Resolving Noun Compounds with Multi-Use Domain Knowledge [*]

Alicia Tribble and Scott E. Fahlman
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, Pennsylvania 15213
{atribble, sef}@cs.cmu.edu

Abstract

In this paper we describe a system for semantic interpretation of noun compounds that relies on world and domain knowledge from a knowledge base. This architecture combines domain-independent compounding rules with a task-independent knowledge representation, allowing both components to be flexibly reused. We present examples from *Scientific American* text, on which the system was developed, and then describe an exercise that tests the portability of the architecture to a new domain: email text on the topic of conference planning.

Introduction

In this paper we describe a system for interpreting noun compounds. It relies on general and domain-specific knowledge from a knowledge base.

Noun compounds are a type of multi-word expression (MWE) made up of a series of two or more nouns. "Computer science", "customer service", and "speaker laptop testing area" are all examples. They occur frequently in natural language: the authors of (Sag *et al.* 2002) claim that MWEs make up 41% of WordNet entries, and name noun compounds as one of the major sub-classes.

Although some of these compounds have become fixed expressions with non-compositional meanings ("attorney general", "Guinea pig"), many can be interpreted by determining the implied relationship between the nouns. Our experiments deal with this second class of compounds. In our system, a knowledge base stores information about concepts and the typical semantic relations that can occur between them. To analyze a noun compound, we search for an interpretation that is consistent both with our stored world knowledge and with the grammatical patterns typically used to create compounds.

Key components of the system include a set of domain-independent interpretation rules and the knowledge base itself. A dictionary lookup maps strings into their knowledge base concepts, a mapping that is many-to-many. The interpretation rules build complex meaning structures by linking the concepts together. To develop the interpretation rules we worked with a collection of concepts from a *Scientific American* article on oil refinement. These concepts were introduced as training material for noun compound resolution in (McDonald 1982).

An appealing feature of this architecture is that the rules and knowledge are independent resources that may be reused in other contexts. After developing on the Scientific American text, we experimented with the system in a totally new domain: text from personal emails on the topic of conference planning. The system is able to interpret a diverse range of compounds in the new domain using the same domain-independent rules. In addition, the knowledge base which we use for email interpretation has been successfully applied to other NLP problems for the same domain: namely, text classification.

An Example of Compound Analysis

We describe noun compounds in terms of concept-specific semantic roles (or "slots") which can be generalized using ontological information in the knowledge base. An example of the system output for "car maintenance" is shown in Figure 1.

Figure 1: System output for the compound "car maintenance". Elements from the knowledge base (concepts and roles) are shown in curly braces.

```
 RESULT FOUND:
{automobile-maintenance-0}
IS A TYPE OF {maintenance} WHERE
THE ROLE OF
{maintainee (role) of
automobile-maintenance-0}
IS A {automobile}
```

In this example, we find that *maintenance* is a concept with a *maintainee* role. The term "car" has triggered the concept *automobile*, which is used as a filler for the role.

Using inheritance links in the knowledge base, we can also determine that maintainee is a specialization of the the-

[*]This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010.

matic role *patient*. This kind of information allows us to assign several labels to the noun compound relations, on varying levels of granularity. For the purposes of these experiments we always choose the most specific label possible. An unordered list of the ancestors of *maintainee of automobile-maintenance-0* is given in Figure 2.

Figure 2: Ancestors of *maintainee of automobile-maintenance-0*

```
{object affected (role)}
{manufactured product}
{automobile}
{wheeled-vehicle}
{vehicle}            {maintainee (role)}
{machine}            {participant (role)}
{physical object}    {tool}
{inanimate}          {thing}
```

Related Work

Several systems including (Finin 1980), (Leonard 1984), and (Vanderwende 1994) have used a collection of semantic concept representations with domain-independent combination rules. We share with these systems a focus on knowledge representation as a core technology for the task of compound resolution. The work of (Fan, Barker, & Porter 2003) demonstrates that applying ontological knowledge makes sense for this task, particularly in the context of knowledge acquisition. In fact, an important effect of using representations from a knowledge base to interpret noun compounds is that the results can be stored back in the ontology and applied to future tasks.

