Classifying Non-Sentential Utterances in Dialogue: A Machine Learning Approach

Raquel Fernández*
Potsdam University

Jonathan Ginzburg**
King's College London

Shalom Lappin†
King's College London

In this article we use well-known machine learning methods to tackle a novel task, namely the classification of non-sentential utterances (NSUs) in dialogue. We introduce a fine-grained taxonomy of NSU classes based on corpus work, and then report on the results of several machine learning experiments. First, we present a pilot study focused on one of the NSU classes in the taxonomy—bare wh*-phrases or "sluices"—and explore the task of disambiguating between the different readings that sluices can convey. We then extend the approach to classify the full range of NSU classes, obtaining results of around an 87% weighted F-score. Thus our experiments show that, for the taxonomy adopted, the task of identifying the right NSU class can be successfully learned, and hence provide a very encouraging basis for the more general enterprise of fully processing NSUs.*

1. Introduction

Non-sentential utterances (NSUs)—fragmentary utterances that do not have the form of a full sentence according to most traditional grammars, but that nevertheless convey a complete clausal meaning—are a common phenomenon in spoken dialogue. The following are two examples of NSUs taken from the dialogue transcripts of the British National Corpus (BNC) (Burnard 2000):

(1) a. A: Who wants Beethoven music?
B: ***Richard and James.*** [BNC: KB8 1024–1025][1]

b. A: It's Ruth's birthday.
B: ***When?*** [BNC: KBW 13116–13117]

* Karl-Liebknecht Strasse 24-25, 14476 Golm, Germany. E-mail: raquel@ling.uni-potsdam.de.
** The Strand, London WC2R 2LS, UK. E-mail: jonathan.ginzburg@kcl.ac.uk.
† The Strand, London WC2R 2LS, UK. E-mail: shalom.lappin@kcl.ac.uk.

1 This notation indicates the name of the file and the sentence numbers in the BNC.

Submission received: 24 September 2004; revised submission received: 10 November 2006; accepted for publication: 9 March 2007.

Arguably the most important issue in the processing of NSUs concerns their resolution, that is, the recovery of a full clausal meaning from a form which is standardly considered non-clausal. In the first of the examples, the NSU in bold face is a typical "short answer," which despite having the form of a simple NP would most likely be understood as conveying the proposition *Richard and James want Beethoven music*. The NSU in (1b) is an example of what has been called a "sluice." Again, despite being realized by a bare *wh*-phrase, the meaning conveyed by the NSU could be paraphrased as the question *When is Ruth's birthday?*

Although short answers and short queries like those in (1) are perhaps two of the most prototypical NSU classes, recent corpus studies (Fernández and Ginzburg 2002; Schlangen 2003) show that other less well-known types of NSUs—each with its own resolution constraints—are also pervasive in real conversations. This variety of NSU classes, together with their inherent concise form and their highly context-dependent meaning, often make NSUs ambiguous. Consider, for instance, example (2):

(2) a. A: I left it on the table.
B: ***On the table.***

b. A: Where did you leave it?
B: ***On the table.***

c. A: I think I put it er...
B: ***On the table.***

d. A: Should I put it back on the shelf?
B: ***On the table.***

An NSU like B's response in (2a) can be understood either as a clarification question or as an acknowledgment, depending on whether it is uttered with raising intonation or not. In (2b), on the other hand, the NSU is readily understood as a short answer, whereas in (2c) it fills a gap left by the previous utterance. Yet in the context of (2d) it will most probably be understood as a sort of correction or a "helpful rejection," as we shall call this kind of NSU later on in this article.

As different NSU classes are typically related to different resolution constraints, in order to resolve NSUs appropriately systems need to be equipped in the first place with the ability of identifying the intended kind of NSU. How this ability can be developed is precisely the issue we address in this article. We concentrate on the task of automatically classifying NSUs, which we approach using machine learning (ML) techniques. Our aim in doing so is to develop a classification model whose output can be fed into a dialogue processing system—be it a full dialogue system or, for instance, an automatic dialogue summarization system—to boost its NSU resolution capability.

As we shall see, to run the ML experiments we report in this article, we annotate our data with small sets of *meaningful* features, instead of using large sets of arbitrary features as is common in some stochastic approaches. We do this with the aim of obtaining a better understanding of the different classes of NSUs, their distribution, and their properties. For training, we use four machine learning systems: the rule induction learner SLIPPER (Cohen and Singer 1999), the memory-based learner TiMBL (Daelemans et al. 2003), the maximum entropy algorithm MaxEnt (Le 2003), and the Weka toolkit (Witten and Frank 2000). From the Weka toolkit we use the J4.8 decision tree learner, as well as a majority class predictor and a one-rule classifier to derive baseline systems that help us to evaluate

the difficulty of the classification task and the ML results obtained. The main advantage of using several systems that implement different learning techniques is that this allows us to factor out any algorithm-dependent effects that may influence our results.

The article is structured as follows. In Section 2, we introduce the taxonomy of NSU classes we adopt, present a corpus study done using the BNC, and give an overview of the theoretical approach to NSU resolution we assume. After these introductory sections, in Section 3 we present a pilot study that focuses on bare *wh*-phrases or sluices. This includes a small corpus study and a preliminary ML experiment that concentrates on disambiguating between the different interpretations that sluices can convey. We obtain very encouraging results: around 80% weighted F-score (an 8% improvement over a simple one-rule baseline). After this, in Section 4, we move on to the full range of NSUs. We present our main experiments, whereby the ML approach is extended to the task of classifying the full range of NSU classes in our taxonomy. The results we achieve on this task are decidedly positive: around an 87% weighted F-score (a 25% improvement over a four-rule baseline where only four features are used). Finally, in Section 5, we offer conclusions and some pointers for future work.

2. A Taxonomy of NSUs

We propose a taxonomy that offers a comprehensive inventory of the kinds of NSUs that can be found in conversation. The taxonomy includes 15 NSU classes. With a few modifications, these follow the corpus-based taxonomy proposed by Fernández and Ginzburg (2002). In what follows we exemplify each of the categories we use in our work and characterize them informally.

Clarification Ellipsis (CE). We use this category to classify reprise fragments used to clarify an utterance that has not been fully comprehended.

(3) a. A: There's only two people in the class
B: ***Two people?*** [BNC: KPP 352–354]

b. A: [. . .] You lift your crane out, so this part would come up.
B: ***The end?*** [BNC: H5H 27–28]

Check Question. This NSU class refers to short queries, usually realized by conventionalized forms like *alright?* and *okay?*, that are requests for explicit feedback.

(4) A: So <*pause*> I'm allowed to record you.
Okay?
B: Yes. [BNC: KSR 5–7]

Sluice. We consider as sluices all *wh*-question NSUs, thereby conflating under this form-based NSU class *reprise* and *direct* sluices like those in (5a) and (5b), respectively.[2] In the taxonomy of Fernández and Ginzburg (2002) reprise sluices are classified as CE. In the taxonomy used in the experiments we report in this article, however, CE only includes clarification fragments that are not bare *wh*-phrases.

2 This distinction is due to Ginzburg and Sag (2001). More on it will be discussed in Section 2.2.

(5) a. A: Only wanted a couple weeks.
B: ***What?*** [BNC: KB1 3311–3312]

b. A: I know someone who's a good kisser.
B: ***Who?*** [BNC: KP4 511–512]

Short Answer. This NSU class refers to typical responses to (possibly embedded) *wh*-questions (6a)/(6b). Sometimes, however, *wh*-questions are not explicit, as in the context of a short answer to a CE question, for instance (6c).

(6) a. A: Who's that?
B: ***My Aunty Peggy.*** [BNC: G58 33–35]

b. A: Can you tell me where you got that information from?
B: ***From our wages and salary department.*** [BNC: K6Y 94–95]

c. A: Vague and?
B: ***Vague ideas and people.*** [BNC: JJH 65–66]

Plain Affirmative Answer and Plain Rejection. The typical context of these two classes of NSUs is a polar question (7a), which can be implicit as in CE questions like (7b). As shown in (7c), rejections can also be used to respond to assertions.

(7) a. A: Did you bring the book I told you?
B: ***Yes./ No.***

b. A: That one?
B: ***Yeah.*** [BNC: G4K 106–107]

c. A: I think I left it too long.
B: ***No no.*** [BNC: G43 26–27]

Both plain affirmative answers and rejections are strongly indicated by lexical material, characterized by the presence of a '*yes*' word (*yeah*, *aye*, *yep*...) or the negative interjection *no*.

Repeated Affirmative Answer. We distinguish plain affirmative answers like the ones in (7) from repeated affirmative answers like the one in (8), which respond affirmatively to a polar question by verbatim repetition or reformulation of (a fragment of) the query.

(8) A: Did you shout very loud?
B: ***Very loud***, yes. [BNC: JJW 571-572]

Helpful Rejection. The context of helpful rejections can be either a polar question or an assertion. In the first case, they are negative answers that provide an appropriate alternative (9a). As responses to assertions, they correct some piece of information in the previous utterance (9b).

(9) a. A: Is that Mrs. John <*last or full name*>?
B: ***No, Mrs. Billy.*** [BNC: K6K 67-68]

b. A: Well I felt sure it was two hundred pounds a, a week.
B: ***No fifty pounds ten pence per person.*** [BNC: K6Y 112–113]

Plain Acknowledgment. The class plain acknowledgment refers to utterances (like *yeah*, *mhm*, *ok*) that signal that a previous declarative utterance was understood and/or accepted.

