No Evidence Left Behind: Understanding Semantics in Dialogs using Relational Evidence Based Learning

Asli Celikyilmaz, Dilek Hakkani-Tur, Minwoo Jeong
Microsoft
asli,dilek,minwoo@ieee.org

Abstract

We describe a new structural learning approach to semantic analysis of utterances from conversational dialogs of low-resource domains. Typically an utterance is represented with a multi-layered semantic tag schema: a higher level global context (tag) defines the user's intent, and associated arguments or slot tags define the local context. To deal with the low resource domains, the existing models encode prior information on either the global or the local context, but not on both. Because these components are highly correlated given the domain, we argue that paired priors on both components is more beneficial for semantic analysis of utterances. We introduce a new multi-layer structural learning approach, which integrates *paired* prior information about the global and local components of the utterances. Specifically we encode inter-correlations between the multi-layered components into the joint learner by way of lexicons of paired tags provided by domain experts. Secondly, we introduce systematic ways to extend the paired tag lexicons for low-resource domains from Web-scale data. Across real dialogs from different domains, our approach results in an average improvement of 12%on intent classification and 3% on slot tagging over the baselines.

1 Introduction

A typical spoken language understanding (SLU) engine of a conversational dialog system represents utterances of different domains (e.g., news, travel, etc.) with semantic components. These components are described in two layers: At the top layer is the global semantics, namely the target *domain* (for multi-domain systems) and the user's *intent* (goal). The bottom layer defines sentence-internal (local) *arguments* or *slots*, see examples in Table 1. Supervised methods are generally used to build models for domain/intent classification and slot detection, which have been shown to be quite efficient and robust for resource rich domains (Tur and Mori, 2011). Unfortunately the same does not apply to low-resource (new or uncommon) SLU domains such as text-messaging, computer-games, etc., because of the issues involving language variability, semantic ambiguity caused by limited labeled training data. Researchers usually seek solutions in semi-supervised or unsupervised methods by using additional prior information to boost the performance of models for low resource SLU domains.

In this paper we focus on a new supervised structural learning with priors to deal with these problems of modeling SLU. There are several research that address different methods to injecting prior information into structural learning to improve NLP tasks. While we describe these prior work in §2.1, our work is closely related to the evidence based learning (Reynolds and Bilmes, 2005; Li, 2009) in the way priors are used in structural learning. In (Li, 2009) canonical instances of tags (slot values) are implicitly injected into the Conditional Random Fields (CRF) (Lafferty et al., 2001) as constraining variables on slots (Fig. 1.(a)). Our model differs from the previous work primarily in encoding of prior information into a multi-layered structural

learning method. Specifically, we make two contributions toward semantic analysis of utterances, especially for low resource domains, as follows:

First, we describe a new structural model that can jointly learn the intent and slot tags by extending the earlier hierarchical model, Triangular-Chain Conditional Random Fields (TRICRF) (Jeong and Lee, 2008). We make an assumption that the intent and slots of an utterance are correlated, so we encode the prior information about the correlation between these components by introducing an observed layer in between the intent and slot variables of the graph as shown in (Fig.1.(c)). The prior information to constrain such a joint learner is provided by a set of tag lexicons of paired intent and slot variables. Our aim is to shift the expected tag distribution of words (which are rare or out-of-vocabulary or have noisy tags as in Table (1)) towards the provided paired tag lexicons. Unfortunately, manually obtaining seed prior information this way does not scale for SLU tasks. Because we deal with tag pairings in a multi-layered framework, obtaining paired information on intents and slots from external resources is a tedious and time consuming task. However, there is an opportunity to tailor search query logs to expand the intent specific vocabulary, because as explained in §2.3, query logs can reflect user's intents. As a second contribution we present different ways to automatically extend expert defined information on paired tags using external data sources, specifically web search queries, to generalize our joint model against limited labeled data. Such an approach can be used to easily bootstrap SLU models for new and emerging domains.

We compare our structural learning model against three baselines, a supervised TRICRF (which structurally encodes no priors), a CRF model with virtual evidence (a model that encodes prior information into CRF), and posterior regularization using GE (the model that changes the objective function of CRF to encode expected values on word-tag relations). Relative to the best performing baseline, our approach achieves a 12%increase in intent accuracy, and four point increase in F-score on slot detection performance. We analyze the performance of our system on various domain categories, highlighting our model's strengths (correcting out-of-vocabulary errors and semantic ambiguities) as well as its weaknesses (lacking the power of long term dependencies). We also assess the learning rate of our model, showing that very little annotation is needed to achieve high performance. Finally, to showcase a potential multi-domain application, we use our model to learn aspects in utterances across different domains. While generally vindicating the domain specific models for spoken language understanding, the results point to subtleties in variation across domains that merit further investigation.

2 Background and Motivation

We start by reviewing the structural output learning approaches that use prior information and then lay the background for our joint learning approach suited to SLU tasks.

2.1 Prior Knowledge in Structural Learning

Among some noteworthy work on NLP tasks, pertaining to injecting prior information into structural learning especially when labeled data is limited, are:

Prototype driven learning (PDL) (Haghighi and Klein, 2006), which use prototype phrases[1] to guide their algorithm for learning PCFGs. They use the information about words and their possible tags as features for a Markov Random Fields model.

Constraint driven learning (CoDL) (Chang et al., 2007; Chang et al., 2008) combine constraint inference and perception learning and showed good results on information extraction. Unlabeled instances are selected based not only on predicted output, but also on their consistency with user-defined constraints as functions of input and output variables.

Posterior regularization (Ganchev et al., 2010) incorporate side-information as linear constraints on posterior expectations. *Generalized expectation* (GE) (Mann and McCallum, 2008) is a recent posterior regularization approach. It includes the expected constraint loss term, w.r.t. the model parameter vector to encode preferences by specifying constraints on feature expectations, which is more generalized than CoDL. Specifically, it computes the covariance between the model features and constraint functions making GE inference rather expensive.

[1] A *prototype* is defined as frequently used canonical examples of a lexical type (e.g., entity, slot) and usually provided declaratively (by a domain expert).

