

# the morning paper

a random walk through Computer Science research, by Adrian Colyer

Made delightfully fast by

≡ MENU

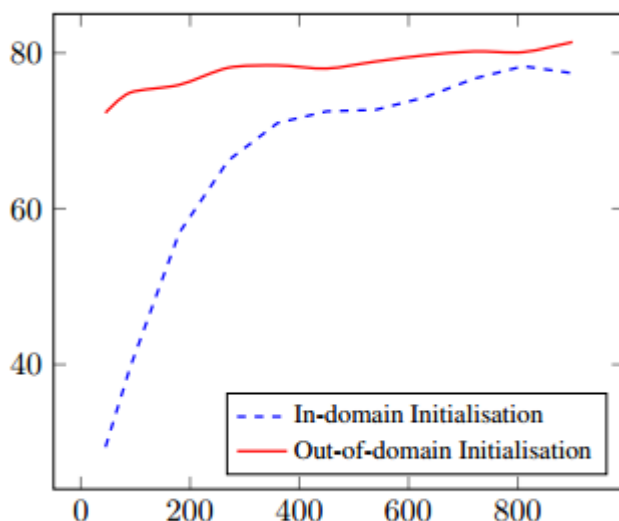
## Multi-domain dialog state tracking using recurrent neural networks

JULY 7, 2016 ~ ADRIAN COLYER

[Multi-domain dialog state tracking using recurrent neural networks](#) a Mrksic et al. 2015

Suppose you want to build a chatbot for a domain in which you don't have much data yet. What can you do to bootstrap your dialog state tracking system? In today's paper the authors show how to transfer knowledge across domains, and through a series of experiments demonstrate that this can be very effective at quickly improving model performance.

Consider a model trained on *out of domain* data – in this case for answering questions about restaurants, hotels, and tourism, that is then used to initialise a model for answering questions about laptops (the *in domain*). The chart below shows how goal accuracy improves as more in-domain training dialogs are made available. Note that the multi-domain model always does better in this case, and the difference is very significant with low amounts of in-domain data.



There are two factors at play here: exposing the training procedure to substantially different out-of-domain dialogs allows it to learn delexicalised features not present in the in-domain training data. These features are applicable to the Laptops domain, as evidenced by the very strong starting performance. As additional in-domain dialogs are introduced, the delexicalised features not present in the out-of-domain data are learned as well, leading to consistent improvements in belief tracking performance.

In fact, it turns out that true out-of-domain training data may be even *more* beneficial than training data from similar domains:

In the context of these results, it is clear that the out-of-domain training data has the potential to be even more beneficial to tracking performance than data from relatively similar domains. This is especially the case when the available in-domain training datasets are too small to allow the procedure to learn appropriate delexicalised features.

Let's take a step back and see how the multi-domain dialog state tracking (DST) training process works. (See yesterday's write-up for [a general overview of the dialog state tracking problem](#)). RNN's are well-suited to DST since they can capture contextual information and hence model and label complex sequences. "The approach is particularly well-suited to our goal of building open-domain dialog systems, as it does not require handcrafted domain-specific resources for semantic interpretation."

The high-level approach is easy to understand. First all of the data available (across multiple domains) is used to train a very general belief tracking model.

This model learns the most frequent and general dialog features present across the various domains.

The starting point for this in Henderson et al.'s RNN framework for belief tracking: a single hidden layer RNN that outputs a distribution over all goal slot-value pairs for each user utterance in a dialog. At each turn the input consists of the ASR (automatic speech recognition) hypothesis, the last system action, the current memory vector, and the previous belief state. *n*-grams are extracted from the input, and to aid generalisation they are *delexicalised*. In Henderson's model, the delexicalised features are used in *addition* to the lexicalised (i.e. regular) ones (not as a replacement). This allows generalisation across slots within the same domain. For the *multi-domain* application studied by authors, all of the

training data from the different (out of domain) domains available is delexicalised, and *the combined slot-agnostic delexicalised dialogs are used to train the parameters of a shared RNN model*. I.e, the delexicalised features are used *instead of* the regular ones, not in addition to.



Delexicalised n-gram features ... replace all references to slot names and values with generic symbols. Lexical n-grams such as [want cheap price] and [want Chinese food] map to the same delexicalised feature, represented by [want tagged-slot-value tagged-slot-name]. Such features facilitate transfer learning between slots and allow the system to operate on unseen values or entirely new slots. As an example, [want available internet] would be delexicalised to [want tagged-slot-value tagged-slot-name] as well, a useful feature even if there is no training data available for the internet slot. The delexicalised model learns the belief state update corresponding to this feature from its occurrences across the other slots and domains. Subsequently, it can apply the learned behaviour to slots in entirely new domains.

We're not quite done building the shared model at this point. The delexicalised training phase does a good job of extracting general dialog dynamics. But it misses out on the affinity of certain features for certain slots. As an example 'near *tagged-slot-value*' is likely to be a feature relating to location.



