Automatic Bare Sluice Disambiguation in Dialogue*

Raquel Fernández Jonathan Ginzburg Shalom Lappin

Department of Computer Science
King's College London
{raquel,ginzburg,lappin}@dcs.kcl.ac.uk

Abstract

The capacity to recognise and interpret sluices—bare *wh*-phrases that exhibit a sentential meaning—is essential to maintaining cohesive interaction between human users and a machine interlocutor in a dialogue system. In this paper we present a machine learning approach to sluice disambiguation in dialogue. Our experiments, based on solid theoretical considerations, show that applying machine learning techniques using a compact set of features that can be automatically identified from PoS markings in a corpus can be an efficient tool to disambiguate between sluice interpretations.

1 Introduction

Most theoretical analyses of *sluicing*—bare *wh*-phrases that exhibit a sentential meaning—focus on embedded sluices like e.g (1a), considered out of dialogue context (see e.g. Ross 1969; Chung et al. 1995). They rarely look at *direct* sluices—sluices used in queries to request further elucidation of quantified parameters (e.g. (1b)). With a few isolated exceptions, these analyses also ignore a class of uses we refer to (following Ginzburg and Sag 2001 (G&S)) as *reprise* sluices. These are used to request clarification of the reference of a constituent in a partially understood utterance, as in (1c).

(1) a. Jo left someone/something/somewhere.
Bo knows who/what/where/why.

b. Cassie: I know someone who's a good kisser.
Catherine: Who? [KP4, 512][1]

c. Sue: I think you were getting a real panic then.
Angela: When? [KB6, 1888]

*We wish to thank Lief Arda Nielsen and Mattew Purver for useful discussion and suggestions. The work presented in this paper has been funded by grant RES-000-23-0065 from the Economic and Social Council of the United Kingdom.

[1]This notation indicates the BNC file (KP4) and the sluice sentence number (512).

Our corpus investigation shows that the combined set of direct and reprise sluices constitutes more than 75% of all sluices in the British National Corpus (BNC). In fact, they make up approximately 33% of all *wh*-interrogative queries in the BNC. Therefore, they represent an empirically important construction which has been understudied from both theoretical and computational perspectives.

The capacity to recognise and interpret bare sluices correctly is essential to maintaining cohesive interaction between human users and a machine interlocutor in an automated dialogue system. If a dialogue system does not assign the correct interpretation to a sluice, it will not respond correctly to the question. Consider the following example.

(2) A: A LAN support person contacted you recently.
B: Who?

If A interprets *who* in (2) as a query for more specific information on the indefinite description "A LAN support person", then the appropriate response to B's question is to provide a proper name or identifying description. However, if A takes B to be indicating that he/she does not understand the description in this context, then A should either repeat the NP, or supply a paraphrase designed to make its meaning clear to B, such as "a technician to help with your network problem".

The next section sketches the theory underlying our study. Section 3 describes our corpus investigation of classifying sluices into interpretation classes. In section 4 we use a set of features to manually annotate a data set of sluices, and run two machine learning algorithms: SLIPPER and TiMBL, which yield similar success rates of approx. 89%. Section 5 presents a procedure for automatically annotating our data set with the features that we use with our two machine learning algorithms. When we apply the algorithms to its output both achieve a success rate of 79%. In Section 6 we compare our machine learning results with those of a simple algorithm that determines the sluice reading solely on the basis of the raw frequency counts of different interpretations for that sluice type. We finally present our conclusions and future work in Section 7.

2 Sluicing: Theory and Implementation

In this section we briefly introduce the theoretical framework that underlies our investigation, and provide a schematic description of SHARDS, a system which implements the basic ideas of this theory to resolve the interpretation of clausal fragments in dialogue.

