



Using Word Confusion Networks for Slot Filling in Spoken Language Understanding

Xiaohao Yang, Jia Liu

Tsinghua National Laboratory for Information Science and Technology
 Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

yangxiaohao@gmail.com, liuj@tsinghua.edu.cn

Abstract

Semantic slot filling is one of the most challenging problems in spoken language understanding (SLU) because of automatic speech recognition (ASR) errors. To improve the performance of slot filling, a successful approach is to use a statistical model that is trained on ASR one-best hypotheses. The state of the art models for slot filling rely on using discriminative sequence modeling methods, such as conditional random fields (CRFs), recurrent neural networks (RNNs) and the recent recurrent CRF (R-CRF) model. In our previous work, we have also proposed the combination model of CRF and deep belief network (CRF-DBN). However, they are mostly trained with the one-best hypotheses from the ASR system. In this paper, we propose to exploit word confusion networks (WCNs) by taking the word bins in a WCN as training or testing units instead of the independent words. The units are represented by vectors composed of multiple aligned ASR hypotheses and the corresponding posterior probabilities. Before training the model, we cluster similar units that may originate from the same word. We apply our proposed method to the CRF, CRF-DBN and R-CRF models. The experiments on ATIS corpus show consistent improvements of the performance by using WCNs.

Index Terms: spoken language understanding, slot filling, word confusion network, conditional random field, deep belief network, recurrent neural network

1. Introduction

The semantic parsing of input utterances in SLU typically consists of three tasks: domain detection, intent determination and slot filling. Slot filling aims at parsing semantic slots from the results of ASR [1] and is typically modeled as a sequence classification problem in which sequences of words are assigned semantic class labels. For example, users ask for the flight information when booking tickets by the utterance “I want to fly to Denver from Boston tomorrow”. In this case, slot filling is expected to extract semantic slots and the associated values of flight information such as *Departure=Boston*, *Destination=Denver* and *Departure Date=tomorrow*.

The state-of-the-art approaches for slot filling rely on statistical machine learning models. These approaches exploit traditional discriminative models such as maximum entropy markov models (MEMM) [2] and conditional random fields (CRFs) [3] or recent deep neural network models such as deep belief networks (DBNs) [4], convolutional neural networks (CNNs) [5], recurrent neural networks (RNN) [6, 7] and recurrent CRF (R-CRF) [8]. The combination of DBN and CRF model is presented in our previous work [9], achieving the state of art performance for slot filling task.

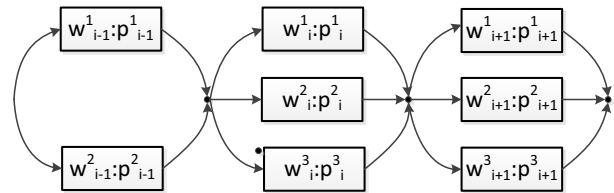


Figure 1: An example of word confusion network (WCN).

Most slot filling models are trained with the ASR one-best results instead of manual transcriptions in order to model the nature of recognition errors [4]. However, extracting target semantic slots with simple one-best hypotheses is still challenging. This paper aims at using the word confusion network (WCN) which contains more information than one-best lists for a more robust slot filling system.

WCNs have first been exploited to improve quality of ASR results [10] and applied to many spoken language processing tasks, including SLU tasks [11, 12]. Recent papers [13, 14] proposed a novel approach for training CRF models using n -gram features extracted from the WCNs. In this paper, going a step further, we propose a general methodology for training and evaluation based on WCNs which can apply to various models. This is done by regarding the word bins in the WCN as vectors composed of associated posterior probabilities. Based on the assumption that the same word tend to produce similar word bins whether or not the word is correctly recognized, we cluster the word bins in WCNs according to the distance between the vectors of bins. Thus the WCNs can be represented as the sequences of the cluster IDs. Then a variety of modeling approaches can be used for training and recognizing, such as the CRF model, R-CRF model and our proposed DBN-CRF model.

