Semi-Supervised Semantic Tagging of Conversational Understanding using Markov Topic Regression

Asli Celikyilmaz
Microsoft
Mountain View, CA, USA
asli@ieee.org

Dilek Hakkani-Tur, Gokhan Tur
Microsoft Research
Mountain View, CA, USA
dilek@ieee.org
gokhan.tur@ieee.org

Ruhi Sarikaya
Microsoft
Redmond, WA, USA
rusarika@microsoft.com

Abstract

Finding concepts in natural language utterances is a challenging task, especially given the scarcity of labeled data for learning semantic ambiguity. Furthermore, data mismatch issues, which arise when the expected test (target) data does not exactly match the training data, aggravate this scarcity problem. To deal with these issues, we describe an efficient semi-supervised learning (SSL) approach which has two components: (*i*) *Markov Topic Regression* is a new probabilistic model to cluster words into semantic tags (concepts). It can efficiently handle semantic ambiguity by extending standard topic models with two new features. First, it encodes word n-gram features from labeled source and unlabeled target data. Second, by going beyond a bag-of-words approach, it takes into account the *inherent sequential nature* of utterances to learn semantic classes based on context. (*ii*) *Retrospective Learner* is a new learning technique that adapts to the unlabeled target data. Our new SSL approach improves semantic tagging performance by 3% absolute over the baseline models, and also compares favorably on semi-supervised syntactic tagging.

1 Introduction

Semantic tagging is used in natural language understanding (NLU) to recognize words of semantic importance in an utterance, such as entities. Typically, a semantic tagging model require large amount of domain specific data to achieve good performance (Tur and DeMori, 2011). This requires a tedious and time intensive data collection and labeling process. In the absence of large labeled training data, the tagging model can behave poorly on test data (target domain). This is usually caused by data mismatch issues and lack of coverage that arise when the target data does not match the training data.

To deal with these issues, we present a new semi-supervised learning (SSL) approach, which mainly has two components. It initially starts with training supervised Conditional Random Fields (CRF) (Lafferty et al., 2001) on the source training data which has been semantically tagged. Using the trained model, it decodes unlabeled dataset from the target domain. With the data mismatch issues in mind, to correct errors that the supervised model make on the target data, the SSL model leverages the additional information by way of a new clustering method. Our first contribution is a new probabilistic topic model, *Markov Topic Regression* (MTR), which uses rich features to capture the degree of association between words and semantic tags. First, it encodes the n-gram context features from the labeled source data and the unlabeled target data as prior information to learn semantic classes based on context. Thus, each latent semantic class corresponds to one of the semantic tags found in labeled data. MTR is not invariant to reshuffling of words due to its Markovian property; hence, word-topic assignments are also affected by the topics of the surrounding words. Because of these properties, MTR is less sensitive to the errors caused by the semantic ambiguities. Our SSL uses MTR to smooth the semantic tag posteriors on the unlabeled target data (decoded using the CRF model) and later obtains the best tag sequences. Using the labeled source and automati-

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 914–923,
Sofia, Bulgaria, August 4-9 2013. ©2013 Association for Computational Linguistics

cally labeled target data, it re-trains a new CRF-model.

Although our iterative SSL learning model can deal with the training and test data mismatch, it neglects the performance effects caused by adapting the source domain to the target domain. In fact, most SSL methods used for adaptation, e.g., (Zhu, 2005), (Daumé-III, 2010), (Subramanya et al., 2010), etc., do not emphasize this issue. With this in mind, we introduce a new iterative training algorithm, *Retrospective Learning*, as our second contribution. While retrospective learning iteratively trains CRF models with the automatically annotated target data (explained above), it keeps track of the errors of the previous iterations so as to carry the properties of both the source and target domains.

In short, through a series of experiments we show how MTR clustering provides additional information to SSL on the target domain utterances, and greatly impacts semantic tagging performance. Specifically, we analyze MTR's performance on two different types of semantic tags: named-entities and descriptive tags as shown in Table 1. Our experiments show that it is much harder to detect descriptive tags compared to named-entities.

Our SSL approach uses probabilistic clustering method tailored for tagging natural language utterances. To the best of our knowledge, our work is the first to explore the unlabeled data to iteratively adapt the semantic tagging models for target domains, preserving information from the previous iterations. With the hope of spurring related work in domains such as entity detection, syntactic tagging, etc., we extend the earlier work on SSL part-of-speech (POS) tagging and show in the experiments that our approach is not only useful for semantic tagging but also syntactic tagging.

The remainder of this paper is divided as follows: §2 gives background on SSL and semantic clustering methods, §3 describes our new clustering approach, §4 presents the new iterative learning, §5 presents our experimental results and §6 concludes our paper.

2 Related Work and Motivation

(I) Semi-Supervised Tagging. Supervised methods for semantic tagging in NLU require a large number of in-domain human-labeled utterances and gazetteers (movie, actor names, etc.), increasing the need for significant manual labor (Tur and DeMori, 2011). Recent work on similar tasks overcome these challenges using SSL methods as follows:

• Are there any [*comedies*] with [*Ryan Gosling*]? • How about [*oscar winning*] movies by [*James Cameron*]? • Find [*Woody Allen*] movies similar to [*Manhattan*].
[Named Entities] director: *James Cameron*, *Woody Allen*,... actor: *Ryan Gosling*, *Woody Allen*,... title: *Manhattan*, *Midnight in Paris*,...
[Descriptive Tags] restriction: *similar*, *suitable*, *free*,*rate*,... description: *oscar winning*, *new release*, *gardening*,... genre: *spooky*, *comedies*, *feel good*, *romance*,...

