# Interpreting Anaphoric Shell Nouns using Antecedents of Cataphoric Shell Nouns as Training Data

Varada Kolhatkar Department of Computer Science University of Toronto varada@cs.toronto.edu Heike Zinsmeister Institut für Germanistik Universität Hamburg heike.zinsmeister@uni-hamburg.de

Graeme Hirst Department of Computer Science University of Toronto gh@cs.toronto.edu

#### Abstract

Interpreting anaphoric shell nouns (ASNs) such as this issue and this fact is essential to understanding virtually any substantial natural language text. One obstacle in developing methods for automatically interpreting ASNs is the lack of annotated data. We tackle this challenge by exploiting *cataphoric* shell nouns (CSNs) whose construction makes them particularly easy to interpret (e.g., the fact that X). We propose an approach that uses automatically extracted antecedents of CSNs as training data to interpret ASNs. We achieve precisions in the range of 0.35 (baseline = 0.21) to 0.72 (baseline = 0.44), depending upon the shell noun.

## 1 Introduction

Anaphors such as *this fact* and *this issue* encapsulate complex abstract entities such as propositions, facts, and events. An example is shown below.

Here is another bit of advice: Environmental Defense, a national advocacy group, notes that "Mowing the lawn with a gas mower produces as much pollution in half an hour as driving a car 172 miles." <u>This fact</u> may help to explain the recent surge in the sales of the good old-fashioned push mowers or the battery-powered mowers.

Here, the anaphor *this fact* is interpreted with the help of the clausal antecedent marked in bold. The antecedent here is complex because it involves a

number of entities and events (e.g., mowing the lawn, a gas mower) and relationships between them, and is abstract because the antecedent itself is not a purely physical entity.

The distinguishing property of these anaphors is that they contain semantically rich abstract nouns (e.g., *fact* in (1)) which *characterize* and *label* their corresponding antecedents. Linguists and philosophers have studied such abstract nouns for decades (Vendler, 1968; Halliday and Hasan, 1976; Francis, 1986; Ivanic, 1991; Asher, 1993). Our work is inspired by one such study, namely that of Schmid (2000). Following Schmid, we refer to these abstract nouns as *shell nouns*, as they serve as conceptual *shells* for complex chunks of information. Accordingly, we refer to the anaphoric occurrences of shell nouns (e.g., *this fact* in (1)) as *anaphoric shell nouns* (ASNs).

An important reason for studying ASNs is their ubiquity in all kinds of text. Schmid (2000) observed that shell nouns such as *fact, idea, point,* and *problem* were among the 100 most frequently occurring nouns in a corpus of 225 million words of British English. Moreover, ASNs can play several roles in organizing a discourse such as encapsulation of complex information, cohesion, and topic boundary marking. So correct interpretation of ASNs can be an important step for correct interpretation of a discourse, and in a number of NLP applications such as text summarization, information extraction, and non-factoid question answering.

Despite their importance, ASNs have not received much attention in Computational Linguistics, and research in this field remains in its earliest stages. At present, the major obstacle is that there is very little annotated data available that could be used to train a supervised machine learning system for robustly interpreting these anaphors, and manual annotation is an expensive and time-consuming task.

We tackle this challenge by exploiting a category of examples, as shown in (2), whose construction is particularly easy to interpret.

#### (2) Congress has focused almost solely on <u>the fact</u> that special education is expensive – and that it takes away money from regular education.

Here, in contrast with (1), the fact is not anaphoric in the traditional sense, but is an easy case of a forward-looking anaphor - a cataphor. While the resolution process of this fact in (1) is quite challenging as it requires the use of semantics and world knowledge, it is fairly easy to interpret the fact in (2) based on the syntactic structure alone. We refer to these easy-to-interpret cataphoric occurrences of shell nouns as cataphoric shell nouns (CSNs). The interpretation of both ASNs and CSNs will be referred to as *antecedent*.<sup>1</sup> The antecedent of *the fact* in (2) is given in the post-nominal that clause. We use the term *shell concept* to refer to the general notion of a shell noun, i.e., the semantic type of the antecedent. For example, the notion of an issue is an important problem which requires a solution.

In this work, we propose an approach to interpret ASNs that exploits *unlabelled* but easy-to-interpret CSN examples to extract characteristic features associated with the antecedent of different shell concepts. We evaluate our approach using crowdsourcing. Our results show that these unlabelled CSN examples provide useful linguistic properties that help in interpreting ASNs.

## 2 Related work

The resolution of anaphors to non-nominal antecedents has been well analyzed taking discourse structure and semantic types into account (Webber, 1991; Passonneau, 1989; Asher, 1993). Most work in machine anaphora resolution, however, is restricted to anaphora that involve nominal antecedents only (Poesio et al., 2011). There are some notable exceptions which have tackled the challenge of interpreting non-nominal antecedents (Eckert and Strube, 2000; Strube and Müller, 2003; Byron, 2004; Müller, 2008). These approaches are limited as they either rely heavily on domain-specific syntactic and semantic annotation or prepossessing, or mark only verbal proxies for non-nominal antecedents.