2. Word confusion networks

WCNs are compact representations of lattices in which competing words at the same approximate time stamp in ASR are aligned within the same position [10]. A posterior probability to measure the confidence of the result is assigned to each word. Figure 1 shows the structure of a WCN. In this example, there are three competing words w_i^1, w_i^2, w_i^3 at the position i . These words are assigned with associated posterior probabilities p_i^1, p_i^2, p_i^3 . At each position, the summation of the posterior probabilities is one as $\sum_j p_i^j = 1$. These bundled words and their corresponding probabilities at the position i is called Bin_i .

In order to exploit WCNs as units instead of words to train a model, we need to represent the WCN in a proper way. By

10.21437/Interspeech.2015-47

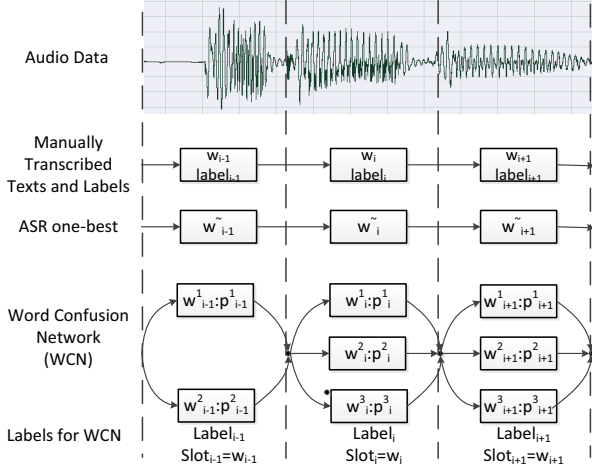


Figure 2: Alignment of the WCNs and the corresponding semantic labels.

considering the posterior probabilities of the words which don't appear in Bin_i as 0, we represent Bin_i as a V -dimension vector \mathbf{b}_i and $\mathbf{b}_i = (p_i^1, \dots, p_i^V)$. V is the size of the vocabulary used in the ASR system which generated the WCNs.

3. Using WCNs for slot filling

Noticing that the traditional slot filling systems are mostly trained on word sequences with associated labels, this paper aims at training a slot filling system on labeled WCNs which are sequences of word bins. Due to the difference between the word sequences and bin sequences, we implement the system in the following steps.

3.1. Labeling word confusion networks

Properly labeled data is essentially required when training a statistical model. In most traditional slot filling systems, manually transcribed texts or one-best results are labeled by semantic slots word-by-word. However, labeling word confusion networks is not as simple as labeling texts. We start with the training data which consist of audio data and the associated manually transcribed texts. The texts are annotated with the semantic slots. By performing a forced alignment between the audio data and the transcribed texts, each word and the assigned semantic slot are both tagged with time stamps. Then audio data is recognized by the ASR system and the one-best result is also tagged with time stamps. By comparing the time stamps between the one-best result and the transcribed texts, semantic slots are labeled for the words in the one-best result. For the WCNs, we can also label each bin with the semantic slot according to the time stamps in the same way. Therefore, each bin in the WCN is assigned a slot and a value ($Bin_i : Slot_i = w_i$). Figure 2 shows an overview of the labeling process.

3.2. Clustering

Now we have the bin sequences and the semantic label sequences and each bin is represented with a vector. Noticing that the same word should produce similar bins in the WCN, we cluster the bin vectors and each cluster contains the bins which are probably produced by the same word. Additionally, we find that the same mis-recognized word also produces similar bins,

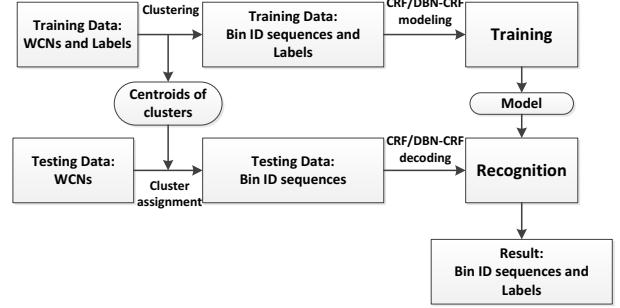


Figure 3: Flow of training and recognition with WCNs.

which can help us extract as much as possible information from the ASR results. In fact, the number of clusters is usually bigger than the size of vocabulary since the same word in different context may split into different clusters.