2.1 Sluicing: A Constructional Approach

G&S provide a detailed analysis of a number of classes of non-sentential utterances (NSUs), including short answers, sluicing, and Clarification Ellipsis

(CE). The framework they develop combines the basic approach to grammatical constructions developed by Sag (1997) within Head-driven Phrase Structure Grammar (HPSG) and a theory of context in dialogue, the KOS framework (Ginzburg 1996; Ginzburg frth; Larsson 2002). The essential idea they develop is that NSUs get their main predicates from context, specifically via unification with the question that is currently *under discussion*, an entity dubbed the *maximal question under discussion* (MAX-QUD).[2] NSU resolution is, consequently, tied to conversational topic, viz. the MAX-QUD. The resolution of NSUs, on this approach, involves one other parameter, an antecedent sub-utterance dubbed the *salient-utterance* (SAL-UTT). This plays a role similar to the role played by the *parallel element* in higher order unification–based approaches to ellipsis resolution (see e.g. Dalrymple et al. 1991; Pulman 1997). Intuitively, the SAL-UTT provides a partial specification of the focal (sub)utterance—it is computed as the (sub)utterance associated with the role bearing widest scope within MAX-QUD. SAL-UTT is used to encode syntactic and phonological parallelism between the fragment and an antecedent which non-sentential utterances often exhibit. For instance, case matching or even identity at the level of phonological segments. As we will see in subsequent sections, many of the heuristics we develop for disambiguating sluices relate to identifying the SAL-UTT.

What determines the MAX-QUD? In the most prototypical case, the MAX-QUD is the content of the most recent utterance, as in short answers; the SAL-UTT in such a case is the sub-utterance of the *wh*-phrase:

(3) a. A: Who phoned?
B: Bo (= Bo phoned).
b. MAX-QUD: $\lambda x.Phone(x,t)$

For propositional lexemes such as 'yes', 'no', and 'probably' the MAX-QUD is a polar question $p?$, which is either the (content of the) most recent utterance or the most recent utterance is an assertion p:[3]

(4) a. A: Did Bo phone?
B: Yes/No/Probably (= Bo phoned/didn't phoned /probably phoned).
b. A: Bo phoned.
B: Yes/No/Probably (= Bo phoned/didn't phoned/ probably phoned).
c. MAX-QUD: $?Phone(b,t)$

[2]In G&S's framework, questions are represented as semantic objects comprising a set of parameters—that is, restricted indices—and a proposition PROP as in (i). This is the feature structure counterpart of the λ-abstract $\lambda\pi(\ldots\pi\ldots)$. In a *wh*-question the PARAMS set represents the abstracted INDEX values associated with the *wh*-phrase(s). For a polar question the PARAMS set is empty.

(i)
$$\begin{bmatrix} \textit{question} & \\ \text{PARAMS} & \{\pi,\ldots\} \\ \text{PROP} & \begin{bmatrix} \textit{proposition} & \\ \text{SIT} & \\ \text{SOA} & \left[\textit{soa}(\ldots\pi\ldots)\right] \end{bmatrix} \end{bmatrix}$$

[3]There is no SAL-UTT in such cases.

A similar characterisation applies to direct sluicing, with the additional requirement that p be a quantified proposition; the SAL-UTT in such a case is the sub-utterance associated with the widest scoping quantifier:[4]

(5) a. A: A student phoned. B: Who? (= Which student phoned?)
b. A: Did someone phone? B: Who? (= Who phoned?).
c. MAX-QUD: $?\exists x.Phone(x,t)$

MAX- QUD can also arise in a somewhat less 'direct' way, via a process of *utterance coercion* (see Ginzburg and Cooper 2001, 2004), triggered by the inability to *ground* (Clark 1996; Traum 1994) the previous utterance. The output of the coercion process is a question about the content of a sub-utterance which the addressee cannot resolve. This sub-utterance constitutes the SAL-UTT. Such a question is the MAX-QUD for reprise sluices and CE. For instance, if the original utterance is (6a), with 'Bo' as the unresolvable sub-utterance, one possible output from the coercion operations defined by Ginzburg and Cooper (2004) is the question in (6b):

