Recurrent Support Vector Machines For Slot Tagging In Spoken Language Understanding

Yangyang Shi and **Kaisheng Yao** and **Hu Chen** and **Dong Yu**
Yi-Cheng Pan and **Mei-Yuh Hwang**
Microsoft
{yanshi,kaisheng.yao,huch,dong.yu,ycpan,mehwang}@microsoft.com

Abstract

We propose recurrent support vector machine (RSVM) for slot tagging. This model is a combination of the recurrent neural network (RNN) and the structured support vector machine. RNN extracts features from the input sequence. The structured support vector machine uses a sequence-level discriminative objective function. The proposed model therefore combines the sequence representation capability of an RNN with the sequence-level discriminative objective. We have observed new state-of-the-art results on two benchmark datasets and one private dataset. RSVM obtained statistical significant 4% and 2% relative average F1 score improvement on ATIS dataset and Chunking dataset, respectively. Out of eight domains in Cortana live log dataset, RSVM achieved F1 score improvement on seven domains. Experiments also show that RSVM significantly speeds up the model training by skipping the weight updating for non-support vector training samples, compared against training using RNN with CRF or minimum cross-entropy objectives.

1 Introduction

One of the key tasks in natural language understanding (Hemphill et al., 1990a; He and Young, 2003; De Mori, 2007; Dinarelli et al., 2008; Wang et al., 2005) is slot tagging that labels user queries with semantic tags. It is a sequence labeling problem that transcribes a sequence of observations $X = [x(1), x(2), ..., x(M)]$ to a sequence of discrete labels $Y = [y(1), y(2), ..., y(M)]$. For example, in the query "show me flights from Seattle to Boston", the words "Seattle" and "Boston" should be labeled, respectively, as the *from-city-name* slot and the *to-city-name* slot.

Recently recurrent neural networks (RNNs) and their variants achieved state-of-the-art performances on slot tagging tasks (Yao et al., 2013; Yao et al., 2014b; Yao et al., 2014a; Graves, 2012; Shi et al., 2015; Mesnil et al., 2015; Peng and Yao, 2015). One direction to improve the sequence labeling is to strengthen the model memorization capability by designing dedicated special structures, for example, using long-short-term-memory (LSTM) networks (Hochreiter and Schmidhuber, 1997; Graves et al., 2013; Yao et al., 2014a), gated RNN and RNN with external memory (RNN-em) (Peng and Yao, 2015). The other direction is to optimize the sequence-level discrimination criterion. For example, recurrent conditional random fields (RCRFs) (Yao et al., 2014b) is trained to optimize the sequence conditional likelihood rather than minimizing frame level cross-entropy applied in conventional RNN based sequence labeling (Prez-ortiz et al., 2001; Yao et al., 2013; Mikolov et al., 2010; Shi et al., 2015; Mesnil et al., 2015).

In this paper, we propose recurrent support vector machines (RSVMs) to improve the discrimination ability of RNNs. Different from RCRFs and conventional RNNs that in essence apply the multinomial logistic regression on the output layer, RSVMs optimize the sequence-level max-margin training criterion used by structured support vector machines (Tsochantaridis et al., 2005) on the output layer of RNNs. There are several advantages of using sequence-level max-margin training over maximum likelihood or minimum cross-entropy. Firstly, the sequence-level max-margin criterion is a *global un-normalized* criterion in which there is no computation cost for normalization. Secondly, using max-margin training, only training samples from support vectors generate non zero errors. In other words, model training can be sped up by skipping the weight updating for non-support vector training samples. Finally, as proven in (Vapnik, 1995), margin maximization is equivalent to minimization of an upper bound on the generalization errors. Max-margin training has no assumption about the model distribution. To use maximum likelihood or minimum cross-entropy, it assumes that the model distribution is peaked. However, especially in natural language processing where the ambiguity is ubiquitous, this assumption does not hold.

Proceedings of NAACL-HLT 2016, pages 393–399,
San Diego, California, June 12-17, 2016.

