Weakly Supervised Slot Tagging with Partially Labeled Sequences from Web Search Click Logs

Young-Bum Kim[†] Minwoo Jeong[†] Karl Stratos[‡] Ruhi Sarikaya[†]

[†]Microsoft Corporation, Redmond, WA
[‡]Columbia University, New York, NY
{ybkim, minwoo.jeong, ruhi.sarikaya}@microsoft.com
stratos@cs.columbia.edu

Abstract

In this paper, we apply a weakly-supervised learning approach for slot tagging using conditional random fields by exploiting web search click logs. We extend the constrained lattice training of Täckström et al. (2013) to non-linear conditional random fields in which latent variables mediate between observations and labels. When combined with a novel initialization scheme that leverages unlabeled data, we show that our method gives significant improvement over strong supervised and weakly-supervised baselines.

1 Introduction

A key problem in natural language processing (NLP) is to effectively utilize large amounts of unlabeled and partially labeled data in situations where little or no annotations are available for a task of interest. Many recent work tackled this problem mostly in the context of part-of-speech (POS) tagging by transferring POS tags from a supervised language via automatic alignment and/or constructing tag dictionaries from the web (Das and Petrov, 2011; Li et al., 2012; Täckström et al., 2013).

In this work, we attack this problem in the context of slot tagging, where the goal is to find correct semantic segmentation of a given query, which is an important task for information extraction and natural language understanding. For instance, answering the question "when is the new bill murray movie release date?" requires recognizing and labeling key phrases: e.g., "bill murray" as `actor` and "movie" as `media type`.

The standard approach to slot tagging involves training a sequence model such as a conditional random field (CRF) on manually annotated data. An obvious limitation of this approach is that it relies on fully labeled data, which is both difficult to adapt and changing tasks and schemas. Certain films, songs, and books become more or less popular over time, and the performance of models trained on outdated data will degrade. If not updated, models trained on live data feeds such as movies, songs and books become obsolete over time and their accuracy will degrade. In order to achieve high accuracy continuously data and even model schemas have to be refreshed on a regular basis.

To remedy this limitation, we propose a weakly supervised framework that utilizes the information available in web click logs. A web click log is a mapping from a user query to URL link. For example, users issuing queries about movies tend to click on links from the IMDB.com or rottentomatoes.com, which provide rich structured data for entities such as title of the movie ("The Matrix"), the director ("The Wachowski Brothers"), and the release date ("1999"). Web click logs present an opportunity to learn semantic tagging models from large-scale and naturally occurring user interaction data (Volkova et al., 2013).

While some previous works (Li et al., 2009) have applied a similar strategy to incorporate click logs in slot tagging, they do not employ recent advances in machine learning to effectively leverage the incomplete annotations. In this paper, we pursue and extend learning from partially labeled sequences, in particular the approach of Täckström et al. (2013).

Instead of projecting labels from a high-resource to a low-resource languages via parallel text and word alignment, we project annotations from structured data found in click logs. This can be seen as a benefit since typically a much larger volume of click log data is available than parallel text for low-resource languages.

We also extend the constrained lattice training method of Täckström et al. (2013) from linear CRFs to non-linear CRFs. We propose a perceptron training method for hidden unit CRFs (Maaten et al., 2011) that allows us to train with partially labeled sequences. We show that combined with a novel pretraining methodology that leverages large quantities of unlabeled data, this training method achieves significant improvements over several strong baselines.

2 Model definitions and training methods

In this section, we describe the two sequence models in our experiments: a conditional random field (CRF) of Lafferty et al. (2001) and a hidden unit CRF (HUCRF) of Maaten et al. (2011). Note that since we only have partially labeled sequences, we need a technique to learn from incomplete data. For a CRF, we follow a variant of the training method of Täckström et al. (2013). In addition, we make a novel extension of their method to train a HUCRF from partially labeled sequences. The resulting perceptron-style algorithm (Figure 2) is simple but effective. Furthermore, we propose an initialization scheme that naturally leverages unlabeled data for training a HUCRF.

