A FACTORIZATION NETWORK BASED METHOD FOR MULTI-LINGUAL DOMAIN CLASSIFICATION

Yangyang Shi, Yi-Cheng Pan, Mei-Yuh Hwang, Kaisheng Yao, Hu Chen, Yuanhang Zou, Baolin Peng

Microsoft

ABSTRACT

In many spoken language understanding systems (SLUS), domain classification is the most crucial component, as system responses based on wrong domains often yield very unpleasant user experiences. In multi-lingual domain classification, the training data for some poor-resource languages often comes from machine translation. Some of the higher order *n*-gram features are distorted during machine translation. Feature co-occurrence becomes reliable feature in multi-lingual domain classification. In this paper, in order to effectively model feature co-occurrences, we propose Factorization Networks that are combinations of Factorization Machines (FMs) with Neural Networks (NNs). FNs extend the linear connections from the input feature layer to the hidden layer in NNs to factorization connections that represent the weights of feature co-occurrences using factorized method. In addition to FNs, we also propose a hybrid model that integrates FNs, NNs and Maximum Entropy (ME) models together. The component models in the hybrid model share the same input features. Based on two data sets (ATIS data set and Microsoft Cortana Chinese data), the proposed models shows promising results. Especially for large Microsoft Cortana Chinese data which is translated from well annotated English data, FNs using unigram, class and query length features achieve more than 20% relative error reduction over linear (SVMs).

Index Terms— Factorization Networks, Spoken Language Understanding, Domain Classification

1. INTRODUCTION

Spoken language understanding (SLU) is an emerging area that involves speech and natural language processing. In a typical SLU system, human input queries are first classified into different domains. The intents and slots of the queries are further determined by specific domain dependent models. In such an architecture, domain classification is the most crucial component for user experiences. An error in domain classification will trigger wrong intent and slot extraction models.

SLU in multi-lingual conditions is a challenge research, especially for poor-resource languages and new domains. Training an SLU system usually requires supervised data that involves expensive and time consuming manual data collection and domain/intent/slot annotation. However, multilingual SLU applications usually have one language with rich resource and the other languages that don't have much annotation and labeled data. A widely used approach is based on machine translation (MT), which translates queries from the resource rich language to the queries from the resource poor language.

When the MT systems are properly used, the translated queries have reliable unigram statistics. But higher order statistics are often distorted due to alignment and word reordering. An example is shown in Fig.1 that compares a translated query with a real query. In real query, the Chinese word *tomorrow* is put at the beginning of a query, but the translated query puts the word *tomorrow* to the end. Therefore, the valid Chinese bigram *tomorrow I* is lost in the translated query.

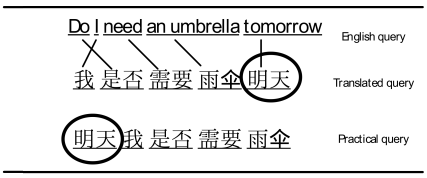

Fig. 1. A noisy query generated from machine translation.

However, one observation we have from the figure is that the word *I* co-occurs with the word *tomorrow* even though they are not right next to each other. Such observation is called co-occurrence.

Support Vector Machines (SVMs) are widely used to model feature co-occurrence using polynomial kernels. However, polynomial-kernel SVMs model each co-occurrence by completely separated weights, which require sufficient numbers of co-occurrences to appear in training data for reliable estimation of the weights. For resource poor languages, it is impossible to have suchlarge number of co-occurrences.

In this paper, we propose using factorization networks (FNs) to model co-occurrences. A FN combines neural networks (NNs) with factorization machines (FMs) [1]. It models co-occurrences by factorizing the inter-feature dependency, which makes FNs more trainable, and generalizable to unobserved feature co-occurrences. For example, suppose we

have two queries in the training data: *Where is the nearby McDonalds and KFC* and *How can I go to the parking-lot near KFC*. From the co-occurrence of (*McDonalds*, *KFC*) and (*parking-lot*, *KFC*), the FNs can model the co-occurrence of (*parking-lot*, *McDonalds*) even though it does not appear in the training data. The generalization capability compensates the insufficiency of SLU training data.