(10) A: I know that they enjoy debating these issues.
B: ***Mhm.*** [BNC: KRW 146–147]

Repeated Acknowledgment. This class is used for acknowledgments that, as repeated affirmative answers, also repeat a part of the antecedent utterance, which in this case is a declarative.

(11) A: I'm at a little place called Ellenthorpe.
B: ***Ellenthorpe.*** [BNC: HV0 383–384]

Propositional and Factual Modifiers. These two NSU classes are used to classify propositional adverbs like (12a) and factual adjectives like (12b), respectively, in stand-alone uses.

(12) a. A: I wonder if that would be worth getting?
B: ***Probably not.*** [BNC: H61 81–82]

b. A: There's your keys.
B: ***Oh great!*** [BNC: KSR 137–138]

Bare Modifier Phrase. This class refers to NSUs that behave like adjuncts modifying a contextual utterance. They are typically PPs or AdvPs.

(13) A: [. . .] they got men and women in the same dormitory!
B: ***With the same showers!*** [BNC: KST 992–996]

Conjunct. This NSU class is used to classify fragments introduced by conjunctions.

(14) A: Alistair erm he's, he's made himself coordinator.
B: ***And section engineer.*** [BNC: H48 141–142]

Filler. Fillers are NSUs that fill a gap left by a previous unfinished utterance.

(15) A: [. . .] twenty two percent is er <*pause*>
B: ***Maxwell.*** [BNC: G3U 292–293]

2.1 The Corpus Study

The taxonomy of NSUs presented herein has been tested in a corpus study carried out using the dialogue transcripts of the BNC. The study, which we describe here briefly, supplies the data sets used in the ML experiments we will present in Section 4.

The present corpus of NSUs includes and extends the subcorpus used in Fernández and Ginzburg (2002). It was created by manual annotation of a randomly selected section of 200-speaker-turns from 54 BNC files. Of these files, 29 are transcripts of

conversations between two dialogue participants, and 25 files are multi-party transcripts. The total of transcripts used covers a wide variety of domains, from free conversation to meetings, tutorials and training sessions, as well as interviews and transcripts of medical consultations. The examined subcorpus contains 14,315 sentences. Sentences in the BNC are identified by the CLAWS segmentation scheme (Garside 1987) and each unit is assigned an identifier number.

We found a total of 1,299 NSUs, which make up 9% of the total of sentences in the subcorpus. These results are in line with the rates reported in other recent corpus studies of NSUs: 11.15% in (Fernández and Ginzburg 2002), 10.2% in (Schlangen and Lascarides 2003), 8.2% in (Schlangen 2005).[3]

The NSUs found were labeled according to the taxonomy presented previously together with an additional class *Other* introduced to catch all NSUs that did not fall in any of the classes in the taxonomy. All NSUs that could be classified with the taxonomy classes were additionally tagged with the sentence number of their antecedent utterance. The NSUs not covered by the classification only make up 1.2% (16 instances) of the total of NSUs found. Thus, with a rate of 98.8% coverage, the present taxonomy offers a satisfactory coverage of the data.

The labeling of the entire corpus of NSUs was done by one expert annotator. To assess the reliability of the annotation, a small study with two additional, non-expert annotators was conducted. These annotated a total of 50 randomly selected instances (containing a minimum of two instances of each NSU class as labeled by the expert annotator) with the classes in the taxonomy. The agreement obtained by the three annotators is reasonably good, yielding a κ score of 0.76. The non-expert annotators were also asked to identify the antecedent sentence of each NSU. Using the expert annotation as a gold standard, they achieved 96% and 92% accuracy in this task.

The distribution of NSU classes that emerged after the annotation of the subcorpus is shown in detail in Table 1. By far the most common class can be seen to be Plain Acknowledgment, which accounts for almost half of all NSUs found. This is followed in frequency by Short Answer (14.5%) and Plain Affirmative Answer (8%). CE is the most common class among the NSUs that denote questions (i.e., CE, Sluice, and Check Question), making up 6.3% of all NSUs found.

2.2 Resolving NSUs: Theoretical Background and Implementation

The theoretical background we assume with respect to the resolution of NSUs derives from the proposal presented in Ginzburg and Sag (2001), which in turn is based on the theory of context developed by Ginzburg (1996, 1999).

Ginzburg and Sag (2001) provide a detailed analysis of a number of classes of NSUs—including Short Answer, Sluice, and CE—couched in the framework of Head-driven Phrase Structure Grammar (HPSG). They take NSUs to be first-class grammatical constructions whose resolution is achieved by combining the contribution of the NSU phrase with contextual information—concretely, with the current **question under discussion**, or QUD, which roughly corresponds to the current conversational topic.[4]

3 For a comparison of our NSU taxonomy and the one proposed by Schlangen (2003), see Fernández (2006).

4 An anonymous reviewer asked about the distinction between NSUs that are meaning complete and those which are not. In fact we take all NSUs to be interpreted as full propositions or questions.

Table 1
Distribution of NSU classes.

NSU class	Total	%
Plain Acknowledgment	599	46.1
Short Answer	188	14.5
Plain Affirmative Answer	105	8.0
Repeated Acknowledgment	86	6.6
Clarification Ellipsis	82	6.3
Plain Rejection	49	3.7
Factual Modifier	27	2.0
Repeated Affirmative Answer	26	2.0
Helpful Rejection	24	1.8
Check Question	22	1.7
Filler	18	1.4
Bare Modifier Phrase	15	1.1
Propositional Modifier	11	0.8
Sluice	21	1.6
Conjunct	10	0.7
Other	16	1.2
Total	**1,299**	**100**

The simplest way of exemplifying this strategy is perhaps to consider a direct short answer to an explicit *wh*-question, like the one shown in (16a).

(16) a. A: Who's making the decisions?
B: ***The fund manager.*** (= *The fund manager is making the decisions.*)
[BNC: JK7 119–120]

b. QUD: $\lambda(x).Make\_decision(x,t)$
Resolution: $Make\_decision(fm,t)$

In this dialogue, the current QUD corresponds to the content of the previous utterance—the *wh*-question *Who's making the decisions?* Assuming a representation of questions as lambda abstracts, the resolution of the short answer amounts to applying this question to the phrasal content of the NSU, as shown in (16b) in an intuitive notation.[5]

Ginzburg and Sag (2001) distinguish between direct and reprise sluices. For direct sluicing, the current QUD is a polar question *p?*, where *p* is required to be a quantified proposition.[6] The resolution of the direct sluice consists in constructing a *wh*-question by a process that replaces the quantification with a simple abstraction. For instance:

(17) a. A: A student phoned.
B: ***Who?*** (= *Which student phoned?*)

5 To simplify matters, throughout the examples in this section we use lambda abstraction for *wh*-questions and a simple question mark operator for polar questions. For a far more accurate representation of questions in HPSG and Type Theory with Records, see Ginzburg and Sag (2001) and Ginzburg (2005), respectively.

6 In Ginzburg's theory of context an assertion of a proposition *p* raises the polar question *p*? for discussion.

b. QUD: ?$\exists x Phone(x, t)$
Resolution: $\lambda(x).Phone(x, t)$

In the case of reprise sluices and CE, the current QUD arises in a somewhat less direct way, via a process of utterance *coercion* or *accommodation* (Larsson 2002; Ginzburg and Cooper 2004), triggered by the inability to ground the previous utterance (Traum 1994; Clark 1996). The output of the coercion process is a question about the content of a (sub)utterance which the addressee cannot resolve. For instance, if the original utterance is the question *Did Bo leave?* in (18a), with *Bo* as the unresolvable sub-utterance, one possible output from the coercion operations defined by Ginzburg and Cooper (2004) is the question in (18b), which constitutes the current QUD, as well as the resolved content of the reprise sluice in (18a).

(18) a. A: Did Bo leave?
B: ***Who?*** (= *Who are you asking if s/he left?*)

b. QUD: $\lambda(b).Ask(A, ?Leave(b, t))$
Resolution: $\lambda(b).Ask(A, ?Leave(b, t))$

The interested reader will find further details of this approach to NSU resolution and its extension to other NSU classes in Ginzburg (forthcoming) and Fernández (2006).

The approach sketched here has been implemented as part of the SHARDS system (Ginzburg, Gregory, and Lappin 2001; Fernández et al., in press), which provides a procedure for computing the interpretation of some NSU classes in dialogue. The system currently handles short answers, direct and reprise sluices, as well as plain affirmative answers to polar questions. SHARDS has been extended to cover several types of clarification requests and used as a part of the information-state-based dialogue system CLARIE (Purver 2004b). The dialogue system GoDiS (Larsson et al. 2000; Larsson 2002) also uses a QUD-based approach to handle short answers.

3. Pilot Study: Sluice Reading Classification

The first study we present focuses on the different interpretations or readings that sluices can convey. We first describe a corpus study that aims at providing empirical evidence about the distribution of sluice readings and establishing possible correlations between these readings and particular sluice types. After this, we report the results of a pilot machine learning experiment that investigates the automatic disambiguation of sluice interpretations.

3.1 The Sluicing Corpus Study

We start by introducing the corpus of sluices. The next subsections describe the annotation scheme, the reliability of the annotation, and the corpus results obtained.

Because sluices have a well-defined surface form—they are bare *wh*-words—we were able to use an automatic mechanism to reliably construct our subcorpus of sluices. This was created using SCoRE (Purver 2001), a tool that allows one to search the BNC using regular expressions.

