Final Request Analysis Component

Andreas Vlachos, Stephen Clark, Tiphaine Dalmas, Robin Hill,
Srini Janarthanam, Oliver Lemon, Michael Minock, Bonnie Webber

Distribution: Public

SpaceBook
Spatial & Personal Adaptive Communication Environment: Behaviors & Objects & Operations & Knowledge
270019 Deliverable 4.1.2

August 31, 2013

Project funded by the European Community
under the Seventh Framework Programme for
Research and Technological Development

The deliverable identification sheet is to be found on the reverse of this page.

Project ref. no.	270019
Project acronym	SpaceBook
Project full title	Spatial & Personal Adaptive Communication Environment: Behaviors & Objects & Operations & Knowledge
Instrument	STREP
Thematic Priority	Cognitive Systems, Interaction, and Robotics
Start date / duration	01 March 2011 / 36 Months

Security	Public
Contractual date of delivery	M30 = August 2013
Actual date of delivery	August 31, 2013
Deliverable number	4.1.2
Deliverable title	Final Request Analysis Component
Type	Prototype
Status & version	Final 1.0
Number of pages	14 (excluding front matter)
Contributing WP	4
WP/Task responsible	UCAM, UMU
Other contributors	
Author(s)	Andreas Vlachos, Stephen Clark, Tiphaine Dalmas, Robin Hill, Srini Janarthanam, Oliver Lemon, Michael Minock, Bonnie Webber
EC Project Officer	Franco Mastroddi
Keywords	Semantic Parsing Corpus, Dialog Act Tagging

The partners in SpaceBook are:

Umeå University	UMU
University of Edinburgh HCRC	UE
Heriot-Watt University	HWU
Kungliga Tekniska Högskola	KTH
Liquid Media AB	LM
University of Cambridge	UCAM
Universitat Pompeu Fabra	UPF

For copies of reports, updates on project activities and other SPACEBOOK-related information, contact:

The SPACEBOOK Project Co-ordinator:
Dr. Michael Minock
Department of Computer Science
Umeå University
Sweden 90187
mjm@cs.umu.se
Phone +46 70 597 2585 - Fax +46 90 786 6126

Copies of reports and other material can also be accessed via the project's administration homepage, `http://www.spacebook-project.eu`

Contents

Executive summary

This document describes progress towards the final prototype for the Natural Language Understanding component of SPACEBOOK, which will be referred to as the Semantic Parser. A large part of the document will describe the creation of the Semantic Parsing Corpus, which will be used to train the Semantic Parser. The document will also describe preliminary Machine Learning experiments on the corpus, in which a Dialog Act Tagger has been trained and tested. Dialog act tagging is the first phase in semantic parsing. These developments have occurred by month 30 of the project. Completion of the full Semantic Parser will take place during the remainder of the project and be reported in D4.1.3, the Report on learning semantic analysis components.

1 Introduction

This report follows on from Deliverable 4.1.1 — Initial Request Analysis Component — in which the semantic parsing problem was formulated and an initial rule-based semantic parser was described. This report focuses on the main approach to semantic parsing described in the earlier deliverable, namely to automatically learn a semantic grammar and parser from training data, where the data consists of pairs of SPACEBOOK-type utterances and their corresponding logical forms. The SPACEBOOK-type utterances are transcriptions of real speech captured during Wizard-of-Oz experiments, with the intention that these will be close to the output of the speech recognizer used in the SPACEBOOK-system. The logical forms are represented using the Meaning Representation Language (MRL) developed for SPACEBOOK, which was described in detail in the earlier deliverable.

We focus on the creation of the semantic parsing corpus, which will provide the training and testing data for the development of the semantic parser. Creating such a corpus is a substantial contribution in its own right, requiring careful guidlelines and many person-hours of annotation work. Much of the annotation was carried out by Diane Nichols, a freelance linguist, and Andreas Vlachos (UCAM), with further checking by Stephen Clark (UCAM).