Virtual evidence (VE) based learning, e.g., VE-based HMM (Reynolds and Bilmes, 2005) or VE-based CRF (Li, 2009), incorporate external knowledge into sequence models as constraints on potential functions, rather than adding constraints on the objective function, such as in CoDL or GE. Therefore, the inference method does not need to be altered compared to GE methods and they have shown improvements on semantic tagging tasks against PDL approach. Initially, given a tag (slot) they define a list of seed words (prototypes). These values are injected into the CRF as discrete random variables to constrain only the state space, see Fig. (1.a).

2.2 Joint Learning with Paired Tag Lexicons

Recent research has shown that the joint learning methods outperform cascaded or singleton models on several NLP tasks, e.g., sentiment analysis (McDonald et al., 2007; Titov and McDonald, 2008), POS and NER (Finkel and Manning, 2009), joint tagging of web search query classes and their constituent labels (Reisinger and Pasca, 2011; Pantel et al., 2012), etc. Building SLU models is not much different than these research. Hence, the cascaded approaches to building SLU systems (Margolis et al., 2010; Begeja et al., 2004; Wang et al., 2009) learn each component separately; i.e., first the domain is detected, then the intent and slots based on the domain. Unfortunately, this propagates errors through the pipeline, mainly because the inter-correlations between the components are not considered at learning time. Recent work on building joint learning models for SLU, e.g., (Jeong and Lee, 2008; Celikyilmaz and Hakkani-Tur, 2012) inject priors on singleton components (individually on global or local context of utterances). Very little attention has been put on leveraging paired prior knowledge to boost the joint learning.

Our goal is not only to learn a model that can utilize the correlation between the semantic tags of the utterances, but also to handle the noisy tag issues of insufficient labeled data, a common problem in low research domains. For example, the utterance from the computer-games domain:

change my hair to red

has observed intent *i*-UPDATE-AVATAR and slots s-BODY-PART("*hair*") and s-COLOR("*red*"). Because such slot tags are mainly correlated with updating the appearance of an avatar, they are unlikely to be observed in utterances with *i*-CHECK-GAME-SCORE intent such as:

what is my opponent's highest score

On the opposite end, we observe that words can hold different meanings for different user intents. Table 1 shows three utterances with different intents, all containing the word "*two*", but assigned to a different slot tag in each utterance.

Utterance	**Intent Tag / Slot Tag of *"two"***
shooting games played by [*two*].	*i*-FIND-GAME / *s*-NUM-PEOPLE
only players with scores greater than [*two*].	*i*-FIND-PLAYERS / *s*-GAME-SCORE
when did "Pokemon Version [two]" released ?	*i*-FIND-RELEASE-DATE / *s*-GAME-NAME

Table 1: The word ***"two"*** is mapped to there different slot tags in three utterances with different intent tags. The intent tags and surrounding words can guide the joint model to predict the correct slot tag for the same word used in different context.

To tackle with these issues, we propose to jointly constrain the state variables (slot sequences), and the latent intent variable by using paired tag lexicons. Unlike previous work which use slot tag lexicons of words, we define paired tag lexicons of *types* indicative of paired intent and slot. A *type* is an n-gram which has two parts: the "focus" word, which instances the slot tag, and the "context" which mainly instances the utterance's intent. For example

just shooting games

is a 3-gram type with focus word "*shooting*" surrounded by context words. The word "*just*" indicates filtering intent and the focus word indicate the genre slot tag. Therefore, this type is an instance of {*i*-FILTER; *s*-GAME-GENRE} lexicon. We define different n-gram types where focus word is at the center (like above), at the beginning or end of the n-gram (see examples in Table 2). We also define types as patterns to cover long term dependency between the focus word and its context such as:

rate (*) five stars

which is an instance of {*i*-RATE-GAME; *s*-STAR-RATING} tag lexicon. This pattern style type can

Paired Tag Lexicon	Type examples with focus word
{i-FILTER-GAME; s-GAME-GENRE}	• only (*) horror games • for older kids • action games only
{i-FIND-GAME; s-GAME-TYPE}	• competitive games ⟨EOS⟩ • which (*) kids friendly • arcade video games • violent games online
{i-RATE-GAME; s-STAR-RATE}	• assign (*) five stars • rate that three

Table 2: Examples of 3-gram and patterned types (with (*)) of three paired intent-slot tag lexicons from games domain. The focus word of each type is underlined. ⟨EOS⟩ indicate types matching the end of sentence.

constrain the intent of an utterance such as *"rate the last movie that i watched as five stars"* along with the slot tag of word *"five"*. In §3.3, we define how we integrate the paired priors into the structure of the TRICRF model.

2.3 Expanding Lexicons via Search Logs

One of the issues of dealing with low resource domains is learning the unknown words. In the context of SLU, unknown words (or out-of-vocabulary (OOV)) are defined as having never appeared in the training corpus. When the training corpus is small, the percent of words which are unknown can be high. Suffice to say that manually defining the prior information (e.g., populating tag lexicons or writing rules) to handle the OOV words in learning algorithms does not scale for language understanding tasks. Figure Xshow some statistics regarding the coverage of the seed examples in training data. On average less than five percent of the seed instances are observed in the training set. Luckily, search query click-logs (QCL) have been a useful resource for extracting in domain queries to bootstrap SLU models. For example, (Li et al., 2008) uses query click logs to bootstrap intent detection models when the labeled data is limited.

The QCL can provide implicit supervision of the broad category of user's intent because of the way the clicked URLs are structured. For instance, assume these two queries are issued by two separate users: "*reviews for skyfall*" and "*skyfall critics*". After entering the query, they both click on "rottentomatoes.com" from the list returned by the search engine. This indicate that the two users are likely to have the same intent, which is most probably the *i-*CHECK-REVIEWS. The same is true for the clicked hosts: users clicking on links with "games.com" domain are likely to purchase/check-out online games.