Given two vectors, cosine distance and Euclidean distance can be used as the distance metric. We use cosine similarity here as the metric of similarity. The cosine similarity between the two bin vectors \mathbf{b}_i and \mathbf{b}_j is defined as

$$sim(i, j) = \frac{\mathbf{b}_i \cdot \mathbf{b}_j}{|\mathbf{b}_i| |\mathbf{b}_j|} \quad (1)$$

We cluster all of the bins in the WCNs of the training data into K clusters using the k -means clustering or the repeated bisection algorithm [15]. K is a hyper-parameter in the experiments.

3.3. Training

Each bin has a cluster ID after clustering. Then the training data can be represented as pairs of a sequence of cluster IDs and a sequence of semantic labels. Based on these pairs, we can train a model to predict a label sequence from a cluster ID sequence in various frameworks such as CRF [3, 16], DBN-CRF [9] and R-CRF [8].

3.4. Evaluation

After clustering we have K centroid vectors. Before we predict the semantic tags of a WCN using the trained model, each bin in the WCN is assigned to one of the nearest clusters according to the similarity between the bin and the centroid vectors. Thus the evaluation data is also represented by cluster ID sequences. We assign the slots to the cluster ID sequences using the trained model and fill the slots with the 1-best words from the WCN bins.

Figure 3 shows the whole training and evaluation process with WCNs.

3.5. Considering contexts of bins in a WCN

The above representation of the WCN bin with the corresponding vector can take into account the acoustic feature of a word in various acoustic environments. In order to model the language feature of a word, we can consider the contexts of bins in a WCN. For example, each Bin_i is represented with a vector \mathbf{b}_i with the dimension V , the size of the vocabulary. By considering the previous and the next bins, Bin_i can be represented with a vector of $3V$ dimensions like $(\sigma \mathbf{b}_{i-1}, \mathbf{b}_i, \sigma \mathbf{b}_{i+1})$, where σ is a weighting factor which is another hyper-parameter in our experiment. If $\sigma = 0$, we experiment without contexts.

4. Applied to models

Noticing that our proposed approach can be seen as a preprocessing step for training and recognition, we can train a model in different frameworks. The traditional discriminative model CRF, the hybrid model DBN-CRF in our previous work [9] and R-CRF [8] are used in this work to evaluate the effect of the proposed approach.

4.1. CRF modeling with WCNs

CRF is a discriminative sequence model which can frame slot filling task in SLU as a sequence labeling problem to obtain the most likely slot sequence given the input sequence:

$$\hat{Y} = \underset{Y}{\operatorname{argmax}} P(Y|X) \quad (2)$$

where $X = x_1, \dots, x_T$ is the input word sequence and $Y = y_1, \dots, y_T$ is the output label sequence. The goal is to obtain the label sequence \hat{Y} which has the highest conditional probability. CRF is shown to outperform other discriminative models due to its global sequence training ability. In the basic linear CRF model, the above conditional probability $P(Y|X)$ can be defined in an exponential form:

$$P(Y|X) = \frac{1}{Z(X)} \exp\left(\sum_k \lambda_k f_k(y_{t-1}, y_t, x_t)\right) \quad (3)$$

where the function f_k represents the input features extracted from training data and the label transition features with associated weights λ_k . $Z(X)$ is the normalization term [3]. The features $\{f_k\}$ are predefined in advance according to the input sequences and their labels, and the weights $\{\lambda_k\}$ are learned during the training process. After the parameters are optimized with annotated training data, the most likely label sequence \hat{Y} can be determined using the Viterbi algorithm. Note that, each label y_t depends on the whole sequence X , instead of corresponding observations x_t . CRF model can overcome the *label bias* problem, which is the main advantage against local models like MEMM [2] or the latest DBN [4].