2.1 Partially Observed CRF

A first-order CRF parametrized by $\theta \in \mathbb{R}^d$ defines a conditional probability of a label sequence $y = y_1 \ldots y_n$ given an observation sequence $x = x_1 \ldots x_n$ as follows:

$$p_\theta(y|x) = \frac{\exp(\theta^\top \Phi(x, y))}{\sum_{y' \in \mathcal{Y}(x)} \exp(\theta^\top \Phi(x, y'))}$$

where $\mathcal{Y}(x)$ is the set of all possible label sequences for x and $\Phi(x, y) \in \mathbb{R}^d$ is a global feature function that decomposes into local feature functions $\Phi(x, y) = \sum_{j=1}^n \phi(x, j, y_{j-1}, y_j)$ by the first-order Markovian assumption. Given fully labeled sequences $\{(x^{(i)}, y^{(i)})\}_{i=1}^N$, the standard training method is to find θ that maximizes the log likelihood of the label sequences under the model with l_2-regularization:

$$\theta^* = \arg\max_{\theta \in \mathbb{R}^d} \sum_{i=1}^N \log p_\theta(y^{(i)}|x^{(i)}) - \frac{\lambda}{2} ||\theta||^2$$

Unfortunately, in our problem we do not have fully labeled sequences. Instead, for each token x_j in sequence $x_1 \ldots x_n$ we have the following two sources of label information:

- A set of allowed label types $\mathcal{Y}(x_j)$. (Label dictionary)
- A label $\tilde{y}_j$ transferred from a source data. (Optional: transferred label)

Täckström et al. (2013) propose a different objective that allows training a CRF in this scenario. To this end, they define a constrained *lattice* $\mathcal{Y}(x, \tilde{y}) = \mathcal{Y}(x_1, \tilde{y}_1) \times \ldots \times \mathcal{Y}(x_n, \tilde{y}_n)$ where at each position j a set of allowed label types is given as:

$$\mathcal{Y}(x_j, \tilde{y}_j) = \begin{cases} \{\tilde{y}_j\} & \text{if } \tilde{y}_j \text{ is given} \\ \mathcal{Y}(x_j) & \text{otherwise} \end{cases}$$

In addition to these existing constraints, we introduce constraints on the *label structure*. In our segmentation problem, labels are structured (e.g., some label types cannot follow certain others). We can easily incorporate this restriction by disallowing invalid label types as a post-processing step of the form:

$$\mathcal{Y}(x_j, \tilde{y}_j) \leftarrow \mathcal{Y}(x_j, \tilde{y}_j) \cap \overline{\mathcal{Y}}(x_{j-1}, \tilde{y}_{j-1})$$

where $\overline{\mathcal{Y}}(x_{j-1}, \tilde{y}_{j-1})$ is the set of valid label types that can follow $\mathcal{Y}(x_{j-1}, \tilde{y}_{j-1})$.

Täckström et al. (2013) define a conditional probability over label lattices for a given observation sequence x:

$$p_\theta(\mathcal{Y}(x, \tilde{y})|x) = \sum_{y \in \mathcal{Y}(x, \tilde{y})} p_\theta(y|x)$$

Given a label dictionary $\mathcal{Y}(x_j)$ for every token type x_j and training sequences $\{(x^{(i)}, \tilde{y}^{(i)})\}_{i=1}^N$ where $\tilde{y}^{(i)}$ is (possibly non-existent) transferred labels for

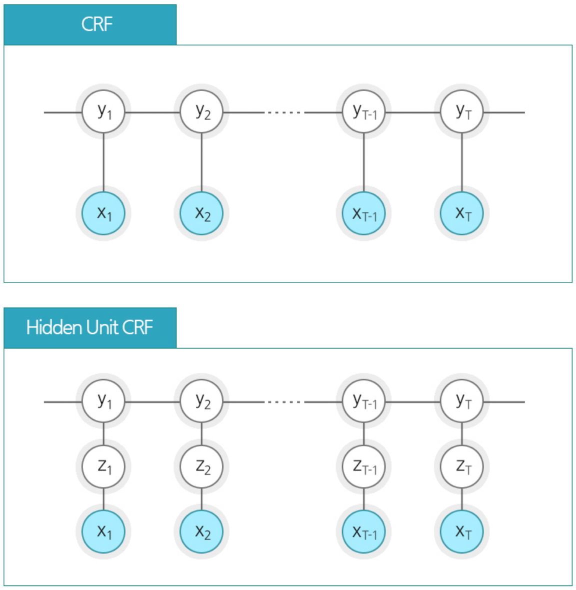


Figure 1: Illustration of CRFs and hidden unit CRFs

$x^{(i)}$ and, the new training method is to find θ that maximizes the log likelihood of the label lattices:

