Graph-Based Semi-Supervised Conditional Random Fields For Spoken Language Understanding Using Unaligned Data

Mohammad Aliannejadi
Amirkabir University
of Technology
(Tehran Polytechnic)
m.aliannejadi@aut.ac.ir

Masoud Kiaeeha
Sharif University of
Technology
kiaeeha@ce.sharif.edu

Shahram Khadivi &
Saeed Shiry Ghidary
Amirkabir University
of Technology
(Tehran Polytechnic)
{khadivi, shiry}@aut.ac.ir

Abstract

We experiment graph-based Semi-Supervised Learning (SSL) of Conditional Random Fields (CRF) for the application of Spoken Language Understanding (SLU) on unaligned data. The aligned labels for examples are obtained using IBM Model. We adapt a baseline semi-supervised CRF by defining new feature set and altering the label propagation algorithm. Our results demonstrate that our proposed approach significantly improves the performance of the supervised model by utilizing the knowledge gained from the graph.

1 Introduction

The aim of Spoken Language Understanding (SLU) is to interpret the intention of the user's utterance. More specifically, a SLU system attempts to find a mapping from user's utterance in natural language, to the limited set of concepts that is structured and meaningful for the computer. As an example, for the sample utterance:

I want to return to Dallas on Thursday

It's corresponding output would be:

```
GOAL : RETURN
TOLOC.CITY = Dallas
RETURN.DATE = Thursday.
```

SLU can be widely used in many real world applications; however, data processing costs may impede practicability of it. Thus, attempting to train a SLU model using less training data is a key issue.

The first statistical SLU system was based on hidden Markov model and modeled using a finite state semantic tagger employed in AT&T's CHRONUS system (Pieraccini et al., 1992). Their semantic representation was flat-concept; but, later He and Young (2005) extended the representation to a hierarchical structure and modeled the problem using a push-down automaton. There are other works which have dealt with SLU as a sequential labeling problem. Raymond and Riccardi (2007) and Wang and Acero (2006) have fully annotated the data and trained the model in discriminative frameworks such as CRF. CRF captures many complex dependencies and models the sequential relations between the labels; therefore, it is a powerful framework for SLU.

The Semi-Supervised Learning (SSL) approach has drawn a raft of interest among the machine learning community basically because of its practical application. Manual tagging of data can take considerable effort and time; however, in the training phase of SSL, a large amount of unlabeled data along with a small amount of labeled data is provided. This makes it more practicable and cost effective than providing a fully labeled set of training data; thus, SSL is more favorable.

Graph-based SSL, the most active area of research in SSL in the recent years, has shown to outperform other SSL methods (Chapelle et al., 2006). Graph-based SSL algorithms are generally run in two steps: graph construction and label propagation. Graph construction is the most important step in graph-based SSL; and, the fundamental approach is to assign labeled and unlabeled examples to nodes of the graph. Then, a similarity function is applied to compute similarity between pairs of nodes. The computed similarities are then assigned as the weight of the edges connecting the nodes (Zhu et al., 2003). Label propagation operates on the constructed graph. Based on the constraints or properties derived from the graph, labels are propagated from a few labeled nodes to the entire graph. These constraints include smoothness (Zhu et al., 2003; Subramanya et al., 2010; Talukdar et al., 2008; Garrette and Baldridge, 2013), and sparsity (Das and Smith, 2012; Zeng et al., 2013).

Labeling unaligned training data requires much

Mohammad Aliannejadi, Masoud Kiaeeha, Shahram Khadivi and Saeed Shiry Ghidary. 2014. Graph-Based Semi-Supervised Conditional Random Fields For Spoken Language Understanding Using Unaligned Data. In *Proceedings of Australasian Language Technology Association Workshop*, pages 98−103.

less effort compared to aligned data (He and Young, 2005). Nevertheless, unaligned data cannot be used to train a CRF model directly since CRF requires *fully-annotated* data. On the other hand, robust parameter estimation of a CRF model requires a large set of training data which is unrealistic in many practical applications. To overcome this problem, the work in this paper applies semi-supervised CRF on unlabeled data. It is motivated by the hypothesis that data is aligned to labels in a monotone manner, and words appearing in similar contexts tend to have same labels. Under these circumstances, we were able to reach 1.64% improvement on the F-score over the supervised CRF and 1.38% improvement on the F-score over the self trained CRF.

