ACCURATE STATISTICAL SPOKEN LANGUAGE UNDERSTANDING FROM LIMITED DEVELOPMENT RESOURCES

Ivan V. Meza-Ruiz, Sebastian Riedel, Oliver Lemon

School of Informatics
University of Edinburgh
Edinburgh EH8 9LW, UK
{I.V.Meza-Ruiz,S.R.Riedel}@sms.ed.ac.uk,olemon@inf.ed.ac.uk

ABSTRACT

Robust Spoken Language Understanding (SLU) is a key component of spoken dialogue systems. Recent statistical approaches to this problem require additional resources (e.g. gazetteers, grammars, syntactic treebanks) which are expensive and time-consuming to produce and maintain. However, simple datasets annotated only with slot-values are commonly used in dialogue systems development, and are easy to collect, automatically annotate, and update. We show that it is possible to reach state-of-the-art performance using minimal additional resources, by using Markov Logic Networks (MLNs). We also show that performance can be further improved by exploiting long distance dependencies between slot-values. For example, by representing such features in MLNs, but without using a gazetteer, we outperform the Hidden Vector State (HVS) model of He and Young 2006 (1.26% improvement, a 13% error reduction).

***Index Terms*—** Natural language interfaces, Adaptive systems, Speech processing, Cooperative systems

1. INTRODUCTION

Spoken Language Understanding (SLU) systems produce representations of the meaning of utterances recognised by automatic speech recognition (ASR) modules of spoken dialogue systems. After SLU the dialogue manager module interprets the representation in context and produces a response for interaction with the user. Recently, statistical approaches have been explored for this task [1, 2, 3, 4], rather than more brittle and labour-intensive grammar-based frameworks .

For rapid development of robust dialogue systems it is important that SLU components are:

- accurate and robust,
- easy to build, update, and maintain.

We thank Alex Lascarides and James Henderson for comments. This work is partially funded by EPSRC grant number EP/E019501/1 and the EC FP7 project "CLASSiC" (ICT-216594).

In this paper we show how to meet both of these requirements in a system with state-of-the-art performance. In particular we study scenarios where dialogue system developers need to create SLU components from limited resources (e.g. a slot-value annotated corpus), and compare this with cases where extra information is available. In this case, the meaning of an utterance is presented as a set of slot values, as is commonly used in spoken dialogue systems e.g. [2]. Table 1 presents an example of slot-values as semantic representation[1].

USER:*what flights are there arriving in Chicago on continental airlines after 11 pm*
$GOAL$ =FLIGHT
$TOLOC.CITY\_NAME$ =Chicago
$AIRLINE\_NAME$ =continental_airlines
$ARRIVE\_TIME.TIME\_RELATIVE$ =after
$ARRIVE\_TIME.TIME$ =n2300

Table 1. Example of slot-values as a semantic representation.

The SLU task is then to create a labelling of slot-values for each word of a recognised user utterance. In particular, we explore the use of long distance dependency features for statistical SLU. Typically, statistical SLU approaches produce a labelling based on observable information at a specific point of the utterance and the n previous labels. In the case of $n = 0$ we have a simple classifier. For $n > 0$ we have a linear chain model which uses an n^{th} order Markov assumption. In this work, we also explore the use of more complex relations in order to capture long distance dependencies within a Markov Logic Framework.

2. PREVIOUS WORK

There is renewed interest in statistical SLU given new state of the art techniques in the field. [1] proposed a parsing framework for the SLU task. However, this approach requires a cor-

[1]From the ATIS 3 corpus [5].

pus labelled with semantically augmented syntactic trees. [2] presents the Hidden Vector State model which can be thought as an extended HMM which can handle stacks of labels instead of single labels, but additionally uses a gazetteer. Two very recent state of the art results are presented in [3, 4]. Both approaches tackle the problem as a parsing problem and they learn a weighted grammar which is used to parse utterances. In both cases, a corpus annotated with logical forms is required while syntactic trees are handled as hidden variables. [6] describes a mixed model for SLU where a statistical classifier identifies user intentions and a rule-based grammar detects the named entities (a shallow semantic representation similar to slot-values). However, here a rule-based grammar has to be developed for each new domain, in addition to a labelling of the corpus with intentions.