For example, "seven eleven" can be labeled as time tag or place name (super market name) tag. The conditional probability of tag given "seven eleven" should not be sharp for time or place name.

Recently, SVM is also applied on top of a deep neural network for speech recognition (Zhang et al., 2015). In their work, a cutting-plane algorithm (Joachims et al., 2009) is used, which is computationally expensive for speech recognition tasks. In this paper, we use the stochastic gradient descent algorithm (SGD) (Panagiotakopoulos and Tsampouka, 2013) for model training. The loss function is critical to the sequence level max-margin training criterion, which defines the margin. In this paper, we apply the sequence level hard loss function rather than traditional Hamming loss function (Nguyen and Guo, 2007). In sequence level hard loss function, the wrong sequence is assigned loss one without considering the number of wrong slot labels in the sequence. In the experiments on two bench mark datasets, namely the ATIS (Airline Travel Information Systems) dataset (Hemphill et al., 1990b; Yao et al., 2014b) and the CoNLL 2000 Chunking dataset [1], and private Cortana live log dataset, RSVMs outperformed previous results.

2 Recurrent Support Vector Machines

In this section, we propose RSVM that uses the structured SVM algorithm (Tsochantaridis et al., 2005) to estimate the weights for RNN and label transition probabilities based on the entire training sequence. The training objective in RSVM is the following constrained optimization.

$$\min_{W,A} \frac{1}{2}||W||_2^2+\frac{1}{2}||A||_2^2+C\sum_{k=1}^{K}\zeta_k$$

$$s.t.\ f(Y^{(k)^*})+\zeta_k \geq f(Y^{(k)})+L(Y^{(k)}) \quad \forall Y^{(k)} \tag{1}$$

$$L(Y^{(k)}) \geq 0 \quad \forall Y^{(k)} \tag{2}$$

$$\zeta_k \geq 0 \tag{3}$$

where

$$f(Y^{(k)}) = \sum_{t=1}^{T} a_{y_k(t-1)y_k(t)} + W_{y_k(t)}^T H(t) \tag{4}$$

where C is regularization weight for empirical loss. $Y^{(k)}$ represents the slot label sequence $y^{(k)}(1:T)$ for training sample $X^{(k)}$. $Y^{(k)^*}$ is the ground truth slot sequence $y^{(k)^*}(1:T)$ for $X^{(k)}$. $a_{y^{(k)}(t-1)y^{(k)}(t)}$ is one element in matrix A, representing the weight for the slot label transition features from $y^{(k)}(t-1)$ to $y^{(k)}(t)$. $L(Y^{(k)})$ defines the loss function of a possible slot label sequence for a training sample $X^{(k)}$, which is actually used as a margin to separate the score $f(Y^{(k)^*})$ for ground truth slot label sequence with all other possible slot sequences in Eq. (1). $\zeta^{(k)}$ is the slack variable that penalize the slot label sequence that violates the margin constraint.

The constrained optimization problem can be transformed to an unconstrained optimization problem as

$$\min_{W,A} \quad F(W,A) = \frac{1}{2C}||W||_2^2+\frac{1}{2C}||A||_2^2$$
$$+\sum_{k=1}^{K}[\max_{Y^{(k)}\neq Y^{(k)^*}}\left(f(Y^{(k)})+L(Y^{(k)})\right)-f(Y^{(k)^*})]_+. \tag{5}$$

where $[x]_+$ is the Hinge function that maps x to zero when x is smaller than zero, otherwise $[x]_+ = x$.

The loss function $L(Y^{(k)})$ is critical to the structured SVM training. The following two types of loss functions have been investigated:

$$L(Y^{(k)}) = \sum_{t=1}^{T} 1\left(y^{(k)}(t) \neq y^{(k)^*}(t)\right) \tag{6}$$

$$L(Y^{(k)}) = 1\left(y^{(k)}(1:T) \neq y^{(k)^*}(1:T)\right) \tag{7}$$

Eq. (6) is Hamming loss that is applied by (Nguyen and Guo, 2007) for structured SVM sequence labeling. Eq. (7) is sequence level hard loss function that always give loss one to wrong slot label sequences no matter how many words are labeled with wrong slot labels. In our experiment, we find that the margin defined by Eq. (7) gives the best performance.