In the following section we describe the algorithm this work is based on and our proposed algorithm. In Section 3 we evaluate our work and in the final section conclusions are drawn.

2 Semi-supervised Spoken Language Understanding

The input data is unaligned and represented as a semantic tree, which is described in (He and Young, 2005). The training sentences and their corresponding semantic trees can be aligned monotonically; hence, we chose IBM Model 5 (Khadivi and Ney, 2005) to find the best alignment between the words and nodes of the semantic tree (labels). Thus, we have circumvented the problem of unaligned data. More detailed explanation about this process can be found in our previous work (Aliannejadi et al., 2014). This data is then used to train the supervised and semi-supervised CRFs.

2.1 Semi-supervised CRF

The proposed semi-supervised learning algorithm is based on (Subramanya et al., 2010). Here, we quickly review this algorithm (Algorithm 1).

In the first step, the CRF model is trained on the labeled data ($\mathcal{D}_l$) according to (1):

$$\Lambda^* = \underset{\Lambda \in \mathbb{R}^K}{\arg\min} \Big[-\sum_{i=1}^{l} \log \mathrm{p}(\mathbf{y_i}|\mathbf{x_i};\Lambda) + \gamma \|\Lambda\|^2 \Big], \tag{1}$$

where Λ^* is the optimal parameter set of the base CRF model and $\|\Lambda\|^2$ is the squared ℓ_2-norm regularizer whose impact is adjusted by γ. At the first line, Λ^* is assigned to $\Lambda_{(n=0)}$ i.e. the initial parameter set of the model.

Algorithm 1 Semi-Supervised Training of CRF

```
 1: $\Lambda_{(n=0)}$ = TrainCRF($\mathcal{D}_l$)
 2: G = BuildGraph($\mathcal{D}_l \cup \mathcal{D}_u$)
 3: {r} = CalcEmpiricalDistribution($\mathcal{D}_l$)
 4: while not converged do
 5:     {m} = CalcMarginals($\mathcal{D}_u, \Lambda_n$)
 6:     {q} = AverageMarginals(m)
 7:     {q̂} = LabelPropagation(q, r)
 8:     $\mathcal{D}_u^v$ = ViterbiDecode({q̂}, $\Lambda_n$)
 9:     $\Lambda_{n+1}$ = RetrainCRF($\mathcal{D}_l \cup \mathcal{D}_u^v, \Lambda_n$);
10: end while
11: Return final $\Lambda_n$
```

In the next step, the k-NN similarity graph (G) is constructed (line 2), which will discussed in more detail in Section 2.3. In the third step, the empirical label distribution (r) on the labeled data is computed. The main loop of the algorithm is then started and the execution continues until the results converge.

Marginal probability of labels (m) are then computed on the unlabeled data ($\mathcal{D}_u$) using Forward-Backward algorithm with the parameters of the previous CRF model (Λ^n), and in the next step, all the marginal label probabilities of each trigram are averaged over its occurrences (line 5 and 6).

In label propagation (line 7), trigram marginals (q) are propagated through the similarity graph using an iterative algorithm. Thus, they become smooth. Empirical label distribution (r) serves as the priori label information for labeled data and trigram marginals (q) act as the seed labels. More detailed discussion is found in Section 2.4.

Afterwards, having the results of label propagation (q̂) and previous CRF model parameters, labels of the unlabeled data are estimated by combining the interpolated label marginals and the CRF transition potentials (line 8). For every word position j for i indexing over sentences, interpolated label marginals are calculated as follows:

$$\hat{\mathrm{p}}(y_i^{(j)} = y|\mathbf{x}_i) = \alpha \mathrm{p}(y_i^{(j)} = y|\mathbf{x}_i;\Lambda_n) + (1-\alpha)\hat{\mathrm{q}}_{T(i,j)}(y), \tag{2}$$

where $T(i,j)$ is a trigram centered at position j of the ith sentence and α is the interpolation factor.

