Statistical Models for Organizing Semantic Options in Knowledge Editing Interfaces

Jill Nickerson

Division of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138
nickerso@eecs.harvard.edu

Abstract

This paper describes the design and empirical evaluation of statistical models that use domain and lexical knowledge to organize new semantic options in interfaces for editing knowledge bases. We employ the models in a system that allows a domain expert to perform language-neutral knowledge editing by interacting with natural language text generated by a natural language generation system. This editing produces a knowledge base used for natural language generation in multiple languages. To create statistical models, we use natural language techniques to automatically process a large corpus and then collect statistics based on the frequency of objects and verb-object pairs. The models are used to produce methods for organizing new semantic options based on 1) overall object frequency and 2) verb-object pair frequency. We compare these two innovative methods to the more traditional method of organizing options alphabetically. We conduct a set of experiments to evaluate these three methods for organizing new semantic options and analyze subjects' interactions with the different methods in terms of speed and accuracy.

1 Introduction

Representing the content of a document in a formal knowledge base (KB) enables documents expressing that content to be generated in multiple languages. Traditionally, building a KB required an expert in the knowledge to be contained in the KB *and* an expert in the knowledge representation (KR) language being used to build it. The WYSIWYM (What You Say Is What You Meant) knowledge-editing method (Power *et al.*, 1998), however, allows a domain expert to build and extend a KB representation without previous exposure to computational linguistics or KR languages. In WYSIWYM, the expert interacts with natural language that has been generated by a natural language generation system, rather than a KR language, to build a KB that is used for multiple language generation of documents. As the domain expert builds a KB representation, the system must organize the semantic options that the expert can choose to extend the KB.[1] The organization method the system uses should allow the expert to quickly and accurately make selections. Throughout this paper, we use examples from our test domain of vehicle service manuals. Figure 1 shows a partially specified manual in this domain.

Previous work involving WYSIWYM (Power *et al.*, 1998; Bouayad-Agha *et al.*, 2002) organizes new semantic options alphabetically. Unlike previous work, we use methods that take advantage of domain and lexical knowledge. To acquire domain and lexical knowledge, we automatically extract relevant syntactic arguments from a background corpus. Then, we use the frequency of these arguments in the background corpus to organize alternative semantic options that are presented to the domain expert as he builds a KB representation.

After briefly explaining WYSIWYM, we describe the statistical models that use domain and lexical knowledge to organize new semantic options. We then discuss the empirical evaluation we are currently conducting to determine when alternative methods for organizing new semantic options are preferred.

2 WYSIWYM: A Method for Editing Knowledge Bases

WYSIWYM (Power *et al.*, 1998) allows a domain expert to build a KB directly through interacting with natural language 'feedback text.' The feedback text, which is generated by a natural language generation system using a partial KB, communicates the current state of the KB and ways to extend it. Because the expert is directly editing a

[1] In this paper, we focus on new semantic options. Other work, (Nickerson, 2002; Van Deemter and Power, 1998) has discussed the organization of objects that have already been introduced into the KB.

[A] To replace ***some part*** for the 2001 Buick Le Sabre:

First perform the removal method:

[B] 1. Remove ***some part*** with a flat bladed tool.
2. Do ***some action***.

Then, perform the installation method:

[C] 1. Install the drain hoses with ***some tool.***
2. Clean ***some object*** with ***some tool***.
3. Fasten the transmitter battery into ***some position***.

Figure 1. Partially Specified Vehicle Service Manual.

KB, the system does not need to interpret text. In addition, it is possible to use the same KB to generate texts in multiple languages, alleviating the need to translate a document from one language into others.

WYSIWYM has been deployed in several systems,[2] including DRAFTER for generating software documentation (Power *et al.*, 1998) and PILLS for generating patient leaflets (Bouayad-Agha *et al.*, 2002). We used WYSIWYM and domain knowledge gleaned from a corpus study of vehicle manuals (GM, 2002) to develop AUTO, a system technical writers can use to *auto*matically generate vehicle manuals.

In the feedback text for a partially specified manual generated by AUTO in Figure 1, bold, italicized words indicate places where knowledge can be expanded. When the expert clicks on one of these phrases, he is presented with a menu of semantic options for extending the KB. For example, clicking the phrase 'some object' [C2] results in the menu of objects shown in Figure 3 [1]. After the expert selects an object, the feedback text is updated to reflect the revised KB. When the expert has completed adding knowledge, the system can construct the 'output text.' This text is intended to be the actual manual text and, therefore, does not include markings to indicate where knowledge may be added.