2.1 Training Procedure For Recurrent Support Vector Machines

Fig. 1 depicts the architecture of RSVM that can be viewed as the conventional RNN unrolled over the sequence $x(1:T)$. For each single training sample $x(1:T)$, a forward inference and a backward learning are carried out to sweep over the network shown in Fig. 1.

In the forward inference, an unnormalized slot score vector $y(t)$ is computed based on each word input $x(t)$ and its corresponding auxiliary feature $Cx(t)$. The word input and auxiliary feature are encoded in one-hot representation. As shown in Fig. 1, a slot label lattice is generated for the training sample $x(1:T)$. Using Viterbi algorithm, two best slot label sequences $Y^{(k)^{\text{top}}}$ and $Y^{(k)^{\text{second}}}$ are derived from the lattice. In the decoding phase, only the best slot label sequence is computed.

In backward learning, the sub-gradient (Ratliff et al., 2007) are calculated to update the weights for RSVMs. When $Y^{(k)^{\text{top}}} \neq Y^{(k)^*}$ the sub-gradient is

$$\frac{\partial \zeta_k}{\partial \theta} = \frac{\partial f(Y^{(k)^{\text{top}}})}{\partial \theta} - \frac{\partial f(Y^{(k)^*})}{\partial \theta}. \tag{8}$$

When $Y^{(k)^{\text{top}}} = y^{(k)^*}$ and $y^{(k)^*} - Y^{(k)^{\text{second}}} < L(Y^{(k)})$, the

[1] See http://www.cnts.ua.ac.be/conll2000/chunking.

sub-gradient is

$$\frac{\partial \zeta_k}{\partial \theta} = \frac{\partial f(Y^{(k)^{\text{second}}})}{\partial \theta} - \frac{\partial f(Y^{(k)^*})}{\partial \theta}. \quad (9)$$

When $Y^{(k)^{\text{top}}} = Y^{(k)^*}$ and $y^{(k)^*} - Y^{(k)^{\text{second}}} \geq L(Y^{(k)})$, the subgradient is zero. Our experiment show that the RSVM training can be substantially sped up by skipping the backward weight updating for non-support vector training samples that obtain zero subgradient.

In Eq. (8) and (9), θ represents the weights in RSVMs. Specifically, the weights W, A, U and V are updated using sequence level mini-batch method. The weights O connecting hidden layers are updated using Backpropagation Through Time (BPTT) (Werbos, 1990).

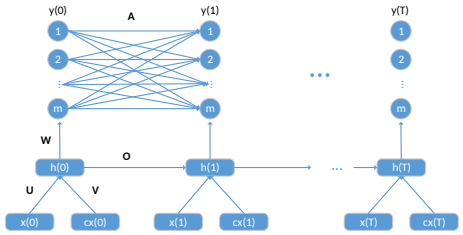

Figure 1: Recurrent support vector machines for slot tagging. U is the weight matrix connecting the word input to the hidden layer, V is the weight matrix connecting auxiliary feature input to the hidden layer, O is the weight matrix connecting previous hidden state to the current hidden state and W is the weight matrix connecting the hidden layer to the output layer. A represents the weight for slot label transition features.

3 Experiments

3.1 Data

To evaluate performances of the proposed model, three sets of experiments were conducted. The first set of experiments are based on ATIS dataset (Hemphill et al., 1990b; Yao et al., 2014b). There are 893 queries from ATIS-III, Nov93 and Dec94 for testing, and 4978 utterances from the rest of ATIS-III and ATIS-II used for training. The training data contains 127 unique slot tags.

The second dataset used in the experiment is CoNLL 2000 Chunking dataset. Chunking is also called shallow parsing that assigns syntactic labels to segments of a sequence of words. Chunking and slot tagging are typical sequence labeling problems. In this paper, we use the chunking task to further verify the performance of the proposed RSVM model. In the CoNLL 2000 Chunking task, the training data are from sections 15-18 of WSJ data and the test data are from section 20. In the training data, there are 220663 tokens with 19123 unique words and additional 45 different types of Part-Of-Speech (POS).

