Using Machine Learning for Non-Sentential Utterance Classification

Raquel Fernández, Jonathan Ginzburg and **Shalom Lappin**
Department of Computer Science
King's College London
UK
{raquel,ginzburg,lappin}@dcs.kcl.ac.uk

Abstract

In this paper we investigate the use of machine learning techniques to classify a wide range of non-sentential utterance types in dialogue, a necessary first step in the interpretation of such fragments. We train different learners on a set of contextual features that can be extracted from PoS information. Our results achieve an 87% weighted f-score—a 25% improvement over a simple rule-based algorithm baseline.

Keywords Non-sentential utterances, machine learning, corpus analysis

1 Introduction

Non-Sentential Utterances (NSUs)—fragmentary utterances that convey a full sentential meaning—are a common phenomenon in spoken dialogue. Because of their elliptical form and their highly context-dependent meaning, NSUs are a challenging problem for both linguistic theories and implemented dialogue systems. Although perhaps the most prototypical NSU type are short answers like (1), recent corpus studies (Fernández and Ginzburg, 2002; Schlangen, 2003) have shown that other less studied types of fragments—each with its own resolution constraints—are also pervasive in real conversations.

(1) Kevin: Which sector is the lawyer in?
Unknown: Tertiary. [KSN, 1776–1777][1]

[1]This notation indicates the British National Corpus file, KSN, and the sentence numbers, 1776–1777.

Arguably the most important issue in the processing of NSUs concerns their resolution, i.e. the recovery of a full clausal meaning from a form which is incomplete. However, given their elliptical form, NSUs are very often ambiguous. Hence, a necessary first step towards this final goal is the identification of the right NSU type, which will determine the appropriate resolution procedure.

In the work described in this paper we address this latter issue, namely the classification of NSUs, using a machine learning approach.[2] The techniques we use are similar to those applied by (Fernández et al., 2004) to disambiguate between the different interpretations of bare *wh*-phrases. Our investigation, however, takes into account a much broader range of NSU types, providing a wide coverage NSU classification system.

We identify a small set of features, easily extractable from PoS information, that capture the contextual properties that are relevant for NSU classification. We then use several machine learners trained on these features to predict the most likely NSU class, achieving an 87% weighted f-score. We evaluate our results against a baseline system that uses an algorithm with four rules.

The paper is structured as follows. First we introduce the taxonomy of NSU classes we adopt. In section 3 we explain how the empirical data has been collected and which restrictions have been adopted in selecting the data set to be used in our experiments. The features we use to characterise such data, and the generation process of the data set are presented in section 4. Next we introduce some very

[2]A related task, namely that of automatically identifying NSUs and their antecedents, is investigated by Schlangen (2005).

simple algorithms used to derive a baseline for our NSU classification task, and after that present the machine learners used in our experiments. In section 7 we report the results obtained, evaluate them against the baseline systems, and discuss the results of a second experiment performed on a data set created by dropping one of the restrictions adopted before. Finally, in Section 8, we offer conclusions and some pointers for future work.

2 NSU Taxonomy

We propose a taxonomy of 14 NSU classes. With a few modifications, these classes follow the corpus-based taxonomy proposed in (Fernández and Ginzburg, 2002). In what follows we exemplify each of the categories we use in our work and characterise them informally.

2.1 Question-denoting NSUs

Sluices and Clarification Ellipsis (CE) are the two classes of NSUs that denote questions.

Sluice We consider as sluices all *wh*-question NSUs,[3] like the following:

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

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

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

(4) A: There's only two people in the class
B: Two people? [KPP, 352–354]

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

2.2 Proposition-denoting NSUs

The remaining NSU classes denote propositions.

[3] In (Fernández and Ginzburg, 2002)'s taxonomy, this category is used for non-reprise bare *wh*-phrases, while reprise sluices are classified as CE. We opt for a more form-based category that can convey different readings, without making distinctions between these readings. Recent work by (Fernández et al., 2004) has shown that sluice interpretations can be efficiently disambiguated using machine learning techniques.