In the final step, the previous CRF model parameters are regularized using the labels estimated for the unlabeled data in the previous step (line 9)

Description	Feature
Context	$x_1\ x_2\ x_3\ x_4\ x_5$
Left Context	$x_1\ x_2$
Right Context	$x_4\ x_5$
Center Word in trigram	$\_\ x_3\ \_$
Center is Class	$IsClass(x_3)$
Center is Preposition	$IsPreposition(x_3)$
Left is Preposition	$IsPreposition(x_2)$

Table 1: Context Features used for constructing the similarity graph

as follows:

$$\Lambda_{n+1} = \underset{\Lambda \in \mathbb{R}^K}{\arg\min} \Big[-\sum_{i=1}^{l} \log \mathrm{p}(\mathbf{y_i}|\mathbf{x_i};\Lambda_n) \\ - \eta \sum_{i=l+1}^{u} \log \mathrm{p}(\mathbf{y_i}|\mathbf{x_i};\Lambda_n) + \gamma \|\Lambda\|^2 \Big], \quad (3)$$

where η is a trade-off parameter whose setting is discussed later in Section 3.

2.2 CRF Features

By aligning the training data, many informative labels are saved which are omitted in other works (Wang and Acero, 2006; Raymond and Riccardi, 2007). By saving these information, the first order label dependency helps the model to predict the labels more precisely. Therefore the model manages to predict the labels using less lexical features and the feature window that was [-4,+2] in previous works is reduced to [0,+2]. Using smaller feature window improves the generalization of the model (Aliannejadi et al., 2014).

2.3 Similarity Graph

In our work we have considered trigrams as the nodes of the graph and extracted features of each trigram $x_2\ x_3\ x_4$ according to the 5-word context $x_1\ x_2\ x_3\ x_4\ x_5$ it appears in. These features are carefully selected so that nodes are correctly placed in neighborhood of the ones having similar labels. Table 1 presents the feature set that we have applied to construct the similarity graph.

$IsClass$ feature impacts the structure of the graph significantly. In the pre-processing phase specific words are marked as classes according to the corpus' accompanying database. As an example, city names such as Dallas and Baltimore are represented as *city_name* which is a class type. Since these classes play an important role in calculating similarity of the nodes, $IsClass$ feature is used to determine if a given position in a context is a class type.

Furthermore, prepositions like *from* and *between* are also important, e.g. when two trigrams like "*from Washington to*" and "*between Dallas and*" are compared. The two trigrams are totally different while both of them begin with a preposition and are continued with a class. Therefore, $IsPreposition$ feature would be particularly suitable to increase the similarity score of these two trigrams. In many cases, these features have a significant effect in assigning a better similarity score.

To define a similarity measure, we compute the Pointwise Mutual Information (PMI) between all occurrences of a trigram and each of the features. The PMI measure transforms the independence assumption into a ratio (Lin, 1998; Razmara et al., 2013). Then, the similarity between two nodes is measured as the cosine distance between their PMI vectors. We carefully examined the similarity graph on the training data and found out the head and tail trigrams of each sentence which contain *dummy* words, make the graph sparse. Hence, we have ignored those trigrams.

2.4 Label Propagation

After statistical alignment, the training data gets noisy. Hence, use of traditional label propagation algorithms causes an error propagation over the whole graph and degrades the whole system performance. Thus, we make use of the Modified Adsorption (MAD) algorithm for label propagation.