The remainder of the report provides further description of the MRL (also detailed in D4.1.1, but included here to make this report self-contained); and then describes some preliminary semantic parsing experiments focusing on dialog act tagging, which is the first stage in the semantic parsing pipeline. In summary, we have created a substantial semantic parsing corpus based on two dialog scenarios, consisting in total of 17 dialogs and 2374 user utterances. We also conducted what we believe is the first inter-annotator agreement study on semantic parsing annotation, reporting 80.4 exact match agreement between the two annotators. The highest accuracy achieved on the dialog act tagging task was 81.0, thus suggesting that the task is feasible but not trivial.

2 Meaning Representation Language

The MRL uses a flat syntax composed of elementary predications, based loosely on minimal recursion semantics [1], but without an explicit treatment of scope. Each meaning representation (MR) consists of a dialog act representing the overall function of the utterance, followed for some dialog acts by an unordered set of predicates. All predicates are implicitly conjoined and the names of their arguments specified to improve readability and to allow for some of the arguments to be optional. The argument values can be either constants from the controlled vocabulary, verbatim string extracts from the utterance (enclosed in quotes) or variables (`Xno`). Negation is denoted by a tilde ($\sim$) in front of predicates. The variables are used to bind together the arguments of different predicates within an utterance, as well as to denote coreference across utterances. Figure 1 contains part of an example dialog annotated with the MRL.

Annotation with the MRL is not text bound; i.e. we do not specify the alignment between tokens in the utterance and elements in the MR, apart from the verbatim string extracts. Even though some supervision can be helpful for automatically inferring such alignments [2], the annotation task is non-trivial, and often it is not possible to decide on a single correct alignment. For example, in the first utterance in Figure 1 the `isA` predicate denoting the restaurant cannot be aligned with a particular token or span in the utterance since neither "italian" nor "meal" on their own can represent it. Furthermore, including the tokens between "italian" and "meal" would not be ideal either since many of them are not relevant to this predicate.

Finally, even though the proposed MRL is designed to be machine-interpretable, it is not executable

USER what's the nearest italian, em, for a meal?

```
dialogAct(set_question)
*isA(id:X1, type:restaurant)
def(id:X1)
hasProperty(id:X1, property:cuisine,
            value:"italian")
distance(location:@USER,
         location:X1, value:X2)
argmin(argument:X1, value:X2)
```

WIZARD vapiano's.

```
dialogAct(inform)
isA(id:X4, type:restaurant)
*isNamed(id:X4, name:"vapiano's")
equivalent(id:X1, id:X4)
```

USER take me to vapiano!

```
dialogAct(set_question)
*route(from_location:@USER,
       to_location:X4)
isA(id:X4, type:restaurant)
isNamed(id:X4, name:"vapiano")
```

WIZARD keep walking straight down clerk street.

```
dialogAct(instruct)
*walk(agent:@USER, along_location:X1,
      direction:forward)
isA(id:X1, type:street)
isNamed(id:X1, name:"clerk street")
```

USER what is this church?

```
dialogAct(set_question)
*isA(id:X2, type:church)
index(id:X2)
```

WIZARD sorry, can you say this again?

```
dialogAct(repeat)
```

USER i said what is this church on my left!

```
dialogAct(set_question)
*isA(id:X2, type:church)
index(id:X2)
position(id:X2, ref:@USER,
         location:left)
```

WIZARD it is saint john's.

```
dialogAct(inform)
isA(id:X3, type:church)
*isNamed(id:X3, name:"saint john's")
equivalent(id:X2, id:X3)
```

USER A sign here says it is saint mark's.

```
dialogAct(inform)
isA(id:X4, type:church)
*isNamed(id:X4, name:"saint mark's")
equivalent(id:X2, id:X4)
```

Figure 1: Sample annotated dialog

directly on a database. While such a property would be desirable in some respects — for example it would allow response-based learning — it would also restrict us to an MRL that cannot capture semantics beyond the database. However, for SPACEBOOK such semantics are necessary since many user utterances are not interpretable as database queries or would be difficult to represent as such, e.g. repetition requests. We assume instead that the Interaction Manager handles the interaction with the database, and in order to facilitate this interaction the controlled vocabulary used in the MRL corresponds closely to the schema of the SPACEBOOK database.

