A Factored Discriminative Spoken Language Understanding for Spoken Dialogue Systems

Filip Jurčíček, Ondřej Dušek, and Ondřej Plátek

Charles University in Prague
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
{jurcicek, odusek, oplatek}@ufal.mff.cuni.cz
https://ufal.mff.cuni.cz

Abstract. This paper describes a factored discriminative spoken language understanding method suitable for real-time parsing of recognised speech. It is based on a set of logistic regression classifiers, which are used to map input utterances into dialogue acts. The proposed method is evaluated on a corpus of spoken utterances from the Public Transport Information (PTI) domain. In PTI, users can interact with a dialogue system on the phone to find intra- and inter-city public transport connections and ask for weather forecast in a desired city. The results show that in adverse speech recognition conditions, the statistical parser yields significantly better results compared to the baseline well-tuned handcrafted parser.

Keywords: spoken language understanding, dialogue systems, meaning representation

1 Introduction

Semantic parsing is a key component of any spoken dialogue system. Its purpose is to map natural language to a formal meaning representation – semantics, which can be defined either by a grammar, e.g. LR grammar for the GeoQuery domain [1], or by frames and slots, e.g. the TownInfo domain [2], or dialogue acts [3]. In this work, dialogue acts are used to represent the meaning. A dialogue act (DA) is composed of one or more dialogue act items (DAI). A dialogue act item represents basic intents (such as `inform`, `request`, etc.) and optionally the semantic content, also referred to as slots, in the input utterance (e.g. `vehicle=bus`, `time=1:30`). In some cases, the value of a slot can be omitted, for example, where the intention is to query it, as in `request(arrival_time)`. Table 1 shows examples of the dialogue acts in the public transport information (PTI) domain. As dialogue managers commonly use semantics in the form of frames and slots [3,4], the presented approach learns to map directly from natural language into the frame and slot semantics.

This paper describes a probabilistic discriminative Spoken Language Understanding (SLU) based on Dialogue Act Item Classifiers using Logistic Regression (DAICLR), where logistic regression classifiers are used to map input utterances into dialogue acts. To obtain a compact probabilistic representation, the predicted dialogue acts are factored according to dialogue act items, each associated with a corresponding item

P. Sojka et al. (Eds.): TSD 2014, LNAI 8655, pp. 579–586, 2014.

Table 1. Examples of the PTI semantics.

1.	ZE STANICE NÁDRAŽÍ HOLEŠOVICE *from the NÁDRAŽÍ HOLEŠOVICE station* `inform(from_stop="Nádraží Holešovice")`
2.	JÁ BYCH CHTĚLA JET V OSM HODIN RÁNO *I would like to leave at 8 am* `inform(ampm="morning")&inform(departure_time="8:00")`

3.	JEDE TO NA ZVONAŘKU
	`is it going to ZVONAŘKA`
	`confirm(to_stop="Zvonařka")`

4.	V KOLIK HODIN TO BUDE NA STANICI VELETRŽNÍ PALÁC *what time does it arrive to the VELETRŽNÍ PALÁC stop* `inform(to_stop="Veletržní palác")&request(arrival_time)`

Top to bottom for each statement: Czech user utterance, English literal translation, dialogue act.

marginal probability instead of representing the uncertainty of dialogue acts in N-best lists. The proposed DAICLR method is evaluated on a corpus of spoken utterances from the Public Transport Information (PTI) domain. In PTI, users can interact in Czech language with a telephone-based dialogue system to find intra- and inter-city public transport connections and ask for weather forecast in a desired city. The PTI system is publicly available at a toll-free phone number and covers virtually all cities in the Czech Republic [5].

A successful SLU component in a spoken dialogue system must be robust to recognition errors, easy to build, computationally efficient, and provide accurate predictive probabilities. As the DAICLR parser is directly trained on the output of the ASR component which includes recognition errors, the DAICLR is capable of learning to deal with systematic recognition errors. The parser learns from data which has no alignment between words and semantics. It can efficiently parse not only 1-best ASR output but also N-best lists with ASR hypotheses. In addition, it learns a small set of classifiers that allows real-time parsing. Finally, the output probability estimates accurately model the chance of the predicted items being correct due to the use of discriminatively trained logistic regression classifiers. This is particularly important in probabilistic dialogue state tracking components [6].

In the next section, related work on mapping natural language into formal meaning representations is described. Section 3 presents the proposed DAICLR parser and describes the training process. Section 4 compares the proposed DAICLR parser to a well-tuned handcrafted parser developed for the PTI domain. Finally, Section 5 concludes this work.

2 Related Work

There has already been a substantial amount of work on data-driven approaches to SLU. This section briefly describes some of the main contributions to this in literature.