Table 2
Total of sluices in the BNC.

what	*why*	*who*	*where*	*which N*	*when*	*how*	*which*	Total
3,045	1,125	491	350	160	107	50	15	**5,343**

The dialogue transcripts of the BNC contain 5,183 bare sluices (i.e., 5,183 sentences consisting of just a *wh*-word). We distinguish between the following classes of bare sluices: *what*, *who*, *when*, *where*, *why*, *how*, and *which*. Given that only 15 bare *which* were found, we also considered sluices of the form *which N*. Including *which N*, the corpus contains a total of 5,343 sluices, whose distribution is shown in Table 2.

For our corpus study, we selected a sample of sluices extracted from the total found in the dialogue transcripts of the BNC. The sample was created by selecting all instances of bare *how* (50) and bare *which* (15), and arbitrarily selecting 100 instances of each of the remaining sluice classes, making up a total of 665 sluices.

Note that the sample does not reflect the frequency of sluice types found in the full corpus. The inclusion of sufficient instances of the lesser frequent sluice types would have involved selecting a much larger sample. Consequently it was decided to abstract over the true frequencies to create a balanced sample whose size was manageable enough to make the manual annotation feasible. We will return to the issue of the true frequencies in Section 3.1.3.

3.1.1 Sluice Readings. The sample of sluices was classified according to a set of four semantic categories—drawn from the theoretical distinctions introduced by Ginzburg and Sag (2001)—corresponding to different sluice interpretations. The typology reflects the basic direct/reprise divide and incorporates other categories that cover additional readings, including an *Unclear* class intended for those cases that cannot easily be classified by any of the other categories. The typology of sluice readings used was the following:

Direct. Sluices conveying a direct reading query for additional information that was explicitely or implicitly quantified away in the antecedent, which is understood without difficulty. The sluice in (19) is an example of a sluice with direct reading: It asks for additional temporal information that is implicitly quantified away in the antecedent utterance.

(19) A: I'm leaving this school.
B: ***When?*** [BNC: KP3 537–538]

Reprise. Sluices conveying a reprise reading emerge as a result of an understanding problem. They are used to clarify a particular aspect of the antecedent utterance corresponding to one of its constituents, which was not correctly comprehended. In (20) the reprise sluice has as antecedent constituent the pronoun *he*, whose reference could not be adequately grounded.

(20) A: What a useless fairy he was.
B: ***Who?*** [BNC: KCT 1752–1753]

Clarification. As reprise, this category also corresponds to a sluice reading that deals with understanding problems. In this case the sluice is used to request clarification of the entire antecedent utterance, indicating a general breakdown in communication. The following is an example of a sluice with a clarification interpretation:

(21) A: Aye and what money did you get on it?
B: ***What?***
A: What money does the government pay you? [BNC: KDJ 1077–1079]

***Wh*-anaphor.** This category is used for the reading conveyed by sluices like (22), which are resolved to a (possibly embedded) *wh*-question present in the antecedent utterance.

(22) A: We're gonna find poison apple and I know where that one is.
B: ***Where?*** [BNC: KD1 2370–2371]

Unclear. We use this category to classify those sluices whose interpretation is difficult to grasp, possibly because the input is too poor to make a decision as to its resolution, as in the following example:

(23) A: <*unclear*> <*pause*>
B: ***Why?*** [BNC: KCN 5007]

3.1.2 Reliability. The coding of sluice readings was done independently by three different annotators. Agreement was moderate ($\kappa = 0.59$). There were important differences among sluice classes: The lowest agreement was on the annotation of *how* (0.32) and *what* (0.36), whereas the agreement on classifying *who* was substantially higher (0.74).

Although the three coders may be considered "experts," their training and familiarity with the data were not equal. This resulted in systematic differences in their annotations. Two of the coders had worked more extensively with the BNC dialogue transcripts and, crucially, with the definition of the categories to be applied. Leaving the third annotator out of the coder pool increases agreement very significantly ($\kappa = 0.71$). The agreement reached by the *more expert* pair of coders was acceptable and, we believe, provides a solid foundation for the current classification.[7]

3.1.3 Distribution Patterns. The sluicing corpus study shows that the distribution of readings is significantly different for each class of sluice. The distribution of interpretations is shown in Table 3, presented as row counts and percentages of those instances where

7 Besides the difficulty of annotating fine-grained semantic distinctions, we think that one of the reasons why the κ score we obtain is not too high is that, as shall become clear in the next section, the present annotation is strongly affected by the *prevalence problem*, which occurs when the distributions for categories are skewed (highly unequal instantiation across categories). In order to control for differences in prevalence, Di Eugenio and Glass (2004) propose an additional measure called *PABAK* (*prevalence-adjusted bias-adjusted kappa*). In our case, we obtain a *PABAK* score of 0.60 for agreement amongst the three coders, and a *PABAK* score of 0.80 for agreement between the pair of more expert coders. A more detailed discussion of these issues can be found in Fernández (2006).

Table 3
Distribution patterns.

Sluice	Direct n (%)	Reprise n (%)	Clarification n (%)	*Wh*-anaphor n (%)
what	7 (9.60)	17 (23.3)	48 (65.7)	1 (1.3)
why	55 (68.7)	24 (30.0)	0 (0)	1 (1.2)
who	10 (13.0)	65 (84.4)	0 (0)	2 (2.6)
where	31 (34.4)	56 (62.2)	0 (0)	3 (3.3)
when	50 (63.3)	27 (34.1)	0 (0)	2 (2.5)
which	1 (8.3)	11 (91.6)	0 (0)	0 (0)
whichN	19 (21.1)	71 (78.8)	0 (0)	0 (0)
how	23 (79.3)	3 (10.3)	3 (10.3)	0 (0)

at least two annotators agree, labeled taking the majority class and leaving aside cases classified as *Unclear*.

Table 3 reveals significant correlations between sluice classes and preferred interpretations (a chi square test yields $\chi^2 = 438.53$, $p \leq 0.001$). The most common interpretation for *what* is Clarification, making up more than 65%. *Why* sluices have a tendency to be Direct (68.7%). The sluices with the highest probability of being Reprise are *who* (84.4%), *which* (91.6), *which N* (78.8%), and *where* (62.2%). On the other hand, *when* (63.3%) and *how* (79.3%) have a clear preference for Direct interpretations.

As explained in Section 3.1, the sample used in the corpus study does not reflect the overall frequencies of sluice types found in the BNC. Now, in order to gain a complete perspective on sluice distribution in the full corpus, it is therefore appropriate to combine the percentages in Table 3 with the absolute number of sluices contained in the BNC. The number of estimated tokens is displayed in Table 4.

For instance, the combination of Tables 3 and 4 allows us to see that although almost 70% of *why* sluices are Direct, the absolute number of *why* sluices that are Reprise exceeds the total number of *when* sluices by almost 3 to 1. Another interesting pattern revealed by this data is the low frequency of *when* sluices, particularly by comparison with what one might expect to be its close cousin, *where*. Indeed the Direct/Reprise splits are almost mirror images for *when* versus *where*. Explicating the distribution in Table 4 is important in order to be able to understand among other issues whether we would expect a similar distribution to occur in a Spanish or Mandarin dialogue corpus; similarly, whether one would expect this distribution to be replicated across different domains.

Table 4
Sluice class frequency (estimated tokens).

$what_{cla}$	2,040	$whichN_{rep}$	135
why_{dir}	775	$when_{dir}$	90
$what_{rep}$	670	who_{dir}	70
who_{rep}	410	$where_{dir}$	70
why_{rep}	345	how_{dir}	45
$where_{rep}$	250	$when_{rep}$	35
$what_{dir}$	240	$whichN_{dir}$	24

We will not attempt to provide an explanation for these patterns here. The reader is invited to check a sketch of such an explanation for some of the patterns exhibited in Table 4 in Fernández, Ginzburg, and Lappin (2004).

3.2 Automatic Disambiguation

In this section, we report a pilot study where we use machine learning to automatically disambiguate between the different sluice readings using data obtained in the corpus study presented previously.

3.2.1 Data. The data set used in this experiment was selected from our classified corpus of sluices. To generate the input data for the ML experiments, all three-way agreement instances plus those instances where there is agreement between the two coders with the highest agreement were selected, leaving out cases classified as Unclear. The total data set includes 351 datapoints. Of these, 106 are classified as Direct, 203 as Reprise, 24 as Clarification, and 18 as *Wh*-anaphor. Thus, the classes in the data set have significantly skewed distributions. However, as we are faced with a very small data set, we cannot afford to balance the classes by leaving out a subset of the data. Hence, in this pilot study the 351 data points are used in the ML experiments with their original distributions.

3.2.2 Features and Feature Annotation. In this pilot study—as well as in the extended experiment we will present later on—instances were annotated with a small set of features extracted automatically using the POS information encoded in the BNC. The annotation procedure involves a simple algorithm which employs string searching and pattern matching techniques that exploit the SGML mark-up of the corpus. The BNC was automatically tagged using the CLAWS system developed at Lancaster University (Garside 1987). The ∼100 million words in the corpus were annotated according to a set of 57 POS codes (known as the C5 tag-set) plus 4 codes for punctuation tags. A list of these codes can be found in Burnard (2000). The BNC POS annotation process is described in detail in Leech, Garside, and Bryant (1994).

Unfortunately the BNC mark-up does not include any coding of intonation. Our features can therefore not use any intonational data, which would presumably be a useful

Table 5
Sluice features and values.