3 Statistical Models for Organizing Semantic Options

We create two statistical models, each of which supports different methods for organizing new semantic options in menus like those in Figure 3. The first model takes object frequency into account, whereas the second one uses verb-object frequency. To create these models, we process a background corpus of approximately 600 vehicle repair procedures[3] (GM, 2002) using natural language techniques and then compute statistics based on the frequency of objects and verb-object cooccurrence patterns.

Figure 2 shows how the models are created. Each word contained in the instruction steps of the vehicle repair procedures is first annotated with a part of speech tag (Ratnaparkhi, 1996). Next, the instruction steps are parsed (Collins, 1997).[4] We then extract matrix verb and direct object pair(s) for each instruction step using syntactic patterns written in TGrep2[5] (Rohde, 2001). To create Model 1, which is based on object frequency alone, we keep track of the extracted object and the number of times it has been observed. For Model 2, we track the number of times that verbs and objects cooccur.[6]

Model 1 is used to create AUTO-O. This version of AUTO organizes new semantic options by their overall frequency in the background corpus. Objects that were not observed in the background corpus are organized alphabetically. Model 2 is used to create AUTO-VO. Once the domain expert chooses a verb, AUTO-VO organizes valid objects of the verb based on the number of times they cooccurred with the verb in the background corpus. In AUTO-VO, verb-object pairs are used as selection patterns to give preference to certain new options given the context, namely the preceding verb, in which they are to appear. Figure 3 shows partial menus for specifying an object that the subject is presented with if he clicks on the phrase 'some object' in instruction step [C2] in Figure 1. Figure 3 [1] shows the menu that would be presented to the subject if he were interacting with AUTO, and [2] is the menu he would be given if he were interacting with AUTO-VO. AUTO-VO places those objects that most frequently occur with the verb 'clean' towards the top of the menu. AUTO simply alphabetizes all possible objects contained in the system's ontology.

[2] Our KB is similar in structure to those in other systems using WYSIWYM.

[3] The 600 vehicle repair procedures contain approximately 8,500 instruction steps. The steps in the installation and removal procedures in Figure 4 are examples of instruction steps.

[4] Both the tagger and parser were trained on the *Wall Street Journal* (WSJ). We, therefore, introduce some errors that could be avoided if we had available manually tagged and parsed vehicle service manuals that could be used to retrain the tagger and parser. Also, since imperative sentences are, for the most part, nonexistent in the WSJ, we insert the subject 'you' and the modal verb 'should' in all imperative sentences to avoid parsing errors.

[5] TGrep is a software tool which, given a set of parse trees and a pattern using node names contained in the parse trees and relationships between these nodes, extracts those trees that match the pattern.

[6] The cooccurrence statistics collected for Model 2 are similar to those used by (Dagan and Itai, 1990). In their work, verb-object frequencies observed in a background corpus are used to organize the possible referents of an anaphor. We keep one file for each initial letter of the observed verbs.