$$\theta^* = \arg\max_{\theta \in \mathbb{R}^d} \sum_{i=1}^{N} \log p_\theta(\mathcal{Y}(x^{(i)}, \tilde{y}^{(i)})|x^{(i)}) - \frac{\lambda}{2} ||\theta||^2$$

Since this objective is non-convex, we find a local optimum with a gradient-based algorithm. The gradient of this objective at each example $(x^{(i)}, \tilde{y}^{(i)})$ takes an intuitive form:

$$\begin{aligned} &\frac{\partial}{\partial\theta} \log p_\theta(\mathcal{Y}(x^{(i)}, \tilde{y}^{(i)})|x^{(i)}) - \frac{\lambda}{2} ||\theta||^2 \\ &= \sum_{y \in \mathcal{Y}(x^{(i)}, \tilde{y})} p_\theta(y|x^{(i)}) \Phi(x^{(i)}, y) \\ &\quad - \sum_{y \in \mathcal{Y}(x^{(i)})} p_\theta(y|x^{(i)}) \Phi(x^{(i)}, y) - \lambda\theta \end{aligned}$$

This is the same as the standard CRF training except the first term where the gold features $\Phi(x^{(i)}, y^{(i)})$ are replaced by the expected value of features in the constrained lattice $\mathcal{Y}(x^{(i)}, \tilde{y})$.

2.2 Partially Observed HUCRF

While effective, a CRF is still a linear model. To see if we can benefit from nonlinearity, we use a HUCRF (Maaten et al., 2011): a CRF that introduces a layer of binary-valued hidden units $z = z_1 \dots z_n \in \{0, 1\}$ for each pair of label sequence $y = y_1 \dots y_n$ and observation sequence $x = x_1 \dots x_n$. A HUCRF parametrized by $\theta \in \mathbb{R}^d$ and $\gamma \in \mathbb{R}^{d'}$ defines a joint probability of y and z conditioned on x as follows:

$$p_{\theta,\gamma}(y, z|x) = \frac{\exp(\theta^\top \Phi(x, z) + \gamma^\top \Psi(z, y))}{\sum_{\substack{z' \in \{0,1\}^n \\ y' \in \mathcal{Y}(x, z')}} \exp(\theta^\top \Phi(x, z') + \gamma^\top \Psi(z', y'))}$$

where $\mathcal{Y}(x, z)$ is the set of all possible label sequences for x and z, and $\Phi(x, z) \in \mathbb{R}^d$ and $\Psi(z, y) \in \mathbb{R}^{d'}$ are global feature functions that decompose into local feature functions:

$$\Phi(x, z) = \sum_{j=1}^{n} \phi(x, j, z_j)$$

$$\Psi(z, y) = \sum_{j=1}^{n} \psi(z_j, y_{j-1}, y_j)$$

In other words, it forces the interaction between the observations and the labels at each position j to go through a latent variable z_j: see Figure 1 for illustration. Then the probability of labels y is given by marginalizing over the hidden units,

$$p_{\theta,\gamma}(y|x) = \sum_{z \in \{0,1\}^n} p_{\theta,\gamma}(y, z|x)$$

As in restricted Boltzmann machines (Larochelle and Bengio, 2008), hidden units are conditionally independent given observations and labels. This allows for efficient inference with HUCRFs despite their richness (see Maaten et al. (2011) for details).

2.2.1 Training with partially labeled sequences

We extend the perceptron training method of Maaten et al. (2011) to train a HUCRF from partially labeled sequences. This can be viewed as a modification of the constrained lattice training method of Täckström et al. (2013) for HUCRFs.