Table 1: Samples of semantically tagged utterances from movie domain, named-entities and descriptive tags.

• (Wang et al., 2009; Li et al., 2009; Li, 2010; Liu et al., 2011) investigate web query tagging using semi-supervised sequence models. They extract semantic lexicons from unlabeled web queries, to use as features. Our work differs from these, in that, rather than just detecting named-entities, our utterances include *descriptive* tags (see Table 1).

• Typically the source domain has different distribution than the target domain, due to topic shifts in time, newly introduced features (e.g., until recently online articles did not include facebook "*like*" feature.), etc. Adapting the source domain using unlabeled data is the key to achieving good performance across domains. Recent adaptation methods for SSL use: expectation minimization (Daumé-III, 2010) graph-based learning (Chapelle et al., 2006; Zhu, 2005), etc. In (Subramanya et al., 2010) an efficient iterative SSL method is described for syntactic tagging, using graph-based learning to smooth POS tag posteriors. However, (Reisinger and Mooney, 2011) argues that vector space models, such as graph-learning, may fail to capture the richness of word meaning, as similarity is not a globally consistent metric. Rather than graph-learning, we present a new SSL using a probabilistic model, MTR, to cluster words based on co-occurrence statistics.

• Most iterative SSL methods, do not keep track of the errors made, nor consider the divergence from the original model. (Lavoie et al., 2011) argues that iterative learning models should mitigate new errors made by the model at each iteration by

keeping the history of the prior predictions. This ensures that a penalty is paid for diverging from the previous model's predictions, which will be traded off against the benefit of reducing classification loss. We present a *retrospective* SSL for CRF, in that, the iterative learner keeps track of the errors of the previous iterations so as to carry the properties of both the source and target domains.

(II) Semantic Clustering. A common property of several context-based word clustering techniques, e.g., Brown clustering (Brown et al., 1992), Clustering by Committee (Pantel, 2003), etc., is that they mainly cluster based on local context such as nearby words. Standard topic models, such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003), use a *bag-of-words* approach, which disregards word order and clusters words together that appear in a similar global context. Such models have been effective in discovering lexicons in many NLP tasks, e.g., named-entity recognition (Guo et al., 2009), word-sense disambiguation (Boyd-Graber et al., 2007; Li et al., 2010), syntactic/semantic parsing (Griffiths et al., 2005; Singh et al., 2010), speaker identification (Nyugen et al., 2012), etc. Recent topic models consider word sequence information in documents (Griffiths et al., 2005; Moon et al., 2010). The Hidden Topic Markov Model (HTMM) by (Gruber et al., 2005), for instance, models sentences in documents as Markov chains, assuming all words in a sentence have the *same* topic. While MTR has a similar Markovian property, we encode features on words to allow each word in an utterance to sample from any of the given semantic tags, as in "what are $[\textit{scary}]_{\mathbf{genre}}$ movies by $[\textit{Hitchcock}]_{\mathbf{director}}$?".

In LDA, common words tend to dominate all topics causing related words to end up in different topics. In (Petterson et al., 2010), the vector-based features of words are used as prior information in LDA so that the words that are synonyms end up in same topic. Thus, we build a semantically rich topic model, MTR, using *word context* features as side information. Using a smoothing prior for each word-topic pair (instead of a constant β smoother), MTR assures that the words are distributed over topics based on how similar they are. (e.g., *"scary"* and *"spooky"*, which have similar context features, go into the same semantic tag, *"genre"*). Thus, to best of our knowledge, MTR is the first topic model to incorporate word features while considering the sequence of words.

3 Markov Topic Regression - MTR

3.1 Model and Abstractions

LDA assumes that the latent topics of documents are sampled independently from one of K topics. MTR breaks down this independence assumption by allowing Markov relations between the hidden tags to capture the relations between consecutive words (as sketched in Figure 1 and Algorithm 1).

(I) Semantic Tags (s_i): Each word w_i of a given utterance with N_j words, u_j=$\{w_i\}_{i=1}^{N_j} \in U$, j=1,..$|U|$, from a set of utterances U, is associated with a latent semantic tag (state) variable $s_i \in \mathcal{S}$, where $\mathcal{S}$ is the set of semantic tags. We assume a fixed K topics corresponding to semantic tags of labeled data. In a similar way to HTMM (Gruber et al., 2005) described for documents, MTR samples each s_i from a Markov chain that is specific to its utterance u_j. Each state s_i generates a word, w_i, based on the word-state co-occurrences. MTR allows for sampling of consecutive words from different tag clusters. The initial probabilities of the latent states are sampled from a Dirichlet distribution over state variables, θ_j, with α hyper-parameter for each u_j.

(II) Tag Transition Indicator ($\psi_{\mathbf{v}}$): Given utterance u_j, the decision to sample a w_i from a new topic is determined by an indicator variable, $c_{j,i}$, that is sampled from a *Binomial*($\psi_{v=w_i}$) distribution with a *Beta* conjugate prior. (There are v binomials for each vocabulary term.) $c_{j,i}$=1 suggests that a new state be sampled from K possible tags for the word w_i in u_j, and $c_{j,i}$=0 suggests that the state s_i of w_i should be the same as the previous word's latent state s_{i-1}. The first position of the sequence is sampled from a new state, hence $c_{j,i=1}$=1.