MAD algorithm controls the label propagation more strictly. This is accomplished by limiting the amount of information that passes from a node to another (Talukdar and Pereira, 2010). Soft label vectors $\hat{Y}_v$ are found by solving the unconstrained optimization problem in (4):

$$\min_{\hat{Y}} \quad \sum_{l \in C} \Big[\mu_1 (Y_l - \hat{Y}_l)^\top S \, (Y_l - \hat{Y}_l) \\ + \mu_2 \hat{Y}_l^\top L' \hat{Y}_l \; + \; \mu_3 \big\| \hat{Y}_l - R_l \big\|^2 \Big], \quad (4)$$

where μ_i are hyper-parameters and R_l is the empirical label distribution over labels i.e. the prior belief about the labeling of a node. The first term of the summation is related to label score injection from the initial score of the node and

	% of Labeled Data		
	10	20	30
Supervised CRF	86.07	87.69	88.64
Self-trained CRF	86.34	87.73	88.64
Semi-supervised CRF	87.72	88.75	89.12

Table 2: Comparison of training results. Slot/Value F-score in %.

makes the output match the seed labels Y_l (Razmara et al., 2013). The second term is associated with label score acquisition from neighbor nodes i.e. smooths the labels according to the similarity graph. In the last term, the labels are regularized to match a priori label R_l in order to avoid false labels for high degree unlabeled nodes. A solution to the optimization problem in (4) can be found with an efficient iterative algorithm described in (Talukdar and Crammer, 2009).

Many errors of the alignment model are corrected through label propagation using the MAD algorithm; whereas, those errors are propagated in traditional label propagation algorithms such as the one mentioned in (Subramanya et al., 2010).

2.5 System Overview

We have implemented the Graph Construction in Java and the CRF is implemented by modifying the source code of CRFSuite (Okazaki, 2007). We have also modified Junto toolkit (Talukdar and Pereira, 2010) and used it for graph propagation. The whole source code of our system is available online[1]. The input utterances and their corresponding semantic trees are aligned using GIZA++ (Och and Ney, 2000); and then used to train the base CRF model. The graph is constructed using the labeled and unlabeled data and the main loop of the algorithm continues until convergence. The final parameters of the CRF are retained for decoding in the test phase.

3 Experimental Results

In this section we evaluate our results on Air Travel Information Service (ATIS) data-set (Dahl et al., 1994) which consists of 4478 training, 500 development and 896 test utterances. The development set was chosen randomly. To evaluate our work, we have compared our results with results from Supervised CRF and Self-trained CRF (Yarowsky, 1995).

For our experiments we set hyper-parameters as follows: for graph propagation, $\mu_1 = 1, \mu_2 = 0.01$, $\mu_3 = 0.01$, for Viterbi decoding, $\alpha = 0.1$, for CRF-retraining, $\eta = 0.1$, $\gamma = 0.01$. We have chosen these parameters along with graph features and graph-related parameters by evaluating the model on the development set. We employed the L-BFGS algorithm to optimize CRF objective functions; which is designed to be fast and low-memory consumer for the high-dimensional optimization problems (Bertsekas, 1999).

We have post-processed the sequence of labels to obtain the slots and their values. The slot-value pair is compared to the reference test set and the result is reported in F-score of slot classification. Table 2 demonstrates results obtained from our semi-supervised CRF algorithm compared to the supervised CRF and self-trained CRF. Experiments were carried out having 10%, 20% and 30% of data being labeled. For each of these tests, labeled set was selected randomly from the training set. This procedure was done 10 times and the reported results are the average of the results thereof. The Supervised CRF model is trained only on the labeled fraction of the data. However, the Self-trained CRF and Semi-supervised CRF have access to the rest of the data as well, which are unlabeled. Our Supervised CRF gained 91.02 F-score with 100% of the data labeled which performs better compared to 89.32% F-score of Raymond and Riccardi (2007) CRF model.

As shown in Table 2, the proposed method performs better compared to supervised CRF and self-trained CRF. The most significant improvement occurs when only 10% of training set is labeled; where we gain 1.65% improvement on F-score compared to supervised CRF and 1.38% compared to self-trained CRF.

4 Conclusion

We presented a simple algorithm to train CRF in a semi-supervised manner using unaligned data for SLU. By saving many informative labels in the alignment phase, the base model is trained using fewer features. The parameters of the CRF model are estimated using much less labeled data by regularizing the model using a nearest-neighbor graph. Results demonstrate that our proposed algorithm significantly improves the performance compared to supervised and self-trained CRF.