Recently, Kolhatkar and Hirst (2012) presented a machine-learning based resolution system for *this issue* anaphora, identifying full syntactic phrases as antecedents. Although they achieved promising results, their approach was limited in two respects. First, it focused on only one type of shell noun anaphora (*issue* anaphora). Second, their training data was restricted to MEDLINE abstracts in which *this issue* is used in a rather systematic way. Furthermore, their work is based on manually labelled ASN antecedents, whereas we use automatically identified CSN antecedents, which we interpret as explicitly expressed antecedents in comparison to the more implicitly expressed ASN antecedents.

Using explicitly expressed structure in the text to identify implicit structure is not new. The same idea has been applied before in computational linguistics. Marcu and Echihabi (2002) identified implicit discourse relations using explicit ones. Markert and Nissim (2005) used Hearst's (1992) explicit patterns to learn lexical semantic relations for NPcoreference and *other*-anaphora resolution from the web. Although our work focuses on a different topic, the methodology is in the same vein.

#### **3** Hypothesis of this work

The hypothesis of this work is that CSN antecedents and ASN antecedents share some linguistic properties and hence linguistic knowledge encoded in CSN antecedents will help in interpreting ASNs. Accordingly, we examine which features present in CSN antecedents are relevant in interpreting ASNs.

The motivation and intuition behind this hypothesis is as follows. The antecedents of both ASNs and CSNs represent the corresponding shell concept. So are there any characteristic features associated with this shell concept? Do speakers of English follow certain patterns of syntactic shape or words, for instance, when they state facts, decisions,

<sup>&</sup>lt;sup>1</sup>Sadly, the more-logical term for the interpretation of a CSN, *succedent*, does not actually exist.



Figure 1: Overview of our approach

or issues? There is an abundance of data for CSN antecedents and if we are able to capture particular linguistic characteristic features associated with a shell concept using this data, we can use this information to interpret ASNs. For instance, example (2) demonstrates characteristic properties of antecedents of the shell noun *fact* including that (a) they are propositions and are generally expressed with clauses or sentences rather than noun phrases, and (b) they are generally expressed in the present tense. Observe that these properties also hold for the antecedent of *this fact* in example (1).

We test our hypothesis by building machine learning models that are trained on automatically extracted CSN antecedents and then applying these models to recover ASN antecedents. Figure 1 shows an overview of our methodology.

## 4 Background

**Formal definition** *Shell-nounhood* is a functional notion; it is defined by the *use* of an abstract noun rather than the inherent properties of the noun itself (Schmid, 2000). An abstract noun is a shell noun when the speaker decides to *use* it as a shell noun.

**Shell noun categorization** Schmid (2000) gives a list of 670 English nouns which are frequently used as shell nouns. He divides them into six broad semantic classes: *factual*, *linguistic*, *mental*, *modal*, *eventive*, and *circumstantial*. Table 1 shows this classification, along with example shell nouns for each category. For this work, we selected six frequently occurring shell nouns covering four of Schmid's six classes: *fact* and *reason* from factual, *issue* and *decision* from mental, *question* from linguistic, and *possibility* from modal. These shell nouns tend to have antecedents that lie within a single sentence. We excluded *eventive* and *circumstan*.

Class	Description	Examples
factual	states of affairs	fact, reason
linguistic	linguistic acts	question
mental	ideas	issue, decision
modal	judgements	possibility
eventive	events	act, reaction
circumstantial	situations	situation, way

Table 1: Schmid's classification of shell nouns. The nouns given in the *Example* column tend to occur frequently with the respective class. The shell nouns used in this work are shown in boldface.

Pattern	Example					
N-to	Several people at the group said the deci-					
	sion to write the letters was not controver- sial internally.					
N-be-to	The principal reason is to create a repre-					
	sentative government rather than to select					
	the most talented person.					
N-that	Mr. Shoval left open the <b>possibility</b> that					
	Israel would move into other West Bank					
	cities.					
N-be-that	The simple and reassuring fact is that a					
	future generation of leaders is seeking new					
	challenges during challenging times.					
N-wh	There is now some question whether the					
	country was ever really in a recession.					
N-be-wh	Of course, the central, and probably in-					
	soluble, issue is whether animal testing is					
	cruel.					

Table 2: Easy-to-interpret CSN patterns given by Schmid (2000). In the *Example* column, the patterns are marked in boldface and the antecedents are marked in italics.