(III) Tag Transition Base Measure ($\boldsymbol{\eta}$): Prior probability of a word given a tag should increase the chances of sampling words from the correct semantic tag. MTR constrains the generation of a tag s_i given the previous tag s_{i-1} and the current w_i based on $c_{j,i}$ by using a vocabulary specific Beta prior, $\psi_v \sim \textit{Beta}(\eta_v)$ [1], on each word in vocabulary $w_{v=1,..V}$. We inject the prior information on semantic tags to define values of the base measure η_v using external knowledge from two sources:

(a) **Entity Priors (η_S):** Prior probability on named-entities and descriptive tags denoted as

[1] For each beta distribution we use symmetric $\textit{Beta}(\eta_v)$=$\textit{Beta}(\alpha$=η_v, β=$\eta_v)$.

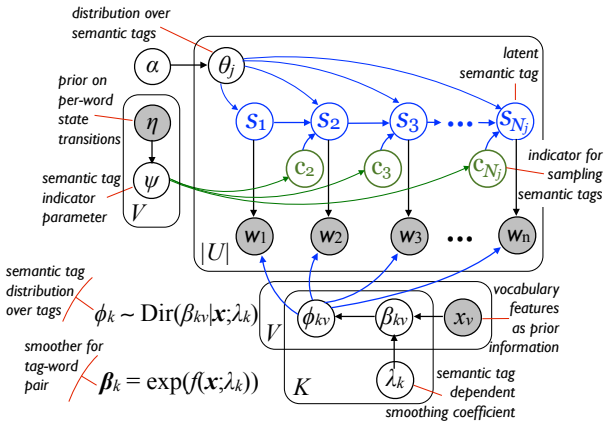

Figure 1: The graph representation of the Markov Topic Regression (MTR). To demonstrate hidden state Markov Chain, the generation of each word is explicitly shown (inside of the plate).

Algorithm 1 Markov Topic Regression

for each semantic tag topic s_k, $k \leftarrow 1,...,K$ **do**
  – draw a topic mixture $\phi_k \sim Dir(\boldsymbol{\beta}_k|\lambda_k,\mathbf{x})$,
  – let $\boldsymbol{\beta}_k$=exp($f(\mathbf{x};\lambda_k)$); $\mathbf{x}$=$\{x_v\}_{v=1}^{V_l}$, $\boldsymbol{\beta}_k \in \mathcal{R}^{V_l}$
for each word w_v in vocabulary $v \leftarrow 1,...,V$ **do**
  – draw a tag indicator mixture $\psi_v \sim Beta(\eta)$,
for each utterance $j \leftarrow 1,...,|U|$ **do**
  –draw transition distribution $\theta_j^s \sim Dir(\alpha)$
  over states s_i and set c_{j1}=1.
  –**for** words w_i in u_j, $i \leftarrow 1,...,N_j$ **do**
    ◇ if i >1, toss a coin $c_{j,i} \sim Binomial(\psi_{w_i})$.
    ◇ If $c_{j,i}$=1, draw $s_i \sim Multi(\theta_j^{s_{i,s_{i-1}}})^\dagger$
    otherwise s_i=s_{i-1}.
    ◇ Sample $w_i \sim$Multi(ϕ_{s_i}).

† Markov assumption over utterance words is used (See Eq.(4)).

η_S=$p(s_i|s_{i-1},w_i$=$v,w_{i-1})$. We use web sources (wiki pages on movies and urls such as imdb.com) and labeled training data to extract entity lists that correspond to the semantic tags of our domains. We keep the frequency of each n-gram to convert into (empirical) prior probability distribution.

(*b*) **Language Model Prior** (η_W)**:** Probabilities on word transitions denoted as η_W=$p(w_i$=$v|w_{i-1})$. We built a language model using SRILM (Stolcke, 2002) on the domain specific sources such as top wiki pages and blogs on online movie reviews, etc., to obtain the probabilities of domain-specific n-grams, up to 3-grams. The observed priors, η_S and η_W, are used for calculating the base measure η for each vocabulary w_v as:

$$\eta_v^{s_i|s_{i-1}} = \begin{cases} \eta_S^{s_i|s_{i-1},w_i=v}, & \text{if } \eta_S^{s_i|s_{i-1},w_i=v} \text{ exists,} \\ \eta_W^{w_i=v,w_{i-1}}, & \text{otherwise} \end{cases} \quad (1)$$

In Eq.(1), we assume that the prior on the semantic tags, η_S, is more indicative of the decision for sampling a w_i from a new tag compared to language model posteriors on word sequences, η_W. Here we represent the base-measure (hyper-parameter) of the semantic tag indicator variable, which is not to be confused with a probability measure [2]

We update the indicator parameter via mean criteria, $\psi_{v=w_i}$=$\sum_{i,j=1}^{K} \eta_{v=w_i}^{s_i|s_j}/(K^2)$. If no prior on a specific word exists, a default value is used for base measure, η_v=0.01.