2. RELATED WORK

In this section, the related work about Factorization Machines (FMs) and Neural Networks (NNs) will be discussed.

FMs are first introduced by Steffen Rendle [1] to estimate higher order feature co-occurrences in huge sparse problems by combining polynomial-kernel SVMs [2] with factorization models. In that setting, SVMs fail to model feature co-occurrences effectively due to data sparseness. SVMs directly and independently estimate the parameter for each co-occurrence, yet most of the co-occurrences are not observed in the training data. The basic idea of FMs is to break the independence of the feature co-occurrences by factorization in which the parameters of the feature co-occurrences are represented by dot products of latent vectors that represent the individual features which have inter-dependency. In addition to using Stochastic Gradient Descent (SGD) [3], [4] applies alternatively the least-square method and Bayesian method in training FMs. In this paper, we apply the factorization method in NNs to model the feature co-occurrence for domain classification in SLU, especially for multi-lingual domain classification.

In the recent years we have witnessed a boost of applying NN-based technology in speech recognition and natural language processing [5–10]. In particular, some of these applications are targeting at improving SLU domain classification [11–14] using deep architecture. In addition to using deep architecture, some of previous work [15–17] proposed to use nonlinear connections between different layers in the network. In this paper, we propose to use second-order FMs connecting the input features to the hidden layer.

3. FACTORIZATION NETWORKS

3.1. Factorization Networks

A natural language query t is represented as a real value vector $\mathbf{x}^{(t)} \in R^L$. The task of domain classification is having a function that maps query $\mathbf{x}^{(t)}$ to a domain label.

Illustrated in the Fig.2, a FN has three layers: input feature layer, hidden layer and output layer. The input feature layer extracts second-order co-occurrence and first order occurrence features from the inputs $\mathbf{x}_i$. The output from this layer is therefore represented as

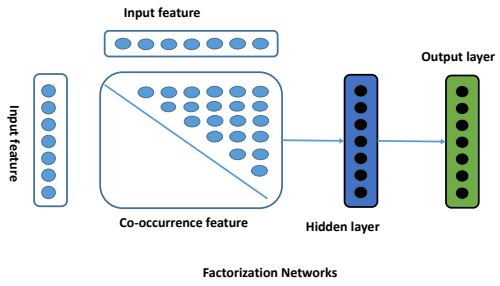

Fig. 2. The Factorization Networks. Blue nodes use sigmoids as the activation function, while green nodes use softmax.

$$\mathbf{h}_k = \mathrm{G}_k + \sum_{i=1}^{L} \mathbf{x}_i w_{ki}, \quad (1)$$

$$\mathrm{G}_k = \sum_{i=1}^{L} \sum_{j=i+1}^{L} \langle \beta_{ki}, \beta_{kj} \rangle \mathbf{x}_i \mathbf{x}_j \quad (2)$$

where the k-th element is a linear weighted summation of the input occurrence. The input feature weight for the occurrence is w_{ki} for combining the i-th occurrence with the k-th element of the feature. $\langle \beta_{ki}, \beta_{kj} \rangle$ is the weight for co-occurrence of input i and j.

A factorization network rephrases the co-occurrence weight as follows:

$$\mathrm{G}_k = \frac{1}{2} \sum_{f=1}^{F} \left[(\sum_{i=1}^{L} \beta_{kif} \mathbf{x}_i)^2 - \sum_{i=1}^{L} \beta_{kif}^2 \mathbf{x}_i^2 \right] \quad (3)$$

where F specifies a factor size and is usually an order smaller than the input feature dimension L. By expanding $(\sum_{i=1}^{L} \beta_{kif} x_i)^2$ in Eq. 3 [1], each pair of co-occurrence $\mathbf{x}_i \mathbf{x}_j$ is weighted by $\sum_f \beta_{kif} \beta_{kjf}$. In other words, the weight for co-occurrence $\mathbf{x}_i \mathbf{x}_j$ can be computed as long as some co-occurrence $(\mathbf{x}_i \mathbf{x}_{j'})$ and $(\mathbf{x}_{i'} \mathbf{x}_j)$ exist in the training data. The number of parameters from the input layer to the hidden layer is HLF (the set of $\{\beta_{kif}\}$) + HL (the linear connections), where H is the number of hidden units.

