

# SconeEdit: a Text-guided Domain Knowledge Editor

Alicia Tribble

Language Technologies  
Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213  
atribble@cs.cmu.edu

Benjamin Lambert

Language Technologies  
Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213  
benlambert@cmu.edu

Scott E. Fahlman

Language Technologies  
Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213  
sef@cs.cmu.edu

## Abstract

We will demonstrate SconeEdit, a new tool for exploring and editing knowledge bases (KBs) that leverages interaction with domain texts. The tool provides an annotated view of user-selected text, allowing a user to see which concepts from the text are in the KB and to edit the KB directly from this Text View. Alongside the Text View, SconeEdit provides a navigable KB View of the knowledge base, centered on concepts that appear in the text. This unified tool gives the user a text-driven way to explore a KB and add new knowledge.

## 1 Introduction

We will demonstrate SconeEdit, a new tool for exploring and editing knowledge bases that integrates domain text. SconeEdit expands on the function of traditional ontology editors by showing the user an interactive text window (Text View) where the user can view and edit concepts from the knowledge base as highlighted terms in their original context. The Text View augments a traditional KB View, allowing the user to leverage existing knowledge as well as domain-focused text examples to perform a variety of knowledge-based tasks.

Consider the task of assessing the quality of a knowledge base as a resource for a new AI or natural language system. In SconeEdit, a user can view

the knowledge base alongside a text document from the target domain. SconeEdit searches for instances of KB concepts in the text and highlights them in the Text View. Already the user can see a concise visual sample of the coverage of the KB for this domain.

Now the user can work with the KB View and Text View together to navigate the ontology. Double-clicking on a highlighted concept like “keyboard” opens a detailed view of that concept in the KB View. Inside the KB View, the user can click on the superclass of the *keyboard* concept to see the concept *computer input device* and all of its children. Next, SconeEdit selectively highlights all instances of *computer input device* in the text. The system uses type inference from the KB to highlight “mouse”, “touchpad”, and “wireless keyboard.” If “scanner” appears in the text but isn’t included in the knowledge base, the user can spot the omission quickly.

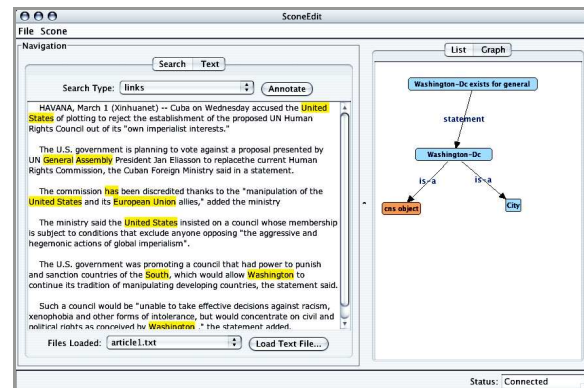


Figure 1. The SconeEdit Interface

In this way, domain text is used as a measuring tool for coverage of domain knowledge. Our demonstration allows the user to try SconeEdit and to explore the interaction of text and knowledge.

## 2 The Knowledge Base

SconeEdit is a software client to the Scone Knowledge Base System, or simply “Scone” (Fahlman, 2005). Scone is an efficient, open-source knowledge base (KB) system being developed in the Language Technologies Institute of Carnegie Mellon University. Scone is intended to be a practical KB system that can be used as a component in a wide range of AI and natural language software applications. One of the goals in developing Scone is to make it easy to use, especially when adding new knowledge.

The SconeEdit interface makes Scone more usable in several ways: the Text View display gives the user a convenient and intuitive starting point for exploring the knowledge base. SconeEdit also provides an easy way of adding knowledge to the KB without learning the formal input language for Scone. This demonstration focuses on the effectiveness of SconeEdit and Scone together, but the design principles of SconeEdit are applicable to knowledge bases written in other formalisms.

Figure 1 shows the SconeEdit window with a document and KB loaded. The left side of the interface contains the Text View, and the KB View is on the right. Each of these views is described in detail below.

## 3 Architecture

### 3.1 Text View

In a traditional ontology browser, the user starts looking for concepts of interest by typing words and phrases into a search field. This is the model for several existing tools, including the VisDic viewer for WordNet (Horák and Smrž, 2004), the INOH ontology viewer (INOH, 2004), and the Gene Ontology viewer presented by Koike and Takagi (2004), among others.

