Probabilistic matching for dialog state tracking with limited training data

Julien Perez and Will Radford

Abstract This report details our submission to the fourth Dialog State Tracking Challenge (DSTC4), the first time Xerox has participated. Accordingly, we have taken a segment-specific approach that attempts to identify ontology values as precisely as possible using a statistical model. Our model is inspired by work in Named Entity Linking that extracts mentions, then searches and reranks candidates. This is mainly motivated by the small amount of data available relative to the high complexity of the task. However, we believe this setting is realistic in the industrial environment where few data are generally available for a given dialog context to automate. This relatively simple approach performs reasonably at 38.5% F1 using schedule 2 evaluation, and is the most precise at 59.4% on the DSTC4 test set.

1 Introduction

Dialog systems are a rapidly growing research area driven by the spread of smart mobile devices. One prominent challenge is to track the so-called state of the dialog, rather than conduct an interaction. A tracker should provide a compact representation of user and system actions and responses in a *dialog state*. However, errors from Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) make the true user utterance not directly observable, and so deducing the true dialog state is challenging. A common interpretation of the task uses schemata [WY07], that must be filled with a predefined set of slots and values. We considered several approaches such as rule-based [Com07] and Partially Observable Markov Decision Process (POMDP)[TY10a, TY10b, GY11, YGTW13] models, but settled

Julien Perez
Xerox Research Centre Europe, e-mail: julien.perez@xrce.xerox.com

Will Radford
Xerox Research Centre Europe, e-mail: will.radford@xrce.xerox.com

on one inspired by Named Entity Linking [DMR+10, SSB12]. We learn a fairly simple statistical model, in a process that can also be viewed as metric learning. This approach allows our system to be performed reasonably well despite limited access to training data. In our case, we define the condition of "limited data" when a strict a posterior estimation of the value of a given slot can not be computed by classic supervised machine learning technique where values would be used as target value to estimate due to the lack of data.

2 Architecture

Our system sequentially updates its belief of the dialog state using information extracted from utterances. This process has several steps: (1) mention detection, (2) searching for candidate ontology slot-value pairs for mentions and (3) ranking of candidates and classification of INFO values. Once, ranking is computed, the scores associated to the value beyond a given threshold γ. In addition, inspired by the DSTC4 baseline system, we prune inconsistent values from the state, preferring NEIGHBOURHOOD to PLACE values in the ATTRACTION topic, and TO or FROM values to STATION in TRANSPORTATION. We describe each of these in the sections below.

2.1 Detecting mentions

The purpose of this first step is to extract candidate mentions in the current utterance of an on-going dialog that will be used has basis for value matching in an available ontology. We make the assumption that most slot values (not including INFO slots) will be explicitly mentioned in the dialog, usually as a noun phrase. We preprocess the text, tokenizing using NLTK[1] and normalize disfluency markers (e.g. um, %UH), removing markers such as ~. As such, we use a statistical model, SENNA[2], to identify NP chunks, which we term as *mentions* in the rest of this report.

2.2 Searching for ontology candidates

We search a full-text index of ontology values for candidate matches for each mention. The index is built using WHOOSH[3] and indexes each value using an English language analyzer, that includes stemming, and a character n-gram analyzer (*n* rang-

[1] http://www.nltk.org

[2] http://ml.nec-labs.com/senna

[3] https://bitbucket.org/mchaput/whoosh

Name	Description	Example of feature values
ir_score	The full-text index score.	ir_score=89.50
ir_score_pct	The full-text index score divided by the *highest* scoring candidate.	ir_score=0.67
ir_rank	The rank of this candidate in the full-text index results.	ir_rank=4
ir_size	The number of full-text index results.	ir_size=15
ratio	The edit-distance using FUZZY between the mention and value (case-insensitive).	ratio=45
prt_ratio	The partial edit-distance using FUZZY between the mention and value (case-insensitive).	prt_ratio=100
stem_ratio	The same as ratio, but using Porter-stemmed tokens.	stem_ratio=47
stem_prt_ratio	The same as partial_ratio, but using Porter-stemmed tokens.	stem_prt_ratio=100
slot	The slot of the candidate.	slot-FROM=1
slot_mention	The slot and mention tokens of the candidate.	slot_mention:FROM:airport=1
slot_context	Bag-of-words from the utterance, not including the mention, joined with the slot.	slot_context:FROM:And=1
left_context	Bag-of-words from the three tokens (padded with ###) before the mention, joined with the slot.	left_context:FROM:from=1
right_context	Bag-of-words from the three tokens (padded with ###) after the mention, joined with the slot.	right_context:FROM:###=1
cos_emb	The cosine similarity between phrase embeddings of the mention and value. Phrase embeddings are the average of token WORD2VEC embeddings.	cos_emb=0.72

Table 1 Features used by the ranking model for the mention airport and ontology value (TRANSPORTATION, FROM, SINGAPORE CHANGI AIRPORT).

ing from 2 to 6) for distant matching. We limit searches at 30 candidates and build a ranked list of matches with their scores. Each candidate is a tuple of (TOPIC, SLOT, VALUE), although the topic is given and so the same for each candidate.