In contrast, polynomial-kernel SVMs model the feature co-occurrences in the following way:

$$\mathrm{G}_s = \sum_{i=1}^{L} w_{ii} \mathbf{x}_i \mathbf{x}_i + \sum_{i=1}^{L} \sum_{j=i+1}^{L} \sqrt{2} w_{ij} \mathbf{x}_i \mathbf{x}_j, \quad (4)$$

where w_{ij} represents the weight for the co-occurrence of $\mathbf{x}_i \mathbf{x}_j$. Since w_{ij} in Eq. 4 is explicitly estimated for $\mathbf{x}_i \mathbf{x}_j$, it requires the exact and sufficient number of co-occurrence of both $\mathbf{x}_i$ and $\mathbf{x}_j$ to have w_{ij} reliably estimated. In addition,

the factorization network has much smaller number of parameters than polynomial-kernel SVMs because H and F are much smaller than L. The forward computation of Eq.3 can be efficiently calculated in $O(FL)$.

Based on Eq.3, the gradient of the factorization parameters can be easily derived as

$$\frac{\mathrm{d}(\mathrm{G}_k)}{\mathrm{d}(\beta_{kif})} = \mathbf{x}_i(\sum_{j=1}^{L} \beta_{kjf}\mathbf{x}_j) - \beta_{kif}\mathbf{x}_i^2 \quad (5)$$

The hidden layer uses sigmoid function for activation. The softmax function is used in the output layer. Training FNs uses cross-entropy minimization criterion. During test, the index with the maximum score, which is the conditional probability of a label given inputs, is selected as the decoded domain label.

3.2. Hybrid Models

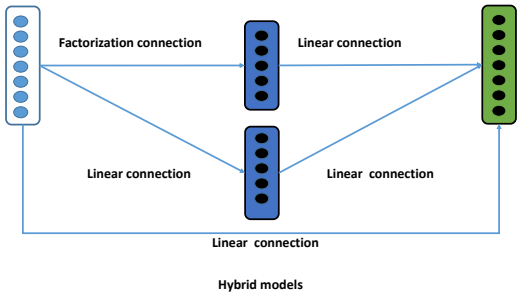

Fig. 3. The hybrid networks. Blue nodes use sigmoids as the activation function, while green nodes use softmax.

A hybrid model, shown in the Fig.3, consists of neural networks (NN), factorization networks (FNs) and maximum entropy (ME) models [18]. A maximum entropy model [19] directly connects the input feature to the activity of the output label. The neural network model combines the input feature linearly. The ME model has been applied in [19] to improve Recurrent Neural Network based language models. The proposed hybrid model uses ME in the same way as [19]. The FN models the second-order co-occurrence but have $O(FHL)$ number of parameters. The NN has $O(HL)$ number of parameters but only models the first order occurrence. Using a hybrid model corresponds to interpolation of the three models.

4. EXPERIMENTS

4.1. Data Sets

Two data sets are used to evaluate the proposed method. The ATIS dataset [20, 21] is mainly about air travel. It has 22 domains/intents such as airline, ground service and etc. There are 893 utterances for testing (ATIS-III, Nov93 and Dec94), and 4978 utterances for training (rest of ATIS-III and ATIS-II). In our experiment, we actually use 888 utterances in the testing by removing the out-of-vocabulary domain utterance such as (day time, flight no airline and airfare flight). In the training data, 170 utterances are randomly selected for validation. The training data has 899 unigram features, $6K$ bigram features and $13K$ trigram features.

The second data set that we use is a Microsoft internal Cortana data in Chinese that is translated from English version Cortana data. It has 8 domains: alarm, calendar, communication, note, reminder, weather, places and web-queries. We use 5.4 million queries for training and 23,300 queries for testing. The testing data is human annotated data. The vocabulary size of the training data is 144,800. 16,000 queries are randomly selected for validation and they are excluded from training. Unigrams, bigrams, trigrams, classes and query length are used as features to represent each query. The total number of unigrams, bigrams and trigrams are respectively 144,800, 4.3 million and 10.2 million.

