RESOLVING SHELL NOUNS

by

Varada Kolhatkar

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Abstract

Resolving Shell Nouns

Varada Kolhatkar

Doctor of Philosophy

Graduate Department of Computer Science

University of Toronto

2015

Shell nouns are abstract nouns, such as *fact*, *issue*, *idea*, and *problem*, which, among other functions, facilitate efficiency by avoiding repetition of long stretches of text. An example is shown in (1). Shell nouns encapsulate propositional content, and the process of identifying this content is referred to as *shell noun resolution*.

(1) Living expenses are much lower in rural India than in New York, but **this fact** is not fully captured if prices are converted with currency exchange rates.

This dissertation presents the first computational work on resolving shell nouns. The research is guided by three primary questions: first, how an automated process can determine the interpretation of shell nouns; second, the extent to which knowledge derived from the linguistics literature can help in this process; and third, the extent to which speakers of English are able to interpret shell nouns.

I start with a pilot study to annotate and resolve occurrences of *this issue* in the Medline abstracts. The method follows a typical problem-solving procedure used in computational linguistics: manual annotation, feature extraction, and supervised machine learning. The results illustrate the feasibility of annotating and resolving shell nouns, at least in the closed domain of Medline abstracts. Next, I move to developing general algorithms to resolve a variety of shell nouns in the newswire domain. The primary challenge was that each shell noun has its own idiosyncrasies and there was no annotated data available for this task. I developed a number of computational methods for resolving shell nouns that do not rely on manually annotated data.

The methods combine lexico-syntactic knowledge and features derived from the linguistic literature and techniques in statistical natural language processing.

For evaluation, I developed annotated corpora for shell nouns and their content using crowdsourcing. The annotation results showed that the annotators agreed to a large extent on the shell content. The evaluation of resolution methods showed that knowledge derived from the linguistics literature helps in the process of shell noun resolution, at least for shell nouns with strict semantic expectations.

Acknowledgements

I enjoyed my time as a graduate student here at the University of Toronto, and I would like to thank the people who made this experience so joyful. First of all, I express my heartfelt gratitude towards my advisor Graeme Hirst. I deeply appreciate all the advice, opportunities, support, and encouragement that he has given me during my Ph.D. I can't thank him enough for his prompt and insightful comments on my writing. His endless energy and dedication to his students inspired me everyday. A big "thank you", Graeme!

I sincerely thank Suzanne Stevenson and Gerald Penn for being on my thesis committee and their investment in this project in their super-busy schedules. Suzanne, I am deeply grateful to you for your detailed reading of this thesis and your insightful comments and suggestions. Gerald, I thank you for many illuminating discussions and giving me new perspectives on my research. (Apart from your help in research, thank you for being so kind to me and taking me to the temples in Toronto.) It has been an honour to have Massimo Poesio as my external examiner. Thank you very much Massimo for your constructive comments and questions. I also thank Daphna Heller for being on my committee.

Heike Zinsmeister has been a vital part of my Ph.D. She's been a great collaborator, a mentor, and a friend during my Ph.D. I am forever grateful to her for her support, her invaluable feedback, and her overall attention to this project. I could not have asked for more in a collaborator and a mentor. Thank you very much, Heike!

I am grateful to Brian Budgell for discussing different research possibilities in the clinical trial domain during the beginning of my Ph.D. These discussions launched this project. Also, I thank him for annotating data for the pilot study of this project. I also thank Sara Scharf, Michel Fiallo-Perez, and all CrowdFlower annotators for sincerely annotating our data.

I thank the United States Air Force and the Defense Advanced Research Projects Agency, Ontario/Baden-Wurttemberg Faculty Research Exchange, and University of Toronto for the financial support that allowed me to pursue my Ph.D.

This acknowledgement won't be complete without thanking my Master's advisor Ted Ped-

ersen, who introduced me to the field of computational linguistics. I thank him for being a great mentor throughout my graduate studies.

I must also thank the entire staff from the department of computer science for their excellent administrative support, especially, Luna Keshwah, Marina Haloulos, Relu Patrascu; the entire computational linguistics group for CL seminars, CL Teas, reading groups, and for many interesting discussions, especially, Abdel-Rahman Mohamed, Aditya Bhargava, Afsaneh Fazly, Aida Nematzadeh, Alistair Kennedy, Chris Parisien, Eric Corlett, Erin Grant, Frank Rudzicz, Jackie C.K. Cheung, Julian Brook, Katie Fraser, Leila Chan Currie, Libby Barak, Naama Ben-David, Nona Naderi, Patricia Thaine, Paul Cook, Sean Robertson, Shunan Zhao, Siavash Kazemian, Timothy Fowler, Tong Wang, Ulrich Germann, and Vanessa Wei Feng; and post-docs and fellow graduate students from other groups for a stimulating research environment, especially, Alexander Schwing, Amin Tootoonchian, Charlie Tang, Emily Denton, Fernando Flores-Mangas, George Dahl, Ilya Sutskever, Laurent Charlin, Micha Livne, Mohammad Norouzi, Navdeep Jaitly, Niloofar Razavi, and Ryan Kiros.

Finally, I would like to thank my family and friends for their love and support. A big "thank you" to my chéri Ryan for his enormous help in this project. His insightful questions and comments helped me better understand the phenomenon of shell nouns and often gave me new perspectives on my research.

Contents

List of Tables

List of Figures

Chapter 1

Shell Nouns

"The history of language is the history of a process of abbreviation."

— Friedrich Nietzsche[1]

Languages present various techniques to avoid repetition and to convey information both efficiently and elegantly. Perhaps the most obvious and useful technique is the use of pronouns. Example (2) illustrates the use of the pronouns *he* and *him* to avoid repetition of the noun phrase *the little prince*.

(2) **The little prince** also pulled up, with a certain sense of dejection, the last little shoots of the baobabs. **He** believed that **he** would never want to return. But on this last morning all these familiar tasks seemed very precious to **him**. And when **he** watered the flower for the last time, and prepared to place her under the shelter of her glass globe, **he** realised that **he** was very close to tears.[2]

Somewhat more complex technique is the use of a parallel structure, as shown in the last sentence of example (2). Here, the repetition of the phrase *when he* is avoided in the parallel clause *prepared to place her under the shelter of her glass globe*.

This dissertation focuses on a computational perspective of one such technique, the use of *shell nouns*. *Shell nouns* are abstract nouns, such as *fact*, *issue*, *idea*, and *problem*, which,

[1] From: Ian Johnston (translator). *Beyond Good and Evil: Prelude to a Philosophy of the Future.* 1886, section 268. Thanks to Ryan Beaton for this quote.

[2] From: Katherine Woods (translator). *The Little Prince* by Antoine de Saint Exupéry.

among other functions, facilitate efficiency by avoiding repetition of even longer stretches of text. The *shell* metaphor comes from Schmid (2000), and it captures different functions of these nouns in a discourse: containment, signalling, pointing, and encapsulating. In example (3), the shell noun phrase *this fact* avoids repetition of the propositional clause *Living expenses are much lower in rural India than in New York*, and in example (4), the shell noun phrase *this issue* effectively avoids repetition of the verb phrase *allow some form of audio-visual coverage of court proceedings*.[3]

(3) **Living expenses are much lower in rural India than in New York**, but **this fact** is not fully captured if prices are converted with currency exchange rates.

(4) New York is one of only three states that do not **allow some form of audio-visual coverage of court proceedings**. Some lawmakers worry that cameras might compromise the rights of the litigants. But a 10-year experiment with courtroom cameras showed that televised access enhanced public understanding of the judicial system without harming the legal process. New York's backwardness on **this issue** hurts public confidence in the judiciary...

Similar to pronouns, shell nouns themselves are incomplete (Vendler, 1968) and unspecific (Francis, 1994). They can only be interpreted together with the *shell content*, i.e., the propositional content they encapsulate in the given context. In order for a computer to understand text containing shell nouns, the links between the shell content (e.g., *Living expenses are much lower in rural India than in New York*) and shell noun phrases (e.g., *this fact*) must be identified. The process of identifying the content of a shell noun phrase in the given context is referred to as *shell noun resolution* or *interpretation*. In the examples above, the shell noun phrases are shown in boldface and are underlined, and the shell content is shown in boldface.

Shell nouns share similarities with two phenomena in computational linguistics. First, the relation between shell nouns and shell content is similar to the relation of *anaphora*. *Anaphora* is the relation between an *anaphor* and an *antecedent*, where the interpretation of the anaphor is

[3]All examples in this dissertation are either from the New York Times corpus (`http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2008T19`) or Medline (`http://www.nlm.nih.gov/bsd/pmresources.html`), except where indicated.

determined via that of the antecedent (Huddleston and Pullum, 2002). In the above examples, the shell noun phrases act as anaphors and the shell content as their antecedents. Example (3) shows a backward-looking anaphora relation between the clause *Living expenses are much lower in rural India than in New York* and the shell noun phrase *this fact*. Similarly, example (4) shows a backward-looking anaphora relation between the shell noun phrase *this issue* and the idea *whether to allow some form of audio-visual coverage of court proceedings*. Example (5) is a bit different from the previous examples in that it is similar to a forward-looking case of anaphora, i.e., the shell content follows the shell noun phrase. We refer to such occurrences of shell nouns *cataphoric shell nouns* to contrast them with typical anaphoric occurrences of shell nouns (i.e., forward-looking shell content vs. backward-looking shell content).[4]

(5) **The issue** that this country and Congress must address is **how to provide optimal care for all without limiting access for the many**.

Second, unlike well-studied anaphoric expressions such as pronouns (e.g., *he* and *she*), shell noun phrases generally refer to complex and abstract objects. They are complex because they involve a number of entities and events and relationships between them, and are abstract because they do not represent purely physical entities. Asher (1993) calls this phenomenon, in which an anaphoric expression refers to an abstract object such as a proposition, a property, or a fact, *abstract anaphora*. In the examples above, the relation between the shell noun phrases and their shell content is similar to abstract anaphora.

Shell nouns play an important role in organizing a discourse and maintaining its coherence, and resolving them is an important component of various computational linguistics tasks that rely on discourse structure. Yet, their understanding from a computational linguistics perspective is only in the preliminary stage. There have been attempts at manually annotating the interpretation of anaphoric occurrences of shell nouns (Botley, 2006), and resolving expres-

[4]Calling the predication structures such as (5) cataphora stretches the traditional use of the term where the antecedent and the anaphor occur in different clauses (e.g., *If you want **them**, there are **cookies** in the kitchen.*). That said, some annotation schemes such as MUC (Grishman and Sundheim, 1996) consider predication structures to be anaphoric or cataphoric.

sions with similar kinds of abstract object interpretation, such as antecedents of personal and demonstrative pronouns (Eckert and Strube, 2000; Byron, 2003; Müller, 2008; Poesio and Artstein, 2008, inter alia). But all these approaches do not particularly focus on the phenomenon of shell nouns.

This thesis provides a computational treatment of shell nouns. Accordingly, I consider three primary questions: first, the extent to which speakers of English are able to interpret shell nouns; second, how an automated process can determine the interpretation of shell nouns; and third, the extent to which knowledge derived from the linguistics literature can help in this process.

1.1 Central thesis

The central thesis presented in this dissertation is *that knowledge and features derived from the linguistic literature can help in automatically resolving both anaphoric and cataphoric occurrences of shell nouns, at least for shell nouns with strict semantic and syntactic expectations*.

Linguists have studied a variety of shell nouns, their classification, different patterns they follow, and their semantic and syntactic properties in detail (Vendler, 1968; Winter, 1977; Ivanic, 1991; Asher, 1993; Francis, 1994; Schmid, 2000, inter alia). However this information has not been exploited for their automatic resolution. In fact, shell noun resolution has hardly received any attention in computational linguistics. Indeed, most of the available anaphora resolution systems (e.g., GuiTAR[5], BART[6], Reconcile[7], and that of Durrett and Klein (2013)) are designed to resolve reference relations only between pronouns and nouns, and between other noun phrases which are not pronouns (e.g., *the president of the United States of America* and *Barack Obama*). These systems will not attempt to resolve the shell noun phrases in examples (3), (4), and (5). The driving motivation of this thesis is to fill this gap and to expand

[5] http://dces.essex.ac.uk/research/nle/GuiTAR
[6] http://bart-anaphora.org/
[7] http://www.cs.utah.edu/nlp/reconcile/

Table 1.1: Frequently occurring shell nouns in the New York Times corpus.

Noun	*way*	*point*	*issue*	*problem*	*decision*	*fact*	*question*	*idea*
Frequency	706,322	260,896	222,903	222,311	221,458	206,770	168,408	165,667

the range of anaphoric expressions that an automatic anaphora resolution system can tackle.

1.2 Importance of shell nouns in computational linguistics

Ubiquity of shell nouns Unlike pronouns, shell nouns are open-class expressions. Schmid provides a list of 670 English nouns that tend to occur as shell nouns (given in Appendix A). Although individual shell nouns do not occur as frequently as pronouns in text, shell nouns as a group occur frequently in all kinds of text from newspaper articles to novels to scientific articles. In fact, many shell nouns are among the most frequently occurring nouns in English. Schmid (2000) observed that shell nouns such as *fact*, *idea*, *point*, and *problem* were among the one-hundred most frequently occurring nouns in a corpus of 225 million words of British English.

Shell nouns occur frequently in argumentative texts, such as academic discourse, newswire text, and political debates. The pervasiveness of shell nouns in academic discourse and their importance for English as a second language (ESL) learners have been noted in the literature (e.g., Francis (1988); Flowerdew (2003)). We observed about 25 million occurrences of different shell nouns in the New York Times corpus.[8] Table 1.1 shows some frequently occurring shell nouns with their frequencies in the New York Times corpus. In political debates, they are used to indicate personal attitudes and evaluations (Schmid, 2000; Botley, 2006). For instance, politicians are proficient in characterizing their ideas as *facts* and *advantages*, and their opponents' ideas as *issues* and *problems*.

[8]This number is just an initial indication of how frequently shell nouns occur, and it should be taken with caution. I am reporting the lexical occurrences of shell nouns, which might not be in fact shell noun usages. The problem of whether a particular usage is a shell noun usage or not is not straightforward.

Current challenges in anaphora resolution Anaphora resolution involves more than just relations between simple nouns and pronouns. There are many expressions in natural texts other than simple pronouns that cannot be interpreted by themselves. To fully understand the phenomenon of reference, all such expressions must be addressed. Researchers of anaphora resolution have identified the current challenge in the field as expanding the range of anaphoric expressions and going beyond nominal anaphora (Eckert and Strube, 2000; Modjeska, 2003; Byron, 2004; Poesio et al., 2011, inter alia). According to Byron (2004):

> it is clear that pronoun interpretation software must be able to understand both noun-phrase-coreferential as well as non noun-phrase-coreferential pronouns.

In Poesio et al.'s (2011, p. 85) words:

> One major challenge for the next decade will be to expand the range of anaphoric phenomena considered and accordingly to go beyond nominal anaphora — e.g., develop models able to deal with reference to abstract objects, bridging and ellipsis ...

Resolving shell nouns is a step towards tackling a class of expressions with abstract object interpretation that has so far not been addressed in a robust computational implementation.

Function of shell nouns in discourse Shell nouns are important in structuring a discourse efficiently and elegantly, and language without them will be chaotic and unwieldy. In Schmid (2000, p. 14)'s words:

> Discourse without shell nouns can be compared to an egg-and-spoon race using eggs without shells. One would not be able to get on in discourse (and in the race), if it were not for the encapsulating function of shell nouns (or egg shells). In other words, shell nouns can supply propositions with conceptual shells which allow speakers to grab them and carry them along as they move on in discourse.

Shell nouns play three important roles in organizing a discourse. First, they are used metadiscursively to talk about the current discourse. In example (4), the author *characterizes* and *labels* the information presented in the context by referring to it as an *issue* — an

important topic or problem for debate or discussion. Second, they are used as cohesive devices in a discourse. In (4), for example, *this issue* on the one hand summarizes one stage of an argument by referring to it as a *issue* and on the other, faces forward and serves as the starting point of the following argument. Finally, as Schmid (2000) points out, like the conjunctions *so* and *however*, anaphoric occurrences of shell nouns may function as topic boundary markers and topic change markers.

Potential applications A practical reason for studying shell nouns is their potential use in natural language understanding applications. Shell nouns are powerful linguistic tools and understanding their interpretation is essential to understanding virtually any substantial natural language text. The correct interpretation of shell nouns will help a number of natural language understanding tasks such as text summarization, information extraction, question answering, and discourse analysis. In example (4), for instance, knowing the interpretation of *this issue* suggests which discourse relations occur between the elementary discourse units (which are generally clauses) from the first sentence and the last sentence.

Moreover, resolving occurrences of the shell noun *issue* in a domain will spell out important unsolved problems from that domain. So extraction of this information would be useful in any information retrieval system or a summarization system.

Another application of shell nouns is in ESL learning. It has been observed that ESL students tend to make errors when they use shell nouns (Francis, 1988; Flowerdew, 2003, 2006; Hinkel, 2004). Pointing out the shell content of different shell nouns might help ESL learners in learning these complex abstract concepts. For instance, Francis (1988) suggests the following different tasks in order to teach shell nouns to ESL learners. First, the students were given short texts in which shell nouns were used effectively, and they were asked to identify the referents for the shell nouns. Second, students discussed the function of evaluative modifiers to the shell nouns. Third, students were asked what the effect would be if shell nouns were replaced by the demonstrative *this*. Fourth, shell nouns were deleted and students were asked

to select an appropriate one with or without alternatives provided. ESL learners could possibly get help with all these tasks from a computational system that is able to deal with shell nouns.

1.3 Challenges posed by shell nouns

Shell nouns demonstrate an interplay between different kinds of linguistic knowledge such as syntactic, semantic, and pragmatic knowledge. In this section, I describe the different challenges one has to deal with when interpreting shell nouns.

Semantic challenge The relation between a shell noun and its content is in many crucial respects a semantic phenomenon. Each shell noun has its idiosyncrasies. In particular, different shell nouns have different semantic and syntactic expectations, and the primary semantic challenge is developing a general shell noun resolution method that is able to identify and deal with these idiosyncrasies. Shell nouns take different types of one or more semantic arguments: one introducing the shell content and others expressing circumstantial information about the shell noun. For instance, *fact* typically takes a single *that* clause as an argument, whereas *reason* is relational and expects two semantic arguments: cause and effect, as shown in example (6). The cause argument is generally the shell content, and it acts as the ground or motivation for the effect argument.

(6) **The reason** [that they are together]$^{\text{effect}}$ is [**that they're not like each other**]$^{\text{cause}}$.

Similarly, *decision* takes an agent making the decision and the shell content is represented as an action or a proposition, as shown in (7).[9]

(7) I applaud loudly **the decision** of [Greenburgh]$^{\text{agent}}$ [**to ban animal performances**]$^{\text{action}}$.

[9]Observe that this aspect of shell nouns of taking different numbers and kinds of complement clauses is similar to verbs having different sub-categorization frames, except that in case of shell nouns, the propositional shell content is given in one of the semantic arguments.

In (8), the same propositional content has been referred to as a *fact* and a *reason*. The shell content of the shell noun *reason* is the ground for the clause *that you'd wind up suing*. The shell noun *fact*, on the other hand, does not display any relationship between clauses.

(8) **The fact that people were misled and information was denied**, that's **the reason** that you'd wind up suing.

Thus the semantic challenge of shell noun interpretation involves realizing the idiosyncrasies of different shell nouns, and identifying and recognizing whether a particular text segment represents the intended semantic concept, i.e., the concept of a *fact* or an *issue*. In particular, resolving a shell noun involves: a) identifying the expected semantic arguments for that noun, b) identifying which of these arguments represents the shell content, and c) extracting the constituent representing the desired argument in the given context.

Syntactic challenge Shell nouns pose a challenge in terms of possible syntactic shapes of the shell content. Shell content represents complex abstract entities. Typically, such entities cannot be expressed with simple noun phrases. Accordingly, in examples (3), (4), and (5) the shell content is expressed by a *that* clause, a verb phrase, and a *wh* clause respectively. This leads to a large shell content candidate search space and a number of spurious shell content candidates, i.e., candidates that are clearly not eligible candidates for the given shell noun as they do not satisfy the basic syntactic, semantic, and lexical constraints of that shell noun. Hobbs (1978) in his seminal paper about pronoun resolution summarizes this challenge when resolving anaphors with antecedents of different syntactic types.

> One might suggest that the algorithm be modified to accept an S node as the antecedent of a pronoun occurring in certain contexts. However, the problem of avoiding spurious antecedents would then be quite severe. In
>
> **(9)** The newspaper reported that Ford had claimed the economy was improving, but I didn't believe it.
>
> the algorithm allowing both S and NP nodes would recommend the following plausible antecedents, in the given order:

The newspaper reported that Ford had claimed the economy was improving

the newspaper

Ford claimed the economy was improving

the economy was improving

A short sentence given above has at least four antecedent candidates. So considering multiple surrounding sentences, which is usually required for shell noun resolution, just adds to the number of spurious antecedent candidates.

Pragmatic challenge Another challenge is in regard with pragmatics, i.e., the ways in which the context contributes to the meaning. Even if we resolve syntactic and semantic ambiguities correctly, i.e., if we have a text segment with appropriate syntactic type representing the given semantic concept plausibly, it is still not enough for accurate shell noun interpretation. Consider the constructed examples in (10). Here options a) and b) give two possible continuations of the preceding text. As we can see, the shell noun phrase *this fact* is common in both options, which enforces the same semantic constraints in both cases. Also, both options follow the same sentence structure. However, the shell noun phrases refer to different facts. Here we need to deal with pragmatics and make use of the context to correctly identify the shell content in both cases. In (10)a, the fact infuriated John so it is more likely that it refers to the act of the teacher and not his own act. Conversely, in (10)b, the fact infuriated the teacher and so it is more likely that it refers to John's act.

(10) 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) **This fact** 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) **This fact** infuriated the teacher, who had already told John several times to focus on class work.

In this dissertation we deal with semantic and syntactic challenges, leaving pragmatic challenges for future work.

1.4 Thesis organization

This thesis presents an end-to-end shell-noun resolution system. Chapter 2 describes the linguistic background and related work in terms of annotation and resolution of shell nouns and similar expressions. The chapter demonstrates a lack of attention to shell nouns from a computational linguistic perspective.

Chapter 3 examines the feasibility of annotating and resolving shell nouns. In particular, it focuses on annotating and resolving anaphoric occurrences of the frequently occurring shell noun *issue* in Medline abstracts. First, it describes our procedure to annotate the shell content of *this-issue* instances and measuring inter-annotator agreement. Then it explains our candidate-ranking model for *this-issue* resolution that explores various syntactic, semantic, and lexical features. Unlike previous approaches we do not restrict ourselves to nominal or verbal antecedents; rather, we are able to identify antecedents that are arbitrary spans of text. The inter-annotator agreement and evaluation results show the feasibility of reliably annotating and automatically resolving *this-issue* instances to their shell content, at least in the restricted domain of Medline abstracts. This chapter is based on the work presented in Kolhatkar and Hirst (2012).

The next step of an end-to-end shell noun resolution system is to generalize this approach to other shell nouns in a broader domain. Accordingly, the next two chapters describe approaches for resolving a variety of shell nouns occurring in two common constructions in the newswire domain: cataphoric and anaphoric constructions.

Chapter 4 describes a general shell noun resolution approach for shell nouns occurring in cataphoric constructions. The approach can resolve a variety of shell nouns in a broader newswire domain by exploiting lexico-syntactic knowledge and semantic classification of shell nouns derived from the linguistics literature. We evaluate the approach against crowdsourced data. This chapter is based on the work presented in Kolhatkar and Hirst (2014).

Chapter 5 describes a general approach to resolve shell nouns occurring in anaphoric constructions. First, it describes our approach for automatically creating labelled data for training

and interpreting such occurrences using supervised machine learning ranking models. Second, it describes our methodology for reliably annotating the shell content, the quality of crowd annotation using experts, and the challenges we faced in doing so. This chapter is based on the work presented in Kolhatkar et al. (2013a,b).

Finally, Chapter 6 summarizes the contributions of this dissertation and identifies potential directions for future work.

Chapter 2

Background

This chapter lays out the necessary background for the problem of resolving shell nouns to which the whole thesis is directed. In line with the theme of the thesis, I talk about related work in three different areas: the linguistic account of shell nouns, annotation, and resolution of expressions with similar kinds of abstract object interpretation.

Section 2.1 provides the linguistic account of shell nouns from the perspective of their automatic resolution. It starts with the definition of shell nouns, and then describes the linguistic framework on which this thesis is founded. It also discusses the similarity of shell nouns with abstract anaphora and deictic expressions.

Sections 2.2 and 2.3 describe the attempts at annotation and resolution of expressions with similar kinds of abstract interpretation. There have been efforts in annotating antecedents of demonstrative NPs and of anaphoric shell nouns. Also, a number of approaches have been suggested to resolve pronouns with abstract antecedents (e.g., *this*, *that*, *it*) in restricted domains such as TRAINS93 dialogues. In these sections we discuss these approaches in detail.

2.1 Linguistic account of shell nouns

2.1.1 Terminology and definition

Same phenomenon, different names Shell nouns have been a subject of interest for linguists and philosophers for decades. In the literature, they have been discussed in various contexts from various perspectives.

Vendler (1968) calls them *container nouns*. He defines them in terms of two patterns: *N* is *nominalization* and *nominalization* is *N* where *N* is a container noun and *nominalization* is a *that* clause, *to* clause, *wh* clause, or a deverbal noun (noun derived from a verb). Unlike other nouns, container nouns can take a verbal complement or a clause in the form of a nominalization as in (11).

(11) The real **issue** is *to get on with rebuilding society*.

Halliday and Hasan (1976) discuss a similar set of nouns, referred to as *general nouns*, in connection with lexical cohesion. Halliday and Hasan's class of general nouns contains nouns having generalized reference within the major noun classes, e.g., human nouns such as *people*, *man*, and *girl*; inanimate abstract nouns such as *affair* and *matter*; and fact nouns such as *question* and *idea*. According to Halliday and Hasan, when these nouns are combined with specific determiners, e.g., *the fact* or *the thing*, they act as anaphoric expressions and in such cases "the referent is not being taken up at face-value but is being transmuted into a fact or a thing".

Francis (1994) studies a set of nouns referred to as *label nouns* or anaphoric nouns as "they serve to encapsulate or package a stretch of discourse". Francis emphasizes the encapsulation and labelling aspect of these nouns. She requires two criteria for a noun to be a label noun. First, the noun is not a *repetition* or a *synonym* of any preceding element, and second the noun replaces text segments from the preceding text, naming them for the first time.

Ivanic (1991) refers to such nouns as *carrier* nouns, as these nouns carry a specific meaning within their context in addition to their constant dictionary meaning. Ivanic points out that

carrier nouns are good candidates for the core vocabulary of a language: they are not domain-dependent and occur in almost all domains with the same constant meaning. The specific or variable meaning is drawn from the given context.

Flowerdew (2003) focuses on a similar set of nouns in the context of ESL learning. He refers to such nouns as *signalling* nouns, due to their function of establishing links across and within clauses. He distinguishes three different usages of signalling nouns: meaning realized within the clause, meaning realized across clauses (i.e., anaphoric and cataphoric), and when there is nothing earlier or later in the text to realize the meaning of the signalling noun (i.e., exophoric).

Terminology used in this thesis Schmid (2000) uses the metaphorical term *shell noun*, which incorporates all the essential aspects of these nouns, such as containment, encapsulation, pointing, and signalling. According to Schmid, shell nouns are used by speakers to create conceptual shells for complex and elaborate chunks of information. He defines *shell-nounhood* as a functional notion; it is defined by the *use* of a particular abstract noun rather than the inherent properties of the noun itself. An abstract noun is a shell noun when the speaker decides to *use* it as a shell noun.[1]

An instance of a shell noun refers to a large chunk of information in the context, characterizing that information by encapsulating it as a temporary concept, e.g., by instantiating a *discourse entity* for it (Webber, 1991). We will talk about the notion of a discourse entity later in this chapter (Section 2.1.4).

If we consider full-content nouns, such as *cat* and *table*, on the one end of a spectrum and anaphoric pronouns, such as *he*, *she*, and *it*, on the other end of the spectrum, where do shell nouns lie? Schmid discusses this question in terms of three functions of shell nouns: characterization, concept-formation, and linking, as discussed below.

[1]From this functional perspective, it is not possible to provide an exhaustive list of shell nouns, and one might argue that the list given in Appendix A is misleading. There are many nouns, not included in the list, which can be used as shell nouns in certain contexts. That said, the purpose of the list is simply to illustrate typical nouns which are frequently used as shell nouns.

Characterization Full-content nouns have more or less stable and rich denotation and so they have strong potential for characterization of what speakers want to talk about. In contrast, anaphoric pronouns have limited potential for characterization. For instance, they can tell only about limited semantic dimensions such as gender, number, and spatial proximity (e.g., *this* and *that*). Shell nouns fall somewhere in the middle. As Schmid (2000, p. 16) explains, shell nouns characterize a piece of experience, such as a *fact* or a *problem*, which are quite stable notions, but they characterize an experience in a fairly general way. In other words, shell nouns represent abstract and unspecific meanings, which only become specific and detailed in context. In this respect they are similar to anaphoric pronouns, which are dependent on their context for their interpretation.

Concept-formation According to Schmid (2000, p. 18), concept formation captures two illusions: a) a word stands for a single, neatly-bounded entity, and b) this neatly-bounded entity has a thing-like quality. Full-content nouns have a relatively strong relation to the experience they encapsulate as a concept. In contrast, the experience that pronouns encapsulate is context-dependent. Shell nouns fall in between: similar to full-content nouns they indicate specific recurrent experience, to facts, reasons, and issues, and similar to pronouns, the concepts created by shell nouns are variable and context-dependent.

Linking Anaphoric pronouns have a great potential for linking. They instruct the reader to interpret two groups of linguistic elements together. In contrast, full-content nouns have hardly any potential to create cohesive links, except for lexical cohesion as described by Halliday and Hasan (1976). Shell nouns are more similar to anaphoric pronouns in this respect than to full-content nouns.

In sum, shell nouns combine the three functions of characterizing, concept-formation, and linking in a special way. These functions are otherwise performed separately by different linguistic elements. Moreover, while carrying out these functions, shell nouns try to maintain balance between conceptual stability and information flexibility.

Note that the sets of nouns discussed above, e.g., shell nouns, carrier nouns, and label nouns, overlap considerably. However, they are not exactly equivalent sets. For instance, the criteria for a noun to be a general noun in the sense of Halliday and Hasan are not exactly the same as the criteria for Schmid's shell noun. Schmid's shell nouns include Halliday and Hasan's fact nouns or inanimate abstract nouns. By contrast, human nouns such as *man* are general nouns in that they will be near the top of a concept hierarchy such as WordNet (Fellbaum, 1998). But they do not qualify to be shell nouns. Overall, there are three primary characteristic properties that are common in all different perspectives on shell noun: they encapsulate propositional content, often this content is a mental state or mental perspective, and often the content is communicated in the form of indirect speech.

This dissertation follows Schmid's terminology of shell nouns and primarily draws on his extensive analysis of shell nouns, in particular, his categorization of shell nouns in terms of the lexico-syntactic patterns they follow. As for the definition, I do not stick to a particular definition. I do not concentrate on the borderline nouns that satisfy one of the above definitions but not the others. Rather I focus on the set of nouns such as *issue*, *fact*, and *decision* that satisfy all of the definitions given above.

2.1.2 Linguistic properties of shell nouns and their content

2.1.2.1 Lexico-syntactic patterns

The primary aspect of shell nouns on which this dissertation is founded on is the lexico-syntactic patterns they follow. Precisely defining the notion of shell nounhood is tricky. There is no straightforward procedure to identify whether a particular usage of a potential shell noun is a shell noun usage or not. Schmid suggests a number of lexico-syntactic constructions to identify shell noun usages. In this section, we discuss these constructions briefly.

A necessary property of shell nouns is that they are capable of taking clausal arguments, primarily with two lexico-syntactic constructions: *Noun + postnominal clause* and *Noun + be + complement clause* (Vendler, 1968; Biber et al., 1999; Schmid, 2000; Huddleston and

Table 2.1: Lexico-grammatical patterns of shell nouns from Schmid (2000). Shell noun phrases are underlined, the pattern is marked in boldface, and the shell content is marked in italics.

	Cataphoric	
1	*N-be-to*	**Our plan is *to*** *hire and retain the best managers we can*.
2	*N-be-that*	**The major reason is *that*** *doctors are uncomfortable with uncertainty*.
3	*N-be-wh*	Of course, **the central, and probably insoluble, issue is *whether*** *animal testing is cruel*.
4	*N-to*	**The decision *to*** *disconnect the ventilator* came after doctors found no brain activity.
5	*N-that*	Mr. Shoval left open **the possibility *that*** *Israel would move into other West Bank cities*.
6	*N-wh*	If there ever is **any doubt *whether*** *a plant is a poppy or not*, break off a stem and squeeze it.
7	*N-of*	**The concept of** *having an outsider as Prime Minister* is outdated.
	Anaphoric	
8	*th-N*	*Living expenses are much lower in rural India than in New York*, but **this fact** is not fully captured if prices are converted with currency exchange rates.
9	*th-be-N*	*People change*. **This is a fact**.
10	*Sub-be-N*	If the money is available, however, ***cutting the sales tax* is a good idea**.

Pullum, 2002). In particular, Schmid provides a number of typical lexico-syntactic patterns that are indicative of shell noun occurrence. Table 2.1 shows these patterns with examples. Note that these are likely patterns for the shell noun usages, but this is not a comprehensive list of all possible patterns. The patterns are either *cataphoric*, where the shell content follows the shell noun, or *anaphoric*, where the shell content precedes the shell noun. I use the terms *cataphoric* and *anaphoric* for lack of a better alternatives. The motivation to use these terms is the similarity between such constructions and pronouns with forward- or backward-looking antecedents.

Cataphoric We refer to the patterns following the predication structure shown in Table 2.1 as cataphoric patterns because they suggest that the shell content follows the shell noun phrase. These patterns primarily follow two constructions: N-be-clause and N-clause.

N-be-clause In this construction, the shell noun phrase occurs as the subject in a subject-verb-clause construction, with the linking verb *be*, and the shell content embedded as a *wh* clause, *that* clause, or *to*-infinitive clause. The linking verb *be* indicates the semantic identity between the shell noun and its content in the given context. The construction follows the patterns in rows 1, 2, and 3 of Table 2.1.

N-clause This construction includes the cataphoric patterns 4–7 in Table 2.1. For these patterns the link between the shell noun and the content is much less straightforward: whether the postnominal clause expresses the shell content or not is dependent on the shell noun and the syntactic structure of the specific example. For the shell noun *fact*, typically the shell content is embedded in the postnominal *that* clause, if it is not a relative clause. In (12), the postnominal *that* clause represents the shell content, as it is not a relative clause: the fact in question is not an argument of *exploit and repackage*. On the other hand, the shell noun *reason* typically occurs with two complement clauses as arguments expressing cause (or ground) and effect (or consequence), with the shell content expressed in the cause argument. In example (13), the postnominal *that* clause is a complement clause, and still it is not the shell content because it is not the cause argument of the shell noun *reason*.

(12) **The fact that a major label hadn't been at liberty to exploit and repackage the material on CD** meant that prices on the vintage LP market were soaring.

(13) **One reason** that 60 percent of New York City public-school children read below grade level is **that many elementary schools don't have libraries**.

The *N-of* pattern is different from other patterns: it follows the construction *N-prepositional phrase* rather than *N-clause*, and since a prepositional phrase can take different kinds of embedded constituents such as a noun phrase, a sentential complement, and a verb phrase, the pattern offers flexibility in the syntactic type of the shell content.

Anaphoric For these patterns, the link between the shell noun and the content is created using linguistic elements such as *the*, *this*, *that*, *other*, *same*, and *such*. For the patterns 8 and 9

Table 2.2: Distribution of cataphoric patterns for six shell nouns in the New York Times corpus. Each column shows the percentage of instances following that pattern. The last column shows the total number of cataphoric instances of each noun in the corpus.