[1] https://github.com/maxxkia/g-ssl-crf

References

Mohammad Aliannejadi, Shahram Khadivi, Saeed-Shiry Ghidary, and MohammadHadi Bokaei. 2014. Discriminative spoken language understanding using statistical machine translation alignment models. In Ali Movaghar, Mansour Jamzad, and Hossein Asadi, editors, *Artificial Intelligence and Signal Processing*, volume 427 of *Communications in Computer and Information Science*, pages 194–202. Springer International Publishing.

Dimitri P Bertsekas. 1999. Nonlinear programming.

Olivier Chapelle, Bernhard Schölkopf, Alexander Zien, et al. 2006. *Semi-supervised learning*, volume 2. MIT press Cambridge.

Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. Expanding the scope of the atis task: The atis-3 corpus. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 43–48, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dipanjan Das and Noah A. Smith. 2012. Graph-based lexicon expansion with sparsity-inducing penalties. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 677–687, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dan Garrette and Jason Baldridge. 2013. Learning a part-of-speech tagger from two hours of annotation. In *Proceedings of NAACL-HLT*, pages 138–147.

Yulan He and Steve Young. 2005. Semantic processing using the hidden vector state model. *Computer Speech & Language*, 19(1):85 – 106.

Shahram Khadivi and Hermann Ney. 2005. Automatic filtering of bilingual corpora for statistical machine translation. In Andrés Montoyo, Rafael Muńoz, and Elisabeth Métais, editors, *Natural Language Processing and Information Systems*, volume 3513 of *Lecture Notes in Computer Science*, pages 263–274. Springer Berlin Heidelberg.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, ACL '98, pages 768–774, Stroudsburg, PA, USA. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2000. Giza++: Training of statistical translation models.

Naoaki Okazaki. 2007. Crfsuite: a fast implementation of conditional random fields (crfs). *URL http://www.chokkan.org/software/crfsuite*.

R. Pieraccini, E. Tzoukermann, Z. Gorelov, J. Gauvain, E. Levin, Chin-Hui Lee, and J.G. Wilpon. 1992. A speech understanding system based on statistical representation of semantics. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, volume 1, pages 193–196 vol.1, Mar.

Christian Raymond and Giuseppe Riccardi. 2007. Generative and discriminative algorithms for spoken language understanding. In *International Conference on Speech Communication and Technologies*, pages 1605–1608, Antwerp, Belgium, August.

Majid Razmara, Maryam Siahbani, Gholamreza Haffari, and Anoop Sarkar. 2013. Graph propagation for paraphrasing out-of-vocabulary words in statistical machine translation. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*.

Amarnag Subramanya, Slav Petrov, and Fernando Pereira. 2010. Efficient graph-based semi-supervised learning of structured tagging models. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 167–176, Stroudsburg, PA, USA. Association for Computational Linguistics.

ParthaPratim Talukdar and Koby Crammer. 2009. New regularized algorithms for transductive learning. In Wray Buntine, Marko Grobelnik, Dunja Mladeni, and John Shawe-Taylor, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 5782 of *Lecture Notes in Computer Science*, pages 442–457. Springer Berlin Heidelberg.

Partha Pratim Talukdar and Fernando Pereira. 2010. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1473–1481, Stroudsburg, PA, USA. Association for Computational Linguistics.

Partha Pratim Talukdar, Joseph Reisinger, Marius Paşca, Deepak Ravichandran, Rahul Bhagat, and Fernando Pereira. 2008. Weakly-supervised acquisition of labeled class instances using graph random walks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 582–590, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ye-Yi Wang and Alex Acero. 2006. Discriminative models for spoken language understanding. In *International Conference on Speech Communication and Technologies*.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, ACL '95, pages 189–196, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xiaodong Zeng, Derek F Wong, Lidia S Chao, and Isabel Trancoso. 2013. Graph-based semi-supervised model for joint chinese word segmentation and part-of-speech tagging. In *ACL*, pages 770–779.

Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919.