4.2. Training Settings

The FNs and hybrid models are trained using stochastic gradient descent (SGD) with $L2$ regularization. The initial learning rate for SGD is 0.1. After each epoch of training, the model is tested on the validation data. If the likelihood on the validation data is not improved, learning rate gets halved. Training stops if validation set likelihood is not improved for two times.

4.3. Results On ATIS

The dimensions for NN and FN are both set to 22. The factor size in FN is 8. Table 1 shows the domain classification error rate of the proposed method, together with the results from linear SVM, SVM using Radial Basis Function kernel (RBF) [22], SVM using polynomial function kernel [22].

Results show that the best performance is obtained using unigram plus bigram features (1,2-gram). Using additional trigram feature actually degrades performances. This is probably due to over-fit. The best classification error rate is achieved by linear SVM, RBF SVM and a hybrid model using both FN and ME. With unigram features only, polynomial SVM has the lowest error rate.

With both unigram and bigram features, factorization alone ('FN') achieves better performance than the linear summation in NN. This gain is attained even with ME added. Compared against NN + ME, FN + ME has 21% relative error rate reduction.

model	1-gram	1,2-gram	1,2,3-gram
linear SVM	5.7	4.4	5.7
RBF SVM	5.6	4.4	4.8
polynomial SVM	5.0	4.5	5.2
FM	6.8	4.7	4.9
FN	6.4	4.5	4.6
NN	5.4	5.2	5.0
NN +ME	5.4	5.6	5.3
FN +ME	5.4	4.4	4.9
FN +ME +NN	5.3	4.5	4.9

Table 1. Domain classification error rates on ATIS dataset. "1-gram" stands for unigram feature. "1,2-gram' stands for unigram plus bigram feature. "1,2,3-gram" stands for unigram plus bigram and trigram feature.

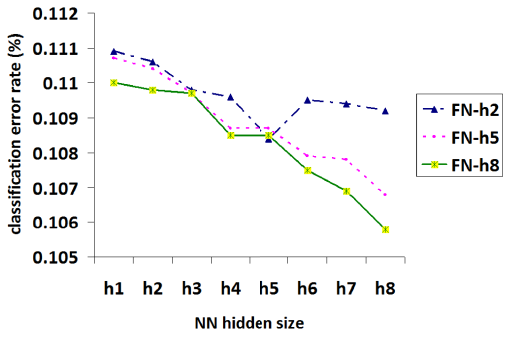

Fig. 4. Microsoft Internal Cortana Data classification error rates using hybrid models. X-axis represents the hidden size of NN neurons. "FN-h2" means 2 FN hidden neurons.

4.4. Results on Microsoft Internal Cortana Training Data

FN is expensive to use on large dataset, because of its memory consumption. We therefore uses the hybrid model described in Section 3.2. FNs and NNs have their own hidden layers. By maintaining a small hidden layer dimension for FN but a large hidden layer for NN, the hybrid model may efficiently model the co-occurrence without overfit and much memory. We plot classification error rate versus hidden layer dimension in Fig. 4. The models use unigram, class and query length features, and the factor size is 8. It is shown in the figure that lower classification error rate can be achieved using bigger hidden sizes. Our experiments show that the best performing hidden layer dimension is 8, which coincidentally is the number of domains.

The Microsoft Internal Cortana training Data is much bigger than ATIS data set. SVM with RBF kernel and polynomial kernels were not able to finish experiments because of large memory consumption. We therefore only use linear SVM as our baseline model. Table 2 compares the classification error rates of the proposed models with linear SVM models, which is trained using Liblinear [23] with $L1$ regularization and $L2$ loss function. Based on the previous results, hidden layer sizes for NN and FN, in addition to the factor size, are all set to 8.