4.2. DBN-CRF modeling with WCNs

While CRF exploits the sequence training approach and can alleviate the *label bias* problem in locally normalized models, the input features are manually defined and cannot be learned automatically. Thus we use DBNs to generate the features for the CRF, which is called DBN-CRF [9]. Figure 4 shows the DBN-CRF model architecture. The input sequences are bin cluster ID sequences instead of word sequences.

4.3. R-CRF modeling with WCNs

In the recurrent CRF model [8], a RNN is used to generate the input features for a CRF. The features used are the RNN scores before softmax normalization to avoid label bias problem. In this paper, we use WCNs to train the R-CRF model as an extension of the work in [8]. Figure 5 shows the R-CRF model architecture.

5. Experiments

We conduct experiments to verify whether the performance of slot filling is improved by using WCNs for training or recognition. In order to confirm our proposed method is relatively

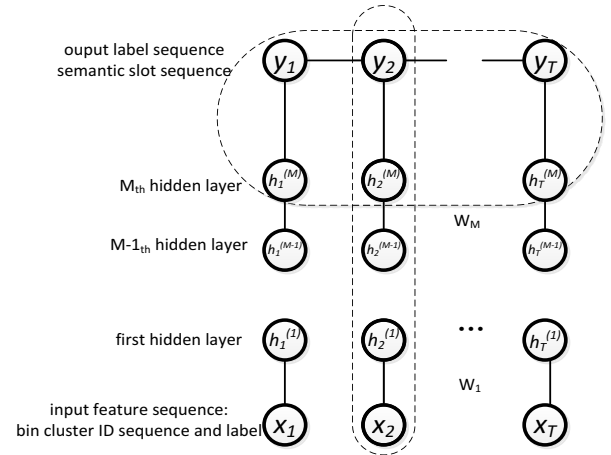


Figure 4: DBN based CRF model using WCNs.

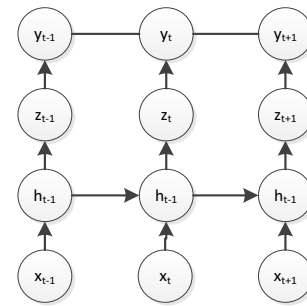


Figure 5: Recurrent CRF model using WCNs.

general and unaffected by the modeling approach, we experiment with three different kinds of models, CRF, DBN-CRF and R-CRF.

5.1. Experimental setup

We evaluate our proposed method on slot filling task with the most widely used data set for SLU research, the ATIS corpus [17]. The training set contains 4978 utterances with the transcribed texts and corresponding semantic labels while the test set contains 905 utterances also with texts and labels. 1000 utterances of the training set are held out as the development set to tune the hyper-parameters in the experiment. Additional 8000 unlabeled utterances in the same scenario are used to pre-train the RBMs for DBN initialization. The ATIS corpus are annotated using semantic frames in In-Out-Begin (IOB) representation which is shown in Table 1. Notice that “*dc*” represents “*departure city*” and “*ac*” represents “*arrival city*”.

To obtain the ASR one-best hypotheses and WCNs for the above 3 data sets, we prepared an ASR systems [18]. The vocabulary size of the dictionary is 19800, meaning that the dimension of the vector representing the bin of the WCN is also 19800. This dimension is 19800×3 when the contexts of the bin are considered. The Word Error Rate (WER) of the ASR one-best of the test set is 28.7%.

5.2. Feature selection

The input features for the CRF, DBN-CRF and R-CRF are extracted from the labeled WCN bin cluster ID sequences. We

Table 1: ATIS corpus example with IOB annotation.

