tekstiilit (30),
säilyttimet (2), leikkikalut (2),
sisustus (1)

Materiaali (koko luokittelu) (ryhmittele kohteet)
materiaalit (37)

Valmistaja (koko luokittelu) (ryhmittele kohteet)
yritykset (34)

Hakuehdot

Hakusana: *nokia* (ryhmittele kohteet) (poista)

Kohteet ryhmiteltyinä hakusanan *nokia* mu
(näytä ilman ryhmittelyä)

Valmistaja > Nokia, kohteet 1-4/14 (ryhmittele koh

leluauto:henkilöauto (ECM 3594 47)
leluauto:lelukilpa-auto (ECM 3598 1)

Valmistuspaikka > Nokia, kohteet 1-4/32 (ryhmitt

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 that correspond to classes in the underlying ontologies. At the same time, also hit lists of museum collection items are generated. This idea expanding queries over hierarchies has been applied also, e.g., in the Open Directory Project search engine. However, in our case the 9 category views 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 indirect and more flexible. For example, in the search results of figure 1, the category 'Nokia' appears twice as a place (town). This is because the category can appear in the content of the portal in two different roles. Simply choosing e.g. the category 'Nokia (the place)' would not disambiguate the meaning sufficiently, since the same resource has the role of place of manufacture (Valmistuspaikka>...>Nokia) or place of usage (Käyttöpaikka>...>Nokia), or both, in the metadata of the museum artifacts. 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.

3.2 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[6][9]. This portal provides the user with semantic search and browsing facilities similar to MuseumFinland but to a database of some 2200 video and audio clips[7] and learning object metadata (LOM)[8] related to them.

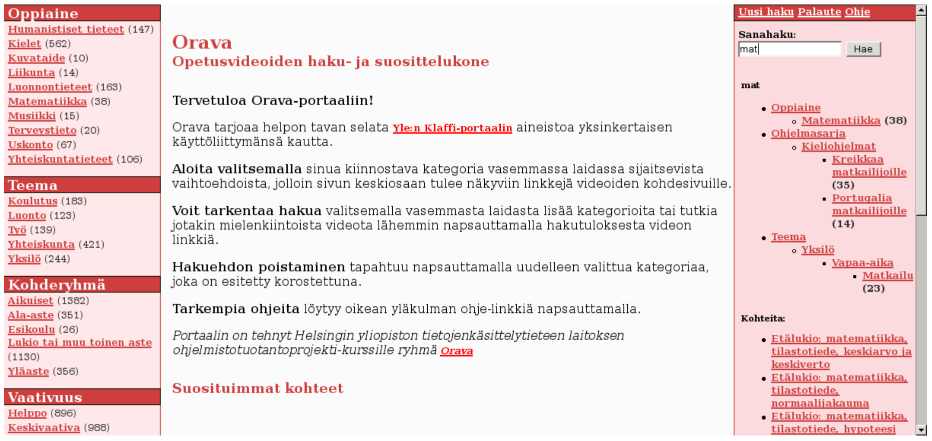
Figure 2 depicts the home page of the portal with the on-the-fly semantic autocompletion in action in the upper right corner. The user has typed in the characters 'mat', aiming perhaps at the word 'matkailu' (travel). The autocompletion function dynamically and automatically updates the category trees 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.

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

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

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



Oppiaine
Humanistiset tieteet (147)
Kielet (562)
Kuvataide (10)
Liikunta (14)
Luonnontieteet (163)
Matematiikka (38)
Musiikki (15)
Terveystieto (20)
Uskonto (67)
Yhteiskuntatieteet (106)

Teema
Koulutus (183)
Luonto (123)
Työ (139)
Yhteiskunta (421)
Yksilö (244)

Kohderyhmä
Aikuiset (1382)
Ala-aste (351)
Esikoulu (26)
Lukio tai muu toinen aste (1130)
Yläaste (356)

Vaativuus
Helppo (896)
Keskivaativa (988)

Orava
Opetusvideoiden haku- ja suosittelukone

Tervetuloa Orava-portaaliin!