A sketch of our training algorithm is shown in Figure 2. At each example, we predict the most likely label sequence with the current parameters. If this sequence does not violate the given constrained lattice, we make no updates. If it does, we predict the most likely label sequence within the con-

```
Input: constrained lattices $\{(x^{(i)}, \tilde{y}^{(i)})\}_{i=1}^{N}$, step size $\eta$
Output: HUCRF parameters $\Theta := \{\theta, \gamma\}$
  1. Initialize $\Theta$ randomly.
  2. Repeatedly select $i \in \{1 \dots N\}$ at random:
     (a) $y^* \leftarrow \arg\max_{y \in \mathcal{Y}(x^{(i)})} p_\Theta(y|x^{(i)})$
     (b) If $y^* \notin \mathcal{Y}(x^{(i)}, \tilde{y}^{(i)})$:
         i. $y^+ \leftarrow \arg\max_{y \in \mathcal{Y}(x^{(i)}, \tilde{y}^{(i)})} p_\Theta(y|x^{(i)})$
         ii. Make parameter updates:
             $$\Theta \leftarrow \Theta + \eta \times \frac{\partial}{\partial \Theta} \Big( p_\Theta(y^+, z^+|x^{(i)}) - p_\Theta(y^*, z^*|x^{(i)}) \Big)$$
             where the following hidden units are computed in closed-form (see Gelfand et al. (2010)):
             $$z^+ := \arg\max_z p_\Theta(z|x^{(i)}, y^+)$$
             $$z^* := \arg\max_z p_\Theta(z|x^{(i)}, y^*)$$
```

Figure 2: A sketch of the perceptron training algorithm for a partially observed hidden unit CRF.

strained lattice. We treat this as the gold label sequence, and perform the perceptron updates accordingly (Gelfand et al., 2010). Even though this training algorithm is quite simple, we demonstrate its effectiveness in our experiments.

2.2.2 Initialization from unlabeled data

Rather than initializing the model parameters randomly, we propose an effective initialization scheme (in a similar spirit to the pre-training methods in neural networks) that naturally leverages unlabeled data.

First, we cluster observation types in unlabeled data and treat the clusters as labels. Then we train a fully supervised HUCRF on this clustered data to learn parameters θ for the interaction between observations and hidden units $\Phi(x, z)$ and γ for the interaction between hidden units and labels $\Phi(z, y)$. Finally, for task/domain specific training, we discard γ and use the learned θ to initialize the algorithm in Figure 2. We hypothesize that if the clusters are non-trivially correlated to the actual labels, we can capture the interaction between observations and hidden units in a meaningful way.

3 Mining Click Log Data

We propose using search click logs which consist of queries and their corresponding web documents. Clicks are an implicit signal for related entities and information in the searched document. In this work, we will assume that the web document is *structured* and generated from an underlying database. Due to the structured nature of the web, this is not an unrealistic assumption (see Adamic and Huberman (2002) for discussion). Such structural regularities make obtaining annotated queries for learning a semantic slot tagger almost cost-free.

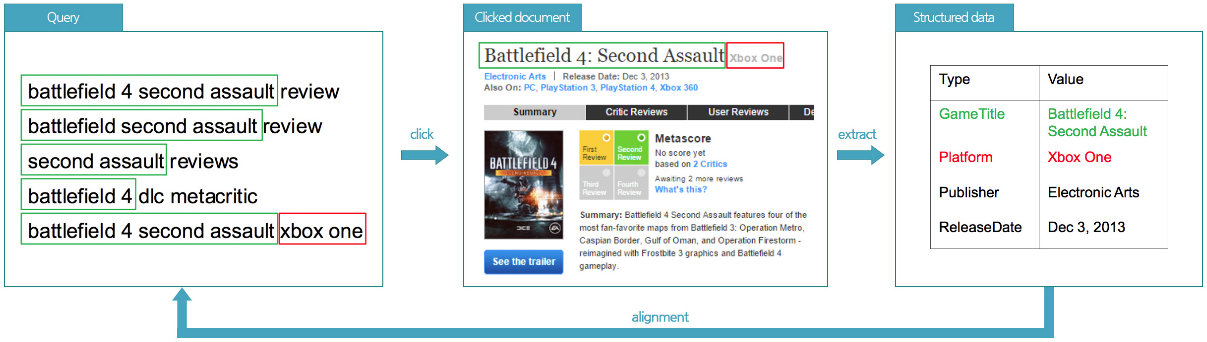
As an illustration of how to project annotation, consider Figure 3, where we present an example taken from queries about video games. In the figure, the user queries are connected to a structured document via a click log, and then the document is parsed and stored in a structured format. Then annotation types are projected to linked queries through structural alignment. In the following subsections we describe each step in our log mining approach in detail.