(IV) Topic-Word Distribution Priors ($\boldsymbol{\beta}_k$): Different from (Mimno et al., 2008), which uses asymmetric hyper-parameters on document-topic distributions, in MTR, we learn the asymmetric hyper-parameters of the semantic tag-word distributions. We use blocked Gibbs sampling, in which the topic assignments s_k and hyper-parameters $\{\boldsymbol{\beta}_k\}_{k=1}^K$ are alternately sampled at each Gibbs sampling lag period g given all other variables. We impose the prior knowledge on naturally related words, such that if two words "*funny*" and "*hilarious*" indicate the same given "*genre*" class, then their latent tag distributions should also be similar. We enforce this on smoothing parameter $\beta_{k,v}$, e.g., $\beta_{k,'funny'} \sim \beta_{k,'hilarious'}$ for a given tag k as follows:

At each g lag period of the Gibbs sampling, K log-linear models with parameters, $\lambda_k^{(g)} \in \mathcal{R}^M$, is trained to predict $\beta_{kv}^{(g)} \in \boldsymbol{\beta}_k$, for each w_v of a tag s_k:

$$\boldsymbol{\beta}_k^{(g)} = \exp(f(\boldsymbol{x}^l;\lambda_k^{(g)})) \quad (2)$$

where the log-linear function f is:

$$n_{kv}^{(g)} = f(x_v^l;\lambda_k^{(g)}) = \sum_m \lambda_{k,m}^{(g)} x_{v,m}^l \quad (3)$$

Here $\boldsymbol{x} \in \mathcal{R}^{V \times M}$ is the input matrix $\boldsymbol{x}$, wherein rows $x_v \in \mathcal{R}^M$ represents M-dimensional scalar vector of explanatory features on vocabulary words. We use the word-tag posterior probabilities obtained from a CRF sequence model trained on labeled utterances as features. The $\boldsymbol{x}$=$\{\boldsymbol{x}^l,\boldsymbol{x}^u\}$ has labeled (l) and unlabeled (u) parts. The labeled part contains V_l size vocabulary of which we know the semantic tags, $\boldsymbol{x}^l$=$\{(x_1^l,s_1),...,(x_{V_l}^l,s_{V_l})\}$. At the start of the Gibbs sampling, we designate the

[2]The base-measure used in Eq.(1) does not relate to a back-off model in LM sense. Here, instead of using a constant value for the hyper-parameters, we use probability scores that we obtain from LM.

K latent topics to the K semantic tags of our labeled data. Therefore, we assign labeled words to their designated topics. This way we use observed scalar counts of each labeled word v associated with its semantic tag k, $n^{(g)}_{kv}$, as the output label of its input vector, x^l_v; an indication of likelihood of words getting sampled from the corresponding semantic label s_k. Since the impact of the asymmetric prior is equivalent to adding pseudo-counts to the sufficient statistics of the semantic tag to which the word belongs, we predict the pseudo-counts $\beta^{(g)}_{kv}$ using the scalar counts of the labeled data, $n^{(g)}_{kv}$, based on the log-linear model in Eq. (2). At g=0, we use $\beta^{(0)}_{kv}$=2^8, if $x_v \in X^l$; otherwise $\beta^{(0)}_{kv}$=2^{-2}, commonly used values for large and small β. Note that larger β-values indicate correlation between the word and the topic.

3.2 Collapsed Sampler

The goal of MTR is to infer the degree of relationship between a word v and each semantic tag k, ϕ_{kv}. To perform inference we need two components:

- a sampler which can draw from conditional $P_{\texttt{MTR}}(s_{ji}{=}k|s_{ji-1}, s_{\setminus ji}, \alpha, \psi_i, \beta_{ji})$, when $c_{j,i}$=1, where s_{ji} and s_{ji-1} are the semantic tags of the current w_i=v of vocabulary v and previous word w_{i-1} in utterance u_j, and $s_{\setminus ji}$ are the semantic tag topics of all words except for w_i; and,
- an estimation procedure for (β_{kv}, λ_k) (see §3.1).

We integrate out the multinomial and binomial parameters of the model: utterance-tag distributions θ_j, binomial state transition indicator distribution per each word ψ_v, and ϕ_k for tag-word distributions. We use collapsed Gibbs sampling to reduce random components and model the posterior distribution by obtaining samples $(s_{ji}, c_{j,i})$ drawn from this distribution. Under the Markov assumption, for each word w_i=v in a given utterance u_j, if $c_{j,i}$=1, we sample a new tag s_i=k given the remaining tags and hyper-parameters $\boldsymbol{\beta}_k$, α, and $\eta^{s_i|s_{i-1}}_{w_i=v}$. Using the following parameters; $n^{(s_i)}_{ji}$, which is the number of words assigned to a semantic class s_i=k excluding case i, and $n^{(s_{i-1})}_{s_i}$ is the number of transitions from class s_{i-1} to s_i, where indicator $\mathbb{I}(s_{i-1}, s_i)$=1 if slot s_i=s_{i-1}, the update equation is formulated as follows:

$$p(s_{ji}=k|\mathbf{w}, s_{-ji}, \alpha, \eta^{s_i|s_{i-1}}_{w_i}, \boldsymbol{\beta}_k) \propto \frac{n^{(s_i)}_{ji}+\beta_{kw_i}}{n^{(k)}_{(.)}+\sum_v \beta_{kv}} * (n^{(s_{i-1})}_{s_i}+\alpha) * \frac{(n^{(s_i)}_{s_{i+1}}+\mathbb{I}(s_{i-1}, s_i)+\mathbb{I}(s_{i+1}, s_i)+\alpha)}{n^{(s_i)}_{(.)}+\mathbb{I}(s_{i-1}, k)+K\alpha} \quad (4)$$