To ensure that the model learns the relative importance of different features for each of the slots, we train slot specific models for each slot across all the available domains. To train these *slot-specialised* models, the shared RNN's parameters are replicated for each slot and specialised further by performing additional runs of stochastic gradient descent using only the slot-specific (delexicalised) training data.

This completes the training of the shared general model. The general model can then be specialised for the target domain which enables it to learn domain-specific behaviour while retaining the previously learned cross-domain dialog patterns.

## What difference does the out-of-domain pre-training make?

The authors work with six dialog datasets: three based on restaurant bookings (in Cambridge, SF, and Michigan respectively); a tourist information dataset containing dialogs about hotels, restaurants, pubs, and coffee shops; a hotels dataset; and a laptops dataset. Using these they train three different general models:

- The *all restaurants* model uses the three restaurant datasets
- The R+T+H model trains on the restaurant, tourist, and hotel datasets
- The R+T+H+L model trains on all of the datasets.

As a baseline, we can ask how well these models perform in each of the domains (i.e., with no domain-specific training at all).



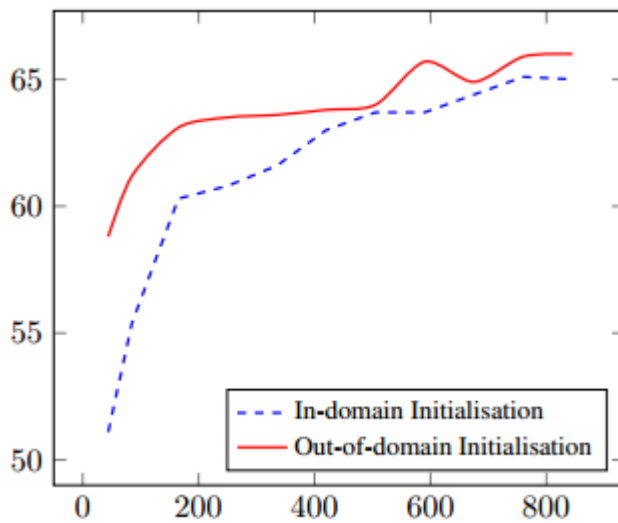
The parameters of the three multi-domain models are not slot or even domain specific. Nonetheless, all of them improve over the domain-specific model for all but one of their constituent domains... The R+T+H+L model is much better at balancing its performance across all six domains, achieving the highest geometric mean (for accuracy) and still improving over all but one of the domain-specific models.

Adding in the slot-specialisation phase to the general model further improves performance in the vast majority of experiments. In the table below, moving down through a column corresponds to adding more out-of-domain training data, and moving right corresponds to slot-specialising the shared model for each slot *in the current domain*.

Model	Cambridge Restaurants		SF Restaurants		Michigan Restaurants	
	Shared Model	Slot-specialised	Shared Model	Slot-specialised	Shared Model	Slot-specialised
<b>Domain Specific</b>	75.0	75.4	51.6	<b>56.5</b>	64.2	65.6
<b>All Restaurants</b>	75.5	77.3	49.6	53.6	67.4	65.9
<b>R+T+H</b>	76.8	<b>77.4</b>	51.2	54.6	<b>68.7</b>	65.8
<b>R+T+H+L</b>	76.8	77.0	50.8	54.1	64.4	66.9
	Tourist Information		SF Hotels		Laptops	
	Shared Model	Slot-specialised	Shared Model	Slot-specialised	Shared Model	Slot-specialised
<b>Domain Specific</b>	62.9	65.1	57.1	57.4	74.7	78.4
<b>R+T+H</b>	65.0	<b>67.1</b>	58.8	60.7	-	-
<b>R+T+H+L</b>	63.6	65.5	57.8	<b>61.6</b>	76.7	<b>78.9</b>

Table 3: Impact of slot specialisation on performance across the six domains (ensembles of 12 models)

The really interesting question, and the one we opened with, is what happens when the out-of-domain shared model is used to initialise a model for a new domain. We've already seen the results for training a laptop dialog model starting with a shared model built from all the other domains. Here's what happens when the R+T+H datasets (minus Michigan) are used to pre-train a shared model for the Michigan restaurant dialogs:



In both experiments, the use of out-of-domain data helps to initialise the model to a much better starting point when the in-domain training dataset is small. The out-of-domain initialisation consistently improves performance: the joint goal accuracy is improved even when the entire in-domain dataset becomes available to the training procedure.

Looks like those [available corpora for building dialog systems](#) might come in handy!

POSTED IN [UNCATEGORIZED](#)

[MACHINE LEARNING](#)

[< PREVIOUS](#)

[Machine learning for dialog state tracking: a review](#)

[NEXT >](#)

[Ten challenges in highly-interactive dialog systems](#)

No comments

Sign up

Sign in



Start a conversation ...