	Proportion							
Noun	***N-be-to***	***N-be-that***	***N-be-wh***	***N-to***	***N-that***	***N-wh***	***N-of***	**total**
idea	7	2	-	5	23	10	53	91,277
issue	-	1	5	7	14	2	71	55,088
concept	1	-	-	6	12	-	79	14,301
decision	-	-	-	80	12	1	5	55,088
plan	5	-	-	72	17	-	4	67,344
policy	4	1	-	16	25	2	51	24,025

Table 2.3: Distribution of anaphoric and cataphoric patterns for six shell nouns in the New York Times corpus. Each column shows the percentage of instances following that pattern. The last column shows the total number of instances of each noun in the corpus.

	Proportion					
Noun	***th-N***	***Sub-be-N***	***th-be-N***	***cataphoric***	***unknown***	**total**
idea	40	1	3	42	15	219,797
issue	36	2	2	23	38	239,189
concept	35	1	2	34	28	42,453
decision	28	-	1	27	44	231,971
plan	23	-	-	24	52	275,054
policy	9	-	-	13	77	190,374

the shell content does not typically occur in the sentence containing the shell noun phrase. For the pattern 10, the shell content is the subject in a subject-verb-N construction.

Pattern preferences Table 2.3 shows distribution of anaphoric patterns, proportion of anaphoric patterns in comparison with the cataphoric patterns, and proportion of instances with unknown patterns. Among anaphoric patterns, the pattern *th_N* is the dominant one. The table shows that to fully cover the phenomenon of shell nouns from a computational perspective, it is important to resolve shell nouns following both kinds of constructions: anaphoric and cataphoric. Moreover, a significant portion of instances of shell nouns fall under the *unknown* category. For these instances, it is not straightforward to identify whether the occurrences are anaphoric, cataphoric, or even non-shell noun usages. Schmid's patterns are lecixo-syntactic:

they are constrained with lexical items, such as *that* and *to*, and syntactic structure of a complement clause. So they are not able to capture examples such as (14) due to the absence of the lexical part of the patterns.

(14) My **idea** is: **They're going to 25 next year, anyway, so why not go to 25 right now"?**

That said, there are some example where the shell content is indefinite, as shown in (15).

(15) A bad **idea** does no harm until someone acts upon it.

Similarly, some examples demonstrate a non-shell usage of typical shell nouns, as in (16).

(16) Mathis is the cover subject of **this week's issue** of Sports Illustrated.

2.1.2.2 Cognitive status of shell content

Another aspect of shell nouns is the cognitive status of their shell content. Although this dissertation does not rely on this aspect of shell nouns, it is worth discussing the work done by Gundel et al. (1993) in this area, as research presented in this dissertation contrasts with their ideas. According to Gundel et al., different determiners and pronominal forms conventionally signal different cognitive statuses of the referents, described by the *givenness hierarchy*, shown in Figure 2.1. Each status entails all lower statuses. Also, if a referent has a particular status, the expression can take a form under it or any from the lower statuses.

Shell nouns following anaphoric constructions are a subset of expressions following the patterns *this N* and *the N*, and shell nouns following cataphoric constructions are a subset of expressions following the patterns *the N* and *a N*. The hierarchy suggests that *this*, *that*, and *this N* are associated with referents that are *activated*. According to Gundel et al., these referent are represented in current short-term memory and they include entities from the immediate discourse context. An example given by Gundel et al. is shown in (17).

(17) I couldn't sleep last night. **That** kept me awake.

Poesio and Modjeska (2002) test the hypothesis that THIS-NPs are activated but not in focus on texts from two domains in the GNOME corpus: the museum texts that contain de-

in focus	> activated	> familiar	> uniquely identifiable	> referential	> type identifiable
{*it*}	{*this*, *that*, *this* N}	{*that* N}	{*the* N}	{indefinite *this* N}	{*a* N}

Figure 2.1: Givenness hierarchy (Gundel et al., 1993)

scriptions of museum objects and the artists that produced them, and pharmaceutical texts from leaflets providing patients with mandatory information about their medicine. The texts contained about 500 sentences and 900 finite clauses. Poesio and Modjeska analyzed 112 THIS-NPs from these texts. To carry out this analysis, first, they operationalized the terms *activated* and *in focus* on the basis of *centering theory* (Grosz et al., 1995).

Centering theory assumes that each *utterance* introduces new discourse entities in the discourse and in this process updates the *local focus*. Moreover, the discourse entities introduced by an *utterance* are *ranked* and the most highly-*ranked* entity is referred to as the *preferred center (CP)*. The *backward-looking-center (CB)* of an *utterance* U_i is the highest *ranked* element of the *utterance* U_{i-1} that is *realized* in U_i. The theory itself keeps the notions of *local focus*, *ranking*, *utterance*, and *realized* open and researchers define these notions according to what they need. An *utterance* is generally defined as a sentence or a clause. *Ranking* is generally based on the grammatical function, e.g., subject is ranked higher than object, or on information status, e.g., hearer-old entities are ranked more highly than hearer-new entities. An entity could be *realized* implicitly or explicitly.

The term *activated* means that the entity is in the *global focus* (Grosz and Sidner, 1986) and is sufficiently salient. According to Grosz and Sinder, global focus characterizes the entire set of entities which are in some sense part of the attentional state of the participants of the discourse. It has been argued that two structures are required to characterize the entire set of entities in the global focus: a stack-like structure for the linguistic component of the global focus that contains every discourse entity introduced and the other situation-based structure that contains every entity from the visual scene.

Based on this idea of global focus, Poesio and Modjeska call an entity active if it is

1. in the visual situation; or
2. a forward looking center of the previous *utterance*; or
3. a part of the implicit linguistic focus. An entity is considered as part of the implicit linguistic focus if it can be *constructed* out of the previous *utterance*. An entity can be constructed out of an *utterance* if:
   (a) it is a plural object (e.g., John and Mary) whose elements or subsets have been explicitly mentioned in that *utterance*; or
   (b) it is an abstract entity, e.g., propositions, introduced by that *utterance*.

Poesio and Modjeska interpret *not in focus* as not CB(U_{i-1}), i.e., not in the backward-looking-center of the previous *utterance*. Accordingly, they verified Gundel et al.'s idea of THIS-NPs not being *in focus* in three different ways on 112 THIS-NPs instances.

1. About $90-93\%$ of THIS-NP instances referred to entities other than CB(U_i), the backward-looking center of the *utterance* containing THIS-NPs.
2. About $75-80\%$ of THIS-NP instances referred to entities other than CB(U_{i-1}), the backward-looking center of the previous *utterance*.
3. About $61-65\%$ of THIS-NP instances referred to entities other than CP(U_{i-1}), the most highly-*ranked* entity of the previous *utterance*.

Poesio and Modjeska conclude that *this*-NPs are used to refer to entities which are active in the sense explained above albeit not in focus, i.e., they are not the center of the previous utterance. Since NP referents are more easily rendered in focus than other referents (Müller, 2008), the antecedents of THIS-NPs are generally non-nominal.

2.1.2.3 Distributional properties of shell nouns

Distributional properties of shell nouns have been studied in the context of different genres (Botley, 2006; Castro, 2013).

Botley refers to the linguistic phenomenon where the antecedent is not directly recoverable and requires a complex process of inference as *indirect anaphora* (IA). He studied indirect

anaphora in a corpus of about 300,000 words from three different genres: newswire text, parliamentary debates, and literary texts. He annotated about 462 demonstrative NP instances from these genres. The work is not intended to give insights about how humans might resolve such anaphoric expressions. Accordingly, the corpus was annotated by only one annotator, without any reliability measurement and the data itself is no longer available (S. Botley, p.c.). Botley reports several observations about the distribution of IA in three different genres.

- A majority of IA instances occurred with the demonstrative *this*.
- The cases of cataphora are much rarer than the cases of anaphora.
- Hansard has more cases of IA than other two genres.
- Proximal demonstratives (*this* and *these*) have indirectly recoverable antecedents more frequently than the distant ones.
- Singular demonstratives have indirectly recoverable antecedents more often than plural ones.

Botley also makes a larger point that IA poses a difficulties for corpus-based linguistics in that about 30% of the cases of IA were hard to analyze. So in case of IA, it is hard to examine whether a corpus-based study of language is able to provide observations which either confirm or deny rationalistic intuitions about language. He also points out two main reasons for difficulty in analyzing cases of IA: lack of clear surface linguistic boundaries and complex or unclear inference process for retrieving the antecedent.

2.1.2.4 Other characterizing properties

Ivanic (1991) points out a number of properties associated with shell nouns. First, shell nouns fall midway on the continuum between open-class nouns and closed-class pronouns as they share properties with both. On the one hand they can take full range of determiners, quantifiers, and modifiers similar to open-class nouns and are more informative signposts than pronouns, and on the other they resemble pronouns in the sense that they have both a constant meaning

and a variable meaning. Second, shell nouns are *countable* abstract nouns. For example, *issue* is a valid shell noun but *contempt* and *justice* are not. Third, when shell nouns are used anaphorically, they almost always are associated with definite reference items such as the demonstratives *this*, *that*, plural forms *these* and *those*, and definite article *the*.

2.1.3 Categorization of shell nouns

In the literature, shell nouns have been categorized based on various properties. Asher (1993) categorizes them based on the levels of abstractness their shell content demonstrate. For instance, he distinguishes between eventualities and factualities, i.e., facts and propositions. Lyons (1977) refers to simple objects as *first-order entities*, events as *second-order entities*, and facts and propositions as *third-order entities*. Francis categorizes them based on the type of their linguistic act. Schmid groups together shell nouns with similar semantic properties. Below I discuss the classification of shell nouns by Francis (1994) and Schmid (2000).

2.1.3.1 Francis's categorization

Francis isolates a set of shell nouns and calls them *metalinguistic* nouns because they label a stretch of discourse as *a linguistic act*. Metalinguistic nouns are further divided into four subgroups: *illocutionary*, *language activity*, *mental process*, and *text*. *Illocutionary nouns* are nominalizations of verbal processes, usually acts of communication (e.g., *claim*, *remark*). They typically have cognate illocutionary verbs. *Language activity* nouns describe language activities or the results thereof (e.g., *debate*, *controversy*). They themselves do not communicate the meaning, but they are the description of the product of the language activity. *Mental process* nouns are cognition nouns and are nominalizations of cognition verbs (e.g., *belief*). *Text nouns* refer to formal textual structure of the discourse (e.g., *page*, *section*). Shell nouns that are not metalinguistic fall under the *ownerless* category. This category includes nouns like *problem* and *issue*. According to Francis, metalinguistic nouns such as *argument*, *point*, or *statement* are used to label a stretch of discourse as a linguistic act. In contrast, ownerless nouns *issue*

and *problem*, for example, exist outside the discourse. In other words, although the textual description of the concepts like *issues* and *problems* might be present in the current discourse, their actual existence is outside the discourse. On the other hand, metalinguistic nouns such as *argument* and *statement* come into existence in the current discourse. The classification is shown in Table 2.4.

Although this classification makes sense at a higher level, for the purposes of this thesis, it is more constructive to think of this classification as a spectrum rather than hard categories. For instance, Francis puts the bulk of shell nouns in the metalinguistic class. At the one extreme we have strictly metalinguistic shell nouns such as *statement*, which have their full existence in the given text. That is, *statement* is usually understood to refer to a specific sequence of words. But many shell nouns fall somewhere in the middle on this spectrum. For example, according to Francis, *argument* is a metalinguistic noun, whereas *issue* is not, as the mere existence of an argument is in the text, whereas issues lie outside of the text. But one can argue that the shell noun *argument* generally refers to a conceptual entity that can be formulated in a number of different ways. So it is not entirely the *words* presented in the given text that make an argument: there could be other sets of words or choices of linguistic constructs that would express the same argument. Conversely, although an *issue* typically refers to facts and events outside the text, it is possible that it must be formulated in words before it can be recognized as an *issue*.

Moreover, sometimes it is hard to distinguish between different categories of shell nouns. For instance, the mental shell nouns have a lot in common with linguistic shell nouns and sometimes it is hard to distinguish between the two usages. The former are used to report ideas whereas the latter are used to report utterances. Recall that *statement* is a good example of a linguistic noun. When speakers utter statements, they have the particular thought or idea expressed by the statement. When we say for instance *this statement is false*, we generally mean the idea expressed is false, not merely that the wording is wrong. So even an apparently a clear-cut case of a linguistic noun can be used as a mental noun.

Table 2.4: Categorization of shell nouns

Work	Class	Examples
Francis (1994)	**metalinguistic**	
	illocutionary	*accusation*, *claim*, *decision*, *proposition*, *appeal*, *explanation*, *reply*
	language activity	*contrast*, *debate*, *definition*, *example*, *proof*, *reasoning*
	mental process	*analysis*, *belief*, *concept*, *hypothesis*, *idea*, *view*
	text	*excerpt*, *page*, *paragraph*, *passage*, *quotation*, *section*
	ownerless	*problem*, *issue*, *context*, *fact*, *aspect*, *approach*
Schmid (2000)	factual	*fact*, *thing*, *point*, *problem*, *reason*, *difference*
	linguistic	*news*, *message*, *report*, *order*, *proposal*, *question*
	mental	*idea*, *notion*, *belief*, *plan*, *aim*, *decision*
	modal	*possibility*, *truth*, *permission*, *obligation*, *need*, *ability*
	eventive	*act*, *move*, *measure*, *reaction*, *attempt*, *tradition*
	circumstantial	*situation*, *context*, *area*, *time*, *way*, *approach*

2.1.3.2 Schmid's categorization

Schmid takes a more systematic approach than Francis. He classifies shell nouns at three levels. At the most abstract level, he classifies shell nouns into six semantic classes: *factual*, *linguistic*, *mental*, *modal*, *eventive*, and *circumstantial*, as shown in Table 2.4. Each semantic class indicates the type of experience the shell noun is intended to describe. For instance the *mental* class describes ideas, cognitive states, and processes, whereas the *linguistic* class describes utterances, linguistic acts, and products thereof.

The next level of classification includes more-detailed semantic features. Each broad class from the abstract level categorization is sub-categorized into a number of *groups*. A group of an abstract class tries to capture the semantic features associated with the fine-grained differences between different usages of shell nouns in that class. For instance, groups associated with the *mental* class are: *conceptual*, *creditive*, *dubiative*, *volitional*, and *emotive*.

The third level of classification consists of *families*. A family groups together shell nouns with similar semantic features. Schmid provides 79 distinct families of 670 shell nouns. Each family is named after the primary noun in that family. Table 2.5 shows six families: *Idea*, *Plan*, *Trouble*, *Problem*, *Thing*, and *Reason*. A shell noun can be a member of multiple families. The nouns subsumed in a family share semantic features, which come from the first two levels of categorization: classes and groups. For instance, all nouns in the *Idea* family are *mental* and

Table 2.5: Example families of shell nouns from Schmid (2000).

***Idea* family**	***Plan* family**
Semantic features: [mental], [conceptual]	**Semantic features**: [mental], [volitional], [manner]
Frame: mental; focus on propositional content of IDEA	**Frame**: mental; focus on IDEA
Nouns: *idea*, *issue*, *concept*, *point*, *notion*, *theory*, ...	**Nouns**: *decision*, *plan*, *policy*, *idea*, *strategy*, *principle*, *rationale*, ...
Patterns: *N-be-that/of*, *N-that/of*	**Patterns**: *N-be-to/that*, *N-to/that*
***Trouble* family**	***Problem* family**
Semantic features: [eventive], [attitudinal], [manner], [deontic]	**Semantic features**: [factual], [attitudinal], [impeding]
Frame: general eventive	**Frame**: general factual
Nouns: *problem*, *trouble*, *difficulty*, *dilemma*, *snag*	**Nouns**: *problem*, *trouble*, *difficulty*, *point*, ...
Patterns: *N-be-to*	**Patterns**: *N-be-that/of*
***Thing* family**	***Reason* family**
Semantic features: [factual]	**Semantic features**: [factual], [causal]
Frame: general factual	**Frame**: causal; attentional focus on CAUSE
Nouns: *fact*, *phenomenon*, *point*, *case*, *thing*, ...	**Nouns**: *reason*, *cause*, *ground*, *thing*
Patterns: *N-that*, *N-be-that*	**Patterns**: *N-be-that/why*, *N-that/why*

conceptual. They are mental because ideas are only accessible through thoughts, and conceptual because they represent reflection or an application of a concept. Each family activates a *semantic frame*. The idea of these semantic frames is similar to that of frames in Frame semantics Fillmore (1985) and in semantics of grammar Talmy (2000). In particular, Schmid follows Talmy's conception of frames. A semantic frame describes conceptual structures, its elements, and their interrelationships. For instance, the *Reason* family invokes the causal frame, which has cause and effect as its elements with the attentional focus on the cause. According to Schmid, the nouns in a family also share a number of lexico-syntactic features. The *patterns* attribute in Table 2.5 shows prototypical lexico-syntactic patterns, which *attract* the members of the family. Schmid defines *attraction* as the degree to which a lexico-grammatical pattern attracts a certain noun. For instance, the patterns *N-to* and *N-that* attract the shell nouns in the *Plan* family, whereas the *N-that* pattern attracts the nouns in the *Thing* family. The pattern *N-of*

is restricted to a smaller group of nouns such as *concept*, *problem*, and *issue*.[2,3]

2.1.4 Relation to deictic expressions

Shell noun phrases overlap with anaphoric expressions as well as deictic expressions. In this section, we discuss where exactly they lie on the spectrum of anaphoric and deictic expressions.

The interpretation of anaphoric expressions depends upon the linguistic context, i.e., the surrounding text (Poesio et al., 2011). Pronouns are great examples of anaphoric expressions as they strongly depend on the textual context.[4] There are other expressions whose reference is determined in relation to the features of the utterance-act: the time, the place, and the participants (Huddleston and Pullum, 2002). An example from Huddleston and Pullum is shown below.

(18) Could **you** pick **this** up and put it with **those boxes**, please?

Here, *this* refers to something that is close to the speaker and *those boxes* refers to the boxes that are further away. These expressions cannot be interpreted simply based on the linguistic context. The correct interpretation also requires visual context — the overall utterance situation around the speaker. Such expressions are referred to as *deictic* expressions.

If we consider a spectrum with anaphoric expressions on one end and deictic expressions on the other, where do shell nouns lie? An example of anaphoric occurrence of *this issue* is given in (19). Here the antecedent of *this issue* is clearly given in the linguistic context.

(19) There is a controversial debate **whether back school program might improve quality of life in back pain patients**. This study aimed to address **this issue**.

Most of the shell noun occurrences of *this issue* in the NYT are similar to example (19). That said, it is always possible to construct examples that fall on the other end of the spectrum.

[2]Schmid used the British section of COBUILD'S *Bank of English* for his classification.

[3]Schmid's families could help enrich resources such as FrameNet (Baker et al., 1998) with the shell content information.

[4]Some mentions of the pronoun *it* are non-anaphoric and are *pleonastic* (Lappin and Leass, 1994), i.e., they are used just for the sake of satisfying grammar as in *It is raining*.

Imagine a town holding a referendum on whether to allow mining operations to begin near a residential area. There are pro and con banners up everywhere and the issue is extremely contentious. A resident might point to all the banners, saying *this issue is tearing down the town*. This is an example of a deictic occurrence of *this issue*, as resolution of *this issue* requires visual context.

The following example will fall on the midpoint of this spectrum. Here, the textual antecedent of *this problem* is *garage space*, however, the actual problem is *lack of garage space*, which is assumed to be understood by the reader.

(20) On a recent Friday, Mr. Ferraro of Avis stood in a steamy garage and described the problem of keeping up with weekend demand. In car rental parlance this is called fleet management, and it is a nightmare in Manhattan, where the primary problem is **garage space**.

"We can hold 40 or 50 cars," said Mr. Ferraro, who, like his counterparts at other companies, was deliberately unspecific to avoid tipping off the competition. "But we are renting hundreds today."

Avis and other big rental car companies solve **this problem** by paying 30 to 50 drivers to shuttle autos in from their airport and suburban locations, which is cheaper than renting more parking space.

For the anaphoric occurrences of shell nouns, as in (19), it is possible, at least theoretically, for a computer program to resolve them, as the complete shell content occurs in the linguistic context. On the other hand, with access to the textual information only, it is not possible to resolve deictic examples. For examples such as (20), a computer program can provide an interpretation, which is available in the text. Although this will not be completely satisfactory, as it will not exactly be a full interpretation, such interpretation might still be useful in practical applications.

Lyons (1977) introduces the term *textual deixis*, when the referring expression are linked to the spatio-temporal co-ordinates of the act of utterance. He distinguishes between *pure textual deixis*, where the referring expression is related to a textual unit, and *impure textual deixis*, where the referring expression is related not exactly to the textual unit, i.e., the words in the

text, but to the underlying interpretation. For instance, in example (21) from Lyons (1977), *that* refers not exactly to the words mentioned here, but to the underlying fact.

(21) A: You look about fifteen.
B: Is *that* meant to be a compliment?

Webber (1988) and Asher (1993) use the term *discourse deictic* expressions for Lyons's impure textual deixis, as very often the discourse segments they refer to have their own mental reality, distinct from the individual entities described in that discourse segment. The term *discourse deictic expression* originates from the concept of a *discourse model* (Webber, 1979). A discourse model contains representation of entities that have been referred to in the discourse, the attributes of these entities, and relationships between them. According to Prince (1981) discourse entities in a discourse model are represented by NPs:

> Let us say that a text is a set of instructions from a speaker to a hearer on how to construct a particular discourse model. The model will contain discourse entities, attributes, and links between entities. A discourse entity is a discourse-model object, akin to Karttunen (1976)'s discourse referent; it may represent an individual (existent in the real world or not), a class of individuals, an exemplar, a substance, a concept, etc. Following Webber (1979) entities may be thought of as hooks on which to hang attributes. All discourse entities in a discourse model are represented by NPs in a text, though not all NPs in a text represent discourse entities.

Webber (1988) takes into account discourse deictic expressions, such as *this*, *that*, and *it*, which have non-nominal referents. She leaves it open whether referents of such expressions should be considered discourse entities or not (Webber, 1988, p. 119):

> I have not said anything about whether or not these discourse segment referents$_m$ should be considered discourse entities like their NP-evoked counterparts. This is because I do not believe there is enough evidence to warrant taking a stand. Part of the problem is that there is no precise criterion for "discourse entity-hood".

If we want to build a computational system to interpret shell noun phrases based on the notion of a discourse model, we need to build two components: a) a method for constructing a discourse model that contains referents of shell noun phrases, which are typically non-nominal

abstract objects, and b) a method to map these referents to shell noun phrases. Implementation of these components raises several questions. First, while constructing a discourse model, what discourse entities do we include in the discourse model? If we include only NPs, the model won't account for the phenomenon of shell nouns. If we want to include referents of discourse deictic referents, the question is how do we construct those, as such referents are created on the spot by knowing the context of the expression, as we will see in Section 2.1.5.1. Second, how do we represent abstract referents, such as events, propositions, and situations? Third, what type of objects are the referents of specific shell noun phrases? These are all interesting but poorly understood questions. In this dissertation, I do not commit to any specific model or theory. Rather I focus on two more basic questions: a) to what extent humans agree on the text representing shell content of shell noun phrases, and b) to what extent a computational system can identify such text in the given context. I believe these questions come before developing a reasonable theory for the reference of such kind of expressions.

Note that Huddleston and Pullum (2002) consider the example such as (19), where the antecedent is present in the given text as *anaphoric*, but Lyons will consider it as pure textual deixis. In this dissertation, I do not get into the philosophical discussion of what is deictic and what is anaphoric, or how abstract referents might be constructed in the readers discourse model. For general text understanding, it will be useful if a computer program is able to identify a text segment that provides full or even partial interpretation for the actual referent of the given shell noun. Accordingly, in this dissertation, I stick to the term *anaphora* for the shell nouns that occur with demonstratives.

2.1.5 Relation to abstract anaphora

Recall that the relation between shell nouns and their content is similar to Asher (1993)'s abstract anaphora, in which the anaphoric expressions refer to an abstract object such as a proposition, a property, or a fact. According to Asher, abstract objects have three properties: they

have no spatio-temporal location, usually no causal effect, and are not perceived by senses.[5]

Although the properties of abstract anaphora and their antecedents from the literature do not specifically describe properties of the shell content of shell nouns, there is an overlap between them, as both represent abstract objects. Below I discuss a number of characteristic properties of abstract anaphora that are relevant for shell nouns.

2.1.5.1 Referent coercion

It has been suggested that referents of discourse deictic anaphora involve *referent coercion* (Dahl and Hellman, 1995) or *ostension* (Webber, 1991). The idea is that for such anaphors the referent does not exist in the discourse model on its own and the anaphor itself creates the referent. An example from Eckert and Strube (2000) is shown in (23).

(23) John crashed the car.

1. **This** annoyed his parents. (event)
2. Jane did **that** too. (concept)
3. **This** shows how careless he is. (fact)
4. His girlfriend couldn't believe **it**. (proposition)

In this example, the same referent is conceptualized as an event, a concept, a fact, and a proposition. Similarly, the shell content of shell noun phrases do not exist in the discourse model before they are conceptualized using shell nouns. But shell noun phrases differ from examples such as (23). In example (23), the semantic type of the referent has to be identified from the context of the anaphor, i.e., using the predicative context, whereas for shell noun phrases, the shell nouns themselves provide the semantic type of the referent. That said, the problem

[5] By *no causal effect*, Asher likely means *no purely physical causal effect*. For instance, in example (22), *this decision* is an abstract anaphor that also has causal effect, e.g., on the behaviour of the agents making the decision or affected by the decision. However, such causal effect is mediated through the mental states of the agents and in this sense is not purely physical.

(22) In principle, he said, airlines should be allowed to sell standing-room-only tickets for adults — as long as **this decision** was approved by their marketing departments.

of identifying the actual shell content is still hard as there is no one-to-one correspondence between the syntactic type of shell content and semantic type of its referent. 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.

2.1.5.2 Right-frontier constraint

Polanyi (1985), Webber (1991), and Asher (1993) provide theoretical accounts for anaphors with similar properties, i.e., pronouns with abstract antecedents. They suggest that abstract antecedents can be recovered using a discourse model which represents the discourse entities that have been referred to in it. Each discourse entity is associated with a set of attributes and its relationship to other discourse entities. Both Asher and Webber suggest that such abstract antecedents must be linearly or hierarchically adjacent to their anaphors. Webber argues that only those discourse segments can yield referents for abstract anaphors that correspond to nodes on the *right frontier* of a formal discourse tree structure, where the right frontier of a tree is the nodes along the path from root to tip defined by the sequence of rightmost daughters, starting at the root. An example from Webber (1991) is given in (24).

(24) (a) There's two houses you might be interested in.

(b) House A is in Palo Alto. It's got 3 bedrooms and 2 baths, and was built in 1950. It's on a quarter acre, with a lovely garden, and the owner is asking $425K. But **that**'s all I know about it.

(c) House B is in Portola Valley. It's got 3 bedrooms, 4 baths and a kidney-shaped pool, and was also built in 1950. It's on 4 acres of steep wooded slope, with a view of the mountains. The owner is asking $600K. I heard all **this** from a real-estate friend of mine.

(d) Is **that** enough information for you to decide which to look at?

(e) *But **that**'s all I know about House A.

Here, parts (b) and (c) are central parts of the text. According to Webber, the continuation (e) is ill-formed, as at this point the information of House A is closed off and it is not possible to

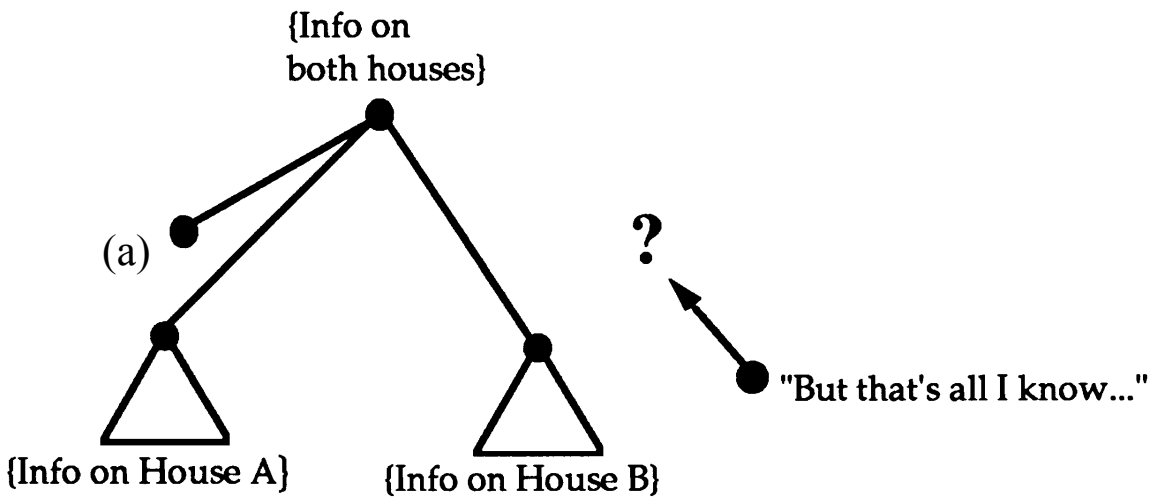


Figure 2.2: Right-frontier rule (Webber, 1991)

access it, as shown in Figure 2.2. The only accessible antecedents at this point are the ones on the right frontier: 1) information on both houses, i.e., the information spanned by the root node and 2) the information on House B.

Asher (1993)'s principle of *availability* states that only the current constituent itself, its discourse referents, sub-constituents, and a constituent which stands in a discourse relation to the current constituent are available as antecedents for abstract anaphora.

Although, the right-frontier rule and the principle of availability seem to work for example (24), it is unclear how strict these rules are, as noted by Poesio et al. (2005). For instance, the substitution

(e′) *That*'s all I know about House A, but I can give you more information about House B if you are interested.

in the place of (e) (or of (d)) will violate the right-frontier rule because *That* in (e′) accesses the closed-off information about the House A. However it seems a fairly natural continuation of the conversation, certainly in the context of a spontaneous conversation that has not been prepared in detail in advance. (Indeed, even if the conversation were structured in advance, it may be perfectly reasonable to give basic information on both House A and House B, before then qualifying what you know of House A in the manner of (e′).) In any case, whatever one

thinks as to whether the continuation (e′) is ideal, it would seem a fairly natural occurrence in spoken language.

Leaving aside this general question about the strictness of the right-frontier rule or the principle of availability, there is a more significant concern specific to shell nouns. This concern is that anaphoric shell nouns such as *this issue* and *this fact* can have shell content that can take several syntactic shapes. So the discourse trees derived from the state-of-the-art discourse parsers such as those of Joty et al. (2013) and Feng and Hirst (2014), which are grounded on clauses as their elementary discourse units, will not always correspond smoothly with the syntactic shape or size of typical shell content. That is, typical shell content may include other syntactic shapes such as noun phrases and verb phrases which would not be accessible in such discourse trees.

Moreover, in the context of anaphoric shell nouns, the state-of-the-art discourse parsers do not provide the perfect discourse representation that is needed to tackle anaphoric shell nouns. For instance, running the state-of-the-art discourse parser by Feng and Hirst on the anaphoric shell noun examples disregards the anaphoric relation between anaphoric shell noun phrases and their shell content. Figure 2.3 shows the discourse representation of the example we saw in Chapter 1 given by Feng and Hirst. The parser first splits the discourse into *elementary discourse units* (EDUs). For readability, I have numbered each EDU in the given discourse and used these numbered EDUs in the discourse tree representation. As we can see in the figure, there is no direct relation between EDU_8 and EDU_2, and the right frontier in the discourse tree does not give the correct antecedent.

2.1.5.3 Syntactic preferences

Asher (1993, p. 226) notes that the range of syntactic constructs of abstract antecedents is quite wide: the antecedents arise from six different linguistic constructions.

1. *that* clause (e.g., *that Mary was sick*)
2. infinitival phrases (e.g., *to go to the movies*)
3. gerund phrases (e.g., *John's hitting Fred*)

(25) [New York is one of only three states]^{EDU_1} [that do not allow some form of audio-visual coverage of court proceedings]^{EDU_2}. [Some lawmakers worry]^{EDU_3} [that cameras might compromise the rights of the litigants]^{EDU_4}. [But a 10-year experiment with courtroom cameras showed]^{EDU_5} [that televised access enhanced public understanding of the judicial system]^{EDU_6} [without harming the legal process]^{EDU_7}. [New York's backwardness on **this issue** hurts public confidence in the judiciary...]^{EDU_8}

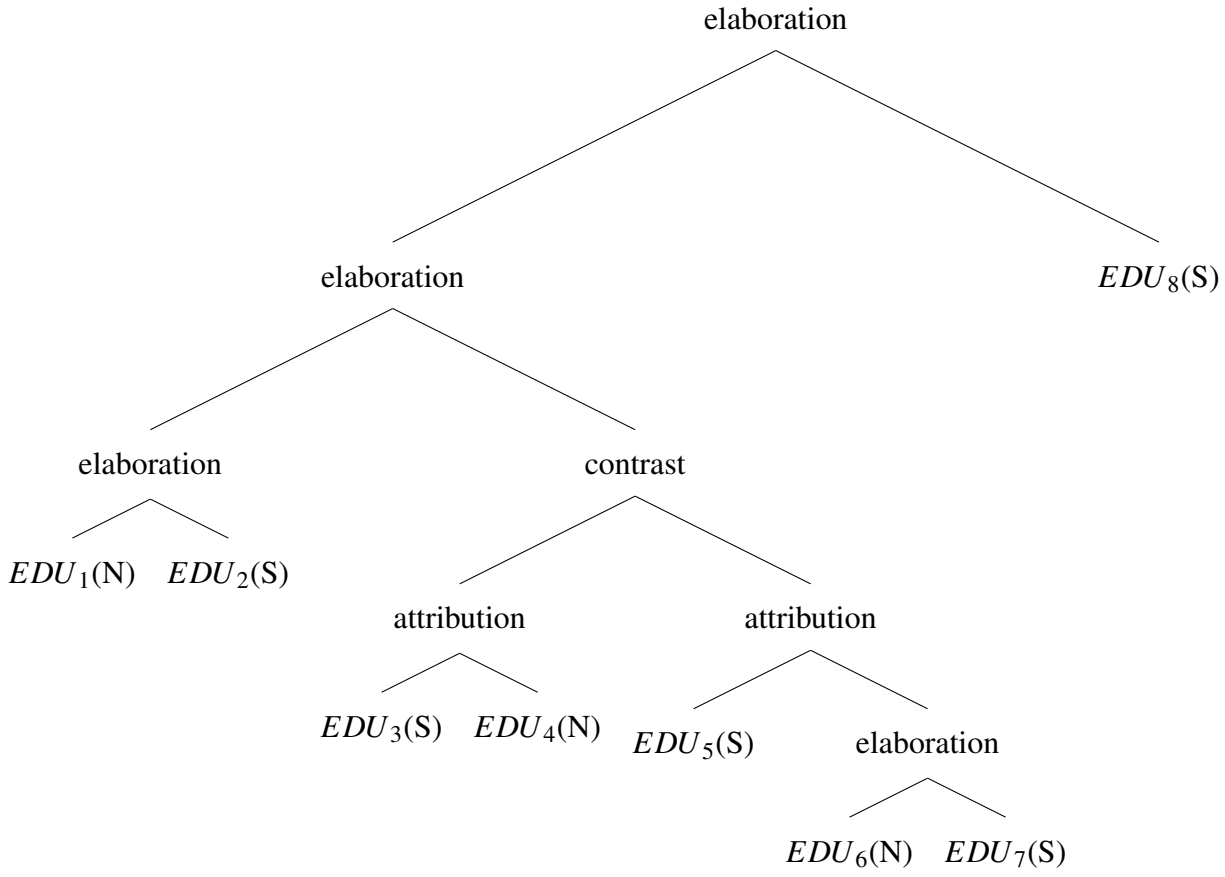


Figure 2.3: Discourse tree for example (25), given by Feng and Hirst (2014). N = nucleus, S = satellite.

4. naked infinitive complements (e.g., *John saw Mary leave.*)
5. noun phrases that appear to denote proposition-like entities (e.g., *The claim that Susan got a C on the test.*)
6. clauses with parasitic gaps, i.e., implicitly expressed argument

Navarretta (2011)'s analysis of usages of *this*, *that*, and *it* in Danish (455 instances) and Italian (114 instances) shows that the antecedents of such anaphors most frequently occur in subordinate and simple main clauses, and matrix clauses are very rare for such antecedents.