3.1 Click Logs

Web search engines keep a record of search queries, clicked document and URLs which reveal the user behavior. Such records are proven to be useful in improving the quality of web search. We focus on utilizing query-to-URL click logs that are essentially a mapping from queries to structured web documents. In this work, we use a year's worth of query logs (from July 2013 to June 2014) at a commercial search engine. We applied a simple URL normalization procedure to our log data including trimming and removal of prefixes, e.g. “www”.

3.2 Parsing Structured Web Document

A simple wrapper induction algorithm described in Kushmerick (1997) is applied for parsing web documents. Although it involves manually engineering a rule-based parser and is therefore website-specific, a single wrapper often generates large amounts of data for large structured websites, for example IMDB. Furthermore, it is very scalable to large quantities of data, and the cost of writing such a rule-based sys-



Summary: Battlefield 4 Second Assault features four of the most fan-favorite maps from Battlefield 3: Operation Metro, Caspian Border, Gulf of Oman, and Operation Firestorm - reimagined with Frostbite 3 graphics and Battlefield 4 gameplay.

Type	Value
GameTitle	Battlefield 4: Second Assault
Platform	Xbox One
Publisher	Electronic Arts
ReleaseDate	Dec 3, 2013

Figure 3: An example illustrating annotation projection via click-log and wrapper induction.

tem is typically much lower than the annotation cost of queries.

Figure 4 shows the statistics of parsed web documents on 24 domains with approximately 500 template rules. One of the chosen domains in our experiment, Music, has over 130 million documents parsed by our approach.

3.3 Annotation Projection via Structural Alignment

We now turn to the annotation projection step where structural alignment is used to transfer type annotation from structured data to queries. Note that this is different from the word-based or phrase-based alignment scenario in machine translation since we need to align a word sequence to a type-value pair.

Let us assume that we are given the user query as a word sequence, $w = w_1, w_2, \ldots, w_n$ and a set of structured data, $s = \{s_1, s_2, \ldots, s_m\}$, where s_i is a pair of slot-type and value. We define a measurement of dissimilarity between word tokens and slots, $dist(w_i, s_j) = 1 - sim(w_i, s_j)$ where $sim(\cdot, \cdot)$ is cosine similarity over character trigrams of w_i and s_j. Next we construct a n-by-n score matrix S of which element is $\max_j dist(w_{t' \ldots t}, s_j)$ meaning that a score of the most similar type-value s_j and a segment $\{t' \ldots t\}$ where $1 \leq t' < t \leq n$. Finally, given this approximate score matrix S, we use a dynamic programming algorithm to find the optimal segments to minimize the objective function:

$$T(t) = \min_{t' < t} T(t') S(t', t).$$

Our approach results in a large amount of high-quality partially-labeled data: 314K, 1.2M, and 1.1M queries for the Game, Movie and Music domain, respectively.

4 Experiments

To test the effectiveness of our approach, we perform experiments on a suite of three entertainment domains for slot tagging: queries about movies, music, and games. For each domain, we have two types of data: engineered data and log data. *Engineered data* is a set of synthetic queries to mimic the behavior of users. This data is created during development at which time no log data is available. *Log data* is a set of queries created by actual users using deployed spoken dialogue systems: thus it is directly transcribed from users' voice commands with automatic speech recognition (ASR). In general we found log data to be fairly noisy, containing many ASR and grammatical errors, whereas engineered data consisted of clean, well-formed text.