In the past, QCL's have been used to extend dictionaries for several tasks: (Li, 2009; Hillard et al., 2011) extend sparse tag lexicons using in-domain query log data to improve SLU tasks, (Komachi et al., 2009) construct semantic categories to improve information retrieval, (Pantel et al., 2012)'s joint model extracts query intent and entity types of new search queries, leveraging signals from query context, click and entity information extracted from web search queries logs. In an analogical way to the previous work, we use a large query click log (QCL) data to automatically generate new instances for *paired* tag lexicons, which has not been investigated for SLU tasks before. We mainly focus on the rare and noisy intent-slot tag pairs to handle the OOV issue in §4.

3 Structural Learning for SLU with Priors

We present progression of structural learning methods leading up to our proposed joint learner.

3.1 CRF with Priors

CRFs (Lafferty et al., 2001) are undirected graphical models, which specify the conditional probability of an assignment of output labels given a set of input observations (Fig. 1.(a)). Let $\mathbf{w}=(w_1, w_2, \ldots, w_T)$ be an input sequence (e.g., utterance) of T tokens, w_t denotes word at location t and $\mathbf{s}=(s_1, s_2, ..., s_T)$ is the state sequence (e.g., slot tags). A linear chain CRF defines conditional probability of $\boldsymbol{s}$:

$$p_\lambda(\mathbf{s}|\mathbf{w}) = \frac{1}{Z_\lambda} \prod_{t=1}^{T} \phi^{(t)}\left(s_t, s_{t-1}, \mathbf{w}\right) \qquad (1)$$

where $\phi_t(\cdot)$ is the local potential function represented with maximum cliques of the graph. Partition function $Z_\lambda(\mathbf{w}) \triangleq \sum_{\mathbf{s}} \prod_{t=1}^{T} \phi_t(s_t, s_{t-1}, \mathbf{w})$ ensures that the probability of all state sequences sum to one. In (Li, 2009), slot tag dictionaries are provided as define prior information on selected slot tags. Then they are encoded into CRF graphical models as sequence of virtual evidences (VE) $v_1, v_2, ..., v_T$, in parallel to the latent state variables (Fig. 1.(a)). A constant value, v_t=1, is assigned to each piece of evidence and connected with s_t and s_{t-1} forming a new set of maximum cliques. The

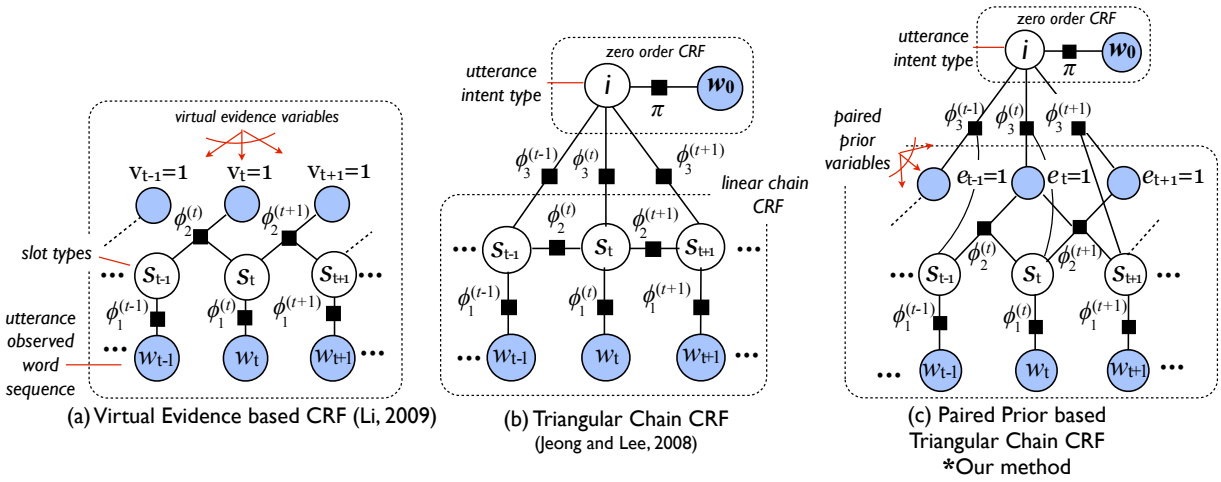

Figure 1: Factor graph representation of different types of CRF models with prior information. Solid and empty nodes denote observed and hidden variables and w_t and s_t's are token and slot sequences and $\boldsymbol{w}_0$ is the utterance with intent i. v's denote virtual evidence (tag priors) and e's denote paired priors.

VE based CRF extends the potential functions of CRF, $\phi^{(t)}$, to include priors $v_{1:T}$ as:

$$p_\lambda(\mathbf{s}, i|\mathbf{w}, \mathbf{v}) = \frac{1}{Z_\lambda} \prod_{t=1}^{T} \phi^{(t)}(s_t, s_{t-1}, v_t, \mathbf{w}) \quad (2)$$

$$\phi^{(t)}(s_t, s_{t-1}, v_t, \mathbf{w}) = \overbrace{\phi_{\mathbf{1}}^{(t)}(s_t, \mathbf{w})}^{observation} \cdot \overbrace{\phi_{\mathbf{2}}^{(t)}(s_t, s_{t-1}, v_t)}^{transition\text{-}v} \quad (3)$$

$$\phi_{\mathbf{1}}^{(t)}(s_t, \mathbf{w}) = \exp\left(\sum_k \lambda_1^k f_1^k(s_t, \mathbf{w}_t)\right) \quad (4)$$

$$\phi_{\mathbf{2}}^{(t)}(s_t, s_{t-1}, v_t) = \exp(\sum_k \lambda_2^k f_2^k(s_t, s_{t-1}, v_t, t)) \quad (5)$$

where $f_1^k(s_t, \mathbf{w}_t)$ is a feature function that encodes aspects of the current word and $f_2^k(s_t,s_{t-1},v_t,t)$ encodes the VE feature functions of the state transitions at the current index t. $\lambda_{1/2}$ are the weight vectors, where λ_2 is pre-defined and fine-tuned using cross-validation.

The VE based CRF adds constraints only on the state variables (slots). Because we have an additional latent variable, the intent (global) label of an utterance, and we need to consider the correlation between intents and slots, we investigate joint learning, more suitable to utterance semantic analysis.