Our work expands most directly on the algorithms described in (McDonald 1982), We use an updated implementation of the knowledge base and interpretation algorithm, and our goal is to broaden the applicability of the system by testing and adapting to new text domains.

Empirical approaches have also become increasingly important for this task. Lauer (1995) compares several corpus-based techniques for bracketing noun compounds. Rosario and Hearst (2001) cast the problem as a classification task, and train a machine learning approach on hand-labeled examples from biomedical texts. In applications where domain knowledge has not yet been created or collected, empirical methods offer a good solution. In addition they often provide principled ways of ranking their results based on corpus frequency. But Rosario and Hearst (2001), along with the number of experiments combining empirical NLP systems with WordNet (Fellbaum 1998), demonstrate that ontological knowledge helps when it is available. We describe some ways to combine our approach with empirical analysis in the section on Future Work.

Nominal Relations and the Knowledge Base

As the example above demonstrates, the relationship labels we assign to compound nouns rely on the domain-specific concept definitions stored in the knowledge base. To store and access these definitions we use the Scone Knowledge Representation System (Fahlman 2005).

The Scone Knowledge Representation System includes a representation language and an inference engine. The engine provides functions for adding new knowledge, and for modifying or querying existing knowledge. Scone is open-source, and it also includes a small but growing collection of core "common sense" facts. The work presented here is part of our larger effort to acquire additional Scone knowledge by understanding statements in Simple English.

Scone knowledge is stored in a semantic graph, where every concept is a Scone *element*. Scone can represent taxonomic information ("apple is a type of fruit"), as well as role relations for objects ("a piece of fruit has seeds") and for actions ("eating has an agent and a thing-eaten"). Statements are also supported, and they may take other statements as arguments ("Sue believes that John owns this car."). During analysis of a noun compound, Scone provides a list of role relations that are candidate labels for the compound ("{driving} is an {action} with a {driver (role)} and a {vehicle (role)}"). Scone imposes selectional restrictions on role fillers ("A {driver} must be a {person}"), and the engine provides an efficient mechanism to check for violation of these restrictions.

Building The Domain Model

Knowledge bases are hand-coded in the representation language, and may be saved and loaded as independent subdomain models. For our examples, we have augmented a general world-knowledge model by adding domain-specific knowledge based on the development text. Our goal is to establish the re-usability of domain knowledge and the interpretation algorithm, so we will first give some examples from the development domain, and then describe how we can apply the system to a new domain and re-use domain knowledge in new tasks.

We begin by choosing a development set of noun compounds that employ a semantically diverse set of relations. The development set described in (McDonald 1982) was our starting point. A sample of these compounds is shown in Figure 3.

Next, we expand our cross-domain core knowledge base to cover this development set. The core ontology includes 430 elements. The upper levels of the ontology include (among many others): {intangible}, {tangible}, {stuff}, {physical object}, {solid} {liquid} and {gas}, {animate}, {inanimate}, {person}, {animal} {plant}, {organization}, {time}. We build up domain coverage by adding knowledge to this re-usable core. To cover the compounds in our development set, we added 137 additional elements with role-structures, bringing the total size of the knowledge base to 567 elements. This knowledge base is very small, but it demonstrates our approach by being diverse enough to include a range of syntactic and semantic compound classes.

The scope of the knowledge base is important for the success of the algorithm because the vocabulary of noun compound relations that we can identify comes directly from the

Figure 3: A Sample from the Development Set with their Relations. When role relations are inherited from a supertype of the head noun, the supertype is also given.

fuel economy {fuel} fills {economize-what (role)}
oil production {oil} fills {produces-what (role)}
underground oil {underground} fills {source (role)} of a {material}
passenger car {person} fills {transports-what (role)} of a {vehicle}
pickup truck {pickup} fills {intended-use (role)} of a {tool}
car wheel the {wheel} which is {part-of (role)} of a {automobile}
petroleum products {petroleum} fills {composing-material (role)} of {physical object}
oil embargo {oil} fills {restricts-what (role)}
gasoline shortage {shortage-of (role)}
oil imports {Equality} of {imports-what (role)} and {oil}
automobile industry {industrial-product (role)}
consumer preference {who-prefers (role)}

roles and statements that can join two nominal concepts in the knowledge base.