4 Semi-Supervised Semantic Labeling

4.1 Semi Supervised Learning (SSL) with CRF

In (Subramanya et al., 2010), a new SSL method is described for adapting syntactic POS tagging of sentences in newswire articles along with search queries to a target domain of natural language (NL) questions. They decode unlabeled queries from target domain (*t*) using a CRF model trained on the POS-labeled newswire data (source domain (*o*)). The unlabeled POS tag posteriors are then smoothed using a graph-based learning algorithm. On graph, the similarities are defined over sequences by constructing the graph over *types*, word 3-grams, where *types* capture the local context of words. Since CRF tagger only uses local features of the input to score tag pairs, they try to capture all the context with the graph with additional context features on *types*. Later, using viterbi decoding, they select the 1-best POS tag sequence, $\boldsymbol{s}^*_j$ for each utterance u_j. Graph-based SSL defines a new CRF objective function:

$$\Lambda^{(t)}_{n+1} = \underset{\Lambda \in \mathcal{R}^K}{\operatorname{argmin}} \left\{ -\sum_{j=1:l} \log p(\boldsymbol{s}_j|u_j; \Lambda^{(t)}_n) + \mu \|\Lambda^{(t)}_n\|^2 \right\} - \left\{ \tau \sum_{j=l}^{l+u} \log p_n(\boldsymbol{s}^*_j|u_j; \Lambda^{(t)}_n) \right\} \quad (5)$$

The first bracket in Eq.(5) is the loss on the labeled data and $\mathcal{L}_2$ regularization on parameters, $\Lambda^{(t)}_n$, from *n*th iteration, same as standard CRF. The last term is the loss on unlabeled data from target domain with a hyper-parameter τ. They use a small value for τ to enable the new model to be as close as possible to the initial model trained on source data.

4.2 Retrospective Semi-Supervised CRF

We describe a *Retrospective* SSL (R-SSL) training with CRF (Algorithm 2), using MTR as a

smoothing model, instead of a graph-based model, as follows:

I. DECODING and **SMOOTHING**. The posterior probability of a tag s_{ji}=k given a word w_{ji} in unlabeled utterance u_j from target domain (t) $\hat{p}_n(j,i)$=$\hat{p}_n(s_{ji}$=k$|w_{ji};\Lambda_n^{(t)})$, is decoded using the n-th iteration CRF model. MTR uses the decoded probabilities as semantic tag prior features on vocabulary items. We generate a word-tag matrix of posteriors, $\boldsymbol{x}\in(0,1)^{V\times K}$, where K is the number of semantic tags and V is the vocabulary size from n-th iteration. Each row is a K dimensional vector of tag posterior probabilities x_v=$\{x_{v1},\ldots x_{vK}\}$ on the vocabulary term, w_v. The labeled rows $\boldsymbol{x}^l$ of the vocabulary matrix, $\boldsymbol{x}$=$\{\boldsymbol{x}^l,\boldsymbol{x}^u\}$, contain only $\{0,1\}$ values, indicating the word's observed semantic tags in the labeled data. Since a labeled term w_v can have different tags (e.g., "*clint eastwood*" may be tagged as *actor-name* and *director-name* in the training data), $\sum_k^K x_{vk}\geq 1$ holds. The $\boldsymbol{x}$ is used as the input matrix of the kth log-linear model (corresponding to kth semantic tag (topic)) to infer the β hyper-parameter of MTR in Eq. (2). MTR generates smoothed conditional probabilities ϕ_{kv} for each vocabulary term v given semantic tag k.

II. INTERPOLATION. For each word w_{ji}=v in unlabeled utterance u_j, we interpolate tag marginals from CRF and MTR for each semantic tag $s_{ji}=k$:

$$\hat{q}_n(s_{ji}|w_{ij};\Lambda_n^{(t)}) = \pi \overbrace{\hat{p}_n(s_{ji}|w_{ij};\Lambda_n^{(t)})}^{\text{CRF } posterior} + (1-\pi)\overbrace{\phi_{kv}}^{\text{MTR}} \quad (6)$$

III. VITERBI. Using viterbi decoding over the tag marginals, $\hat{q}_n(s_{ji}|w_{ij};\Lambda_n^{(t)})$, and transition probabilities obtained from the CRF model of n-th iteration, we get $\hat{p}_n(s_j^*|u_j;\Lambda_n^{(t)})$, the 1-best decode $\boldsymbol{s}_j^*$ of each unlabeled utterance $u_j\in\mathcal{U}_n^u$.

IV. RETROSPECTIVE SSL (R-SSL). After we decode the unlabeled data, we re-train a new CRF model at each iteration. Each iteration makes predictions on the semantic tags of unlabeled data with varying posterior probabilities. Motivated by (Lavoie et al., 2011), we want the loss function to have a dependency on the prior model predictions. Thus, R-SSL encodes the history of the prior predictions, as follows:

Algorithm 2 Retrospective Semi-Supervised CRF

Input: Labeled $\mathcal{U}^l$, and unlabeled $\mathcal{U}^u$ data.
Process: $\Lambda_n^{(o)}$=crf-train($\mathcal{U}_l$) at n=0, n=n+1 †.
While not converged
  $\hat{p}$=posterior-decode($\mathcal{U}_n^u$,$\Lambda_n^{(o)}$)
  ϕ=smooth-posteriors($\hat{p}$) using MTR,
  $\hat{q}$=interpolate-posteriors($\hat{p}$,ϕ),
  $\mathcal{U}_n^u$=viterbi-decode($\hat{q}$)
  $\Lambda_{n+1}^{(t)}$=crf-retrospective($\mathcal{U}^l$, $\mathcal{U}_n^u,\ldots,\mathcal{U}_1^u$,$\Lambda_n^{(t)}$)
† (n):iteration, (t):target, (o):source domains.