Not surprisingly, synthetic queries in engineered data are not necessarily representative of real queries in log data since it is difficult to accurately simulate what users' queries will be before a fully functioning system is available and real user data can be gathered. Hence this setting can greatly benefit from weakly-supervised learning methods such as ours since it is critical to learn from new incoming log data. We use search engine log data to project lattice constraints for weakly supervised learning.

In this setup, a user issues a natural language query to retrieve movies, music titles, games and/or information there of. For instance, a user could say

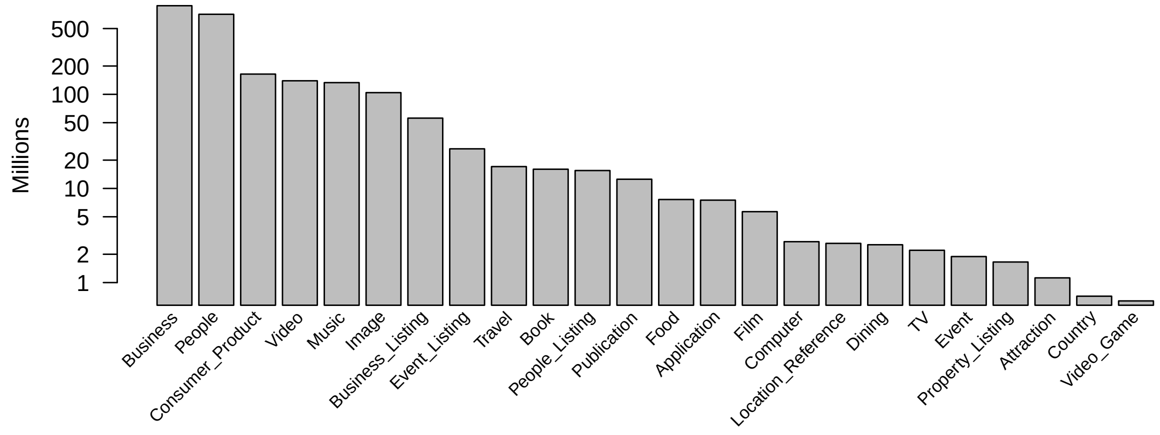


Figure 4: Statistics of structured web documents. The vertical axis shows the number of documents (in millions); the horizontal axis shows the web domain types.

"play the latest batman movie" or "find beyonce's music". Our slot sequence tagger is trained with variants of CRF using lexical features, gazetteers, Brown clusters and context words. The domains consist of 35 slot types for movies, 25 for music and 24 for games. Slot types correspond to both named entities (e.g., game name, music title, movie name) as well as more general categories (genre, media type, description). Table 1 shows the size of the datasets used in our experiments.

Domains	Training	Test
games	32017	5508
movies	48173	7074
music	46377	8890

Table 1: Labeled data set size for games, movies and music domains partitioned into training and test set.

Domains	Engineered	Log	Diff.
games	89.63	68.58	21.05
movies	88.67	74.21	14.45
music	88.77	37.13	51.64
AVG.	89.02	59.97	29.05

Table 2: The difference in F1 performance of CRF models trained only on engineered data but tested on both engineered and log data.

4.1 Discrepancy between Engineered Data and Log Data

To empirically highlight the need for learning from real user queries, we first train a standard CRF on the (fully labeled) engineered data and test it on the log data. We have manually annotated some log data for evaluation purposes. For features in the CRF, we use n-grams, gazetteer, and clusters. The clusters were induced from a large body of unlabeled data which consist of log data and click log data. Table 2 shows the F1 scores in this experiment. They indicate that a model fully supervised with engineered data performs very poorly on log data. The difference between the scores within engineered data and the scores in log data is very large (29.05 absolute F1).

4.2 Experiments with CRF Variants

Our main contribution is to leverage search log data to improve slot tagging in spoken dialogue systems. In this section, we assume that we have no log data in training slot taggers.[1]

For parameter estimation, both CRFs and POCRFs employ L-BFGS, while POHUCRF uses

[1] In practice, this assumption is not necessarily true because a deployed system can benefit from actual user logs. However, this controlled setting allows us to show the benefits of employing web search click log data.