Short Answer Short Answers are typical responses to (possibly embedded) *wh*-questions.

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

(7) A: Can you tell me where you got that information from?
B: From our wages and salary department. [K6Y, 94–95]

However, there is no explicit *wh*-question in the context of a short answer to a CE question (8), nor in cases where the *wh*-phrase is ellided (9).

(8) A: Vague and?
B: Vague ideas and people. [JJH,65–66]

(9) A: What's plus three time plus three?
B: Nine.
A: Right. And minus three times minus three?
B: Minus nine. [J91, 172–176].

Plain Affirmative Answer and Rejection The typical context of these two classes of NSUs is a polar question.

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

They can also answer *implicit* polar questions, e.g. CE questions like (11).

(11) A: That one?
B: Yeah. [G4K, 106–107]

Rejections can also be used to respond to assertions:

(12) A: I think I left it too long.
B: No no.[G43, 26–27]

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

Repeated Affirmative Answer Typically, repeated affirmative answers are responses to polar questions. They answer affirmatively by repeating a fragment of the query.

(13) A: Did you shout very loud?
B: Very loud, yes. [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 (14). As responses to assertions, they correct some piece of information in the previous utterance (15).

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

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

Plain Acknowledgement The class plain acknowledgement refers to utterances (like e.g. "yeah", "mhm", "ok") that signal that a previous declarative utterance was understood and/or accepted.

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

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

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

Propositional and Factual Modifiers These two NSU classes are used to classify propositional adverbs like (18) and factual adjectives like (19), respectively, in stand-alone uses.

(18) A: I wonder if that would be worth getting?
B: Probably not. [H61, 81–82]

(19) A: So we we have proper logs? Over there?
B: It's possible.
A: Brilliant! [KSV, 2991–2994]

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

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

Conjunction + fragment This NSU class is used to classify fragments introduced by conjunctions.

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

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

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

3 The Corpus

To generate the data for our experiments, we collected a corpus of NSUs extracted from the dialogue transcripts of the British National Corpus (BNC) (Burnard, 2000).

Our corpus of NSUs includes and extends the subcorpus used in (Fernández and Ginzburg, 2002). It was created by manual examination of a randomly selected section of 200-speaker-turns from 54 BNC files. The examined sub-corpus contains 14,315 sentences. We found a total of 1285 NSUs. Of these, 1269 were labelled according to the typology presented in the previous section. We also annotated each of these NSUs with the sentence number of its antecedent utterance. The remaining 16 instances did not fall in any of the categories of the taxonomy. They were labelled as 'Other' and were not used in the experiments.

NSU class	**Total**
Plain Acknowledgement	582
Short Answer	105
Affirmative Answer	100
Repeated Ack.	80
CE	66
Rejection	48
Repeated Aff. Ans.	25
Factual Modifier	23
Sluice	20
Helpful Rejection	18
Filler	16
Bare Mod. Phrase	10
Propositional. Modifier	10
Conjunction + frag	5
Total dataset	**1109**

Table 1: NSU sub-corpus

The labelling of the entire corpus of NSUs was done by one expert annotator. To assess the reliability of the taxonomy we performed a pilot study with two additional, non-expert annotators. These annotated a total of 50 randomly selected instances (containing a minimum of 2 instances of each NSU class as labelled by the expert annotator) with the classes in the taxonomy. The agreement obtained by the three annotators is reasonably good, yielding a kappa 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 achieve 96% and 92% accuracy in this task.

The data used in the experiments was selected from our classified corpus of NSUs (1269 instances as labelled by the expert annotator) following two simplifying restrictions. First, we restrict our experi-

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(mpty)
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

Table 2: Features and values

ments 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 classified according to the taxonomy, whose antecedent is the immediately preceding utterance contains a total of 1109 datapoints. Table 1 shows the frequency distribution for NSU classes.