As can be appreciated in the results of approaches which handle SLU as a parsing task, the inclusion of long distance dependencies has been helpful. However, such approaches require a corpus annotated with full logical forms. Such a corpus is an expensive resource during the development of a dialogue system, and is costly to produce, maintain, and update. In contrast, we constrain our work to use only a slot-value annotated corpus, which is easier to annotate and maintain, since slot-values are less complex structures than logical forms. Slot-value labellings are also widely used in dialogue systems development.

To handle long distance dependencies and capture (to some extent) the advantages of parsing approaches statistically we use densely connected sequential Markov Networks. Markov Logic Networks (MLNs)[7] provide a compact way of defining such networks and allow efficient inference and training. Here first order logic (FOL) formulae with associated weights are used as templates for loglinear models. Using Markov Logic as modelling framework not only allows us to incorporate a large class of global dependencies into our models in order to improve accuracy, it also ensures that the underlying technology is widely accessible and can be easily reused in different contexts. Freely available toolkits such as [8, 9] provide efficient means of training and inference in such models.

3. THE CORPUS

For our experiments we use the Air Travel Information System (ATIS, [5]) corpora. These corpora are in the domain of flight booking and car rental. In particular, we used the extended version created by [2]. This version is composed of slot-value labellings of the ATIS-2 and ATIS-3 training sets (4978 utterances), and the ATIS-3 $NOV93$ testing set (448 utterances). To select features, perform error analysis, and decide the number of iterations in the implementation of the MLNs we split the corpus in to training and test sets (4582/396 utterances).

We have also used the OVIS [1] corpus with similar results to those presented here.

4. THE MLN MODELS

We approach the SLU task using two models. One for the goal slot which depends on the whole utterance, and the other for the argument slots (i.e., slots which have a word of the utterance as argument). Figure 1 presents the Graphical Model for the goal. There are 22 possible labels for the goal (e.g., `FLIGHT`, `GROUND_SERVICE`, `AIRFARE`). For the case of the argument slot we treat the slot as single label. Figure 2 presents the Graphical Model for the slot arguments model. There are 112 possible labels for the slot arguments.

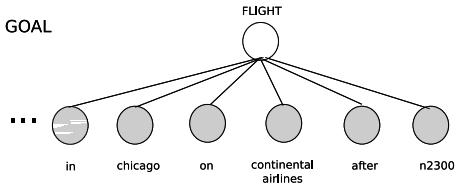

Fig. 1. Example using Local Features

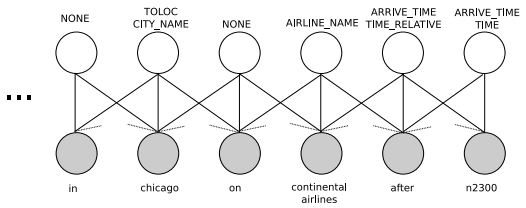

Fig. 2. Model for slots as a single label

4.1. Local Features

So far, the models of figures 1 and 2 employ local features. In these figures the hidden variables are connected only to observable variables. In particular, we can define features based on the current word and its context, for instance: Orthography, membership of a type (e.g., gazetteers, numbers), or extra information (e.g., Part-of-Speech tags).

In the approach presented here, we show how to perform accurate SLU without using features requiring additional development effort, such as a gazetteer, or POS tagging. We use only standard orthography and properties of a slot-value-labelled corpus.

4.2. Adding Markov assumptions

Figure 3 shows the model with the inclusion of first and second order Markov assumptions for the argument slots, which only involves hidden variables. In this case, a feature is defined using only hidden variables, and these features are straightforward to define using the MLN.