(6) a. A: Did Bo leave? B: Who? (= Who are you asking if s/he left?)
b. MAX-QUD: $\lambda bAsk(A, ?leave(b,t0))$

2.2 Implementing the direct/reprise divide

SHARDS (Ginzburg et al. 2001; Fernández et al. frth) is an implemented system which provides a procedure for computing the interpretation of NSUs in dialogue. The system comprises two main components: an HPSG-based grammar and a resolution procedure. The grammar employed is an implemented version of the wide coverage grammar proposed by G&S and it is encoded in ProFIT (Erbach 1995). Once an elliptical sentence has been parsed, the resolution component of the system resolves its interpretation by assigning values to the MAX-QUD and SAL-UTT features of the clause on the basis of information located in a structured record of previously processed sentences stored in memory. 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 2004a, 2004b). In particular, CLARIE can parse and generate reprise sluices by implementing the aforementioned analysis of grounding/clarification interaction.

[4] Adjuncts sluices are possible even without an overt antecedent:

(i) A: John saw Mary.
B: Why?/With who? (= Why/With who did John see Mary?)

In such cases the value of SAL-UTT needs to be null—there is no antecedent quantificational NP and none of the parallelism effects that argument sluices show. (Fernández et al. frth) suggest that such cases should be analysed in a way akin to the propositional lexemes above, rather than like argument-sluices such as *who* or *what*.

3 Corpus Study

Our corpus-based investigation of bare sluices has been performed using the dialogue transcripts of the BNC. The corpus of bare sluices has been constructed using SCoRE (Purver 2001), a tool that allows one to search the BNC using regular expressions.

The dialogue transcripts of the BNC contain a total of 5343 sluices, whose distribution is shown in Table 1. From this total, we selected two different samples of sluices, created by arbitrarily selecting 50 sluices of each class (15 in the case of *which*). The first sample included all instances of bare *how* and bare *which* found, making up a total of 365 sluices. The second sample contained 50 instances of the remaining classes, making up a total of 300 sluices.

what	*why*	*who*	*where*	*which N*	*when*	*how*	*which*	**Total**
3045	1125	491	350	160	107	50	15	**5343**

Table 1: Total of sluices in the BNC

The annotation procedure consisted of classifying the two samples of sluices according to a set of domain independent categories. The categories used, drawn from the theoretical distinctions referred to in the previous section, correspond to different sluice interpretations. The classification was done independently by 3 different annotators.

To classify the sluices in the first sample of our sub-corpus, we used the following categories:

Direct The utterer of the sluice understands the antecedent of the sluice without difficulty. The sluice queries for additional information that was explicitly or implicitly quantified away in the previous utterance.

(7) Caroline: I'm leaving this school.
Lyne: When? [KP3, 538]

Reprise The utterer of the sluice cannot understand a particular aspect of the previous utterance, corresponding to one of its constituents, which the previous (or possibly not directly previous) speaker assumed as presupposed (typically a contextual parameter, except for *why*, where the relevant "parameter" is something like speaker intention or speaker justification).

(8) Geoffrey: What a useless fairy he was.
Susan: Who? [KCT, 1753]

Clarification The sluice is used to ask for clarification of the entire preceding utterance.

(9) June: Only wanted a couple weeks.
Ada: What? [KB1, 3312]

Unclear It is difficult to understand what content the sluice conveys, possibly because the input is too poor to make a decision as to its resolution, as in the following example:

(10) Unknown : <unclear> <pause>
Josephine: Why? [KCN, 5007]

After annotating the first sample, we decided to add a new category to the above set. The sluices in the second sample were classified according to a set of five categories, including the following:

Wh-anaphor The antecedent of the sluice is a *wh*-phrase.