Similarly, shell content can take different syntactic shapes. Moreover, there are other noun-specific preferences for difference shell nouns. For instance, *purpose* and *decision* take a *to*-clause; *explanation* and *fact* take a *that*-clause; and *question* takes a *wh*-question clause.

2.1.5.4 Distance between anaphor and antecedent

In anaphora resolution, an important factor that affects the accessibility of antecedents is the distance between the anaphor and antecedent. A short distance between the anaphor and the antecedent implies a smaller search space and a smaller number of competing antecedent candidates, whereas a long-distance implies a larger search space with many competing antecedent candidates. In case of expressions with abstract antecedents, such as *this*, *that*, and *it*, typically this distance is short. In particular, demonstrative pronouns alone are not particularly informative by themselves and so the distance between the anaphor and the antecedent is fairly small and the textual coherence fairly strong, i.e., there are fewer competing candidates. In contrast, shell nouns are informative because of the presence of shell nouns in them and they can allow long-distance as well as short-distance antecedents, as shown in the following examples.

(26) Once an international poverty line is set, it must be converted to local currencies. This is trickier than it sounds. Currency exchange rates are inappropriate because most of the items that the poor consume are not traded on world markets. **Living expenses are much lower in rural India than in New York**, but **this fact** is not fully captured if prices are converted with currency exchange rates.

Here, the distance between the anaphor and the antecedent is small: the antecedent of *this fact* occurs in the preceding clause. In contrast, in (27), the antecedent of *this issue* occurs four sentences away from the anaphor sentence.

(27) Among Roman Catholics, the differences were even more striking. Only 28 percent of Catholics who said religion was very or extremely important to them favored keeping abortion legal, but 72 percent of Catholics for whom religion was less important favored the legal status quo.

The sense of a public struggling with a morally difficult issue was dramatically conveyed when the survey asked: "**Would you approve or disapprove of someone you know having an abortion?**"

Thirty-nine percent said they would approve and 32 percent said they would disapprove. But 25 percent more volunteered a response not included in the question: they said their view would depend on the circumstances involved. An additional 5 percent did not know. The lack of a clear majority for either of the unequivocal responses to **this question** may be the best indicator of where public opinion really stands on abortion.

2.2 Related work in annotation

2.2.1 Introduction

In the previous section we talked about the linguistic account of shell nouns: different terminologies in the literature, shell noun categorization, and their similarity with abstract anaphora and deictic expressions. In this section, we talk about attempts to carry out annotation of shell content of shell nouns and other anaphors with similar properties. In computational linguistics, most of the annotated corpora focus only on anaphoric relations between noun phrases and there exist only a few corpora that consider non-nominal antecedents. Dipper and Zinsmeister (2011) provide a good survey of various annotated corpora for abstract anaphora. Most of these deal with English and consider only anaphoric instances of personal and demonstrative pronouns (Passonneau, 1989; Eckert and Strube, 2000; Byron, 2003; Müller, 2008; Hedberg et al., 2007; Poesio and Artstein, 2008; Navarretta, 2011).

Müller (2008) focused on annotation of *it*, *this*, and *that* to their antecedents in the ICSI Meeting Corpus to collect training and test data that could be used by an automatic anaphora resolution system. He asked naive annotators who did not have any bias about the task to annotate 59 anaphoric chains. He defined a simple annotation scheme to deal with non-nominal VP antecedents that suggests marking finite or infinite verbs as proxies which provide sufficient information for the identification of the larger units such as sentences or clauses. Hedberg et al. (2007) annotate 321 demonstratives from the New York Times corpus with the goal of identifying the cognitive status of their antecedents. For that they marked the type of the antecedent (DIRECT or INDIRECT), the antecedent, and its cognitive status (INFOCUS or ACTIVE). They report $\kappa = 0.46$ (moderate agreement) for identifying the cognitive status of the antecedent, and $\kappa = 0.70$ (substantial agreement) for identifying the type of the antecedent. They do not report agreement in identifying the actual antecedents. Poesio and Artstein (2008) created the ARRAU corpus — the largest annotated corpus, containing 455 anaphors pointing to non-nominal antecedents. Navarretta (2011) studied use of *this*, *that*, and *it* in Danish (455 instances) and Italian (114 instances) written and spoken data. In particular, she annotated the following properties for each instance: the type of the pronoun, the antecedent, the semantic and syntactic type of the antecedent, and the anaphoric distance in terms of clauses.

The OntoNotes project[6] has created a multilingual corpus that includes reliably annotated event coreference, among annotation of other shallow semantic structures. The task of event coreference annotation is identifying co-referring event verbs, as shown in example (28) taken from Lee et al. (2012).

(28) a. The New Orleans Saints placed Reggie Bush on the injured list on Wednesday.
b. Saints put Bush on I.R.

OntoNotes 2.0 contains 300K of English newswire data from the *Wall Street Journal* and 200K of English broadcasting news from various sources including ABC and CNN. But as Chen et al. (2011) note, the distribution of event chains in the corpus is quite sparse and the chains

[6] http://www.bbn.com/ontonotes/

are quite short.

Dipper and Zinsmeister (2011) point out that for annotating non-nominal antecedents as spans of text, there is no standard way of reporting inter-annotator agreement. Some studies report only observed percentage agreement with results in the range of about 0.40–0.55 (Vieira et al., 2002; Dipper and Zinsmeister, 2011). Artstein and Poesio (2006) discuss Krippendorff's α for chance-corrected agreement. They considered antecedent strings as bags of words and computed the degree of difference between them by different distance measures, such as Jaccard and Dice. Depending on the distance measure, they observed agreement between 0.47 and 0.57, which resulted in only slightly lower chance-corrected α between 0.45 and 0.55.

A few projects annotate demonstrative NPs (Vieira et al., 2002; Poesio and Modjeska, 2002; Artstein and Poesio, 2006; Botley, 2006). Although these projects do not specifically consider shell nouns, when demonstrative NPs occur with abstract head nouns, they largely overlap with anaphoric shell nouns (about 80% of the time (Vieira et al., 2002)).

2.2.2 Annotating demonstrative NPs

Poesio and Modjeska (2002) As noted in Section 2.1.2, Poesio and Modjeska (2002) analyzed 112 this-NPs from two corpora: the museum subcorpus consisting of descriptions of museum objects and brief texts about the artists that produced them, and the pharmaceutical subcorpus which is a selection of leaflets providing patients with mandatory information about their medicine. Poesio and Modjeska were interested in the cognitive status of this-NPs in the given discourse. Due to the difficulties associated with identifying the precise antecedent of this-NPs, they developed an annotation scheme where the annotators do not have to mark the actual antecedents; rather the scheme instructs the annotators to classify this-NPs into different categories such as visual deixis, discourse deixis, and anaphoric, and based on these categories, they assign a cognitive status to each THIS-NP instance. They tested the reliability of this scheme by measuring the agreement among two annotators on about 87 this-NPs in the corpus. They get a $\kappa = 0.82$ on this classification task: the annotators disagreed on 3 this-NPs

and 5 were classified as problematic instances.

Vieira et al. (2002) Vieira et al. point out the need to focus on the specific treatment of demonstrative NP anaphora. They analyzed syntactic, semantic, and discourse features related to demonstrative NPs in French and Portuguese from the MLCC multilingual and parallel corpora.[7] They considered 250 demonstrative noun phrases for each language.

Vieira et al. carried out their annotation in three phases. In the first phase, one annotator marked all demonstrative descriptions from the corpus as markables. In the second phase, two native speakers marked the antecedents of the previously identified markables. Finally, in the third phase, the annotators identified the relationship between the demonstrative NPs and their marked antecedents. For the antecedent identification task, they report inter-annotator agreement of 51% and 69.8% for Portuguese and French, respectively. Below are a few relevant observations from their analysis that apply to both French and Portuguese.

First, they observed a clear predominance of abstract nouns in demonstrative noun phrases: about 80% of the markables occurred with abstract head nouns. They also analyzed annotations with respect to the semantic relation between the anaphors and their antecedents and showed that there was a clear predominance of hypernymy relation (e.g., antecedent = *Canada*, anaphor = *this country*).

Second, there is a relation between the syntactic complexity and discourse roles of demonstrative NPs. For instance, for both languages, demonstrative NPs followed the simple syntactic structure of DET N about 80% of the time. By contrast, definite descriptions occurred with a variety of simple as well as complex syntactic structures, e.g., adjectival (e.g., `DET (ADJ N|N ADJ)`), prepositional or relative clause modification (e.g., `DET (N|ADJ N|N ADJ) OF N)`).

Third, about 62% of the time the antecedents were noun-phrases. However, the remaining cases, the antecedents were either a single sentence, part of a sentence or a paragraph. So an anaphora resolution system designed to resolve demonstrative NPs that considers NP structures only will fail on these remaining 38% of the cases.

[7] `http://catalog.elra.info/product_info.php?products_id=764`

Fourth, they observed that about 80% of demonstratives with concrete head nouns have a coreference relation with their antecedent whereas only about 40% of demonstratives with abstract head nouns have a coreference relation with their antecedents. They also point out that this observation could be used as a baseline to evaluate systems for demonstratives with abstract head nouns.

2.2.3 Summary

There is not much work done in annotating shell content. There is no suitable annotation scheme or a systematic way to compute inter-annotator agreement for annotating non-nominal antecedents. Müller's scheme of annotating proxies and keeping the scope of the antecedent flexible is attractive, as it can deal with the problem of imprecise boundaries of shell content. However, there are two problems with this approach. First, the verb gives only partial information about the antecedent. For instance, in example (23), marking the verb *crashed* as the antecedent does not tell us whether we are talking about the event or the concept or the fact: in (23) 2., the antecedent of *that* must exclude the subject of the verb *crashed*, whereas in (23) 4., both arguments of the verb have to be included in the antecedent. Second, shell content in the form of nominalizations or containing multiple verbs (two clauses connected by a conjunction) cannot be expressed effectively with this annotation scheme. So there is a need to examine the feasibility of shell content annotation and the extent to which humans agree on the shell content.

2.3 Related work in resolution

2.3.1 Introduction

There is a cline from plain nominal-anaphora down to anaphors with abstract antecedents, and it passes through cases such as *bridging reference*, *other*-anaphora, and *event*-anaphora. There

is no holistic approach that tackles all these anaphoric expressions.

Annotated datasets such as MUC and ACE are limited to annotations of a particular kind of coreferent entities. In particular, they consider only multiple ambiguous mentions of a single entity representing a person, an organization, or a location. Note that the term *coreference resolution* is closely related but not identical with the term *anaphora resolution*. Two discourse entities co-refer if they refer to the same object. But not all anaphoric expressions in a language co-refer in this sense. Also, two entities across documents can co-refer but they might not be anaphoric.

Other hard cases of anaphora have been studied sparsely in the field. *Bridging* reference (e.g., *We went for **a concert** by the Toronto Symphony Orchestra. **The violinist** played some solo pieces for the audience.*) has received fair amount of attention both in terms of recognition and antecedent selection (Poesio et al., 2011). *Other*-anaphora (e.g., *British and **other** European steelmakers*) has been studied by Modjeska (2003), but the work is limited to *other* anaphors with nominal antecedents only. With the availability of the OntoNotes corpus[8] and EventCorefBank corpus[9], there has been some work on resolving *event anaphora* (Pradhan et al., 2007; Chen et al., 2011; Lee et al., 2012). Pradhan et al. (2007) applied a conventional coreference resolution algorithm on the OntoNotes corpus, but they do not report separate performance on event anaphora. Chen et al. (2011) defined seven distinct mention-pair models for different syntactic types (NPs, verbs, and pronouns) of coreferent mentions. Along with traditional coreference resolution features, they explored syntactical structural information embedded in the parse trees.

Apart from these specific approaches, the more general problem of resolving anaphors with non-nominal abstract antecedents has received some attention in the field, which are closest to shell noun resolution. So I will discuss in detail some approaches that tackle abstract anaphora, i.e., anaphora where the antecedent is an abstract object.

[8] `http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2008T04`
[9] `http://faculty.washington.edu/bejan/data/ECB1.0.tar.gz`

2.3.2 Resolving abstract anaphora in spoken dialogues

Eckert and Strube (2000) The work of Eckert and Strube (2000) is one of the earliest that tackles abstract anaphora in spontaneous spoken dialogues. The work outlines an algorithm that 1) identifies whether a given occurrence of a pronoun or demonstrative has an abstract antecedent and 2) provides a resolution process for pronouns with abstract antecedents as well as concrete antecedents.

The first step is identifying whether the pronoun or demonstrative participates in abstract anaphora or individual, i.e., concrete anaphora. For that, Eckert and Strube came up with a list of rules shown below. They suggest that if a pronoun or a demonstrative occurs in any of the following contexts, it is considered to be a preferred candidate that exhibits abstract anaphora.

- Equating constructions where a pronominal referent is equated with an abstract object, e.g., *x is making it easy*. *x is a suggestion*.
- Copula constructions whose adjectives can only be applied to abstract entities, e.g., *x is true*, *x is false*, *x is correct*, *x is right*, *x isn't right*
- Arguments of verbs describing propositional attitude which only take sentential complements, e.g., *assume*.
- Object of *do*.
- Anaphoric referent is equated with a 'reason', e.g., *x is because I like her*, e.g., *x is why he is late*.

Following Asher (1993), Eckert and Strube assume that the predicate of a discourse deictic anaphor determines its semantic type. For instance, for the anaphor in the subject position of *is true*, the antecedent must be of type proposition with a full sentential syntactic form (e.g., ***John sang***. ***That****'s true.*), whereas if the anaphor is in the object position of the *do* verb, then the antecedent must be an event type antecedent, (e.g., ***John sang***. *Bill did* ***that*** *too.*).

Following Webber (1991)'s idea of antecedent coercion for discourse deictic anaphors, Eckert and Strube assume that an abstract object referent of such an anaphor is introduced to the discourse model by the anaphor itself. Accordingly, they maintain a list of abstract objects

accessed so far by means of using anaphoric expressions that replace them. This list is referred to as *A-List*.

For actual resolution, they suggest the *context ranking* algorithm, which is inspired by Webber (1991)'s right-frontier rule. The context ranking algorithm considers the following antecedent candidates.

- A-List (all abstract objects previously referred to anaphorically)
- Clause to the left of the clause containing the anaphor
- Rightmost main clause and subordinating clause to its right
- Rightmost complete sentence

Given these candidates, a linear search is carried out to find the correct antecedent. For every candidate, the algorithm examines the candidate's compatibility with the anaphor and returns the candidate as the correct antecedent if they are compatible. In particular, if the anaphor is an argument of *do*, the VP of the candidate is added to the A-List and returned as the correct antecedent. Otherwise, the candidate itself is added to the A-List and returned as the correct antecedent.

Eckert and Strube's algorithm is not implemented. Manual execution of the algorithm resulted in 63.6% precision and 70% recall on 70 instances of discourse deictic anaphors.

Byron (2004) Byron (2004) implemented a system called PHORA that tackles abstract antecedents. PHORA resolves personal and demonstrative pronouns in task-oriented TRAINS93 dialogues. PHORA works as follows. The first step is building a discourse model for the given discourse. The model is composed of the set of discourse entities (DEs) and the set of proxy discourse entities (proxy DEs). DEs are all noun phrases (NPs) from the entire discourse so far, whereas proxy DEs represent the semantic content of other constituents such as sentences, embedded sentences, predicates, and verbals. DEs and proxy DEs are also referred to as mentioned entities and activated entities following Gundel's hierarchy, shown earlier in Figure 2.1. Unlike DEs, proxy DEs are only available to be used by anaphors for a short time — only the

Table 2.6: Example discourse model (Byron, 2004)

Input	Number	Type	Composition	Specificity	Interpretation	Salience
Engine 1	Sing	ENGINE	Homogenous	Indiv	ENG1	focus
Avon	Sing	CITY	Homogenous	Indiv	AVON	mentioned
the oranges	Plural	ORANGE	Homogenous	Indiv	ORANGES1	mentioned
to get oranges	Sing	Functional	Homogenous	Kind	proxy	activated
all of (29)	Sing	Functional	Homogenous	Indiv	proxy	activated

Table 2.7: Referring functions (Byron, 2004)

Function	DE/Proxy details	Output Types
Ident(d,t)	Any mentioned DE if its type meets type constraint t	**The train** is red.
Kind(d,t)	Descriptive NP (not bare plural NP) meeting of type t	That's a **great route**.
Situation(d)	Sentence with tensed stative verb	The train **is** red.
Event(d)	Sentence with tensed eventive verb	It **gets there** late.
Kind_A(d)/Kind_E(d)	Infinitive form of action/event	**To load them** takes an hour.
	Gerund form of action/event	**Loading them** takes an hour.
Proposition(d)	Each TELL or YN-question	**I think that he's an alien.**
	that sentential	I think that **he's an alien**.
	if/when subordinating clause	If **he's an alien...**
Endtime(d)	Sentence with tensed eventive verb	**Then we go to Avon.**
	Gerund or infinitive from eventive verb	I need **to load** the boxcars.

most recent ones are stored in the discourse model (proxy DEs from the preceding discourse unit, i.e., preceding clause). Each DE is stored with the following attributes: the surface linguistic constituent, number (singular or plural), semantic type, composition (heterogeneous or homogeneous), specificity (individual or kind), interpretation (the referent or proxy associated with this DE). For instance, given sentence (29), the DEs and proxy DEs of the discourse model will be as shown in Table 2.6. The mentioned DE *Engine 1* is designated as the focus of the discourse. The proxy DEs are activated entities.

(29) Engine 1 goes to Avon to get the oranges.

The resolution of pronouns works as follows. When a pronoun is encountered, the first step of pronoun resolution is to identify the semantic type of its antecedent based on its context. The semantic type constraints come from verbs, predicate NPs, and predicate adjectives. For instance, in *It's right*, the semantic type of the pronoun *it* is PROPOSITION, which comes from the predicate adjective *right*.

The next step of the resolution process is a search for the referent that satisfies the semantic type constraints and agreement features for the pronoun. For mentioned entities the search is carried out backwards in the order of clauses appeared in the discourse. For activated entities, the proxy DEs are searched. The proxy DEs can be used to access many different types of referents by applying *referring functions*, as shown in Table 2.7. The referring functions coerce the proxy into a referent of the desired type. A referring function returns the referent if the semantic and agreement constraints are satisfied and returns NIL otherwise.

Byron assumes a linear discourse model and carries out the antecedent search as follows. For personal pronouns, the entities are searched in the following order.

- mentioned entities in the current clause are searched in right-to-left order
- the focused entity from the preceding clause
- the remaining mentioned entities
- activated entities from the preceding clause

For demonstrative pronouns, the entities are searched in the following order.

- activated entities from the preceding clause
- the focused entity only if it can coerced to a Kind
- mentioned DEs from the entire discourse

For instance, for the following continuation of example (29).

(30) **That** takes two hours.

The demonstrative *That* is of type TAKE_TIME(X), where X must be an event. Following the search order for demonstratives, first we will look for the activated DEs and call the referring function Event(d). The function will be successful on the proxy DE (29) as the verb *goes* in this proxy DE is a tensed eventive verb.

With this approach, Byron achieved an accuracy of 72% (baseline = 37%) in resolving 180 test pronouns from ten problem-solving dialogues in the TRAINS93 corpus. Byron does not report separate numbers for anaphors with NP and non-NP antecedents. However, she

reports that the performance increases from 43% to 67% with the inclusion of abstract non-NP antecedents.

2.3.3 Resolving *this*, *that*, and *it* in multi-party dialogues

Müller (2007, 2008) presents an implemented system for the resolution of *it*, *this*, and *that* in spoken multi-party dialogue. He proposes to resolve these pronouns as a binary classification task. In particular, a pronoun is resolved by creating a number of candidate antecedents and searching this set based on two factors: constraint and preferences. Müller controls the search space of the antecedent candidates by excluding all non-nominal and non-verbal candidates. His system is the first one that applies supervised machine learning on this problem with a large number of automatically extracted novel features. Müller (2008, p. 150) discusses all features he incorporated in detail. The traditional anaphora resolution features such as number and gender of the anaphor and the antecedent are not relevant in case of shell nouns. Below we discuss some of his features that could be relevant for shell noun resolution.

Type The type feature includes the morphological type of the anaphor (e.g., demonstrative or pronoun) and the antecedent (e.g., noun, proper noun, infinite verb, finite verb). The intuition behind this feature is high-level separation of instances based on the type of different expressions.

Distance Müller incorporates a number of distance features including the distance in words between the anaphor and the antecedent, the distance in seconds between the anaphor and the antecedent, the distance of the anaphor and the antecedent to the previous disfluency, and the distance in sentences between the anaphor and the antecedent.

Syntactic features Müller includes a number of features extracted from constituency and dependency parse trees. The embedding level feature encodes the embedding of the candidate in the immediate clause and the top clause in a constituency parse tree. According to Müller,

this feature is an approximation of the syntactic complexity of the constructions containing the antecedent.

The grammatical function features capture the detailed grammatical functions of the anaphor and the antecedent (e.g., subject, object), whether the antecedent is an object of the verb *do*, and the tense of the governing verb of the antecedent. Müller also includes a feature that captures whether the antecedent or the anaphor is the object in a clause with an existential *there* as subject. The feature is based on Lappin and Leass (1994)'s *existential emphasis* salience factor.

Lexical features Müller includes two classes of lexical features. The first class of features is based on the list of subordinating conjunctions given by Passonneau (1994) (e.g., *as, because, whether, yet*). Passonneau claims that being governed by these conjunctions is a necessary condition for a clause to be functionally independent. Müller incorporates the feature that examines whether the immediate clause containing the expression is governed by the conjunctions given by Passonneau in order to identify whether the clause is a main clause or a subordinating clause.

The second class of features examines whether the candidate is headed by one of the prepositions appearing in the list given by Paice and Husk (1987) (e.g., *among, before, beside, despite*). Paice and Husk use this list for detecting non-referential *it*. According to them, the prepositions in this list are indicators for referential usages.

Compatibility of anaphora and antecedents Another set of features are the *tipster* features. These features represent conditional probabilities from TIPSTER corpus[10] ($\approx 250,000,000$ words) for compatibility of anaphor and antecedent. The first tipster feature captures incompatibility of concrete antecedents with the given anaphor. Recall that Eckert and Strube (2000) note that pronouns in a subject position with a copula construction with adjectives (e.g., *x is true*) are incompatible with concrete objects and show preference for abstract objects. Müller

[10] `http://catalog.ldc.upenn.edu/LDC93T3A`

(2008) operationalizes Eckert and Strube (2000)'s I-Incompatibility as the conditional probability of an adjective to appear with a *to*-infinitive and *that*-sentence complement. For instance, for the former the probability is calculated as:

$$\frac{\#it\ (\textit{'s} \mid \textit{is} \mid \textit{was} \mid \textit{were})\ \text{ADJ}\ \textit{to}}{\#it\ (\textit{'s} \mid \textit{is} \mid \textit{was} \mid \textit{were})\ \text{ADJ}} \tag{2.1}$$

For instance, for the adjective *true*, *to*-infinitive probability is 0.0045 and *that*-infinitive probability is 0.5412.

The second tipster feature is related to the semantic compatibility of the anaphor and NP antecedent. The semantic compatibility is computed by substituting the anaphor with the antecedent head and performing corpus queries. For instance, if the anaphor is subject in an adjective copula construction (e.g., *This is true.*) the following corpus query is performed to quantify the compatibility between the predicated adjective and the NP antecedent. In particular, the query quantifies how many times the head of the antecedent occurs with the given adjective relative to the total number of occurrences of the adjective.

$$\frac{\#\text{ADJ (ANTE} \mid \text{ANTES)} + \#\ \text{ANTE}\ (\textit{is} \mid \textit{was})\ \text{ADJ} + \#\ \text{ANTE}\ (\textit{are} \mid \textit{were})\ \text{ADJ}}{\#ADJ} \tag{2.2}$$

For instance, adjectives such as *hungry*, *guilty*, and *naughty* are more compatible with NPs. So the probability associated with the compatibility of the adjective *guilty* and the noun *verdict* is 0.017.

Other features Other features include the lemma of the anaphor, category of the antecedent (VP or NP), size of the antecedent, presence of determiners and the person for NP antecedents, and number of arguments for VP antecedents.

Müller achieved a very low F-measure of 18.63 (baseline = 8.13). He concluded that the pronouns *it*, *this*, and *that* in spoken multi-party dialog to a large extent defy an automatic

resolution. He points out a number of potential reasons for the low performance. First reason is the nature of spontaneous spoken language in contrast to written text. In a spoken language pronouns are more often vague or ambiguous. Second reason is that the corpus consisted of multi-party dialogue which adds to referential ambiguities. Third reason is the abstract nature of the antecedents of the anaphors he considers.

2.3.4 Summary

To the best of my knowledge, there is no automatic approach that addresses resolution of shell nouns. Previous work on resolution of abstract anaphora in general is rather limited. Eckert and Strube (2000)'s approach is not implemented. Moreover, it is dependent on non-trivial information about the incompatibility of a pronoun or a demonstrative with a concrete antecedents. Byron's knowledge-deep, rule-based, and non-probabilistic approach seems promising. However, it is implemented only in a closed domain of TRAINS93 dialogues — it is based on the manually extracted semantic types of the abstract and concrete objects in the TRAINS93 dialogues.

Most of the current methods are domain-specific, heavily use manually extracted feature values and domain knowledge, and are restricted to particular syntactic shapes. Müller (2008)'s approach is an automatic approach. But it is limited to abstract antecedents expressed with verb phrases and noun phrases to control the large search space of non-nominal antecedents. Also, Müller's achieved a very low resolution performance in the spoken dialogue domain. The sparse related work in resolution of related anaphora shows the need to develop a more flexible algorithm that can tackle arbitrary spans of text as antecedents. But is it practical to develop such an algorithm? What do we mean by arbitrary — a phrase, a clause, a sentence, a paragraph, or even longer stretches of text? In the subsequent chapters, I try to answer these questions.

Chapter 3

A pilot study of resolving shell nouns

3.1 Introduction

This chapter reports on a pilot study of annotating and resolving shell nouns. In particular, it presents an end-to-end procedure for annotating and resolving anaphoric occurrences of the prototypical shell noun *issue* in the Medline[1] abstracts.[2] The shell noun *issue* was chosen for the following reasons. First, it occurs frequently in all kinds of text from newspaper articles to novels to scientific articles. There are more than 90,000 anaphoric instances of the shell noun *issue* in the New York Times corpus (Table 2.3). Second, the shell noun *issue* is abstract enough that it can take several syntactic and semantic forms, which makes the problem interesting and non-trivial. Third, *issue* referents in scientific literature such as Medline abstracts generally lie in the previous sentence or two, which makes the problem tractable.

Since we focus on only anaphoric instances of the shell noun *issue*, i.e., *this issue* instances, we use the terms *antecedent* and *shell content* interchangeably.

[1] `http://www.ncbi.nlm.nih.gov/pubmed`

[2] This work is presented in Kolhatkar and Hirst (2012).

3.2 Annotation

Although native speakers have almost no trouble in understanding the text containing shell nouns, pinpointing their precise shell content is tricky even for humans. As we noted in Section 2.2, there has been some work in annotating anaphors with similar properties, i.e., anaphors with abstract antecedents (Dipper et al., 2011). But there is no work that systematically annotates shell nouns and reports the reliability of this task. Also, there have been open questions on measuring inter-annotator agreement, as the boundaries of shell content are imprecise. For instance, in example (31), it is hard to tell with the given context whether *this decision* refers to *selling the standing-room-only tickets* or *allowing to sell the standing-room-only tickets*.

(31) In principle, he said, airlines should be allowed to sell standing-room-only tickets for adults — as long as **this decision** was approved by their marketing departments.

So before any computational treatment, it is necessary to examine whether it is possible to create a well-grounded corpus of shell nouns and their corresponding shell content. Accordingly, the first question is to what extent human annotators agree on shell content. The next sections describe our procedure to annotate *this issue* instances in the Medline abstracts.

3.2.1 The corpus

Medline is a freely available database that contains bibliographical information for articles from life sciences. For the pilot study, we chose Medline abstracts as domain for their clear structure and restrictive context. Medline abstracts are generally well-structured and are rather long, as shown in (32).

(32) **A comparative evaluation of nitrous oxide-isoflurane vs isoflurane anesthesia in patients undergoing craniotomy for supratentorial tumors: A preliminary study.**

BACKGROUND: Neuroanesthesiologists are a highly biased group; so far the use of nitrous oxide in their patient population is concerned. We hypothesized that any adverse consequence with use of nitrous oxide should affect the patient so as to prolong his/her stay in the hospital. The primary aim of this preliminary trial was to evaluate if avoidance

of nitrous oxide could decrease the duration of Intensive Care Unit (ICU) and hospital stay after elective surgery for supratentorial tumors.

PATIENTS AND METHODS: A total of 116 consecutive patients posted for elective craniotomy for various supratentorial tumors were enrolled between April 2008 and November 2009. Patients were randomly divided into Group I: Nitrous oxide - Isoflurane anesthesia (Nitrous oxide-based group) and Group II - Isoflurane anesthesia (Nitrous oxide-free group). Standard anesthesia protocol was followed for all the patients. Patients were assessed till discharge from hospital.

RESULTS: The median duration of ICU stay in the nitrous group and the nitrous-free group was 1 (1 - 11 days) day and 1 (1 - 3 days) day respectively (P = 0.67), whereas the mean duration of hospital stay in the nitrous group was 4 (2 - 16) days and the nitrous free group was 3 (2 - 9) days (P = 0.06). The postoperative complications in the two groups were comparable.

CONCLUSION: From this preliminary study with a low statistical power, it appears that avoidance of nitrous oxide in one's practice may not affect the outcome in the neurosurgical patients. Further large systemic trials are needed to address **this issue**.

There are several benefits in working with the Medline abstracts for the pilot study. First, the antecedents of *this issue* are relatively well-defined in this domain. Second, limited context of abstracts restricts the antecedent search space. Finally, *issues* in Medline abstracts are generally associated with clinical problems in the medical domain and spell out the motivation of the research presented in the article. So extraction of this information would be useful in any biomedical information retrieval system.

We started with 200 Medline abstracts (similar to example (32)) containing *this-issue* instances. After removing duplicates, we were left with 188 abstracts.

3.2.2 Annotation procedure

From the original collection of 188 instances, five instances were discarded as they had an unrelated (publication related) sense. Among the remaining 183 instances, 132 instances were independently annotated by two annotators, a domain expert, Dr. Brian Budgell from the Canadian Memorial Chiropractic College, and a non-expert (myself), and the remaining 51

instances were annotated only by the domain expert. Following Dipper et al. (2011), we chose to mark free spans of text. We used the former instances for training and the latter instances (unseen by myself) for testing. The annotators' task was to identify and mark text segments as antecedents, without concern for their linguistic types. The annotation comprises the following information:

- ANTECEDENT marked with <ANTECEDENT> tag.

- REFERENT_TYPE, which encodes the syntactic type of the antecedent. The type could be NP (noun phrase), CLAUSE (clause), SENT (sentence), SSENT (sequence of sentences), VP (verb phrase), or MIXED (combination of different syntactic constituents).

- DIST, which encodes the distance of the antecedent from the anaphor. This attribute can take three values: ADJA (adjacent sentence), SAME (same sentence), and REM (2 or more sentences away from the anaphor)

- EXTRA attribute which contains any extra information that the annotator wants to include, in the form: "field$_1$:value$_1$, field$_2$:value$_2$, ..., field$_n$:value$_n$".

Below, I show annotation of example (32). The antecedent *that avoidance of nitrous oxide in one's practice may not affect the outcome in the neurosurgical patients* is marked for the anaphor with ID="2". The REFERENT_TYPE of this antecedent is "CLAUSE" and the DIST attribute has value "ADJA" as it lies in the adjacent sentence. The annotator includes an EXTRA attribute PARAPHRASE in the annotation because the actual referent is not explicitly stated in the text which would be *whether avoidance of nitrous oxide in one's practice affects the outcome in the neurosurgical patients*, a variant of the textual antecedent. The detailed annotation guidelines are given in Appendix D.

```
<AbstractText>
<AbstractText Label="CONCLUSION" NlmCategory="CONCLUSIONS">
From this preliminary study with a low statistical power,
it appears <ANTECEDENT ID="2">that avoidance of nitrous
oxide in one's practice may not affect the outcome in the
neurosurgical patients</ANTECEDENT>. Further large
systemic trials are needed to address <ANAPHOR ID="2"
DET="this" NOUN="issue" REFERENT_TYPE="CLAUSE"
DIST="ADJA" EXTRA="PARAPHRASE:whether avoidance of
nitrous oxide in one's practice affects the outcome in the
neurosurgical patients">this issue</ANAPHOR>.
</AbstractText>
```

3.2.3 Inter-annotator Agreement

Annotating shell content as free spans of text involves marking different kinds of syntactic constituents.[3] There is no standard way in the literature to report inter-annotator agreement for such kind of annotation. Some studies report only observed percentage agreement with results in the range of about 0.40–0.55 (Vieira et al., 2002; Dipper and Zinsmeister, 2011).[4]

It is well known that in order to get figures that are comparable across studies, observed agreement has to be adjusted for chance agreement (Artstein and Poesio, 2008). Moreover, for this kind of annotation, we need more than just a chance corrected agreement. For instance, agreement coefficients such as Cohen's κ, which are adjusted for chance agreement, underestimate the degree of agreement for such annotation, suggesting disagreement even between two very similar annotated units (e.g., two text segments that differ in just a word or two). For example, suppose annotator1, annotator2, and annotator3 mark antecedents for *issue* in (33) as shown in a), b), and c) below, respectively. Cohen's κ, in this case, will consider this as a complete disagreement, whereas, in fact, the annotations are pretty close and represent essentially the same concept. A reasonable inter-annotator agreement coefficient for such annotation

[3] Sometimes they are not even well-defined syntactic constituents as defined by an automatic parser.

[4] The variation in the results is mainly due to the variation in the number of annotators, types of anaphors, and language of the corpora.

should be able to realize that a) and b) are more distant than (b) and (c), and that b) and c) are very close to each other.

(33) This prospective study suggested that oral carvedilol is more effective than oral metoprolol in the prevention of AF after on-pump CABG. It is well tolerated when started before and continued after the surgery. However, further prospective studies are needed to clarify **this issue**.

a) This prospective study suggested that oral carvedilol is more effective than oral metoprolol in the prevention of AF after on-pump CABG.

b) that oral carvedilol is more effective than oral metoprolol in the prevention of AF after on-pump CABG

c) oral carvedilol is more effective than oral metoprolol in the prevention of AF after on-pump CABG

In the context of this kind of annotation, Artstein and Poesio (2006) suggest two inter-annotator agreement measures that try to deal with the problem of fuzzy boundaries. The first measure is the percentage of annotators that chose the most common choice for each markable. In particular, this measure considers the most common choice for the beginning and end for an antecedent across annotators. However, this measure does not take into account the fact that there might be a substantial overlap in the antecedents that the annotators mark although they do not agree on the exact beginning and ending of their antecedents. Also, it's unclear how this measure works for split antecedents, i.e., where an annotator marks an antecedent as a discontinuous string. Second, they discuss Krippendorff's α for chance-corrected agreement. They considered antecedent strings as bags of words and computed the degree of difference between them by different distance measures such as Jaccard and Dice. The bag-of-words approach ignores the information about the exact location where the words in the antecedent lie, and gives rather optimistic scores. So we need an inter-annotated agreement coefficient that goes beyond binary match or mismatch of annotations and incorporates distance between strings more elegantly than Krippendorff's α with distance metrics.

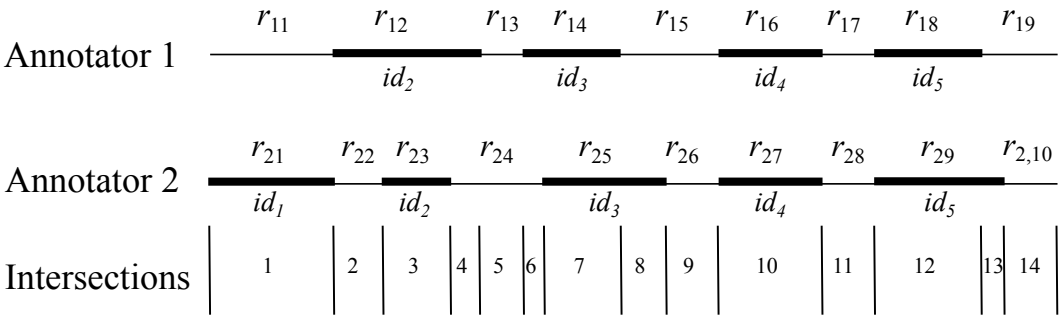


Figure 3.1: Example of annotated data. Bold segments denote the marked antecedents for the corresponding anaphor *ids*. r_{hj} is the j^{th} section identified by the annotator h.