Orava tarjoaa helpon tavan selata Yle:n Klaffi-portaalin aineistoa yksinkertaisen käyttöliittymänsä kautta.

Aloita valitsemalla sinua kiinnostava kategoria vasemmassa laidassa sijaitsevista vaihtoehdoista, jolloin sivun keskiosaan tulee näkyviin linkkejä videoiden kohdesivuille.

Voit tarkentaa hakua valitsemalla vasemmasta laidasta lisää kategorioita tai tutkia jotakin mielenkiintoista videota lähemmin napsauttamalla hakutuloksesta videon linkkiä.

Hakuehdon poistaminen tapahtuu napsauttamalla uudelleen valittua kategoriaa, joka on esitetty korostettuna.

Tarkempia ohjeita löytyy oikean yläkulman ohje-linkkiä napsauttamalla.

Portaalin on tehnyt Helsingin yliopiston tietojenkäsittelytieteen laitoksen ohjelmistotuotantoprojekti-kurssille ryhmä ***Orava***

Suosituimmat kohteet

mat

- Oppiaine
  - Matematiikka (38)
- Ohjelmasarja
  - Kieliohjelmat
    - Kreikkaa matkailijoille (35)
    - Portugalia matkailijoille (14)
- Teema
  - Yksilö
    - Vapaa-aika
      - Matkailu (23)

Kohteita:

- Etälukio: matematiikka, tilastotiede, keskiarvo ja keskiverto
- Etälukio: matematiikka, tilastotiede, normaalijakauma
- Etälukio: matematiikka, tilastotiede, hypoteesi

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

Below the dynamic autocompleted 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 this time on-the-fly.

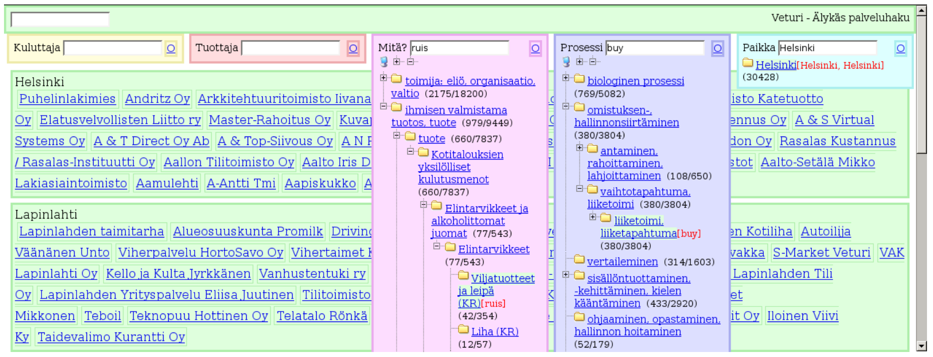
3.3 Semantic Autocompletion Facet by Facet: Case Veturi

In the semantic yellow page portal Veturi [10], created in the Intelligent Web Services (IWebS) project[9], 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 are Consumer ('Kuluttaja'), Producer ('Tuottaja'), Target ('Mitä?'), Process ('Prosessi'), and Location of the Service ('Paikka'). The views are located on the top horizontally, 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. In this box its is possible to write a sequence of (possible partial) keywords, e.g. 'buy marmelade', that are completed one after another against the views.

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

[9] http://www.seco.tkk.fi/projects/iwebs/



Mitä? ruis

- toimija: eliö, organisaatio, valtio (2175/18200)
- ihmisen valmistama tuotos, tuote (979/9449)
  - tuote (660/7837)
    - Kotitalouksien yksilölliset kulutusmenot (660/7837)
      - Elintarvikkeet ja alkoholittomat juomat (77/543)
        - Elintarvikkeet (77/543)
          - Viljatuotteet ja leipä (KR)[ruis] (42/354)
          - Liha (KR) (12/57)

Prosessi buy

- biologinen prosessi (769/5082)
- omistuksen-, hallinnonsiirtäminen (380/3804)
  - antaminen, rahoittaminen, lahjoittaminen (108/650)
  - vaihtotapahtuma, liiketoimi (380/3804)
    - liiketoimi, liiketapahtuma[buy] (380/3804)
- vertaileminen (314/1603)
- sisällöntuottaminen, -kehittäminen, kielen kääntäminen (433/2920)
- ohjaaminen, opastaminen, hallinnon hoitaminen (52/179)