2.1 Dialog acts

The dialog acts are utterance-level labels which capture the overall function of the utterance in the dialog, for example whether an utterance is a question seeking a list as an answer, a statement of information, an acknowledgement, an instruction or a repetition request (`set_question`, `inform`, `acknowledge`, `instruct` and `repeat`). The acts defined in the MRL follow the guidelines proposed by [3] and [4].

The dialog acts in the MRL are divided into two categories. The first category contains those that are accompanied by a set of predicates to represent the semantics of the sentence, such as `set_question` and `inform`. For these acts we denote their focal points — for example the piece of information requested in a `set_question` — with an asterisk (`*`) in front of the relevant predicate. The focal point together with the act provide similar information to the intent annotation in ATIS [5]. The second category contains dialog acts that are not accompanied by predicates, such as `acknowledge` and `repeat`. These are used to annotate utterances whose function in the dialog is clear and simple, even though their actual semantics might be rather complex and possibly beyond what the controlled vocabulary of the MRL can accommodate.

2.2 Predicates

The MRL contains predicates with a variety of arguments to introduce entities, properties of entities, and their relations:

- Predicates introducing entities and their properties: `isA`, `isNamed` and `hasProperty`.
- Predicates describing user actions, such as `walk` and `turn`, with arguments such as `direction` and `along_location` to express the various modes of action.
- Predicates describing geographic relations, such as `distance`, `route` and `position`. The latter uses the argument `ref` in order to denote relative positioning.
- Predicates denoting whether an entity is introduced using a definite article (`def`), an indefinite (`indef`) or an indexical (`index`), which are useful in determining which real-world entity is being referred to.
- Predicates expressing numerical relations such as `argmin` and `argmax`.

2.3 Coreference

In order to model coreference we adopt the notion of discourse referents (DRs) and discourse entities (DEs) from Discourse Representation Theory (DRT) [6] based on earlier work such as [7].

DRs are referential expressions appearing in utterances which denote DEs. DEs are mental entities in the speaker's model of discourse (which do not necessarily correspond to real-world entities). Note also that multiple DEs can refer to the same real-world entity; for example, in Figure 1 "vapiano's" refers to a different DE from the restaurant in the previous sentence ("the nearest italian"), even though they are likely to be the same real-world entity. We considered DEs instead of actual entities in the MRL because they allow us to capture the semantics of interactions such as the last exchange between the wizard and user, in which the disagreement would not have been possible to represent without using DEs. The MRL represents multiple DEs referring to the same real-world entity through the predicate `equivalent`.

Coreference is achieved by using identical variables across predicate arguments within an utterance or across utterances. The main principle in determining whether DRs corefer is that it must be possible to infer this from the dialog context alone, without using world knowledge. For example, in Figure 1 "vapiano's" and "vapiano" are assumed to refer to the same DE even though they are different names, because it is clear from the dialog that the user is referring to the DE mentioned by the wizard.

3 The Corpus

The utterances for corpus annotation were collected using Wizard-of-Oz experiments with pairs of human subjects, described in Deliverable 6.1.2. We give a short description here for completeness. In each experiment, one human pretended to be a tourist visiting Edinburgh (by physically walking around the city), while the other performed the role of the system responding through a suitable interface using a text-to-speech system.

Each user-wizard pair was given one of two scenarios involving requests for directions to different points of interest and information about them, as well as the system offering information considered of interest to the user. The first scenario involves seeking directions to the national museum of Scotland, then wanting to go to a nearby coffee shop, followed by a pub via a cash machine and finally looking for a park. The second scenario involves looking for a Japanese restaurant, then asking for directions to the university gym, requesting information about the Flodden Wall monument, visiting the Scottish parliament and the Dynamic Earth science centre, and going to the Royal Mile and the Surgeon's Hall museum. Each experiment formed one dialog which was manually transcribed from recorded audio files. 17 dialogs were collected in total, seven from the first scenario and 10 from the second.