The last dataset is Cortana live log dataset which is constituted by 8 domains, namely alarm, calender, communication, note, ondevice, places, reminder and weather. In total, there are 71 slots. There are 42506 queries used for training and 5290 queries for testing. The data distribution is described in Table. 1. The last column of Table. 1 shows the average query length (the number of words in one query) on different domains.

domain	train	test	length
alarm	3816	452	4.3
calendar	4138	475	4.5
communication	9551	1262	3.8
note	829	139	4.6
ondevice	4384	572	2.4
places	6167	753	4.7
reminder	5359	720	4.1
weather	8270	917	3.4

Table 1: Cortana live log data distribution.

3.2 Settings

In this paper, we use a predefined maximum iteration number to terminate the training. The learning rate is dynamically adjusted using AdaGrad (Duchi et al., 2011).

In all the experiments, we set the hidden layer size to 300 and initial learning rate to 0.1. In RSVMs, the surrounding two words of the current word are used as auxiliary feature which is represented as bag of words. We set the maximum iteration to 20 for ATIS and 30 for Chunking and Cortana live log. For each dataset, we trained 10 models with the same parameter settings except using different random initialization.

3.3 Results on ATIS

ATIS is a well studied benchmark dataset. Table. 2 gives the slot tagging F1 scores achieved by different models in the literature, using the same data settings. There are three blocks in Table. 2. The top block gives the F1 score obtained by CRF and simple RNN. The middle block gives the results obtained by applying advanced RNN architectures such as LSTM, Gated RNN and RNN with external memories (RNN-em) (Peng and Yao, 2015). These advanced RNNs improves RNN by enhancing its memory (sequence representation capability). The bottom block gives results using the proposed RSVMs.

The bottom part gives F1 scores of the proposed RSVM method. We show results generated by 10 models with different random seeds. The average F1 score of RSVM is similar to the best score of RNN-em. F1 score distribution of 10 RSVM models gets significant improvement over the average score of RNN-em (z-test $p - value = 0.0002 << 0.05$). Fundamentally, the proposed RSVM

model	F1(%)
CRF (Mesnil et al., 2015)	92.9
DBN (Deoras and Sarikaya, 2013)	93.2
RNN (Yao et al., 2013)	94.1
RNN-Jordan(Mesnil et al., 2015)	94.3
RNN-embed(Xu and Sarikaya, 2013)	94.4
RNN-joint (Shi et al., 2015)	94.6
RNN-hybrid(Mesnil et al., 2015)	95.1
deep-LSTM (Yao et al., 2014a)	95.0
GRNN-max(Peng and Yao, 2015)	94.7
RNN-em-min(Peng and Yao, 2015)	94.7
RNN-em-average(Peng and Yao, 2015)	95.0
RNN-em-max(Peng and Yao, 2015)	95.2
RSVM-min	94.9
RSVM-average	95.2
RSVM-max	95.5

Table 2: F1 score (in %) for slot tagging on ATIS achieved by different models using only lexical feature. "-min", "-max", and "-average" each denotes mininum, maximum and average F1 scores for a corresponding method.

is based on simple RNN. Comparing LSTM and RNN-em, the proposed model has simpler topology. Note that in (Yao et al., 2014a) and (Peng and Yao, 2015), their advanced models are trained using local normalization method without using sequence level optimization. So the superiority of the proposed RSVM may come from the sequence training and the powerful discriminant capability of SVM. Applying the proposed RSVM method to LSTM or other advanced RNN can be a promising direction for future work.

3.4 Results on CoNLL 2000 Chunking

Table. 3 gives the F1 scores of different models on CoNLL 2000 chunking experiment. To our best knowledge, the first neural network (NN) based chunking model is proposed in (Collobert et al., 2011). Using four basic natural language processing tasks, namely POS tagging, chunking, name entity recognition and semantic role labeling, they demonstrate the ability of NN to discover hidden representations. In their work, only simple input feature is used. There is not any task-specific feature engineering work in their proposed system. Their model purely relies on the NN feature representation that are learned from large amount of unlabeled data. As shown in Table. 3, their system performs better than all the previous systems on CoNLL 2000 chunking dataset.