*tial* classes because the shell nouns in these classes tend to have rather unclear and long antecedents.<sup>2</sup>

**Shell noun patterns** Schmid (2000) also provides a number of lexico-grammatical patterns for shell nouns. In Section 1, we noted two such patterns: *this-N* (*this fact* in example (1)) and *N-that* (*fact that* in example (2)). We also noted that CSNs with pattern *N-that* are fairly easy to interpret compared to the ASN pattern *this-N*. Table 2 shows some other easy-to-interpret CSN patterns given by Schmid. Generally, for all these patterns, the antecedent is

<sup>&</sup>lt;sup>2</sup>These observations are based on an exploratory pilot annotation we carried out on sample data of 150 ASN instances.

quite easy to extract with a few predefined rules.

Shell antecedent properties Antecedents of CSNs and ASNs share some properties while they are distinguished by others. The distinguishing property is that CSNs, by their construction, have their antecedents in the same sentence, as shown in example (2). On the other hand, ASNs can have long-distance as well as short-distance antecedents.<sup>3</sup> The common properties are as follows. First, antecedents of both ASNs and CSNs represent the corresponding shell concept, e.g., the notion of a fact or an issue. Second, in both cases, the antecedents are complex abstract entities, which involve a number of entities and relationships between them. Finally, in both cases, there is no one-to-one correspondence between the syntactic type of an antecedent and semantic type of its referent (Webber, 1991). For instance, a semantic type such as *fact* can be expressed with different syntactic shapes such as a clause, a verb phrase, or a complex sentence. Conversely, a syntactic shape, such as a clause, can function as several semantic types, including fact, proposition, and event.

#### 5 Training phase

As shown in Figure 1, the goal of the training phase is to build training data from CSNs and their antecedents and train models which can be used for resolving ASNs.

## 5.1 The CSN corpus

We automatically constructed a corpus, a subset of the New York times (NYT) corpus<sup>4</sup>, which contains 211,722 sentences following CSN patterns from Table 2. We considered part-of-speech information<sup>5</sup> while looking for the patterns. For instance, instead of the pattern *N*-that, we actually looked for {shell\_noun\_NN that\_IN}.

#### 5.2 Antecedent extractor for CSNs

The goal of the antecedent extractor is to create automatically labelled CSN antecedent data. Recall that antecedents of CSNs can be extracted using simple predefined rules that are based on the syntactic structure alone. For instance, the antecedent extraction rule for example (2) would be: if the example follows the pattern *fact-that*, extract the post-nominal *that* clause as the antecedent. To come up with a list of such extraction rules, we systematically analyzed a sample of examples (about 20 examples) of each pattern for each shell noun. Table 3 summarizes the resulting antecedent extraction rules.

The actual antecedent extraction works as follows. First, we parsed the examples from the CSN corpus using the Stanford parser.<sup>6</sup> Then for each example, we applied rules from Table 3 depending on the shell noun and the pattern it follows to extract an appropriate syntactic constituent as the CSN antecedent. For instance, for the noun fact following the N-that pattern, as in example (2), we first looked for the NP constituent containing the shell noun *fact*, and then extracted the sentential constituent following the NP constituent as the CSN antecedent. Although, in most of the cases, the antecedent is given in the post-nominal wh, that, or infinitive clauses, sometimes it is not present in the immediately following clause but is given only as a predicate, as shown in (3).

(3) The primary reason that the archdiocese cannot pay teachers more is that its students cannot afford higher tuition.

In such cases, we looked for the pattern (*VP* (*VB*  $be\_verb$ ) *X*) in the right sibling of the NP containing the pattern *shell\_noun-that* and extracted *X* as the CSN antecedent.

Two contradictory goals need to be achieved while extracting antecedents of CSNs. The first requires only considering CSNs with high-confidence patterns, whereas the second requires considering as many patterns as possible to allow a wide variety of antecedent examples with different linguistic properties (e.g., syntactic shape). Our antecedent extractor tries to find a balance between the two goals.

<sup>&</sup>lt;sup>3</sup>In our annotated sample data, we observed ASN antecedents as close as the same sentence and as far as 7 sentences away from the anaphor.

<sup>&</sup>lt;sup>4</sup>http://www.ldc.upenn.edu/Catalog/

catalogEntry.jsp?catalogId=LDC2008T19

<sup>&</sup>lt;sup>5</sup>http://nlp.stanford.edu/software/tagger. shtml

<sup>&</sup>lt;sup>6</sup>http://nlp.stanford.edu/software/lex-parser. shtml

	fact	reason	issue	decision	question	possibility
N-to	_	_	_	inf clause	predicate	inf clause
N-be-to	-	inf clause	inf clause	inf clause	<i>inf/wh</i> clause	inf clause
N-that	that clause	predicate	predicate	-	predicate	that clause
N-be-that	that clause	that clause				
N-wh	_	predicate	wh clause	wh clause	wh clause	_
N-be-wh	wh clause	wh clause				

Table 3: Content extraction patterns for CSNs. Patterns in boldface are the prominent patterns for the respective shell noun. *inf* clause = infinitive clause. Discarded patterns are denoted by -.