Krippendorff's *unitizing* α I argue that Krippendorff's *reliability coefficient for unitizing* ($_u\alpha$) (Krippendorff, 2013) is a better measure of inter-annotator agreement of segment antecedents. This coefficient is appropriate when the annotators work on the same text, identify the units in the text that are relevant to the given research question, and then label the identified units (Krippendorff, priv. comm.). The general form of coefficient α is:

$$\alpha = 1 - \frac{D_o}{D_e} \tag{3.1}$$

where D_o and D_e are observed and expected disagreements respectively. Both disagreement quantities express the average squared differences between the mismatching pairs of values assigned by annotators to given units of analysis. $\alpha = 1$ indicates perfect reliability and $\alpha = 0$ indicates the absence of reliability. When $\alpha < 0$, either the sample size is very small or the disagreement is systematic.

In our context, the annotators work on the same text, the *this issue* instances. We define an elementary annotation unit (the smallest separately judged unit) to be a word token. The annotators identify and locate *this issue* antecedents in terms of sequences of elementary annotation units.

The general idea of Krippendorff's unitizing α, in our context, is as follows. The annotators mark the antecedents corresponding to each anaphor in their respective copies of the text, as shown in Figure 3.1. The marked antecedents are mutually exclusive sections r; we denote

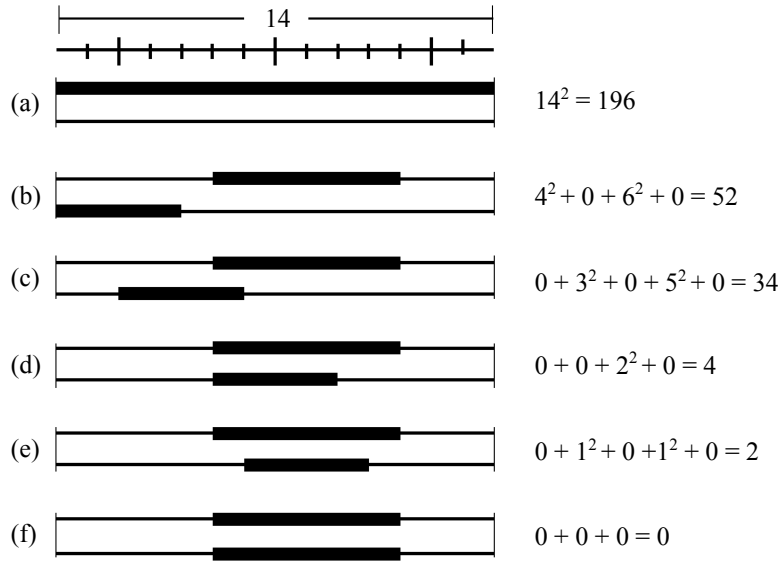


Figure 3.2: The distance function. Adopted from Krippendorff (2004)

the j^{th} section identified by the annotator h by r_{hj}. Each marked section has an identifier (*id*) associated with it which denotes the anaphor for which the antecedent has been marked. For the gaps between the units (irrelevant text) the *id*s have "not-applicable" (NA) value. In Figure 3.1, annotators 1 and 2 have reached different conclusions by identifying 9 and 10 sections respectively in their copies of the text. The NA values are not shown in the Figure. Annotator 1 has not marked any antecedent for the anaphor with $id = 1$, while annotator 2 has marked r_{21} for the same anaphor. Both annotators have marked exactly the same antecedent for the anaphor with $id = 4$. The difference between two annotated sections is defined in terms of the square of the distance between the non-overlapping parts of the sections. The distance is 0 when the sections are unmarked by both annotators or are marked and exactly the same, and is the summation of the squares of the unmatched parts if they are different. Figure 3.2 shows few examples of the distance function. Each example has sections marked by two annotators. We refer to the bold sections as *identified* sections. Example (a) shows the maximum difference possible between the annotations. In example (b), the annotators have nothing in common. Example (c) has identified sections similar to that of (b). However, in this case, the distance drops from 52 to 34 because of one overlapping unit. Example (d) and (e) show the location

sensitivity of the distance function. In (d), one annotator's identified section is contained in other annotator's identified section and the distance is 4 because of the difference of 2 units at the end. Similarly, example (e) also depicts the difference of two units. However, distance drops to 2 from 4 because of the difference between the locations of the contained identified units. The distance is 0 if the annotators completely agree with each other, as shown in example (f).

The coefficient is computed using *intersections* of the marked sections. An intersection boundary is marked when there is a transition between a marked and an unmarked sections in the annotation of any of the annotators. In Figure 3.1, annotators 1 and 2 have a total of 14 intersections. The observed and expected disagreements are computed using the equations given in Krippendorff (2013, p. 313). In brief, the observed disagreement ${}_uD_o$ is the weighted sum of the differences between all mismatching intersections of sections marked by the annotators. And the expected disagreement is the summation of all possible differences of pairwise combinations of all sections of all annotators normalized by the length of the text (in terms of the number of tokens) and the number of pairwise combinations of annotators. For our data, the inter-annotator agreement was $\alpha_u = 0.86$ (${}_uD_o = 0.81$ and ${}_uD_e = 5.81$), which is considered to be a strong indicator for reliably annotated data.[5]

3.2.4 Gold corpus statistics

A gold-standard corpus was created by resolving the cases where the annotators disagreed. Among 132 training instances, the annotators could not resolve 6 instances and we broke the tie by writing to the authors of the articles and using their response to resolve the disagreement. In the gold-standard corpus, 95.5% of the antecedents were in the current or previous sentence and 99.2% were in the current or previous two sentences. Only one antecedent was found more than two sentences back and it was six sentences back. One instance was a cataphor, but the

[5] Artstein and Poesio (2006) observed a low agreement when they gave an option to annotate free spans of text. One reason for high agreement in our case is the restricted domain of Medline abstracts.

antecedent occurred in the same sentence as the anaphor. Also, in our data, antecedents were always continuous spans of text and they did not span multiple sentences but always occurred in a single sentence. This might be because this corpus only contains singular *this-issue* anaphors in a well-structured text.

Antecedent type	Distribution	Example
clause	37.9%	There is a controversial debate (**SBAR** *whether back school program might improve quality of life in back pain patients*). This study aimed to address **this issue**.
sentence	26.5%	(**S** *Reduced serotonin function and abnormalities in the hypothalamic-pituitary-adrenal axis are thought to play a role in the aetiology of major depression.*) We sought to examine **this issue** in the elderly ...
mixed	18.2%	(S (PP Given these data) (, ,) (**NP** *decreasing HTD to* $< or = 5$ *years*) (**VP** *may have a detrimental effect on patients with locally advanced prostate cancer*) (. .)) Only a randomized trial will conclusively clarify **this issue**.
nominalization	17.4%	As (**NP** *the influence of estrogen alone on breast cancer detection*) is not established, we examined **this issue** in the Women's Health Initiative trial...

Table 3.1: Antecedent types. In examples, the antecedent type is in **bold** and the marked antecedent is in *italics*.

Table 3.1 shows the distribution of the different linguistic forms that an antecedent of *this issue* can take. The majority of antecedents are clauses or whole sentences. A number of antecedents are noun phrases, but these are generally nominalizations that refer to abstract concepts (e.g., *the influence of estrogen alone on breast cancer detection*). Some antecedents are not even well-defined syntactic constituents[6] but are combinations of several well-defined constituents. We denote the type of such antecedents as *mixed*. In the corpus, 18.2% of the antecedents are of this type. Indeed, many of mixed-type antecedents (nearly three-quarters of them) are the result of parser attachment errors, but some are not.

In our data, we did not find any anaphoric chains for any of the *this-issue* anaphor instances. This observation supports the THIS-NPs hypothesis (Gundel et al., 1993; Poesio and Modjeska, 2002) that *this*-NPs are used to refer to entities which are *active* albeit not in *focus*, i.e., they are not the center of the previous utterance and thus are not referred to again and again by means

[6]We refer to every syntactic constituent identified by the parser as a *well-defined syntactic constituent*.

of anaphoric chains. That said, the lack of anaphoric chains in our case might be due to the nature of the data we use: the Medline abstracts.

3.3 Resolution

Now that we have reliably annotated data, we can train supervised machine learning models to resolve *this-issue* anaphora. Given a *this-issue* anaphor a_i, the problem of resolution consists of two steps: extracting the set of eligible candidates $C = \{c_1, c_2, ..., c_k\}$ and identifying the best candidate c_i from C that provides interpretation to the anaphor a_i. The following subsections describe each step in detail.

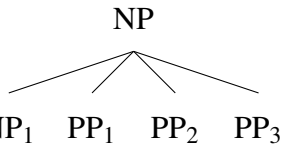
3.3.1 Candidate extraction

For correct resolution, the set of extracted candidates must contain the correct antecedent in the first place. In nominal-anaphora resolution, every noun phrase in the search span is considered as a candidate. But in shell content resolution, it is not enough to extract all noun phrases, because shell content is of many different types such as clauses, sentences, and nominalizations, as we noted in Table 3.1. We extract a set of suitable candidates corresponding to each anaphor instance as follows. First, we consider three sentences as a source of candidates: the anaphor sentence and the two preceding sentences. Second, we parse all these sentences using the Stanford Parser[7]. Third, we extract a set of candidates associated with every parse tree as follows. Initially the set of candidates contains all well-defined constituents from the parse tree. We require that the node type of the candidate be in the set {S, SBAR, NP, SQ, SBARQ, S+V}. In particular, we do not extract prepositional phrases and verb phrases as candidates because our data did not contain any antecedent with this syntactic type. Then we extract mixed-type candidates by concatenating the original constituent with its sister constituents. For example, in (34), the set of well-defined eligible candidate constituents is {NP, NP_1} and so NP_1 PP_1

[7] `http://nlp.stanford.edu/software/lex-parser.shtml`

and NP_1 PP_1 PP_2 are mixed type candidates.

(34)



Extracting mixed-type candidates serves two purposes: a) including the candidates which are not well-defined syntactic constituents (see Table 3.1), and b) rectifying parsing errors to include candidates that would not have been in a list of well-defined syntactic constituents given by the Stanford parser.

The set of candidate constituents is updated with the extracted mixed type constituents. Combining different well-defined candidates offers two advantages: a) it allows us to deal with mixed type instances; b) as a side effect, it also corrects some attachment errors made by the parser. The constituents having a number of leaves (words) less than a threshold[8] are discarded to give the set of candidate constituents associated with the given candidate sentence. This process is repeated for every candidate sentence to create a set of candidates per candidate sentence. The final set of candidates per anaphor instance is the union of these sets.

3.3.2 Features

As noted in Chapter 2, automatic resolution of shell nouns has not been attempted and hence we do not have a set of features that have been shown to work for this task. We explored the effect of including 43 automatically extracted features (12 feature classes), which are summarized in Table 3.2. The features were drawn from three sources: properties of shell noun antecedents as discussed in the linguistics literature, features used in resolution of anaphors with similar properties, i.e., *it*, *this*, and *that*, and our observations from annotation. Note that the process of feature extraction was rather flexible in that a feature was included when it seemed appropriate and useful for shell noun resolution, and hence it is possible for two features to be equivalent or dependent on each other. The next subsections describe various syntactic, semantic, lexical,

[8]The threshold 5 was empirically derived. Antecedents in our training data had on average 17 words.

Table 3.2: Feature sets for *this-issue* resolution. All features are extracted automatically.

ISSUE PATTERN (IP)	
ISWHETHER	1 iff the candidate follows the pattern `SBAR` → `(IN whether) (S ...)`
ISTHAT	1 iff the candidate follows the pattern `SBAR` → `(IN that) (S ...)`
ISIF	1 iff the candidate follows the pattern `SBAR` → `(IN iff) (S ...)`
ISQUESTION	1 iff the candidate node is `SBARQ` or `SQ`
SYNTACTIC TYPE (ST)	
ISNP	1 iff the candidate node is of type NP
ISS	1 iff the candidate node is a sentence node
ISSBAR	1 iff the candidate node is an SBAR node
ISSQ	1 iff the candidate node is an SQ or SBARQ node
MIXED	1 iff the candidate node is of type *mixed*
EMBEDDING LEVEL (EL) Müller (2008)	
TLEMBEDDING	level of embedding of the given candidate in its top clause (the root node of the syntactic tree)
ILEMBEDDING	level of embedding of the given candidate in its immediate clause (the closest parent of type S or SBAR)
MAIN CLAUSE (MC)	
MCLAUSE	1 iff the candidate is in the main clause
DISTANCE (D)	
ISSAME	1 iff the candidate is in the same sentence as anaphor
SADJA	1 iff the candidate is in the adjacent sentence
ISREM	1 iff the candidate occurs 2 or more sentences before the anaphor
POSITION	1 iff the antecedent occurs before anaphor
SEMANTIC ROLE LABELLING (SR)	
IVERB	1 iff the governing verb of the given candidate is an *issue* verb
ISA0	1 iff the candidate is the *agent* of the governing verb
ISA1	1 iff the candidate is the *patient* of the governing verb
ISA2	1 iff the candidate is the *instrument* of the governing verb
ISAM	1 iff the candidate plays the role of *modification*
ISNOR	1 iff the candidate plays no well-defined semantic role in the sentence
DEPENDENCY TREE (DT)	
IHEAD	1 iff the candidate head in the dependency tree is an issue word (e.g., *controversial*, *unknown*)
ISSUBJ	1 iff the dependency relation of the candidate to its head is of type *nominal, controlling* or *clausal subject*
ISOBJ	1 iff the dependency relation of the candidate to its head is of type *direct object* or *preposition obj*
ISDEP	1 iff the dependency relation of the candidate to its head is of type *dependent*
ISROOT	1 iff the candidate is the root of the dependency tree
ISPREP	1 iff the dependency relation of the candidate to its head is of type *preposition*
ISCONT	1 iff the dependency relation of the candidate to its head is of type *continuation*
ISCOMP	1 iff the dependency relation of the candidate to its head is of type *clausal* or *adjectival complement*
ISSENT	1 iff candidate's head is the root node
PRESENCE OF MODALS (M)	
MODAL	1 iff the given candidate contains a modal verb
PRESENCE OF SUBORDINATING CONJUNCTION (SC)	
ISCONT	1 iff the candidate starts with a contrastive subordinating conjunction (e.g., *however*, *but*, *yet*)
ISCAUSE	1 iff the candidate starts with a causal subordinating conjunction (e.g., *because*, *as*, *since*)
ISCOND	1 iff the candidate starts with a conditional subordinating conjunction (e.g., *if*, *that*, *whether or not*)
LEXICAL OVERLAP (LO)	
TOS	normalized ratio of the overlapping words in candidate and the title of the article
AOS	normalized ratio of the overlapping words in candidate and the anaphor sentence
DWS	proportion of domain-specific words in the candidate
CONTEXT (C)	
ISPPREP	1 iff the preceding word of the candidate is a preposition
ISFPREP	1 iff the following word of the candidate is a preposition
ISPPUNCT	1 iff the preceding word of the candidate is a punctuation
ISFPUNCT	1 iff the following word of the candidate is a punctuation
LENGTH (L)	
LEN	length of the candidate in words

and other features we used in resolution of *this-issue* anaphora.

3.3.2.1 Syntactic features

***Issue* pattern (IP)** Vendler (1968) points out that *wh*-question clause is common with the shell noun *issue*. An issue can take several semantic forms such as controversy (*X is controversial*), hypothesis (*It has been hypothesized X*), or lack of knowledge (*X is unknown*), where *X* is the *issue*. These semantic forms are generally expressed with certain syntactic patterns such as *whether X or not* and *that X*. Based on this observation, we include four *issue* pattern features: whether the candidate is a *wh*-clause, *that*-clause, if-clause, or a question clause.

Syntactic type (ST) Shell nouns act as placeholders or shells for complex pieces of information. There is a range of objects that could fit in these placeholders. For shell nouns such as *issue* this range is quite wide: it can incorporate objects with different syntactic shapes. That said, shell nouns have different syntactic preferences. For instance, Table 3.1 demonstrates that more than 60% of *this-issue* instances had clausal or sentential antecedents and verbal antecedents did not occur at all in the Medline abstracts. The ST features capture the syntactic preferences for the antecedents of *this issue*.

Embedding level (EL) Müller (2008) includes the EL feature in resolving *it*, *this*, and *that* in the spoken dialogue domain. The EL feature is a shallow approximation of the syntactic complexity of the candidate within the candidate sentence. Following Müller (2008), 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.

Main clause (MC) Navarretta (2011) notes that the antecedents of anaphors with abstract antecedents in Italian and Danish frequently occur in subordinate clauses and simple main clauses, and matrix clauses are very rare. Moreover, Gundel et al. (1993) and Poesio and Modjeska (2002) suggest that the antecedents of *this*-NP anaphors are not the center of the previous utterance. To capture these ideas, we include the MC feature, which checks whether the given candidate is part of the subordinating clause or the matrix clause.

3.3.2.2 Semantic and lexical features

Dependency tree (DT) The DT features examine the relation of the candidate to its head in the dependency tree given by the Stanford parser.

The feature IHEAD checks whether the candidate head in the dependency tree is an *issue* word. We extracted *issue* words as follows: first we started with a few seed words and then expanded this set using WordNet.[9] The *issue* words include: *speculate, theorize, theorise, conjecture, hypothesize, hypothesise, hypothecate, suppose, assume, presume, believe, propose, suggest, advise, look, appear, seem, think, believe, consider, conceive, analyze, analyse, study, examine, challenge, dispute, argue, reason, argue, contend, debate, fence, argue, indicate, establish, set_up, found, launch, prove, demonstrate, establish, show, shew, establish, found, plant, constitute, institute, test, prove, try, try_out, examine, restrict, restrain, trammel, limit, bound, confine, throttle, necessitate, ask, postulate, need, require, take, involve, call_for, demand, stay, remain, show, demo, exhibit, present, demonstrate, evaluate, pass_judgment, judge, uncertain, unsure, incertain, potential, possible, ill-defined, unknown, known, unclear, clear, demonstrate, evaluate, test, studied, establish, limited, need, demand, question, essential, controversy, controversial, inconclusive, unsettle, unsolved, unresolved, controversy, contention, contestation, disputation, disceptation, tilt, argument, arguing*.

Paice and Husk (1987) propose a list of prepositions that are indicators of referential usages for detecting non-referential *it*. Müller (2008) uses this as a feature in his *this*, *that*, and *it*

[9] `http://wordnet.princeton.edu/`

resolution system. The list includes the following prepositions: *among, before, beside, despite, from, in, near, of, onto, through, under, via, with, at, below, between, during, inside, off, outside, to, until, within, beneath, by, into, on, over, without*. The ISPREP features check whether the head of the candidate is a preposition from this list.

We also consider other dependency relations of the candidate to its head such as subject and direct object.

Semantic role (SR) The SR features capture the semantic role of the candidate with respect to its governing verb. We used the Illinois Semantic Role Labeler[10] for SR features.

The feature IVERB checks whether the governing verb of the candidate is an *issue* verb. Again we started with a set of few seed verbs and then expanded the set using WordNet. The *issue* verbs include: *speculate, theorize, theorise, conjecture, hypothesize, hypothesise, hypothecate, suppose, assume, presume, believe, propose, suggest, advise, look, appear, seem, think, believe, consider, conceive, analyze, analyse, study, examine, challenge, dispute, argue, reason, argue, contend, debate, fence, argue, indicate, establish, set_up, found, launch, prove, demonstrate, establish, show, determine, found, plant, constitute, institute, test, prove, try, try_out, examine, restrict, restrain, trammel, limit, bound, confine, throttle, necessitate, ask, postulate, need, require, take, involve, call_for, demand, stay, remain, show, demo, exhibit, present, demonstrate, evaluate, pass_judgment, judge*.

Modals (M) While annotating the data, we observed that issues often represent uncertainty which is typically denoted by using modal verbs, as shown in (35).

(35) Previous research indicated shared neurochemical substrates for gambling and psychostimulant reward. This suggests **that dopamine substrates may directly govern the reinforcement process in pathological gambling**. To investigate **this issue**, the present study assessed the effects of the relatively selective dopamine D2 antagonist...

[10] `http://cogcomp.cs.illinois.edu/page/software_view/SRL`

The M feature checks for the presence of modals in the given candidate.

Subordinating conjunction (SC) Passonneau (1994) provides a list of subordinating conjunctions and argues that being governed by one of these conjunctions is a necessary requirement for a clause to be independent. The conjunctions include: *after, albeit, although, as, because, before, ergo, forasmuch, how, if, inasmuch, lest, like, once, providing, since, so, then, though, till, til, until, unless, when, whence, whenever, where, whereas, whereat, whereby, wherefrom, whether, while, yet*. The SC features approximate this idea. We further divide these conjunctions into three groups: contrastive, causal, and conditional conjunctions and examine whether the candidate starts with conjunctions from the appropriate group.

Lexical overlap (LO) In nominal anaphora resolution, lexical overlap between the anaphor and the antecedents is one of the useful features. However in case of shell nouns, this feature is not directly applicable, as the anaphor and the antecedent rarely have overlapping words. We propose three lexical overlap features. First, during annotation we noted that issues in Medline abstract tend to spell out the motivation of the article which is also generally expressed in the title of the article. So we consider lexical overlap between the candidate and the title as one of the lexical overlap features. Second, although for anaphoric shell nouns the anaphor and the antecedent do not generally have overlapping words, the context of the anaphor, i.e., the anaphor sentence might share some words with the antecedent. The second lexical overlap feature encodes this information. Third, issues in Medline abstracts tend to include more domain-specific content words and exclude phrases with non-domain-specific words such as *it is well known that*. So the third feature encodes the proportion of domain-specific words in the given candidate. A list of domain-specific words was extracted based on *odds ratio* (OR) (Cornfield, 1951). For that, we compare how frequently a word occurs in Medline text compared to non-Medline text. We considered the NYT corpus as our non-Medline text. Each word *w* in the Medline abstracts gets an OR score based on equation 3.2. Domain-specific score for a candidate is the normalized score of domain-specific words in the candidate.

$$OR(w) = log\left(\frac{Pr(\text{domain = medline} \mid w)}{Pr(\text{domain} \neq \text{medline} \mid w)}\right) \quad (3.2)$$

$$= log\left(\frac{\text{Count(domain = medline and } w)}{\text{Count(domain} \neq \text{medline and } w)}\right) \quad (3.3)$$

3.3.2.3 Other features

Distance (D) The distance between the anaphor and the antecedent is a traditional feature in anaphora resolution. The intuition behind the distance feature is that, very often the distance between an anaphor and an antecedent is short to keep up with the pressure of strong textual coherence. For instance, in our annotated data, we observed that more than 95% of the instances had their antecedents in the same or immediately preceding sentences. The distance feature tries to capture this idea. We incorporate three binary distance features: whether the antecedent is in the same sentence as that of the anaphor, whether it is in the immediately adjacent sentence, and whether it is far away from the anaphor. In this class, we also include the feature *position* that encodes the position of the antecedent with respect to the anaphor.

Context (C) This set of features checks whether the candidate is preceded by a preposition or a punctuation mark.

Length (L) This feature encodes the length of the candidate in words. The intuition behind this feature is that the antecedents of *this issue* tend to be long: even if they are NPs, they tend to be long and complex NPs (e.g., *the influence of estrogen alone on breast cancer detection*)

3.3.2.4 Feature summary

We extract a number of features for *this-issue* anaphora resolution. Some of these features are derived empirically from the training data (e.g., ST, L, D). Others are based on ideas presented in the linguistics literature and our observations during annotation. Our long-term goal is to

generalize *this-issue* resolution to other shell nouns. So it is important to distinguish the features that are specific to the word *issue* (*issue*-specific features) and other features that might be relevant to other shell nouns (general abstract-anaphora features). The *Issue*-specific features make use of our common-sense knowledge of the concept of *issue* and the different semantic forms it can take; e.g., controversy (*X is controversial*), hypothesis (*It has been hypothesized X*), or lack of knowledge (*X is unknown*), where *X* is the *issue*. The *issue*-specific features include IVERB, IHEAD, and IP features. All other features are not particularly associated with the semantic properties of the word *issue*.

3.3.3 Candidate ranking model

We follow the candidate-ranking model proposed by Denis and Baldridge (2008). The advantage of the candidate-ranking model over the mention-pair model is that it overcomes the strong independence assumption made in mention-pair models and evaluates how good a candidate is relative to *all* other candidates.

We train our model as follows. If the anaphor is a *this-issue* anaphor, the set C is extracted using the candidate extraction algorithm from Section 3.3.1. Then a corresponding set of feature vectors, $C_f = \{c_{f1}, c_{f2}, ..., c_{fk}\}$, is created using the features in Table 3.2. For every anaphor a_i and eligible candidates $C_f = \{c_{f1}, c_{f2}, ..., c_{fk}\}$, we create training examples $(a_i, c_{fj}, label), \forall c_{fj} \in C_f$. The label is 1 if c_i is the true antecedent of the anaphor a_i, otherwise the label is -1. The examples with label 1 get the rank of 1, while other examples get the rank of 2. Note that the instance creation is simpler than for general coreference resolution because of the absence of anaphoric chains in our data. We use SVM^{rank} (Joachims, 2002) for training the candidate-ranking model. During testing, the trained model is used to rank the candidates of each test instance of *this-issue* anaphor.

3.4 Evaluation

Almost all current abstract anaphora resolution implementations report the resolution performance of the annotated antecedents in terms of the usual precision and recall. However, several reasons make it hard to compare the resolution results of these implementations: the variety of anaphoric expressions signalling abstract anaphora, the lack of common corpora, and different methods to represent abstract antecedents. That said, abstract antecedents across all domains share a common property: most of the time they are non-nominal. We argue that the usual precision and recall metric is rather a strict evaluation of non-nominal antecedents and that we need a more flexible way to evaluate such antecedents because often the boundaries of such antecedents are unclear and inclusion or exclusion of a few words or phrases does not make a big difference; the underlying meaning of the antecedent could still be the same. For instance, in example (36), inclusion of the phrase *a controversial debate* is unnecessary, but including it would not probably make a difference in the end application.

(36) There is a controversial debate **whether back school program might improve quality of life in back pain patients**. This study aimed to address **this issue**.

In this section, we present two evaluation metrics that we used for *this-issue* anaphora evaluation. We present our evaluation results of each stage of resolution. Finally, we discuss limitations of our current evaluation metrics.

3.4.1 Evaluation of candidate extraction

The set of candidate antecedents extracted by the method from Section 3.3.1 contained the correct antecedent 92% of the time. Each anaphor had, on average, 23.80 candidates, of which only 5.19 candidates were of nominal type. The accuracy dropped to 84% when we did not extract mixed type candidates. The error analysis of the 8% of the instances where we failed to extract the correct antecedent revealed that most of these errors were parsing errors which

could not be corrected by our candidate extraction method.[11] In these cases, the parts of the antecedent had been placed in completely different branches of the parse tree. For example, in (37), the correct antecedent is a combination of the NP from the $S \rightarrow VP \rightarrow NP \rightarrow PP \rightarrow$ **NP** branch and the PP from $S \rightarrow VP \rightarrow$ **PP** branch. In such a case, concatenating sister constituents does not help.

(37) The data from this pilot study (VP (VBP provide) (NP (NP no evidence) (PP (IN for) (NP **a difference in hemodynamic effects between pulse HVHF and CPFA**))) (PP **in patients with septic shock already receiving CRRT**)). A larger sample size is needed to adequately explore **this issue**.

3.4.2 Evaluation of *this-issue* resolution

We propose two metrics for *this-issue* anaphora evaluation. The simplest metric is the percentage of antecedents on which the system and the annotated gold data agree. We denote this metric as *EXACT-M* (Exact Match) and compute it as the ratio of number of correctly identified antecedents to the total number of marked antecedents. This metric is a good indicator of a system's performance; however, it is a rather strict evaluation because, as we noted in section 1, issues generally have no precise boundaries in the text. So we propose another metric called RLL, which is similar to the ROUGE-L metric (Lin, 2004) used for the evaluation of automatic summarization. Let the marked antecedents of the gold corpus for k anaphor instances be $G = \langle g_1, g_2, ..., g_k \rangle$ and the system-annotated antecedents be $A = \langle a_1, a_2, ..., a_k \rangle$. Let the number of words in G and A be m and n respectively. Let $LCS(g_i, a_i)$ be the number of words in the longest common subsequence of g_i and a_i. In our context, a string X is a subsequence of Y if it is a sequence of words that are not necessarily contiguous but are nevertheless taken in order from Y. For example (39) is a subsequence of (38).

(38) I seldom arrive in Paris, where work takes me four or five times a year, without some feeling of being an ugly duckling or, at any rate, a small-town person.

[11] Extracting candidate constituents from the dependency trees did not add any new candidates to the set of candidates.

(39) I seldom arrive in Paris without some feeling of being an ugly duckling or a small-town person.

Then the precision (P_{RLL}) and recall (R_{RLL}) over the whole data set are computed as shown in equations (2) and (3). If the system picks too much text for antecedents, R_{RLL} is high but P_{RLL} is low. The F-score, F_{RLL}, combines these two scores.

$$P_{RLL} = \frac{1}{n}\sum_{i=1}^{k} LCS(g_i, a_i) \tag{3.4}$$

$$R_{RLL} = \frac{1}{m}\sum_{i=1}^{k} LCS(g_i, a_i) \tag{3.5}$$

$$F_{RLL} = \frac{2 \times P_{RLL} \times R_{RLL}}{P_{RLL} + R_{RLL}} \tag{3.6}$$

The lower bound of F_{RLL} is 0, where no true antecedent has any common subsequence with the predicted antecedents and the upper bound is 1, where all the predicted and true antecedents are exactly the same. In our results we represent these scores in terms of percentages.

There are no implemented systems that resolve *issue* anaphora or abstract anaphora signalled by shell nouns in arbitrary text to use as a comparison. So we compare our results against two baselines: *adjacent sentence* and *random*. The adjacent sentence baseline chooses the previous sentence as the correct antecedent. This is a high baseline because in our data 84.1% of the antecedents lie within the adjacent sentence. The random baseline chooses a candidate drawn from a uniform random distribution over the set of candidates.[12]

We carried out two sets of systematic experiments in which we considered all combinations of our twelve feature classes. The first set consists of 5-fold cross-validation experiments on our training data. The second set evaluates how well the model built on the training data works on the unseen test data.

Table 3.3 gives results of our system. The first two rows are the baseline results. Rows 3 to 8 give results for some of the best performing feature sets. All systems based on our features

[12]Note that our F_{RLL} scores for both baselines are rather high because candidates often have considerable overlap with one another; hence a wrong choice may still have a high F_{RLL} score.

Table 3.3: *this-issue* resolution results with SVM^{rank}. *All* means evaluation using all features. *Issue*-specific features = {IP, IVERB, IHEAD}. EX is EXACT-M. Boldface is best in column.

		5-fold Cross-Validation				Test			
		P_{RLL}	R_{RLL}	F_{RLL}	EX	P_{RLL}	R_{RLL}	F_{RLL}	EX
1	Adjacent sentence	66.5	86.2	74.9	22.9	61.7	87.7	72.5	24.0
2	Random	50.7	32.8	39.6	8.4	43.6	35.0	38.9	15.7
3	{IP, D, C, LO, EL, M, MC, L, SC, SR, DT}	79.4	83.7	81.1	59.8	71.9	85.7	78.2	58.8
4	{IP, D, C, LO, M, MC, L, SC, DT}	78.7	83.9	81.1	59.9	70.6	88.1	78.4	54.9
5	{IP, D, C, EL, L, SC, SR, DT}	78.0	83.1	80.3	57.4	72.0	84.9	77.9	**60.8**
6	{IP, D, EL, MC, L, SR, DT}	80.0	84.8	**82.2**	**59.9**	68.9	85.3	76.2	56.9
7	{IP, D, M, L, SR}	73.4	83.2	77.9	52.3	70.7	91.0	**79.6**	51.0
8	{D, C, LO, L, SC, SR, DT}	79.2	85.3	82.0	56.1	67.4	86.3	75.7	52.9
9	*issue*-specific features	74.7	45.7	56.6	41.4	64.2	45.9	53.5	41.4
10	non-*issue* features	76.4	79.4	77.8	51.5	71.2	83.2	76.8	58.8
11	All	78.2	82.9	80.4	56.8	71.3	83.2	76.8	56.9
12	Oracle candidate extractor + row 3	79.6	82.3	80.7	58.3	74.7	87.1	80.4	**64.7**
13	Oracle candidate sentence extractor + row 3	86.7	92.1	**89.3**	**63.7**	79.7	91.5	**85.2**	62.0

beat both baselines on F-scores and EXACT-M. The empirically derived feature sets IP (issue patterns) and D (distance) appeared in almost all best feature set combinations. Removing D resulted in a 6 percentage points drop in F_{RLL} and a 4 percentage points drop in EXACT-M scores. Surprisingly, feature set ST (syntactic type) was not included in most of the best performing set of feature sets. The combination of syntactic and semantic feature sets {IP, D, EL, MC, L, SR, DT} gave the best F_{RLL} and EXACT-M scores for the cross-validation experiments. For the test-data experiments, the combination of semantic and lexical features {D, C, LO, L, SC, SR, DT} gave the best F_{RLL} results, whereas syntactic, discourse, and semantic features {IP, D, C, EL, L, SC, SR, DT} gave the best EXACT-M results. Overall, row 3 of the table gives reasonable results for both cross-validation and test-data experiments with no statistically significant difference to the corresponding best EXACT-M scores in rows 6 and 5 respectively.[13] To pinpoint the errors made by our system, we carried out three experiments. In the first experiment, we examined the contribution of *issue*-specific features versus non-*issue* features (rows 9 and 10). Interestingly, when we used only non-*issue* features, the

[13] We performed a simple one-tailed, k-fold cross-validated paired t-test at significance level $p = 0.05$ to determine whether the difference between the EXACT-M scores of two feature classes is statistically significant.

performance dropped only slightly. The F_{RLL} results from using only *issue*-specific features were below baseline, suggesting that the features that are not directly associated with the word *issue* play a crucial role in resolving *this-issue* anaphora.

In the second experiment, we determined the error caused by the candidate extractor component of our system. Row 12 of the table gives the result when an oracle candidate extractor was used to add the correct antecedent in the set of candidates whenever our candidate extractor failed. This did not affect cross-validation results by much because of the rarity of such instances. However, in the test-data experiment, the EXACT-M improvements that resulted were statistically significant. This shows that our resolution algorithm was able to identify antecedents that were arbitrary spans of text.

In the last experiment, we examined the effect of the reduction of the candidate search space. We assumed an oracle candidate sentence extractor (Row 13) which knows the exact candidate sentence in which the antecedent lies. We can see that both RLL and EXACT-M scores markedly improved in this setting. In response to these results, we trained a decision-tree classifier to identify the correct antecedent sentence with simple location and length features and achieved 95% accuracy in identifying the correct candidate sentence.

3.5 Discussion

This chapter reports analysis of the narrow problem of resolution of *this-issue* anaphora in the medical domain to get a good grasp of the general shell noun resolution problem. In particular, it described in detail the methodology of annotating and resolving *this issue* in the Medline abstracts. The inter-annotator agreement in terms of Krippendorff's unitizing α of 0.86 and the resolution results as high as 60% in terms of EXACT-M, i.e., accuracy, and about 82% in terms of F_{RLL} illustrate the feasibility of annotating and resolving shell nouns automatically, at least in a closed domain of Medline abstracts.

The results of *this-issue* resolution show that reduction of search space markedly improves

the resolution performance, suggesting that a two-stage process that first identifies the broad region of the antecedent and then pinpoints the exact antecedent might work better than the current single-stage approach. The rationale behind this two-stage process is twofold. First, the main challenge in dealing with non-nominal anaphora is that the search space of candidate antecedents is quite large and the problem of spurious antecedents is quite severe.[14] And second, it is possible to reduce the search space and accurately identify the broad region of the antecedents using simple features such as the location of the anaphor in the anaphor sentence (e.g., if the anaphor occurs at the beginning of the sentence, the antecedent is most likely present in the previous sentence).