Paikka Helsinki

- Helsinki[Helsinki, Helsinki] (30428)

Helsinki

Puhelinlakimies | Andritz Oy | Arkkitehtuuritoimisto Iivana
Oy | Elatusvelvollisten Liitto ry | Master-Rahoitus Oy | Kuva
Systems Oy | A & T Direct Oy Ab | A & Top-Siivous Oy | A N F
/ Rasalas-Instituutti Oy | Aallon Tilitoimisto Oy | Aalto Iris D
Lakiasiaintoimisto | Aamulehti | A-Antti Tmi | Aapiskukko | A
isto Katetuotto
ennus Oy | A & S Virtual
don Oy | Rasalas Kustannus
stot | Aalto-Setälä Mikko

Lapinlahti

Lapinlahden taimitarha | Alueosuuskunta Promilk | Driving
Väänänen Unto | Viherpalvelu HortoSavo Oy | Vihertaimet K
Lapinlahti Oy | Kello ja Kulta Jyrkkänen | Vanhustentuki ry
Oy | Lapinlahden Yrityspalvelu Eliisa Juutinen | Tilitoimisto
Mikkonen | Teboil | Teknopuu Hottinen Oy | Telatalo Rönkä
Ky | Taidevalimo Kurantti Oy
en Kotiliha | Autoilija
vakka | S-Market Veturi | VAK
Lapinlahden Tili
et
it Oy | Iloinen Viivi

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

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

In the view Target view ('Mitä?'), the user has typed in the word 'rye' ('ruis'). While the annotation ontology used does not contain different grains, the concept 'grain products and bread' ('Viljatuotteet ja Leipä (KR)') 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 into 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.

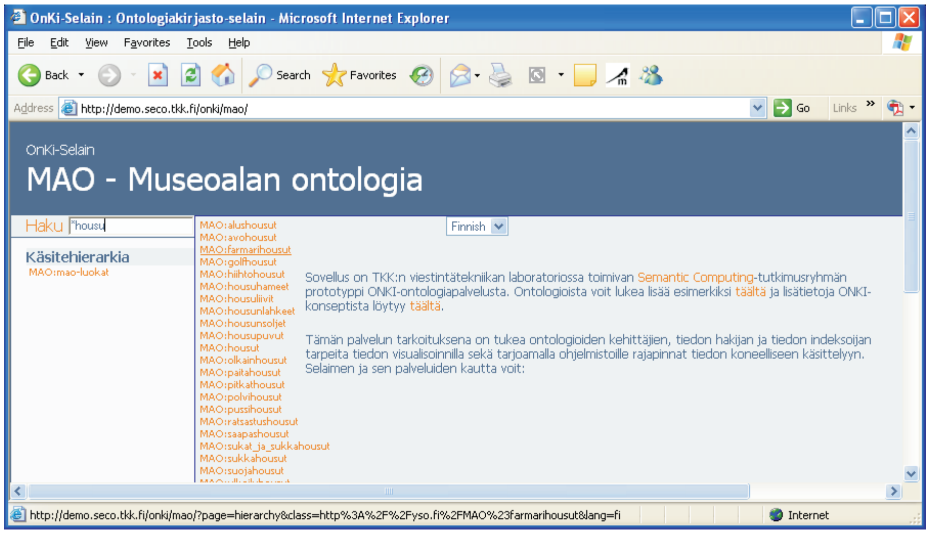
The search query entered in the view Process ('Prosessi') divulges another feature of semantic autocompletion: multilanguage support. Typing in the word 'buy' matches the appropriate business transaction, even though the word for 'buy' in Finnish would be 'ostaa'.