The second restriction concerns the instances classified as plain acknowledgements. Taking the risk of ending up with a considerably smaller data set, we decided to leave aside this class of feedback NSUs, given that (i) they make up more than 50% of our sub-corpus leading to a data set with very skewed distributions, and (ii) a priori, they seem one of the easiest types to identify (a hypothesis that was confirmed after a second experiment—see below). We therefore exclude plain acknowledgements and concentrate on a more interesting and less skewed data set containing all remaining NSU classes. This makes up a total of 527 data points ($1109 - 582$). In section 7.3 we will compare the results obtained using this restricted data set with those of a second experiment in which plain acknowledgements are incorporated.

4 Experimental Setup

In this section we present the features used in our experiments and describe the automatic procedure that we employed to annotate the 527 data points with these features.

4.1 Features

We identify three types of properties that play an important role in the NSU classification task. 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 2 shows the set of nine features used in our experiments.

NSU features A set of four features are related to properties of the NSUs. These are nsu_cont,wh_nsu,aff_neg and lex. We expect the feature nsu_cont to distinguish between question-denoting and proposition-denoting NSUs. The feature wh_nsu is primarily introduced to identify Sluices. The features aff_neg and lex signal the presence of particular lexical items. They include a value (e)mpty which allows us to encode the absence of the relevant lexical items as well. We expect these features to be crucial to the identification of Affirmative Answers and Rejection on the one hand, and Propositional Modifiers, Factual Modifiers, Bare Modifier Phrases and Conjunction + fragment NSUs on the other.

Note that the feature lex could be split into four binary features, one for each of its non-empty values. We have experimented with this option as well, and the results obtained are virtually the same. We therefore opt for a more compact set of features. This also applies to the feature aff_neg.

Antecedent features We use the features `ant_mood`, `wh_ant`, and `finished` to encode properties of the antecedent utterance. The presence of a *wh*-phrase in the antecedent seems to be the best cue for classifying Short Answers. We expect the feature `finished` to help the learners identify Fillers.

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 our feature set. The feature `repeat` is introduced as a clue to identify Repeated Affirmative Answers and Repeated Acknowledgements. The feature `parallel` 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.

4.2 Data generation

Our feature annotation procedure is similar to the one used in (Fernández et al., 2004), which exploits the SGML markup of the BNC. All feature values are extracted automatically using the PoS information encoded in the BNC markup. The BNC was automatically annotated with a set of 57 PoS codes (known as the C5 tagset), plus 4 codes for punctuation tags, using the CLAWS system (Garside, 1987).

Some of our features, like `nsu_cont` and `ant_mood`, for instance, are *high level* features that do not have straightforward correlates in PoS tags. Punctuation tags (that would correspond to intonation patterns in a spoken dialogue system) help to extract the values of these features, but the correspondence is still not unique. For this reason we evaluate our automatic feature annotation procedure against a small sample of manually annotated data.

We randomly selected 10% of our dataset (52 instances) and extracted the feature values manually. In comparison with this gold standard, our automatic feature annotation procedure achieves 89% accuracy.

We use only automatically annotated data for the learning experiments.

5 Baseline

The simplest baseline we can consider is to always predict the majority class in the data, in our case Short Answer. This yields a 6.7% weighted f-score.

A slightly more interesting baseline can be obtained by using a one-rule classifier. It chooses the feature which produces the minimum error. This creates a single rule which generates a decision tree where the root is the chosen feature 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. We use the implementation of a one-rule classifier provided in the Weka toolkit (Witten and Frank, 2000).

In our case, the feature with the minimum error is `aff_neg`. It produces the following one-rule decision tree, which yields a 32.5% weighted f-score.