Sentence	flights	from	Denver	to	New	York
Labels	O	O	B-dc	O	B-ac	I-ac

consider previous two cluster IDs, current cluster ID, next two cluster IDs as the basic feature and use the “*1 of N coded*” binary vectors to represent the feature. If the number of the clusters is K , the input feature can be represent as a vector of size $K \times 5$, with 5 bits switched on.

For the CRF framework, we use CRFsuite for our experiment since the feature generation code is simple and general so we can change or add an arbitrary number of features (<http://www.chokkan.org/software/crfsuite/>). We use Stochastic Gradient Descent (SGD) optimization for the CRF training.

For the DBN-CRF framework, we choose three hidden layers of 100 – 200 – 100 units as basic DBN architecture, with additional input layer and output layer. The threshold for weight constraining is 2 [4]. The training process is divided into 2 phases, pre-training step and weights tuning step with back propagation.

For the R-CRF framework, the dimension of hidden layer is 200. We implement a forward-backward algorithm during training and Viterbi algorithm during decoding [8].

5.3. Evaluation

We evaluate the total F-measure for all the 79 semantic slots. The results are shown in Table 2.

For the comparison of performance, we estimate the models trained and evaluated on manually transcribed texts, the ASR one-best hypotheses and the proposed WCNs respectively.

In the experiments, we have two hyper-parameters which are the number of clusters K and the context weighting factor σ . They are tuned with the development set in the experiment. For the number of clusters K , we increase it from 10000 (smaller than the vocabulary size 19800) to 60000 (larger than the vocabulary size). For the contexts of WCN bins, we choose the weighting factor σ from $\{0.2, 0.5, 0.7\}$. Figure 6 shows the total F-measure in the development set when varying K and σ . In the experiment, we figured out that $K = 51000$ and $\sigma = 0.2$ achieve the best performance and we use these parameter values in the evaluation.

In [13] and [14], n -gram features are extracted from WCNs for training a slot filling system. We repeat the work, using WCN bins with size of 3 and the corresponding trigram features for training.

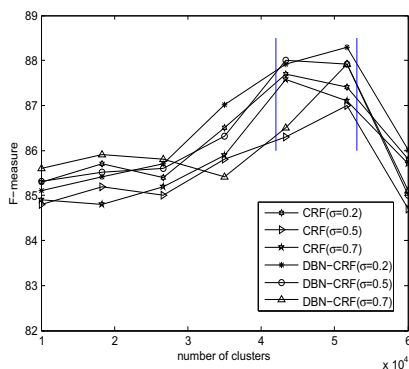


Figure 6: F-measure in development set when varying K and σ .

Table 2: F-measure in evaluation set using different methods.

Training / Evaluation / Parameters	CRF	DBN-CRF	R-CRF
Manually transcribed text / ASR one-best	0.786	0.802	0.852
ASR one-best / ASR one-best	0.794	0.808	0.865
WCNs (no contexts) / WCNs (no contexts) / $K = 51000$	0.842	0.858	0.891
WCNs (with contexts) / WCNs (with contexts) / $K = 51000, \sigma = 0.2$	0.877	0.883	0.903
WCNs / WCNs [13, 14]	0.861	-	-

5.4. Discussion

Taking an overview of the results in Table 2, our proposed approach shows consistent improvements in CRF, DBN-CRF and R-CRF model. The R-CRF model with WCNs and consideration of contexts achieves the best performance.

(1) Comparing the 2nd and the 3rd rows in Table 2, the models trained on the ASR one-best results are slightly superior. This is because the training data and the test data match and the trained model can take into account the ASR errors.

(2) Comparing the 3rd and the 4th rows, the F-measure improved by using WCNs for both training and evaluation. The improvement illustrates that the model trained with WCNs can effectively recover much more information from the ASR errors than one-best results.

(3) Comparing the 4th and the 5th rows, the F-measure improved by considering contexts of WCN bins. The improvement illustrates that the richer representation of the context feature is helpful in slot filling.