Certainly, the approach presented in this chapter needs further development to make it useful. One broad goal is to resolve shell nouns with a variety of shell nouns and for different kinds of text. At present, the major obstacle is that there is very little annotated data available that could be used to train a machine learning system to resolve shell nouns. The next chapter explains how we overcome these challenges and presents an approach that tackles a variety of shell nouns in a broader domain.

[14] If we consider all well-defined syntactic constituents of a sentence as issue candidates, in our data, a sentence has on average 43.61 candidates. Combinations of several well-defined syntactic constituents only add to this number. Hence if we consider the antecedent candidates from the previous 2 or 3 sentences, the search space can become quite large and noisy.

Chapter 4

Resolving Cataphoric Shell Nouns

4.1 Introduction

In Section 2.1.2.1 (p. 17), we noted that shell nouns occur fairly frequently with cataphoric lexico-syntactic patterns. According to Schmid, the presence of these patterns suggest that the shell content occurs in the same sentence as a postnominal or a complement clause. We refer to such instances of shell nouns as cataphoric shell nouns (CSNs).[1] This chapter presents a general approach to resolve CSNs.[2]

Recall that shell nouns take different types of one or more semantic arguments, and the problem of shell noun resolution is identifying the appropriate semantic argument that is the shell content of the shell noun in the given context. For CSNs, the shell content typically occurs as the syntactic argument of the shell noun. For instance, in examples (40) and (41), the shell nouns are resolved to the postnominal *that* clause and the complement *that* clause, respectively.[3]

[1]Recall that the phenomenon of cataphoric shell nouns is similar to the phenomenon of cataphora in that both have forward-looking antecedent or shell content. That said, the major difference between the two is that in case of cataphora, the antecedent is not specified by the sentence's syntax; such structures are common in case of CSNs. To avoid confusion between the well-known concept of cataphora and the cataphora-like phenomenon that shell nouns exhibit, we use the term *cataphoric shell nouns*, i.e., CSNs.

[2]This work is presented in Kolhatkar and Hirst (2014).

[3]Recall that the postnominal *that* clause in (40) is not a relative clause: the fact in question is not an argument of *exploit and repackage*.

(40) **The fact** [**that a major label hadn't been at liberty to exploit and repackage the material on CD**]$^{\text{general factual content}}$ meant that prices on the vintage LP market were soaring.

(41) Although there are many technical objections, **the usual reason** [why courts have rejected DNA tests that seem to show guilt]$^{\text{effect}}$ is [**that scientists disagree about how to calculate the odds that there is a match between cells from a suspect and cells from a crime scene**]$^{\text{cause}}$.

Although resolving examples such as (40) and (41) seems straightforward using syntactic structure alone, the relation between a CSN and its content is in many crucial respects a semantic phenomenon. For instance, resolving the shell noun phrase *the usual reason* to its shell content in (41) involves identifying a) that *reason* generally expects two semantic arguments: cause and effect, b) that the cause argument (and not the effect argument) represents the shell content, and c) that a particular constituent in the given context is the cause argument.

To obtain the semantic knowledge required to resolve CSNs, I exploit Schmid's semantic classification and semantic families from Section 2.1.2. In Section 4.2 (p. 80), I point out the difficulties associated with resolving CSNs. Section 4.3 describes a general method to resolve CSNs. Section 4.4 describes how we gathered the evaluation data using crowdsourcing. Section 4.5 shows the comparison between the baseline and our method. Finally, Section 4.6 discusses the successes and failures of our algorithm and demonstrates how far one can get with simple, deterministic shell content extraction, and to what extent knowledge derived from the linguistic literature can be useful to resolve CSNs.

4.2 Challenges

A number of challenges are associated with the task of resolving cataphoric shell noun examples, especially when it comes to developing a holistic approach for a variety of shell nouns.

First, each shell noun has idiosyncrasies. Different shell nouns have different semantic and syntactic expectations, and hence they take different types of one or more semantic arguments: one introducing the shell content, and others expressing circumstantial information about the

shell noun. For instance, *fact* typically takes a single factual clause as an argument, which is its shell content, as we saw in example (40), whereas *reason* expects two arguments: the cause and the effect, with the content introduced in the cause, as we saw in example (41). Similarly, *decision* takes an agent making the decision and the shell content is represented as an action or a proposition, as shown in (42). Recall that this aspect of shell nouns of taking different numbers and kinds of complement clauses is similar to verbs having different subcategorization frames.

(42) I applaud loudly **the decision** of [Greenburgh]$^{\text{agent}}$ [**to ban animal performances**]$^{\text{action}}$.

Second, at the conceptual level, once we know which semantic argument represents shell content, resolving examples such as (41) seems straightforward using syntactic structure, i.e., by extracting the complement clause. But at the implementation level, this is a non-trivial problem for two reasons. The first reason is that examples containing shell nouns often follow syntactically complex constructions, including embedded clauses, coordination, and sentential complements. An automatic parser is not always accurate for such examples. So the challenge is whether the available tools in computational linguistics such as syntactic parsers and discourse parsers are able to provide us with the information that is necessary to resolve these difficult cases. The second reason is that the shell content can occur in many different constructions, such as apposition (e.g., ***parental ownership of children***, ***a concept*** *that allows* ...), postnominal and complement clause constructions, as we saw in examples (40) and (41), and modifier constructions (e.g., ***the liberal trade policy*** *that* ...). Moreover, in some constructions, the content is indefinite (e.g., ***A bad idea*** *does not harm until someone acts upon it.*) or *None* because the example is a non-shell noun usage (e.g., ***this week's issue*** *of Sports Illustrated*), and the challenge is to identify such cases.

Finally, whether the postnominal clause introduces the shell content or not is dependent on the context of the shell noun phrase. The resolution can be complicated by complex syntactic constructions. For instance, when the shell noun follows verbs such as *expect*, it becomes difficult for an automatic system to identify whether the postnominal or the complement clause is of the verb or of the shell noun (e.g., *they did not expect* ***the decision*** *to reignite tension*

in Crown Heights vs. *no one expected* ***the decision to call an election***). Similarly, shell noun phrases can be objects of prepositions, and whether the postnominal clause introduces the shell content or not is dependent on this preposition. For instance, for the pattern *reason that*, the postnominal *that* clause does not generally introduce the shell content, as we saw in (41); however, when the shell noun phrase containing *reason* follows the preposition *for*, the cause argument, i.e., the shell content, is typically introduced in the postnominal clause. An example is shown in (43).

(43) Low tax rates give people an incentive to work, for **the simple reason that they get to keep more of what they earn**.

4.3 Resolution algorithm

This section describes an algorithm to resolve CSNs. The algorithm addresses the primary challenge of idiosyncrasies of shell nouns by exploiting Schmid's semantic families (see Section 2.1.3.2, p. 27). The input of the algorithm is a CSN instance, and the output is its shell content or *None* if the shell content is not present in the given sentence. The algorithm follows four steps. First, we parse the given sentence using the Stanford parser as in Chapter 3. Second, we look for the noun phrase (NP), where the head of the NP is the shell noun to be resolved.[4] Third, we identify whether the shell content occurs in the given sentence or not, as described in Section 4.3.1. Finally, we extract the appropriate postnominal or complement clause as directed by Schmid's semantic families, as described in Section 4.3.2.

4.3.1 Identifying potentially anaphoric shell-noun constructions

Before starting the actual resolution, first we identify whether the shell content occurs in the given sentence or not. According to Schmid, the lexico-syntactic patterns signal the position of the shell content. For instance, if the pattern is of the form *N-be-clause*, the shell content is

[4]We extract the head of an NP following the heuristics proposed by Collins (1999, p. 238).

more likely to occur in the complement clause in the same sentence. That said, although on the surface level, the shell noun seems to follow a cataphoric pattern, it is possible that the shell content is not given in a postnominal or a complement clause, as shown in (44).

(44) Just as weekend hackers flock to the golf ball most used by PGA Tour players, **recreational skiers, and a legion of youth league racers, gravitate to the skis worn by Olympic champions**. It is **the reason** that top racers are so quick [to][5] flash their skis for the cameras in the finish area.

Here, the shell noun and its content are linked via the pronoun *it*. For such constructions, the shell noun phrase and shell content do not occur in the same sentence. Shell content occurs in the preceding discourse, typically in the preceding sentence. We identify such cases, and other cases where the shell content is not likely to occur in the postnominal or complements clauses, by looking for the patterns below in the given order, returning the shell content when it occurs in the given sentence.

Sub-be-N This pattern corresponds to the lexico-grammatical pattern in Figure 4.1(a). If this pattern is found, there are three main possibilities for the *subject*. First, if an existential *there* occurs at the subject position, we move to the next pattern. Second, if the subject is *it* (example (44)), *this* or *that*, we return *None*, assuming that the content is not present in the given sentence. Finally, if the first two conditions are not satisfied, i.e., if the subject is neither a pronoun nor an existential *there*, we assume that subject contains a valid shell content, and return it. An example is shown in (45). Note that in such cases, unlike other patterns, the shell content is expressed as a noun phrase.

(45) **Strict liability** is **the biggest issue** when considering what athletes put in their bodies.

Apposition Another case where shell content does not typically occur in the postnominal or complement clause is the case of apposition. Indefinite shell noun phrases often occur in apposition constructions, as shown in (46).

[5]The word is missing in the NYT corpus.

(46) The LH lineup, according to Gale, will feature **"cab-forward" design**, **a concept** that particularly pleases him.

In this step, we check for this construction and return the sentential, verbal, or nominal left sibling of the shell noun phrase.

Modifier For shell nouns such as *issue*, *phenomenon*, and *policy*, often the shell content is given in the modifier of the shell noun, as shown in (47).

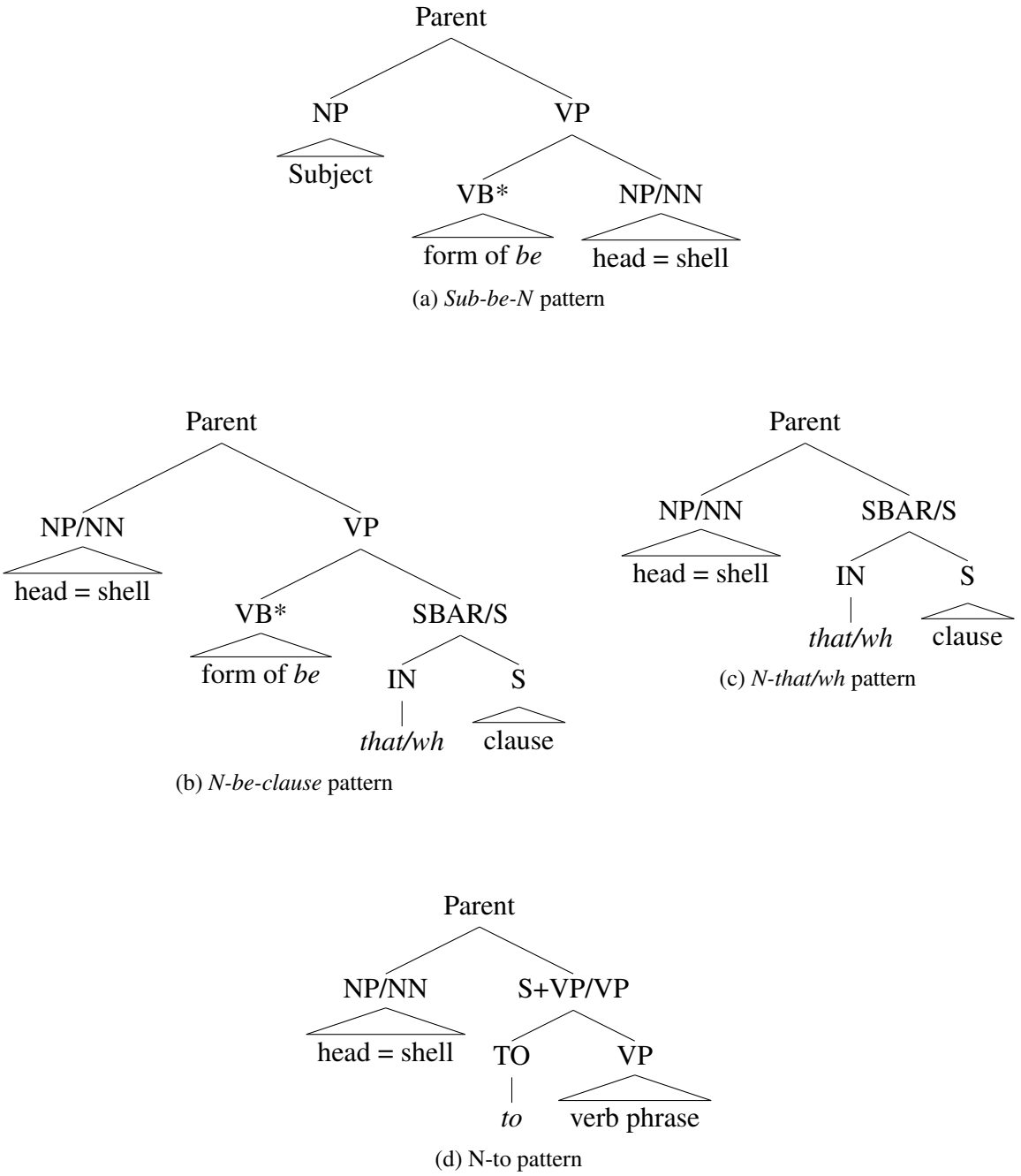
(47) But in the 18th century, Leipzig's central location in German-speaking Europe and **the liberal trade policy** of the Saxon court fostered publishing.

We deal with such cases as follows. First, we extract the modifier phrases by concatenating the modifier words having noun, verb, or adjective part-of-speech tags. To exclude unlikely modifier phrases as shell content (e.g., *good idea*, *big issue*), we extract a list of modifiers for a number of shell nouns and create a stoplist of modifiers. If any of the words in the modifier phrases is a pronoun or occurs in the stoplist, we move to the next pattern. If the modifier phrase passes the stoplist test, to distinguish between non-shell content and shell content modifiers, we examine the hypernym paths of the words in the modifier phrase in WordNet (Fellbaum, 1998). If the synset *abstraction.n.06* occurs in the path, we consider the modifier phrase to be valid shell content, assuming that the shell content of shell nouns most typically represents an abstract entity.

4.3.2 Resolving remaining instances

At this stage we are assuming that the shell content occurs either in the postnominal clause or the complement clause. So we look for the patterns below, returning the shell content when found.

N-be-clause The lexico-grammatical pattern corresponding to the pattern *N-be-clause* is shown in Figure 4.1(b). This is one of the more reliable patterns for shell content extraction, as the



(a) *Sub-be-N* pattern

(b) *N-be-clause* pattern

(c) *N-that/wh* pattern

(d) N-to pattern

Figure 4.1: Lexico-syntactic patterns for shell nouns

be verb suggests the semantic identity between the shell noun and the complement clause. The *be*-verb does not necessarily have to immediately follow the shell noun. For instance, in example (48), the head of the NP *The issue that this country and Congress must address* is the shell

noun *issue*, and hence it satisfies the construction in Figure 4.1(b).

(48) **The issue** that this country and Congress must address is **how to provide optimal care for all without limiting access for the many**.

N-clause Finally, we look for this pattern. An example of this pattern is shown in Figure 4.1(c). This is the most common (see Table 2.2, p. 20) and trickiest pattern in terms of resolution, and whether the shell content is given in the postnominal clause or not is dependent on the properties of the shell noun under consideration and the syntactic construction of the example. For instance, for the shell noun *decision*, the postnominal *to*-infinitive clause typically represents shell content. But this did not hold for the shell noun *reason*, as shown in (49).

(49) **The reason** to resist becoming a participant is obvious.

Here, Schmid's semantic families come in the picture. We wanted to examine a) the extent to which the previous steps help in resolution, and b) whether knowledge extracted from Schmid's families add value to the resolution. So we employ two versions of this step.

Include Schmid's cues (+SC) This version exploits the knowledge encoded in Schmid's semantic families (Section 2.1.3.2, p. 27), and extracts postnominal clauses only if Schmid's pattern cues are satisfied. In particular, given a shell noun, we determine the families in which it occurs and list all possible patterns of these families as shell content cues. The postnominal clause is a valid shell content only if it satisfies these cues. For instance, the shell noun *reason* occurs in only one family: *Reason*, with the allowed shell content patterns *N-that* and *N-why*. Schmid's patterns suggest that the postnominal *to*-infinitive clauses are not allowed as shell content for this shell noun, and thus this step will return *None*. This version helps correctly resolving examples such as (49) to *None*.

Exclude Schmid's cues (–SC) This version does not enforce Schmid's cues in extracting the postnominal clauses. For instance, the *Problem* family does not include *N-that/wh/to/of*

Table 4.1: Shell nouns and the semantic families in which they occur.

Shell noun	Families
idea	*Idea*, *Plan*, *View*, *Aim*, *Belief*, *Purpose*
problem, *difficulty*, *trouble*	*Problem*, *Trouble*
fact	*Thing*, *Certainty*
issue, *concept*	*Idea*
decision, *plan*, *policy*	*Plan*
reason	*Reason*
phenomenon	*Thing*

patterns, but in this condition, we nonetheless allow these patterns in extracting the shell content of the nouns from this family.

4.4 Evaluation data

We claim that our algorithm is able to resolve a variety of shell nouns. That said, creating evaluation data for all of Schmid's 670 English shell nouns was infeasible in the given time and money constraints. So we create a sample of representative evaluation data to examine how well the algorithm works

a) on a variety of shell nouns,

b) for shell nouns within a semantic family,

c) for shell nouns across families with different semantic and syntactic expectations, and

d) for a variety of shell noun patterns from Table 2.1 (p. 18).

4.4.1 Selection of nouns

Recall that each shell noun has its idiosyncrasies. So in order to evaluate whether our algorithm is able to address these idiosyncrasies, the evaluation data must contain a variety of shell nouns with different semantic and syntactic expectations. To examine a), we consider the six families shown in Table 2.5 (p. 28). These families span three abstract categories: *mental*, *eventive*, and

Table 4.2: Semantic families of the twelve selected shell nouns.

***Idea* family**	***Plan* family**
Semantic features: [mental], [conceptual] **Frame**: mental; focus on propositional content of IDEA **Nouns**: *idea*, *issue*, *concept*, *point*, *notion*, *theory*, *thesis*, *position*, *hypothesis*, ... **Patterns**: *N-be-that/of*, *N-that/of*	**Semantic features**: [mental], [volitional], [manner] **Frame**: mental; focus on IDEA **Nouns**: *decision*, *plan*, *policy*, *idea*, *strategy*, *principle*, *rationale*, ... **Patterns**: *N-be-to/that*, *N-to/that*
***View* family**	***Aim* family**
Semantic features: [mental], [creditive], [attitudinal] **Frame**: mental; focus on psychological state **Nouns**: *idea*, *view*, *notion*, *line*, *opinion*, *conviction*, *experience*, ... **Patterns**: *N-that*, *N-be-that*	**Semantic features**: [mental], [volitional], [conclusive] **Frame**: mental; focus on psychological state **Nouns**: *point*, *idea*, *hope*, *aim*, *goal*, *ambition*, *interest*, *objective*, ... **Patterns**: *N-to*, *N-be-to*
***Belief* family**	***Purpose* family**
Semantic features: [mental], [creditive] **Frame**: mental; focus on psychological state **Nouns**: *idea*, *belief*, *hope*, *feeling*, *impression*, *speculation*, *knowledge*, ... **Patterns**: *N-that/of*, *N-be-that/of*	**Semantic features**: [mental], [volitional], [detached] **Frame**: mental; focus on psychological state **Nouns**: *idea*, *purpose*, *function* **Patterns**: *N-be-to*, *N-be-to*
***Trouble* family**	***Problem* family**
Semantic features: [eventive], [attitudinal], [manner], [deontic] **Frame**: general eventive **Nouns**: *problem*, *trouble*, *difficulty*, *dilemma*, *snag* **Patterns**: *N-be-to*	**Semantic features**: [factual], [attitudinal], [impeding] **Frame**: general factual **Nouns**: *problem*, *trouble*, *difficulty*, *point*, ... **Patterns**: *N-be-that/of*
***Thing* family**	***Certainty* family**
Semantic features: [factual] **Frame**: general factual **Nouns**: *fact*, *phenomenon*, *point*, *case*, *thing*, *business*... **Patterns**: *N-that*, *N-be-that*	**Semantic features**: [modal], [epistemic], [necessary] **Frame**: epistemic modality **Nouns**: *fact*, *truth*, *reality*, *certainty*, ... **Patterns**: *N-that*, *N-be-that*

***Reason* family**
Semantic features: [factual], [causal] **Frame**: causal; attentional focus on CAUSE **Nouns**: *reason*, *cause*, *ground*, *thing* **Patterns**: *N-be-that/why*, *N-that/why*

```
be_pat = (r" ((‘s)|(is)|(has_VB[A-Z]* been)|(are)|(was)|(were)|
              (will_MD be)|(would_MD be)|
              (would_MD have_VB[A-Z]* been))_((VB[A-Z]*)|(MD)) ")
wh_pat = (r’ ((whether)|(what)|(when)|(where)|(which)|
             (who)|(whom)|(why)|(how))_’
shell = shell + ’_NN’
pats = ({ ’_to_pat’: shell + r’ to_TO ’,
          ’_be_to_pat’: shell + be_pat + r’to_TO ’,
          ’_that_pat’: shell + r’ that_IN ’,
          ’_be_that_pat’: shell + be_pat+ r’that_IN ’,
          ’_wh_pat’: shell + wh_pat,
          ’_be_wh_pat’: shell + be_pat + wh_pat.lstrip(),
          ’_of_pat’: shell + r’ of_IN ’
        })
```

Figure 4.2: Python regular expressions used in extracting CSN instances.

factual, and five distinct groups: *conceptual*, *volitional*, *factual*, *causal*, and *attitudinal*. Also, the families have considerably different syntactic expectations. For instance, the nouns in the *Idea* family can have their content in *that* or *of* clauses occurring in *N-clause* or *N-be-clause* constructions, whereas the *Trouble* and *Problem* families do not allow *N-clause* pattern. The shell content of the nouns in the *Plan* family is generally represented with *to* infinitives. To examine b) and c), we choose three nouns from each of the first four families from Table 2.5 (p. 28), i.e., *idea*, *issue*, *concept*, *decision*, *plan*, *policy*, *problem*, *trouble*, and *difficulty*. To add diversity, we also include two shell nouns from the *Thing* family and a shell noun from the *Reason* family, i.e., *fact*, *phenomenon*, and reason. So we selected total 12 shell nouns for evaluation: *idea*, *issue*, *concept*, *decision*, *plan*, *policy*, *problem*, *trouble*, *difficulty*, *reason*, *fact*, and *phenomenon*. Recall that a shell noun can occur in more than one semantic family having distinct pattern preferences. Table 4.1 shows the twelve shell nouns and the families in which they occur, and Table 4.2 shows the detailed description of these families, including their pattern cues.

4.4.2 Selection of instances

Recall that the shell content varies based on the shell noun and the pattern it follows. Moreover, shell nouns have pattern preferences, as we saw in Table 2.2 (p. 20). To examine d), i.e., how well our algorithm works for a variety of shell noun patterns, we need shell noun examples following different patterns from Table 2.1 (p. 18). We consider the New York Times corpus as our base corpus, and from this corpus extract all sentences following the cataphoric lexico-grammatical patterns from Table 2.1 (p. 18) for the twelve selected shell nouns. We considered part-of-speech information[6] while looking for the patterns. For instance, instead of the pattern *N-that*, we actually looked for {shell_noun_*NN that_IN*}. In other words, for *N-that* pattern, we only consider instances when *that* is a subordinating conjunction and discard instances when *that* is used as a relative pronoun. Figure 4.2 shows the regular expressions used to extract CSN examples corresponding to the shell noun #shell.

Then we arbitrarily pick 100 examples for each shell noun. In particular, for each shell noun the 100 examples include 10 examples of each of the seven cataphoric patterns from Table 2.1 (p. 18). The remaining 30 examples are picked randomly from all the cataphoric occurrences of that shell noun. As a result, among these 30 examples, the most dominant pattern for that shell noun will normally have a greater representation than the other patterns.

4.4.3 Crowdsourcing annotation

We wanted to examine to what extent non-expert native speakers of English with minimal annotation guidelines would agree on shell content of CSNs. We explored the possibility of using *crowdsourcing*, which is an effective way to obtain annotations for natural language research (Snow et al., 2008). There has been some prior effort to annotate anaphora and coreference using *Games with a Purpose* as a method of crowdsourcing (Chamberlain et al., 2009; Hladká et al., 2009) and it has been shown that crowdsourced data can successfully be used as training data for NLP tasks (Hsueh et al., 2009).

[6] `http://nlp.stanford.edu/software/tagger.shtml`

With the instances of the twelve selected shell nouns above, we designed a crowdsourcing experiment to obtain the annotated data for evaluation. We parse each sentence using the Stanford parser, and extract all possible candidates, i.e., syntactic arguments of the shell noun from the parser's output. Since our examples include embedding clauses and sentential complements, often the parser is inaccurate. For instance, in example (50), the parser attaches only the first clause of the coordination (*that people were misled*) to the shell noun phrase *the fact*.

(50) **The fact that people were misled and information was denied**, that's the reason that you'd wind up suing.

To deal with such parsing errors, we consider the 30-best parses given by the parser. From these parses, we extract a list of eligible candidates. This list includes the arguments of the shell noun given in the appositional clauses, modifier phrases, postnominal *that*, *wh*, or *to* infinitive clauses, complement clauses, objects of postnominal prepositions of the shell noun, and subject if the shell noun follows *subject-be-N* construction. On average, there were three unique candidates per instance.

After extracting the candidates, we present the annotators with the sentence, with the shell noun highlighted, and the extracted candidates. We ask the annotators to choose the option that provides the correct interpretation of the highlighted shell noun. We also provide them the option *None of the above*, and ask them to select it if the shell content is not present in the given sentence or the shell content is not listed in the list of candidates.

CrowdFlower We used CrowdFlower[7] as our crowdsourcing platform, which in turn uses various worker channels such as Amazon Mechanical Turk[8]. CrowdFlower offers a number of features. First, it provides a *quiz* mode which facilitates filtering out spammers by requiring an annotator to pass a certain number of test questions before starting the real annotation. Second, during annotation, it randomly presents test questions with known answers to the annotators to keep them on their toes. Based on annotators' responses to these questions, each

[7] http://crowdflower.com/

[8] https://www.mturk.com/mturk/welcome

Table 4.3: Annotator agreement on shell content. Each column shows the percentage of instances on which at least *n* or fewer than *n* annotators agree on a single answer.

	≥ 5	≥ 4	≥ 3	< 3
idea	53	67	95	5
issue	44	65	95	5
concept	40	56	96	4
decision	50	72	98	2
plan	41	55	95	5
policy	42	61	94	6
problem	52	70	100	0
trouble	44	69	99	1
difficulty	45	61	96	4
reason	48	60	93	7
fact	52	68	98	2
phenomenon	39	56	95	5
all	46	63	96	4

annotator is assigned a trust score: an annotator performing well on the test questions gets a high score. CrowdFlower later uses these trust scores as weights when computing the majority vote. Finally, CrowdFlower allows the user to select the permitted demographic areas and skills required.

Settings We asked for at least 5 annotations per instance by annotators from the English-speaking countries. The evaluation task contained a total of 1200 instances, 100 instances per shell noun. To maintain the annotation quality, we included 105 test questions, distributed among different answers. We paid 2.5 cents per instance and the annotation task was completed in less than 24 hours. The annotation guidelines are given in Appendix E.

Results Table 4.3 shows the agreement of the crowd on instances of different shell nouns. In most cases, at least 3 out of 5 annotators agreed on a single answer. We took this answer as the gold standard in our evaluation, and discard the instances where fewer than three annotators agreed. The option *None of the above* was annotated for about 30% of time. We include these cases in the evaluation, as we wanted to examine to what extent our algorithm is able

Table 4.4: Shell noun resolution results. LSC = lexico-syntactic clause baseline. Each column shows the percent accuracy of resolution using the corresponding method. Boldface indicates best in row.

	Nouns	LSC	A–SC	A+SC
1	*idea*	74	82	**83**
2	*issue*	60	75	**77**
3	*concept*	51	67	**68**
4	*decision*	70	71	**73**
5	*plan*	51	**63**	62
6	*policy*	58	**70**	52
7	*problem*	66	**69**	59
8	*trouble*	63	**68**	50
9	*difficulty*	68	**75**	49
10	*reason*	43	53	**77**
11	*fact*	43	55	**68**
12	*phenomenon*	33	**62**	50
13	*all*	57	**69**	64

to successfully predict if the shell content is not in the given sentence. In total we had 1,257 instances (1,152 instances where at least 3 annotators agreed + 105 test questions).

4.5 Evaluation results

Baseline We evaluate our algorithm against crowd-annotated data using a *lexico-syntactic clause* (LSC) baseline. Given a sentence containing a shell instance and its parse tree, this baseline extracts the postnominal or complement clause from the parse tree depending only upon the lexico-syntactic pattern of the shell noun. For instance, for the *N-that* and *N-be-to* patterns, it extracts the postnominal *that* clause and the complement *to*-infinitive clause, respectively.[9]

Results Table 4.4 shows the evaluation results for the LSC baseline, the algorithm without Schmid's cues (A–SC), and the algorithm with Schmid's cues (A+SC). Overall, we see that our

[9]Note that we only extract subordinating clauses (e.g., (SBAR (IN *that*) (*clause*))) and *to*-infinitive clauses, and not relative clauses.

algorithm is adding value. The A–SC condition in all cases and the A+SC condition in some cases outperform the LSC baseline, which proves to be rather low, especially for the shell nouns with strict syntactic and semantic expectations, such as *fact* and *reason*. These nouns have strict expectations in the sense that their shell content can take very few semantic and syntactic forms. Accordingly, the families *Thing* and *Certainty* of the shell noun *fact* suggest only a *that* clause, and the *Reason* family of the shell noun *reason* suggests only *that* and *because* clauses for the shell content.

That said, we observe a wide range of performance for different shell nouns. On the up side, the A+SC results for the shell nouns *idea*, *issue*, *concept*, *decision*, *reason*, and *fact* outperform the baseline and the A–SC results. In particular, the A+SC results for the shell nouns *fact* and *reason* are markedly better than the baseline results. These cues help in correctly resolving examples such as (51) to *None*, where the postnominal *to*-infinitive clause describes the purpose or the goal for the reason, but not the shell content itself.

(51) There was still **reason** to expect the Fed to raise interest rates in July.

On the down side, adding Schmid's cues hurts the performance of more versatile nouns, which can take a variety of clauses. Although the A–SC results for the shell nouns *plan*, *policy*, *problem*, *trouble*, *difficulty*, and *phenomenon* are well above the baseline, the A+SC results are markedly below it. That is, Schmid's cues were deleterious. Our error analysis revealed that these nouns are versatile in terms of the clauses they take as shell content, and Schmid's cues restrict these clauses to be selected as shell content. For instance, the shell noun *problem* occurs in two semantic families with *N-be-that/of* and *N-be-to* as pattern cues (Table 4.2, p. 88), and postnominal clauses are not allowed for this noun. Although these cues help in filtering some unwanted cases, we observed a large number of cases where the shell content is given in postnominal clauses, as shown in (52).

(52) I was trying to address **the problem** of **unreliable testimony by experts** in capital cases.

Similarly, the *Plan* family does not allow the *N-of* pattern. This cue works well for the shell noun *decision* from the same family because often the postnominal *of* clause is the agent for

this shell noun and not the shell content. However, it hurts the performance of the shell noun *policy*, as *N-of* is a common pattern for this shell noun (e.g., ... *officials in Rwanda have established* ***a policy*** *of* ***refusing to protect refugees***...). Other failures of the algorithm are due to parsing errors and lack of inclusion of context information.

4.6 Discussion and conclusion

In this chapter, we proposed a general method to resolve CSNs, which exploits information derived from the linguistics literature. This is a first step towards general shell noun resolution.

The first goal of this work was to point out the difficulties associated with the resolution of CSNs. The low resolution results of the LSC baseline demonstrate the difficulties of resolving such cases using lexico-syntactic structure alone, suggesting the need for incorporating more linguistic knowledge in the resolution.

The second goal of this work was to examine to what extent knowledge derived from the linguistics literature helps in resolving shell nouns. We conclude that Schmid's patterns and clausal cues are useful for resolving nouns with strict syntactic expectations (e.g., *fact*, *reason*); however, these cues are defeasible: they miss a number of cases in our corpus. It is possible to improve on Schmid's cues using crowdsourcing annotation and by exploiting lexico-syntactic patterns associated with different shell nouns from a variety of corpora.

Shell nouns take a number of semantic arguments. In this respect, they are similar to the general class of argument-taking nominals as given in the NomBank (Meyers et al., 2004). Similarly, there is a small body of literature that addresses nominal semantic role labelling (Gerber et al., 2009) and nominal subcategorization frames (Preiss et al., 2007). That said, the distinguishing property of shell nouns is that one of their semantic arguments is the shell content, but the literature in computational linguistics does not provide any method that is able to identify the shell content. Schmid's families and crowdsourcing annotation of CSN shell content could help enrich the existing resources such as NomBank.

One limitation of our approach is that in our resolution framework, we do not consider the problem of ambiguity of nouns that may not be used as shell nouns. The occurrence of nouns with the lexical patterns in Table 2.1 (p. 18) does not always guarantee shell noun usage. For instance, in our data, we observed a number of instances of the noun *issue* with the publication sense (e.g., *this week's issue of Sports Illustrated*).

Our algorithm is able to deal with only a restricted number of shell noun usage constructions, but the shell content can be expressed in a variety of other constructions. A robust machine learning approach that incorporates context and deeper semantics of the sentence, along with Schmid's constraints, could mitigate this limitation.

Chapter 5

Resolving Anaphoric Shell Nouns

5.1 Introduction

Last chapter described an algorithm to resolve CSNs when the shell content occurs in the same sentence as that of the shell noun phrase. This resolution approach cannot resolve examples such as (53).

(53) The municipal council had to decide **whether to balance the budget by raising revenue or cutting spending**. The council had to come to a resolution by the end of the month. **This issue** was dividing communities across the country.[1]

The goal of this chapter is to develop a general computational method to resolve anaphoric shell nouns (ASNs), i.e., shell nouns occurring in anaphoric constructions.[2] In case of CSNs, the lexico-syntactic patterns provide strong clues about where to find the shell content. In case of ASNs, the shell content can occur anywhere in the given text, and there are no obvious lexio-syntactic clues that can help identifying the shell content. Moreover, there is no restriction on the syntactic type of the shell content: they can be of different syntactic shapes such as verb phrases, noun phrases, clauses, and sentences. Consequently, the search space of ASN shell content candidates is large. In a sample of the NYT corpus, we observed approximately 50 to

[1] This is a constructed example.

[2] The work presented in this chapter is based on Kolhatkar et al. (2013a) and Kolhatkar et al. (2013b).

60 distinct syntactic constituents per sentence.

Chapter 3 described an approach to resolve a particular case of anaphoric shell nouns (ASNs), namely *this issue* in the Medline domain. The approach followed a typical problem-solving procedure used in computational linguistics: annotation, supervised machine learning with the characteristic features occurring in the subset of the annotated data, and evaluation on the held-out test data. But it is not straightforward to generalize *this issue* annotation and resolution to other ASNs, such as *this fact*, *this decision*, and *this question*, primarily because there is no large-scale annotated corpus available for a variety of ASNs and their shell content and manual annotation is an expensive and time-consuming task.

In this chapter, I describe a general approach to resolve ASNs. The first two sections are based on Kolhatkar et al. (2013b). Section 5.2 explains our hypothesis and the trick we use to resolve ASNs without any manually annotated training data. Section 5.3 describes our resolution algorithm: the generation of labelled training data, feature extraction, and supervised machine learning models for shell nouns. Section 5.4 describes how we created annotated data for evaluation. This section is based on Kolhatkar et al. (2013a). Section 5.5 demonstrates how far can we get with our resolution algorithm.

5.2 Hypothesis

The goal of this chapter is to identify shell content of shell nouns occurring in anaphoric constructions, such as *this issue* in (53). We know that shell nouns occur fairly frequently in cataphoric constructions, as shown in (54).

(54) Of course, **the central, and probably insoluble, issue** is **whether animal testing is cruel**.

In Chapter 4, we proposed a method to resolve such examples that does not rely on annotated data.