To address the first goal, we filter examples following noisy patterns, i.e., the patterns that do not unambiguously encode antecedents of that CSN. For instance, the pattern *N-to* is a highly preferred pattern for *decision*, as shown in (4). The antecedent extraction rule here is relatively simple: if the example follows the pattern *decision-to*, extract the postnominal infinitive clause as the correct antecedent.

 (4) President Jacques Chirac's arrogant <u>decision to</u> defy the world and go ahead with two nuclear bomb tests in Polynesia deserves contempt.

But the same pattern is noisy for *reason*. In (5), for example, the actual reason is not given anywhere in the sentence. So we discard the examples following the pattern *N*-to for *reason*.

(5) Investors have had reason to worry about stocks.

We also discard examples with negative determiners, as in (6), because in such cases, the extraction rules do not precisely give the antecedent of the given CSN.

(6) He was careful to repeat anew that he had made **no decision to** go to war.

For the *N*-wh pattern, we exclude certain wh words for certain nouns. For example, we exclude the wh word which for question as the Penn Treebank tagset<sup>7</sup> does not distinguish between which as a relative pronoun and as a question. We are interested in the latter but not the former. Other discarded wh words include which and when for fact; all wh words except when and why for reason, all wh words except how and whether for issue; which, whom, when, and why for decision; which and when for question; and all wh words for possibility.

To address the second goal of allowing a wide variety of antecedent examples, we try to include as many patterns as possible for each shell noun, as shown in Table 3. For instance, the patterns *question-to* and *question-be-to* will have infinitive clauses as antecedents (marked as VP or S+VP by the parser), whereas for the examples following patterns *question-wh* and *question-be-wh* the antecedent will be in the *wh* clauses (marked as SBAR). For the pattern *question-that*, the antecedent will be in the predicate (similar to example (3)), which can be a prepositional phrase, a noun phrase or a clause.

#### 5.3 Models for CSN antecedents

The antecedent extractor gives labels for each instance in the CSN corpus. Using this labelled data, we train machine learning ranking models for different shell concepts that capture the characteristic features associated with that shell concept. The following sections describe each step of our ranking models in detail.

#### 5.3.1 Candidate extraction

The first step is to extract the set of eligible antecedent candidates  $C = \{C_1, C_2, ..., C_k\}$  for the CSN instance  $a_i$ . To train a machine learning model we need positive and negative examples. We already have positive examples for antecedent candidates the *true* antecedents given by the antecedent extractor. But we also need negative examples of antecedent candidates. By their construction, CSNs have their antecedents in the same sentence. So we extract all syntactic constituents of this sentence, given by the Stanford parser. All the syntactic constituents, except the *true* antecedent, are considered as negative examples. With this candidate extraction method, we end up with many more negative exam-

<sup>&</sup>lt;sup>7</sup>The Stanford tagger we employ uses the Penn Treebank tagset (Marcus et al., 1993).

ples than positive examples, but that is exactly what we expect with ASN antecedent candidates, i.e., the test data on which we will be applying our models.

#### 5.3.2 Features

Although our problem is similar to anaphora resolution, we cannot make use of the usual anaphora or coreference resolution features such as agreement or string matching (Soon et al., 2001) because of the nature of ASN and CSN antecedents. We came up with a set of features based on the properties that were common in both ASN and CSN antecedents, according to our judgement.

**Syntactic type of the candidate (S)** We observed that each shell noun prefers specific CSN patterns and each pattern involves a particular syntactic type. For instance, *decision* prefers the pattern *N-to* and consequently realizes as its antecedents more verb phrases than, for example, noun phrases. We employ two versions of syntactic type: fine-grained syntactic type given by the Stanford parser (e.g., NP-TMP, RRC) and coarse-grained syntactic type (e.g., NP, VP, S, PP) in which we consider ten basic syntactic categories and map all fine-grained syntactic types to these categories.

**Context features (C)** Context features allow our models to learn about the contextual clues that signal the antecedent. This class contains two features: (a) coarse-grained syntactic type of left and right siblings of the candidate, and (b) part-of-speech tag of the preceding and following words of the candidate.

**Embedding level features (E)** These features (Müller, 2008) encode the embedding level of the candidate within its sentence. We consider two embedding level features: top embedding level and immediate embedding level. Top embedding level is the level of embedding of the given candidate with respect to its top clause (the root node), and immediate embedding level is the level of embedding with respect to its immediate clause (the closest ancestor of type S or SBAR). The intuition behind this feature is that if the candidate is deep in the parse tree, it is possibly not salient enough to be an antecedent. As we consider all syntactic constituents as potential candidates, there are many that clearly cannot be antecedents. This feature will allow us to get rid of

this noise.