The Hidden Vector State (HVS) technique has been used to model an approximation of a pushdown automaton with semantic concepts as non-terminal symbols [7,8]. From the output parse trees, a deterministic algorithm was used to recover slot names and their values.

A probabilistic parser using Combinatory Categorical Grammar (PCCG) has been used to map utterances to lambda calculus [10]. The combinatory categorical grammar is converted into a probabilistic model by learning a log-linear model. An online learning algorithm updates weights of features representing the parse tree of an input utterance. However, apart from using the lexical categories (city names, airport names, etc.) readily available from the ATIS corpus [11], this method also needs a considerable number of handcrafted entries in its initial lexicon.

Markov Logic Networks (MLN) have been used to extract slot values by combining probabilistic graphical models and first-order logic [12]. In this approach, weights are attached to first-order clauses which represent the relationship between slot names and their values. Such weighted clauses are used as templates for features of Markov networks.

Semantic Tuple Classifiers (STC) based on support vector machines have been used to build semantic trees by recursively calling classifiers that predict fragments of the semantic representation from N-gram features [2].

The domain-independent semantic role labelling was used as a form of preprocessing to reduce complexity of mapping to domain-dependent meaning representations [14].

Machine translation techniques [15] have been used with a translation model based on synchronous context-free grammars.

Inductive logic programming [16] has been used to incrementally develop a theory including a set of predicates. In each iteration, the predicates were generalised from predicates in the theory and predicates automatically constructed from examples.

The DAICLR approach is similar to the STC parser; however, its implementation is more straightforward and consequently more computationally efficient.

3 Methodology

This section describes the DAICLR parsers. First, a description of an utterance and dialogue act abstraction using the in-domain gazetteers is provided. Second, the method of training dialogue act item classifiers is described. Third, features used in the dialogue act classifiers are detailed.

3.1 Utterance and dialogue act abstraction

The DAICLR model uses a set of independent classifiers for dialogue act items that can appear in the output dialogue act. Since the number of possible slot values for each slot is generally very high, this would lead to a very large set of classifiers specialised to classify individual combinations of dialogue act type, slot name, and slot value in the input utterance. Consequently, the training process would suffer from severe data sparsity since most of the slot values are never seen in training data. Therefore, to reduce the data sparsity, a form generalisation using gazetteers with surface forms for

slot categories such as city and stop is implemented [7,10,12,13]. A simple deterministic procedure is used to abstract an utterance and associated dialogue act. This procedure has two variants: the first one is used in training, the second one in decoding. In training, surface forms of slot values found in the dialogue act are replaced by their category labels in both the utterance and the dialogue act. This is demonstrated in the next example:

chtěla bych jet z Anděla ⇒ `inform(from_stop="Anděl")`
i want to leave from Anděl (Eng. lit. tran.)

is abstracted to

chtěla bych jet z STOP ⇒ `inform(from_stop="STOP")`
i want to leave from STOP (Eng. lit. tran.)

Then an abstract classifier for `inform(from_stop="STOP")` is trained. However, gazetteers are not always accurate and do not include all possible surface forms. Therefore, when no surface forms for a given slot value can be found in the utterance, the slot value is left un-abstracted and a specialised classifier just for this specific slot value is trained. To prevent creation of too many specialised classifiers, only classifiers for values with occurence counts larger than a certain threshold are created.

In decoding, the utterance is first abstracted by replacing the surface forms of slot values by their category labels. Each substitution of a surface form is recorded together with the corresponding slot value. Then, dialogue act item classification is performed using both abstract and specialised classifiers. Finally, outputs predicted by abstract classifiers are converted back, replacing the category labels with the corresponding slot values from the substitution records. A similar approach to generalisation was used in [2].

3.2 DAICLR model

The DAICLR dialogue act predictive model is factored according to dialogue act items as follows:

$$p(d|u) = \prod_i p(i|u), \tag{1}$$

where d is a dialogue act, u is the input utterance, i is the i-th item of the dialogue act d, and each probability $p(i|w)$ is modelled by a logistic regression classifier. Let the feature function $f(i,u) \in R^d$, defined as a d-dimensional vector, represent features extracted from the input utterance u for the dialogue act item i. Let be the $w \in R^d$ a parameter vector. Then the probability of a dialogue act item i is defined as

$$p(i|u) = \frac{1}{1 + \exp\{-w_i \cdot f(i,u))\}}. \tag{2}$$

In this work, the logistic regression classifier training was performed using the Scikit-Learn [18][1] software package. However, any other tool could be used. Other types of classifiers such as support vector machines and kernelised logistic regression [9] were

[1] `http://scikit-learn.org/stable/`

evaluated as well. As these approaches provided similar performance on the evaluation task, their evaluation is omitted in this work. The source code of the DAICLR parser is available on GitHub[2].