Each scenario was designed to last approximately one hour, but the actual execution time (and number of utterances collected) varied in each experiment depending on the amount of interaction between the user and the wizard. The users were encouraged to ask for information according to their interests, which resulted in a wide range of tourism-related discussions, such as foreign currency exchange rates and architecture. Furthermore, allowing the wizards to answer in natural language instead of restricting them to responding via database queries (as in the ATIS corpus, for example) led to more varied dialogs. Thus, the data collected from the Wizard-of-Oz experiments allowed us to gain insights into the kinds of information users might expect from the SPACEBOOK application and how they might ask for it.

However, this also resulted in some of the user requests not being within the scope of the system. Furthermore, the proposed MRL has its own limitations; for example it does not have predicates to express temporal relationships. Thus, it was necessary to filter the utterances collected and decide which ones to annotate with MRs.[1] In particular, we did not annotate utterances falling into one or more of the following

[1] A similar filtering process was used for GeoQuery (Section 7.5.1 in [8]) and ATIS (principles of interpretation

	new corpus	GeoQuery	ATIS
user utterances	2374	880	5871
utterances/dialog	139.7	1	8.8
unique NL words	896	280	611
MRL vocabulary	115	35	85

Table 1: Corpus comparison. Figures for GeoQuery are from [9], and for ATIS from [10] except for average dialog length which is from [11].

categories:

- Utterances that are not human-interpretable, e.g. utterances that were interrupted too early to be interpretable.
- Utterances that are human-interpretable but outside the scope of the system, e.g. questions about historical events which are not included in the database of the application considered.
- Utterances that are within the scope of the system but too complex be represented by the proposed MRL, e.g. an utterance requiring representation of time to be interpreted.

Note that when the core of an utterance can be captured adequately by the MRL, we opt to annotate it with an appropriate MR even if some of the entities mentioned are outside the scope of the system, or if some of the language used is too complex. For example, the final utterance in Figure 1 mentions a sign which is outside the scope of the system, but we still annotated the utterance since its interpretation with respect to the application is not affected. We argue that determining which utterances should be translated into MRs is an important subtask for real-world applications of semantic parsing and decided to keep these utterances in the corpus.

3.1 Annotation

The annotation was performed by Andreas Vlachos and Diane Nichols, a freelance linguist with no experience in semantic parsing, with further checking by Stephen Clark. As well as annotating the user utterances, we also annotated the wizard utterances with dialog acts and the entities mentioned, as they provide the necessary context to perform context-dependent interpretation. In practice, though, we expect this information to be used by the natural language generation component to produce the SPACEBOOK's response and thus be available to the semantic parser.

The total number of user utterances annotated was 2374, out of which 1906 were annotated with MRs, the remaining not translated due to the reasons discussed above. Table 1 has more statistics, including a comparison with the popular GeoQuery and ATIS corpora. The number and types of the MRL vocabulary terms used in the annotation appear in Table 2. The full lists of controlled vocabulary terms can be found in the appendix.

In order to assess the quality of the guidelines and the annotation, we conducted an inter-annotator agreement study. For this purpose, the two annotators annotated one dialog consisting of 510 utterances. Exact

document (`/atis3/doc/pofi.doc`) in the NIST CDs).

vocabulary type	number of terms
dialog acts	15
predicates	19
arguments	41
constants	9
entity types	26
properties	4

Table 2: MRL vocabulary used in the annotation

match agreement at the utterance level, which requires that the MRs by the annotators for an utterance agree on dialog act, predicates and within-utterance variable assignment, was 0.804, which is a strong result given the complexity of the annotation task, and which suggests that the MRL and guidelines for it can be applied consistently. We also assessed the agreement on predicates using F-score, which was 0.914.