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

Figure 1: One-rule tree

Finally, we consider a more substantial baseline that uses the combination of four features that produces the best results. The four-rule tree is constructed by running the J4.8 classifier (Weka's implementation of the C4.5 system (Quinlan, 1993)) with all features and extracting only the four first features from the root of the tree, which interestingly are all NSU features. This creates a decision tree with four rules, one for each feature used, and nine leaves corresponding to nine different NSU classes.

```
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 2: Four-rule tree

This four-rule baseline yields a 62.33% weighted

f-score. Detailed results for the three baselines considered are shown in Tables 3, 4 and 5, respectively. All results reported were obtained by performing 10-fold cross-validation.

The results (here and in the sequel) are presented as follows: The tables show the recall, precision and f-measure for each class. To calculate the overall performance of the algorithm, we normalise those scores 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 the bottom row of the tables, is then the sum of the corresponding weighted scores.

NSU class	**recall**	**prec**	**f1**
ShortAns	100.00	20.10	33.50
Weighted Score	**19.92**	**4.00**	**6.67**

Table 3: Majority class baseline

NSU class	**recall**	**prec**	**f1**
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**

Table 4: One-rule baseline

NSU class	**recall**	**prec**	**f1**
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
ConjFrag	100.00	71.43	83.33
Weighted Score	**70.40**	**55.92**	**62.33**

Table 5: Four-rule baseline

6 Machine Learners

We use three different machine learners, which implement three different learning strategies: SLIPPER, a rule induction system presented in (Cohen and Singer, 1999); TiMBL, a memory-based algorithm created by (Daelemans et al., 2003); and MaxEnt, a maximum entropy algorithm developed by Zhang Le (Le, 2003). They are all well established, freely available systems.

SLIPPER As in the case of Weka's J4.8, SLIPPER is based on the popular C4.5 decision tree algorithm. SLIPPER improves this algorithm by using iterative pruning and confidence-rated boosting to create a weighted rule set. We use SLIPPER's option `unordered`, which finds a rule set that separates each class from the remaining classes, giving rules for each class. This yields slightly better results than the default setting. Unfortunately, it is not possible to access the output rule set when cross-validation is performed.

TiMBL As with all memory-based learning algorithms, TiMBL computes the similarity between a new test instance and the training instances stored in memory using a distance metric. As a distance metric we use the *modified value difference metric*, which performs better than the default setting (*overlap metric*). In light of recent studies (Daelemans and Hoste, 2002), it is likely that the performance of TiMBL could be improved by a more systematic optimisation of its parameters, as e.g. in the experiments presented in (Gabsil and Lemon, 2004). Here we only optimise the distance metric parameter and keep the default settings for the number of nearest neighbours and feature weighting method.

MaxEnt Finally, we experiment with a maximum entropy algorithm, which computes the model with the highest entropy of all models that satisfy the constraints provided by the features. The maximum entropy toolkit we use allows for several options. In our experiments we use 40 iterations of the default L-BFGS parameter estimation (Malouf, 2002).

7 Results: Evaluation and Discussion

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 those of the other algorithms is not statistically significant. Detailed recall, precision and f-measure scores are shown in Appendix I (Tables 8, 9 and 10).

7.1 Comparison with the baseline

The four-rule baseline algorithm discussed in section 5 yields a 62.33% weighted f-score. Our best result, the 87.75% weighted f-score obtained with the maximal entropy model, shows a 25.42% improvement over the baseline system. A comparison of the scores obtained with the different baselines considered and all learners used is given in Table 6.