The table shows that by modeling feature co-occurrence, the FN using unigram features is better than the linear SVM with bigram features, and achieves more than 20% error rate reduction relatively, compared to the SVM unigram model. Used together with ME and NN features, FN is able to obtain better performances. The hybrid model ('FN + ME + NN') using unigram plus bigram and trigram features obtains the lowest error rate of 9.7%, $7\%+$ relative error reduction over the SVM models with the same features. We also observed that maximum entropy features are useful for all of the evaluated neural network models. We also observed that NN has similar performances as SVM, especially with unigram and bigram features.

model	1+c+len	1,2+c+len	1,2,3+c+len
linear SVM	14.7	11.6	10.5
FM	13.9	11.8	10.5
FN	11.4	11.5	10.4
NN	14.8	11.6	11.0
NN +ME	11.0	10.4	10.3
FN +ME	10.7	10.6	9.9
FN +ME +NN	10.6	10.4	9.7

Table 2. Classification error rates ($\%$) using n-gram, class (c) and query length feature (len). "1-gram" stands for unigram feature. "1,2-gram' stands for unigram plus bigram feature. "1,2,3-gram" stands for unigram plus bigram and trigram feature.

5. CONCLUSIONS AND FUTURE WORK

Feature co-occurrence is a important feature in multi-lingual spoken language understanding domain classification, especially when the training data is obtained via machine translation where the high order *n*-gram features are distorted. To model the second order co-occurrence effectively, we have proposed a method based on factorization network. Compared with polynomial linear SVM that also models co-occurrence, the proposed method has much smaller model size. On ATIS dataset, the proposed method achieved similar performance with polynomial linear SVM. On Microsoft Internal Cortana Chinese data that was a large training data set obtained via machine translation, the proposed models performed significantly better than polynomial linear SVM.

6. REFERENCES

[1] S. Rendle, "Factorization machines.," in *IEEE International Conference on Data Mining*, 2010, pp. 995–1000.

[2] V. N. Vapnik, *The Nature of Statistical Learning Theory*, 1995.

[3] L. Bottou, "Online algorithms and stochastic approximations," in *Online Learning and Neural Networks*. Cambridge University Press, Cambridge, UK, 1998.

[4] S. Rendle, "Factorization machines with libfm," *ACM Transactions on Intelligent Systems and Technology*, vol. 3, no. 3, pp. 57, 2012.

[5] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing - Special Issue on Deep Learning for Speech and Language Processing*, vol. 20, pp. 33–42, 2012.

[6] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *Signal Processing Magazine*, vol. 29, pp. 82–97, 2012.

[7] H. Schwenk, "Continuous space language models," *Computer Speech and Language*, vol. 21, no. 3, pp. 492–518, 2007.

[8] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[9] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the International Conference on Machine Learning*, 2008, pp. 160–167.

[10] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[11] R. Sarikaya, G. E. Hinton, and B. Ramabhadran, "Deep belief nets for natural language call-routing," in *ICASSP*, 2011, pp. 5680–5683.

[12] L. Deng, G. Tür, X. He, and D. Z. Hakkani-Tür, "Use of kernel deep convex networks and end-to-end learning for spoken language understanding.," in *SLT*, 2012, pp. 210–215.

[13] G. Tür, L. Deng, D. Hakkani-Tür, and X. He, "Towards deeper understanding: Deep convex networks for semantic utterance classification," in *ICASSP*, 2012, pp. 5045–5048.

[14] P. Xu and R. Sarikaya, "Contextual domain classification in spoken language understanding systems using recurrent nerual network," in *ICASSP*, 2014, p. to appear.

[15] D. Yu and L. Deng, "Deep convex net: A scalable architecture for speech pattern classification," in *INTERSPEECH*, 2011, pp. 2285–2288.

[16] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep Fisher networks for large-scale image classification," in *Advances in Neural Information Processing Systems*, 2013, pp. 163–171.

[17] T. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "Pcanet: A simple deep learning baseline for image classification?," *CoRR*, vol. abs/1404.3606, 2014.

[18] K. Nigam, "Using maximum entropy for text classification," in *The proceedings of International Joint Conference on Artificial Intelligence Workshop on Machine Learning for Information Filtering*, 1999, pp. 61–67.

[19] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Cernocký, "Strategies for training large scale neural network language models," in *ASRU*, 2011, pp. 196–201.

[20] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, "The atis spoken language systems pilot corpus," in *Proceedings of the Workshop on Speech and Natural Language*, 1990, pp. 96–101.

[21] K. Yao, B. Peng, G. Zweig, D. Yu, X.Li, and F. Gao, "Recurrent conditional random field for language understanding," in *ICASSP*, 2014.

[22] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[23] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, "Liblinear: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.