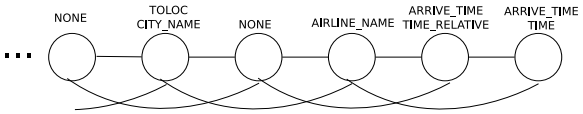

Fig. 3. 1^{st} and 2^{nd} order Markov assumptions

Figure 4 shows the definition of these relations using a FOL formulae. For the case of the 1^{st} order Markov assumption the formula says that there is a relation for every pair of argument slots which are separated by one position. This is useful for example in capturing a pattern of slots about time, which often appear together. This can be seen it the example of table 1.

$$\forall Slot\ s, Slot\ s_{prev}, Position\ p. slot\_argument(p, s) \wedge slot\_argument(p-1, s_{prev})$$

Fig. 4. Rule for 1^{st} order Markov assumption

4.3. Global Features

To capture long distance dependencies between the slots we use global features. Figure 5 shows some of the relations we can define. Figure 5.A shows a relation between two consecutive labels and a previous label, and 5.B shows a relation between a label and any other previous label which is not otherwise considered by the Markov assumptions of section 4.2. The first relation corresponds to a common pattern where two consecutive argument slots depend on their context. In the case of our example, the fact that the time slots are `ARRIVE` is linked to the type of slot assigned to *Chicago* (i.e., `TOLOC.city_name`). The second relation links the argument slots with the rest of the slots, in this case we look to avoid cases where an argument slot such as `TOLOC.city_name` appears more than once in a sentence.

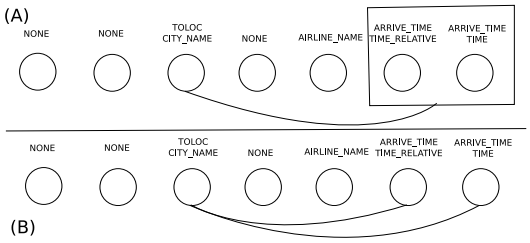

Fig. 5. Examples of global features

4.4. Training and Inference

To learn the weights of the MLN we use single-best MIRA[10], a discriminative Online Learner. For Maximum A Posteriori Inference at test time and during Online Learning we employ a Cutting Plane Method that incrementally instantiates and solves Integer Linear Programs representing the Markov Network[11]. For a large class of MLNs (in particular the ones we train in this work) this yields exact and efficient inference.

5. EXPERIMENTS: METHOD AND RESULTS

We developed two sets of experiments. The first one aims to establish the empirical superiority of a local MLN model over a MaxEnt model, thus showing that a local MLN model reaches suitable baseline performance. We choose MaxEnt as a baseline because a straight-forward implementation of the task provides comparable results to those reported by [2]. This was trained using minimal resources and two classifiers, one for the goal and one for the argument slot. Features for each utterance help to identify the goal, while features from a word and its context help to identify the argument slots. A similar set-up was tried with implementations of Conditional Random Fields but the training times were from 3 to 5 days making experimentation with different features impractical. For the second set of experiments we increase the complexity of the model. In both sets of experiments we constrained the models to use minimum resources as discussed above.

For both models in the first set of experiments, we use the same local features: Orthography of the current word, the two previous words and the two following words. We also use a feature indicating the presence of the words: *arrive, arriving, leave* and *leaving*. The MLN_{local} model corresponds to a local model where the argument slots were considered as a unique label. This is directly comparable with the MaxEnt model.

For the second set of experiments we create the model MLN_{global} by adding first and second Markov order assumptions and the global features shown in 5.B to the local model.

To evaluate the models we use two measurements: *Global* and *Exact* match scores as presented in [2, 3]. The global scores measure precision, recall and $F1$-score for recovering slot-values in the whole experiment. The exact match score measures precision, recall and $F1$-scores for recovering the exact set of slot-values for each utterance.

5.1. Results: MaxEnt and MLN

For our first set of experiments we obtain the results presented in table 2. Here, MLN_{local} performs better than the MaxEnt model. Statistical significance of MLN_{local} against the MaxEnt model for both precision and recall is at $\rho < 0.05$.

5.2. Results: Global MLN vs. Hidden Vector State model

For the second set of experiments we obtained the results presented in table 3. In this table we include the results presented in [2] under the label HVS, although the techniques are not

		Precision	Recall	$F1$-score
MaxEnt	Global	89.45%	88.82%	89.13%
	Exact	66.21%	64.95%	65.57%
MLN_{local}	Global	91.46%	91.30%	91.38%
	Exact	72.64%	70.54%	71.57%

Table 2. Baselines: MaxEnt versus local MLN

directly comparable since the HVS model uses additional information in a gazetteer as a preprocessing step. We can see that both the MLN_{local} and MLN_{global} scores outperform the HVS model, based on only a slot-value annotated corpus. Note that Exact match and Precision/Recall scores are not available for the HVS system. We also measured the statistical significance of MLN_{global} compared with MLN_{local} and for both precision and recall we have $\rho < 0.05$.