$$\Lambda_{n+1}^{(t)} = \underset{\Lambda\in\mathcal{R}^K}{\text{argmin}} \left\{-\sum_{j=1:l}\log p(\boldsymbol{s}_j|u_j;\Lambda_n^{(t)}) + \mu\|\Lambda_n^{(t)}\|^2\right\} \left\{-\sum_{j=1:(l+u)}\max\{0,\hat{p}_n^{**}\}\right\} \quad (7)$$

where, $\hat{p}_n^{**}$=$1-\log h_n(u_j)\hat{p}_n(s_j^*|u_j;\Lambda_n^{(t)})$. The first two terms are same as standard CRF. The last term ensures that the predictions of the current model have the same sign as the predictions of the previous models (using labeled and unlabeled data), denoted by a maximum margin hinge weight, $h_n(u_j)$=$\frac{1}{n-1}\sum_1^{n-1}\hat{p}_n(s_j^*|u_j;\Lambda_n^{(t)})$. It should also be noted that with MTR, the R-SSL learns the word-tag relations by using features that describe the words in context, eliminating the need for additional *type* representation of graph-based model. MTR provides a separate probability distribution θ_j over tags for each utterance j, implicitly allowing for the same word v in separate utterances to differ in tag posteriors ϕ_{kv}.

5 Experiments

5.1 Datasets and Tagsets

5.1.1 Semantic Tagging Datasets

We focus here on audiovisual media in the *movie* domain. The user is expected to interact by voice with a system than can perform a variety of tasks such as browsing, searching, querying information, etc. To build initial NLU models for such a dialog system, we used crowd-sourcing to collect and annotate utterances, which we consider our source domain. Given movie domain-specific tasks, we asked the crowd about how they would

interact with the media system as if they were talking to a person.

Our data from target domain is internally collected from real-use scenarios of our spoken dialog system. The transcribed text forms of these utterances are obtained from speech recognition engine. Although the crowd-sourced data is similar to target domain, in terms of pre-defined user intentions, the target domain contains more descriptive vocabulary, which is almost twice as large as the source domain. This causes data-mismatch issues and hence provides a perfect test-bed for a domain adaptation task. In total, our corpus has a 40K semantically tagged utterances from each source and target domains. There are around 15 named-entity and 10 descriptive tags. We separated 5K utterances to test the performance of the semantic tagging models. The most frequent entities are: *movie-director* ('*James Cameron*'), *movie-title* ('*Die Hard*'), etc.; whereas top descriptive tags are: *genre* ('*feel good*'), *description* ('*black and white*', '*pg 13*'), *review-rate* ('*epic*', '*not for me*'), *theater-location* ('*near me*','*city center*'), etc.

Unlabeled utterances similar to the movie domain are pulled from a month old web query logs and extracted over 2 million search queries from well-known sites, e.g., IMDB, Netflix, etc. We filtered queries that are similar to our target set that start with *wh*-phrases ('*what*', '*who*', etc.) as well as imperatives '*show*', '*list*', etc. In addition, we extracted web n-grams and entity lists (see §3) from movie related web sites, and online blogs and reviews. We collected around 300K movie review and blog entries on the entities observed in our data. We extract prior distributions for entities and n-grams to calculate entity list η and word-tag β priors (see §3.1).

5.1.2 Syntactic Tagging Datasets

We use the Wall Street Journal (WSJ) section of the Penn Treebank as our labeled source data. Following previous research, we train on sections 00-18, comprised of 38,219 POS-tagged sentences. To evaluate the domain adaptation (DA) approach and to compare with results reported by (Subramanya et al., 2010), we use the first and second half of QuestionBank (Judge et al., 2006) as our development and test sets (target). The QuestionBank contains 4000 POS-tagged questions, however it is difficult to tag with WSJ-trained taggers because the word order is different than WSJ and contains a test-set vocabulary that is twice as large as the one in the development set. As for unlabeled data we crawled the web and collected around 100,000 questions that are similar in style and length to the ones in QuestionBank, e.g. "*wh*" questions. There are 36 different tag sets in the Penn dataset which includes tag labels for verbs, nouns, adjectives, adverbs, modal, determiners, prepositions, etc. More information about the Penn Tree-bank tag set can be found here (Marcus et al., 1993).

5.2 Models

We evaluated several baseline models on two tasks:

5.2.1 Semantic Clustering

Since **MTR** provides a mixture of properties adapted from earlier models, we present performance benchmarks on tag clustering using: (*i*) **LDA**; (*ii*) Hidden Markov Topic Model **HMTM** (Gruber et al., 2005); and, (*iii*) **w-LDA** (Petterson et al., 2010) that uses word features as priors in LDA. When a uniform β hyper-parameter is used with no external information on the state transitions in MTR, it reduces to a HMTM model. Similarly, if no Markov properties are used (bag-of-words), MTR reduces to w-LDA. Each topic model uses Gibbs sampling for inference and parameter learning. We sample models for 1000 iterations, with a 500-iteration burn-in and a sampling lag of 10. For testing we iterated the Gibbs sampler using the trained model for 10 iterations on the testing data.

5.2.2 SSL for Semantic/Syntactic Tagging

We evaluated three different baselines against our SSL models:

⋆ **CRF**: a standard supervised sequence tagging.