(11) Larna: We're gonna find poison apple and I know where that one is.
Charlotte: Where? [KD1, 2371]

The reliability of the annotation was evaluated using the *kappa* coefficient (K) (Carletta 1996). The agreement on the coding of the first sample of sluices was moderate ($K = 52$). Agreement on the annotation of the 2nd sample was considerably higher although still not entirely convincing ($K = 61$). 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 "less expert" coder out of the coder pool increases agreement very significantly: $K = 70$ in the first sample, and $K = 71$ in the second one.

The distribution of interpretations for each class of sluice is shown in Table 2. The distributions are presented as percentages of pairwise agreement (i.e. agreement between pairs of coders), leaving aside the `unclear` cases. The results of the study show that the distribution of readings is significantly different for each class of sluice.

	First Sample			**Second Sample**			
	`dir`	`rep`	`cla`	`dir`	`rep`	`cla`	`wh-a`
what	9	22	69	7	23	66	4
why	57	43	0	83	14	0	3
who	24	76	0	0	95	0	5
where	25	75	0	22	69	0	9
when	67	33	0	65	29	0	6
which N	12	88	0	20	80	0	0
which	4	96	0	—	—	—	—
how	87	8	5	—	—	—	—

Table 2: Distributions as pairwise agreement percentages

4 Applying Machine Learning

In (Fernández et al. 2004) we used the results of the corpus study described in the previous section to identify a number of heuristic principles for assigning an interpretation to bare sluice types. We formulated these principles as probability weighted Horn clauses to achieve the most general and declarative expression of these conditions. We then used the predicates in the antecedents of the Horn clauses as features to manually annotate an input data set of sluices. To evaluate the predictive power of these features in the sluice disambiguation

task, we applied two machine learning algorithms to our data set: SLIPPER, a rule-based learning algorithm, and TiMBL, a memory-based system.

4.1 Experimental Setup

The input data set was generated from all three-way agreement instances plus those instances where there is agreement between the two more experienced coders, leaving out cases classified as `unclear`. We reclassified 9 instances in the first sample as `wh-anaphor`, and also included these data.[5] The total data set includes 351 datapoints. These were manually annotated according to the set of features shown in Table 3.

Features	Informal Description	Values
sluice	type of sluice	`what, why,...`
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 ant. expression	`yes, no, ?`

Table 3: Features

We use a total of 10 features. All features are nominal. Except for the `sluice` feature that indicates the sluice type, they are all boolean, i.e. they can take as value either `yes` or `no` (`decl`, `n_decl` for mood, and `pos`, `neg` for polarity). The features `mood`, `polarity` and `frag` 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 item or construction contained in the utterance. They will take `yes` as a value if this element or construction exists *and*, it matches the semantic restrictions imposed by the sluice type. Unknown or irrelevant values are indicated by a question mark. This allows us to express, for instance, that the presence of a proper name is irrelevant to determining the interpretation of a *where* sluice, while 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 (question mark value) when there is an antecedent expression defined by another feature.

4.2 SLIPPER and TiMBL

In our first experiment we use a rule-based learning algorithm called SLIPPER (for Simple Learner with Iterative Pruning to Produce Error Reduction) (Cohen and Singer 1999). We performed a 10-fold cross-validation on the total data

[5] We reclassified those instances in the first sample that had motivated the introduction of the `wh-anaphor` category for the second sample. Given that there were no disagreements involving this category, this reclassification was straightforward.

set, obtaining an average success rate of 85%. For the holdout method, we held over 100 instances as a testing data, and used the remainder (251 datapoints) for training. This yielded a success rate of 89%.

For the second experiment we used TiMBL, a memory-based learning algorithm developed at Tilburg University (Daelemans et al. 2003). The results obtained are similar to those for SLIPPER. In 10-fold cross-validation TiMBL achieves an average success rate of 92%. Using the holdout method on the same training and testing data sets as SLIPPER, yields a success rate of 88%. Table 4 shows a slightly revised version of the results reported in (Fernández et al. 2004).