In the next Section, the features used in the classifiers are detailed.

3.3 Feature Extraction

To make the DAICLR parser computationally efficient, only lexical N-gram features extracted from the input utterance are used. In informal experiments, it was observed that N-grams for N up to 4 bring consistent improvement in accuracy. In addition, skipping bigrams, which can skip up to 3 words, were used. The skipping bigrams have a large span, yet they do not suffer from data sparsity as much as high-order N-grams. To prevent overfitting, simple feature pruning based on counts of occurrences in the training data was introduced. The threshold was set to the size of the N-gram plus a fixed small constant, which was tuned on development data. While abstraction was used to generate abstract features (see Section 3.1), the final feature set also includes examples of original surface forms. This enables classifiers to adjust its classification to common values, such as Prague (the capital city of the Czech Republic) or Anděl (a stop with high public transport traffic). For example, the features extracted for the sentence "chtěla bych jet z Anděla" include N-grams 'chtěla', 'z Anděla', 'chtěla * Anděla', 'z STOP', 'chtěla * STOP', and so on.

So far, extraction from text or 1-best ASR hypothesis was described. To process ASR N-best lists, the same feature extraction process is performed for each hypothesis in the list. The final feature set is then a weighted combination of features extracted from individual ASR hypotheses where the weights correspond to the probabilities of the hypotheses.

4 Experiments

In this section, the DAICLR parser is evaluated on the corpus of user interactions with a statistical spoken dialogue system in the PTI domain [5].

4.1 Data

The PTI corpus consist of approximately 1800 dialogue call logs, which amount to about 11870 user utterances. All audio recordings in the data were transcribed by professional transcribers. The transcriptions are orthographic and capture several kinds of non-speech events as well as incompletely pronounced words and foreign words used in Czech discourse. To obtain semantic annotation, a semi-automatic transcription process was employed. A handcrafted parser was built by an expert in an iterative manner: The parser was first used to obtain semantic annotations by processing human transcriptions, these automatic annotations were then verified by an independent expert and identified errors were corrected in the handcrafted parser.

[2] `https://github.com/UFAL-DSG/alex/blob/master/alex/components/slu/dailrclassifier.py`

This approach seems to be appropriate since most of advantages of a trainable SLU come from the ability to adapt to ASR errors. In addition, obtaining semantic annotation for Czech data is relatively slow and complicated; using crowdsourcing is not a possibility due to lack of speakers of Czech on platforms such as Amazon Mechanical Turk[3].

The data were divided into training, development, and test sections, where the corresponding data sizes were 9,496, 1,188, 1,188 utterances respectively. Apart from manual transcriptions, the data includes the 1-best and 10-best lists of ASR hypotheses, which allows us to evaluate the robustness of our models to recognition errors. Any tuneable parameters such as pruning thresholds were set on the development data and the reported results were obtained using the test set.

To obtain ASR hypotheses, we used the Google cloud-based ASR[4]. The main advantage of Google ASR is that it is fast, can be used off-the-shelf without any additional modifications, and provides state-of-the-art quality for many tasks [17]. This setup yields performance of 45.2% WER (Word Error Rate) on our data. The high WER is presumably caused by adverse acoustic conditions presented in the PTI speech data, e.g. street noise, and mismatch between the Google's language model and the PTI domain.

Table 2. Dialogue act item precision, recall and F-measure for the PTI test set.

Parser	Precision	Recall	F-measure
1-best ASR output			
handcrafted parser	50.83	47.39	49.12
DAICLR	68.39	67.52	67.95
N-best ASR output			
handcrafted parser	50.89	47.49	49.13
DAICLR	68.59	67.74	68.16

4.2 Results

The results for the PTI test data are shown in Table 2. The model accuracy is measured in terms of precision, recall, and F-measure (harmonic mean of precision and recall) on dialogue act items. A dialogue act item is correct only if the dialogue act type, slot, and value are correct.

The DAICLR parser significantly outperforms the baseline handcrafted parser. Regarding the 1-best ASR output, Table 2 shows that DAICLR produces 67.95% of F-measure, which represents a 18.83% improvement over the handcrafted parser.

[3] `https://www.mturk.com/mturk/welcome`

[4] The API is located at `https://www.google.com/speech-api/v1/recognize` for the Czech version of the service, and its use is described in a blog post at `http://mikepultz.com/2013/07/google-speech-api-full-duplex-php-version/`.

Concerning the N-best ASR output, results shows only modest improvement over the 1-best results. Presumably, the probabilities in the N-best ASR output do not reflect the accuracy of the ASR hypotheses well. On manual inspection of the data, we observed that only a small portion of the total probability mass was distributed among alternative hypotheses on average.