The performance of Bidirectional LSTM (BLSTM), RCRF and the proposed RSVM on chunking task further confirms the conclusion in (Collobert et al., 2011) that NN is able to discover the internal representations that are useful for different natural language processing tasks. Additionally, the results of BLSTM, RCRF and RSVM, indicate that RNNs have better capabilities to discover the sequence representation than NN. The average F1 score of RCRF and RSVM are 94.9% and 95.0%, respectively. Comparing the F1 score distribution, RSVM achieves the significant improvement over RCRF (paired t-test $p-value = 0.012 < 0.05$). As shown in Table. 3, replacing the CRF objective function with structured SVM max-margin criterion could generate further improvement. The average performance of RSVM is better than the best result of RCRF shown in the table.

model	F1(%)
SVM (Kudo and Matsumoto, 2000)	93.5
gen-Winnow(Zhang et al., 2001)	93.9
SVM (Kudo and Matsumoto, 2001)	93.9
CRF (McDonald et al., 2005)	94.3
CRF (Sun et al., 2008)	94.3
CRF (Sha and Pereira, 2005)	94.4
NN (Collobert et al., 2011)	94.5
BLSTM (Wang et al., 2015)	94.6
RCRF-min	94.7
RCRF-average	94.9
RCRF-max	94.9
RSVM-min	94.7
RSVM-average	95.0
RSVM-max	95.1

Table 3: F1 score (in %) for chunking on CoNLL 2000 shared task using different models. All these models use word features as well as POS features. "-min", "-max", and "-average" each denotes minimum, maximum and average F1 scores for a corresponding method.

3.5 Results on Internal Live dataset

In this section, we compare different slot models on different domains based on Cortana live log data.

Table. 4 compares the F1 score on CRF, RNN, RCRF, joint-RNN and the proposed RSVM on alarm, calendar, communication, and note. Table. 5 presents the F1 score of different models on the rest domains. "RNN" denotes the Elman type of RNN for slot tagging which uses current word information, previous slot output information and context window information (surrounding four words) (Yao et al., 2013). "RCRF" represents the RCRF slot tagging models that use the same feature as "RNN" (Yao et al., 2014b). "joint-RNN" (Shi et al., 2015) also uses the same features as "RNN" and "RCRF". However, "joint-RNN" implicitly makes use of query domain, intent and slot information by training the domain classifier, intent classifier and slot labeling jointly via multi-task learning.

Overall, the proposed RSVM obtains significant improvement over CRF, RNN,RCRF and joint-RNN on alarm, communication, note and reminder (z-test $p-value < 5E-5$). On the calendar, places and weather, RSVM achieves similar performance as joint-RNN. Even joint-

RNN is built on the basis of conventional RNN using local normalization, it actually takes the sequence representation information implicitly from domain and intent classification. However, in ondevice domain, RCRF performs the best and the proposed RSVM model performs even worse than CRF. We notice that, in ondevice model, user queries tend to be short, with on average 2.4 words in a query, shown in Table. 1. Also the loss function in the proposed model only uses the top and the second most hypothesis, which may be less informative, especially with short sentences, as compared to using all hypothesis in RCRF.

model	alarm	calen	commu	note
CRF	93.2	93.7	91.6	76.5
RNN	93.8	95.2	92.5	77.0
RCRF	93.9	95.9	93.2	76.9
joint-RNN	94.9	96.4	92.9	74.9
RSVM-min	95.8	95.8	93.3	86.9
RSVM-aver	95.9	96.3	93.9	88.3
RSVM-max	96.2	96.6	94.9	89.7

Table 4: F1 score comparison on different slot tagging models on alarm, calendar, communication and note.

model	ondev	place	remin	weath
CRF	98.2	87.5	93.3	95.7
RNN	98.4	88.4	90.5	95.0
RCRF	98.8	88.3	91.9	94.6
joint-RNN	98.6	90.6	92.4	96.7
RSVM-min	97.4	88.2	94.3	96.2
RSVM-aver	97.7	89.7	94.5	96.6
RSVM-max	97.9	90.8	95.1	96.8

Table 5: F1 score comparison on different slot tagging models on ondevice, places, reminder and weather.