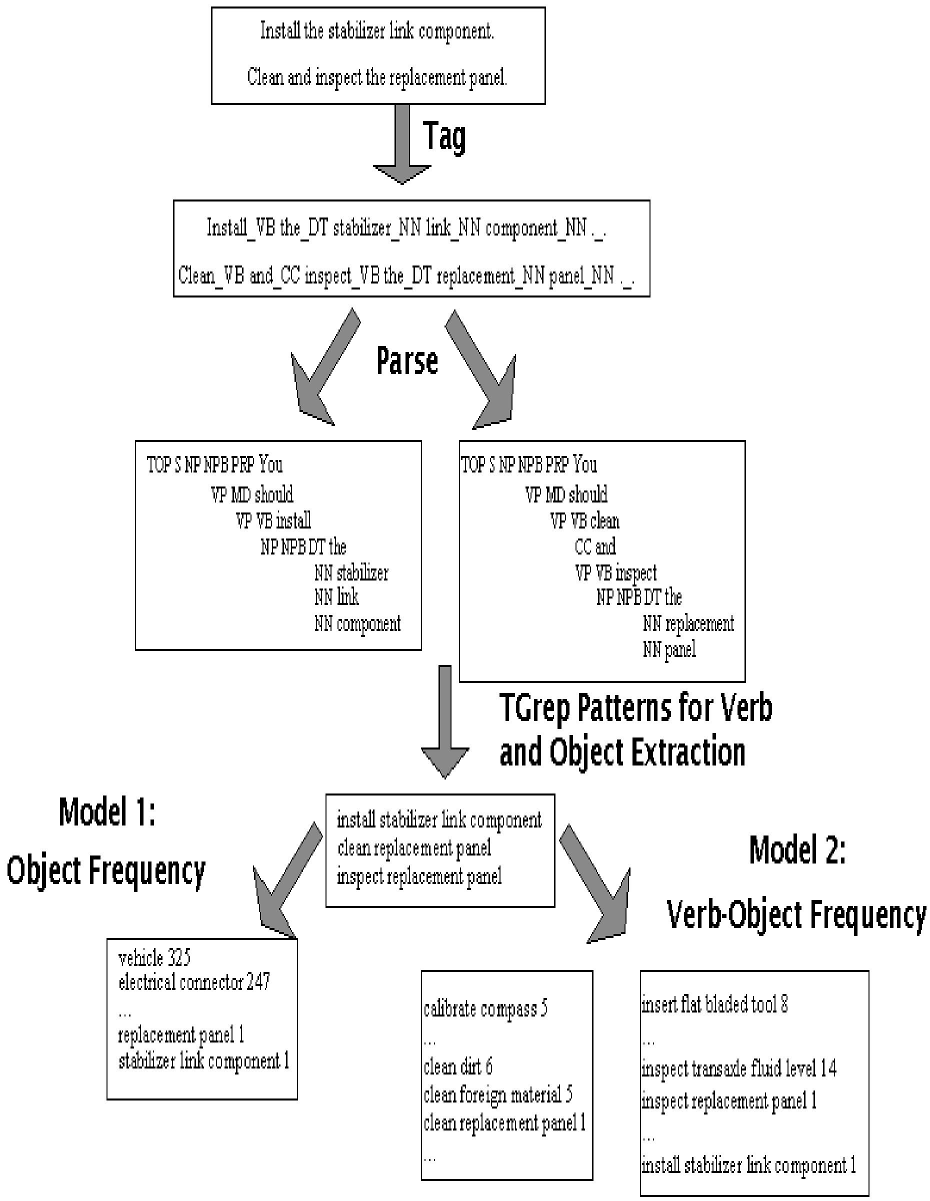

Figure 2. Creating Statistical Models Using Domain and Lexical Knowledge.

4 Experimental Design

The main goal in evaluating the different versions of AUTO is to determine if subjects find the semantic option organization methods that our statistical models produce useful. To test the contribution of the statistical models, we compare six versions of our system for authoring vehicle service manuals: AUTO, AUTO-O, AUTO-VO, and a version of each employing a two-level hierarchy for organizing options.

AUTO, like many existing knowledge editing systems (e.g., (Power *et al.*, 1998; Bouayad-Agha *et al.*, 2002)), organizes new semantic options alphabetically. Both AUTO-O and AUTO-VO use innovative organization methods that employ the statistical models described in Section 3. For each version of the system, we also create

CHOOSE AN OBJECT
brake pressure modulator
cylinder head
dirt
foreign material
glass
…
replacement panel

CHOOSE AN OBJECT
dirt
foreign material
brake pressure modulator
cylinder head
glass
…
replacement panel

[1] **[2]**

Figure 3. Menu Organization for Specifying ‘some object’ in Step [C2] of Figure 1: AUTO [1] vs. AUTO-VO [2].

a version that uses a two-level hierarchy of objects as opposed to a flat presentation style. The hierarchy is based on the objects that appear in the background corpus of manuals for repairing different parts of a vehicle such as ‘brakes’ and ‘engine.’ In versions of the system using hierarchies, the subject must first, for example, specify that the part he wishes to select is related to the brakes; and then, if he were using AUTO-VO, he would see a list of ‘brake parts’ ordered according to their cooccurrence with the selected verb. The following three factors are controlled in the experiments:

1. **System version**: We use a within-subjects experimental design to avoid individual subject differences. All participants interact with each of the six versions of AUTO: AUTO, AUTO-O, AUTO-VO, and a version of each using a two-level hierarchy for presenting objects.

2. **Procedures**: To alleviate effects of repair procedures that may be too easy or difficult, each subject is asked to author twelve different repair procedures.[7] An example problem is shown in Figure 4. The subject is told that he may not be able to reproduce the procedure word for word. His goal is to author a procedure in which a vehicle repairperson would be able to successfully follow the procedure.