The DAICLR parser is very computationally efficient on domains such as PTI because the final number of abstracted and specialised classifiers is small. There are 24 unique dialogue act types and 21 unique slots in the PTI domain and the total number of learnt classifiers was 135. On a standard workstation, the DAICLR parser can process one utterance in under 50 milliseconds.

5 Conclusion

This paper presents a novel spoken language understanding parser based on a factored probabilistic model with individual factors modelled using logistic regression classifiers. This approach learns a small set of abstract and specialised classifiers which generalise or specialise to any slot value in a domain. The concept of a factored model for dialogue act item classification and a small set of classifiers is the core of the parser's computational efficiency. It was verified that in adverse acoustic conditions, the trainable DIACLR parser significantly outperforms a well-tuned handcrafted parser and can significantly mitigate the problem of inaccurate speech recognition.

Acknowledgments. This research was partly funded by the Ministry of Education, Youth and Sports of the Czech Republic under the grant agreement LK11221, core research funding and grant GAUK 2058214 of Charles University in Prague. This work has been using language resources distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2010013).

References

1. Kate, R.J., Wong, Y.W., Mooney, R.J.: Learning to Transform Natural to Formal Languages In: Proceedings of AAAI, pp. 1062–1068 (2005)
2. Mairesse, F., Gasic, M., Jurčíček, F., Keizer, S., Thomson, B., Yu, K., Young, S.: Spoken language understanding from unaligned data using discriminative classification models. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4749–4752 (2009)
3. Thomson, B., Gašić, M., Keizer, S., Mairesse, F., Schatzmann, J., Yu, K., Young, S.: User study of the Bayesian update of dialogue state approach to dialogue management In: Proceedings of Interspeech, pp. 483–486 (2008)
4. Williams, J., Young, S.: Partially observable Markov decision processes for spoken dialog systems, In: Computer Speech and Language, 21-2, pp. 393–422 (2007)
5. Public Transport Information System for Czech Republic, `https://ufal.mff.cuni.cz/alex-dialogue-systems-framework/ptics` (2014)
6. Žilka, L., Marek, D., Korvas, M., Jurčíček, F.: Comparison of Bayesian Discriminative and Generative Models for Dialogue State Tracking, In: SIGDIAL '13: Proc. of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Metz, France, pp. 452–457 (2013).

7. He, Y., Young, S.: Semantic processing using the Hidden Vector State model, In: Computer Speech & Language, 19-1, pp. 85–106 (2005)
8. Jurčíček, F., Švec, J, Müller, L: Extension of the HVS semantic parser by allowing left-right branching, In: Proceedings of ICASSP, pp. 4993–4996 (2008)
9. Zhu, J., Hastie, T.: Kernel logistic regression and the import vector machine. Journal of Computational and Graphical Statistics, 14(1), (2005) 185–109
10. Zettlemoyer, L. S., Collins, M.: Online learning of relaxed CCG grammars for parsing to logical form. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 678–687 (2007)
11. Dahl, D.A., Bates, M., Brown, M., Fisher, W., Hunicke-Smith, K., Pallett, D., Pao, C., Rudnicky, A., Shriberg, E.: Expanding the scope of the ATIS task: The ATIS-3 corpus, In: Proceedings of the ARPA HLT Workshop, pp. 43–48 (1994)
12. Meza-Ruiz, I.V., Riedel, S., Lemon, O.: Spoken Language Understanding in dialogue systems, using a 2-layer Markov Logic Network: Improving semantic accuracy, Proceedings of Londial (2008)
13. Tür, G., Hakkani-Tür, D. Z., Hillard, D., Celikyilmaz, A.: Unsupervised Spoken Language Understanding: Exploiting Query Click Logs for Slot Filling. In: Proceedings of Interspeech, pp. 1293–1296 (2011)
14. Henderson, J.: Semantic Decoder which Exploits Syntactic-Semantic Parsing, for the Town-Info Task, In: CLASSiC Project Deliverable 2.2 (2009)
15. Wong,Y.W., Mooney, R.J.: Learning for Semantic Parsing with Statistical Machine Translation, In: Proceedings of HLT/NAACL, pp. 439–446 (2006)
16. Tang, L.R., Mooney,R.J.: Using Multiple Clause Constructors in Inductive Logic Programming for Semantic Parsing, In: Proceedings of ECML, pp. 466–477 (2001)
17. Morbini, F., Audhkhasi, K., Sagae, K., Arstein, R., Can, D., Georgiou, P. G., Narayanan, S. S., Leuski, A., Traum, D.: Which ASR should I choose for my dialogue system?, Proc. of SIGDIAL, Metz, France, pp. 394–403 (2013)
18. Pedregosa, F. et al.: Scikit-learn: Machine Learning in Python. JMLR 12, pp. 2825–2830, 2011.