Observe that there are two striking similarities between examples (53) and (54). First, in both examples, the shell content represents similar kinds of abstract objects, as both of them represent the general notion of the shell noun, e.g., the notion of an *issue* is an important

problem which requires a solution. Second, in both cases the shell content is expressed with a similar syntactic construct, a *wh*-clause.

We exploit these similarities between shell content of CSNs and ASNs to resolve ASNs. Accordingly, we hypothesize that CSN shell content and ASN shell content share some linguistic properties and hence linguistic knowledge encoded in CSN shell content will help in interpreting ASNs.

To test the hypothesis, we examine which features present in CSN shell content are relevant in interpreting ASNs. For instance, Vendler (1968) points out prototypical syntactic constructs preferred by various shell nouns (e.g., *question* and *issue* take a *wh*-question clause). Schmid (2000) discusses the strong present and past tense association with the shell noun *fact*. We aim to automatically identify all such cues that are common between shell content of CSNs and ASNs using machine learning algorithms.

5.3 Resolving ASNs using shell content of CSNs

Figure 5.1 shows an overview of our methodology. Given a shell noun, we collect examples following CSN patterns. Then to get automatically labelled CSN shell content data, we extract shell content of these examples by applying the resolution algorithm in Chapter 4. This automatically labelled CSN shell content data serves as our training data to resolve ASNs. With this training data, we train supervised machine learning models. Later, we apply these models to rank ASN shell content candidates. Finally, we evaluate our approach using crowdsourcing.

5.3.1 Training phase

As shown in Figure 5.1, the goal of the training phase is to extract training data using the algorithm given in Chapter 4, and to train shell noun models which can be used to predict ASN shell content.

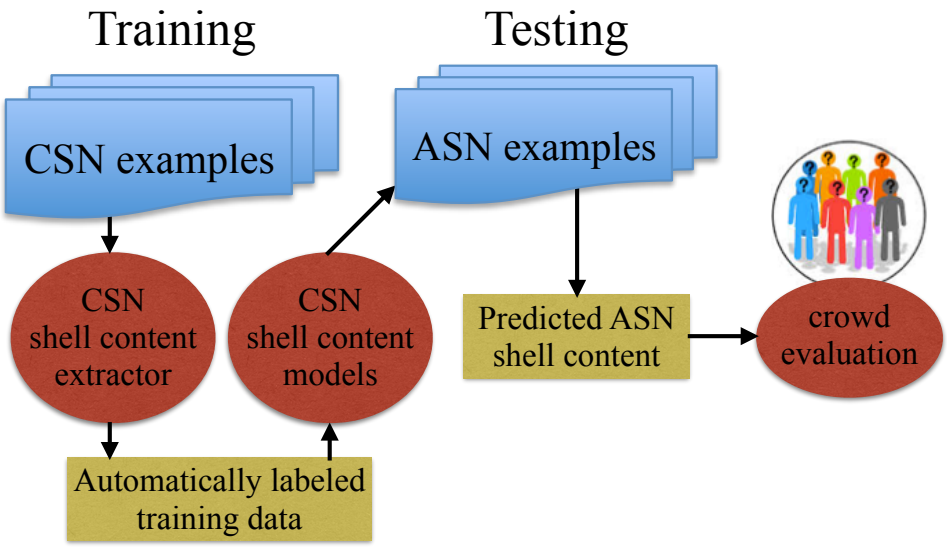


Figure 5.1: Overview of resolving ASNs using shell content of CSNs

5.3.1.1 Generating training data

For each shell noun to be resolved, we collect the CSN examples for that noun from the given corpus following seven cataphoric patterns and one anaphoric pattern. For instance, for the shell noun *issue*, the corresponding patterns are: *issue-be-to*, *issue-be-that*, *issue-be-wh*, *issue-to*, *issue-that*, *issue-wh*, *issue-of*, and *Sub-be-issue*. Then we extract CSN instances following the procedure described in Section 4.4.2. Next, for each CSN example, we get its shell content by applying the CSN resolution algorithm from Chapter 4. If the algorithm returns *None*, we skip the example. Finally, we have automatically labelled CSN shell content data, i.e., the pairs {CSN example, its shell content}.

5.3.1.2 Models for CSN shell content

With the generated training data, we build supervised machine learning models. Recall that our goal is to identify shell content of ASNs. So the question is how to formulate the machine learning problem so that we can use models built on automatically labelled CSN shell content data to predict ASN shell content.

In case of ASNs, there are many eligible candidates for the shell content, and one of these

candidates represents the correct shell content. Moreover, not all other candidates are equally mistaken. Some candidates are very close to the correct shell content, but they are not the right answer just because there is a better option present in the set of candidates. So keeping our test scenario in mind, we create a set of eligible candidates for CSN examples, and train machine learning ranking models, as in Chapter 3. The following sections describe each step of the ranking models in detail.

Candidate extraction The first step is to extract the set of eligible shell content 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 shell content candidates — the *true* shell content given by the method in Chapter 4. But we also need negative examples of shell content candidates. By their construction, CSNs have their shell content 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* shell content, are considered as negative examples. With this candidate extraction method, we end up with many more negative examples than positive examples, but that is exactly what we expect with ASN shell content candidates, i.e., the test data on which we will be applying our models.

Features We came up with a set of features based on the properties that we found to be common in both ASN and CSN shell content. The features syntactic type, embedding level, subordinating conjunctions, and length are directly taken from *this issue* resolution (see Section 3.3.2). That said, not all features from *this issue* resolution are relevant here due to the nature of the CSN training data.

Syntactic type of the candidate (S) In Table 2.2 (p. 20) we noted 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 shell content more verb phrases than, for example, noun phrases. This feature tries to capture the syntactic type

{S, SBAR, SINV, SQ, SBARQ, ROOT} → S
{ADVP} → ADVP
{VP, +VP} → VP
{NP, QP, NX, NAC, NP-TMP} → NP
{PP} → PP
{ADJP} → ADJP
{PRN} → PRN
{CONJP} → CONJP
{FRAG, UCP, PRN+S, RRC, LST} → FRAG
{X, INTJ, PRT} → X
other → POS-level

Table 5.1: Mapping between fine-grained syntactic types and coarse-grained syntactic types.

preferences for the given shell noun. 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, as shown in Table 5.1.[3]

Context features (C) Context features allow our models to learn about the contextual clues that signal the shell content. This class contains two features: (a) 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. We employ both coarse-grained and fine-grained versions here.

Embedding level features (E) Müller (2008)'s embedding level features turned out to be useful in *this issue* resolution (see section 3.4.2). So we employ them in general ASN resolution. 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

[3]Note that the table does not provide a comprehensive list of all possible syntactic types in our data. We check whether the syntactic type starts with the fine-grained syntactic types in the table. If it does, we map it to the corresponding coarse-grained syntactic type. For instance, S+VP is mapped to S, although it is not present in the list of fine-grained syntactic types in the table.

salient enough to be an shell content. Although we consider all syntactic constituents as potential candidates, there are many that clearly cannot be shell content. This feature will allow us to get rid of this noise.

Subordinating conjunctions (SC) Subordinating conjunctions are common with CSN and ASN shell content. Vendler (1968) points out that the shell noun *fact* prefers a *that*-clause, and *question* and *issue* prefer a *wh*-question clause. Also, the pattern *because X* is common with *reason* (Schmid, 2000). The subordinating conjunction feature encodes these preferences for different shell nouns. The feature checks whether the candidate follows the pattern *SBAR → (IN sconj) (S ...)*, where *sconj* is a subordinating conjunction from the list: *about*, *after*, *although*, *as*, *because*, *before*, *by*, *except*, *for*, *if*, *in*, *lest*, *like*, *once*, *since*, *so*, *than*, *that*, *though*, *till*, *unless*, *until*, *upon*, *whereas*, *whether*, *while*, and *with*.

Verb features (V) CSNs and ASNs encapsulate propositional content, which tends to contain verbs. All examples from Table 2.1 (p. 18), 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 non-finite, and (c) presence of modals.

For (a), we look for the presence of a verb phrase in the constituent. For (b), we look for the first VP node in the constituent; if the part-of-speech tag of the main verb in this node is in the set {VBZ, VBP, VBD, MD}, we set the finite verb feature, else we set the non-finite verb feature. For (c), we look for the part-of-speech tag MD in the constituent.

Length features (L) The intuition behind these features is that CSN and ASN shell content tends 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 shell content.

Lexical features (LX) The CSN resolution algorithm from Chapter 4 provides us a large number of shell content examples for each shell noun. A natural question is whether certain words tend to occur more frequently in the shell content than non-content parts of the sentence. To deal with this question, we extracted all shell content unigrams (i.e., unigrams occurring in shell content part of the sentence) and non-content unigrams (i.e., unigrams occurring in non-content parts of the sentence) for each shell noun. Then for all shell content unigrams for a particular shell noun, we computed the most informative unigrams in terms of information gain (Yang and Pedersen, 1997) and considered the first 50 highly ranked unigrams as the lexical features for that noun. In contrast with the other features, these lexical features are tailored for each shell noun and are extracted *a priori*. For instance, the first 50 most-informative lexical features (i.e., words with high information gain) for the shell noun *question* are: *question*, *whether*, *be*, *will*, *to*, *how*, *say*, *can*, *what*, *the*, *of*, *”*, *but*, *or*, *would*, *do*, *in*, *“*, *big*, *have*, *they*, *it*, *why*, *real*, *and*, *go*, *enough*, *much*, *should*, *only*, *make*, *‘,’*, *this*, *a*, *on*, *unanswered*, *that*, *their*, *key*, *get*, *from*, *which*, *who*, *long*, *them*, *for*, *use*, *we*, *his*. The word *question* is informative because it does not commonly occur in the shell content part and occurs in the non-shell content part in almost all cases due to the nature of CSN examples. During training, we have a binary feature for each of these words and the appropriate features are set based on the presence of these words in the shell content candidate.

Candidate ranking models Now that we have the set of shell content candidates and a set of features, we are ready to train CSN shell content models. As in Chapter 3, we follow the *candidate-ranking* models proposed by Denis and Baldridge (2008).

For every shell noun, we gather automatically extracted shell content data given by the extractor for all instances of that shell noun, as explained in Section 5.3.1.1. Then for each instance in this data, we extract the set of candidates $C = \{c_1, c_2, ..., c_k\}$. 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 eli-

gible 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* shell content, i.e., the automatically extracted shell content for that CSN, otherwise the rank is 2. We use the `svm_rank_learn` call of SVM^{rank} Joachims (2002) for training the candidate-ranking models.

5.3.2 Testing phase

In this phase, we use the learned candidate ranking models to predict the shell content of ASNs.

5.3.2.1 Shell content identification

Candidate extraction Recall that ASNs can have short-distance as well as long-distance shell content. For this reason, the algorithm considers n preceding sentences and the sentence containing the ASN as the source of shell content candidates. Later in Section 5.5, we report results with two different values of n. From these n sentences, we extract all syntactic constituents given by the Stanford parser.[4] Similar to CSN resolution, there could be parsing errors and the correct candidate might not be present in the set of constituents. To mitigate this limitation, we consider 30-best parses given by the parser, and extract a set of unique eligible shell content candidates from all of these parses.

Feature extraction and candidate ranking Given the shell content candidates, feature extraction and candidate ranking are essentially the same as for the training phase, except of course we do not know the *true* shell content. Once we have the feature vectors for each 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^{rank}.

[4] We discard the leaf-level syntactic constituents.

5.4 Evaluation data

For ASN shell content extraction there is no evaluation data previously available. As in Chapter 4, we create evaluation data for ASNs using crowdsourcing. In particular, we focus on a set of six frequently occurring shell nouns from Schmid's list of 670 shell nouns, given in Appendix A: *fact* and *reason* from the *factual* category (see Table 2.4, p. 26), *issue* and *decision* from the *mental* category, *question* from the *linguistic* category, and *possibility* from the *modal* category. The reason for not selecting shell nouns from the eventive and the circumstantial categories was that the shell content of the shell nouns in these categories are rather vague and hard to pinpoint.[5] Four of the six selected shell nouns, *fact*, *reason*, *issue*, and *decision*, were included in the evaluation of CSN resolution (see Section 4.4). Then we extract CSN and ASN examples of these six shell nouns from the NYT corpus to create the CSN corpus and the ASN corpus.

5.4.1 The CSN corpus

The CSN corpus consists of the examples of six selected shell nouns following seven CSN patterns: *N-be-to*, *N-be-that*, *N-be-wh*, *N-to*, *N-that*, *N-wh*, and *N-of*, and one ASN pattern: *Sub-be-N*. We include the anaphoric pattern *Sub-be-N* because similar to the CSN patterns, if the shell content occurs in the same sentence, the CSN resolution algorithm from Chapter 4 is able to identify shell content of shell nouns with this pattern. To extract examples, we follow the regular expressions from Figure 4.2. Table 5.2 shows the six shell nouns and the number of CSN examples per noun in the NYT corpus.

5.4.2 The ASN corpus

We started with about 500 instances for each of the six selected shell nouns (3,000 total instances), containing the pattern {*this* shell_noun}. The instances were extracted from the NYT

[5]This observation is based on a pilot annotation of about 200 ASN instances carried out by myself and Dr. Heike Zinsmeister. The instances contained about 20 instances of 10 different shell nouns from Schmid's six semantic categories.

Shell Noun	CSN frequency
fact	83,591
reason	43,349
issue	58,941
decision	62,451
question	54,234
possibility	46,049

Table 5.2: Shell nouns and their CSN frequency in the NYT corpus.

corpus. 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 non-abstract sense (e.g., *this issue* with a publication-related sense), we were left with 2,822 instances.[6]

5.4.3 Annotation challenges

An essential first step in ASN resolution is to clearly establish the extent of inter-annotator agreement on shell content of ASNs as a measure of feasibility of the task. The following sections describe our annotation methodology in detail. We also describe how we evaluated the feasibility of the task and the quality of the annotation, and the challenges we faced in doing so.

What to annotate? Any annotation task requires a list of *markables*, i.e., a set of well-defined linguistic units to be annotated. But as noted in Chapters 1 and 3, ASN shell content can be of various syntactic shapes and sometimes is not even a well-defined syntactic constituent. So the question of 'what to annotate' as mentioned by Fort et al. (2012) is not straightforward for ASN shell content, as the notion of *markables* is complex compared to ordinary nominal anaphora: the units on which the annotation work should focus are heterogeneous.[7] Moreover, due to this heterogeneous nature of annotation units, there are a huge number of markables

[6]The duplicates were removed using simple heuristic rules.

[7]Occasionally, shell content is non-contiguous spans of text, but in this work, we ignore such instances for simplicity.

(e.g., all syntactic constituents given by a syntactic parse tree). So there are many options to choose from, while only a few units are actually to be annotated.

Lack of a *right* answer It is not obvious how to define clear and detailed annotation guidelines to create a gold-standard corpus for ASN shell content annotation due to our limited understanding of the nature and interpretation of such nouns. The notion of the *right* answer is not well-defined for ASN shell content. The primary challenge is to identify the conditions when two different candidates for annotation should be considered as representing essentially the same concept, which raises deep philosophical issues that we do not propose to solve in this thesis. For instance, do *whether animal testing is cruel* and *animal testing is cruel* represent the same concept? We believe, this challenge could only be possibly addressed by the requirements of downstream applications of ASN resolution. For our purposes, we consider two candidates equivalent if they are exactly the same or they differ only by the introductory subordinating conjunction.

5.4.4 Annotation methodology

So there were two primary challenges involved in the annotation process: first, to find annotators who can annotate data reliably with minimal guidelines, and second, to design simple annotation tasks that will elicit data useful for our purposes. Now we discuss how we dealt with these challenges.

5.4.4.1 Crowdsourcing

As in Chapter 4, we explored the use of CrowdFlower.

5.4.4.2 Design of the annotation tasks

With the help of well-designed gold examples, CrowdFlower can get rid of spammers and ensures that only reliable annotators perform the annotation task. But the annotation task must

be well-designed in the first place to get a good quality annotation. Following the claim in the literature that with crowdsourcing platforms simple tasks do best (Madnani et al., 2010; Wang et al., 2012), we split our annotation task into two relatively simple sequential annotation tasks. First, identifying the sentence containing the shell content, and second, given the sentence of the shell content, identifying the precise shell content. Now we will discuss each of our annotation tasks in detail.

CrowdFlower experiment 1 The first annotation task was about identifying the sentence containing the shell content of the given ASN without actually pinpointing the precise shell content.[8] We designed a CrowdFlower experiment where we presented to the annotators ASNs from the ASN corpus with three preceding paragraphs as context. Sentences in the vicinity of ASNs were each labelled: four sentences preceding the ASN, the sentence containing the ASN, and two sentences following the ASN. This choice was based on our pilot annotation: the shell content very rarely occurs more than four sentences away from the ASN. The annotation task was to pinpoint the sentence in the presented text that contained the shell content for the ASN and selecting the appropriate sentence label as the correct answer. If no labelled sentence in the presented text contained the shell content, we suggested to the annotators to select *None*. If the shell content spanned more than one sentence, then we suggested to them to select *Combination*. We also provided a link to the complete article from which the text was drawn in case the annotators wanted to have a look at it. Figure 5.2 shows a screenshot of our interface.

Settings We asked for 8 judgements per instance and paid 8 cents per annotation unit. Our job contained in total 2,822 annotation units with 168 gold units. The gold units are created by carrying out a pilot experiment and then considering the units with high agreement. As we were interested in the verdict of native speakers of English, we limited the allowed demographic region to English-speaking countries. The annotation guidelines are given in Appendix F.

[8]The shell nouns we have chosen tend to have shell content that lies within a single sentence.

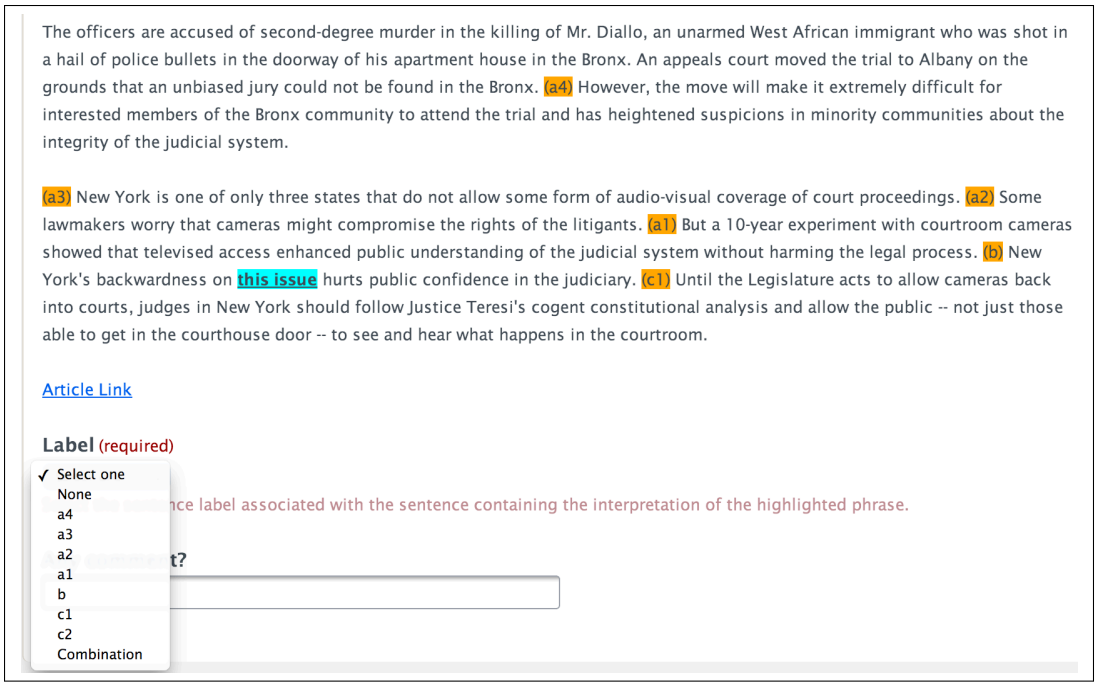


Figure 5.2: CrowdFlower experiment 1 interface

CrowdFlower experiment 2 This annotation task was about pinpointing the exact shell content text of the ASN instances. We designed a CrowdFlower experiment, where we presented to the annotators ASN instances from the ASN corpus with highlighted ASNs and the sentences containing the shell content, the output of experiment 1. One way to pinpoint the exact shell content string is to ask the annotators to mark free spans of text within the sentence containing shell content, similar to Byron (2003) and Artstein and Poesio (2006). However, CrowdFlower quality-control mechanisms work best with multiple-choice annotation labels. So we decided to display a set of labelled candidates to the annotators and ask them to choose the answer that best represents the ASN shell content. A practical requirement of this approach is that the number of options to be displayed be only a handful in order to make it a feasible task for online annotation. But, the number of markables for ASN shell content is large. If, for example, we define markables as all syntactic constituents given by the Stanford parser, there are on average 49.5 such candidates per sentence in the ASN corpus. It is not practical to display all these candidates and to ask CrowdFlower annotators to choose one answer from this many options. Also, some potential candidates are clearly not appropriate candidates for a particular

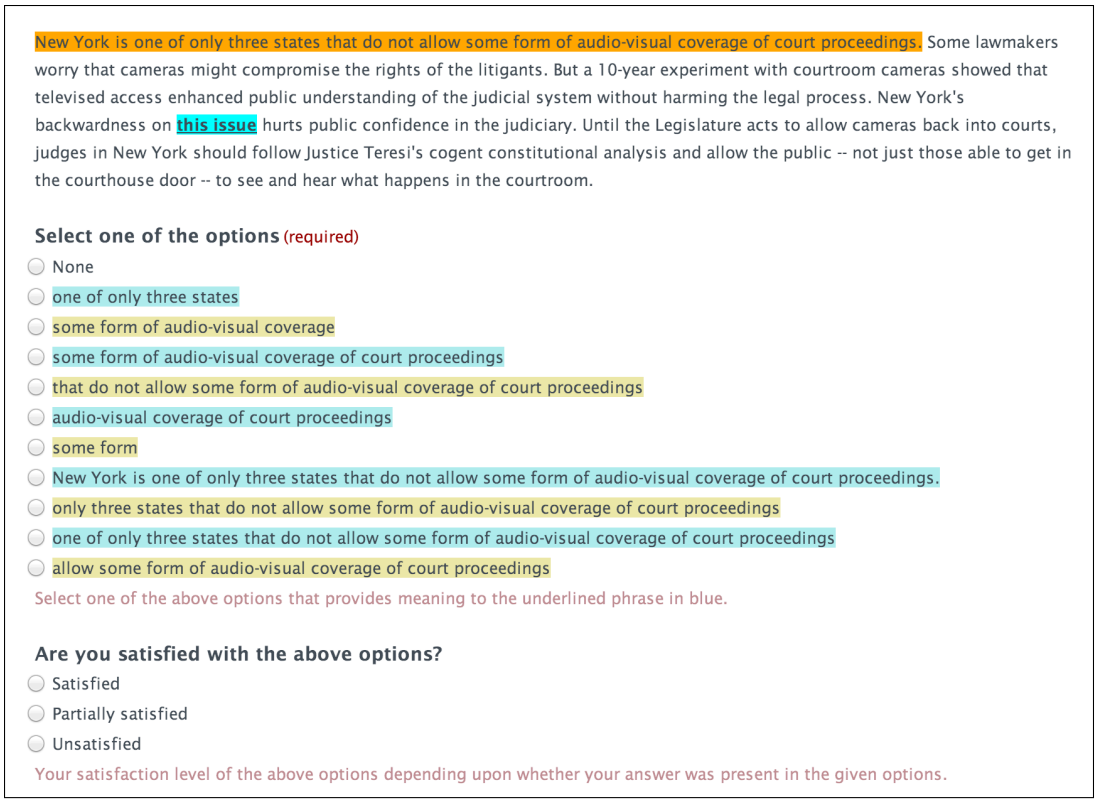


Figure 5.3: CrowdFlower experiment 2 interface

shell noun. For instance, noun phrase candidates are not usually appropriate for the shell noun *fact*, as generally facts are propositions. So the question is whether it is possible to restrict this set of candidates by discarding unlikely ones.

Here, we draw on the most-likely shell content candidates given by our ranking models from Section 5.3.2.1. According to our hypothesis, the models trained on CSN shell content help in identifying ASN shell content. We expect that the ranking models trained on CSN shell content data push down the spurious candidates and bring up the most probable candidates of ASNs. So we apply the appropriate trained CSN shell content model to predict candidate rankings of the given ASN. We displayed the first 10 highly-ranked candidates (randomly ordered) to the annotators. In addition, we made sure not to display two candidates with only a negligible difference. For example, given two candidates, *X* and *that X*, which differ only with respect to the introductory *that*, we chose to display only the longer candidate *that X*. In a controlled annotation, with detailed guidelines, such difficulties of selecting between minor

variations could be avoided. Figure 5.3 shows a screenshot of our interface.

Settings As in experiment 1, we asked for 8 judgements per instance and paid 6 cents per annotation unit. The reason for paying a little bit less in the second experiment is that since we highlight the sentence containing the shell content, the annotators take relatively less time per each instance.[9] For this experiment we considered only 2,323 annotation units with 151 gold units, only the units where at least half of the trustworthy annotators agreed on an answer in experiment 1. This task turned out to be a suitable task for crowdsourcing as it offered a limited number of options to choose from, instead of asking the annotators to mark arbitrary spans of text. The annotation guidelines are given in Appendix F.

5.4.5 Inter-annotator agreement

Our annotation tasks pose difficulties in measuring inter-annotator agreement both in terms of the task itself and the platform used for annotation. In this section, we describe our attempt to compute agreement for each of our annotation tasks and the challenges we faced in doing so.

5.4.5.1 CrowdFlower experiment 1

Recall that in this experiment, annotators identify the sentence containing the shell content and select the appropriate sentence label as their answer. We know from our pilot annotation that the distribution of such labels is skewed: most of the ASN shell content lies in the sentence preceding the anaphor sentence. We observed the same trend in the results of this experiment. In the ASN corpus, the crowd chose the preceding sentence 64% of the time, the same sentence 13% of the time, and long-distance sentences 23% of the time.[10] Considering the skewed distribution of labels, if we use traditional agreement coefficients, such as Cohen's κ (1960) or Krippendorff's α (2013), expected agreement is very high, which in turn results in a low reli-

[9] The payment amount for both experiments followed CrowdFlower payment guidelines.

[10] This confirms Passonneau (1989)'s observation that non-nominal shell content tend to be close to the anaphors.

ability coefficient (in our case $\alpha = 0.61$), which does not necessarily reflect the true reliability of the annotation (Artstein and Poesio, 2008).

One way to measure the reliability of the data, without taking chance correction into account, is to consider the distribution of the ASN instances with different levels of CrowdFlower *confidence*. CrowdFlower assigns a unique answer to each annotation unit along with a *confidence* score (denoted as c henceforth). Each annotator has a trust level based on how she performs on the gold examples, and confidence score is the normalized score of the summation of the trusts. For example, suppose annotators A, B, and C with trust levels 0.75, 0.75, and 1.0 give answers *no*, *yes*, *yes* respectively for a particular instance. Then the answer *yes* will score 1.75 and answer *no* will score 0.75 and *yes* will be chosen as the crowd's answer with c = 0.7 (i.e., $1.75/(1.75+0.75)$). We use these confidence scores in our analysis of inter-annotator agreement below.

Table 5.3 shows the percentages of instances in different confidence level bands for each shell noun as well as for all instances. For example, for the shell noun *fact*, 8% of the total number of *this fact* instances were annotated with $c < 0.5$. As we can see, most of the instances of the shell nouns *fact*, *reason*, *question*, and *possibility* were annotated with high confidence. In addition, most of them occurred in the band $0.8 \leq c \leq 1$. There are relatively few instances with low confidence for these nouns, suggesting the feasibility of reliable shell content annotation for these nouns. By contrast, the mental nouns *issue* and *decision* had a large number of low-confidence ($c < 0.5$) instances, bringing in the question of reliability of shell content annotation of these nouns.

Given these results with different confidence levels, the primary question is what confidence level should be considered acceptable? For our task, we required that at least four trusted annotators out of eight annotators should agree on an answer for it to be acceptable.[11] We will talk about acceptability later in Section 5.4.6.

[11] When at least four trusted annotators agree on an answer the confidence is ≥ 0.5. So we chose 0.5 as the threshold, after systematically examining instances with different confidence levels.

	F	*R*	*I*	*D*	*Q*	*P*	*all*
$c < .5$	8	8	36	21	13	7	16
$.5 \leq c < .6$	6	6	13	8	7	5	8
$.6 \leq c < .8$	24	25	31	31	22	27	27
$.8 \leq c < 1.$	22	23	11	14	19	25	18
$c = 1.$	40	38	9	26	39	36	31
Average c	.83	.82	.61	.72	.80	.83	.76

Table 5.3: CrowdFlower confidence distribution for CrowdFlower experiment 1. Each column shows the distribution in percentages for confidence of annotating antecedents of that shell noun. The final row shows the average confidence of the distribution. Number of ASN instances = 2,822. *F* = *fact*, *R* = *reason*, *I* = *issue*, *D* = *decision*, *Q* = *question*, *P* = *possibility*.

5.4.5.2 CrowdFlower experiment 2

Recall that this experiment was about identifying the precise shell content text segment given the sentence containing the shell content. It is not clear what the best way to measure the amount of such agreement is. Agreement coefficients such as Cohen's κ underestimate the degree of agreement for such annotation, suggesting disagreement even between two very similar annotated units (e.g., two text segments that differ in just a word or two). We present the agreement results in three different ways: Krippendorff's α with distance metrics Jaccard and Dice (Artstein and Poesio, 2006), Krippendorff's unitizing alpha (Krippendorff, 2013), and CrowdFlower confidence values.

Krippendorff's α using Jaccard and Dice The agreement results of Krippendorff's α using distance metrics Jaccard and Dice are shown in Table 5.4. Our agreement results are comparable to Artstein and Poesio's agreement results. They had 20 annotators annotating 16 anaphor instances with segment shell content, whereas we had 8 annotators annotating 2,323 ASN instances. As Artstein and Poesio point out, expected disagreement in case of such shell content annotation is close to maximal, as there is little overlap between segment shell content of different anaphors and therefore α pretty much reflects the observed agreement.

	Jaccard			Dice		
	D_o	D_e	α	D_o	D_e	α
A&P	.53	.95	.45	.43	.94	.55
Our results	.47	.96	.51	.36	.92	.61

Table 5.4: Agreement using Krippendorff's α for CrowdFlower experiment 2. A&P = Artstein and Poesio (2006, p. 4).

	F	*R*	*I*	*D*	*Q*	*P*	*all*
$c < .5$	11	17	32	31	14	28	21
$.5 \leq c < .6$	12	12	19	23	9	19	15
$.6 \leq c < .8$	36	33	34	32	30	36	33
$.8 \leq c < 1.$	24	22	10	10	21	13	18
$c = 1.$	17	16	5	3	26	4	13
Average c	.74	.71	.60	.59	.77	.62	.68

Table 5.5: CrowdFlower confidence distribution for CrowdFlower experiment 2. Each column shows the distribution in percentages for confidence of annotating antecedents of that shell noun. The final row shows the average confidence of the distribution. Number of ASN instances = 2,323. *F* = *fact*, *R* = *reason*, *I* = *issue*, *D* = *decision*, *Q* = *question*, *P* = *possibility*.

Krippendorff's *unitizing* α ($_u\alpha$) As with *this-issue* annotation agreement, we use $_u\alpha$ for measuring reliability of the ASN shell content annotation task. $_u\alpha$ incorporates the notion of distance between strings by using a distance function which is defined as the square of the distance between the non-overlapping tokens in our case. The distance is 0 when the annotated units are exactly the same, and is the summation of the squares of the unmatched parts if they are different. We compute observed and expected disagreement as explained by Krippendorff (2013, p. 313). For our data, $_u\alpha$ was 0.54. Note that $_u\alpha$ reported here is just an approximation of the actual agreement as in our case the annotators chose an option from a set of predefined options instead of marking free spans of text. $_u\alpha$ was lower for the mental nouns *issue* and *decision* and the modal noun *possibility* compared to other shell nouns.

CrowdFlower confidence results We also examined different confidence levels for ASN shell content annotation. Table 5.5 gives confidence results for all instances and for each noun. In contrast with Table 5.3, the instances are more evenly distributed here. As in experiment 1, the mental nouns *issue* and *decision* had many low confidence instances. For the modal noun

possibility, it was easy to identify the sentence containing the shell content, but pinpointing the precise shell content turned out to be difficult.

5.4.5.3 Nature of disagreement in ASN annotation.

Disagreement in experiment 1 There were two primary sources of disagreement in experiment 1. First, the annotators had problems agreeing on the answer *None*. We instructed them to choose *None* when the sentence containing the shell content was not labelled. Nonetheless, some annotators chose sentences that did not precisely contain the actual shell content but just hinted at it. Second, sometimes it was hard to identify the precise shell content sentence as the shell content was either present in the blend of all labelled sentences or there were multiple possible answers, as shown in example (55).

(55) Any biography of Thomas More has to answer one fundamental question. Why? Why, out of all the many ambitious politicians of early Tudor England, did only one refuse to acquiesce to a simple piece of religious and political opportunism? What was it about More that set him apart and doomed him to a spectacularly avoidable execution?

The innovation of Peter Ackroyd's new biography of More is that he places the answer to **this question** outside of More himself.

Here, the author formulates the question in a number of ways and any question mentioned in the preceding text can serve as the shell content of the anaphor *this question*.

Hard instances Low agreement can indicate different problems: unclear guidelines, poor-quality annotators, or difficult instances (e.g., not well understood linguistic phenomena) (Artstein and Poesio, 2006). We can rule out the possibility of poor-quality annotators for two reasons. First, we consider 8 diverse annotators who work independently. Second, we use CrowdFlower's quality-control mechanisms and hence allow only relatively trustworthy annotators to annotate our texts. Regarding instructions, we take inter-annotator agreement as a measure for feasibility of the task, and hence we keep the annotation instruction as simple as possible. This could be a source of low agreement. The third possibility is hard instances.

Our results show that the mental nouns *issue* and *decision* had many low-confidence instances, suggesting the difficulty associated with the interpretation of these nouns (e.g., the very idea of what counts as an issue is fuzzy). The shell noun *decision* was harder because most of its instances were court-decision related articles, which were in general hard to understand.

Different strings representing similar concepts The primary challenge with the ASN annotation task is that different shell content candidates might represent the same concept and it is not trivial to incorporate this idea in the annotation process. When five trusted annotators identify the shell content as *but X* and three trusted annotators identify it as merely *X*, since CrowdFlower will consider these two answers to be two completely different answers, it will give the answer *but X* a confidence of only about 0.6. ${}_u\alpha$ or α with Jaccard and Dice will not consider this as a complete disagreement; however, the coefficients will register it as a difference. In other words, the difference functions used with these coefficients do not disregard semantics, paraphrases, and other similarities that humans might judge as inconsequential. One way to deal with this problem would be clustering the options that reflect essentially the same concepts before measuring the agreement. Some of these problems could also be avoided by formulating instructions for marking shell content so that these differences do not occur in the identified shell content. However, crowdsourcing platforms require annotation guidelines to be clear and minimal, which makes it difficult to control the annotation variations.

5.4.6 Evaluation of crowd annotation

CrowdFlower experiment 2 resulted in 1,810 ASN instances with $c > 0.5$. The question is how good these annotations are from the experts' point of view.

To examine the quality of the crowd annotation we asked two judges A and B to evaluate the *acceptability* of the crowd's answers. The judges were highly-qualified academic editors: A, a researcher in Linguistics and B, a translator with a Ph.D. in History and Philosophy of Science. From the crowd-annotated ASN shell content data, we randomly selected 300 instances, 50

		Judge B				
		P	*R*	*I*	*N*	Total
Judge A	*P*	**171**	44	11	7	233
	R	12	**27**	7	4	50
	I	2	4	**6**	1	13
	N	1	2	0	**1**	4
	Total	186	77	24	13	300

Table 5.6: Evaluation of ASN antecedent annotation. *P* = *perfectly*, *R* = *reasonably*, *I* = *implicitly*, *N* = *not at all*

instances per shell noun. We made sure to choose instances with borderline confidence ($0.5 \leq c < 0.6$), medium confidence ($0.6 \leq c < 0.8$), and high confidence ($0.8 \leq c \leq 1.0$). We asked the judges to rate the acceptability of the crowd-answers based on the extent to which they provided interpretation of the corresponding anaphor. We gave them four options: *perfectly* (the crowd's answer is perfect and the judge would have chosen the same shell content), *reasonably* (the crowd's answer is acceptable and is close to their answer), *implicitly* (the crowd's answer only implicitly contains the actual shell content), and *not at all* (the crowd's answer is not in any way related to the actual shell content).[12] Moreover, if they did not mark *perfectly*, we asked them to provide their shell content string. The two judges worked on the task independently and they were completely unaware of how the annotation data was collected.