		Precision	Recall	$F1$-score
MLN_{global}	Global	93.43%	89.77%	91.56%
	Exact	72.04%	67.86%	69.89%
HVS	Global	N/A	N/A	90.3%

Table 3. Global MLN vs. HVS model (He and Young 2006)

This shows that the extra relations defined in the global MLN provide increased performance whilst also using fewer development resources than HVS [2]. We note that the global precision of the global MLN is higher than that of the local model, although the exact match scores are slightly worse. This is due to data sparsity. Global features which were not seen in the corpus will be penalized. This strategy produces better argument slots (high precision), at the cost of detecting some slot sequences (low recall).

6. SUMMARY AND DISCUSSION

Recent statistical approaches to SLU require external resources (e.g. gazetteers, grammars, syntactic treebanks) which make rapid development and ongoing maintenance of SLU components costly [1, 2, 3, 4]. We present a new method which proceeds from simple corpora annotated only with slot values. Such datasets are commonly used in dialogue systems development and are easy to produce and update.

We used the ATIS 3 corpus [5] in this paper, for comparison with [2] (but have also used the OVIS corpus [1] with similar results). We show that it is possible to reach state-of-the-art performance without using any other extra resources, by using Markov Logic Networks (MLNs). We outperform the Hidden Vector State model of [2] which additionally used a gazetteer. Furthermore we show that using global features (with their ability to capture long-distance dependencies) in statistical learning approaches is a promising method for future advances in SLU. By using them we obtain a 1.26% improvement in performance when compared to the HVS model.

In ongoing work we are are also investigating how the addition of extra information (such as gazetteers and syntactic information) improves the MLN models. We are exploring new configurations for MLN models for this task. Ultimately, such work will improve the state-of-the-art in statistical parsing, from limited and practical development resources, for development of robust spoken dialogue systems.

7. REFERENCES

[1] Rens Bod, "Context-sensitive dialogue processing with the DOP model," *Natural Language Engineering*, vol. 5, no. 4, pp. 310–323, 2000.

[2] Yulan He and Steve Young, "Spoken language understanding using the Hidden Vector State model," *Speech Communication*, vol. 48(3-4), pp. 262–275, 2006.

[3] Luke Zettlemoyer and Michael Collins, "Online learning of relaxed CCG grammars for parsing to logical form," in *Proceedings of EMNLP-CoNLL*, 2007.

[4] Yuk Wah Wong and Raymond J. Mooney, "Learning synchronous grammars for semantic parsing with lambda calculus," in *Proceedings of the 45th Annual Meeting of the ACL*, 2007, pp. 960–967.

[5] Deborah A. Dahl and et al., "ATIS 3 training data," 1994, Linguistic Data Consortium, Philadelphia.

[6] N. Gupta, G. Tur, D. Hakkani-Tur, S. Bangalore, G. Riccardi, and M. Rahim, "The AT&T spoken language understanding system," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, pp. 213–222, 2006.

[7] Matt Richardson and Pedro Domingos, "Markov Logic Networks," *Machine Learning*, vol. 62, pp. 107–136, 2007.

[8] Stanley Kok, Parag Singla, Matthew Richardson, and Pedro Domingos, "The Alchemy System for Statistical Relational AI," Tech. Rep., Department of Computer Science and Engineering, University of Washington, 2005.

[9] Sebastian Riedel, "Markov TheBeast: Pseudo Markov Logic Engine," 2007, http://code.google.com/p/thebeast/.

[10] Koby Crammer and Yoram Singer, "Ultraconservative online algorithms for multiclass problems," in *Proceedings of 14th COLT and EuroCOLT 2001, Amsterdam*, 2001, vol. 2111, pp. 99–115.

[11] Sebastian Riedel and James Clarke, "Incremental integer linear programming for non-projective dependency parsing," in *Proceedings of EMNLP*, 2006.