Feature	Description	Values
`sluice`	type of sluice	`what, why, who,...`
`mood`	mood of the antecedent utterance	`decl, n_decl`
`polarity`	polarity of the antecedent utterance	`pos, neg, ?`
`quant`	presence of a quantified expression	`yes, no, ?`
`deictic`	presence of a deictic pronoun	`yes, no, ?`
`proper_n`	presence of a proper name	`yes, no, ?`
`pro`	presence of a pronoun	`yes, no, ?`
`def_desc`	presence of a definite description	`yes, no, ?`
`wh`	presence of a *wh* word	`yes, no, ?`
`overt`	presence of any other potential antecedent expression	`yes, no, ?`

source of information to distinguish, for instance, between question- and proposition-denoting NSUs, between Plain Acknowledgment and Plain Affirmative Answer, and between Reprise and Direct sluices.

To annotate the sluicing data, a set of 11 features was used. An overview of the features and their values is shown in Table 5. Besides the feature `sluice`, which indicates the sluice type, all the other features are concerned with properties of the antecedent utterance. The features `mood` and `polarity` refer to syntactic and semantic properties of the antecedent utterance as a whole. The remaining features, on the other hand, focus on a particular lexical type or construction contained in the antecedent. These features (`quant`, `deictic`, `proper_n`, `pro`, `def_desc`, `wh`, and `overt`) are not annotated independently, but conditionally on the sluice type. That is, they will take `yes` as a value if the element or construction in question appears in the antecedent *and* it matches the semantic restrictions imposed by the sluice type. For instance, when a sluice with value `where` for the feature `sluice` is annotated, the feature `deictic`, which encodes the presence of a deictic pronoun, will take value `yes` only if the antecedent utterance contains a locative deictic like *here* or *there*. Similarly the feature `wh` takes a `yes` value only if there is a *wh*-word in the antecedent that is identical to the sluice type.

Unknown or irrelevant values are indicated by a question mark (`?`) value. This allows us to express, for instance, that the presence of a proper name is irrelevant to determining the interpretation of say a *when* sluice, although it is crucial when the sluice type is *who*. The feature `overt` takes `no` as value when there is no overt antecedent expression. It takes `yes` when there is an antecedent expression not captured by any other feature, and it is considered irrelevant (`?` value) when there is an antecedent expression defined by another feature.

The 351 data points were automatically annotated with the 11 features shown in Table 5. The automatic annotation procedure was evaluated against a manual gold standard, achieving an accuracy of 86%.

3.2.3 Baselines. Because sluices conveying a Reprise reading make up more than 57% of our data set, relatively high results can already be achieved with a majority class baseline that always predicts the class Reprise. This yields a 42.4% weighted F-score.

A slightly more interesting baseline can be obtained by using a one-rule classifier. We use the implementation of a one-rule classifier provided in the Weka toolkit. For each feature, the classifier creates a single rule which generates a decision tree where the root is the feature in question and the branches correspond to its different values. The leaves are then associated with the class that occurs most often in the data, for which that value holds. The classifier then chooses the feature which produces the minimum error.

```
sluice:
- who     -> Reprise
- what    -> Clarification
- why     -> Direct
- where   -> Reprise
- when    -> Direct
- which   -> Reprise
- whichN  -> Reprise
```

Figure 1
One-rule tree.

Table 6
Baselines' results.

	Sluice reading	Recall	Precision	F1
Majority class baseline	Reprise	100	57.80	73.30
	weighted score	**57.81**	**33.42**	**42.40**
One-rule baseline	Direct	72.60	67.50	70.00
	Reprise	79.30	80.50	79.90
	Clarification	100	64.90	78.70
	weighted score	**73.61**	**71.36**	**72.73**

In this case the feature with the minimum error chosen by the one-rule classifier is `sluice`. The classifier produces the one-rule tree in Figure 1. The branches of the tree correspond to the sluice types; the interpretation with the highest probability for each type of sluice is then predicted.

By using the feature `sluice` the one-rule tree implements the correlations between sluice type and preferred interpretation that were discussed in Section 3.1.3. There, we pointed out that these correlations were statistically significant. We can see now that they are indeed a good rough guide for predicting sluice readings. As shown in Table 6, the one-rule baseline dependent on the distribution patterns of the different sluice types yields a 72.73% weighted F-score.

All results reported (here and in the remainder of the article) were obtained by performing 10-fold cross-validation. They are presented as follows: The tables show the recall, precision, and F-measure for each class. To calculate the overall performance of the algorithm, these scores are normalized according to the relative frequency of each class. This is done by multiplying each score by the total of instances of the corresponding class and then dividing by the total number of datapoints in the data set. The weighted overall recall, precision, and F-measure, shown in boldface for each baseline in Table 6, is then the sum of the corresponding weighted scores. For each of the baselines, the sluice readings not shown in the table obtain null scores.

3.2.4 ML Results. Finally, the four machine learning algorithms were run on the data set annotated with the 11 features. Here, as well as in the more extensive experiment we will present in Section 4, we use the following parameter settings with each of the learners. Weka's J4.8 decision tree learner is run using the default parameter settings. With SLIPPER we use the option `unordered`, which finds a rule set that separates each class from the remaining classes using growing and pruning techniques and in our case yields slightly better results than the default setting. As for TiMBL, we run it using the *modified value difference metric* (which performs better than the default *overlap metric*), and keep the default settings for the number of nearest neighbors ($k = 1$) and feature weighting method (*gain ratio*). Finally, with MaxEnt we use 40 iterations of the default L-BFGS parameter estimation (Malouf 2002).

Overall, in this pilot study we obtain results of around 80% weighted F-score, although there are some significant differences amongst the learners. MaxEnt gives the lowest score (73.24% weighted F-score)—hardly over the one-rule baseline, and more than 8 points lower than the best results, obtained with Weka's J4.8 (81.80% weighted F-score). The size of the data set seems to play a role in these differences, indicating that MaxEnt does not perform so well with small data sets. A summary of weighted F-scores is given in Table 7.

Table 7
Comparison of weighted F-scores.

System	Weighted F-score
Majority class baseline	42.40
One rule baseline	72.73
MaxEnt	73.24
TiMBL	79.80
SLIPPER	81.62
J4.8	81.80

Detailed recall, precision, and F-measure results for each learner are shown in Appendix A. The results yielded by MaxEnt are almost equivalent to the ones achieved with the one-rule baseline. With the other three learners, the use of contextual features improves the results for Reprise and Direct by around 5 points each with respect to the one-rule baseline. The results obtained with the one-rule baseline for the Clarification reading, however, are hardly improved upon by any of the learners. In the case of TiMBL the score is in fact lower—72.16 versus 78.70 weighted F-score. This leads us to conclude that the best strategy is to interpret all *what* sluices as conveying a Clarification reading.

The class *Wh*-anaphora, which, not being the majority interpretation for any sluice type, was not predicted by the one-rule baseline nor by MaxEnt, now gives positive results with the other three learners. The best result for this class is obtained with Weka's J4.8: 80% F-score.

The decision tree generated by Weka's J4.8 algorithm is displayed in Figure 2. The root of the tree corresponds to the feature `wh`, which makes a first distinction between

```
wh:
- yes -> Wh_anaphor
- no  -> sluice:
          - what   -> Clarification
          - where  -> Reprise
          - which  -> Reprise
          - whichN -> Reprise
          - when   -> overt:
                        - yes    -> Reprise
                        - no     -> Direct
          - why    -> ant_mood:
                        - decl   -> Direct
                        - n_decl -> Reprise
          - who    -> quant:
                        - yes -> pro:
                                  - yes -> Reprise
                                  - no  -> proper_n:
                                             - yes -> Reprise
                                             - no  -> Direct
                        - no  -> Reprise
```

Figure 2
Weka's J4.8 tree.

Wh-anaphor and the other readings. If the value of this feature is `yes`, the class Wh-anaphor is predicted. A negative value for this feature leads to the feature `sluice`. The class with the highest probability is the only clue used to predict the interpretation of the sluice types `what`, `where`, `which`, and `whichN` in a way parallel to the one-rule baseline. Additional features are used for `when`, `why`, and `who`. A Direct reading is predicted for a `when` sluice if there is no overt antecedent expression, whereas a Reprise reading is preferred if the feature `overt` takes as value `yes`. For `why` sluices the mood of the antecedent utterance is used to disambiguate between Reprise and Direct: If the antecedent is declarative, the sluice is classified as Direct; if it is non-declarative it is interpreted as Reprise. In the classification of `who` sluices three features are taken into account: `quant`, `pro`, and `proper_n`. The basic strategy is as follows: If the antecedent utterance contains a quantifier and neither personal pronouns nor proper names appear, the predicted class is Direct, otherwise the sluice is interpreted as Reprise.

3.2.5 Feature Contribution. Note that not all features are used in the tree generated by Weka's J4.8. The missing features are `polarity`, `deictic`, and `def_desc`. Although they don't make any contribution to the model generated by the decision tree, examination of the rules generated by SLIPPER shows that they are all used in the rule set induced by this algorithm, albeit in rules with low confidence level. Despite the fact that SLIPPER uses all features, the contribution of `polarity`, `deictic`, and `def_desc` does not seem to be very significant. When they are eliminated from the feature set, SLIPPER yields very similar results to the ones obtained with the full set of features: 81.22% weighted F-score *versus* the 81.66% obtained before. TiMBL on the other hand goes down a couple of points, from 79.80% to 77.32% weighted F-score. No variation is observed with MaxEnt, which seems to be using just the sluice type as a clue for classification.