Variable assignment was more challenging to assess since agreement between two annotations relies on measuring the identity between the variables assigned to different arguments rather than the variable names themselves. Since variable names cannot be taken into account, we decided to treat each variable as a cluster of argument slots and evaluate variable assignment as a clustering task. We chose to use information-theoretic clustering evaluation measures which avoid the problem of cluster mapping; in particular we used the adjusted mutual information (AMI) measure [12] as implemented in the scikit-learn toolkit [13]. AMI scores range from 0 to 1 and, unlike the more commonly used V-measure [14], are adjusted for chance, assigning scores close to 0 for random clusterings. The AMI was found to be 0.974 at the utterance level, while for the whole dialog it was 0.948. Note that the commonly used Kappa statistic [15] could not have been used for any of the evaluations given above, since it can only be applied to classification tasks.

4 Dialog Act Tagging

In this section we describe the implementation of two dialog act taggers for the new corpus, one based on a classification approach, using the AROW online learning algorithm, and the other treating the task as a sequence labelling problem, using a CRF toolkit. Determining the dialog act of an utterance is an important stage in the SPACEBOOK semantic parsing task since, as described in above, some acts are accompanied by predicates thus requiring further processing, while others are not. Hence, good performance at this first stage is crucial to semantic parsing performance.

More specifically, dialog act tagging in the new corpus is the task of assigning to each user utterance one of 14 labels; 13 labels correspond to acts (two of the 15 acts defined are only relevant to wizard utterances) plus an additional label for utterances that cannot be translated into MRs due to the reasons discussed above. We consider the dialog acts for the wizard utterances to be given as part of the input, since we assume that they will be available for the SPACEBOOK application.

The dialog act tagging task as defined by the application is greedy in nature. Every time a new user utterance is encountered it needs to be labeled by the tagger, and this prediction is used to generate the

response by the SPACEBOOK system and it is likely to affect how the dialog evolves. Thus, it is natural to consider implementing a dialog act tagger using a classifier (which will be the first approach). However, the utterances of both the wizard and the user are fixed in the new corpus and thus the dialog is not affected by the acts predicted; i.e. the corpus dialog is not "dynamic" in the sense that a real dialog would be. For example, if in a real dialog the system inferred that a user utterance was a proposition question, then the next system utterance would be determined by that, even if the inference were incorrect. However, for the corpus-based scenario, the next system utterance has already occurred and is fixed. Therefore, the dialog act tagging task as defined by the corpus can also be seen as the prediction of a sequence of acts for the user utterances in a dialog (which will be the second approach).

We developed the classifier-based dialog act tagger by implementing the adaptive regularization of weight vectors (AROW) learning algorithm [16]. AROW is an online learning algorithm for linear predictors that takes into account the rarity of each feature and adjusts the per-feature learning rates so that popular features do not overshadow rare but useful ones. Adaptive learning rates have been shown to be beneficial in other NLP tasks such as text classification [17]. Since we are operating in a batch learning setting (i.e. we have access to all the training instances simultaneously), we perform multiple rounds over the training instances randomly shuffling their order.

For the sequential prediction version of the tagger we used linear chain conditional random fields (CRFs) [18] trained with stochastic gradient descent as implemented in the CRFsuite toolkit [19]. CRFs, unlike classification methods, find the most likely dialog act assignment to a sequence of user utterances jointly. Furthermore, unlike hidden Markov Models, CRFs learn conditional models and thus are able to use arbitrary and possibly overlapping features over the input, so they can use the same features as classification-based approaches.

