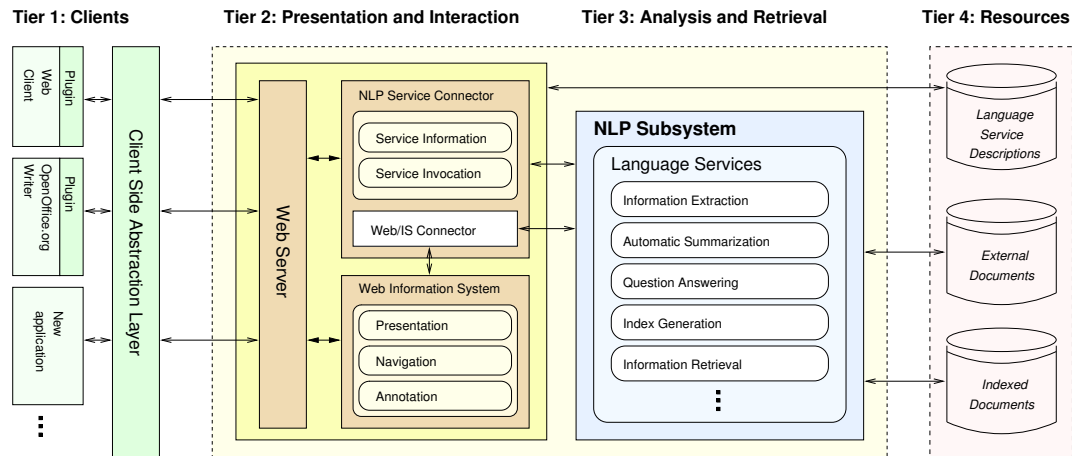


Semantic Assistants: NLP Web Services

Semantic Software Lab, Concordia University, Montréal, Canada

Semantic Assistants support users in content retrieval, analysis, and development, by offering context-sensitive natural language processing (NLP) services directly integrated into standard desktop clients, like a word processor. They are implemented through an open service-oriented architecture, using Semantic Web ontologies and Web Services.



Background

The Semantic Assistants research project aims to bring advanced natural language processing and text mining tools to the modern desktop. The current version is primarily aimed at desktop (application) developers wishing to add NLP functionality, and language engineers that want to bring their analysis services directly to end users.

Architecture: Web Services for NLP

Semantic Assistants are built on a four-tier information system architecture:

Tier 1 of the architecture consists of client applications and a Client-Side Abstraction Layer (CSAL). Currently, two example clients are distributed with the system, a simple command-line client for testing purposes and a plug-in for the OpenOffice.org *Writer* word processor. Client and server communicate through W3C *Web Services*.

Tier 2 includes a Web server and the NLP Service Connector, which is responsible for a number of tasks, including communication with the client, reading and querying the language service descriptions, running requested language services, and generating response messages.

Tier 3 is the NLP subsystem. At present, the *General Architecture for Text Engineering* (GATE) framework is supported (see <http://gate.ac.uk>).

Tier 4 is the resource tier. Here we have the language service descriptions, which are authored in the Web Ontology Language (OWL). Tier 4 further contains external

documents, which the NLP subsystem must be able to access.

System Components

The current distribution contains the following open source components:

Server. The Semantic Assistants server is the core of the architecture. It communicates with the clients through the CSAL on one hand and the NLP framework through the NLP Service Connector on the other. As a service-oriented architecture (SOA), every service is automatically available to all clients connected to the architecture, using standard *Web Services Description Language* (WSDL) interface descriptions.

Client-Side Abstraction Layer (CSAL). A primary goal of this work is to make it as easy as possible for client (plug-in) developers to integrate NLP functionality. This is achieved with an abstraction layer (CSAL), which is located on the client side and performs the actual communication with the server.