3.4 Semantic Indexing: Case ONKI Ontology Server

ONKI [11] is a part of the “Finnish National Ontologies on the Semantic We” (FinnONTO)[10] 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[11] 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. In figure 4, the user has typed in the regular expression '*housu' (trouser), where '*' matches any

[10] http://www.seco.tkk.fi/projects/finnonto/
[11] http://www.seco.tkk.fi/applications/onki/


OnKi-Selain : Ontologiakirjasto-selain - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://demo.seco.tkk.fi/onki/mao/ Go Links
Haku
Finnish
Internet

OnKi-Selain

MAO - Museoalan ontologia

Haku *housu

Käsitehierarkia

MAO:mao-luokat

MAO:alushousut
MAO:avohousut
MAO:farmarihousut
MAO:golfhousut
MAO:hiihtohousut
MAO:housuhameet
MAO:housuliivit
MAO:housunlahkeet
MAO:housunsoljet
MAO:housupuvut
MAO:housut
MAO:olkainhousut
MAO:paitahousut
MAO:pitkathousut
MAO:polvihousut
MAO:pussihousut
MAO:ratsastushousut
MAO:saapashousut
MAO:sukat_ja_sukkahousut
MAO:sukkahousut
MAO:suojahousut

Sovellus on TKK:n viestintätekniikan laboratoriossa toimivan Semantic Computing-tutkimusryhmän prototyppi ONKI-ontologiapalvelusta. Ontologioista voit lukea lisää esimerkiksi täältä ja lisätietoja ONKI-konseptista löytyy täältä.

Tämän palvelun tarkoituksena on tukea ontologioiden kehittäjien, tiedon hakijan ja tiedon indeksoijan tarpeita tiedon visualisoinnilla sekä tarjoamalla ohjelmistoille rajapinnat tiedon koneelliseen käsittelyyn. Selaimen ja sen palveluiden kautta voit:

http://demo.seco.tkk.fi/onki/mao/?page=hierarchy&class=http%3A%2F%2Fyso.fi%2FMAO%23farmarihousut&lang=fi

Fig. 4. Semantic autocompletion in the ontology server ONKI.

sequence of characters, and ONKI browser has completed the input into several concept categories of different types of trousers defined in the underlying cultural ontology MAO of the MuseumFinland portal. After selecting a concept by clicking on, the semantic neighborhood of the concept can be browsed further, if needed. Using ONKI, data of the selected concept such as label and the corresponding URI can read into an external application via a web service interface. ONKI can in this way be used as a service for accurate semantic indexing.

4 Implementation

The portals discussed are based on the semantic portal tool OntoViews [3], and share the same implementation of semantic autocompletion. In the implementation, the user interface component is a shallow HTML/JavaScript wrapper, whose only responsibility is to forward typed keypresses to the server. In MuseumFinland the user interface elements are static HTML, but all the newer on-the-fly implementations make use of Ajax (Asynchronous JavaScript and XML) and the XMLHttpRequest-object[12] technologies to make HTTP queries to the server in the background while viewing a page. 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.

In OntoViews, all the actual keyword matching is done on the server by Ontogator [12], the view-based search engine of OntoViews. This gives the benefit of tight integration with the main multi-facet search facilities of the engine. The search is accomplished as follows:

[12] See e.g. http://en.wikipedia.org/wiki/AJAX

Firstly, the complex ontological mapping, navigation and processing associated with semantic autocompletion is accomplished as a precalculation, alongside the view projection for the multi-facet search. For each category to be projected, a set of logic rules expressed in Prolog is consulted that dictate which labels of which ontological entities are to be associated with that category. By using such rules, the ontology manipulation involved is abstracted into chunks that are quite general, as well as easy to understand, combine and implement. For example, the Veturi system includes the following rules:

```
annotation(Category,Value):- rdf(Category,'rdfs:comment',Value).

annotation(Category,Value) :-
    sumoclass(Category), rdfs_subclassof(Category,SubCategory),
    not_projected(SubCategory), annotation(SubCategory,Value).
```

The first rule states that for all classes, also their rdfs:comment should be indexed for keyword search. The second rule then states that for each class to be projected, any annotations of subclasses *not* projected will be added. In Veturi, these two rules result in adding to the quite abstract descriptions Suggested Upper Merged Ontology (SUMO) classes used, more concrete descriptions from the mid level ontology MILO that provides example subclasses for the SUMO concepts.