System	w. f-score
Majority class baseline	6.67
One rule baseline	32.50
Four rule baseline	62.33
SLIPPER	86.35
TiMBL	86.66
MaxEnt	87.75

Table 6: Comparison of weighted f-scores

It is interesting to note that the four-rule baseline achieves very high f-scores with Sluices and CE—around 97% (see Table 5). Such results are not improved upon by the more sophisticated learners. This indicates that the features `nsu_cont` and `wh_nsu` used in the four-rule tree (figure 2) are sufficient indicators to classify question-denoting NSUs. The same applies to the classes Propositional Modifier and Factual Modifier. The baseline already gives f-scores of 100%. This is in fact not surprising, given that the disambiguation of these categories is tied to the presence of particular lexical items that are relatively easy to identify.

Affirmative Answers and 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. By using features that help to identify other categories with the machine learners we are able to improve the precision for Short Answers 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.

7.2 Error analysis: some comments

The class with the lowest scores is clearly Helpful Rejection. TiMBL achieves a 39.92% f-measure for this class. The maximal entropy model, however, yields only a 10.37% f-measure. Examination of the confusion matrices shows that ∼27% of Help Rejections were classified as Rejections, ∼15% as Repeated Acknowledgements, and ∼26% as Short Answers. This indicates that the feature `parallel`, introduced to identify this type of NSUs, is not a good enough cue. Whether similar techniques to the ones used e.g. in (Poesio et al., 2004; Schlangen, 2005) to compute semantic similarity could be used here to derive a notion of semantic contrast that would complement this structural feature is an issue that requires further investigation.

7.3 Incorporating plain acknowledgements

As explained in section 3, the data set used in the experiments reported in the previous sections excluded plain acknowledgements. The fact that plain acknowledgements are the category with the highest probability in the sub-corpus (making up more than 50% of our total data), and that they do not seem particularly difficult to identify could affect the performance of the learners by inflating the results. Therefore we left them out in order to work with a more balanced data set and to minimise the potential for misleading results. In a second experiment we incorporated plain acknowledgements 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 acknowledgements (“mhm”, “aha”, “right”, etc.). The total data set (1109 data points) was automatically annotated with the features modified in this way by means of the procedure described in section 4.2. The three machine learners were then run on the annotated data.

As in our first experiment the results obtained are very similar across learners. All systems yield around an 89% weighted f-score, a slightly higher result than the one obtained in the previous experiment. Detailed scores for each class are shown in Appendix II (Tables 11, 12 and 13). As expected, the class Plain Acknowledgement obtains a high f-score (95%). This, combined with its high probability, raises the overall performance a couple of points (from ∼87% to ∼89% weighted f-score). The improvement with respect to the baselines, however, is not as large: a simple majority class baseline already yields 36.28% weighted f-score. Table 7 shows a comparison of all weighted f-scores obtained in this second experiment.

System	**w. f-score**
Majority class baseline	36.28
One rule baseline	54.26
Four rule baseline	68.38
SLIPPER	89.51
TiMBL	89.65
MaxEnt	89.88

Table 7: Comparison of w. f-scores - with ack.

The feature with the minimum error used to derived the one-rule baseline is again `aff_neg`, this time with the new value `ack` as part of its possible values (see figure 3 below). The one-rule baseline yields a weighted f-score of 54.26%, while the four-rule baseline goes up to a weighted f-score of 68.38%.[4]

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

Figure 3: One-rule tree - with ack.