Table. 6 gives the overall performance comparison of different models in internal live log dataset using the weighted average F1 score over all domains. In this table, we can find that the proposed RSVM on average can achieve 0.6% and 0.7% F1 score improvement over joint-RNN and RCRF, respectively.

3.6 Training Speed Up In RSVM

Using max-margin criteria, backward weight updating only happens to support vector samples. While using cross-entropy or maximum likelihood based training criteria, backward weight updating has to sweep over the whole training data. Fig. 2 shows that RSVM can substantially speed up the model training by skipping the backward weight updating for non-support vector samples. As depicted in Fig. 2, RSVM only executes backward weight updating for 337 training samples (7% of whole training data) at epoch 20.

model	F1(%)
CRF	92.6
RNN	92.8
RCRF	93.9
joint-RNN	94.0
RSVM-min	94.0
RSVM-average	94.6
RSVM-max	95.2

Table 6: The weighted average F1 score of different slot tagging models over all the domains.

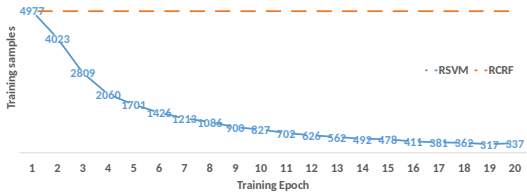

Figure 2: Training sample usage comparison for RSVM and RCRF on ATIS data in backward weight updating in each training epoch.

4 Conclusions

We have proposed a recurrent support vector machine (RSVM) which applies the structured SVM on top of the conventional RNN for slot tagging. Different from previous RNN sequence training approaches that use maximum conditional likelihood as objective function, the proposed method uses sequence level max-margin criterion with hard loss function. The model is trained to discriminate the score of ground-truth slot sequences with respect to other competing slot sequences by a margin. Viterbi algorithm is used in decoding to select a slot sequence that gives the largest score. To verify the performance of the proposed method, three datasets, namely ATIS dataset, CoNLL 2000 Chunking dataset and Cortana live log dataset, were used. The proposed RSVM achieved a new state-of-the-art performances on these datasets. In addition, RSVM showed substantial training speed up by skipping the weight updating for non-support vector training samples. On ATIS data, after 20 epoches, backward weight updating only happened for almost 7% of whole training samples.

The proposed RSVM is built on top of conventional RNN structure. Though RSVM doesn't have advanced topology used in LSTM and RNN-em, it achieves comparable or better performances. Therefore, the improvement comes from its sequence level max-margin criterion. For future works, we plan to apply the structured SVM on top of other advanced models.

References

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Renato De Mori. 2007. Spoken language understanding: a survey. In *The Proceedings of IEEE Workshop on Automatic Speech Recognition Understanding*, pages 365–376.

Anoop Deoras and Ruhi Sarikaya. 2013. Deep belief network based semantic taggers for spoken language understanding. In *ISCA Interspeech*. ISCA, September.

Marco Dinarelli, Alessandro Moschitti, and Giuseppe Riccardi. 2008. Joint generative and discriminative models for spoken language understanding. In *The Proceeding of Spoken Language Technology Workshop*, pages 61–64.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. 2013. Speech recognition with deep recurrent neural networks. In *The Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649.

Alex Graves. 2012. *Supervised Sequence Labelling with Recurrent Neural Networks*, volume 385. Springer.

Yulan He and S. Young. 2003. A data-driven spoken language understanding system. In *The Proceedings of Automatic Speech Recognition and Understanding*, pages 583–588.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990a. The atis spoken language systems pilot corpus. In *The Proceedings of the DARPA Speech and Natural Language Workshop*, pages 96–101.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990b. The atis spoken language systems pilot corpus. In *The Proceedings of the Workshop on Speech and Natural Language*, pages 96–101.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. 2009. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59.