4. Classifying the Full Range of NSUs

So far we have presented a study that has concentrated on fine-grained semantic distinctions of one of the classes in our taxonomy, namely Sluice, and have obtained very encouraging results—around 80% weighted F-score (an improvement of 8 points over a simple one-rule baseline). In this section we show that the ML approach taken can be successfully extended to the task of classifying the full range of NSU classes in our taxonomy.

We first present an experiment run on a restricted data set that excludes the classes Plain Acknowledgement and Check Question, and then, in Section 4.6, report on a follow-up experiment where all NSU classes are included.

4.1 Data

The data used in the experiments was selected from the corpus of NSUs following some simplifying restrictions. Firstly, we leave aside the 16 instances classified as *Other* in the corpus study (see Table 1). Secondly, we restrict the experiments to those NSUs whose antecedent is the immediately preceding utterance. This restriction, which makes the feature annotation task easier, does not pose a significant coverage problem, given that the immediately preceding utterance is the antecedent for the vast majority of NSUs (88%). The set of all NSUs, excluding those classified as *Other*, whose antecedent is the immediately preceding utterance, contains a total of 1123 datapoints. See Table 8.

Table 8
NSU subcorpus.

NSU class	Total
Plain Acknowledgment	582
Short Answer	105
Affirmative Answer	100
Repeated Acknowledgment	80
CE	66
Rejection	48
Repeated Affirmative Answer	25
Factual Modifier	23
Sluice	20
Helpful Rejection	18
Filler	16
Check Question	15
Bare Modifier Phrase	10
Propositional Modifier	10
Conjunct	5
Total data set	**1,123**

Finally, as mentioned previously, the last restriction adopted concerns the instances classified as Plain Acknowledgment and Check Question. Taking the risk of ending up with a considerably smaller data set, we decided to leave aside these meta-communicative NSU classes given that (1) plain acknowledgments make up more than 50% of the subcorpus leading to a data set with very skewed distributions; (2) check questions are realized by the same kind of expressions as plain acknowledgments (*okay*, *right*, etc.) and would presumably be captured by the same feature; and (3) a priori these two classes seem two of the easiest types to identify (a hypothesis that was confirmed after a second experiment—see Section 4.6). We therefore exclude plain acknowledgments and check questions and concentrate on a more interesting and less skewed data set containing all remaining NSU classes. This makes up a total of 526 data points ($1123 - 582 - 15$). In Subsection 4.6 we shall compare the results obtained using this restricted data set with those of a second experiment in which plain acknowledgements and check questions are incorporated.

4.2 Features

A small set of features that capture the contextual properties that are relevant for NSU classification was identified. In particular three types of properties that play an important role in the classification task were singled out. The first one has to do with semantic, syntactic, and lexical properties of the NSUs themselves. The second one refers to the properties of its antecedent utterance. The third concerns relations between the antecedent and the fragment. Table 9 shows an overview of the nine features used.

4.2.1 NSU Features. A set of four features are related to properties of the NSUs. These are `nsu_cont`, `wh_nsu`, `aff_neg`, and `lex`. The feature `nsu_cont` is intended to distinguish between question-denoting (`q` value) and proposition-denoting (`p` value) NSUs. The feature `wh_nsu` encodes the presence of a *wh*-phrase in the NSU—it is primarily introduced to identify Sluices. The features `aff_neg` and `lex` signal the appearance of

Table 9
NSU features and values.

Feature	Description	Values
nsu_cont	content of the NSU (either prop or question)	p, q
wh_nsu	presence of a *wh* word in the NSU	yes, no
aff_neg	presence of a *yes/no* word in the NSU	yes, no, e(mpty)
lex	presence of different lexical items in the NSU	p_mod, f_mod, mod, conj, e
ant_mood	mood of the antecedent utterance	decl, n_decl
wh_ant	presence of a *wh* word in the antecedent	yes, no
finished	(un)finished antecedent	fin, unf
repeat	repeated words in NSU and antecedent	0-3
parallel	repeated tag sequences in NSU and antecedent	0-3

particular lexical items. They include a value e(mpty) which allows us to encode the absence of the relevant lexical items as well. The values of the feature aff_neg indicate the presence of either a *yes* or a *no* word in the NSU. The values of lex are invoked by the appearance of modal adverbs (p_mod), factual adjectives (f_mod), and prepositions (mod) and conjunctions (conj) in initial positions. These features are expected to be crucial to the identification of Plain/Repeated Affirmative Answer and Plain/Helpful Rejection on the one hand, and Propositional Modifiers, Factual Modifiers, Bare Modifier Phrases, and Conjuncts on the other.

Note that the feature lex could be split into four binary features, one for each of its non-empty values. This option, however, leads to virtually the same results. Hence, we opt for a more compact set of features. This also applies to the feature aff_neg.

4.2.2 Antecedent Features. We use the features ant_mood, wh_ant, and finished to encode properties of the antecedent utterance. The first of these features distinguishes between declarative and non-declarative antecedents. The feature wh_ant signals the presence of a *wh*-phrase in the antecedent utterance, which seems to be the best cue for classifying Short Answers. As for the feature finished, it should help the learners identify Fillers. The value unf is invoked when the antecedent utterance has a hesitant ending (indicated, for instance, by a pause) or when there is no punctuation mark signalling a finished utterance.

4.2.3 Similarity Features. The last two features, repeat and parallel, encode similarity relations between the NSU and its antecedent utterance. They are the only numerical features in the feature set. The feature repeat, which indicates the appearance of repeated words between NSU and antecedent, is introduced as a clue to identify Repeated Affirmative Answers and Repeated Acknowledgments. The feature parallel, on the other hand, is intended to capture the particular parallelism exhibited by Helpful Rejections. It signals the presence of sequences of POS tags common to the NSU and its antecedent.

As in the sluicing experiment, all features were extracted automatically from the POS information encoded in the BNC mark-up. However, as with the feature mood in the sluicing study, some features like nsu_cont and ant_mood are *high level* features that do not have straightforward correlates in POS tags. Punctuation tags (that would correspond to intonation patterns in spoken input) help to extract the values of these features, but the correspondence is still not unique. For this reason the automatic

```
aff_neg:
- yes -> AffAns
- no  -> Reject
- e   -> ShortAns
```

Figure 3
One-rule tree.

feature annotation procedure was again evaluated against a small sample of manually annotated data. The feature values were extracted manually for 52 instances (∼10% of the total) randomly selected from the data set. In comparison with this gold standard, the automatic feature annotation procedure achieves 89% accuracy. Only automatically annotated data is used for the learning experiments.

4.3 Baselines

We now turn to examine some baseline systems that will help us to evaluate the classification task. As before, the simplest baseline we can consider is a majority class baseline that always predicts the class with the highest probability in the data set. In the restricted data set used for the first experiment, this is the class Short Answer. The majority class baseline yields a 6.7% weighted F-score.

When a one-rule classifier is run, we see that the feature that yields the minimum error is `aff_neg`. The one-rule baseline produces the one-rule decision tree in Figure 3, which yields a 32.5% weighted F-score (see Table 10). Plain Affirmative Answer is the class predicted when the NSU contains a *yes*-word, Rejection when it contains a *no*-word, and Short Answer otherwise.

Finally, we consider a more substantial baseline that uses the four NSU features. Running Weka's J4.8 decision tree classifier with these features creates a decision tree with four rules, one for each feature used. The tree is shown in Figure 4.

Table 10
Baselines' results.

	NSU Class	Recall	Precision	F1
Majority class baseline	ShortAns	100.00	20.10	33.50
	weighted score	**19.92**	**4.00**	**6.67**
One-rule baseline	ShortAns	95.30	30.10	45.80
	AffAns	93.00	75.60	83.40
	Reject	100.00	69.60	82.10
	weighted score	**45.93**	**26.73**	**32.50**
Four-rule baseline	CE	96.97	96.97	96.97
	Sluice	100.00	95.24	97.56
	ShortAns	94.34	47.39	63.09
	AffAns	93.00	81.58	86.92
	Reject	100.00	75.00	85.71
	PropMod	100.00	100.00	100.00
	FactMod	100.00	100.00	100.00
	BareModPh	80.00	72.73	76.19
	Conjunct	100.00	71.43	83.33
	weighted score	**70.40**	**55.92**	**62.33**

```
nsu_cont:
- q -> wh_nsu:
        - yes    -> Sluice
        - no     -> CE
- p -> lex:
        - conj   -> ConjFrag
        - p_mod  -> PropMod
        - f_mod  -> FactMod
        - mod    -> BareModPh
        - e      -> aff_neg:
                     - yes -> AffAns
                     - no  -> Reject
                     - e   -> ShortAns
```

Figure 4
Four-rule tree.

The root of the tree corresponds to the feature `nsu_cont`. It makes a first distinction between question-denoting (`q` branch) and proposition-denoting NSUs (`p` branch). Not surprisingly, within the `q` branch the feature `wh_nsu` is used to distinguish between Sluice and CE. The feature `lex` is the first node in the `p` branch. Its different values capture the classes Conjunct, Propositional Modifier, Factual Modifier, and Bare Modifier Phrase. The `e(mpty)` value for this feature takes us to the last, most embedded node of the tree, realized by the feature `aff_neg`, which creates a sub-tree parallel to the one-rule tree in Figure 3. This four-rule baseline yields a 62.33% weighted F-score. Detailed results for the three baselines considered are shown in Table 10.