3.2 Triangular Chain CRF - TRICRF

TRICRF (Jeong and Lee, 2008) is a joint structural learning method which assumes that the utterances are intrinsically grouped into latent topic (intent) clusters i. Hence, the latent variables, e.g., intent i and slots $\boldsymbol{s}$ can be jointly learned (Fig. 1.(b)). The conditional probability $p(\boldsymbol{s},i|\mathbf{w})$ of utterances to predict slot tag s_t of token t and intent i is:

$$p_\lambda(\mathbf{s}, i|\mathbf{w}) = \frac{1}{Z_\lambda} \left(\prod_{t=1}^{T} \phi^{(t)}(i, s_t, s_{t-1}, \mathbf{w})\right) \cdot \pi(i, \mathbf{w}) \quad (6)$$

$\phi^{(t)}$ and π are potentials over the triangular chain graph and partition function $Z_\lambda \triangleq \sum_{\mathbf{s},i} \prod_{t=1}^{T} \phi^{(t)}(\mathrm{i},s_t,s_{t-1},\mathbf{w}) \cdot \pi(i, \mathbf{w})$ ensures that the distribution is normalized. TRICRF is an integration of: (*i*) zero-order CRF, which defines the potential function $\pi(\cdot)$ over the entire utterance to predict the hidden intent variable i; (*ii*) linear-chain CRF (Lafferty et al., 2001), which defines potential functions $\phi_t(\cdot)$ to predict the latent slot sequence $\mathbf{s}$. The two are combined with edges in the graph.

In linear chain CRF, the slot tag of a word depends on the current and surrounding words (context) of the utterance, formulated as, $\phi_{\mathbf{1}}^{(t)}(s_t, \mathbf{w})$. Whereas, the latent variable intent i does not directly influence how the random variables $w_{1:T}$ are emitted given the latent slot sequence $s_{1:T}$. Specifically, user's intent can be inferred in most utterances using a simple language model, $p(i|\mathbf{w}) = \pi(i, \mathbf{w})$, which discovers intent bearing phrases. For instance, it is trivial to capture phrases such as "*show me*" or "*find me*", which imply i-FIND intent, or *"remove comedies"*, which implies i-FILTER intent. Slot tags can also provide information about the intent of the utterance, e.g., in restaurants domain, s-MEAL-TYPE slot

is more likely to appear in utterances with *i*-FIND-RESTAURANT intent than utterances with *i*-FIND-OPENING-TIME intent. This is conveniently formulated in TRICRF with $\phi_{\mathbf{3}}^{(t)}(i, s_t)$ potentials. Under these assumptions, we factorize the $\pi(\cdot)$ and $\phi_t(\cdot)$ in Eq. (6) as follows:

$$\pi(i, \mathbf{w}) = \exp\left(\sum_k \lambda_0^k f_0^k(i, \mathbf{w}_0)\right) \quad (7)$$

$$\begin{cases} \phi^{(t)}(i, s_t, s_{t-1}, \mathbf{w}) = \\ \overbrace{\phi_{\mathbf{1}}^{(t)}(s_t, \mathbf{w})}^{\textit{observation}} \cdot \overbrace{\phi_{\mathbf{2}}^{(t)}(s_t, s_{t-1})}^{\textit{transition}} \cdot \overbrace{\phi_{\mathbf{3}}^{(t)}(i, s_t)}^{\textit{i-s edge}} \end{cases} \quad (8)$$

$$\phi_{\mathbf{1}}^{(t)}(s_t, \mathbf{w}) = \exp\left(\sum_k \lambda_1^k f_1^k(s_t, \mathbf{w})\right) \quad (9)$$

$$\phi_{\mathbf{2}}^{(t)}(s_t, s_{t-1}) = \exp\left(\sum_k \lambda_2^k f_2^k(s_t, s_{t-1})\right) \quad (10)$$

$$\phi_{\mathbf{3}}^{(t)}(i, s_t) = \exp\left(\sum_k \lambda_3^k f_3^k(i, s_t)\right) \quad (11)$$

$\lambda_{0/1/2}$ are the weight vectors, $\mathbf{w}_0$ is the utterance, and $f_1^k(s_t, \mathbf{w})$ and $f_2^k(s_t, s_{t-1})$ are kth feature functions that encode aspects of the current word and state transitions at index *t*. $\phi_{\mathbf{3}}^{(t)}$ is a constraint on the dependency between the latent intent *i* and slot ***s*** variables, parameterized with functions $f_3^k(i, s_t)$.

3.3 Proposed Paired Prior Based TRICRF

We introduce a sequence of observed variables $\{e_1, e_2, ..., e_T\} \in E_{is}$ as additional level of priors to constrain utterance's global (intent *i*) and local context (slots *s*). We modify the TRICRF by joining the zero-order and linear-chain CRF and the priors by adding edges between *i*, and **s**, in the factor graph model as shown in Fig. 1.(c). Hence the conditional probability distribution of TRICRF of Eq. (6) is reconstructed with the added e_t's as:

$$p_\lambda(\mathbf{s}, i|\mathbf{w},\mathbf{r}) = \frac{1}{Z_\lambda} \prod_{t=1}^{T} \phi^{(t)}(i, s_t, s_{t-1}, e_t, \mathbf{w}) \cdot \pi(i, \mathbf{w})$$

Then the potentials factorize as follows:

$$\pi(i, \mathbf{w}) = \exp\left(\sum_k \lambda_0^k f_0^k(i, \mathbf{w}_0)\right) \quad (12)$$

$$\begin{cases} \phi^{(t)}(i, s_t, s_{t-1}, e_t, \mathbf{w}) = \\ \overbrace{\phi_{\mathbf{1}}^{(t)}(s_t, \mathbf{w})}^{\textit{observation}} \cdot \overbrace{\phi_{\mathbf{2}}^{(t)}(s_t, s_{t-1}, e_t)}^{\textit{transition-r*}} \cdot \overbrace{\phi_{\mathbf{3}}^{(t)}(i, s_t, e_t)}^{\textit{i-e-s edge*}} \end{cases}$$

$$\phi_{\mathbf{2}}^{(t)}(s_t, s_{t-1}, e_t) = \exp\left(\sum_k \lambda_2^k f_2^k(s_t, s_{t-1}, e_t, t)\right) \quad (13)$$

$$\phi_{\mathbf{3}}^{(t)}(i, s_t, e_t) = \exp\left(\sum_k \lambda_3^k f_3^k(i, s_t, e_t)\right) \quad (14)$$

The $\phi_{\mathbf{1}}^{(t)}$ is as same as in Eq. (4), but $\phi_{\mathbf{2}}^{(t)}$ and $\phi_{\mathbf{3}}^{(t)}$ include the paired prior variables e_t's. λ_1 and λ_2 are the weight vectors of the state transition and intent-slot dependency feature functions respectively.