Interpretation Rules

While the knowledge base provides a vocabulary of semantic relations along with some important constraints, as a generative source for semantic labels it will over-generalize. For example, the knowledge base contains is-not-a links in addition to is-a links. But behavioral research shows that these relationships don't occur in noun compounds (Downing 1977).

For this reason, the search for which relation holds in a given compound is driven by interpretation rules. Our rules are modeled on the cases described in (McDonald 1982), and they impose an order of preference based on the syntactic structures that can occur in a compound.

Part of the difficulty in analyzing noun compounds is that there are few syntactic cues. In fact, early linguistic literature on noun compounds (Lees 1960) describes them as a deletion phenomenon, where the compound is the short form for an entire relative clause: "A man who takes away the garbage" becomes "garbage man." However some syntax does remain in the form of word order. Using a combination of word order and the interpretation rules to drive our search, we are able to recognize the difference between *passenger car* and *car passenger*. More details are given below.

Compounds of Length Two

For compounds of length two (N_1, N_2), English word order usually places the head noun on the right (N_2), with a nominal pre-modifier on the left (N_1). In this case, we can drive the search for a semantic relation label with the following rule: examine the list of roles attached to N_2, and determine which of these may be filled by N_1. Return all licensed roles. This is the strategy which finds the meaning of *pickup truck* in Figure 3. When this rule fails to return a licensed meaning, the reverse word order may be tried: take N_1 as the head noun and search for roles that may be filled by N_2. This strategy is necessary for analyzing *car wheel*, also in Figure 3.

Compounds of Length Three or More

Several systems treat the two-noun, left-to-right compound as the basic case, and derive interpretations for longer compounds by recursively attaching left-hand nouns as modifiers to the current right-hand head. Systems like (Barker & Szpakowicz 1998), on the other hand, use a trained model to choose a bracketing first, then assign semantic labels to the bracketed constituents afterward.

Our system is intermediate between these two, in that it applies several additional rules to account for compounds of length three or more, but does not consider all possible head-modifier bracketings. These rules build role-filler chains by parsing from left to right (*coal mine supervisor*) or from right to left (*glass wine glass*). There are also two special rules for nominalized verbs, which are allowed to take sub-heads as fillers to their left and to fill roles to their right (*tension adjustment screw*).

The interpretation rules listed above are ordered, so that they can be used to perform heuristic disambiguation. In general the system applies all of the rules and delivers all possible interpretations that are licensed by the knowledge base and which fit any of the rules. When a single answer is required, the higher-ordered rules fire first and the first licensed interpretation is returned. For long compounds, this rule ordering means that simple left-to-right head-modifier bracketing is tried first. It is a priority in our future work to add a more sophisticated selection mechanism for choosing the best interpretation.

Re-using the Components

In an experiment to test the domain-independence of the interpretation rules, we tested the system on a number of example compounds from text in a new domain.

The Radar Email Corpus

Scone is currently in use as the knowledge representation system for the Radar project (CMU-RADAR 2004). Radar is a software tool that emulates a personal assistant; features include scheduling, handling email and allocating resources like conference rooms and video equipment. When a user receives a new email, Radar labels it automatically with a set of actions that the user may need to take in response. The classifier that assigns these labels is an SVM trained on lexical features of the email. It was implemented by the authors of (Klimt & Yang 2004) and uses a similar algorithm to the one described there. Scone plays a role in the classification of emails by providing semantic features which augment the lexical features in the classifier. The Scone analyzer for Radar does not include compound noun analysis.

Scone developers created a new knowledge base for the Radar project. It models the relationships that are important in the domain of conference planning, which was a task for users of Radar during recent evaluations. It includes instances of people and locations that occur in training emails, collected semi-automatically. We refer to this knowledge base as the Radar-kb. It contains more than 6,500 concepts and instances.

Expansion of the Knowledge Base

This body of domain knowledge gives us the framework we need to test the independence of our analysis rules. We have hypothesized that the rules can be applied to text in any domain where we have sufficient knowledge. The Radar-kb provides background knowledge for a new domain so that we can test our hypothesis.

We have a corpus of emails that was used to train the classifier described above. From these texts we automatically extracted a collection of 100 new noun compounds which we use to test our analysis rules. First, we have to augment the Radar-kb with concepts that occur in the noun compounds. Only 30 of 282 vocabulary items in our collection had definitions in the Radar-kb. Without modifying the analysis rules at all, we expanded the knowledge base by hand to cover several examples from the new domain.