**Subordinating conjunctions (SC)** As we can see in Table 2, subordinating conjunctions are common with CSN and ASN antecedents. Vendler (1968) points out that the shell noun *fact* prefers a *that*clause, and *question* and *issue* prefer a *wh*-question clause. We observed that the pattern *because X* is common with *reason*. The subordinating conjunction feature encodes these preferences for different shell nouns. The feature checks whether the candidate follows the pattern *SBAR*  $\rightarrow$  (*IN sconj*) (*S*...), where *sconj* is a subordinating conjunction.

**Verb features (V)** A prominent property of CSN and ASN antecedents is that they tend to contain verbs. All examples from Table 2, for example, contain verbs. Moreover, certain shell nouns have tense and aspect preferences. For instance, for shell noun *fact*, lexical verbs in past and present tenses predominate (Schmid, 2000), whereas modal forms are extremely common for *possibility*. We use three verb features that capture this idea: (a) presence of verbs in general, (b) whether the main verb is finite or nonfinite, and (c) presence of modals.

**Length features (L)** The intuition behind these features is that CSN and ASN antecedents tend to be long, especially for nouns such as *fact*. We consider two length features: (a) length of the candidate in words, and (b) relative length of the candidate with respect to the sentence containing the antecedent.

Lexical features (LX) Our extractor gives us a large number of antecedent examples for each shell noun. A natural question is whether certain words tend to occur more frequently in the antecedent than non-antecedent parts of the sentence. To deal with this question, we extracted all antecedent unigrams (i.e., unigrams occurring in antecedent part of the sentence) and non-antecedent unigrams (i.e., unigrams occurring in non-antecedent parts of the sentence) for each shell noun. Then for all antecedent unigrams for a particular shell noun, we computed term goodness in terms of information gain (Yang and Pedersen, 1997) and considered the first 50 highly ranked unigrams as the lexical features for that noun. Note that, in contrast with the other features, these lexical features are tailored for each shell noun and are extracted a priori.

#### 5.3.3 Candidate ranking models

Now that we have the set of candidate antecedents and a set of features, we are ready to train CSN antecedent models. We follow the *candidate-ranking* models proposed by Denis and Baldridge (2008) because they allow us to evaluate how good an antecedent candidate is relative to *all* other candidates.

For every shell noun, we gather automatically extracted antecedent data given by the extractor for all instances of that shell noun. Then for each instance in this data, we extract the set *C* as explained in Section 5.3.1. For each candidate  $C_i \in C$ , we extract a feature vector to create a corresponding set of feature vectors,  $C_f = \{C_{f1}, C_{f2}, ..., C_{fk}\}$ . For every CSN  $a_i$  and a set of feature vectors corresponding to its eligible candidates  $C_f = \{C_{f1}, C_{f2}, ..., C_{fk}\}$ , we create training examples  $(a_i, C_{fi}, rank), \forall C_{fi} \in C_f$ . The rank is 1 if  $C_i$  is same as the *true* antecedent, i.e., the automatically extracted antecedent for that CSN, otherwise the rank is 2. We use the svm\_rank\_learn call of SVM<sup>rank</sup> (Joachims, 2002) for training the candidate-ranking models.

#### 6 Testing phase

In this phase, we use the learned candidate ranking models to identify the antecedents of ASNs.

#### 6.1 The ASN corpus

We started with about 450 instances for each of the six selected shell nouns (2,700 total instances), containing the pattern {*this* shell\_noun}. The instances were extracted from the NYT. Each instance contains three paragraphs from the corresponding NYT article: the paragraph containing the ASN and two preceding paragraphs as context. After automatically removing duplicates and ASNs with a nonabstract sense (e.g., *this issue* with a publicationrelated sense), we were left with 2,323 instances.

#### 6.2 Antecedent identification

**Candidate extraction** The search space of ASN antecedents is quite large for two reasons: ASNs tend to have long-distance as well as short-distance antecedents, and there is no clear restriction on the syntactic type of the antecedents. In the ASN corpus, each sentence on average had 49.5 distinct syntactic constituents given by the Stanford parser. If

we consider *n* preceding sentences, the sentence containing the anaphor, and one following sentence<sup>8</sup> as sources for antecedents, then the average number of antecedent candidates will be  $49.5 \times (n+2)$ . This is large compared to the search space of ordinary nominal anaphora. In our previous work (Kolhatkar et al., 2013), we have developed methods that identify the sentence containing the antecedent of the ASN before identifying the precise antecedent. In brief, given a set of a fixed number of sentences around the sentence containing an ASN, these methods reliably identify the sentence containing the antecedent. In this paper, we treat these methods as a black box.

Given the sentence containing the antecedent, we extract all syntactic constituents given by the Stanford parser from that sentence as potential antecedent candidates as for the training phase. In the training phase, the antecedent is always contained in the set of syntactic constituents given by the Stanford parser because the extractor obtains the appropriate antecedent using the syntactic information. But in the testing phase, we cannot guarantee that the true antecedent occurs in the extracted syntactic constituents due to the parser's errors. So for robust candidate extraction, we extract all distinct constituents from the 30-best parses instead of only considering the best parse, which increases the average number of candidates from 49.5 to 55.2.