SconeEdit improves on this browsing paradigm by giving a user who is unfamiliar with the knowledge base an easy way to start exploring. Rather than generating a series of guesses at what may be

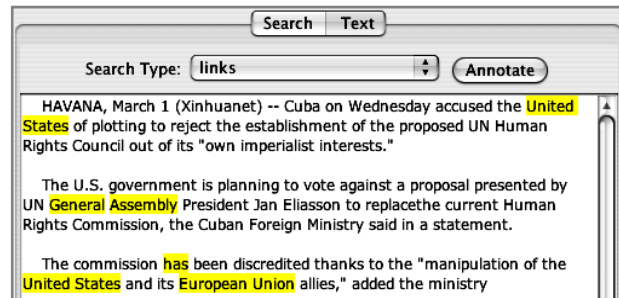


Figure 2. Excerpt from Text View, with Search and Text Tabs

covered by the KB, the user can load natural language text into SconeEdit from a file or the system clipboard. We take an article from Xinhuanet News Service (Xinhuanet, 2006) as an example. Figure 2 shows an excerpt of this text after it has been loaded.

When the text file is loaded, it appears in the Text Tab of the Text View pane. SconeEdit highlights all strings that it can identify as concepts from the knowledge base. In this example, “Washington” is correctly identified as the city, not the state. In many cases the concept may be ambiguous from the string alone. SconeEdit currently uses dynamic programming to highlight the longest-matching concept names it can find (see Section 5). More sophisticated disambiguation is a priority for our future work.

The result of highlighting is a concise visual representation of what is “known” about that text. The Text View helps a user find relevant knowledge quickly, even in a large general-domain KB. Clicking on any highlighted term in the Text View brings up a hierarchical representation of that concept in the KB View.

### 3.2 KB View

The KB View contains two tabs: a Graph Tab and a List Tab. The Graph Tab displays an excerpt from the knowledge base as a network of linked concepts with one *focus concept* in the center. When the user clicks on a highlighted concept in the Text View, a graph focused on that concept appears in the Graph Tab. Continuing with our Xinhuanet example, Figure 3 shows the Graph Tab after a user has clicked on “Washington” in the text. The Graph View now displays concepts that are closely related to *Washington-Dc* in the knowledge base.

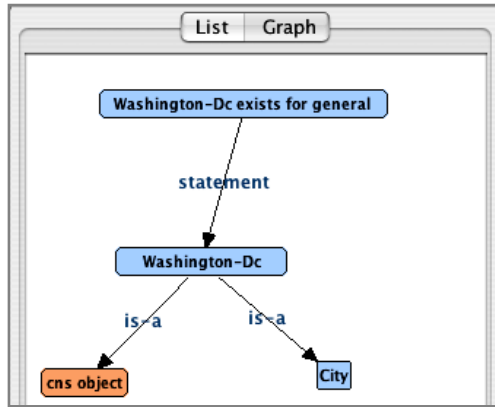


Figure 3. KB View, Graph Tab of *Washington-Dc*

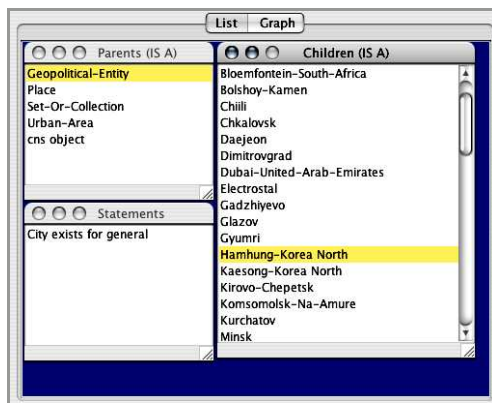


Figure 4. KB View, List Tab of *City*

Clicking on any of these related concepts in the Graph Tab moves the focus of the graph to that concept.

The List Tab shows an alternative view of the same *focus concept*. It displays KB information as a set of property lists. As in the Graph Tab, the user can double-click on any concept in the List Tab to bring that concept into focus. When the focus concept is densely connected to other concepts in the KB, the List Tab can be easier to interpret than the Graph Tab. In general, research has shown that preference for the list style or graph style is personal and varies from user to user (Tribble and Rosé, 2006). Figure 4 shows the List Tab, focused on the concept *City*.

#### 4 Adding Knowledge

Browsing the knowledge base in this way gives the user a detailed, domain-targeted view of its contents. A natural extension of this paradigm is to allow the user to edit the KB while browsing. For example, a user may encounter a concept in the

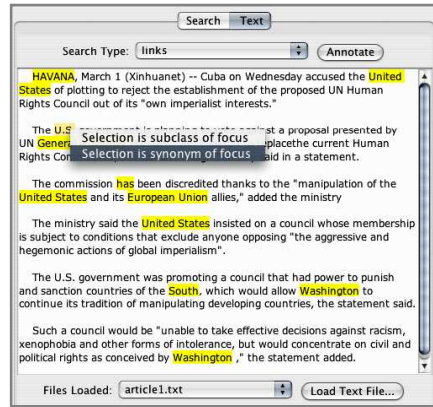


Figure 5. Adding a concept synonym

text that is not present in the knowledge base. SconeEdit allows the user to simply click on a word in the text to create a new concept in the KB (see Figure 5). To specify where the new concept belongs, the user navigates to the appropriate location in the KB View (List Tab or Graph Tab).