In both approaches, utterances are represented using two sets of features: those based on the textual content of the utterance and those based on the utterance preceding it. The former contains all unigrams, bigrams and trigrams and final punctuation mark (if any). Unlike in typical text classification tasks, content words are not always helpful in dialog act tagging; e.g. the word "church" in the user utterances in Figure 1 is not indicative of `set_question`, while n-grams of words typically considered as stopwords, such as "what is this", can be more helpful. The punctuation feature is used as a proxy to the prosodical information commonly used in dialog act tagging for spoken language [3]. The features based on the preceding utterance contain whether it was by the user or the wizard and in the latter case its dialog act. Such features are useful in determining the act of short, ambiguous utterances such as "yes" which is tagged as `yes` when following a `prop_question` utterance by the wizard but as `acknowledge` otherwise.

4.1 Experiments

In order to assess the performance of the two dialog act taggers described in the previous section, we split the annotated dialogs into two sets, one for training (and development) and one for testing. The former consists of four dialogs within the first scenario and seven within the second (11 in total), while the latter consists of three dialogs within each scenario (six in total). All development and feature engineering was conducted using cross-validation at the dialog level on the training set. Cross-validation at the dialog level instead of the utterance level ensures that each fold contains all utterances from complete dialogs, rather than being made up of incomplete dialogs. The results reported below were obtained by training on dialogs from the training set and evaluating on dialogs from the testing set. Since we modeled dialog act tagging as a multiclass classification task we evaluate performance using accuracy.

setup	training	testing	AROW	CRF
mixed	s1+s2	s1+s2	0.81	0.76
same	s1	s1	0.649	0.576
	s2	s2	0.84	0.752
	overall		0.757	0.676
cross	s1	s2	0.665	0.644
	s2	s1	0.706	0.721
	overall		0.672	0.667

Figure 2: Accuracy under different evaluation setups.

We considered three evaluation setups. In the first setup, all training dialogs from both scenarios were used to train the taggers, which are then evaluated on all testing dialogs from both scenarios. We refer to this setup as *mixed*. A similar setup is commonly used in semantic parsing evaluations on the ATIS corpus, in which dialogs concerning the same flight request are included in training, development and testing sets defined. While this is a reasonable evaluation approach, it is likely to be relatively forgiving, since in practice dialog act taggers and semantic parsers are likely to encounter scenarios unseen in their training/development data. Therefore we considered a second evaluation setup in which the dialogs used to train the taggers are from different scenario(s) than the scenario of the dialogs used to evaluate their performance. We refer to this setup as *cross*. Finally, in order to assess the effect of the change in scenarios used for training and testing, we considered a third evaluation setup in which the taggers are trained and evaluated on training and testing dialogs from the same scenario, which we refer to as *same*. Since we have dialogs from two scenarios in the new corpus, there are two experiments for the *same* and *cross* setups, and we also present the overall accuracy across scenarios for each setup.

The accuracy of the two taggers is shown in Figure 2. Both taggers achieve promising performance in all setups, including the more challenging *cross* setup. The AROW-based tagger achieves higher accuracy in all cases except for one of the two *cross* experiments, thus suggesting that structure —at least as exploited by the linear CRF and bearing in mind the other differences between the two learning algorithms— is not as crucial to performance. This can be expected since user utterances usually occur in response to those by the wizard (i.e. user utterances are more dependent on what the wizard just said, rather than the previous user utterance which is what the CRF models as structure). The latter provide useful features which are used by both approaches. Nevertheless, the difference in accuracy in the *cross* setup is much smaller than in the *same* setup, thus suggesting that structure is more useful in that setup, and that the primarily lexicalized features used might be less able to generalize well.

As there is substantial imbalance among the dialog acts in the corpus, we also evaluated performance for each act using recall, precision and F-score. The results for the two taggers in the *mixed* setup are shown in Table 3. Overall, the AROW-based tagger has better-balanced predictions and achieves higher F-scores for most dialog acts, an effect more pronounced in the rarer labels. The same trend was also observed in the other setups. The only act in which the CRF-based tagger performs better is `prop_question` (questions seeking a yes or no answer). Inspection of the training data showed that user utterances with this dialog act are usually followed by an `acknowledge` user utterance, often in response to an answer by the wizard