Let E_{is} denote a paired intent-slot tag lexicon of types (such as in Table 2). At learning time, we define feature functions that sustain the label consistency with each E_{is} as follows:

- For each word token w_t of an utterance, a constant paired prior e_t=1 is used.
- Each e_t is connected to three variables: the intent random variable i, the preceding s_{t-1} and current slot s_t, forming two new set of cliques, $\phi_{\mathbf{2}}^{(t)}(s_t, s_{t-1}, e_t)$ and $\phi_{\mathbf{3}}^{(t)}(i, s_t, e_t)$ (as show in Eq.(13) & Eq.(14)).
- If w_t is a focus word of a type in E_{is}, and the context of that type matches with the context of w_t, then we prefer s_t to be equal to the slot tag of that type. Similarly we prefer the intent i of the utterance be equal to the intent of that type. To this end we set $f_2^k(s_t, s_{t-1}, e_t, t)$=1, and $f_3^k(i, s_t, e_t)$=1.
- We control the strength of this prior by the λ_1 and λ_2 weights of f_1 and f_2. In the experiments, we optimize these weights based on a development set.

Inference. TRICRF with PEs is concerned with two separate inference problems: (1) predicting marginal distributions for each factor: $p_\lambda(s_t|\mathbf{w})$=$\sum_i p_\lambda(i,s_t,e_t|\mathbf{w})$, $p_\lambda(i|\mathbf{w})$, and $p_\lambda(s_t,s_{t-1},e_t|\mathbf{w})$=$\sum_i p_\lambda(i,s_t,s_{t-1},e_t|\mathbf{w})$, and partition Z_λ, (2) computing Viterbi decoding for predicting the slot sequence for the new utterances.

The inference uses multiple viterbi searches for linear chain CRF's computing $\hat{s} = \arg\max_s p_\lambda(s, i = l|w)$ for each given topic i. The complexity of the inference is $\mathcal{O}(\mathcal{T}|\mathcal{S}|^2\mathcal{I})$, where $\mathcal{T}$, $\mathcal{I}$, and $\mathcal{S}$ are the sizes of each sequence, intent and slot spaces respectively.

4 Paired Tag Lexicon Expansion

In the context of many NLP tasks, unknown words or phrases (or out-of-vocabulary, OOV) are defined as having never appeared in the training corpus. When the training corpus is small, the percent of words which are unknown can be high. In addition,

in the low-resource domains most types won't be found in the initial tag lexicons. For these reasons, we automatically expand our initial paired lexicons into ones that has coverage for most of the types. Our ingredients are query-click log data, initial seed lexicons and a supervised CRF method.

4.1 Query-Click Logs (QCL) Data

During a search session, users issue a search query for which the search engine presents a list of result urls. Of the search results, users click on urls that are representative of their intent. This interaction is captured by means of a click, which is logged by most search engines as click-through data. For instance, a search log may contain the clicked urls U for queries Q issued by different users as shown in Fig. (2).

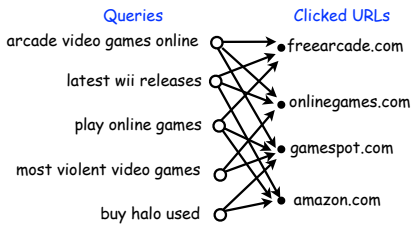

Figure 2: Sketch for query-click relations.

As we discuss in §2.3, QCL can provide implicit supervision about the intent of the users. Importantly, we seek search queries which have similar intents as the utterances in our training dataset. At start, we collect raw query logs observed in a web search engine over a period of time (6 months-1 year). Each query is associated with a list of urls that the users clicked along with their click frequency. From these logs we construct our binary QCL data matrix of search queries Q where columns indicate their clicked urls U as shown in Fig. 3. We use a frequency threshold (threshold=10) to turn on/off a particular clicked url cell in the matrix.

4.2 N-gram Type Expansion

We seek to expand the list of n-gram *type*s of paired tag lexicons. Here, we explain our extraction approach tracing between the queries Q and the clicked urls C (see demonstration in Fig. 3). Our goal is to find new search queries that share the same intent as the seed types, then parse new types out of them. As running example we choose *"arcade video games"*

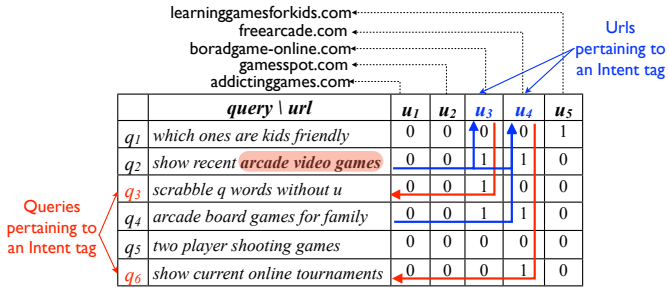

Figure 3: QCL data matrix of selected queries Q and urls U. "1"s indicate url is clicked given the query, "0" otherwise. Given a type (highlighted) with a known intent, the construction of a cluster of urls pertaining to that intent follows the blue arrows starting from the queries (step 1 & 2 of §4.2), and extraction of new queries related to the urls in that intent cluster follows the red arrows starting from the urls (step 3 of §4.2)

as type with focus *"video"* from {i-FIND-GAME; s-GAME-TYPE} paired lexicon.