⋆ **Self-CRF**: a wrapper method for SSL using self-training. First a supervised learning algorithm is used to build a CRF model based on the labeled data. A CRF model is used to decode the unlabeled data to generate more labeled examples for re-training.

⋆ **SSL-Graph**: A SSL model presented in (Subramanya et al., 2010) that uses graph-based learning as posterior tag smoother for CRF model using Eq.(5).

In addition to the three baseline, we evaluated three variations of our SSL method:

⋆ **SSL-MTR**: Our first version of SSL uses MTR to

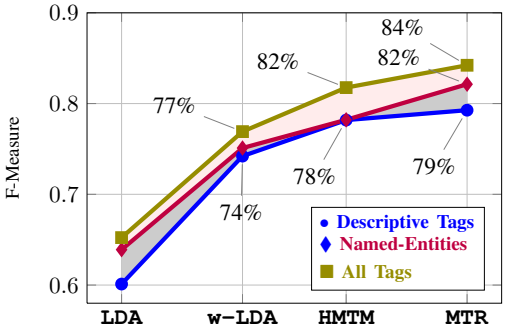

Figure 2: F-measure for semantic clustering performance. Performance differences for three different baseline models and our MTR approach by different semantic tags.

smooth the semantic tag posteriors of a unlabeled data decoded by the CRF model using Eq.(5).

⋆ **R-SSL-Graph**: Our second version uses graph-learning to smooth the tag posteriors and retrain a new CRF model using *retrospective* SSL in Eq.(7).

⋆ **R-SSL-MTR**: Our full model uses MTR as a Bayesian smoothing model, and retrospective SSL in Eq.(7) for iterative CRF training.

For all the CRF models, we use lexical features consisting of unigrams in a five-word window around the current word. To include contextual information, we add binary features for all possible tags. We inject dictionary constraints to all CRF models, such as features indicating label prior information. For each model we use several named entity features, e.g., *movie-title*, *actor-name*, etc., non-named entity (descriptive) features, e.g., *movie-description*, *movie-genre*, and domain independent dictionaries, e.g, *time*, *location*, etc. For graph-based learning, we implemented the algorithm presented in (Subramanya et al., 2010) and used the same hyper-parameters and features. For the rest of the hyper-parameters, we used: α=0.01 for MTR, π=0.5 for interpolation mixing. These parameters were chosen based on the performance of the development set. All CRF objective functions were optimized using Stochastic Gradient Descent.

5.3 Results and Discussions

5.3.1 Experiment 1: Clustering Semantic Tags.

Here, we want to demonstrate the performance of MTR model for capturing relationships between words and semantic tags against baseline topic models: LDA, HMTM, w-LDA. We take the semantically labeled utterances from the movie target domain and use the first half for training and the rest for performance testing. We use all the collected unlabeled web queries from the movie domain. For fair comparison, each benchmark topic model is provided with prior information on word-semantic tag distributions based on the labeled training data, hence, each K latent topic is assigned to one of K semantic tags at the beginning of Gibbs sampling.

We evaluate the performance separately on descriptive tags, named-entities, and all tags together. The performance of the four topic models are reported in Figure 2. LDA shows the worst performance, even though some supervision is provided by way of labeled semantic tags. Although w-LDA improves semantic clustering performance over LDA, the fact that it does not have Markov properties makes it fall short behind MTR. As for the effect of word features in MTR, we see a 3% absolute performance gain over the second best performing HMTM baseline on named-entity tags, a 1% absolute gain on descriptive tags and a 2% absolute overall gain. As expected, we see a drop in F-measure on all models on descriptive tags.

5.3.2 Experiment 2: Domain Adaptation Task.

We compare the performance of our SSL model to that of state-of-the-art models on semantic and syntactic tagging. Each SSL model is built using labeled training data from the source domain and unlabeled training data from target domain. In Table 2 we show the results on Movie and QuestionBank target test datasets. The results of SSL-Graph on QuestionBank is taken from (Subramanya et al., 2010). The self-training model, Self-CRF adds 3% improvement over supervised CRF models on movie domain, but does not improve syntactic tagging. Because it is always inherently biased towards the source domain, self-training tends to reinforce the knowledge that the supervised model already has. SSL-Graph works much better for both syntactic and semantic tagging compared to CRF and Self-CRF models. Our Bayesian MTR efficiently extracts information from the unlabeled data for the target domain. Combined with retrospective training, R-SSL-MTR demonstrates noticeable improvements, ∼2% on descriptive tags, and 1% absolute gains in overall semantic tag-

ging performance over SSL-Graph. On syntactic tagging, the two retrospective learning models is comparable, close to 1% improvement over the SSL-Graph and SSL-MTR.

	Movie Domain			QBank
Model	Desc.	NE	All	POS
CRF	75.05	75.84	75.84	83.80
Self-CRF	78.96	79.53	79.19	84.00
SSL-Graph	80.27	81.35	81.23	86.80
SSL-MTR	79.87	79.31	79.19	86.30
R-SSL-Graph	80.58	81.95	81.52	87.12
R-SSL-MTR	82.76	82.27	**82.24**	**87.34**

Table 2: Domain Adaptation performance in **F-measure** on Semantic Tagging on **Movie** Target domain and POS tagging on **QBank:QuestionBank**. Best performing models are **bolded**.