dialog act	training instances	testing instances	AROW Recall	AROW Precision	AROW F-score	CRF Recall	CRF Precision	CRF F-score
acknowledge	670	172	0.913	0.935	0.924	0.93	0.856	0.891
inform	266	131	0.855	0.862	0.858	0.855	0.824	0.839
set_question	267	96	0.927	0.774	0.844	0.906	0.744	0.817
prop_question	77	33	0.515	0.531	0.523	0.545	0.857	0.667
yes	38	12	1.0	0.6	0.75	0.667	0.727	0.696
repeat	25	18	0.722	0.684	0.703	0.444	0.888	0.593
check_channel	8	29	0.724	1.0	0.84	0.138	1.0	0.242
greet	11	5	1.0	0.833	0.909	0.4	0.667	0.5
no	9	5	0.8	0.572	0.667	0.2	0.5	0.286
wait	7	5	0.4	1.0	0.572	0.0	0.0	0.0
next_instruct	6	4	0.25	1.0	0.4	0.0	0.0	0.0
apologise	9	1	1.0	0.333	0.5	0.0	0.0	0.0
ready	1	1	0.0	0.0	0.0	0.0	0.0	0.0
no_MRL	375	93	0.602	0.7	0.647	0.645	0.522	0.577

Table 3: Per dialog act performance in the mixed evaluation setup for the AROW-based and the CRF-based taggers.

to the question of the user. The CRF-based tagger, unlike the AROW-based one, can take advantage of such patterns in the data and thus achieve better performance. However, as explained in the previous section, such a sequential prediction approach which performs dialog act assignment jointly would not be applicable in the inherently greedy application context considered. Finally, one of the most difficult labels to predict for both taggers is `no_MRL`, i.e. the label assigned to utterances that cannot be translated into MRs. Such utterances are sometimes very similar to ones that should be translated and the distinction is dependent on whether the proposed MRL can capture their meaning adequately. Therefore we expect that performance on this label is likely to improve if dialog act tagging is performed as part of the semantic parsing task defined by the new corpus.

5 Conclusions

In this report we described the SPACEBOOK MRL covering the domain of tourism-related activities and a new corpus annotated with it. The MRL can handle dialog context such as coreference and can accommodate utterances that are not interpretable according to a database. We conducted an inter-annotator agreement study and found 0.804 exact match agreement between two annotators. The SPACEBOOK MRL uses a wider controlled vocabulary than GeoQuery and ATIS (see Table 1 for a comparison). On average, the dialogs are 15 times as long as the the dialogs of ATIS; thus coreference resolution is likely to be harder. Although the new corpus has fewer utterances than ATIS, it has a wider vocabulary of NL words. For these reasons, we expect this new corpus to be more challenging than previous commonly used semantic corpora.

We performed experiments on dialog act tagging (a task that existing corpora do not require) using both

classification-based and sequential prediction approaches achieving good performance. Furthermore, we demonstrated how the new corpus can accommodate more realistic evaluations than those usually conducted, by training on dialogs from different scenarios than the ones used for testing.

6 Difficulties and Future Work

The main reason that the full parser has not been developed in time for this Deliverable is the delay in obtaining the Wizard-of-Oz data at the start of the project. Creating a corpus with manual annotation is a significant, time-consuming exercise, requiring many months of annotation time. Hence once the delay in obtaining the original data was incurred, it was not possible to make that time up in producing the semantic parsing corpus.

In future work, over the next few months, we will develop an approach for the full semantic parsing task defined by the new corpus. This work will be reported in deliverable D4.1.3, the final report on learning semantic analysis components. More generally, as recent approaches to semantic parsing have achieved rather high accuracies on existing corpora, we believe the new corpus will be a useful resource for the wider research community in making further progress on the semantic parsing task. The corpus will be made freely available for this purpose.

References

[1] Ann Copestake, Dan Flickinger, Ivan Sag, and Carl Pollard. Minimal recursion semantics: An introduction. *Research in Language and Computation*, 3(2–3):281–332, 2005.