The user can also modify an existing KB concept by adding English synonyms. For example, the word “United States” may be highlighted in a text example, while “U.S.” is not. To add a synonym for the “United States” concept, the user navigates to this concept in the KB View, and then clicks on the text “U.S.”. A menu offers the choice of adding a synonym to the existing focus concept. Figure 5 illustrates this process.

#### 5 Identifying KB Concepts in Text

Elements in a Scone knowledge base represent specific concepts, rather than words or word senses. Each concept is linked with a list of English names (words or phrases). This association between Scone elements and English names is many-to-many.

To map a sentence to the set of concepts that appear there, a dynamic-programming alignment is performed using the English names in the KB as a dictionary. SconeEdit searches for an alignment that covers as much of the input text as possible. The result of aligning an input string with concepts is a set of triples, each consisting of a concept, an offset, and a length. These triples are used directly by the Text Tab to highlight substrings and associate them with KB concepts.

Consider the sentence “Washington, D.C. is a city.” Table 1 shows some example Scone concepts and their English names. Given a knowledge

Concept Name	English Names
<i>Washington-State</i>	“Washington”, “Washington State”,
<i>Washington-Dc</i>	“Washington”, “Washington, D.C.”
<i>City</i>	“city”

Table 1. Example concepts and their English Name lists

base with these concepts, SconeEdit returns the alignment: (concept: *Washington-DC*, offset: 1, length: 16) (concept: *City*, offset: 23, length: 4).

## 6 Planned Features

A single node in the KB could have hundreds or thousands of outgoing links. For readability, the browser must select a subset of these links to display to the user. We plan to leverage Scone’s reasoning ability, along with SconeEdit’s document-driven design, to select which nodes are likely to be relevant to the user in the context of the loaded document(s). For example, a user who views subclasses of *disease* in a medical ontology may be presented with thousands of disease types. If the current document loaded into SconeEdit is a document about *food*, Scone may be able to prune the subclasses it lists to only food-borne illnesses.

Another feature we hope to add is better integration with an entire corpus. The current system allows the user to work with individual documents. This could be extended to allow a user to navigate to a particular concept in the knowledge base and retrieve all documents in a corpus containing that concept (in its various forms). These documents could then be used to generate more KB concepts of interest.

## 7 Related Work

To the best of our knowledge, existing ontology and KB editors and viewers do not specifically focus on editing and viewing an ontology or KB in the context of natural language text. Other ontology editors such as Protégé (Gennari, 2002) and OntoEdit (Sure, 2002) offer many features for generating complex ontologies, but do not provide the rich interaction with domain text that is the focus of SconeEdit. The CNet Big Picture (CNet News

Online, 2000) is one example of a system that does link ontology knowledge to text, but the concepts in the ontology are limited to a small fixed set.

## Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010. The authors would like to thank Vasco Pedro, Eric Nyberg, and Tim Isganitis for their contributions to SconeEdit.

## References

- CNet News Online. 2000. *The Big Picture*, [http://news.com.com/The+Big+Picture/2030-12\\_3-5843390.html](http://news.com.com/The+Big+Picture/2030-12_3-5843390.html).
- S.E. Fahlman. 2006. *Scone User's Manual*, <http://www.cs.cmu.edu/~sef/scone/>.
- J. Gennari, M. A. Musen, R. W. Ferguson, W. E. Grosso, M. Crubezy, H. Eriksson, N. F. Noy, S. W. Tu. 2002. The Evolution of Protégé: An Environment for Knowledge-Based Systems Development. *International Journal of Human-Computer Interaction*, 58(1), pp. 89—123.
- Aleš Horák and Pavel Smrž. 2004. VisDic -- WordNet Browsing and Editing Tool. *Proceedings of GWC 2004*, pp. 136—141.
- INOH, 2004. INOH Ontology Viewer Website. <http://www.inoh.org:8083/ontology-viewer/>.
- Asako Koike and Toshishisa Takagi, 2004. Gene/protein/family name recognition in biomedical literature. In *Proceedings of BioLINK 2004: Linking Biological Literature, Ontologies, and Databases*, pp. 9-16.
- Alicia Tribble and Carolyn Rosé. 2006. Usable Browsers for Ontological Knowledge Acquisition. To appear in *Proceedings of CHI-2006*. Montréal, Canada. April 22-27, 2006.
- Xinhuanet. 2006. US accused of blocking approval of new UN human rights body. [http://news.xinhuanet.com/english/2006-03/02/content\\_4247159.htm](http://news.xinhuanet.com/english/2006-03/02/content_4247159.htm).
- Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer and D. Wenke. OntoEdit: Collaborative Ontology Engineering for the Semantic Web. In *Proceedings of the first International Semantic Web Conference 2002 (ISWC 2002)*.