3. **Order**: For the experiments, subjects are randomly divided into six groups of two subjects each. To compensate for carryover effects, i.e. learning, due to presentation order, we counterbalance by ordering the versions of the system that the subjects interact with using a balanced Latin Square (Martin, 1996).[8]

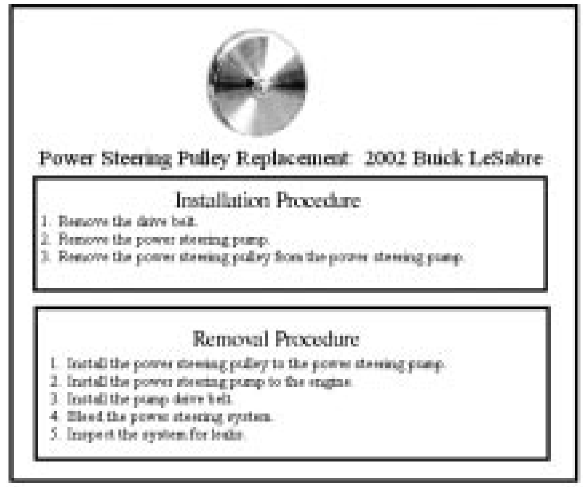

Figure 4. Example Vehicle Repair Procedure.

Twelve subjects interact with each of the six versions of AUTO to author two vehicle repair procedures per system version. The subjects are technical writers who have no previous experience using AUTO.[9] They first authors six repair procedures using the six versions in the order that is randomly assigned to him. Then, he authors six more repair procedures using this same order.

We record two dependent variables: speed and accuracy. For each procedure/subject/version triplet, we record the total time to complete the procedure and also the amount of time that elapses between when the subject is presented with a list of new semantic options and when the subject makes a selection. In addition, a panel of experts, in our case three mechanics, rate the quality of each completed repair procedure on a scale of 1-10.[10]

5 Discussion

We apply an analysis of variance (ANOVA) (Cohen, 1995) to examine the effects of the organization methods suggested by the statistical models that take advantage of domain and lexical knowledge. Overall, we expect subjects to author vehicle repair procedures more quickly and accurately when interacting with versions of the AUTO that use statistical models based on verb and object cooccurrence frequency. In certain situations, though, we hypothesize that different versions of the system will be preferred.

For selecting new objects that are quite rare in the background corpus, we predict subjects will prefer AUTO. Subjects will be able to more efficiently find the

---

[7] None of these procedures are part of the background corpus. Also, their structure is flat; they do not contain nested procedures.

[8] A balanced Latin Square is one in which, in addition to each system version appearing with equal frequency in each position, each system version also appears before and after each other system version an equal number of times.

[9] They will, however, be given adequate training prior to using AUTO in the experimental setting.

[10] 10 being of the highest quality.

desired object when the valid objects are organized alphabetically as opposed to having to scroll through other options that occur frequently in the background corpus (as in AUTO-O) or in this particular context (as in AUTO-VO). In this situation, the subject is not likely to understand the menu organization when interacting with AUTO-O and AUTO-VO. Instead, he is likely to waste time scrolling through common objects. With the alphabetical organization, however, the subject will be able to quickly scan through the list to find the desired option.

Assuming the coverage of the background corpus is comprehensive, subjects are likely to prefer AUTO-VO when they wish to select options that are common in the given context. For example, if the subject wished to select 'seal retaining clamp' following the specification of the verb 'crimp,' the subject is likely to prefer AUTO-VO since there are three objects in the background corpus that are frequently crimped, and 'seal retaining clamp' is one of them. Since this object is quite rare overall when compared to all of the objects in the ontology and since it falls within the second half of the alphabet, it is unlikely that the subject will quickly be able to arrive at the desired selection when either AUTO or AUTO-O is used.

In some cases, however, new semantic options may be quite frequent in the background corpus overall, but the subject may wish to place the new object in a context which was not observed in the background corpus. In situations like this one, the subject should find the desired object most quickly when using AUTO-O.

6 Ongoing Work

We are currently performing the evaluation explained in Section 4. We plan to extend the statistical models to consult both a background corpus and a library of already generated procedures when organizing new semantic options. If both sources of information are consulted, even though the background corpus may not be a reasonable domain representation, as the domain expert continues to author more manuals, the statistical models will become tailored to the expert's use of objects and will then organize new options to more accurately reflect his usage of the ontology.