Step 1: Find Query Intent Clusters (Q^i): We take each seed type entry of paired tag lexicons, canonize the focus word to construct patterns. The pattern of the running type is *"arcade (*) games"*. Using each pattern we pull matching queries and assign them the intent label (i) of that type. In Table 3, Q^i=$\{q_2, q_5\}$ are the queries that match this pattern. We add the newly discovered types to the paired tag lexicon, i.e., *"arcade online games"*, *"arcade board games"*.

Step 2: Find Url Clusters per Intent (U^i): We construct the url clusters pertaining to intents U^i under the assumption that user's with similar intents click on similar urls. We select the list of urls that has the value of "1" for the queries in Q^i to define intent clusters of urls $U^i \in U$. For the running examples the selected list of urls are U^i=$\{u_3, u_4\}$ as shown in Fig. 3.

Step 3: Expand Queries from Url Clusters (Q^{i*}). Once we have the clusters of click urls per selected intent, U^i, we start to search in reverse to collect additional queries Q^{i*} that share the same click urls with Q^i. For example, u_3 and u_4 are clicked by some users who also entered queries $\{q_3;q_6\} \in Q^{i*}$.

Step 4: Extract New Types. Now that we collected new queries Q^{i*} that have the same intent tag as the starting types, we aim to extract potential types from these queries. First, we train a slot tagger model using CRF with n-gram features on the given labeled training data. We decode the new queries in

Q^{i*} and select the predicted slot types. For example, our CRF model decodes the query q_6 and predicts these as slots:

$$\{current\}_{s-GAME-RELEASE-DATE}$$
$$\{online\}_{s-GAME-DESCRIPTION}$$

We construct new types using the predicted slots as focus word. For example, using the new slot *"online"* we collect the following types from Q^{i*}:

current online tournaments
online tournaments ⟨*EOS*⟩
show current online

Lastly, we add the new types to the corresponding paired tag lexicon. Thus, the above three types are added to {*i*-FIND-GAME; *s*-GAME-DESCRIPTION} lexicon.

5 Experiments and Results

To better understand the effect of using paired priors in a joint learning model, we present the results of our experiments on different domains.

5.1 Datasets

Flight Domain Data. ATIS (Air-Travel Information System (Price, 1990)) is the goal-oriented dialog data, which has been mainly used for benchmark analysis of new learning tools for SLU. The utterances are based mainly on reservations, checking flight status, etc. The data is labeled with 16 intents and 119 slot tags, converted to IOB (inside/outside/begin) format. There are 4561 train and 891 test utterances[2].

Entertainment Domain Datasets: The data are internally collected from real-use scenarios of a spoken dialog system. We focus here on three domains on audiovisual media:

(i) Games: about online/offline games;

(ii) Music: information on songs, artists, albums;

(iii) Movies: mainly movies and television shows.

The user is expected to interact by voice with a system that can perform a variety of tasks in relation to each media, including (among others) browsing, searching, querying information, purchasing and playing. We use the transcribed text utterances obtained from ASR engine. There are 7523, 7510, and 29492 training and 2497, 2490, and 10087 testing utterances in games, music and movies datasets, respectively. The intents and slots of each domain are summarized in Table 3.

Domain	Intent (I) ; Slot (S)
flight (ATIS)	I: purchase-ticket, get-info-flight/airline/aircraft.. S: depart-city,meal-type,aircraft-code....
music	I: find-artist, buy-album, S: name, album-type(*single*), source('*top charts*')
games	I: find-game, buy-game, get-screenshot,... S: company,version, edition,...
movie	I: find-movie,actor,get-release-date,find-theater S: , description('*frightening*')

Table 3: List of intents and semantic slot tags for each domain. *italicized* are examples for contextual slots.

More than half of the slot tags in our data are *named-entities* such as s-MOVIE-NAME, s-ACTOR, s-GAME-NAME, etc. Whereas the other half is mainly descriptive slot tags about the items that the user is seeking, e.g., s-GAME-DESCRIPTION, s-MOVIE-GENRE, etc. Learning a model to predict descriptive slots is much harder, because context is the key characteristic for predicting the descriptive tags. Especially the slot detection performance is effected the most due to poor context of low resource domains. Therefore, we asked our domain experts to provide us with around 50 seed types for each intent-slot tag lexicon of mainly the descriptive slot tags. We also chose tags that have less overlap between the training and testing data to better demonstrate the effects of the usage of prior knowledge.

We later expand the paired tag lexicons using search query logs as in §4. The QCL data is formed from query-clicks observed over 6-months in a web search engine, where each query has frequency of at least 10. Among 10 million queries, we collected around 20K queries per each entertainment domain, by using several filtering methods including removal of unigram queries, queries including profanity vocabulary, etc. From the remaining queries we collected 1000 unique click-urls per domain. Using n-gram type expansion method explained in §4, we expand the seed relational prototypes to average ∼1000 vocabulary per-domain.

Different from the earlier work we mention

5.2 Models

Baselines The following baselines are used for the SLU's intent and slot detection tasks:

[2]For joint models utterances with intent and slot labels should be aligned, therefore some utterances are not used.

Paired Intent ; Slot Tag	Seed Types w/ focus word	Variations of the focus words
i-FIND-GAME ; *s*-RELEASE-DATE	ones recently out, out this fall	October, summer, holiday, ...
i-FIND-GAME ; *s*-STAR-RATING	best rated games, ones rated 8	highest, up, higher,...
i-FIND-GAME ; *s*-DESCRIPTION	games about swimming	archer, ancient, battleship,...
i-FIND-INSTRUCTIONS ; *s*-GAME-NAME	hits for shaiya, game halo rules	dance, hip-hop,...
i-BUY-GAME ; *s*-DESCRIPTION	most wanted game, violent games out	trustworthy, best, free,...

Table 4: Sample types for paired tag lexicons of the ENTERTAINMENT-GAMES domain. The seed types indicate the user defined prior information. The last column is the expansion of The last column is the list of focus words, which are variations of the the focus word of seed types in the second column. These are obtained from n-gram expansion method using web search query click logs.

`Discrete`: A traditional approach to SLU is predicting intent and semantic slot variables separately. We used Max-Entropy classifier (MAX-ENT) for intent and CRF for slot detection models.