5.3.3 Experiment 3: Analysis of Semantic Disambiguation.

Here we focus on the accuracy of our models in tagging semantically ambiguous words. We investigate words that have more than one observed semantic tag in training data, such as "*are there any* $[\textbf{\textit{war}}]_{\text{genre}}$ *movies available.*", "*remove all movies about* $[\textbf{\textit{war}}]_{\text{description}}$."). Our corpus contained 30,000 unique vocabulary, 55% of which are contained in one or more semantic categories. Only 6.5% of those are tagged as multiple categories (polysemous), which are the sources of semantic ambiguity. Table-3 shows the precision of two best models for most confused words.

We compare our two best SSL models with different smoothing regularizes: R-SSL-MTR (**MTR**) and R-SSL-Graph (**GRAPH**). We use precision and recall criterion on semantically confused words.

In Table 3 we show two most frequent descriptive tags; *genre* and *description*, and commonly misclassified words by the two models. Results indicate that the R-SSL-MTR, performs better than the R-SSL-Graph, in activating the correct meaning of a word. The results indicate that incorporating context information with MTR is an effective option for identifying semantic ambiguity.

6 Conclusions

We have presented a novel semi supervised learning approach using a probabilistic clustering method to semantically tag spoken language utterances. Our results show that encoding priors on words and context information contributes significantly to the performance of semantic clustering. We have also described an efficient iterative learning model that can handle data inconsistencies that leads to performance increases in semantic and syntactic tagging.

	genre		description	
Vocab.	GRAPH	MTR	GRAPH	MTR
war	50%	100%	75%	88%
popular	90%	89%	80%	100%
kids	78%	86%	–	100%
crime	49%	80%	86%	67%
zombie	67%	89%	67%	86%

Table 3: Classification performance in F-measure for semantically ambiguous words on the most frequently confused descriptive tags in the movie domain.

As a future work, we will investigate using session data, namely the entire dialog between the human and the computer. Rather than using single turn utterances, we hope to utilize the context information, e.g., information from previous turns for improving the performance of the semantic tagging of the current turns.

References

D. Blei, A. Ng, and M. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*.

J. Boyd-Graber, D. Blei, and X. Zhu. 2007. A topic model for word sense disambiguation. *Proc. EMNLP*.

P.F. Brown, V.J.D. Pietra, P.V. deSouza, and J.C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

O. Chapelle, B. Schlkopf, and Alexander Zien. 2006. Semi-supervised learning. *MIT Press*.

H. Daumé-III. 2010. Frustratingly easy semi-supervised domain adaptation. *Proc. Workshop on Domain Adaptation for Natural Language Processing at ACL*.

T.L Griffiths, M. Steyvers, D.M. Blei, and J.M. Tenenbaum. 2005. Integrating topics and syntax. *Proc. of NIPS*.

A. Gruber, M. Rosen-Zvi, and Y. Weiss. 2005. Hidden topic markov models. *Proc. of ICML*.

H. Guo, H. Zhu, Z. Guo, X. Zhang, X. Wu, and Z. Su. 2009. Domain adaptation with latent semantic association for named entity recognition. *Proc. NAACL*.

J. Judge, A. Cahill, and J.Van Genabith. 2006. Question-bank: Creating corpus of parse-annotated questions. *Proc. Int. Conf. Computational Linguistics and ACL*.

A. Lavoie, M.E. Otey, N. Ratliff, and D. Sculley. 2011. History dependent domain adaptation. *Proc. NIPS Workshop on Domain Adaptation*.

X. Li, Y.-Y. Wang, and A. Acero. 2009. Extracting structured information from user queries with semi-supervised conditional random fields. *Proc. of SIGIR*.

L. Li, B. Roth, and C. Sporleder. 2010. Topic models for word sense disambiguation and token-based idiom detection. *Proc. ACL*.

X. Li. 2010. Understanding semantic structure of noun phrase queries. *Proc. ACL*.

J Liu, X. Li, A. Acero, and Ye-Yi Wang. 2011. Lexicon modeling for query understanding. *Proc. of ICASSP*.

M. P. Marcus, B. Santorini, and M.A. Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 27:1–30.

D. Mimno, W. Li, and A. McCallum. 2008. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. *Proc. UAI*.

T. Moon, K. Erk, and J. Baldridge. 2010. Crouching dirichlet, hidden markov model: Unsupervised pos tagging with context local tag generation. *Proc. ACL*.

V.-A. Nyugen, J. Boyd-Graber, and P. Resnik. 2012. Sits: A hierarchical nonparametric model using speaker identity for topic segmentation in multiparty conversations. *Proc. ACL*.

P. Pantel. 2003. Clustering by committee. *Ph.D. Thesis, University of Alberta, Edmonton, Alta., Canada*.

J. Petterson, A. Smola, T. Caetano, W. Buntine, and S. Narayanamurthy. 2010. Word features for latent dirichlet allocation. In *Proc. NIPS*.

J. Reisinger and R. Mooney. 2011. Cross-cutting models of lexical semantics. In *Proc. of EMNLP*.

S. Singh, D. Hillard, and C. Leggetter. 2010. Minimally-supervised extraction of entities from text advertisements. *Proc. NAACL-HLT*.

A. Stolcke. 2002. An extensible language modeling toolkit. *Proc. Interspeech*.

A. Subramanya, S. Petrov, and F. Pereira. 2010. Efficient graph-based semi-supervised learning of structured tagging models. In *Proc. EMNLP*.

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

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*.

X. Zhu. 2005. Semi-supervised learning literature survey. *Technical Report 1530, University of Wisconsin-Madison*.