In general the results obtained when plain acknowledgements are added to the data set are very similar to the ones achieved before. Note however that even though the overall performance of the algorithms is slightly higher than before (due to the reasons mentioned above), the scores for some NSU classes are actually lower. The most striking case is the class Affirmative Answer, which in TiMBL goes down more than 10 points (from 93.61% to 82.42% f-score—see Tables 9 and 12 in the appendices). The tree in figure 3 provides a clue to the reason for this. When the NSU contains a "yes" word (first branch of the tree) the class with the highest probability is now Plain Acknowledgement, instead of Affirmative Answer as before. This is due to the fact that, at least in English, expressions like e.g. "yeah" (considered here as "yes" words) are potentially ambiguous between acknowledgements and affirmative answers.[5] 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 acknowledgements in our taxonomy) and non-backchannel fragments. This is a tricky problem for Schlangen as his fragment identification procedure does not have access to the context. Although 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 Affirmative Answer) is not always trivial.

8 Conclusions and Future Work

We have presented a machine learning approach to the problem of Non-Sentential Utterance (NSU) classification in dialogue. We have described a procedure for predicting the appropriate NSU class from a fine-grained typology of NSUs derived from a corpus study performed on the BNC, using a set of automatically annotated features. We have employed a series of simple baseline methods for classifying NSUs. The most successful of these methods uses a decision tree with four rules and gives a weighted f-score of 62.33%. We then applied three machine learning algorithms to a data set which includes all NSU classes except Plain Acknowledgement and obtained a weighted f-score of approx-

[4] The four-rule tree can be obtained by substituting the last node in the tree in figure 2 (section 5) for the one-rule tree in figure 3.

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

imated 87% for all of them. This improvement, taken together with the fact that the three algorithms achieve very similar results suggests that our features offer a reasonable basis for machine learning acquisition of the typology adopted. However, some features like `parallel`, introduced to account for Help Rejections, are in need of considerable improvement.

In a second experiment we incorporated plain acknowledgements in the data set and ran the machine learners on it. The results are very similar to the ones achieved in the previous experiment, if slightly higher due to the high probability of this class. The experiment does show though a potential confusion between plain acknowledgements and affirmative answers that did not show up in the previous experiment.

In future work we will integrate our NSU classification techniques into an Information State-based dialogue system (based on SHARDS (Fernández et al., to appear) and CLARIE (Purver, 2004)), that assigns a full sentential reading to fragment phrases in dialogue. This will require a refinement of our feature extraction procedure, which will not be restricted solely to PoS input, but will also benefit from other information generated by the system, such as dialogue history and intonation.

Acknowledgements

We would like to thank two anonymous SIGdial reviewers for their comments and suggestions. We would also like to thank Lief A. Nielsen and Matt Purver for discussion, and Zoran Macura and Yo Sato for help in assessing the NSU taxonomy. The research described here is funded by grant number RES-000-23-0065 from the Economic and Social Research Council of the United Kingdom.

References

Claire Beyssade and Jean-Marie Marandin. 2005. Contour Meaning and Dialogue Structure. Ms presented at the workshop Dialogue Modelling and Grammar, Paris, France.

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

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

W. Daelemans and V. Hoste. 2002. Evaluation of machine learning methods for natural language processing tasks. In *In Proceedings of the third International Conference on Language Resources and Evaluation (LREC-02)*, pages 755–760.

W. Daelemans, 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.