(4) The comparison of last two rows show that our method of exploiting WCNs is more effective than the previous work [13, 14]. The primary reason is that we take into account the bins with full size in a WCN while the previous work used WCN bins with size of 3 which may compromise the accuracy of slot filling.

6. Conclusion and future work

In this paper, we proposed an approach to exploit word confusion networks for slot filling task in spoken language understanding. The key idea is that the same word can produce similar bins in WCNs whether or not the word is correctly recognized. The bins are clustered and the WCN is represented with a sequence of cluster IDs. Thus our proposed approach can be seen as a preprocessing step for modeling and recognition with various techniques. We conducted experiments with CRF, DBN-CRF and R-CRF models and observed that the proposed method consistently improve performances on ATIS dataset.

Future work will explore whether additional dense features such as word embeddings can boost the clustering process, further improving the performance of our method.

7. Acknowledgements

This work is supported by National Natural Science Foundation of China under Grant No. 61370034, No. 61273268, No. 61005019 and No. 61005017.

8. References

- [1] G. Tur and R. De Mori, *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011.
- [2] A. McCallum, D. Freitag, and F. C. Pereira, "Maximum entropy markov models for information extraction and segmentation." in *ICML*, 2000, pp. 591–598.
- [3] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," *ICML*, 2001.
- [4] A. Deoras and R. Sarikaya, "Deep belief network based semantic taggers for spoken language understanding," in *Proceedings of Interspeech*, 2013.
- [5] P. Xu and R. Sarikaya, "Convolutional neural network based triangular crf for joint intent detection and slot filling," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 78–83.
- [6] G. Mesnil, X. He, L. Deng, and Y. Bengio, "Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding," in *Proceedings of Interspeech*, 2013.
- [7] K. Yao, G. Zweig, M.-Y. Hwang, Y. Shi, and D. Yu, "Recurrent neural networks for language understanding," in *Proceedings of Interspeech*, 2013, pp. 104–108.
- [8] K. Yao, B. Peng, G. Zweig, D. Yu, X. Li, and F. Gao, "Recurrent conditional random field for language understanding," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2014.
- [9] X. Yang and J. Liu, "Deep belief network based crf for spoken language understanding," in *Proceedings of ISCSLP*, 2014, pp. 49–53.
- [10] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech and Language*, vol. 14, pp. 373–400, 2000.
- [11] D. Hakkani-Tur, F. Bechet, G. Riccardi, and G. Tur, "Beyond asr 1-best: Using word confusion networks in spoken language understanding," *Computer Speech and Language*, vol. 20, pp. 495–514, 2006.
- [12] G. Tur, D. Hakkani-Tur, and G. Riccardi, "Extending boosting for call classification using word confusion networks," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, vol. 1. IEEE, 2004, pp. 1–437.
- [13] G. Tur, A. Deoras, and D. Hakkani-Tur, "Semantic parsing using word confusion networks with conditional random fields," in *INTERSPEECH*, 2013, pp. 2579–2583.
- [14] M. Henderson, M. Gasic, B. Thomson, P. Tsiakoulis, K. Yu, and S. Young, "Discriminative spoken language understanding using word confusion networks." in *SLT*, 2012, pp. 176–181.
- [15] Y. Zhao and G. Karypis, "Hierarchical clustering algorithms for document datasets," *Data Mining and Knowledge Discovery*, vol. 10, pp. 141–168, 2005.
- [16] C. Raymond and G. Riccardi, "Generative and discriminative algorithms for spoken language understanding." in *INTERSPEECH*, 2007, pp. 1605–1608.
- [17] P. Price, "Evaluation of spoken language systems: The atis domain," in *Proceedings of the Third DARPA Speech and Natural Language Workshop*. Morgan Kaufmann, 1990, pp. 91–95.
- [18] S. F. Chen, B. Kingsbury, and L. Mangu, "Advances in speech transcription at ibm under the darpa ears program," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, pp. 1596–1608, 2006.