**Feature extraction and candidate ranking** Given the antecedent candidates, feature extraction and candidate ranking are essentially the same as for the training phase, except of course we do not know the *true* antecedent. Once we have the feature vectors for each antecedent candidate, the appropriate trained model, i.e., the model trained for the corresponding shell noun, is invoked and the candidates are ranked using the svm\_rank\_classify call of SVM<sup>rank</sup>.

# 7 Evaluation

We evaluate the ranked candidates of ASN instances using crowdsourcing.

<sup>&</sup>lt;sup>8</sup>The ASN corpus contains a few cataphoric examples that do not follow the standard patterns of the CSNs shown in Table 2, but actually refer to an antecedent in the following sentence (e.g., *Mr. Dukakis put this question to him: X*).

**Interface** We chose to use CrowdFlower<sup>9</sup> as our crowdsourcing interface because of its integrated quality-control mechanism. For instance, it throws gold questions randomly at the workers and the workers who do not answer them correctly are not allowed to continue.

We presented to the crowd evaluators the ASN instances from the ASN corpus. Recall that each ASN instance is made up of the paragraph containing the ASN and two preceding paragraphs as context. We displayed the first 10 highly-ranked candidates (ordered randomly) given by our testing phase and asked the evaluators to choose the best answer that represents the ASN antecedent. We encouraged the evaluators to select *None* when they did not agree with any of the displayed answers. We also asked them how satisfied they were with the displayed answers. We provided them with three options: *unsatisfied*, *satisfied*, and *partially satisfied*.

Our job contained 2,323 evaluation units. We asked for 8 judgements per instance and paid 6 cents per evaluation unit. As we were interested in the verdict of native speakers of English, we limited the allowed demographic region to English-speaking countries.

**Results** Among the 2,323 ASN instances, 96% of them were labelled as *satisfied*, 3% as *partially satisfied* and 1% as *unsatisfied*. Only 2% of the instances were labelled as *None*. As expected, evaluators were *unsatisfied* or *partially satisfied* with the options of these instances. These results suggest that our resolution models trained on automatically extracted antecedents of CSNs bring the relevant candidates of ASN antecedents to the top, i.e., within first 10 highly-ranked candidates. This itself is a positive result given the large search space of ASN antecedent candidates (more than 55 candidates on average).

Among the evaluation units, more than half of the evaluators agreed on an answer for 1,810 units. We used these instances for further analysis.

To examine which CSN antecedent features are relevant in identifying ASN antecedents, we carried out ablation experiments with all feature class combinations. We compared the rankings given by our ranker to the crowd's answer using *precision at n* 

(P@n).<sup>10</sup> More specifically, we count the number of instances where the crowd's answers occur within our ranker's first *n* choices. P@n then is this count divided by the total number of instances. Note that P@1 is equivalent to the standard precision.

We compared our results against two baselines: *preceding sentence* and *chance*. The preceding sentence baseline chooses the previous sentence as the correct antecedent. The chance baseline chooses a candidate from a uniform random distribution over the set of 10 top-ranked candidates.

The results are shown in Table 4. Although different feature combinations gave the best results for different shell nouns, the features that occur frequently in many best-performing combinations were embedding level (E), lexical (LX), and subordinating conjunction (SC) features. The SC features were particularly effective for *issue* and *question*, where we expected patterns such as *whether X*.

Surprisingly, the syntactic type features (S) did not show up very often in the best-performing feature combinations, suggesting that the ASN antecedents had a greater variety of syntactic types than what was available in our CSN training data.

The context features (C) did not appear in any of the best-performing feature combinations. In fact, they resulted in a sharp decline in the precision. For instance, for *question*, adding the context features to the best-performing combination {E,SC,V,L,LX} resulted in a drop of 16 percentage points. This result was not surprising because although the antecedents of ASNs and CSNs share similar properties such as common words, we know that their context is generally different.

We did not observe specific features associated with Schmid's semantic categories. An exception was the E features which were particularly effective for the factual nouns *fact* and *reason*: the results with them alone gave high precision (0.68 for *fact* and 0.72 for *reason*). That said, the E features were present in most of the best-performing combinations even for the shell nouns in other semantic categories.

<sup>&</sup>lt;sup>9</sup>http://crowdflower.com/

<sup>&</sup>lt;sup>10</sup>CrowdFlower gives us a unique answer for each instance, which we take to be the crowd's answer. During annotation, every annotator is presented with a few gold questions randomly and each annotator is assigned a trust score based on her performance on these gold questions. The unique answer for an instance is the answer with the highest sum of trusts.