In addition to using statistical models that use domain and lexical knowledge to organize new semantic options, we plan to combine these models with a model that uses discourse structural knowledge to organize coreference candidates (Nickerson, 2002), semantic options that have already been introduced into the KB. This type of model for organizing coreference candidates would take advantage of knowledge at the sentence level as well as at the discourse level. We also plan to extend the statistical models to collect frequency information about other syntactic relations, such as object-object pairs. Equipped with this knowledge, the models would be able to predict which objects are most likely to follow a given object. We may also investigate verb-object-object trigrams, which given the context of a verb-object pair, predict the objects most likely to follow the pair. Also, in addition to demonstrating the usability of statistical models that use domain and lexical knowledge to organize new options represented by linguistic expressions, we plan to demonstrate the usefulness of such models in organizing options in any kind of interface to a knowledge editing system. For example, statistical models can be used to organize new semantic options in graphical browsers (Skuce and Lethbridge, 1995) given a background corpus of network diagrams from an appropriate domain.

Finally, we have been investigating the use of a tool that automatically extracts verb subcategorization information (Kinyon and Prolo, 2002). After extending the tool to associate a lexical item with each syntactic argument type, this type of tool would provide the information necessary for creating statistical models. In addition, it would guide us in constructing generation rules that had previously been constructed manually through a corpus analysis. This tool, however, requires that the parse trees accepted as input are annotated with 'function tags' indicating semantic roles which are not trivial to assign automatically (Blaheta and Charniak, 2000).

7 Conclusions

We have described the creation and empirical evaluation of statistical models that take advantage of domain and lexical knowledge to organize new semantic entities in interfaces to knowledge editing systems. The models extract verb-object pairs from a background corpus and then collect frequency statistics of these arguments. The frequency patterns observed in the corpus are used to organize new semantic options presented to users of knowledge editing systems. We are currently comparing methods for organizing new semantic entities based on object frequency and verb-object frequency to methods used in previous work that organize new semantic entities alphabetically. We will determine which method, overall, leads subjects to produce the highest quality results in a shortest period of time and characterize situations in which certain methods are preferred.

Acknowledgments

This work is supported in part by NSF grant IIS-9811129. We thank the people at the Univ. of Brighton for providing WYSIWYM. Thanks also to Barbara Grosz, Richard Power, and Donia Scott for helpful suggestions. Thanks to Steven Kosslyn for help with the experimental design.

References

Blaheta, D. and Charniak, E. 2000. Assigning function tags to parsed text. *Proceedings of NAACL*, 234-240.

Bouayad-Agha, N., Power, R., Scott, D., and Belz, A. 2002. PILLS: multilingual generation of medical information documents with overlapping content. Proceedings of LREC.

Cohen, P. 1995. *Empirical Methods for Artificial Intelligence*. MIT Press.

Collins, M. 1997. Three Generative, lexicalised models for statistical parsing. *Proceedings of ACL*.

Dagan, I. And Itai, A. 1990. Automatic processing of large corpora for the resolution of anaphora references. *Proceedings of the 13th COLING*.

GM. 1980-2002. GM Service Operations Web Pages at URL http://service.gm.com.

Kinyon, A. and Prolo, C. 2002. Identifying verb arguments and their syntactic function in the Penn treebank. *Proceedings of LREC 2002*.

Martin, D. 1996. *Doing Psychology Experiments*. 4th ed. Pacific Grove, CA: Brooks/Cole.

Nickerson, J. 2002. Controlling linguistic coreference in graphical interfaces. *Proceedings of ACL-02 Student Session*.

Power, R., Scott, D., and Evans, R. 1998. What you see is what you meant: direct knowledge editing with natural language feedback. *Proceedings of ECAI-98*, 675-681.

Ratnaparkhi, A. 1996. A maximum entropy part-of-speech tagger. *Proceedings of EMNLP*.

Rohde, D. TGrep2.

Skuce, D. and Lethbridge, T. 1995. CODE4: a unified system for managing conceptual knowledge. *Proceedings of International Journal of Human-Computer Studies*, 42: 413-451.

Van Deemter, K. and Power, R. 1998. Coreference in knowledge editing. *Proceedings of ACL-COLING-98 Workshop on the Computational Treatment of Nominals*, 56-60.