4.4 Feature Contribution

As can be seen in Table 10, the classes Sluice, CE, Propositional Modifier, and Factual Modifier achieve very high F-scores with the four-rule baseline—between 97% and 100%. These results are not improved upon by incorporating additional features nor by using more sophisticated learners, which indicates that NSU features are sufficient indicators to classify these NSU classes. This is in fact not surprising, given that the disambiguation of Sluice, Propositional Modifier, and Factual Modifier is tied to the presence of particular lexical items that are relatively easy to identify (*wh*-phrases and certain adverbs and adjectives), whereas CE acts as a default category within question-denoting NSUs.

There are, however, four NSU classes that are not predicted at all when only NSU features are used. These are Repeated Affirmative Answer, Helpful Rejection, Repeated Acknowledgment, and Filler. Because they are not associated with any leaf in the tree, they yield null scores and therefore don't appear in Table 10. Examination of the confusion matrices shows that around 50% of Repeated Affirmative Answers were classified as Plain Affirmative Answers, whereas the remaining 50%—as well as the overwhelming majority of the other three classes just mentioned—were classified as Short Answer. Acting as the default class, Short Answers achieves the lowest score: 63.09% F-score.

In order to determine the contribution of the antecedent features (`ant_mood`, `wh_ant`, `finished`), as a next step these were added to the NSU features used in the four-rule tree. When the antecedent features are incorporated, two additional NSU classes are predicted. These are Repeated Acknowledgment and Filler, which achieve rather

```
aff_neg:
- yes -> AffAns
- no  -> Reject
- e   -> ant_mood:
          - n_decl -> ShorAns
          - decl   -> finished:
                       - fin    -> RepAck
                       - unfin  -> Filler
```

Figure 5
Node on a tree using NSU and antecedent features.

positive results: 74.8% and 64% F-score, respectively. We do not show the full results obtained when NSU and antecedent features are used together. Besides the addition of these two NSU classes, the results are very similar to those achieved with just NSU features. The tree obtained when the antecedent features are incorporated to the NSU features can be derived by substituting for the last node in the tree in Figure 4 the tree in Figure 5. As can be seen in Figure 5, the features `ant_mood` and `finished` contribute to distinguish Repeated Acknowledgment and Filler from Short Answer, whose F-score consequently rises, from 63.09% to 79%, due to an improvement in precision. Interestingly, the feature `wh_ant` does not have any contribution at this stage (although it will be used by the learners when the similarity features are added). The general weighted F-score obtained when NSU and antecedent features are combined is 77.87%. A comparison of all weighted F-scores obtained will be shown in the next section, in Table 11.

The use of NSU features and antecedent features is clearly not enough to account for Repeated Affirmative Answer and Helpful Rejection, which obtain null scores.

4.5 ML Results

In this section we report the results obtained when the similarity features are included, thereby using the full feature set, and the four machine learning algorithms are trained on the data.

Although the classification algorithms implement different machine learning techniques, they all yield very similar results: around an 87% weighted F-score. The maximum entropy model performs best, although the difference between its results and

Table 11
Comparison of weighted F-scores.

System	Weighted F-score
Majority class baseline	6.67
One rule baseline	32.50
Four rule baseline (NSU features)	62.33
NSU and antecedent features	77.83
Full feature set:	
- SLIPPER	86.35
- TiMBL	86.66
- J4.8	87.29
- MaxEnt	87.75

those of the other algorithms is not statistically significant. Detailed recall, precision, and F-measure scores are shown in Appendix B.

As seen in previous sections, the four-rule baseline algorithm that uses only NSU features yields a 62.33% weighted F-score, whereas the incorporation of antecedent features yields a 77.83% weighted F-score. The best result, the 87.75% weighted F-score obtained with the maximal entropy model using all features, shows a 10% improvement over this last result. As promised, a comparison of the scores obtained with the different baselines considered and all learners used is given in Table 11.

Short Answers achieve high recall scores with the baseline systems (more than 90%). In the three baselines considered, Short Answer acts as the default category. Therefore, even though the recall is high (given that Short Answer is the class with the highest probability), precision tends to be quite low. The precision achieved for Short Answer when only NSU features are used is ∼47%. When antecedent features are incorporated precision goes up to ∼72%. Finally, the addition of similarity features raises the precision for this class to ∼82%. Thus, by using features that help to identify other categories with the machine learners, the precision for Short Answers is improved by around 36%, and the precision of the overall classification system by almost 33%: from 55.90% weighted precision obtained with the four-rule baseline, to the 88.41% achieved with the maximum entropy model using all features.

With the addition of the similarity features (`repeat` and `parallel`), the classes Repeated Affirmative Answer and Helpful Rejection are predicted by the learners. Although this contributes to the improvement of precision for Short Answer, the scores yielded by these two categories are lower than the ones achieved with other classes. Repeated Affirmative Answer achieves nevertheless decent F-score, ranging from 56.96% with SLIPPER to 67.20% with MaxEnt. The feature `wh_ant`, for instance, is used to distinguish Short Answer from Repeated Affirmative Answer. Figure 6 shows one of the sub-trees generated by the feature `repeat` when Weka's J4.8 is used with the full feature set.

The class with the lowest scores is clearly Helpful Rejection. TiMBL achieves a 39.92% F-score for this class. The maximal entropy model, however, yields only a 10.37% F-score. Examination of the confusion matrices shows that ∼27% of Help Rejections were classified as Rejection, ∼26% as Short Answer, and ∼15% as Repeated Acknowledgement. This indicates that the feature `parallel`, introduced to identify this type of NSUs, is not a good enough cue.

```
repeat:
 - =0   -> finished:
            - unf ->    Filler
            - fin ->    parallel:
                         - =0 ->  ShortAns
                         - >0 ->  HelpReject
 - >0   -> ant_mood:
            - decl   -> RepAck
            - n_decl -> repeat:
                        - =1 ->   wh_ant:
                                   - yes ->  ShortAns
                                   - no  ->  RepAffAns
                        - >1 ->   RepAffAns
```

Figure 6
Node on a tree using the full feature set.

```
aff_neg:
- ack -> Ack
- yes -> Ack
- no  -> Reject
- e   -> ShortAns
```

Figure 7
One-rule tree.

4.6 Incorporating Plain Acknowledgment and Check Question

As explained in Section 4.1, the data set used in the experiments reported in the previous section excluded the instances classified as Plain Acknowledgment and Check Question in the corpus study. The fact that Plain Acknowledgment is the category with the highest probability in the subcorpus (making up more than 50% of our total data set—see Table 8), and that it does not seem particularly difficult to identify could affect the performance of the learners by inflating the results. Therefore it was left out in order to work with a more balanced data set and to minimize the potential for misleading results. As the expressions used in plain acknowledgments and check questions are very similar and they would in principle be captured by the same feature values, check questions were left out as well. In a second phase the instances classified as Plain Acknowledgment and Check Question were incorporated to measure their effect on the results. In this section we discuss the results obtained and compare them with the ones achieved in the initial experiment.

To generate the annotated data set an additional value `ack` was added to the feature `aff_neg`. This value is invoked to encode the presence of expressions typically used in plain acknowledgments and/or check questions (*mhm*, *right*, *okay*, etc.). The total data set (1,123 data points) was automatically annotated with the features modified in this way, and the machine learners were then run on the annotated data.

4.6.1 Baselines. Given the high probability of Plain Acknowledgment, a simple majority class baseline gives relatively high results: 35.31% weighted F-score. The feature with the minimum error used to derive the one-rule baseline is again `aff_neg`, this time with the new value `ack` as part of its possible values (see Figure 7). The one-rule baseline yields a weighted F-score of 54.26%.

The four-rule tree that uses only NSU features goes up to a weighted F-score of 67.99%. In this tree the feature `aff_neg` is now also used to distinguish between CE and Check Question. Figure 8 shows the `q` branch of the tree. As the last node of

```
nsu_cont:
 - q ->   wh_nsu:
           - yes  -> Sluice
           - no   -> aff_neg:
                      - ack   -> CheckQu
                      - yes   -> CheckQu
                      - no    -> CE
                      - e     -> CE
```

Figure 8
Node on the four-rule tree.

the four-rule tree now corresponds to the tree in Figure 7, the class Plain Affirmative Answer is not predicted when only NSU features are used.

When antecedent features are incorporated, Plain Affirmative Answers, Repeated Acknowledgments, and Fillers are predicted, obtaining very similar scores to the ones achieved in the experiment with the restricted data set. The feature `ant_mood` is now also used to distinguish between Plain Acknowledgment and Plain Affirmative Answer. The last node in the tree is shown in Figure 9. The combined use of NSU features and antecedent features yields a weighted F-score of 85.44%.

4.6.2 ML Results. As in the previous experiment, when all features are used the results obtained are very similar across learners (around 92% weighted F-score), if slightly lower with Weka's J4.8 (89.53%). Detailed scores for each class are shown in Appendix C. As expected, the class Plain Acknowledgment obtains a high F-score (∼95% with all learners). The F-score for Check Question ranges from 73% yielded by MaxEnt to 90% obtained with SLIPPER. The high score of Plain Acknowledgment combined with its high probability raises the overall performance of the systems almost four points over the results obtained in the previous experiment: from ∼87% to ∼92% weighted F-score. The improvement with respect to the baselines, however, is not as large: we now obtain a 55% improvement over the simple majority class baseline (from 35.31% to 92.21%), whereas in the experiment with the restricted data set the improvement with respect to the majority class baseline is 81% (from 6.67% to 87.75% weighted F-score.).

Table 12 shows a comparison of all weighted F-scores obtained in this second experiment.