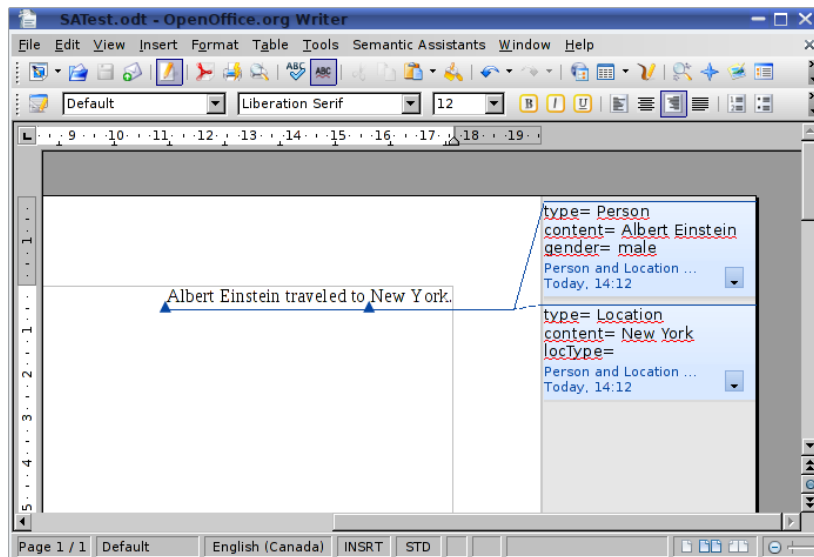
Example Resources. To match clients with suitable services, each NLP service comes with a semantic service description in the *Web Ontology Language* (OWL) format. Three example service descriptions are included in the current distribution: an information extraction (IE) service that detects persons and locations (using GATE's ANNIE pipeline), an IR service (using the Yahoo PR) and a compound service, which combines the IR and the IE service.

More Information, Software Download, and Contact

Semantic Software Lab, Concordia University, Montréal; Contact: Prof. René Witte <witte@semanticsoftware.info>
Open Source Software Download (AGPL3): <http://www.semanticsoftware.info/semantic-assistants-architecture>

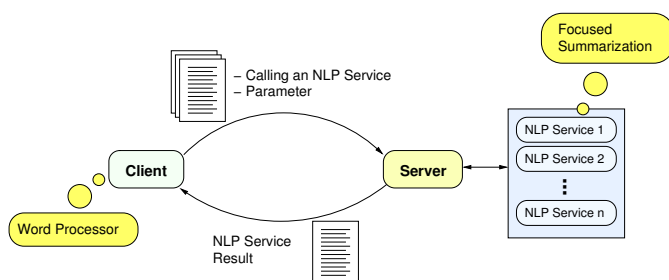
Semantic Assistants: OpenOffice.org Writer Plug-In

The OpenOffice.org Writer plug-in for the Semantic Assistants architecture provides for calling text analysis services directly from within the word processor. Results are either displayed as new documents or as annotations (using side notes) on an existing document.



Background

Today's knowledge workers are often overwhelmed by the vast amount of readily available natural language documents that are potentially relevant for a given task. Natural language processing (NLP) and text mining techniques can deliver automated analysis support, but they are still not integrated into commonly used desktop clients, such as word processors.



Our Semantic Assistants plug-in for the OpenOffice.org word processor *Writer* allows to access any kind of NLP analysis service deployed in the Semantic Assistants architecture. Example services that can help managing large amounts of textual content include information extraction, question-answering, index generation, and automatic summarization. NLP services are executed in the background (asynchronously), allowing users to continue with their work.

More Information, Software Download, and Contact

Semantic Software Lab, Concordia University, Montréal; Contact: Prof. René Witte <witte@semanticsoftware.info>
Open Source Software Download (AGPL3): <http://www.semanticsoftware.info/semantic-assistants-writer-plugin>

Design

Our primary goal for the *Writer* extension was to be able to perform text analysis on a given document. This text can, for instance, be a large document from which information should be extracted, or a problem statement consisting of a few questions, which serves as input for a question-answering (QA) Semantic Assistant. Especially for the last use case, it also allows a user to highlight parts of a document (e.g., a question) and pass only the highlighted part as input to a language service. The plug-in also offers the possibility to specify parameters that need to be passed to a selected NLP service (such as the length of a summary to be generated).

Features

Our plug-in creates a new menu entry "Semantic Assistants:" In this menu, the user can inquire about available services, which are selected based on the client (here *Writer*) and the language capabilities of the deployed NLP services. The integration of a new service does not require any changes on the client side—any new NLP service created and deployed by a language engineer is dynamically discovered through its OWL metadata maintained by the architecture and so becomes immediately available to any connected client. After the service is executed, the result is displayed in *Writer* depending on the type of the server response: either as a new document, as annotations on the existing document, or by opening an external viewer (e.g., a Web browser for HTML documents).