<i>fact</i> (43,00	0 train a	nd 472 te	est instar	nces)	reason (4	4,520 trai	in and 44	3 test inst	tances)
Features	P@1	P@2	P@3	P@4	Features	P@1	P@2	P@3	P@4
{E,L,LX}	.70*	.85	.91	.94	{E,V,L}	.72*	.86	.90	.93
$\{E,V,L,LX\}$	$.68^{*}$	.86	.92	.95	$\{E,V\}$	.72*	.85	.90	.92
$\{E,SC,L,LX\}$	.66*	.83	.92	.95	$\{E,SC,LX\}$	.69*	.84	.90	.94
PSbaseline	.40	-	-	_	PSbaseline	.44	_	_	_
<i>issue</i> (3,00	0 train a	and 303 to	est instar	nces)	decision (	42,332 tr	ain and (	390 test in	istances)
Features	P@1	P@2	P@3	P@4	Features	P@1	P@2	P@3	P@4
{SC,L}	.47*	.59	.71	.78	{E,LX}	.35*	.53	.67	.76
$\{SC,L,LX\}$	.46*	.60	.70	.81	$\{E,SC,LX\}$	.30*	.48	.65	.75
{S,E,SC,L,LX	.40*	.61	.72	.81	{E,SC,V,L,L	X}.27	.44	.57	.69
PSbaseline	.30	_	_	_	PSbaseline	.21	-	-	-
question (9,	336 trai	n and 44(	) test ins	tances)	possibility	(11,735	train and	278 test i	instances)
Features	P@1	P@2	P@3	P@4	Features	P@1	P@2	P@3	P@4
{E,SC,V,L,LX}	.70*	.82	.87	.90	{SC,L,LX}	.56*	.75	.87	.92
{E,SC,LX}	.68*	.83	.88	.91	{E,SC}	.56*	.76	.87	.91
$\{E,SC,V,LX\}$	.69*	.80	.87	.91	${E,L,LX}$	.54*	.76	.86	.91
PSbaseline	.25	-	-	_	PSbaseline	.34	_	_	_

Table 4: Evaluation of our ranker for antecedents of six ASNs. For each noun we show the three best-performing feature combinations. P@n is the precision at rank n (P@1 = standard precision). Boldface indicates the best in the column. PSbaseline = preceding sentence baseline. The P@1 results significantly higher than PSbaseline are marked with \*(two-sample  $\chi^2$  test: p < 0.05). The chance baseline results were 0.1, 0.2, 0.3, and 0.4 for P@1, P@2, P@3, and P@4, respectively.

The only previous work with which our results could be compared is that of Kolhatkar and Hirst (2012). The work reports precision in the range of 0.41 to 0.61 in resolving this issue anaphora in the Medline domain. In our case, for this issue instances from the NYT corpus, we achieved precision in the range of 0.40 to 0.47. Furthermore, we applied our models to resolve this issue instances from Kolhatkar and Hirst's (2012) work.<sup>11</sup> Even with models trained on automatically labelled CSN antecedents, we achieved similar results to Kolhatkar and Hirst's results: P@1 of 0.45, P@2 of 0.59, P@3 of 0.65, and P@4 of 0.67. These results show the domain robustness of our methods with respect to the shell noun issue. Recall that Kolhatkar and Hirst (2012) looked at only very specific cases of this issue and used manually annotated data (Section 2), as opposed to the automatically extracted CSN antecedent data we use.

#### 8 Discussion and conclusion

The goal of this paper was to examine to what extent CSNs help in interpreting ASNs. Based on the evaluators' satisfaction level and very few *None* responses, we conclude that our models trained on CSN antecedents were able to bring the relevant ASN antecedent candidates into the top 10 candidates.

When we applied the models trained on CSN antecedents to interpret ASNs, we achieved precision in the range of 0.35 to 0.72. The precision results as high as 0.72 for *reason* and 0.70 for *fact* and *question* support our hypothesis that the linguistic knowledge provided by CSN antecedents helps in identifying the antecedents of ASNs. We observed different behaviour for different nouns. The mental nouns *issue* and *decision* in general were harder to interpret than other shell nouns. The models trained on CSNs achieved precisions of 0.35 for *decision* and 0.47 for *issue*. So there is still much room for improvement. That said, for the same nouns, the antecedents were in the first four ranks about 76% to 81% of the times,

<sup>&</sup>lt;sup>11</sup>We thank an anonymous reviewer for suggesting this to us.

suggesting that in future research, our models can be used as base models to reduce the large search space of ASN antecedent candidates.

We observed a wide range of performance for different shell nouns. One reason is that the size of the training data was different for different shell nouns. After excluding the noisy examples (Section 5.2), there were about 43,000 training examples for *fact*, but only about 3,000 for *issue*. In addition, a particular shell concept itself can be difficult, e.g., the very idea of what counts as an *issue* is more fuzzy than what counts as a *fact*.