At runtime, the system does only very limited processing, mostly just character manipulation of the query string, such as expanding T9-type ambiguous numerical queries [1, 2] to their possible extensions. Done this way, semantic autocompletion can easily be combined with other advances in predictive text autocompletion, because the ontological navigation happens completely separately from any string matching, similarly to the approach described in [13].

5 Discussion

This paper introduced the idea of semantic autocompletion as a natural extension to traditional autocompletion based on string matching. The idea is to use semantic structures for completing user text input into semantically relevant choices based on the underlying ontologies and content. Several forms of semantic autocompletion were proposed using equivalence relations, indirect semantic relations, semantic roles, and the idea extends seamlessly into semantic search. Semantic autocompletion uses not only string matching but also logical reasoning based on the underlying ontological structures. From the end-users viewpoint the matching occurs on the semantic level. The input text and completed choice labels may be quite different, but their relation to the query can still be understood and useful.

Our implementations and practical application of the idea to multi-facet search in semantic portals suggest that semantic autocompletion should be of practical value on the semantic web. Comprehensive user testing of the approach has not been done yet. However, the intuition obtained in implementing and expanding the view-based user interfaces to support semantic autocompletion point to good results. Combining keyword searching to the visualization capabilities of the facet hierarchies gives the user a quick path into the system, and gives at the same time an overview of what kind of information there is in the vocabulary. This guides the user in formulating the query in terms of appropriate concepts. Furthermore, showing hits inside the hierarchies solves the problems of homonymous query terms: the right meaning can be disambiguated by the view context.

Dealing with large and deep hierarchies is a major bottleneck of the multi-facet search paradigm. According to user tests [14], 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 end-user to deal the wealth of categories used in facets. The value of semantic autocompletion here comes from the integration of the benefits of the keyword-based and multi-facet search paradigms.

Acknowledgements

Samppa Saarela was responsible for creating the server-side search engine used in the OntoViews framework, and Teppo Känsälä integrated autocompletion on-the-fly with Orava. Semantic autocompletion for the ONKI ontology server was implemented by Ville Komulainen.

Our work is a part the "National Semantic Web Ontology Project in Finland (FinnONTO)[13]", funded by the National Funding Agency for Technology and Innovation (Tekes) and a consortium of some 30 public organizations and companies.

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. 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.

[13] http://www.seco.tkk.fi/projects/finnonto/

5. 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
6. Hyvönen, E., Saarela, S., Viljanen, K.: Application of ontology-based techniques to view-based semantic search and browsing. In: The semantic web: research and applications. First European Semantic Web Symposium, ESWS 2004, Heraklion, Greece, Springer–Verlag, Berlin (2004) 92–106.
7. Hyvönen, E., Mäkelä, E., Salminen, M., Valo, A., Viljanen, K., Saarela, S., Junnila, M., Kettula, S.: MuseumFinland—Finnish Museums on the Semantic Web. Journal of Web Semantics **3** (2005)
8. 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.
9. Känsälä, T., Hyvönen, E.: A semantic view-based portal utilizing Learning Object Metadata. Paper, submitted, http://www.seco.hut.fi/publications/2006/kansala-hyvonen-2006-semantic-portal-lom.pdf (2006)
10. Mäkelä, E., Viljanen, K., Lindgren, P., Laukkanen, M., Hyvönen, E.: Semantic yellow page service discovery: The veturi portal. In: Proceedings of the 4rd International Semantic Web Conference (ISWC 2005), Poster papers, Galway, Ireland. (2005)
11. Valo, A., Hyvönen, E., Komulainen, V.: A collaborative ontology development and service framework ONKI. In: Proceedings of Int. Conf. on Dublin Core and Metadata Application (DC-2005), Madrid. (2005)
12. Mäkelä, E., Hyvönen, E., Saarela, S.: Ontogator—a semantic view-based search engine service for web applications. Paper, submitted, http://www.seco.hut.fi/publications/2006/makela-hyvonen-saarela-ontogator-2006.pdf (2006)
13. 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
14. 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)