`Cascaded`: This model simulates a joint learner but learns the intent and semantic slot models separately. Specifically, the predicted intent obtained from the a MAX-ENT model is used as a feature to the CRF based slot detection model, and the predicted slot values from the CRF model are used as features to the MAX-ENT based intent model.

`TRICRF`: Joint modeling baseline that learns the intent and slots together (Jeong and Lee, 2008).

`CRF-VE`: VE based learning, that induces prototypes (word lists per slot type as in Table 4) as constraints into CRF (Li, 2009), making it suitable for ***only*** slot detection benchmark analysis. `CRF-VE-EXP` is another baseline that uses the expanded tag lexicons.

`GE-Discrete`: Our last baseline that uses generalized expectation criteria for learning classifiers via labeled features. Specifically, GE uses the labeled features to set the certain model expectations to be close to the target distributions. We use the tag lexicons as labeled features to train GE for MAX-ENT (Mann and McCallum, 2007) for intent and GE for CRF for slot models separately (Mann and McCallum, 2008). We refer the model that uses the expanded tag lexicons as `GE-Discrete-EXP`.

Proposed Joint Models We tried two versions of our joint learning tool:

`TRICRF-PP`: The joint structural model with paired priors is our first model to use *seed* types from paired tag lexicons. A sample of seed types for the entertainment domains are shown in Table 4).

`TRICRF-PP-EXP`: Our final proposed model uses the expanded types extracted from the query click (§4) as priors for the joint model.

Experimental Setup To train CRF and MAX-ENT, and all the TRICRF models including our proposed approach, we used the open source tool tool[3]. To train the `GE` models, we use the open source MALLET tool (McCallum and Kachites, 2002). To avoid over-fitting, we use $\mathcal{L}_2$ regularization for each CRF and `TRICRF` models and limited memory version of the quasi-Newton method (L-BFGS) to optimize the parameters.

All data is labeled with domain/intent/slot tags. All models are trained using lexical features with up to 5-grams. For quantitative analysis, we use F1 pairwise measure.

(Li, 2009) reports that learning the weights λ_2 in Eq. (5) for priors (VE constraints) does not improve the performance because it undermines the role of other useful features because sufficient training data is not always available. So they use user defined constant values. In our work, we start with user defined values for the paired prior weights λ_2 and λ_3 in Eq.(13) & Eq.(14) and search for the optimum values based on the performance on a held-out set.

The baseline models that use priors, namely `GE-DISCRETE` CRF with VE and GE based models define constraints using tag lexicons of *unigrams* as shown in Table 4. Unlike previous work, our proposed joint model uses paired priors of *types* as additional observed variables. To construct the tag lexicons for these ba

5.3 Experiment Results

Experiment 1. We evaluate the performance of our lexical models on each domain separately. Our

[3]Open source package https://github.com/minwoo/TriCRF

		Flight (ATIS)		Music		Games		Movies		**Avg**	
	Model	Intent	Slot	Intent	Slot	Intent	Slot	Intent	Slot	Intent	Slot
baselines	Discrete	95.43	90.66	81.71	81.84	87.73	86.27	85.13	76.80	87.50	83.89
	Cascaded	95.67	90.67	92.22	81.50	88.71	84.67	85.21	76.71	90.45	83.39
	TRICRF	94.71	91.04	95.11	80.47	90.86	85.06	95.14	77.08	93.96	83.41
GE	GE-Discrete	-		-							
	GE-Discrete-EXP										
VE	CRF-VE	-	91.71	-	81.86	-	86.57	-	76.70	-	84.21
	CRF-VE-EXP	-	-	-	82.03	-	86.57	-	76.50	-	84.20
RE	TRICRF-RE	**96.30**	91.31	95.23	82.01	95.84	86.70	95.18	77.23	**95.64**	84.31
	TRICRF-RE-EXP	-	-	95.22	**82.47**	**95.90**	**86.96**	94.73	**77.69**	95.54	**84.61**

Table 5: Intent and Slot **F-Scores** for various baselines including a virtual evidence based constraint modeling (VE). *RE-ours: proposed models, "**Avg**" denotes macro-average across four domains. The best models are **bolded**.

complete set of results are in Table 5. On ATIS data we did not use query clicks, therefore we don't report expanded model results. As expected, the Discrete and Cascaded models perform the worst. The baseline TRICRF benefits from joint modeling and greatly improves upon other baselines on intent detection but falls short on the slot models. The models that use VE constraints slightly improve the baselines on slot detection. Our model with RE outperforms all the baselines, for almost all the domains. It falls short on the music domain but statistically indistinguishable in terms of F-score (using paired-*t-test* with $p < 0.01$). It performs slightly better than the VE models, up to 2% improvement on intent and 0.5% on slot models. Additionally, TRICRF-RE-EXP models are consistently better than the rest of the models, suggesting that our RE models are robust and perform well even when constraints are discovered from unlabeled web sources such as query click logs.

Different from CRF-VE and GE models, our proposed approach can incorporate prior information on phrases, not just on single words. To enable this, our model accepts paired information on n-gram types.

Experiment 2. We may need to show percentage of the words or phrases that do not appear in the training data.

Here, we evaluate the performance gain with using evidence on intent and slot models when there are limited labeled data. Using a list of RE prototypes, we start with small amount of training data and incrementally increase data size. We build TRICRF and CRF models with and without evidence on ATIS data, however, this time we use the entity feature provided in (Price, 1990).

Incidentally, more training data implies better results in Fig. 4, encoding prior knowledge as relational evidence in joint models (**TRICRF-RE**) when there is not enough labeled training data outperforms all the baselines both for intent and slot models. This suggest that our approach would be useful especially when there is a new domain and SLU systems needs to be built with limited amount of labeled utterances. Additionally, when all the training data is used, the best performing model, using semantic and syntactic features, yielded 95F on slots and 96.98F on intent models (Tur et al., 2012). **TRICRF-RE**, on all data using *only* n-gram and single entity feature yields 93.2F on slots and 96.3F on intent models, showing notable performance against the state-of-the-art.