It is interesting to note that even though the overall performance of the algorithms is slightly higher than before (due to the reasons mentioned previously), the scores for some NSU classes are actually lower. The most striking cases are perhaps the classes Helpful Rejection and Conjunct, for which the maximum entropy model now gives null scores (see Appendix C). We have already pointed out the problems encountered with Helpful Rejection. As for the class Conjunct, although it yields good results with the other learners, the proportion of this class (0.4%, 5 instances only) is now probably too low to obtain reliable results.

A more interesting case is the class Affirmative Answer, which in TiMBL goes down more than 10 points (from 93.61% to 82.42% F-score). The tree in Figure 7 provides a clue to the reason for this. When the NSU contains a *yes*-word (second branch of the tree) the class with the highest probability is now Plain Acknowledgment, instead of Plain Affirmative Answer as before (see tree in Figure 3). This is due to the fact that,

```
aff_neg:
- ack -> Ack
- yes -> ant_mood:
          - n_decl -> AffAns
          - decl   -> Ack
- no  -> Reject
- e   -> ant_mood:
          - n_decl -> ShorAns
          - decl   -> finished:
                       - fin    -> RepAck
                       - unfin  -> Filler
```

Figure 9
Node on a tree using NSU and antecedent features.

Table 12
Comparison of weighted F-scores.

System	Weighted F-score
Majority class baseline	35.31
One rule baseline	53.03
Four rule baseline (NSU features)	67.99
NSU and antecedent features	85.44
Full feature set:	
- J4.8	89.53
- SLIPPER	92.01
- TiMBL	92.02
- MaxEnt	92.21

at least in English, expressions like *yeah* (considered here as *yes*-words) are potentially ambiguous between acknowledgments and affirmative answers.[8] This ambiguity and the problems it entails are also noted by Schlangen (2005), who addresses the problem of identifying NSUs automatically. As he points out, the ambiguity of *yes*-words is one of the difficulties encountered when trying to distinguish between backchannels (plain acknowledgments in our taxonomy) and non-backchannel fragments. This is a tricky problem for Schlangen as his NSU identification procedure does not have access to the context. Although in the present experiments we do use features that capture contextual information, determining whether the antecedent utterance is declarative or interrogative (which one would expect to be the best clue to disambiguate between Plain Acknowledgement and Plain Affirmative Answer) is not always trivial.

5. Conclusions

In this article we have presented results of several machine learning experiments where we have used well-known machine learning techniques to address the novel task of classifying NSUs in dialogue.

We first introduced a comprehensive NSU taxonomy based on corpus work carried out using the dialogue transcripts of the BNC, and then sketched the approach to NSU resolution we assume.

We then presented a pilot study focused on sluices, one of the NSU classes in our taxonomy. We analyzed different sluice interpretations and their distributions in a small corpus study and reported on a machine learning experiment that concentrated on the task of disambiguating between sluice readings. This showed that the observed correlations between sluice type and preferred interpretation are a good rough guide for predicting sluice readings, which yields a 72% weighted F-score. Using a small set of features that refer to properties of the antecedent utterance, we were able to improve this result by 8%.

In the second part of this article we extended the machine learning approach used in the sluicing experiment to the full range of NSU classes in our taxonomy. In order to work with a more balanced set of data, the first run of this second experiment was carried out using a restricted data set that excluded the classes Plain Acknowledgment

8 Arguably this ambiguity would not arise in French given that, according to Beyssade (2005), in French the expressions used to acknowledge an assertion are different from those used in affirmative answers to polar questions.

and Check Question. We identified a small set of features that capture properties of the NSUs, their antecedents and relations between them, and employed a series of simple baseline methods to evaluate the classification task. The most successful of these consists of a four-rule decision tree that only uses features related to properties of the NSUs themselves. This gives a 62% weighted F-score. Not surprisingly, with this baseline very high scores (over 95%) could be obtained for NSU classes that are defined in terms of lexical or construction types, like Sluice and Propositional/Factual Modifier.

We then applied four learning algorithms to the data set annotated with all features and improved the result of the four-rule baseline by 25%, obtaining a weighted F-score of around 87% for all learners. The experiment showed that the classes that are most difficult to identify are those that rely on relational features, like Repeated Affirmative Answer and especially Helpful Rejection.

In a second run of the experiment we incorporated the instances classified as Plain Acknowledgment and Check Question in the data set and ran the machine learners again. The results achieved are very similar to those obtained in the previous run, if slightly higher due to the high probability of the class Plain Acknowledgment. The experiment did show however a potential confusion between Plain Acknowledgment and Plain Affirmative Answer (observed elsewhere in the literature) that obviously had not shown up in the previous run.

As typically different NSU classes are subjected to different resolution constraints, identifying the correct NSU class is a necessary step towards the goal of fully processing NSUs in dialogue. Our results show that, for the taxonomy we have considered, this task can be successfully learned.

There are, however, several aspects that deserve further investigation. One of them is the choice of features employed to characterize the utterances. In this case we have opted for rather high-level features instead of using simple surface features, as is common in robust approaches to language understanding. As pointed out by an anonymous reviewer, it would be worth exploring to what extent the performance of our current approach could be improved by incorporating more low-level features, for instance by the presence of closed-class function words.

Besides identifying the right NSU class, the processing and resolution of NSUs involves other tasks that have not been addressed in this article and that are subjects of our future research. For instance, we have abstracted here from the issue of distinguishing NSUs from other sentential utterances. In our experiments the input fed to the learners was in all cases a vector of features associated with an utterance that had already been singled out as an NSU. Deciding whether an utterance is or is not an NSU is not an easy task. This has for instance been addressed by Schlangen (2005), who obtains rather low scores (42% F-measure). There is therefore a lot of room for improvement in this respect, and indeed in the future we plan to explore ways of combining the classification task addressed here with the NSU identification task.

Identifying and classifying NSUs are necessary conditions for resolving them. In order to actually resolve them, however, the output of the classifier needs to be fed into some extra module that takes care of this task. A route we plan to take in the future is to integrate our classification techniques with the information state-based dialogue system prototype CLARIE (Purver 2004a), which implements a procedure for NSU resolution based on the theoretical assumptions sketched in Section 2.2. The taxonomy which we have tested and presented here will provide the basis for classifying NSUs in this dialogue processing system. The classification system will determine the templates and procedures for interpretation that the system will apply to an NSU once it has recognized its fragment type.

Appendix A: Detailed ML Results for the Sluice Reading Classification Task

Learner	Sluice Reading	Recall	Precision	F1
Weka's J4.8	Direct	71.70	79.20	75.20
	Reprise	85.70	83.70	84.70
	Clarification	100.00	68.60	81.40
	Wh_anaphor	66.70	100.00	80.00
	weighted score	**81.47**	**82.14**	**81.80**
SLIPPER	Direct	81.01	71.99	76.23
	Reprise	83.85	86.49	85.15
	Clarification	71.17	94.17	81.07
	Wh_anaphor	77.78	62.96	69.59
	weighted score	**81.81**	**81.43**	**81.62**
TiMBL	Direct	78.72	75.24	76.94
	Reprise	83.08	83.96	83.52
	Clarification	75.83	68.83	72.16
	Wh_anaphor	55.56	77.78	64.81
	weighted score	**79.85**	**79.98**	**79.80**
MaxEnt	Direct	65.22	75.56	70.01
	Reprise	85.74	76.38	80.79
	Clarification	89.17	70.33	78.64
	Wh_anaphor	0.00	0.00	0.00
	weighted score	**75.38**	**76.93**	**73.24**

Appendix B: Detailed ML Results for the Restricted NSU Classification Task

	Weka's J4.8			SLIPPER		
NSU Class	Recall	Precision	F1	Recall	Precision	F1
CE	97.00	97.00	97.00	93.64	97.22	95.40
Sluice	100.00	95.20	97.60	96.67	91.67	94.10
ShortAns	89.60	82.60	86.00	83.93	82.91	83.41
AffAns	92.00	95.80	93.90	93.13	91.63	92.38
Reject	95.80	80.70	87.60	83.60	100.00	91.06
RepAffAns	68.00	63.00	65.40	53.33	61.11	56.96
RepAck	85.00	89.50	87.20	85.71	89.63	87.62
HelpReject	22.20	33.30	26.70	28.12	20.83	23.94
PropMod	100.00	100.00	100.00	100.00	90.00	94.74
FactMod	100.00	100.00	100.00	100.00	100.00	100.00
BareModPh	80.00	100.00	88.90	100.00	80.56	89.23
ConjFrag	100.00	71.40	83.30	100.00	100.00	100.00
Filler	56.30	100.00	72.00	100.00	62.50	76.92
weighted score	**87.62**	**87.68**	**87.29**	**86.21**	**86.49**	**86.35**
	TiMBL			MaxEnt		
NSU Class	Recall	Precision	F1	Recall	Precision	F1
CE	94.37	91.99	93.16	96.11	96.39	96.25
Sluice	94.17	91.67	92.90	100.00	95.83	97.87
ShortAns	88.21	83.00	85.52	89.35	83.59	86.37
AffAns	92.54	94.72	93.62	92.79	97.00	94.85
Reject	95.24	81.99	88.12	100.00	81.13	89.58
RepAffAns	63.89	60.19	61.98	68.52	65.93	67.20
RepAck	86.85	91.09	88.92	84.52	81.99	83.24
HelpReject	35.71	45.24	39.92	5.56	77.78	10.37
PropMod	90.00	100.00	94.74	100.00	100.00	100.00
FactMod	97.22	100.00	98.59	97.50	100.00	98.73
BareModPh	80.56	100.00	89.23	69.44	100.00	81.97
ConjFrag	100.00	100.00	100.00	100.00	100.00	100.00
Filler	48.61	91.67	63.53	62.50	90.62	73.98
weighted score	**86.71**	**87.25**	**86.66**	**87.11**	**88.41**	**87.75**