Coverage of the Analysis Rules

In general, the analysis rules do capture a significant number of the new compounds. Of the 100 new compounds we collected, 51 could be captured by straightforward role-filling with either left or right bracketing. Three involved nominalized verbs. Example compounds covered by each of the four rules are given in Figure 4.

11 of the remaining compounds were proper names of companies or buildings, which have fixed meanings in the knowledge base and do not need to be parsed by the compound analyzer at all. We also captured the phrase "head start", which we consider to be idiomatic. 7 of the remaining compounds were mis-labeled as noun compounds during test set extraction (*Dear Blake, Mr. Robertson, Monday night*). 28 compounds did represent problematic cases for our algorithm and are discussed below.

Example Output in the New Domain

Below are several examples of real system output for compounds in the Radar domain. In cases where the system has to generate a new element to represent the compound, output includes an intermediate "RESULT FOUND" explanation. In cases where the compound meaning is interpreted as a combination of existing nodes, there is only the final result indicated by "MEANING:".

"event title"

Figure 4: Example Compounds from Radar in Four Analysis Cases

Event Title Left-to-Right: a {title (role)} of an {event}
crowd control Right-to-Left: {crowd} fills {controls-what (role)}
opening reception Nominalized verb at start: {open} fills {purpose-of (role)}
security request submission Nominalized verb at end: {request} with {request-what (role)} of {security} fills {submit-what (role)}

```
 CHECKING PSEUDO ADJS
CONTEXT TOO GENERAL. REFUSING TO
FILL part ROLE.
NO LTR RELATIONSHIP. LOOKING RTL
MEANING: THE {conference event
title} OF A {conference event}
```

"crowd control"

```
 RESULT FOUND: {crowd-control-0}
IS A TYPE OF {control} WHERE THE
ROLE OF
{controls-what (role) of
crowd-control-0}
IS A {crowd}
MEANING: {crowd-control-0}
```

"opening reception"

```
 RESULT FOUND: {event
opening-Reception-0}
IS A TYPE OF {Reception} WHERE THE
ROLE OF
{event purpose (role) of event
opening-Reception-0}
IS A {event opening}
MEANING: {event
opening-Reception-0}
```

"security service request submission"

```
RESULT FOUND: {security
service-request-0}
IS A TYPE OF {request} WHERE THE
ROLE OF
{request-what (role) of security
service-request-0}
IS A {security service}
```

```
RESULT FOUND: {security
service-request-0-submission-0}
IS A TYPE OF {submission} WHERE THE
ROLE OF
{submit-what (role) of security
service-request-0-submission-0}
IS A {security service-request-0}
MEANING: {security
service-request-submission-0-0}
```

"breakfast vendor"

```
RESULT FOUND: {conference
breakfast-vendor-0}
IS A TYPE OF {vendor} WHERE THE
ROLE OF
{provides-what (role) of conference
breakfast-vendor-0}
IS A {conference breakfast}
MEANING: {conference
breakfast-vendor-0}
```

"NSF Conference"

```
RESULT FOUND: {nsf-conference-0}
IS A TYPE OF {conference} WHERE THE
ROLE OF
{sponsor (role) of
nsf-conference-0}
IS A {National Science Foundation}
MEANING: {nsf-conference-0}
```

"planning committee"

```
RESULT FOUND:
{planning-committee-0}
IS A TYPE OF {committee} WHERE THE
ROLE OF
{organizing purpose (role) of
planning-committee-0}
IS A {planning}
MEANING: {planning-committee-0}
```

"committee headquarters"

```
MEANING: THE {headquarters (role)}
OF A {committee}
```

"planning committee headquarters"

```
MEANING: THE {headquarters (role)}
OF A {planning-committee-0}
```

Problematic Cases

There were 28 compounds in the collection we extracted which do not fit into our current analysis paradigm and point to improvements that we can make to the system. They fall into two groups: compounds where the head noun is the name of a role (rather than a role filler) for the modifier, and compounds where the modifier is actually a subtype of the head noun. An example from the role-name category is "event location." The proper interpretation is a role itself, without any filler: the {location (role)} of an {event}. The current rules focus on role fillers, and do not promote the role names themselves as candidate interpretations.