Taku Kudo and Yuji Matsumoto. 2000. Use of support vector learning for chunking identification. In *The proceedings of Conference on Natural Language Learning (CoNLL) and Second Learning Language In Logic WorkShop*.

Taku Kudo and Yuji Matsumoto. 2001. Chunking with support vector machines. In *The Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*, pages 1–8.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Flexible text segmentation with structured multilabel classification. In *The Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 987–994.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, and Geoffrey Zweig. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pages 530–539.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *The Proceedings of Interspeech*, pages 1045–1048.

Nam Nguyen and Yunsong Guo. 2007. Comparisons of sequence labeling algorithms and extensions. *International Conference on Machine Learning*, pages 681–688.

Constantinos Panagiotakopoulos and Petroula Tsampouka. 2013. The stochastic gradient descent for the primal l1-svm optimization revisited. *CoRR*.

Baolin Peng and Kaisheng Yao. 2015. Recurrent neural networks with external memory for language understanding. *CoRR*, abs/1506.00195.

Juan Antonio Prez-ortiz, Mikel L. Forcada, and Departament De Llenguatges I Sistemes. 2001. Part-ofspeech tagging with recurrent neural networks. In *The Proceedings of the International Joint Conference on Neural Networks*.

Nathan Ratliff, J. Andrew (Drew) Bagnell, and Martin Zinkevich. 2007. (online) subgradient methods for structured prediction. In *Eleventh International Conference on Artificial Intelligence and Statistics (AIStats)*.

Fei Sha and Fernando Pereira. 2005. Shallow parsing with conditional random fields. In *The Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics and Human Language Technologies*.

Yangyang Shi, Kaisheng Yao, Hu Chen, Yi-Cheng Pan, Mei-Yuh Hwang, and Baolin Peng. 2015. Contextual spoken language understanding using recurrent neural networks. In *The Proceedings of International Conference on Acoustics, Speech and Signal Processing*.

Xu Sun, Louis-Philippe Morency, Daisuke Okanohara, and Jun'ichi Tsujii. 2008. Modeling latent-dynamic in shallow parsing: A latent conditional model with improved inference. In *The Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 841–848.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large margin methods for structured and interdependent output variables. *Journal Of Machine Learning Research*, 6:1453–1484.

Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*.

Ye-Yi Wang, Li Deng, and Alex Acero. 2005. Spoken language understanding — an introduction to the statistical framework. *IEEE Signal Processing Magazine*, 22(5):16–31.

Peilu Wang, Yao Qian, Frank K. Soong, Lei He, and Hai Zhao. 2015. A unified tagging solution: Bidirectional LSTM recurrent neural network with word embedding. *CoRR*.

P.J. Werbos. 1990. Backpropagation through time: what it does and how to do it. *The Proceedings of the IEEE*, 78(10):1550–1560.

Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular CRF for joint intent detection and slot filling. In *The Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 78–83.

Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu. 2013. Recurrent neural networks for

language understanding. In *The Proceedings of Interspeech*, pages 2524–2528.

Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi. 2014a. Spoken language understanding using long short-term memory neural networks. In *The Proceedings of IEEE workshop on Spoken Language Technology*, pages 189–194.

Kaisheng Yao, Baolin Peng, Geoffrey Zweig, Dong Yu, Xiaolong Li, and Feng Gao. 2014b. Recurrent conditional random field for language understanding. In *The Proceedings of International Conference of Acoustic, Speech and Signal Processing*, pages 4105–4109.

Tong Zhang, Fred Damerau, and David Johnson. 2001. Text chunking based on a generalization of winnow. *Journal of Machine Learning Research*, 2:615–637.

Shi-Xiong Zhang, Chaojun Liu, Kaisheng Yao, and Yifan Gong. 2015. Deep neural support vector machines for speech recognition. In *The Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4275 – 4279, April.