Experiment 3. Multi-domain models are natural extensions to dialog systems, because humans do multiple actions when they interact with personal assistant systems, e.g., browse web, share with friends, purchase, etc. While domain detection tools can discover which SLU system to use at runtime, we argue that joint modeling of multi-domain data can benefit from interrelationships between cross-domain aspects without the need for domain detection models.

We combined the entertainment domain utterances to build a multi-domain **TRICRF-RE-EXP** model. The best performing model (Table 5) yields 95.54F for intent and 84.61F for slot. Our multi-domain model yielded 94.1F for intent and 83.25F for slot, which is still comparable to the best performing average. Even so, the advantage of multi-

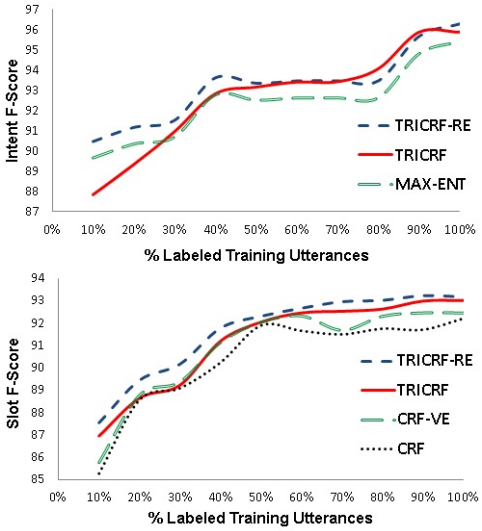

Figure 4: F-score on ATIS for increasing number of labeled samples. (Top) intent; (Bottom) slot performance. blue/dashed=TRICRF-RE; red/solid=TRICRF; green/double-dashed=Max-Ent/CRF-VE; black/dotted= CRF. REs boost the performance even when there is sparse labeled utterances.

domain modeling against single-domain is the reduced number of models to build an SLU system.

6 Conclusions

We have presented a new joint learning model which uses constraints as potential functions for predicting aspects in spoken language utterances. Our model outperforms cascaded approaches and joint modeling baselines on several experiments. Possible feature work is modeling session dialog data where utterances use information from previous dialog turns. Our experiments show that especially for low resource languages using paired priors improve the meaning extraction from utterances.

References

L. Begeja, B. Renger, Z. Liu D. Gibbon, and B. Shahraray. 2004. Interactive machine learning techniques for improving slu models. In *Proc. of WS on Spoken Language Understanding for Conversational Systems and Higher Level Linguistic Information for Speech Processing - HLT-NAACL 2004*.

A. Celikyilmaz and D. Hakkani-Tur. 2012. A joint model for discovery of aspects in utterances. In *Proc. ACL 2012*.

M.-W. Chang, L.-A. Ratinov, and D. Roth. 2007. Guiding semi-supervision with constraint-driven learning. In *Proc. ACL 2007*.

M.W. Chang, L. Ratinov, N. Rizzolo, and D. Roth. 2008. Learning and inference with constraints. In *Proc. AAAI*.

J. Finkel and C. D. Manning. 2009. Joint parsing and named entity recognition. In *Proc. NAACL 2009*.

K. Ganchev, J. Graca, J. Gillenwater, and B. Taskar. 2010. Posterior regularization for structured latent variable models. In *Journal of Machine Learning Research*, volume 11.

A. Haghighi and D. Klein. 2006. Prototype-driven learning for sequence models. In *Proc. HLT-NAACL 2006*.

D.R. Hillard, A. Celikyilmaz, D. Hakkani-Tr, and G. Tr. 2011. Learning weighted entity lists from web click logs for spoken language understanding. In *Proc. Interspeech*.

M. Jeong and G. G. Lee. 2008. Triangular-chain conditional random fields. *EEE Transactions on Audio, Speech and Language Processing (IEEE-TASLP)*.

M. Komachi, S. Makimoto, K. Uciumi, and M. Sassano. 2009. Learning semantic categories from clickthrough logs. In *Proc. ACL 2009*.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML 2001*.

X. Li, Y.-Y. Wang, and A. Acero. 2008. Learning query intent from regularized click graphs. In *Proc. SIGIR 2008*.

X. Li. 2009. On the use of virtual evidence in conditional random fields. In *Proc. EMNLP 2009*.

G. Mann and A. McCallum. 2007. Simple, robust, scalable semi-supervised learning via expectation regularization. In *Proc. ICML 2007*.

G. Mann and A. McCallum. 2008. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *Proc. ACL 2008*.

A. Margolis, K. Livescu, and M. Osterdorf. 2010. Domain adaptation with unlabeled data for dialog act tagging. In *Proc. Workshop on Domain Adaptation for Natural Language Processing at the ACL 2010*.

A. McCallum and A. Kachites. 2002. Mallet. In *mallet.cs.umass.edu*.

R. McDonald, K. Hannan, T. Neylon, M. Wells, and J. Reynar. 2007. Structured models for fine-to-coarse sentiment analysis. In *Proc. ACL 2007*.

P. Pantel, T. Lin, and M. Gamon. 2012. Mining entity types from query logs via user intent modeling. In *Proc. ACL 2012*.

P. J. Price. 1990. Evaluation of spoken language systems: The atis domain. In *Proc. DARPA workshop on Speech and Natural Language*.

J. Reisinger and M. Pasca. 2011. Fine-grained class label markup of search queries. In *Proc. of ACL 2011*.

S. Reynolds and Jeff Bilmes. 2005. Part-of-speech tagging using virtual evidence and negative training. In *In. Proc. HLT-EMNLP 2005*.

I. Titov and R. McDonald. 2008. A joint model of text and aspect rating of sentiment summarization. In *Proc. ACL 2008*.

G. Tur and R. De Mori. 2011. Spoken language understanding: Systems for extracting semantic information from speech. *Wiley*.

G. Tur, D. Hakkani-Tur, L. Heck, and S. Parthasarathy. 2012. Sentence simplification for spoken language understanding. *Proc. ICASSP 2012*.

Y.Y. Wang, R. Hoffman, X. Li, and J. Syzmanski. 2009. Semi-supervised learning of semantic classes for query understanding from the web and for the web. In *The 18th ACM Conference on Information and Knowledge Management*.