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

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

R. Fernández, J. Ginzburg, H. Gregory, and S. Lappin. (to appear). SHARDS: Fragment resolution in dialogue. In H. Bunt and R. Muskens, editors, *Computing Meaning*, volume 3. Kluwer.

M. Gabsil and O. Lemon. 2004. Combining acoustic and pragmatic features to predict recognition performance in spoken dialogue systems. In *In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, Barcelona, Spain.

R. Garside. 1987. The claws word-tagging system. In Roger Garside, Geoffrey Leech, and Geoffrey Sampson, editors, *The computational analysis of English: a corpus-based approach*, pages 30–41. Longman, Harlow.

Zhang Le. 2003. Maximum Entropy Modeling Toolkit for Python and C++. http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.php.

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

M. Poesio, R. Mehta, A. Maroudas, and J. Hitzeman. 2004. Learning to resolve bridging references. In *Proceedings of the 42nd Annual Meeting of the ACL (ACL 2004)*, pages 144–151, Barcelona, Spain.

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

R. Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco.

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

D. Schlangen. 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 ACL (ACL 2005)*, USA. Ann Arbor.

I. H. Witten and E. Frank. 2000. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco. http://www.cs.waikato.ac.nz/ml/weka.

Appendix I: Results w/o plain acknowledgements (527 datapoints)

NSU class	recall	prec	f1
CE	93.64	97.22	95.40
Sluice	96.67	91.67	94.10
ShortAns	83.93	82.91	83.41
AffAns	93.13	91.63	92.38
Reject	83.60	100.00	91.06
RepAffAns	53.33	61.11	56.96
RepAck	85.71	89.63	87.62
HelpReject	28.12	20.83	23.94
PropMod	100.00	90.00	94.74
FactMod	100.00	100.00	100.00
BareModPh	100.00	80.56	89.23
ConjFrag	100.00	100.00	100.00
Filler	100.00	62.50	76.92
Weighted Score	**86.21**	**86.49**	**86.35**

Table 8: SLIPPER

NSU class	recall	prec	f1
CE	94.37	91.99	93.16
Sluice	94.17	91.67	92.90
ShortAns	88.21	83.00	85.52
AffAns	92.54	94.72	93.62
Reject	95.24	81.99	88.12
RepAffAns	63.89	60.19	61.98
RepAck	86.85	91.09	88.92
HelpReject	35.71	45.24	39.92
PropMod	90.00	100.00	94.74
FactMod	97.22	100.00	98.59
BareModPh	80.56	100.00	89.23
ConjFrag	100.00	100.00	100.00
Filler	48.61	91.67	63.53
Weighted Score	**86.71**	**87.25**	**86.66**

Table 9: TiMBL

NSU class	recall	prec	f1
CE	96.11	96.39	96.25
Sluice	100.00	95.83	97.87
ShortAns	89.35	83.59	86.37
AffAns	92.79	97.00	94.85
Reject	100.00	81.13	89.58
RepAffAns	68.52	65.93	67.20
RepAck	84.52	81.99	83.24
HelpReject	5.56	77.78	10.37
PropMod	100.00	100.00	100.00
FactMod	97.50	100.00	98.73
BareModPh	69.44	100.00	81.97
ConjFrag	100.00	100.00	100.00
Filler	62.50	90.62	73.98
Weighted Score	**87.11**	**88.41**	**87.75**

Table 10: MaxEnt

Appendix II: Results with plain acknowledgements (1109 datapoints)

NSU class	recall	prec	f1
Ack	95.42	94.65	95.03
CE	95.00	94.40	94.70
Sluice	98.00	93.33	95.61
ShortAns	87.32	86.33	86.82
AffAns	82.40	86.12	84.22
Reject	79.01	100.00	88.28
RepAffAns	60.33	81.67	69.40
RepAck	81.81	87.36	84.49
HelpReject	37.50	21.88	27.63
PropMod	80.00	80.00	80.00
FactMod	100.00	100.00	100.00
BareModPh	57.14	57.14	57.14
ConjFrag	100.00	100.00	100.00
Filler	59.38	40.62	48.24
Weighted Score	**89.18**	**90.16**	**89.51**

Table 11: SLIPPER

NSU class	recall	prec	f1
Ack	95.61	95.16	95.38
CE	92.74	95.00	93.86
Sluice	100.00	98.00	98.99
ShortAns	85.56	84.58	85.07
AffAns	80.11	84.87	82.42
Reject	95.83	78.33	86.20
RepAffAns	70.37	66.67	68.47
RepAck	85.06	82.10	83.55
HelpReject	31.25	38.54	34.51
PropMod	100.00	100.00	100.00
FactMod	100.00	100.00	100.00
BareModPh	78.57	85.71	81.99
ConjFrag	100.00	87.50	93.33
Filler	40.62	53.12	46.04
Weighted Score	**90.00**	**89.45**	**89.65**

Table 12: TiMBL

NSU class	recall	prec	f1
Ack	95.61	95.69	95.65
CE	95.24	95.00	95.12
Sluice	100.00	98.00	98.99
ShortAns	87.00	83.94	85.44
AffAns	86.12	85.23	85.67
Reject	97.50	79.94	87.85
RepAffAns	68.33	66.67	67.49
RepAck	84.23	77.63	80.80
HelpReject	6.25	75.00	11.54
PropMod	100.00	100.00	100.00
FactMod	96.88	100.00	98.41
BareModPh	71.43	100.00	83.33
ConjFrag	100.00	100.00	100.00
Filler	46.88	81.25	59.45
Weighted Score	**90.35**	**90.63**	**89.88**

Table 13: MaxEnt