[2] Percy Liang, Michael I. Jordan, and Dan Klein. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 91–99, Singapore, 2009.

[3] Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373, 2000.

[4] Harry Bunt, Jan Alexandersson, Jae-Woong Choe, Alex Chengyu Fang, Koiti Hasida, Volha Petukhova, Andrei Popescu-Belis, and David Traum. Iso 24617-2: A semantically-based standard for dialogue annotation. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*, Istanbul, Turkey, 2012.

[5] Gokhan Tur, Dilek Hakkani-Tür, and Larry Heck. What's left to be understood in ATIS? In *IEEE Workshop on Spoken Language Technologies*, 2010.

[6] Hans Kamp and Uwe Reyle. *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer, Dordrecht, 1993.

[7] Bonnie Lynn Webber. *A Formal Approach to Discourse Anaphora*. PhD thesis, Harvard University, 1978.

[8] John M. Zelle. *Using Inductive Logic Programming to Automate the Construction of Natural Language Parsers*. PhD thesis, Department of Computer Sciences, The University of Texas at Austin, 1995.

[9] Yuk Wah Wong. *Learning for Semantic Parsing and Natural Language Generation Using Statistical Machine Translation Techniques*. PhD thesis, Department of Computer Sciences, University of Texas at Austin, 2007.

[10] Yulan He and Steve Young. Spoken language understanding using the hidden vector state model. *Speech Communication*, 48(3-4):262–275, 2006.

[11] Luke S. Zettlemoyer and Michael Collins. Learning context-dependent mappings from sentences to logical form. In *Proceedings of the Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pages 976–984, Singapore, 2009.

[12] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11:2837–2854, 2010.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[14] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 410–420, Prague, Czech Republic, 2007.

[15] Jean Carletta. Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*, 22(2):249–254, 1996.

[16] Koby Crammer, Alex Kulesza, and Mark Dredze. Adaptive regularization of weight vectors. In *Advances in Neural Information Processing Systems 22*, pages 414–422, 2009.

[17] Koby Crammer, Mark Dredze, and Alex Kulesza. Multi-class confidence weighted algorithms. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 496–504. Association for Computational Linguistics, 2009.

[18] John D. Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of 18th International Conference in Machine Learning*, pages 282–289. Morgan Kaufmann Publishers Inc., 2001.

[19] Naoaki Okazaki. CRFsuite: a fast implementation of Conditional Random Fields (CRFs), 2007.

Appendix

Below are the complete lists of the controlled vocabulary terms used in the annotation.

Dialog acts:

```
inform
instruct
prop_question
set_question
ready
processing
greet
acknowledge
yes
no
check_channel
repeat
next_instruct
wait
apologise
```

Predicates with their arguments and constant values:

```
isA(id:X, type:TYPE)
isMany(id:X, type:TYPE)
isNamed(id:X, name:*|X)
hasProperty(id:X, property:PROPERTY, value:*|X)
isVisible(target:X, observer:@)
position(id:X|@, location:X|forward|right|backward|left|at|in|beside, ref:X|@)
distance(location:X|@, location:X|@, value:*|X|near|far)
route(from_location:X|@, to_location:X|@, via_location:X)
turn(agent:@, direction:X|forward|right|backward|left, at_location:X)
walk(agent:@, along_location:X, to_location:X, cross_location:X, via_location:X,
     direction:X|forward|right|backward|left)
def(id:X)
indef(id:X)
index(id:X)
next(id:X)
extraInfo(id:X)
inSet(member:X, set:X)
equivalent(id:X, id:X)
argmin(var:X, value:X)
argmax(var:X, value:X)
```

Entity types:

atm
building
cafe
church_or_religious_centre
financial_institution
gym
health_centre
hill
historic_building
junction
library
museum
park
path
pedestrian_crossing
post_office
pub
restaurant
school_or_university
shop
statue
street
supermarket
theatre_or_concert_hall
traffic_light
walk

Properties:

cuisine_type
price_range
rating
size