Table 5.6 shows the confusion matrix of the ratings of the two judges. Judge B was stricter than Judge A. Given the nature of the task, it was encouraging that most of the crowd-shell content were rated as *perfectly* by both judges (77.7% by A and 62% by B). Note that *perfectly* is rather a strong evaluation for ASN shell content annotation, considering the nature of ASN shell content itself. If we weaken the acceptability criteria and consider the shell content rated as *reasonably* to be also acceptable shell content, 84.6% of the total instances were acceptable according to both judges.

Regarding the instances marked *implicitly*, most of the time the crowd's answer was the closest textual string of the judges' answer. So we again might consider instances marked

[12]Before starting the actual annotation, we carried out a training phase with 30 instances, which gave an opportunity to the judges to ask questions about the task.

implicitly as acceptable answers.

For a very few instances (only about 5%) one or both of the judges marked *not at all*. This was a positive result and suggests success of different steps of our annotation procedure: identifying broad region, identifying the set of most likely candidates, and identifying precise shell content. As we can see in Table 5.6, there were 7 instances where the judge A rated *perfectly* while the judge B rated *not at all*, i.e., completely contradictory judgements. When we looked at these examples, they were rather hard and ambiguous cases. An example is shown in (56). The *whether* clause marked in the preceding sentence is the crowd's answer. One of our judges rated this answer as *perfectly*, while the other rated it as *not at all*. According to her the correct shell content is *whether Catholics who vote for Mr. Kerry would have to go to confession*.

(56) Several Vatican officials said, however, that any such talk has little meaning because the church does not take sides in elections. But the statements by several American bishops that Catholics who vote for Mr. Kerry would have to go to confession have raised the question in many corners about **whether this is an official church position**.

The church has not addressed **this question** publicly and, in fact, seems reluctant to be dragged into the fight..."

There was no notable relation between the judge's rating and the confidence level: many instances with borderline confidence were marked *perfectly* or *reasonably*, suggesting that instances with $c \geq 0.5$ were reasonably annotated instances, to be used as training data for ASN resolution.

5.4.7 The annotated ASN corpus

Finally, an annotated ASN shell content corpus containing 1,810 ASN instances was created. We discarded the instances where fewer than 4 annotators agreed on an answer. Figure 5.4 shows the distribution of syntactic types for different shell nouns. The distribution is similar for the shell nouns from Schmid's same semantic categories. For instance, the factual shell nouns *reason* and *fact* show a similar distribution which is quite different from the mental

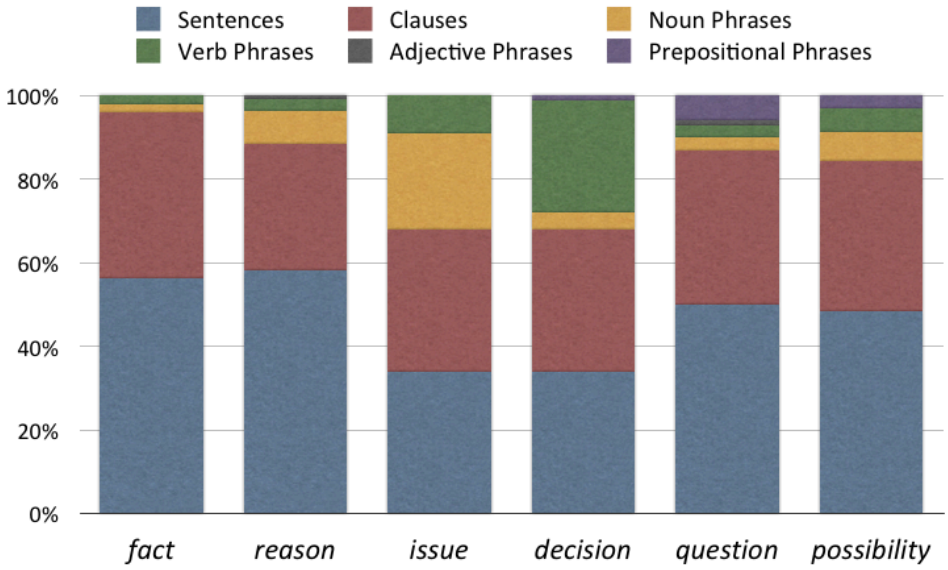


Figure 5.4: Distribution of syntactic types of the shell content in the annotated ASN corpus

shell nouns *decision* and *issue*. A majority of shell content is either full sentences or clauses. Moreover, a fair number of verb phrases and noun phrases were marked as shell content. The syntactic type distributions for the shell nouns *issue* and *decision* show that they can fit different kinds of abstract objects.

5.5 How far can we get with the CSN models?

Now that we have reliably annotated ASN shell content data, we can examine how far we can get with the CSN shell content models. To examine which CSN shell content features are relevant in identifying ASN shell content, we carried out ablation experiments with all feature class combinations for the features from Section 5.3.1.2. We compared the rankings given by our ranker to the crowd's answer using *Success at n* (S@n). More specifically, we count the number of instances where the crowd's answers occur within our ranker's first n choices. S@n then is this count divided by the total number of instances. Note that S@1 is equivalent to the standard precision.

The following sections discuss two sets of ablation experiments. Section 5.5.1 describes the

results of experiments when *n* is set to 5. These experiments consider the sentence containing the ASN and four preceding sentences as the source of candidates. All eligible candidates from these sentences are ranked by applying the corresponding CSN model. Section 5.5.2 describes the results of the experiments with the assumption that we know the sentence containing the shell content. The precise shell content from that sentence is identified using the CSN models.

We compared our results against two baselines: *preceding sentence* (PSbaseline) or *crowd sentence* (CSbaseline), depending upon the experiment, and *chance*. The preceding sentence baseline chooses the previous sentence as the correct shell content, and the crowd-sentence baseline chooses the sentence given by the CrowdFlower experiment 1 as the correct answer. The chance baseline chooses a candidate from a uniform random distribution over the set of 10 top-ranked candidates.

5.5.1 Identifying precise shell content from *n* surrounding sentences

In these experiments, we considered five sentences as the source of shell content candidates: the sentence containing the shell noun phrase and four preceding sentences.[13] From these sentences we extract all syntactic constituents as eligible candidates and then rank them using the ranking models trained on CSN shell content data (see Section 5.3.2.1). Note that the search space of shell content candidates is large (more than 100 candidates per instance). The results of these ablation experiments are shown in Table 5.7. All results are better than the *chance* baseline. That said, not all results are better than the PSbaseline. The shell nouns *possibility*, *issue*, and *decision* turned out to be hard. When we looked at the scores assigned to the candidates for these shell nouns, we observed that many candidates were assigned the same positive score by SVM^{rank}. One reason could be that the range of objects these shell nouns could fit is quite large and hence a large number of candidates are considered as viable candidates. The features that occur frequently in many best-performing combinations were embedding level (E) and subordinating conjunction (SC) features. The lexical features did not

[13]The number 5 was derived from a pilot annotation experiment.

Table 5.7: Evaluation of our ranker for antecedents of six ASNs. Surrounding 5 sentences of the anaphor were considered as the source of candidates. For each noun we show the two best-performing feature combinations. S@n is the Success at rank n (S@1 = standard precision). Boldface indicates best in column. PSbaseline = preceding sentence baseline. S = syntactic type features, C = context features, E = embedding level features, SC = subordinating conjunction features, V = verb features, L = length features, LX = lexical features.

fact (472 instances)

Features	**S@1**	**S@2**	**S@3**	**S@4**
{E,SC}	.24	.38	.53	**.65**
{V}	**.43**	**.45**	**.53**	.56
PSbaseline	.40	–	–	–

reason (443 instances)

Features	**S@1**	**S@2**	**S@3**	**S@4**
{E}	**.57**	**.58**	**.60**	**.62**
{E,V}	.41	.46	.52	.60
PSbaseline	.44	–	–	–

issue (303 instances)

Features	**S@1**	**S@2**	**S@3**	**S@4**
{E}	**.28**	**.32**	**.39**	**.44**
{E,SC,L}	.20	.32	.39	.42
PSbaseline	.19	–	–	–

decision (390 instances)

Features	**S@1**	**S@2**	**S@3**	**S@4**
{SC}	**.28**	**.30**	**.33**	**.35**
{E,SC,LX}	.11	.20	.25	.31
PSbaseline	.21	–	–	–

question (440 instances)

Features	**S@1**	**S@2**	**S@3**	**S@4**
{SC}	**.51**	**.58**	**.67**	**.72**
{E,SC}	.48	.57	.59	.62
PSbaseline	.25	–	–	–

possibility (278 instances)

Features	**S@1**	**S@2**	**S@3**	**S@4**
{SC}	**.31**	**.36**	**.49**	.53
{E,V,L,LX}	.21	.36	.47	**.55**
PSbaseline	**.34**	–	–	–

appear in any of the best performing feature class combination. One reason could be that in case of identifying the precise shell content from the given sentence, the words that tend to occur with a particular shell noun can be used to push down candidates with non-informative words, i.e., words that are not associated with that shell noun. However, the sentences surrounding the shell noun phrase are generally about the same topic and they tend to repeat words a lot, making it hard to distinguish between candidates based on the lexical items.

5.5.2 Identifying precise shell content from the sentence given by the crowd

For these experiments, we assumed that we already have the sentence containing the shell content. We use the crowdsourcing method from Section 5.4.4 that identifies the sentence containing the shell content of the ASN before identifying the precise shell content, and then

given the sentence containing the shell content, we extract all syntactic constituents given by the Stanford parser from that sentence as potential shell content candidates as for the training phase. This results in reduced number of average candidates from $n \times 49.5$ to 49.5.

The results of these ablation experiments are shown in Table 5.8. The results are significantly better than both baselines in all cases. 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 shell content 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 shell content 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.

We compare these results with *this issue* resolution from Chapter 3. For *this issue* resolution in the Medline domain, we observed precision in the range of 0.41 to 0.61. For *this issue* instances from the NYT corpus, we achieved precision in the range of 0.40 to 0.47. Furthermore, we applied the ranking models trained on CSN shell content to resolve *this issue* instances

Table 5.8: Evaluation of our ranker for antecedents of six ASNs. The source of the candidates is the sentence given by the crowd in the first experiment. For each noun we show the three best-performing feature combinations. S@n is the success at rank n (S@1 = standard precision). Boldface indicates best ins column. CSbaseline = crowd sentence baseline. The S@1 results significantly higher than CSbaseline are marked with *(two-sample χ^2 test: $p < 0.05$). The chance baseline results were 0.1, 0.2, 0.3, and 0.4 for S@1, S@2, S@3, and S@4 respectively. S = syntactic type features, C = context features, E = embedding level features, SC = subordinating conjunction features, V = verb features, L = length features, LX = lexical features.

fact (472 instances)

Features	**S@1**	**S@2**	**S@3**	**S@4**
{E,L,LX}	**.70***	.85	.91	.94
{E,V,L,LX}	.68*	**.86**	**.92**	**.95**
{E,SC,L,LX}	.66*	.83	**.92**	**.95**
CSbaseline	.47	–	–	–

reason (443 instances)

Features	**S@1**	**S@2**	**S@3**	**S@4**
{E,V,L}	**.72***	**.86**	**.90**	.93
{E,V}	**.72***	.85	**.90**	.92
{E,SC,LX}	.69*	.84	**.90**	**.94**
CSbaseline	.52	–	–	–

issue (303 instances)

Features	**S@1**	**S@2**	**S@3**	**S@4**
{SC,L}	**.47***	.59	.71	.78
{SC,L,LX}	.46*	.60	.70	**.81**
{S,E,SC,L,LX}	.40*	**.61**	**.72**	**.81**
CSbaseline	.26	–	–	–

decision (390 instances)

Features	**S@1**	**S@2**	**S@3**	**S@4**
{E,LX}	**.35***	**.53**	**.67**	**.76**
{E,SC,LX}	.30*	.48	.65	.75
{E,SC,V,L,LX}	.27	.44	.57	.69
CSbaseline	.29	–	–	–

question (440 instances)

Features	**S@1**	**S@2**	**S@3**	**S@4**
{E,SC,V,L,LX}	**.70***	.82	.87	.90
{E,SC,LX}	.68*	**.83**	**.88**	**.91**
{E,SC,V,LX}	.69*	.80	.87	**.91**
CSbaseline	.38	–	–	–

possibility (278 instances)

Features	**S@1**	**S@2**	**S@3**	**S@4**
{SC,L,LX}	**.56***	.75	**.87**	**.92**
{E,SC}	**.56***	**.76**	**.87**	.91
{E,L,LX}	.54*	**.76**	.86	.91
CSbaseline	.44	–	–	–

from the Medline domain.[14] Even with models trained on automatically labelled data from a completely different domain, we achieved similar results to the *this-issue* resolution results from Chapter 3: S@1 of 0.45, S@2 of 0.59, S@3 of 0.65, and S@4 of 0.67. These results show the domain robustness of these methods with respect to the shell noun *issue*. Recall that in Chapter 3 we looked at only very specific cases of *this issue* and used manually annotated data, as opposed to the automatically extracted CSN shell content data we use here.

[14]We thank an anonymous reviewer for suggesting this to us.

5.6 Discussion and conclusion

The goal of this section 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 shell content were able to bring the relevant ASN shell content candidates into the top 10 candidates.

The results from section 5.5.1 suggest that when the search space is large, the CSN models do not quite identify the viable shell content candidates successfully, especially for the shell nouns *fact*, *issue*, and *possibility*.

But when we know the sentence containing the shell content, 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 shell content helps in identifying the shell content 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 shell content were in the first four ranks about 76% to 81% of the time, suggesting that in future research, these models can be used as base models to reduce the large search space of ASN shell content 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. 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 shell content. However, ASN shell content has additional properties which are not always captured by CSN shell content. For instance, in most cases in the ASN data, the shell content of the ASN *this decision* was a court decision, and it was expressed with a full sentence. On the other hand, in most cases in the CSN data, the shell content of the CSN *decision*

was expressed as an action with an infinitive phrase. Although we observed reasonable inter-annotator agreement on ASN shell content and validated crowd annotations by experts, it is possible that in some cases, the candidates given by the CSN rankers biased the annotators to select one of the displayed answers rather than selecting *None*. One way to systematically investigate this is by examining whether the instances annotated with high confidence have clear CSN shell content patterns (e.g., *whether X* and *that X*).

Moreover, although the models trained on CSN shell content are able to encode characteristic features associated with the general shell concept, they are unable to address the pragmatic challenges mentioned in Section 1.3.

In addition, we only focused on the frequently occurring anaphoric pattern *this N*. We do not address the anaphoric pattern *th-be-N*. For this pattern, the shell content is typically in the preceding sentence, and can be extracted using the right-frontier rule, i.e., extracting the rightmost clause from the preceding sentence (Webber, 1991; Asher, 1993).

Chapter 6

Summary, Contributions, and Future Directions

6.1 Summary of the approach and main results

The goal of this dissertation was to develop computational methods to resolve shell nouns to their shell content, and to examine whether knowledge and features derived from the linguistic literature help in this process. Accordingly, I have developed algorithms that can resolve a variety of shell nouns, occurring in different constructions. In particular, I approached this problem in four steps: pilot study, resolving cataphoric shell nouns, resolving anaphoric shell nouns, and annotating anaphoric shell nouns.

6.1.1 Pilot study

As explained in Chapter 3, to get a good grasp of the shell noun resolution problem, I carried out a pilot study on annotation and identification of shell content of shell nouns. In this study, I focused on the narrow problem of resolution of anaphoric occurrences of the shell noun *issue* in the medical domain. To understand the phenomenon better, I myself and a domain expert, Dr. Brian Budgell, independently annotated a sample of data representing *this issue*

instances from the Medline domain. I pointed out a number of challenges associated with the task: a variety of syntactic types of shell content, large search space of eligible candidates, non-precise boundaries of the shell content. Nonetheless, we achieved an inter-annotator agreement in terms of Krippendorff's unitizing α of 0.86. With this reliably annotated data, we extracted a number of features primarily from three sources: properties of shell content as discussed in the linguistics literature, features used in resolution of anaphors with similar properties, i.e., *it*, *this*, and *that*, and our observations from annotation. With these features, we trained supervised SVM ranking models that learned rankings of eligible shell content candidates. We applied these models to resolve unseen *this issue* instances from the same domain. We achieved accuracies in the range of 0.41 to 0.61 (baseline = 0.24) on the unseen test data. These results illustrate the feasibility of annotating and resolving shell nouns automatically, at least in the closed domain of Medline abstracts. The results also show that reduction of search space markedly improved the resolution performance, suggesting that a two-stage process that first identifies the broad region of the shell content and then pinpoints the exact shell content might work better than a single-stage approach.

6.1.2 Resolving cataphoric shell nouns

Next, I focused on generalizing shell noun resolution to a variety of shell nouns in a broader newswire domain. As explained in Chapter 4, I approached the problem of resolving cataphoric shell nouns (CSNs). An example is shown in (57).

(57) **The reason** that I'm sounding off on menu-driven computer programs is **that they end up taking more time than the old McBee card system**.

I demonstrated the complexities involved with the resolution of such examples, especially when it comes to developing a general algorithm that can deal with the idiosyncrasies of a variety of shell nouns. I proposed an algorithm that exploits Schmid's semantic classification of shell nouns to identify their shell content. The algorithm was evaluated against the crowd-annotated

data. The results showed that syntax alone is not enough to resolve CSNs. I concluded that a) Schmid's pattern and clausal constraints are useful for resolving nouns with strict syntactic expectations (e.g., *fact* and *reason*); however, the overall framework is incomplete from the automatic resolution perspective, and b) enriching the semantic families with more noun-specific semantic constraints or reorganizing the current semantic frames might help the automatic resolution. Later, I use this method as the basis to resolve anaphoric instances of shell nouns.

6.1.3 Resolving anaphoric shell nouns

In the next phase, as explained in Chapter 5, I focused on the problem of resolving anaphoric shell nouns (ASNs) in the newswire domain. ASNs are common in newswire text and are harder to resolve than CSNs because the shell content can occur anywhere in the given context. The primary challenge was that there was no annotated data available that covered a variety of shell nouns. I developed a machine learning approach that learns properties of CSN shell content for different shell nouns and applies the generalization of these learned properties to predict shell content of ASNs. In particular, I hypothesized that shell content of ASNs and CSNs share linguistic properties, and hence linguistic knowledge encoded in CSN shell content will help in interpreting ASNs. Accordingly, I examined which features present in CSN shell content are relevant in interpreting ASNs.

6.1.4 Annotating anaphoric shell nouns

Next, to evaluate our ASN resolution approach, I built an ASN corpus via crowdsourcing. We divided the ASN annotation task into two steps. In the first step, we labelled the sentences in the vicinity of the shell noun phrase and asked the annotators to select the sentence containing the shell content. In the second step, we asked them to select the precise shell content from the annotated sentence of the previous experiment. We presented them with the 10 highly-ranked candidates predicted by the CSN shell content ranking models, and asked them to choose the right answer from these options. Our final annotated ASN corpus contains 1,810 ASN

instances and their shell content for six frequently occurring shell nouns in the NYT corpus. We compared the crowd's answer with the first 4 predicted answers. Our results suggest that when the search space is large, the CSN models do not accurately predict the viable shell content candidates, especially for the the shell nouns *fact*, *issue*, and *possibility*. When we knew the sentence containing the shell content, we achieved precision in the range of 0.35 (baseline = 0.21) to 0.72 (baseline = 0.44), depending upon the shell noun. The precision results as high as 0.72 for *reason* and 0.70 for *fact* and *question* support our hypothesis that the linguistic knowledge, more specifically syntactic and lexical knowledge, provided by CSN shell content helps in identifying the shell content of ASNs. Although the mental nouns such as *issue* and *decision* in general were harder to interpret than other shell nouns, the shell content was in the first four ranks about 76% to 81% of the time, suggesting that in future research, these models can be used as base models to reduce the large search space of ASN shell content candidates.

6.2 Summary of contributions

This dissertation has three main classes of contributions.

The primary contribution of this work is that it sheds light on shell nouns from a computational linguistics perspective, which was not addressed before in the field.

This dissertation has resulted in three shell noun resolution systems: a specialized system that can resolve instances of *this issue* in the medical domain (Kolhatkar and Hirst, 2012), a system that can resolve cataphoric shell nouns (Kolhatkar and Hirst, 2014), and a system that can resolve anaphoric shell nouns (Kolhatkar et al., 2013b). All these systems outperform the corresponding baseline systems. Similarly the dissertation has resulted in four annotated corpora that can be used to study the phenomenon: the *this issue* antecedent corpus (Kolhatkar and Hirst, 2012), the crowd-annotated CSN corpus, the automatically-annotated CSN corpus, and the ASN corpus (Kolhatkar et al., 2013a). I plan to make these corpora available to other

researchers.

An other important contribution is to the field of abstract anaphora resolution. Recall that the relation between shell noun phrases and their content is similar to abstract anaphora, where an anaphor refers to abstract antecedents, such as facts, propositions, and events. The problem of abstract anaphora resolution has been a daunting problem. Traditional linguistic and psycholinguistic principles, such as gender and number agreement, or reflexive constraints (e.g., the subject and object cannot be coreferential in a simple clause such as *John defended him.*) are not applicable in such cases. This dissertation provides the first step towards resolving abstract anaphora.

Finally, I showed that in most cases, Schmid's lexico-syntactic cues help in the process of shell noun resolution. I have also shown the need for more sophisticated linguistic knowledge about syntactic and semantic preferences of shell nouns.

6.3 Short-term future plans

6.3.1 First identifying sentences containing shell content

A natural extension to the current approach is to develop a system that identifies the sentence containing shell content. Developing such a system is justified by three reasons. First, identifying precise shell content for ASNs is tricky, as boundaries of their shell content are fuzzy. Second, the results described in the previous chapter show that it is possible to get reasonable resolution performance when we know the sentence containing the shell content. In particular, our machine learning ranking models trained on CSN shell content help in identifying viable shell content candidates for the given ASN when the search space of the shell content candidates is limited. When we consider only one sentence as the source of candidates (49.5 candidates on average), about 90% of the time the crowd answer was within the first four rankings of our ranker. So automatically identifying the sentences containing ASN shell content could be an important component of an ASN resolution system. Third, our crowd annotation

experiments suggest that humans are better in identifying the sentence containing the shell content than identifying the precise shell content.

I plan to examine whether a two-stage process of first identifying the sentence containing shell content and then the precise shell content works better than directly identifying the precise shell content in a large search space of shell content candidates. Accordingly, I plan to develop methods to identify the sentence containing the shell content. Although such a task provides only partial answers in the resolution process, they might be useful in discourse parsing, as such answers will be suggestive of a specific kind of discourse relation between two sentences.

Along these lines, we examined whether there is a difference in the level of semantic similarity between the sentence containing the shell noun phrase and the sentence containing its shell content, and the sentence containing the shell noun phrase and other nearby sentences.[1] We calculated sentence similarity between pairs of sentences by combining scores of various WordNet similarity measures as well as Google's word2vec[2]. It was found that the sentence containing shell content typically had greater semantic similarity to the sentence containing shell noun phrase than other candidate sentences. The average of different similarity scores for shell content sentences was 0.55 compared to 0.43 for other candidate sentences.

6.3.2 Combining CSN and ASN shell content data

Now that we have CSN shell content models that assign a high score to viable shell content candidates and a lower score to spurious candidates, and reliably annotated ASN shell content data, the next step is to combine these two resources to build a better performing ASN resolver. A natural extension is to train ranking models using ASN data and to exploit differences between ASNs and CSNs. For instance, CSN examples are essentially different from ASN examples, especially with respect to the context of the shell noun phrase and the shell content. Recall that Eckert and Strube (2000) used the predicative context of the anaphor to

[1] This project was carried out with an undergraduate research assistant Leila Chan Currie.

[2] `https://code.google.com/p/word2vec/`

identify the semantic type of the shell content. Although shell nouns themselves provide the semantic type of the shell content, the predicative context can still be suggestive of other semantic constraints on the shell content. For instance, in (58), we can check whether *allowing* or *selling* is more likely to require *approving*. I plan to exploit the predicative context of the anaphor, for instance by using narrative chains (Chambers and Jurafsky, 2009).

(58) In principle, he said, airlines should be allowed to sell standing-room-only tickets for adults — as long as **this decision** was approved by their marketing departments.

Similarly, we excluded features such as distance features, as they are irrelevant for CSNs because the shell content always occurs in the same sentence following the anaphor. Now that we have annotated training data, the next step is to incorporate the features that are relevant to ASNs irrespective of whether they are relevant to CSNs or not. Accordingly, I plan to incorporate features such as semantic roles and dependency tree features that worked well for *this issue* resolution in Medline abstracts (see Section 3.4.2). Moreover, as we noted in Section 5.6, CSN shell content models can serve as base models for an ASN resolver. So the rankings given by CSN ranking models could be one of the features for the ranking models that will be trained on ASN shell content. With these extensions, we will be able to examine the extent to which the ASN shell content data we have gathered differ from the CSN shell content data and whether this data contains any ASN-specific properties, which are not present in the CSN shell content.

6.3.3 One SVM ranker for all shell nouns

Our current method trains a distinct ranking model for each shell noun. So we have a distinct weight vector for each shell noun. Given a feature vector x for a candidate of a test instance of a shell noun, we score that candidate by multiplying it with the weight vector associated with that shell noun.

$$S = w^T \phi(x) \tag{6.1}$$

This approach is not convenient in practice. So I am working on building one ranking model by stacking together all these weight vectors to create $\hat{w}$ and $\hat{\phi}$, as shown below.[3] The function $\delta(\text{SN} = \textit{noun})$ returns 1 if the *noun* is the given shell noun, else it returns 0. We use the appropriate weight vector from $\hat{w}$, depending upon the value of the δ function.

$$\hat{w} = \begin{bmatrix} w_{fact} \\ w_{reason} \\ w_{issue} \\ \vdots \end{bmatrix}$$

$$\hat{\phi} = \begin{bmatrix} \phi(x)\delta(\text{SN} = \textit{fact}) \\ \phi(x)\delta(\text{SN} = \textit{reason}) \\ \phi(x)\delta(\text{SN} = \textit{issue}) \\ \vdots \end{bmatrix}$$

This formulation will give us the performance similar to the performance we get with distinct models for different shell nouns. But it will allow us to add more information in the weight vector $\hat{w}$ that is common among different shell nouns, which might lead to a better resolution performance.

6.4 Long-term future directions

This dissertation opens a number of new research directions. We discuss some of them below.

6.4.1 Clustering shell nouns with similar semantic expectations

In Chapter 4, we noted that shell nouns are similar to verbs in that they occur in a number of sub-categorization frames and take a number of semantic arguments. An interesting future direction is soft clustering of typical usages of a variety of shell nouns similar to verb clustering

[3] This work is in collaboration with Alexander Schwing.

(Merlo and Stevenson, 2000; Schulte im Walde and Brew, 2002). The primary challenge would be identifying and extracting appropriate features for clustering so that the clusters are useful for identifying shell content. Assuming that we have meaningful clusters of shell noun usages, two important questions would need to be answered: whether these clusters are similar to Schmid's semantic families, and whether it is possible and/or useful to reorganize Schmid's semantic families for automatic resolution purposes.

6.4.2 Identifying shell noun usages

Currently, we assume that if a potential shell noun follows Schmid's lexico-syntactic patterns, then it is in fact a shell noun usage. This assumption has two limitations. First, Schmid's lexico-syntactic patterns only suggest and do not guarantee a shell noun usage. Second, Schmid's list of lexico-syntactic patterns is not comprehensive and does not cover all cases of shell noun occurrences. For the shell noun *idea*, the lexico-grammatical patterns cover more than 80% of the instances. In contrast, for the shell noun *policy*, these patterns cover only 23% of the instances. Among the remaining 77% of the instances that do not follow these patterns, some instances are shell noun usages, whereas others are not. An interesting research question is what linguistic or contextual properties suggest a shell noun usage. With this knowledge, comparing the percentage of the instances that follow Schmid's lexico-syntactic patterns and are in fact shell noun usages to the percentage of the instances that do not follow Schmid's lexico-syntactic patterns and are shell noun usages would give us an idea of the coverage of Schmid's lexico-syntactic patterns.

6.4.3 Identifying shell chains

Our current approach focuses on identifying shell content of the given shell noun phrase. But similar to anaphoric or coreference chains, often the same abstract entity has been referred to several times in a discourse to form a *shell chain*. For instance, example (59) shows the shell chain *that an education that includes good schooling and rich supplementary education*

experiences is the right of all children → it → an idea → it.

(59) The bottom line is **that an education that includes good schooling and rich supplementary education experiences is the right of all children**. **It**'s **an idea** that most political and education leaders would subscribe to in principle. Soon they may have the money to make **it** a reality.

An interesting avenue of research would be identifying such equivalence classes of expressions in a discourse that are essentially referring to the same abstract entity. Similar to *lexical chains* (Halliday and Hasan, 1976), shell chains capture the cohesive structure of text. That said, lexical chains are sequences of semantically related words in a text. On the other hand, shell chains are sequences of expressions referring to the same abstract entity. These expressions are typically sentences, clauses, verb phrases, noun phrases, or pronouns. So distributional co-occurrence information or resources such as WordNet, which are typically used to identify lexical chains, may not help much in identifying shell chains.

Recall that the cognitive status of shell content of shell noun phrases, as described by Gundel et al. (1993), is either activated or lower but not in focus, but the cognitive status of the referents of the pronoun *it* are the focus of the discourse. So identifying shell chains will help computational linguistics applications such as automatic summarization, discourse analysis, and question-answering.

Bibliography

Ron Artstein and Massimo Poesio. Identifying reference to abstract objects in dialogue. In *Proceedings of the 10th Workshop on the Semantics and Pragmatics of Dialogue*, pages 56–63, Potsdam, Germany, 2006.

Ron Artstein and Massimo Poesio. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596, December 2008.

Nicholas Asher. *Reference to Abstract Objects in Discourse*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1993.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet Project. In *Proceedings of the 17th International Conference on Computational Linguistics*, volume 1 of *COLING '98*, pages 86–90, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

Douglas Biber, Stig Johansson, Geoffrey Leech, Susan Conrad, and Edward Finegan. *Longman Grammar of Spoken and Written English*. Pearson ESL, November 1999.

Simon Philip Botley. Indirect anaphora: Testing the limits of corpus-based linguistics. *International Journal of Corpus Linguistics*, 11(1):73–112, 2006.

Donna K. Byron. Annotation of pronouns and their antecedents: A comparison of two domains. Technical report, University of Rochester. Computer Science Department, 2003.

Donna K. Byron. *Resolving pronominal reference to abstract entities*. PhD thesis, Rochester, New York: University of Rochester, 2004.

Miguel Ángel Benítez Castro. *Formal, syntactic, semantic and textual features of English shell nouns*. PhD thesis, University of Granada, 2013.

Jon Chamberlain, Udo Kruschwitz, and Massimo Poesio. Constructing an anaphorically annotated corpus with non-experts: Assessing the quality of collaborative annotations. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, pages 57–62, Suntec, Singapore, August 2009. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 602–610, Suntec, Singapore, August 2009. Association for Computational Linguistics.

Bin Chen, Jian Su, Sinno Jialin Pan, and Chew Lim Tan. A unified event coreference resolution by integrating multiple resolvers. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 102–110, Chiang Mai, Thailand, November 2011.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37, 1960.

Michael Collins. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, 1999.

Jerome Cornfield. A method for estimating comparative rates from clinical data. applications to cancer of the lung, breast, and cervix. *Journal of the National Cancer Institute*, (11): 1269–1275, 1951.

Östen Dahl and Christina Hellman. What happens when we use an anaphor. In *Presentation at the XVth Scandinavian Conference of Linguistics*, Oslo, Norway, 1995.

Pascal Denis and Jason Baldridge. 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 2008. Association for Computational Linguistics.

Stefanie Dipper and Heike Zinsmeister. Towards a standard for annotating abstract anaphora. In *Proceedings of the LREC 2010 workshop on Language Resources and Language Technology Standards*, pages 54–59, Valletta, Malta, 2010.

Stefanie Dipper and Heike Zinsmeister. Annotating abstract anaphora. *Language Resources and Evaluation*, 69:1–16, 2011.

Stefanie Dipper, Christine Rieger, Melanie Seiss, and Heike Zinsmeister. Abstract anaphors in German and English. In *Anaphora Processing and Applications*, volume 7099 of *Lecture Notes in Computer Science*, pages 96–107. Springer Berlin / Heidelberg, 2011.

Greg Durrett and Dan Klein. Easy victories and uphill battles in coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, October 2013. Association for Computational Linguistics.

Miriam Eckert and Michael Strube. Dialogue acts, synchronizing units, and anaphora resolution. *Journal of Semantics*, 17:51–89, 2000.

Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

Vanessa Wei Feng and Graeme Hirst. A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 511–521, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

Charles J. Fillmore. Frames and the semantics of understanding. *Quaderni di Semantica*, 6(2): 222–254, 1985.

John Flowerdew. Signalling nouns in discourse. *English for Specific Purposes*, 22(4):329–346, 2003.

John Flowerdew. Use of signalling nouns in a learner corpus. *International Journal of Corpus Linguistics*, 11:345–362, 2006.

Karën Fort, Adeline Nazarenko, and Sophie Rosset. Modeling the complexity of manual annotation tasks: a grid of analysis. In *24th International Conference on Computational Linguistics*, pages 895–910, 2012.

Gill Francis. The teaching of techniques of lexical cohesion in an ESL setting. pages 325–338, 1988.

Gill Francis. Labelling discourse: An aspect of nominal group lexical cohesion. In M. Coulthard, editor, *Advances in written text analysis*, pages 83–101. Routledge, London, 1994.

Matthew Gerber, Joyce Chai, and Adam Meyers. The role of implicit argumentation in nominal srl. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 146–154, Boulder, Colorado, June 2009. Association for Computational Linguistics.

Ralph Grishman and Beth Sundheim. Message understanding conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, pages 466–471, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.

Barbara J. Grosz and Candace L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, July 1986.

Barbara J. Grosz, Scott Weinstein, and Aravind K. Joshi. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21:203–225, 1995.

Jeanette K. Gundel, Nancy Hedberg, and Ron Zacharski. Cognitive status and the form of referring expressions in discourse. *Language*, 69(2):274–307, June 1993.

M. A. K. Halliday and Ruqaiya Hasan. *Cohesion in English*. Longman Publication Group, 1976.

Nancy Hedberg, Jeanette K. Gundel, and Ron Zacharski. Directly and indirectly anaphoric demonstrative and personal pronouns in newspaper articles. In *Proceedings of DAARC-2007 8th Discourse Anaphora and Anaphora Resolution Colloquium*, pages 31–36, 2007.

Eli Hinkel. *Teaching Academic ESL Writing: Practical Techniques in Vocabulary and Grammar (ESL and Applied Linguistics Professional)*. Lawrence Erlbaum, Mahwah, NJ, London, 2004.

Barbora Hladká, Jiří Mírovský, and Pavel Schlesinger. Play the language: Play coreference. In *Proceedings of the Association of Computational Linguistics and International Joint Conference on Natural Language Processing 2009 Conference Short Papers*, pages 209–212, Suntec, Singapore, August 2009. Association for Computational Linguistics.

Jerry Hobbs. Resolving pronoun references. *Lingua*, 44:311–338, 1978.

Pei-Yun Hsueh, Prem Melville, and Vikas Sindhwani. Data quality from crowdsourcing: A study of annotation selection criteria. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 27–35, Boulder, Colorado, June 2009. Association for Computational Linguistics.

Rodney D. Huddleston and Geoffrey K. Pullum. *The Cambridge Grammar of the English Language*. Cambridge University Press, April 2002.

Roz Ivanic. 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, 1991.

Thorsten Joachims. Optimizing search engines using clickthrough data. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 133–142, 2002.