Domains	games	music	movies	AVG.
CRF	74.21	37.13	68.58	59.97
POCRF	77.23	44.55	76.89	66.22
POHCRF	78.93	46.81	76.46	67.40
POHCRF+	**79.28**	**47.35**	**78.33**	**68.32**

Table 3: The F1 performance of variants of CRF across three domains, test on log data

average perceptron. We did not see a significant difference between perceptron and LBFGS in accuracy, but perceptron is faster and thus favorable for training complex HUCRF models. We used 100 as the maximum iteration count and 1.0 for the L2 regularization parameter. The number of hidden variables per token is set to 300. The same features described in the previous section are used here.

We perform experiments with the following CRF variants (see Section 2):

- CRF: A fully supervised linear-chain CRF trained with manually labeled engineered samples.

- POCRF: A partially observed CRF of Täckström et al. (2013) trained with both manually labeled engineered samples and click logs.

- POHUCRF: A partially observed hidden unit CRF (Figure 2) trained with both manually labeled engineered samples and click logs.

- POHUCRF+: POHUCRF with pre-training.

Table 3 summarizes the performance of these CRF variants. All results were tested on log data only. A standard CRF without click log data yields 59.97% of F1 on average. By using click log data, POCRF consistently improves F1 scores across domains, resulting into 66.22% F1 measure. Our model POHUCRF achieves extra gains on games and music, achieving 67.4% F1 measure on average. Finally, the pre-training approach yields significant additional gains across all domains, achieving 68.32% average performance. Overall we achieve a relative error reduction of about 21% over vanilla CRFs.

Domain	CRF	HUCRF	HUCRF+
alarm	91.79	91.79	**91.96**
calendar	87.60	87.65	**88.21**
communication	91.84	92.49	**92.80**
note	87.72	88.48	**88.72**
ondevice	89.37	90.14	**90.64**
places	88.02	88.64	**88.99**
reminder	87.72	89.21	**89.72**
weather	96.93	97.38	**97.63**
AVG.	90.12	90.75	**91.08**

Table 4: Performance comparison between HUCRF and HUCRF with pre-training.

4.3 Weakly-Supervised Learning without Projected Annotations via Pre-Training

We also present experiments within Cortana personal assistant domain where the click log data is not available. The amount of training data we used was from 50K to 100K across different domains and the test data was from 5k to 10k. In addition, the unlabeled log data were used and their amount was from 100k to 200k.

In this scenario, we have access to both engineered and log data to train a model. However, we do not have access to web search click log data. The goal of these experiments is to show the effectiveness of the HUCRF and pre-training method in the absence of weakly supervised labels projected via click logs. Table 4 shows a series of experiments on eight domains.

For all domains other than alarm, using non-linear CRF (HUCRF) improve performance from 90.12% to 90.75% on average. Initializing HUCRF with pretraining (HUCRF+) boosts the performance up to 91.08%, corresponding to a 10% decrease in error relative to a original CRF. Notably in the weather and reminder domains, we have relative error reduction of 23 and 16%, respectively. We speculate that pretraining is helpful because it provides better initialization for training HUCRF: initialization is important since the training objective of HUCRF is non-convex.

In general, we find that HUCRF delivers better performance than standard CRF: when the training procedure is initialized with pretraining (HUCRF+), it improves further.

5 Related Work

Previous works have explored weakly supervised slot tagging using aligned labels from a database as constraints. Wu and Weld (2007) train a CRF on heuristically annotated Wikipedia articles with relations mentioned in their structured infobox data. Li et al. (2009) applied a similar strategy incorporating structured data projected through click-log data as both heuristic labels and additional features. Knowledge graphs and search logs have been also considered as extra resources (Liu et al., 2013; El-Kahky et al., 2014; Anastasakos et al., 2014; Sarikaya et al., 2014; Marin et al., 2014).

Distant supervision methods (Mintz et al., 2009; Riedel et al., 2010; Surdeanu et al., 2012; Agichtein and Gravano, 2000) learn to extract relations from text using weak supervision from related structured data sources such as Freebase or Wikipedia. These approaches rely on named entity recognition as a pre-processing step to identify text spans corresponding to candidate slot values. In contrast, our approach jointly segments and predicts slots.