2.3 Ranking slot values

Our model uses features that aim at encoding the match between an extracted mention and the candidate value. The candidate value are taken at the segment level

that is the only information available for the state tracking challenge. We learn a logistic regression classifier using `scikit-learn`[4]. Table 1 summarises the features, which use external tools such as NLTK for stemming FUZZY[5] for string edit-distances and WORD2VEC for word embeddings.[6]. More formally, the model aims at estimating the probability $p(v|m) = \frac{1}{1+e^{-w^T \phi(v,m)}}$ of a $\{$slot, value$\}$ pair v given a mention m, with w the model's parameters to learn and $\phi(v,m)$ the feature functions presented in the previous section. At learning stage, a mention's candidates are assigned to *1* if it is present in the set of gold-standard tuples for the segment. All other candidates are assigned the value *0*, but note that a list of candidates may include multiple *1* instances depending on the search. We also include NIL candidates to model the lack of a matching candidate, one for each of the slots that were retrieved in the candidates. Where we found a *1* instance, the NIL candidate is labelled *0*, otherwise *1*. These have three features: `NIL_topic`, `NIL_topic_slot` and `NIL_slot`. During initial experimentation, we learn the model using 10-fold cross validation over `dstc4_train` with a grid search to choose the optimal hyperparameters. Such negative sampling strategies tend to warp the distribution of targetted values but we found them to be the most efficient approach in cross validation. There were 190,055 instances, 15% of them *true*, and the best model performed with mean F1 of 89.3% using $l2$ regularization ($C = 1$). During tracking, we apply the same procedure for search and feature extraction, then predict the probability of each candidate using the model. For each slot, we consider the three most probable candidates from the computed list. However, the presence of the NIL candidate in the top-3 list acts as a threshold. Indeed, NIL is used as a special value that represents the fact that no value has been assigned to a given variable

2.4 Classifying INFO *values*

A logistic regression model is used to model the likelihood of a value w.r.t the INFO variable which is present for each topic. The decision is supported by n-grams (1, 2 and 3) of raw and stemmed tokens using the pre-processing and classifiers above and one model has been produced for each topic. The model is $l1$ regularized, with hyper-parameters optimized using 5-fold cross-validation on the training set. We learn independent models for each topic-space and these have varying performance FOOD (78.6% F1), TRANSPORTATION (75.3% F1) and ACCOMMODATION (71.9% F1) perform reasonably, but we see worse performance on ATTRACTION (66.0% F1) and ACCOMMODATION (52.9% F1). We use all segment utterances so far for training and prediction and retain the top value.

[4] http://scikit-learn.org

[5] https://github.com/seatgeek/fuzzywuzzy

[6] Google News embeddings from https://code.google.com/p/word2vec

Topic	Schedule 1			Schedule 2		
	P	R	F	P	R	F
ACCOMMODATION	45.0	22.1	29.6	54.5	29.4	38.2
ATTRACTION	57.3	25.2	35.0	57.8	31.6	40.9
FOOD	66.4	23.8	35.0	64.0	27.8	38.8
SHOPPING	23.9	10.6	14.7	38.6	23.6	29.3
TRANSPORTATION	53.3	22.9	32.0	54.2	25.0	34.2
all	52.7	22.8	31.8	55.6	28.8	38.0

Table 2 Results on `dstc4_dev`.

System	Schedule 1			Schedule 2		
	P	R	F	P	R	F
Top system	53.0 (3)	**50.3** (1)	**51.6** (1)	54.4 (3)	**58.7** (1)	**56.5** (1)
Our system	**56.2** (1)	23.1 (5)	32.8 (5)	**59.4** (1)	28.5 (5)	38.5 (4)

Table 3 Results on `dstc4_test` for topic/slot all/all. Ranks are shown in parentheses. Highest values are in **bold**.

3 Experimental results

Table 2 shows the results of our system trained on `dstc4_train` and evaluated on `dstc4_dev`. Our overall performance is 38% F1 on schedule 2 and shows high precision at the cost of recall, a pattern that reflected in schedule 1 results. The results per-topic are mostly distributed around 38% F1, except for worse performance on shopping and transportation. Within topics, the results are more variable, with 0% F1 on some topic/slot combinations such as food/drink.

We retrained our models on `dstc4_train` and `dstc4_dev` for evaluation on `dstc4_test`. Our system placed 4th ranked by F1 on schedule 2 with a score of 38.5. Table 3 shows overall scores and those of the top-performing system. Notably, although our system's performance is bounded by low recall, it has the highest precision of all systems in the competition.

4 Future work and Conclusion

Our system could be extended in a few different ways that address its current limitations. The foremost of these is to consider state history. This is currently *accretive*, so we add information at each utterance, resetting at segment boundaries. This approach is effective, as it limits the damage of wrong decisions, but many utterances refer to previously identified slot values, so handling them effectively is important. The other main issue is lack of large scale training data, which manifests in two

ways: many values are not encountered during training, and there are few individual sessions, as each tends to be long. Overall, our system performs surprisingly well, given that it almost exclusively operates at an utterance level, scoring 38.5% F1 schedule two.

References

[Com07] Nuance Communications. Grammar developers guide. Technical report, Nuance Communications, 1380 Willow Road, Menlo Park, CA 94025, 2007.

[DMR+10] Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. Entity disambiguation for knowledge base population. In *Conference on Computational Linguistics (Coling)*, 2010.

[GY11] Milica Gasic and Steve Young. Effective handling of dialogue state in the hidden information state POMDP-based dialogue manager. *TSLP*, 7(3):4, 2011.

[SSB12] Rosa Stern, Benoît Sagot, and Frédéric Béchet. A Joint Named Entity Recognition and Entity Linking System. In *EACL 2012 Workshop on Innovative hybrid approaches to the processing of textual data*, April 2012.

[TY10a] Blaise Thomson and Steve Young. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Computer Speech & Language*, 24(4):562–588, 2010.

[TY10b] Blaise Thomson and Steve Young. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Computer Speech & Language*, 24(4):562–588, 2010.

[WY07] Jason D. Williams and Steve Young. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422, 2007.

[YGTW13] Steve Young, Milica Gasic, Blaise Thomson, and Jason D. Williams. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179, 2013.