Appendix C: Detailed ML Results for the Full NSU Classification Task

	Weka's J4.8			SLIPPER		
NSU Class	Recall	Precision	F1	Recall	Precision	F1
Ack	95.00	96.80	95.90	96.67	95.71	96.19
CheckQu	100.00	83.30	90.90	86.67	100.00	92.86
CE	92.40	95.30	93.80	96.33	93.75	95.02
Sluice	100.00	95.20	97.60	94.44	100.00	97.14
ShortAns	83.00	80.70	81.90	85.25	84.46	84.85
AffAns	86.00	82.70	84.30	82.79	87.38	85.03
Reject	100.00	76.20	86.50	77.60	100.00	87.39
RepAffAns	68.00	65.40	66.70	67.71	72.71	70.12
RepAck	86.30	84.10	85.20	84.04	92.19	87.93
HelpReject	33.30	46.20	38.70	29.63	18.52	22.79
PropMod	60.00	100.00	75.00	100.00	100.00	100.00
FactMod	91.30	100.00	95.50	100.00	100.00	100.00
BareModPh	70.00	100.00	82.40	83.33	69.44	75.76
ConjFrag	100.00	71.40	83.30	100.00	100.00	100.00
Filler	37.50	50.00	42.90	70.00	56.33	62.43
weighted score	**89.67**	**89.78**	**89.53**	**91.57**	**92.70**	**92.01**

	TiMBL			MaxEnt		
NSU Class	Recall	Precision	F1	Recall	Precision	F1
Ack	95.71	95.58	95.64	95.54	94.59	95.06
CheckQu	77.78	71.85	74.70	63.89	85.19	73.02
CE	93.32	94.08	93.70	88.89	94.44	91.58
Sluice	100.00	94.44	97.14	88.89	94.44	91.58
ShortAns	87.79	88.83	88.31	88.46	84.91	86.65
AffAns	85.00	85.12	85.06	86.83	81.94	84.31
Reject	98.33	80.28	88.39	100.00	78.21	87.77
RepAffAns	58.70	55.93	57.28	69.26	62.28	65.58
RepAck	86.11	80.34	83.12	86.95	77.90	82.18
HelpReject	22.67	40.00	28.94	00.00	00.00	00.00
PropMod	100.00	100.00	100.00	44.44	100.00	61.54
FactMod	97.50	100.00	98.73	93.33	100.00	96.55
BareModPh	69.44	83.33	75.76	58.33	100.00	73.68
ConjFrag	100.00	100.00	100.00	00.00	00.00	00.00
Filler	44.33	55.00	49.09	62.59	100.00	76.99
weighted score	**91.49**	**90.75**	**91.02**	**91.96**	**93.17**	**91.21**

Acknowledgments

This work was funded by grant RES-000-23-0065 from the Economic and Social Council of the United Kingdom and it was undertaken while all three authors were members of the Department of Computer Science at King's College London. We wish to thank Lief Arda Nielsen and Matthew Purver for useful discussion and suggestions regarding machine learning algorithms. We are grateful to two anonymous reviewers for very helpful comments on an earlier draft of this article. Their insights and suggestions have resulted in numerous improvements. Of course we remain solely responsible for the ideas presented here, and for any errors that may remain.

References

Beyssade, Claire and Jean-Marie Marandin. 2005. Contour meaning and dialogue structure. Paper presented at the workshop Dialogue Modelling and Grammar, Paris, France.

Burnard, Lou. 2000. *Reference Guide for the British National Corpus (World Edition).* Oxford University Computing Services. Available from `ftp://sable.ox.ac.uk/pub/ota/BNC/`.

Clark, Herbert H. 1996. *Using Language.* Cambridge University Press, Cambridge.

Cohen, William and Yoram Singer. 1999. A simple, fast, and effective rule learner. In *Proceedings of the 16th National Conference on Artificial Intelligence,* pages 335–342, Orlando, FL.

Daelemans, Walter, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2003. TiMBL: Tilburg Memory-Based Learner, v. 5.0, Reference Guide. Technical Report ILK-0310, University of Tilburg.

Di Eugenio, Barbara and Michael Glass. 2004. The kappa statistic: A second look. *Computational Linguistics,* 30(1):95–101.

Fernández, Raquel. 2006. *Non-Sentential Utterances in Dialogue: Classification, Resolution and Use.* Ph.D. thesis, Department of Computer Science, King's College London, University of London.

Fernández, Raquel and Jonathan Ginzburg. 2002. Non-sentential utterances: A corpus study. *Traitement automatique des languages,* 43(2):13–42.

Fernández, Raquel, Jonathan Ginzburg, Howard Gregory, and Shalom Lappin. In press. SHARDS: Fragment resolution in dialogue. In H. Bunt and R. Muskens, editors, *Computing Meaning,* volume 3. Kluwer, Amsterdam.

Fernández, Raquel, Jonathan Ginzburg, and Shalom Lappin. 2004. Classifying Ellipsis in Dialogue: A Machine Learning Approach. In *Proceedings of the 20th International Conference on Computational Linguistics,* pages 240–246, Geneva, Switzerland.

Garside, Roger. 1987. The CLAWS word-tagging system. In R. Garside, G. Leech, and G. Sampson, editors, *The Computational Analysis of English: A Corpus-Based Approach.* Longman, Harlow, pages 30–41.

Ginzburg, Jonathan. 1996. Interrogatives: Questions, facts, and dialogue. In Shalom Lappin, editor, *Handbook of Contemporary Semantic Theory.* Blackwell, Oxford, pages 385–422.

Ginzburg, Jonathan. 1999. Ellipsis resolution with syntactic presuppositions. In H. Bunt and R. Muskens, editors, *Computing Meaning: Current Issues in Computational Semantics.* Kluwer, Amsterdam, pages 255–279.

Ginzburg, Jonathan. 2005. Abstraction and ontology: Questions as propositional abstracts in type theory with records. *Journal of Logic and Computation,* 2(15):113–118.

Ginzburg, Jonathan. forthcoming. *Semantics and Interaction in Dialogue.* CSLI Publications and University of Chicago Press, Stanford, California. Draft chapters available from `http://www.dcs.kcl.ac.uk/staff/ginzburg`.

Ginzburg, Jonathan and Robin Cooper. 2004. Clarification, ellipsis, and the nature of contextual updates. *Linguistics and Philosophy,* 27(3):297–366.

Ginzburg, Jonathan, Howard Gregory, and Shalom Lappin. 2001. SHARDS: Fragment resolution in dialogue. In H. Bunt, I. van der Suis, and E. Thijsse, editors, *Proceedings of the Fourth International Workshop on Computational Semantics,* pages 156–172, Tilburg, The Netherlands.

Ginzburg, Jonathan and Ivan Sag. 2001. *Interrogative Investigations.* CSLI Publications, Stanford, California.

Larsson, Staffan. 2002. *Issue-based Dialogue Management.* Ph.D. thesis, Göteborg University, Sweden.

Larsson, Staffan, Peter Ljunglöf, Robin Cooper, Elisabet Engdahl, and Stina Ericsson. 2000. GoDiS: An accommodating dialogue system. In *Proceedings of*

ANLP/NAACL-2000 Workshop on Conversational Systems, pages 7–10, Seattle, WA.

Le, Zhang. 2003. Maximum entropy modeling toolkit for Python and C++. Available from `http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html`.

Leech, Geoffrey, Roger Garside, and Michael Bryant. 1994. The large-scale grammatical tagging of text: Experience with the British National Corpus. In N. Oostdijk and P. de Haan, editors, *Corpus-Based Research into Language.* Rodopi, Amsterdam, pages 47–63.

Malouf, Robert. 2002. A comparision of algorithm for maximum entropy parameter estimation. In *Proceedings of the Sixth Conference on Natural Language Learning,* pages 49–55, Taipei, Taiwan.

Purver, Matthew. 2001. SCoRE: A tool for searching the BNC. Technical Report TR-01-07, Department of Computer Science, King's College London.

Purver, Matthew. 2004a. CLARIE: The Clarification Engine. In J. Ginzburg and E. Vallduví, editors, *Proceedings of the 8th Workshop on the Semantics and Pragmatics of Dialogue (Catalog),* pages 77–84, Barcelona, Spain.

Purver, Matthew. 2004b. *The Theory and Use of Clarification Requests in Dialogue.* Ph.D. thesis, King's College, University of London.

Schlangen, David. 2003. *A Coherence-Based Approach to the Interpretation of Non-Sentential Utterances in Dialogue.* Ph.D. thesis, University of Edinburgh, Scotland.

Schlangen, David. 2005. Towards finding and fixing fragments: Using ML to identify non-sentential utterances and their antecedents in multi-party dialogue. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics,* pages 247–254, Ann Arbor, MI.

Schlangen, David and Alex Lascarides. 2003. The interpretation of non-sentential utterances in dialogue. In *Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue,* pages 62–71, Sapporo, Japan.

Traum, David. 1994. *A Computational Theory of Grounding in Natural Language Conversation.* Ph.D. thesis, University of Rochester, Department of Computer Science, Rochester, NY.

Witten, Ian H. and Eibe Frank. 2000. *Data Mining: Practical Machine Learning Tools with Java Implementations.* Morgan Kaufmann, San Francisco. Available from `http://www.cs.waikato.ac.nz/ml/weka`.