Shafiq Joty, Giuseppe Carenini, Raymond Ng, and Yashar Mehdad. Combining intra- and multi-sentential rhetorical parsing for document-level discourse analysis. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 486–496, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

Lauri Karttunen. Discourse referents. *Syntax and Semantics 7: Notes from the Linguistic Underground*, pages 363–385, 1976.

Varada Kolhatkar and Graeme Hirst. 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 2012. Association for Computational Linguistics.

Varada Kolhatkar and Graeme Hirst. Resolving shell nouns. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, page to appear, Doha, Qatar, October 2014. Association for Computational Linguistics.

Varada Kolhatkar, Heike Zinsmeister, and Graeme Hirst. 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 2013a. Association for Computational Linguistics.

Varada Kolhatkar, Heike Zinsmeister, and Graeme Hirst. Interpreting anaphoric shell nouns using antecedents of cataphoric shell nouns as training data. In *Proceedings of the 2013*

Conference on Empirical Methods in Natural Language Processing, pages 300–310, Seattle, Washington, USA, October 2013b. Association for Computational Linguistics.

Klaus Krippendorff. *Content Analysis: An Introduction to Its Methodology*. Sage, Thousand Oaks, CA, second edition, 2004.

Klaus Krippendorff. *Content Analysis: An Introduction to Its Methodology*. Sage, Thousand Oaks, CA, third edition, 2013.

Shalom Lappin and Herbert J. Leass. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20:535–561, 1994.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 489–500, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

John Lyons. *Semantics, II*. Cambridge University Press, Cambridge, 1977.

Nitin Madnani, Jordan Boyd-Graber, and Philip Resnik. Measuring transitivity using untrained annotators. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 188–194, Los Angeles, June 2010. Association for Computational Linguistics.

Paola Merlo and Suzanne Stevenson. Automatic verb classification based on statistical distributions of argument structure. *Computational Linguistics*, 27(3):373–408, 2000.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. The nombank project: An interim report. In *In Proceedings of the NAACL/HLT Workshop on Frontiers in Corpus Annotation*, 2004.

Natalia N. Modjeska. *Resolving Other-Anaphora*. PhD thesis, School of Informatics, University of Edinburgh, 2003.

Christoph Müller. Resolving *it*, *this*, and *that* in unrestricted multi-party dialog. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 816–823, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Christoph Müller. *Fully Automatic Resolution of It, This and That in Unrestricted Multi-Party Dialog*. PhD thesis, Universität Tübingen, 2008.

Costanza Navarretta. Antecedent and referent types of abstract pronominal anaphora. In *Proceedings of the Workshop Beyond Semantics: Corpus-based investigations of pragmatic and discourse phenomena*, Göttingen, Germany, Feb 2011.

C. D. Paice and G. D. Husk. Towards the automatic recognition of anaphoric features in english text: the impersonal pronoun 'it'. *Computer Speech and Language*, 2:109–132, 1987.

Rebecca J. Passonneau. Getting at discourse referents. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, pages 51–59, Vancouver, British Columbia, Canada, 1989. Association for Computational Linguistics.

Rebecca J. Passonneau. Protocol for coding discourse referential noun phrases and their antecedents. Technical report, Columbia University, 1994.

Massimo Poesio and Ron Artstein. Anaphoric annotation in the ARRAU corpus. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA).

Massimo Poesio and Natalia N. Modjeska. The THIS-NPs hypothesis: A corpus-based investigation. In *Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Conference (DAARC 2002)*, pages 157–162, Lisbon, Portugal, September 2002.

Massimo Poesio, Amrita Patel, and Barbara Di Eugenio. Discourse structure and anaphora in tutorial dialogues: An empirical analysis of two theories of the global focus. *Research on Language and Computation*, 4:229–257, 2005.

Massimo Poesio, Simone Ponzetto, and Yannick Versley. Computational models of anaphora resolution: A survey. Unpublished, 2011.

Livia Polanyi. A theory of discourse structure and discourse coherence. In *Proceedings of the 21st Meeting of the Chicago Linguistics Society*, 1985.

Sameer S. Pradhan, Lance A. Ramshaw, Ralph M. Weischedel, Jessica MacBride, and Linnea Micciulla. Unrestricted coreference: Identifying entities and events in OntoNotes. In *Proceedings of the International Conference on Semantic Computing*, pages 446–453, September 2007.

Judita Preiss, Ted Briscoe, and Anna Korhonen. A system for large-scale acquisition of verbal, nominal and adjectival subcategorization frames from corpora. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 912–919, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Ellen F. Prince. Toward a taxonomy of given-new information. 14:223–255, 1981.

Hans-Jörg Schmid. *English Abstract Nouns As Conceptual Shells: From Corpus to Cognition*. Topics in English Linguistics 34. Mouton de Gruyter, Berlin, 2000.

Sabine Schulte im Walde and Chris Brew. Inducing German semantic verb classes from purely syntactic subcategorisation information. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 223–230, Philadelphia, PA, 2002.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, Hawaii, October 2008. Association for Computational Linguistics.

Leonard Talmy. The windowing of attention. In *Toward a Cognitive Semantics*, volume 1, pages 257–309. The MIT Press, 2000.

Zeno Vendler. *Adjectives and Nominalizations.* Mouton and Co., The Netherlands, 1968.

Renata Vieira, Susanne Salmon-Alt, Caroline Gasperin, Emmanuel Schang, and Gabriel Othero. Coreference and anaphoric relations of demonstrative noun phrases in multilingual corpus. In *Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Conference (DAARC 2002)*, pages 385–427, Lisbon, Portugal, September 2002.

Aobo Wang, Cong Duy Vu Hoang, and Min-Yen Kan. Perspectives on crowdsourcing annotations for natural language processing. In *Language Resources and Evaluation*, volume in press, pages 1–23. Springer, 2012.

Bonnie Lynn Webber. *A Formal Approach to Discourse Anaphora*. Garland, 1979.

Bonnie Lynn Webber. Discourse deixis: Reference to discourse segments. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, pages 113–122, Buffalo, New York, USA, June 1988. Association for Computational Linguistics.

Bonnie Lynn Webber. Structure and ostension in the interpretation of discourse deixis. *Language and Cognitive Processes*, 14:107–135, 1991.

Eugene Winter. A clause-relational approach to English texts: A study of some predictive lexical items in written discourse. *Instructional Science*, 6(1):1–92, 1977.

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

Appendix A

List of shell nouns from Schmid (2000)

ability	absurdity	acceptance	accident	account
achievement	acknowledgement	act	action	adage
advantage	advice	affirmation	age	agenda
aim	allegation	allegory	alternative	amazement
amendment	analogy	analysis	anger	announcement
anomaly	answer	anticipation	anxiety	aphorism
application	appointment	appreciation	apprehension	approach
argument	arrangement	art	aspect	assertion
asset	assignment	assumption	assurance	astonishment
attitude	attraction	attribute	audacity	authority
axiom	bargain	basis	beauty	belief
bet	betting	bid	bitterness	blessing
boast	bonus	breakthrough	brief	burden
calculation	call	campaign	capability	capacity
catastrophe	catch	catchphrase	cause	caveat
challenge	chance	change	characteristic	charge
cheek	choice	claim	cliche	clout

coincidence	comfort	command	comment	commission
compensation	complaint	complication	compliment	compromise
concept	conception	concern	concession	conclusion
confession	confidence	confirmation	conjecture	connection
consequence	consideration	consolation	conspiracy	constraint
context	contract	contradiction	contrast	contribution
convention	conviction	corollary	counterclaim	countermeasure
courage	courtesy	credo	crime	criterion
critique	crusade	crux	cure	curiosity
custom	danger	deal	debacle	decency
decision	declaration	decree	deduction	defence
delight	delusion	demand	demonstration	denial
desire	destiny	determination	diagnosis	dictum
difficulty	dilemma	directive	disadvantage	disappointment
disclosure	discovery	discrepancy	disgrace	disgust
disposition	disquiet	distinction	distortion	doctrine
doubt	downside	drama	drawback	dread
drive	duty	eagerness	edict	effect
effrontery	endeavour	energy	enigma	enterprise
era	error	essence	estimate	ethos
evidence	example	exception	excuse	expectation
explanation	extent	facility	fact	factor
faith	fallacy	fantasy	farce	fate
fear	feature	feeling	fiction	fight
flaw	flexibility	folly	forecast	foresight
foundation	franchise	freedom	frustration	function
gall	gambit	gamble	generalization	goal

grace	gratitude	grief	grievance	gripe
grudge	grumble	guarantee	guess	guilt
habit	handicap	happiness	heart	hint
hunch	hurdle	hypothesis	idea	ideal
image	impact	imperative	impetus	implication
impression	improvement	impulse	inability	incapacity
inclination	inconsistency	indication	indicator	indignation
inevitability	inference	information	initiative	injunction
innovation	insight	insistence	inspiration	instinct
intelligence	intent	intention	interest	interpretation
intuition	invitation	irony	irritation	issue
joke	joy	judgement	justification	keenness
knack	knowledge	lament	law	leeway
legend	lesson	licence	lie	likelihood
line	link	logic	longing	luck
manifestation	manoeuvre	marvel	maxim	measure
message	metaphor	method	miracle	misapprehension
misconception	misfortune	misjudgment	misperception	mission
moment	motion	motivation	motive	motto
mystery	myth	necessity	need	nerve
nightmare	nonsense	norm	notice	notification
nous	novelty	nuisance	oath	object
objective	obligation	observation	obsession	obstacle
oddity	offence	offer	opinion	opportunity
order	orthodoxy	outcome	pact	pain
paranoia	part	passion	payoff	peculiarity
period	permission	permit	perspective	persuasion

phenomenon	philosophy	pity	place	plan
pleasure	pledge	plot	ploy	point
position	possibility	potential	power	practice
precaution	precept	preconception	precondition	predicament
preface	preference	prejudice	premise	premonition
presentiment	pressure	presumption	presupposition	pretence
pride	principle	priority	privilege	prize
problem	procedure	proclamation	prognosis	programme
projection	promise	pronouncement	proof	propensity
proposal	proposition	prospect	protest	proverb
proviso	provocation	punishment	purpose	puzzle
query	quest	question	quibble	race
rationale	reaction	readiness	reading	realisation
reason	reasoning	reassurance	recipe	reckoning
recollection	recommendation	refinement	reflection	refusal
regret	relief	reluctance	remark	remedy
reply	report	reposte (ri ~)	request	requirement
reservation	resistance	resolution	resolve	response
restriction	result	retort	revelation	revolution
right	risk	ritual	role	room
rule	ruling	rumour	ruse	rush
satisfaction	scandal	scenario	scheme	scope
sensation	sentiment	sequel	shame	shock
signal	significance	similarity	sin	site
skill	slogan	snag	solace	solution
space	speciality	speculation	spot	stage
stance	stand	standpoint	statement	step

stipulation	story	strategy	strength	struggle
subtext	success	suggestion	superstition	supposition
surprise	suspicion	symbol	symptom	tactic
talent	talk	target	task	teaching
temerity	temptation	tendency	tenet	terror
testimony	theme	theory	thesis	thing
thought	threat	thrill	time	tip
tradition	tragedy	travesty	trend	trick
trouble	truism	truth	twist	uncertainty
undertaking	unknown	unwillingness	upshot	urge
verdict	version	view	viewpoint	virtue
vocation	vow	warning	way	weakness
whisper	willingness	willpower	wisdom	wish
wonder	worry	yearning	zeal	

Appendix B

Family-Shell Nouns Mapping

Ability	ability, authority, capability, capacity, clout, failure, inability, incapacity, potential, power, skill, talent
Adage	adage, adage, allegory, aphorism, catchphrase, joke, nonsense, preface, proverb, subtext
Advantage	advantage, asset, benefit, blessing, bonus
Advice	recommendation, tip
Agreement	agreement, appointment, arrangement, bargain, compromise, consensus, contract, contract, deal, pact
Aim	aim, ambition, ambition, goal, hope, idea, ideal, interest, object, object, objective, point, target, vision
Argument	acceptance, affirmation, argument, concession, concession, counterclaim, justification, reaction
Aspect	aspect, attribute, characteristic, characteristic, distinction, essence, factor, feature, point
Assessment	assessment, judgement, verdict

Attempt	attempt, campaign, conspiracy, countermeasure, crusade, effort, endeavour, enterprise, fight, gamble, initiative, manoeuvre, measure, move, plot, ploy, precaution, quest, race, ruse, rush, struggle, test, test, trick, venture
Belief	assumption, belief, calculation, confidence, confidence, conjecture, conviction, estimate, expectation, feeling, hope, hunch, idea, impression, inkling, instinct, intuition, knowledge, premise, presumption, presupposition, presupposition, prospect, prospect, speculation, superstition, supposition, surmise, suspicion, understanding
Certainty	certainty, fact, reality, truth
Complaint	complaint, grievance, gripe, grumble, quibble, whinge
Compliment	boast, compliment, excuse, lament, praise
Condition	case, condition, condition, constraint, criterion, criterion, criterion, event, limitation, precondition, provision, proviso, proviso, restriction, stipulation
Desire	concern, desire, dream, inclination, intent, intention, longing, willingness, wish, yearning
Destiny	destiny, fate
Determination	anxiety, audacity, cheek, confidence, confidence, courage, courtesy, decency, determination, eagerness, eagerness, effrontery, energy, flexibility, foresight, gall, grace, gumption, heart, impetus, keenness, motivation, nerve, nous, passion, readiness, resistance, resolution, resolve, resolve, stamina, strength, temerity, willpower, wit, zeal, zeal
Difference	alternative, analogy, contrast, contrast, difference, discrepancy, distinction, inconsistency, similarity
Disclosure	disclosure, revelation
Doubt	doubt, question

Event	act, action, change, event, position, situation
Evidence	clue, corollary, demonstration, demonstration, demonstration, evidence, finding, implication, indication, indicator, intimation, manifestation, proof, reminder, reminder, sign, signal, symbol, symptom
Example	example, exception
Fear	anxiety, apprehension, concern, disquiet, dread, fear, premonition, reservation, worry
Guess	allegation, claim, contention, guess, hint, suggestion
Idea	axiom, concept, credo, doctrine, dogma, hypothesis, idea, image, issue, law, logic, maxim, metaphor, motto, myth, notion, point, position, precept, principle, rationale, rule, rule, scenario, secret, stereotype, subject, teaching, theme, theory, thesis, thought, topic, wisdom
Illusion	deception, delusion, fallacy, fantasy, fiction, illusion, misapprehension, miscalculation, misconception, misjudgment, misperception
Invitation	advice, appeal, application, invitation, petition, plea
Irony	absurdity, accident, anomaly, anomaly, beauty, change, charm, coincidence, curiosity, fault, folly, importance, inevitability, irony, novelty, oddity, paradox, paranoia, peculiarity, travesty, twist, uncertainty, weakness
Job	agenda, assignment, business, challenge, commission, duty, job, mandate, mission, responsibility, role, task
Lie	distortion, lie, pretext
Link	connection, link
Miracle	attraction, breakthrough, comfort, consolation, luck, marvel, merit, miracle, revolution, sensation, solace, virtue, wonder
Mistake	crime, error, fault, folly, mistake, offence, sin

Motivation	compulsion, impulse, incentive, inducement, inspiration, motivation, motive, preoccupation, temptation, urge, vocation
Mystery	conundrum, enigma, mystery, puzzle, puzzle, question, unknown
Myth	axiom, cliche, credo, dictum, doctrine, dogma, formula, law, legacy, legend, maxim, metaphor, motto, myth, orthodoxy, slogan, stereotype, teaching, tenet, truism
Need	imperative, necessity, need, obligation, pressure, requirement
News	argument, information, intelligence, message, news, point, report, story
Offer	bid, offer
Opportunity	approach, area, chance, facility, method, moment, occasion, opportunity, place, position, possibility, region, room, scope, situation, space, stage, step, time, way
Option	alternative, choice, option, preference, priority, speciality
Order	call, command, command, demand, directive, injunction, instruction, motion, order, request
Part	basis, foundation, part
Permission	franchise, freedom, leeway, licence, option, permission, permit, privilege, right
Place	area, place, point, position, region, site, spot
Plan	art, attitude, brief, decision, decision, drive, idea, line, motto, philosophy, plan, policy, principle, programme, project, rationale, routine, rule, rule, scheme, secret, strategy, tactic
Poclamation	announcement, decree, notification, proclamation
Possibility	chance, danger, option, possibility, risk, uncertainty
Prediction	bet, betting, forecast, prediction, prognosis, prophecy
Probability	chance, likelihood, probability

Problem	burden, catch, complication, crux, difficulty, dilemma, disadvantage, downside, drawback, handicap, hurdle, obstacle, point, predicament, problem, snag, thing, trouble
Proclamation	accusation, statement
Promise	assurance, commitment, guarantee, oath, pledge, promise, promise, undertaking, vow
Purpose	function, idea, purpose
Question	query, question
Realisation	analysis, anticipation, appreciation, consideration, deduction, deduction, deduction, deduction, diagnosis, discovery, equation, generalization, inference, insight, insight, interpretation, lesson, projection, reading, realisation, reasoning, reasoning, reckoning, recognition, recollection, reflection, significance
Reason	cause, ground, reason, thing
Reluctance	disinclination, refusal, reluctance, reluctance, unwillingness
Report	account, explanation, explanation, report, story, tale, version
Result	consequence, effect, impact, outcome, payoff, result, result, sequel, upshot
Reward	compensation, prize, punishment, reward
Rumour	gossip, rumour, talk, whisper
Situation	context, position, situation
Solution	cure, key, remedy, solution
Statement	account, assertion, observation, statement
Success	achievement, achievement, coup, improvement, innovation, refinement, success, triumph
Suggestion	proposal, proposition, suggestion

Surprise	amazement, anger, annoyance, astonishment, bitterness, delight, disappointment, disgust, frustration, fury, gratitude, grief, grudge, guilt, happiness, indignation, irritation, joy, pain, pleasure, pride, rage, regret, relief, resentment, sadness, satisfaction, shock, sorrow, surprise, terror, thrill
Tendency	disposition, propensity, tendency, trend
Thing	business, case, fact, phenomenon, point, thing
Threat	caveat, threat, warning
Time	age, era, moment, period, stage, time
Tradition	convention, custom, habit, ritual, tradition
Tragedy	blow, catastrophe, curse, debacle, disaster, disgrace, drama, farce, flaw, misfortune, nightmare, nuisance, offence, pity, scandal, shame, tragedy
Trouble	difficulty, dilemma, problem, snag, trouble
View	attitude, awareness, conception, conviction, ethos, experience, faith, idea, instinct, line, logic, notion, obsession, opinion, perception, perspective, persuasion, philosophy, preconception, prejudice, presentiment, rationale, sentiment, stance, stand, standpoint, thinking, view, viewpoint
Way	approach, gambit, knack, method, norm, norm, practice, procedure, recipe, technique, trick, way

Appendix C

Family-Patterns Mapping

Ability	N_cl, th_N, th_be_N	to
Adage	N_cl, th_N, th_be_N, N_be_cl	that
Advantage	N_be_cl, th_N, th_be_N, N_cl	that, of
Advice	N_cl, th_N, N_be_cl, th_be_N	to, that
Agreement	N_cl, th_N, th_be_N, N_be_cl	to, that
Aim	N_be_cl, th_N, th_be_N, N_cl	to
Argument	N_cl, th_N, th_be_N, N_be_cl	that
Aspect	th_N, N_be_cl, th_be_N	that, of
Assessment	th_N, N_cl, N_be_cl, th_be_N	that
Attempt	N_cl, th_N, th_be_N, N_be_cl	to
Belief	N_cl, th_N, th_be_N, N_be_cl	that
Certainty	N_be_cl, N_cl, th_N, th_be_N	that
Complaint	th_N, N_be_cl, N_cl, th_be_N	that
Compliment	th_N, N_cl, N_be_cl, th_be_N	that
Condition	th_N, N_cl, th_be_N, N_be_cl	that
Desire	N_cl, N_be_cl, th_N, th_be_N	to, that
Destiny	th_N, N_be_cl, th_be_N	to

Determination	N_cl, th_N, th_be_N	to
Difference	N_be_cl, th_N, th_be_N, N_cl	that
Disclosure	N_cl, th_N, N_be_cl, th_be_N	that
Doubt	N_cl, N_be_cl, th_N, th_be_N	wh
Event	th_N, th_be_N, N_be_cl	to
Evidence	N_be_cl, th_N, N_cl, th_be_N	that
Example	th_N, th_be_N, N_be_cl	that
Fear	N_cl, th_N, N_be_cl, th_be_N	to, that
Guess	N_cl, th_N, N_be_cl, th_be_N	that
Idea	th_N, N_cl, N_be_cl, th_be_N	that, of
Illusion	th_N, th_be_N, N_cl, N_be_cl	
Invitation	th_N, N_cl, th_be_N	to, that
Irony	N_be_cl, th_N, th_be_N, N_cl	that
Job	N_cl, N_be_cl, th_N, th_be_N	to
Lie	th_N, N_cl, N_be_cl, th_be_N	that
Link	th_N, N_cl, N_be_cl, th_be_N	that
Miracle	th_N, N_cl, th_be_N, N_be_cl	that
Mistake	th_N, N_be_cl, th_be_N, N_cl	to
Motivation	N_cl, th_N, N_be_cl, th_be_N	to
Mystery	th_N, th_be_N, N_be_cl, N_cl	wh
Myth	N_cl, th_N, th_be_N, N_be_cl	that
Need	N_cl, th_N, N_be_cl, th_be_N	to
News	N_cl, N_be_cl, th_N, th_be_N	that
Offer	N_cl, th_N, th_be_N	to, that
Opportunity	N_cl, th_N, th_be_N	to
Option	N_be_cl, th_N, th_be_N	to
Order	th_N, N_cl, th_be_N	to, that

Part	th_N, th_be_N, N_cl, N_be_cl	that, of
Permission	N_cl, th_N, th_be_N	to
Place	th_N, N_cl, th_be_N	wh
Plan	N_cl, th_N, th_be_N, N_be_cl	to, that
Poclamation	N_cl, th_N, th_be_N, N_be_cl	that
Possibility	N_cl, th_N, th_be_N, N_be_cl	that
Prediction	N_cl, th_N, N_be_cl, th_be_N	that
Probability	N_cl, N_be_cl, th_N, th_be_N	that
Problem	N_be_cl, th_N, th_be_N	that, of
Proclamation	N_cl, th_N, th_be_N, N_be_cl	that
Promise	N_cl, th_N, th_be_N, N_be_cl	to, that
Purpose	th_N, N_be_cl, th_be_N, N_cl	to
Question	th_N, N_be_cl, N_cl, th_be_N	wh
Realisation	th_N, N_cl, N_be_cl, th_be_N	that
Reason	N_be_cl, th_N, N_cl, th_be_N	that, because
Reluctance	N_cl, th_N	to
Report	th_N, N_cl, th_be_N, N_be_cl	that
Result	N_cl, N_be_cl, th_N, th_be_N	that
Reward	th_N, th_be_N, N_be_cl	that
Rumour	N_cl, th_N, th_be_N, N_be_cl	that
Situation	th_N, N_cl, th_be_N	wh
Solution	N_be_cl, th_N, th_be_N	to, that
Statement	N_cl, th_N, th_be_N, N_be_cl	that
Success	th_N, N_be_cl, th_be_N, N_cl	to
Suggestion	N_cl, th_N, N_be_cl, th_be_N	to, that
Surprise	N_cl, th_N, th_be_N, N_be_cl	that
Tendency	N_cl, th_N, th_be_N	to

Thing	N_cl, N_be_cl, th_N, th_be_N	that
Threat	N_cl, th_N, th_be_N, N_be_cl	to, that
Time	th_N, N_cl, th_be_N, N_be_cl	wh
Tradition	th_N, th_be_N, N_cl, N_be_cl	to
Tragedy	th_N, th_be_N, N_cl, N_be_cl	that
Trouble	N_be_cl, th_N, th_be_N	to
View	N_cl, th_N, N_be_cl, th_be_N	that
Way	th_N.N_cl, N_be_cl, th_be_N	to

Appendix D

Annotation guidelines for *this issue* annotation

We followed the guidelines below for annotating *this issue* instances in Medline abstracts.

Namely Test Following Dipper and Zinsmeister (2010), we recommend using the "namely test" for identifying the correct antecedent. Start reading aloud the sentence containing the anaphor. Add a *namely* clause after the anaphor and look for the appropriate text that fits best in the namely clause, as in the following example.

(60) And Jones warned **that with wired products, the longer the cable, the more the sound quality can degrade**. An option that avoids **this problem**, he said, is a wireless connection.

Namely test: ...this problem, namely that with wired products, the longer the cable, the more the sound quality can degrade.

Split Antecedents The antecedent might not always be a single span of text. Mark disconnected spans if necessary. For example, in (61), the phrase *it appears* is not really a part of the reason so we mark a split antecedent.

(61) Ms. Anderson, the performance artist, is preoccupied by the epic form as well. Her "Songs and Stories From Moby Dick," which is to be presented by the Brooklyn Academy of Music later this season, amounts to her own alternative telling of Melville's classic American novel. One senses

that Ms. Anderson is still in pursuit of the elusive core of her multi-media spectacle. What she presented at Spoleto were intriguing songs and scenes that combine her personal response to the novel with bits of whale trivia that may not have been known to Melville.

Her goal, it appears, **is to elucidate what is relevant to a modern audience in "Moby-Dick."** It must be for **this reason** that Ms. Anderson includes a passage of her own devising that would probably do Greenpeace proud, a lecture on how sperm whales communicate and how they got that unusual name. (It dates back, Ms. Anderson explains, to a misunderstanding over the consistency of the whale's brain.)

Closest Antecedent If there is more than one antecedent, mark the closest antecedent to the noun phrase. Mark only the words that are sufficient to be a meaningful antecedent.

Paraphrase Often the actual referent is not explicitly stated in the text and the resolution process requires the reader to infer the actual antecedent from the context and his/her common-sense knowledge of the world. We call such inferred referents as *paraphrased* referents. In (62), the actual referent *lack of garage space* is only implicitly stated in the marked text *garage space*. We mark the textual antecedent in such cases and write the paraphrase to clarify the intended meaning.

(62) On a recent Friday, Mr. Ferraro of Avis stood in a steamy garage and described the problem of keeping up with weekend demand. In car rental parlance this is called fleet management, and it is a nightmare in Manhattan, where the primary problem is **garage space**.

"We can hold 40 or 50 cars," said Mr. Ferraro, who, like his counterparts at other companies, was deliberately unspecific to avoid tipping off the competition. "But we are renting hundreds today."

Avis and other big rental car companies solve **this problem** by paying 30 to 50 drivers to shuttle autos in from their airport and suburban locations, which is cheaper than renting more parking space.

Inflected Forms When marking the antecedents, do not be oversensitive to plurals and verb tenses. In (63), for example, you would mark the textual antecedent *plotted a special route for evening walks to avoid canine sniffer units* even though the precise antecedent would be *to plot a special route for evening walks to avoid canine sniffer units*.

(63) Such raids, in which the police regularly make hauls of five kilos of cocaine with a street value of $150,000, occur so often that some residents have adjusted their routines. A couple that owned a dog **plotted a special route for evening walks to avoid canine sniffer units**. **This decision** came after one tense night when the dog almost got into a fight with an unleashed Rottweiler, which was circling a mound of white powder at the corner deli.

Extra Information Often an antecedent is accompanied by extra information in the form of a reason, time, location, actor, and patient, as in shown in the following examples. We skip such extra information and mark the minimal antecedent. For example, in (64), *the lure of attractive salaries in the high-tech world should be the basis for passing on college* is the reason for the decision, but the actual decision is *to pass on college*.

(64) The computer fields are growing at a torrid pace, according to government figures. In the last decade, there has been a 17 percent annual employment growth among computer systems analysts, a broad category that includes network administrators, Web designers, computer security professionals and computer scientists. That figure compares with an overall employment growth of 1.5 percent annually in the same period.

Salaries in computer-related fields reflect the demand for workers that has accompanied this growth. According to the Census Department's Current Population Survey, the median income in 1999 for computer systems analysts was $1,008 a week and the median income for computer programmers was $898 a week. That compares with an overall median wage of $550 a week, or $29,000 annually.

Still, not everyone is convinced that the lure of attractive salaries in the high-tech world should be the basis for deciding whether **to pass on college**. Students who make **this decision** often face skepticism from teachers and parents.

Shell nouns such as *decision* usually have an agent (who took the decision) and patient (the decision about whom). In (65), *the director of U.S.I.A., acting on behalf of the President* is the agent of the marked antecedent. The preceding sentence also includes information about when the decision was taken (May 7, 1990). We do not include such information in the marked antecedent.

(65) "Poor Peru Stands By as Its Rich Past Is Plundered" (news article, Aug. 25) mentions the United States, Japan and Western Europe as the principal markets for looted material from Peru. However, the United States, as a signatory of the 1970 Unesco Convention on unauthorized international

movement of cultural property, can restrict import of such material when a state that is party to that convention requests relief.

The Government of Peru requested the United States to institute a ban on the importation into the United States of Moche artifacts from the Sipan region. My committee, which examines and makes recommendations on such requests to the director of the United States Information Agency, recommended this restriction. On May 7, 1990, the director of U.S.I.A., acting on behalf of the President, **imposed an emergency import ban on such artifacts**. To the best of my knowledge, **this decision** has virtually stopped their illegal importation into this country.

Insufficient Context If more context is needed to help you identify the correct antecedent, click on the article link at the bottom of the text which will take you the complete article.

Appendix E

Annotation Guidelines for Resolving CSNs

Below are the instructions provided for annotating shell content of CSNs. According to CrowdFlower, the annotator satisfaction level for the instructions was 4.2 out of 5.

Overview

Imagine that you are reading the following sentence to a friend.

> **The primary reason that the archdiocese cannot pay teachers more is that its students cannot afford higher tuition.**

Your friend is somewhat distracted and after finishing the sentence, she asks, please say again what was the primary *reason*? Your response would be something like *that its students cannot afford higher tuition*.

This task is about identifying a clause or a phrase that provides interpretation to such abstract nouns in the given context.

We Provide

1. Sentences from New York Times articles containing abstract nouns such as *plan*, *reason*, *issue*, and *idea*.
2. A number of options for the interpretation of the abstract noun.
3. A box for your comments.

Here is an example.

> **The primary reason that the archdiocese cannot pay teachers more is that its students cannot afford higher tuition.**
>
> (a) that the archdiocese cannot pay teachers more is that its students cannot afford higher tuition
>
> (b) that its students cannot afford higher tuition
>
> (c) that the archdiocese cannot pay teachers more
>
> (d) higher tuition
>
> (e) None of the above
>
> (Select one of the above options that provides meaning to the underlined abstract noun in orange.)
>
> Comments?

Process

Your task is to select the appropriate option that provides interpretation to the highlighted abstract noun in the given context.

1. **Read** the sentence carefully.
2. **Identify** the interpretation of the highlighted abstract noun yourself.
3. **Examine** the list of options.
4. **Select** the option that matches your answer.
5. **Write** your comments in the *Comments* box.

Tips and Examples

Select the shortest possible but complete answer

Always select the shortest possible answer that describes the interpretation of the abstract noun completely. For instance, (a) is not the right answer above, as it is not the shortest possible answer -- it includes the reason as well as the effect, i.e., the circumstantial information of the abstract noun *reason*. That said, make sure that your answer is complete and includes all details about the interpretation of the abstract noun. For instance, in the following example, select the full clause *to change the world and get even richer -- but somewhat more slowly* as the correct answer instead of the partial answer *to change the world and get even richer*.

> **This time, given the size and scope of the energy market, the idea is to change the world and get even richer -- but somewhat more slowly.**
>
> to change the world and get even richer✘
>
> to change the world and get even richer -- but somewhat more slowly✔

Ignore minor variations

Sometimes, one of the candidates matches your answer, but it has some minor variations such as an extra punctuation mark. Ignore such minor variations while selecting your answer.

Be careful while selecting 'None of the above' option

Select this answer in two cases.

First, the actual representation is not present in the given sentence, as shown in the example below.

> **They have good reason to worry.**
>
> to worry✘

None of the above✔

Second, the the interpretation is present in the given sentence, but not listed in the options, as shown below.

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

that the archdiocese cannot pay teachers more✘

higher tuition✘

None of the above✔

In both these cases select *None of the above* option. It'll be great if you can write in the comments whether the answer is actually present in the given sentence or not.

Do not select circumstantial information

While selecting your answer, be careful with the circumstantial information of the abstract noun. For instance, do not select *that the archdiocese cannot pay teachers more* as the interpretation of *reason* in the following example because it is just the effect of the reason and not the actual reason.

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

that the archdiocese cannot pay teachers more✘

that its students cannot afford higher tuition✔

Do not select incomplete answers

Sometimes the candidate represents the correct answer only partially. Although we ask you to select shortest possible answers, do not select such partial answers. For instance, *higher tuition* is incorrect interpretation of *reason* in the above example because it is an incomplete answer.

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

higher tuition ✘

that its students cannot afford higher tuition✔

Summary

The task is about identifying correct interpretation of the abstract noun in the given sentence. We ask you to select the answer as follows. If the interpretation of the abstract noun is

1. **present in the given sentence and in the given options**, select the appropriate option.
2. **not present** in the given sentence or **not present in the given options** select *None of the above*.

Thank You!

Thanks for your hard work, and hope you enjoy annotating our task! Your annotations will be used to evaluate a computer program that identifies such interpretation automatically.

Appendix F

Annotation guidelines for annotating ASNs

F.1 CrowdFlower experiment 1

1. This task is about interpreting phrases such as *this fact*, *this idea*, and *this issue*. Such phrases cannot be interpreted in isolation. They require the help of the preceding (sometimes following) text for proper interpretation. For example:

   (66) (a2) Freud repeatedly stressed the significance of infant sexuality. (a1) The mores of his time were hardly receptive. (b) It would take decades for this idea to gain widespread acceptance.

   Here, we interpret the phrase *this idea* with the help of the phrase *the significance of infant sexuality* from the sentence labelled as (a2).

   In this task, you will see a few excerpts from New York Times articles with highlighted phrases. Each sentence in the vicinity of such highlighted phrases will be labelled (e.g., (b), (a1), (a2)). You'll also see the title of the corresponding New York Times article at the top. Your job is to identify the sentence in the presented text that provides an interpretation for the highlighted phrase and to select the appropriate sentence label from the *Label* list. For instance, you'll select a2 as the correct answer in the above example.

2. The following test might help you identify the interpretation of phrases such as *this fact* and *this issue*. Start reading aloud the sentence containing the phrase. Add a namely clause after the phrase and look for the appropriate text that fits best in the namely clause. For instance, the test for the above example will be: *this idea, namely the significance of infant sexuality*.

3. If you think no labelled sentence has the interpretation for the highlighted phrase, select *None*.

4. If you think the interpretation for the highlighted phrase spans more than one labelled sentences, select *Combination*

5. Feel free to write your comments in the *Any comments* text box.

6. If you need more context to help you identify the correct interpretation, click on *Article Link* at the bottom of the text which will open the complete article.

7. Happy annotating :)!

F.2 CrowdFlower experiment 2

1. Imagine that you are reading the following text to a friend.

   (67) Freud repeatedly stressed the significance of infant sexuality. The mores of his time were hardly receptive. It would take decades for this idea to gain widespread acceptance.

   Your friend is somewhat distracted and when you say *this idea*, she asks, what idea? Your response would be something like *the idea, namely the significance of infant sexuality*.

2. This task is about identifying the meaning of phrases such as *this issue*, *this idea*, and *this fact*. You will see excerpts from New York Times articles with highlighted phrases (in blue). Sentences containing the meaning of these phrases will also be highlighted (in orange). Your job is to identify the exact parts of the sentences that provide these meanings. We suggest a list of possible answers and you have to select the appropriate

answer from this list. For the above example, for instance, you'll select the second answer, *the significance of infant sexuality*, as the correct answer from the following list.

(a) Freud repeatedly stressed the significance of infant sexuality.

(b) the significance of infant sexuality

(c) stressed the significance of infant sexuality

3. If you think the correct answer is not there in the suggested options, select the option that is closest to your answer.

4. If no answer in the provided options makes sense to you, select *None*.

5. If you need more context to help you identify the correct answer, you can access the complete article by clicking on the headline at the top.

6. Tell us how satisfied were you with the provided options by selecting one of the three choices: *Satisfied*, *Partially satisfied*, or *Unsatisfied*.

7. If you have any comments about our task, please write them in the *Comments* text box.

8. Happy annotating :)!