	SLIPPER			TiMBL		
Category	Recall	Prec.	F1	Recall	Prec.	F1
`direct`	96.67	82.86	89.23	86.66	83.87	85.24
`reprise`	87.04	92.16	89.52	87.03	92.15	89.51
`clarification`	83.33	83.33	83.33	83.33	71.42	76.91
`wh_anaphor`	80.00	100.00	88.89	100.00	100.00	100.00

Table 4: Accuracy Results

5 Automatic Feature Annotation

The results presented in (Fernández et al. 2004) were obtained using a manually annotated data set. In this section we describe a procedure for automatically assigning values to the features discussed above, and we report the results obtained from applying SLIPPER and TiMBL to the automatically annotated data set. In the next section, we compare all of these results with those of a simple frequency-based heuristic that assigns to each sluice type the category with the highest probability for that type.

5.1 The Automatic Annotation Procedure

In order to automate the overall task of assigning a sluice interpretation category to bare *wh*-phrases in dialogue, we designed and implemented a procedure to automatically annotate our data set with the features shown in Table 3.

The procedure employs string searching and pattern matching techniques that exploit the SGML markup of the BNC. It relies crucially on the PoS information encoded in the corpus annotation. The BNC is annotated with a set of 57 PoS codes, known as the C5 tagset, plus 4 codes for punctuation tags. A list of these codes can be found in Burnard (2000). The $\sim$ 100 million words of the BNC were automatically tagged using the CLAWS system developed at Lancaster University (Garside 1987). The BNC PoS annotation process is described in detail in Leech et al. (1994).

Our annotation algorithm consists of three basic steps: First, it finds the antecedent utterance of each sluice in our sub-corpus. In the current version, we take the antecedent utterance to be the last sentence in the previous turn.[6]

[6]For an explanation of how sentences and turns are defined in the BNC see the BNC web site (`http://www.hcu.ox.ac.uk/BNC/`).

Feature	Value	Recall	Precision	F1
mood	`decl`	88.69	91.35	90.00
	`n_decl`	72.60	62.35	67.09
polarity	`pos`	83.27	91.45	87.17
	`neg`	82.35	43.75	57.14
	`?`	68.83	62.35	65.43
quant	`yes`	60.00	31.58	41.38
	`no`	55.51	82.93	66.50
	`?`	66.28	38.26	48.51
deictic	`yes`	74.29	86.67	80.00
	`no`	97.87	93.88	95.83
	`?`	97.66	100.00	98.81
proper_n	`yes`	82.61	90.48	86.36
	`no`	95.56	91.49	93.48
	`?`	100.00	100.00	100.00
pro	`yes`	85.71	77.78	81.55
	`no`	66.67	87.50	75.68
	`?`	96.92	95.09	96.00
def_desc	`yes`	78.46	69.86	73.91
	`no`	82.08	79.82	80.93
	`?`	93.89	100.00	96.85
wh	`yes`	66.67	100.00	80.00
	`no`	100.00	98.23	99.11
overt	`yes`	45.00	60.00	51.43
	`no`	85.25	73.24	78.79
	`?`	84.21	90.72	87.34

Table 5: Results - Automatic Feature Annotation

Second, the features are given values independently of the sluice type, using PoS information. For instance, the presence of a tag `<NP0>` triggers a `yes` value for the feature **proper_n**, while the tag `<XX0>`, which is assigned to the negative particle 'not' or 'n't', (partially) determines the value of the feature **polarity**.

Finally, the feature values are filtered according to the sluice type. A feature like **pro**, for example, is not relevant to disambiguate a *why* sluice. *Why* sluices will therefore assign a question mark `?` value for this feature. The **wh** feature, on the other hand, will get a `yes` value if there is a *wh*-word in the antecedent utterance *and* this matches the sluice type to be disambiguated.