Works on weakly supervised POS tagging are also closely related to ours (Toutanova and Johnson, 2007; Haghighi and Klein, 2006). Täckström et al. (2013) investigate weakly supervised POS tagging in low-resource languages, combining dictionary constraints and labels projected across languages via parallel corpora and automatic alignment. Our work can be seen as an extension of their approach to the structured-data projection setup presented by Li et al. (2009). A notable component of our extension is that we introduce a training algorithm for learning a hidden unit CRF of Maaten et al. (2011) from partially labeled sequences. This model has a set of binary latent variables that introduce non-linearity by mediating between observations and labels.

6 Conclusions

In this paper, we applied weakly-supervised learning approach for slot tagging, projecting annotations from structured data to user queries by leveraging click log data. We extended the Täckström et al. (2013) model to nonlinear CRFs by introducing latent variables and applying a novel pre-training methodology. The proposed techniques provide an effective way to leverage incomplete and ambiguous annotations from large amounts of naturally occurring click log data. All of our improvements taken together result in a 21% error reduction over vanilla CRFs trained on engineered data used during system development.

References

Lada A Adamic and Bernardo A Huberman. 2002. Zipfs law and the internet. *Glottometrics*, 3(1):143–150.

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*.

Tasos Anastasakos, Young-Bum Kim, and Anoop Deoras. 2014. Task specific continuous word representations for mono and multi-lingual spoken language understanding. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 3246–3250. IEEE.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 600–609. Association for Computational Linguistics.

Ali El-Kahky, Xiaohu Liu, Ruhi Sarikaya, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2014. Extending domain coverage of language understanding systems via intent transfer between domains using knowledge graphs and search query click logs. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 4067–4071. IEEE.

Andrew Gelfand, Yutian Chen, Laurens Maaten, and Max Welling. 2010. On herding and the perceptron cycling theorem. In *Advances in Neural Information Processing Systems*, pages 694–702.

Aria Haghighi and Dan Klein. 2006. Prototype-driven learning for sequence models. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*.

Nicholas Kushmerick. 1997. *Wrapper induction for information extraction*. Ph.D. thesis, University of Washington.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289.

Hugo Larochelle and Yoshua Bengio. 2008. Classification using discriminative restricted boltzmann machines. In *Proceedings of the 25th international conference on Machine learning*.

Xiao Li, Ye-Yi Wang, and Alex Acero. 2009. Extracting structured information from user queries with semi-supervised conditional random fields. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*.

Shen Li, Joao V Graça, and Ben Taskar. 2012. Wiki-ly supervised part-of-speech tagging. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1389–1398. Association for Computational Linguistics.

Xiaohu Liu, Ruhi Sarikaya, Chris Brockett, Chris Quirk, William B Dolan, and Bill Dolan. 2013. Paraphrase features to improve natural language understanding. In *INTERSPEECH*, pages 3776–3779.

Laurens Maaten, Max Welling, and Lawrence K Saul. 2011. Hidden-unit conditional random fields. In *International Conference on Artificial Intelligence and Statistics*.

Alex Marin, Roman Holenstein, Ruhi Sarikaya, and Mari Ostendorf. 2014. Learning phrase patterns for text classification using a knowledge graph and unlabeled data. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*.

Ruhi Sarikaya, Asli Celikyilmaz, Anoop Deoras, and Minwoo Jeong. 2014. Shrinkage based features for slot tagging with conditional random fields. In *Proc. of Interspeech*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging.

Kristina Toutanova and Mark Johnson. 2007. A bayesian lda-based model for semi-supervised part-of-speech tagging. In *Advances in Neural Information Processing Systems*, pages 1521–1528.

Svitlana Volkova, Pallavi Choudhury, Chris Quirk, Bill Dolan, and Luke S Zettlemoyer. 2013. Lightly supervised learning of procedural dialog systems. In *ACL*.

Fei Wu and Daniel S Weld. 2007. Autonomously semantifying wikipedia. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*.