An example from the subtype category is "banquet event." The proper interpretation is an {event} which is also a {banquet}. Again, the focus on roles and fillers prevents our system from finding the the meaning for this compound. For this example, the system erroneously returns *a {event} which has a {sub-event} {banquet}*.

This kind of error could be described as a pruning error: we fail to consider the appropriate relation when searching for connections in the knowledge base. The place to fix this behavior is in the analysis rules.

Conclusions

We have presented a system for analyzing noun compounds using a knowledge base. The system uses knowledge resources and domain rules that are independent of each other, and can be re-used in new domains and NLP tasks. We have given some successful examples in adapting knowledge developed for text classification to cover noun compounds, and in applying the analysis rules to compounds from a new text domain.

The problematic cases point to specific improvements that we can make to the system. We should expand the analysis rules to allow role names, and not just role fillers, to be considered during the search. We may also expand the rules to allow compounds that join a supertype and subtype. However the prevalence of compounds that did fit into the role-filler paradigm (51 out of the test set of 89) indicates that such new rules should be tried only after our existing rule set has been exhausted.

Further Work

The knowledge in the Radar-kb was developed as a general domain model, and was further adapted with the task of email classification in mind. While this knowledge did not prove broad enough to cover the noun compounds we found in the same domain, we would still claim that it was re-usable on the basis that a few new concepts were added to the existing framework, and still resulted in accurate analyses. We have not yet investigated re-usability of knowledge in the reverse direction: could the concepts we added for noun compound analysis improve performance of the email classifier? This is an experiment which we plan to conduct in the near future.

We will also continue to make improvements to the algorithm itself. In the current system, interpretation rules are applied whenever possible and the result is a full list of licensed interpretations. The next step will be to add the ability to rank these interpretations. Both (Vanderwende 1994) and (Finin 1980), for example, include a matching score for the fitness of a role and filler pair. We are currently implementing automatic measures of semantic distance in Scone that we will use in ranking role-filler matches, as well. This will allow us to rank the resulting compound interpretations.

References

Barker, K., and Szpakowicz, S. 1998. Semi-automatic recognition of noun modifier relationships. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*.

CMU-RADAR. 2004. Radar project homepage. *http://www.radar.cs.cmu.edu/external.asp*.

Downing, P. 1977. On the creation and use of english compound nouns. *Language* 53(4):810–842.

Fahlman, S. E. 2005. Scone User's Manual, *http://www.cs.cmu.edu/ sef/scone* .

Fan, J.; Barker, K.; and Porter, B. W. 2003. The knowledge required to interpret noun compounds. In *IJCAI 2003*, 1483–1485.

Fellbaum, C., ed. 1998. *WordNet An Electronic Lexical Database*. MIT press.

Finin, T. 1980. The semantic interpretation of nominal compounds. In *Proceedings of the First Annual National Conference on Artificial Intelligence (AAAI-80)*.

Klimt, B., and Yang, Y. 2004. Email folder classification using threads. In *submitted*.

Lauer, M. 1995. Corpus statistics meet the noun compound: Some empirical results. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, 47–54. Morristown, NJ, USA: Association for Computational Linguistics.

Lees, R. B. 1960. *The Grammar of English Nominalizations*. Bloomington: Indiana University Press.

Leonard, R. 1984. *The Interpretation of English Noun Sequences on the Computer*. Amsterdam: North-Holland.

McDonald, D. B. 1982. Understanding noun compounds. Technical Report CMU-CS-82-102, Department of Computer Science, Carnegie Mellon University.

Rosario, B., and Hearst, M. 2001. Classifying the semantic relations in noun compounds. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*.

Sag, I. A.; Baldwin, T.; Bond, F.; Copestake, A. A.; and Flickinger, D. 2002. Multiword expressions: A pain in the neck for NLP. In *CICLing '02: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, 1–15. London, UK: Springer-Verlag.

Vanderwende, L. 1994. Algorithm for automatic interpretation of noun sequences. In *Proceedings of the 15th conference on Computational linguistics*, 782–788. Morristown, NJ, USA: Association for Computational Linguistics.