Our algorithm achieves 86% success rate with respect to the manual feature annotation. Table 5 shows the recall, precision and f-measure percentages obtained for the value of each feature. It can be seen that the features with lower scores are those that don't have a trivial PoS correlate. This is the case for `quant` (which would inlcude any potentially quantified expression, from indefinite pronouns, to definite descriptions and temporal and locative expressions) and `overt`, as well as `mood` and `polarity`. Features like `deicitic`, `proper_n` and `wh`, on the other hand, obtain highly accurate results.

5.2 SLIPPER and TiMBL again

Finally, we ran our machine learning algorithms again, but this time we used as input the automatically annotated data set. With the holdout method, both SLIPPER and TiMBL yielded a success rate of 79%, achieving very similar accuracy results for each category. In a 10-fold cross-validation TiMBL remains constant (79% average success rate), while SLIPPER drops 4% points (75%). Recall, precision, and f-measure percentages are given in Table 6.

	SLIPPER			TiMBL		
Category	Recall	Prec.	F1	Recall	Prec.	F1
`direct`	83.33	71.43	76.92	83.33	71.42	76.91
`reprise`	77.78	85.71	81.55	77.77	85.71	81.54
`clarification`	100.00	60.00	75.00	100.00	60.00	75.00
`wh_anaphor`	60.00	100.00	75.00	60.00	100.00	75.00

Table 6: Results - Automatically Annotated Data Set

6 A Simple Frequency-Based Heuristic

In section 3 we showed that distinct sluice types correspond to radically different distributions of sluice interpretations. In this section we consider a disambiguation heuristic that simply assigns to each sluice type the most probable reading, where the probability value of a reading is determined directly by its relative frequency. Following the distribution patterns reported in Table 2, our frequency-based heuristic specifies the following reading for each sluice type: *what*-`clarification`, *why*-`direct`, *who*-`reprise`, *where*-`reprise`, *when*-`direct`, *which*-`reprise`, *whichN*-`reprise`.

This heuristic performs surprisingly well, achieving a success rate of 74%. The accuracy results for each category are shown in Table 7. Using our machine learning algorithms on a data set manually annotated with the appropriate features we are able to improve this result 18% points using TiMBL and 11% points with SLIPPER. Using the automatic feature annotation as input, TiMBL achieves 5% points over the frequency-based result, and SLIPPER 1%. In the case of `clarification`, however, one gets better results assigning a `clarification` reading to all *what* sluices than using automatically annotated features to disambiguate between readings. This indicates that we should aim for a better identification of the context on which a `clarification` reading depends.

Category	Recall	Precision	F1
`direct`	72.64	67.54	70.00
`reprise`	79.31	80.50	79.90
`clarification`	100.00	64.86	78.69
`wh_anaphor`	0.00	0.00	0.00

Table 7: Simple Frequency-Based Heuristic

The fact that our frequency-based heuristic achieves a relatively high degree of success indicates that the raw unconditional probability values for the

possible sluice readings of sluice types are a good rough guide for predicting sluice interpretation. We improve these results significantly by conditioning classification on a set of features. This effectively gives us conditional probabilities for sluice readings. The performance of these conditional probabilities depends upon the accuracy of the feature annotation. When feature accuracy declines under automatic annotation, the conditional probabilities converge on the raw probabilities These comparative results suggest that future work should seek to improve the accuracy of our feature annotation procedure through error analysis.

7 Conclusions and Future Work

We have formulated the problem of identifying the correct reading of a sluice in dialogue as the task of devising a reliable procedure for classifying it according to a set of possible interpretations.

We have presented a small set of features for annotating a dialogue corpus from which we can extract reasonably reliable methods for assigning interpretational classifications to *wh*-fragment questions through machine learning techniques. They can be identified by an automatic procedure that relies solely on the PoS marking of the corpus. When we compared the results of our two machine learning systems to a simple probabilistic classification algorithm that uses the frequency counts associated with each sluice-type, we found that, although the algorithm performs surprisingly well, machine learning applied to our features yields a substantial improvement in success of classification.