One limitation of our approach is that it only learns the properties that are present in CSN antecedents. However, ASN antecedents have additional properties which are not always captured by CSN antecedents. For instance, for the shell noun *decision*, most of the training examples were infinitive phrases of the form *to X*. But antecedents of the ASN *decision* were mostly court decisions and were expressed with full sentences.

Moreover, although the models trained on CSN antecedents are able to encode characteristic features associated with the general shell concept, they are unable to distinguish between two competing candidates both containing the characteristic features of that shell concept. For instance, our approach will not be able to handle the constructed examples in (7).

- (7) The teacher erased the solutions before John had time to copy them out, as he had momentarily been distracted by a band playing outside.
  - a) <u>**This fact**</u> infuriated him, as the teacher always erased the board quickly and John suspected it was just to punish anyone who was lost in thought, even for a moment.
  - b) <u>This fact</u> infuriated the teacher, who had already told John several times to focus on class work.

Here, both propositions possess properties of the shell concept *fact*. Understanding the context of the anaphor itself is crucial in correctly identifying the fact in each case, which cannot be learnt from CSN antecedents due to their specific context patterns.

A number of extensions are planned for this work. First, we plan to use both kinds of data, CSN and ASN antecedent data, which will give us a basis for developing a better performing ASN resolver. We also plan to incorporate contextual features (e.g., right-frontier rule (Webber, 1991) and context ranking (Eckert and Strube, 2000)). Finally, we will examine whether a model trained for one shell noun can be generalized to other shell nouns from the same semantic category.

#### Acknowledgements

We thank the anonymous reviewers for their constructive comments. This material is based upon work supported by the United States Air Force and the Defense Advanced Research Projects Agency under Contract No. FA8650-09-C-0179, Ontario/Baden-Württemberg Faculty Research Exchange, and the University of Toronto.

#### References

- Nicholas Asher. 1993. *Reference to Abstract Objects in Discourse*. Kluwer Academic Publishers, Dordrecht, Netherlands.
- Donna K. Byron. 2004. *Resolving pronominal reference to abstract entities*. Ph.D. thesis, Rochester, New York: University of Rochester.
- Pascal Denis and Jason Baldridge. 2008. Specialized models and ranking for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 660–669, Honolulu, Hawaii, October. Association for Computational Linguistics.
- Miriam Eckert and Michael Strube. 2000. Dialogue acts, synchronizing units, and anaphora resolution. *Journal of Semantics*, 17:51–89.
- Gill Francis. 1986. *Anaphoric Nouns*. Discourse Analysis Monographs 11, Birmingham: English Language Research, University of Birmingham.
- M. A. K. Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*. Longman Pub Group.
- Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *In Proceedings of the 14th International Conference on Computational Linguistics*, pages 539–545, Nantes, France. Association for Computational Linguistics.
- Roz Ivanic. 1991. Nouns in search of a context: A study of nouns with both open- and closed-system characteristics. *International Review of Applied Linguistics in Language Teaching*, 29:93–114.
- Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In ACM SIGKDD Conference

on Knowledge Discovery and Data Mining (KDD), pages 133–142.

- Varada Kolhatkar and Graeme Hirst. 2012. Resolving "this-issue" anaphora. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 1255–1265, Jeju Island, Korea, July. Association for Computational Linguistics.
- Varada Kolhatkar, Heike Zinsmeister, and Graeme Hirst. 2013. Annotating anaphoric shell nouns with their antecedents. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 112–121, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Daniel Marcu and Abdessamad Echihabi. 2002. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 368–375, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.
- Christoph Müller. 2008. Fully Automatic Resolution of It, This and That in Unrestricted Multi-Party Dialog. Ph.D. thesis, Universität Tübingen.
- Rebecca J. Passonneau. 1989. Getting at discourse referents. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, pages 51–59, Vancouver, British Columbia, Canada. Association for Computational Linguistics.
- Massimo Poesio, Simone Ponzetto, and Yannick Versley. 2011. Computational models of anaphora resolution: A survey. Unpublished.
- Hans-Jörg Schmid. 2000. English Abstract Nouns As Conceptual Shells: From Corpus to Cognition. Topics in English Linguistics 34. Mouton de Gruyter, Berlin.
- Wee Meng Soon, Hwee Tou Ng, and Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.
- Michael Strube and Christoph Müller. 2003. A machine learning approach to pronoun resolution in spoken dialogue. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, pages 168–175, Sapporo, Japan, July. Association for Computational Linguistics.
- Zeno Vendler. 1968. *Adjectives and Nominalizations*. Mouton de Gruyter, The Hague.
- Bonnie Lynn Webber. 1991. Structure and ostension in the interpretation of discourse deixis. In *Language and Cognitive Processes*, pages 107–135.

Yiming Yang and Jan O. Pedersen. 1997. A comparative study on feature selection in text categorization. In Proceedings of the 14th International Conference on Machine Learning, pages 412–420, Nashville, TN.