Besides refining our automatic feature annotation method to achieve a more accurate basis for *wh*-fragment classification, we are currently extending the approach presented here to other kinds of non-sentential utterances in dialogue. The results obtained so far in the classification of a wider range of fragments, although still preliminary, are certainly encouraging. In the next phase of our research we aim at integrating the fragment classification procedures into our dialogue interpretation system.

References

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

Carletta, J. (1996). Assessing agreement on classification tasks: the kappa statistics. *Computational Linguistics 2*(22), 249–255.

Chung, S., W. Ladusaw, and J. McCloskey (1995). Sluicing and logical form. *Natural Language Semantics 3*, 239–282.

Clark, H. H. (1996). *Using Language*. Cambridge University Press.

Cohen, W. and Y. Singer (1999). A simple, fast, and effective rule learner. In *Proc. of the 16th National Conference on AI*.

Daelemans, W., J. Zavrel, K. van der Sloot, and A. van den Bosch. (2003). TiMBL: Tilburg Memory Based Learner, Reference Guide. Technical Report ILK-0310, U. of Tilburg.

Dalrymple, M., F. Pereira, and S. Shieber (1991). Ellipsis and Higher Order Unification. *Linguistics and Philosophy 14*, 399–452.

Erbach, G. (1995). ProFIT: Prolog with features, inheritance and templates. In *Proc. of the 7th European Conference of the ACL*.

Fernández, R., J. Ginzburg, H. Gregory, and S. Lappin (frth.). SHARDS: Fragment resolution in dialogue. In H. Bunt and R. Muskens (Eds.), *Computing Meaning*, Volume 3. Kluwer.

Fernández, R., J. Ginzburg, and S. Lappin (2004). Classifying Ellipsis in Dialogue: A Machine Learning Approach. In *Proc. of the 20th International Conference on Computational Linguistics*, pp. 240–246.

Garside, R. (1987). The claws word-tagging system. In R. Garside, G. Leech, and G. Sampson (Eds.), *The computational analysis of English: a corpus-based approach*, pp. 30–41. Longman.

Ginzburg, J. (1996). Interrogatives: Questions, facts, and dialogue. In S. Lappin (Ed.), *Handbook of Contemporary Semantic Theory*. Blackwell.

Ginzburg, J. (frth.). *Semantics and Interaction in Dialogue*. CSLI Publications and University of Chicago Press. Draft chapters available from http://www.dcs.kcl.ac.uk/staff/ginzburg.

Ginzburg, J. and R. Cooper (2001). Resolving ellipsis in clarification. In *Proc. of the 39th Meeting of the ACL*.

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

Ginzburg, J., H. Gregory, and S. Lappin (2001). SHARDS: Fragment resolution in dialogue. In *Proc. of the 4th International Workshop on Computational Semantics*.

Ginzburg, J. and I. Sag (2001). *Interrogative Investigations*. CSLI.

Larsson, S. (2002). *Issue based Dialogue Management*. Ph. D. thesis, Gothenburg University.

Leech, G., R. Garside, and M. Bryant (1994). The large-scale grammatical tagging of text: experience with the British National Corpus. In N. Oostdijk and P. de Haan (Eds.), *Corpus-based research into language*, pp. 47–63.

Pulman, S. (1997). Focus and Higher Order Unification. *Linguistics and Philosophy 20*.

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

Purver, M. (2004a). CLARIE: the Clarification Engine. In *Proc. of the 8th Workshop on the Semantics and Pragmatics of Dialogue*, pp. 77–84.

Purver, M. (2004b). *The Theory and Use of Clarification in Dialogue*. Ph. D. thesis, King's College, London.

Ross, J. (1969). Guess who. In *Proc. of the 5th annual Meeting of the Chicago Linguistics Society*, pp. 252–286. CLS.

Sag, I. A. (1997). English relative clause constructions. *Journal of Linguistics 33*, 431–484.

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