Zero-Shot Learning and Clustering for Semantic Utterance Classification

Article · December 2013

Source: arXiv

Gokhan Tur
Microsoft

Dilek Hakkani-Tur
Microsoft

Larry P. Heck
Institute of Electrical and Electronics Engineers

Zero-Shot Learning and Clustering for Semantic Utterance Classification

Yann N. Dauphin[1] **Gokhan Tur**[2] **Dilek Hakkani-Tür**[2] **Larry Heck**[2]
[1]University of Montreal, Montreal, Canada
[2]Microsoft Research, Mountain View, CA, USA

Abstract

We propose two novel zero-shot learning methods for semantic utterance classification (SUC) using deep learning. Both approaches rely on learning deep semantic embeddings from a large amount of Query Click Log data obtained from a search engine. Traditional semantic utterance classification systems require large amounts of labelled data, whereas our proposed methods make use of the structure of the task to allow classification without labeled data. We also develop a zero-shot semantic clustering algorithm for extracting discriminative features for supervised semantic utterance classification systems. We demonstrate the effectiveness of the zero-shot semantic learning algorithm on the SUC dataset collected by [1]. Furthermore, we show that extracting features using zero-shot semantic clustering for a linear SVM reaches state-of-the-art result on that dataset.

1 Introduction

Conversational understanding systems aim to automatically classify the user request in one of the predefined semantic categories and extract related arguments [2]. Usually, supervised classification methods are used to estimate conditional probabilities, and a set of labeled utterances is used in training. Such systems typically use established classification algorithms, such as Boosting [3], support vector machines (SVMs) [4], or maximum entropy models [5].

Following the recent advances in deep learning techniques, in this paper, we present the application of deep networks trained with large amounts of implicitly annotated data for semantic utterance classification (SUC) in a conversational understanding system. In that respect, this study proposes a novel approach that is significantly different from the previous works which employ deep learning as an alternative classification technique [6, 1, 7]. The deep networks are trained using Bing search query click logs, which consists of user queries and associated clicked URLs, which ideally should reflect high level meaning of the queries. These networks are trained to obtain unstructured text embeddings, which provide the basis for zero-shot semantic clustering and learning.

In the next section, we formally define the task of semantic utterance classification. We review the related work on this task in Section 3 and explain query click logs in Section 4. Sections 5 and 6 present the zero-shot learning and clustering algorithms. In Section 7 we provide experimental results.

2 Semantic Utterance Classification

The semantic utterance classification (SUC) task aims at classifying a given speech utterance X_r into one of M semantic classes, $\hat{C}_r \in \mathcal{C} = \{C_1, \ldots, C_M\}$ (where r is the utterance index). Upon the observation of X_r, $\hat{C}_r$ is chosen so that the class-posterior probability given X_r, $P(C_r|X_r)$, is maximized. More formally,

$$\hat{C}_r = \arg\max_{C_r} P(C_r|X_r). \tag{1}$$

Semantic classifiers need to allow significant utterance variations. A user may say *"I want to fly from San Francisco to New York next Sunday"* and another user may express the same information by saying *"Show me weekend flights between JFK and SFO"*. Not only is there no *a priori* constraint on what the user can say, these systems also need to generalize well from a tractably small amount of training data. On the other hand, the command *"Show me the weekend snow forecast"* should be interpreted as an instance of another semantic class, say, "*Weather*." In order to do this, the selection of the feature functions $f_i(C, W)$ aims at capturing the relation between the class C and word sequence W. Typically, binary or weighted n-gram features, with $n = 1, 2, 3$, to capture the likelihood of the n-grams, are generated to express the user intent for the semantic class C [8]. Once the features are extracted from the text, the task becomes a text classification problem. Traditional text categorization techniques devise learning methods to maximize the probability of C_r, given the text W_r; i.e., the class-posterior probability $P(C_r|W_r)$.

3 Related work

Early work on spoken utterance classification has been done mostly for call routing or intent determination system, such as the AT&T How May I Help You? (HMIHY) system [9], relying on salience phrases, or the Lucent Bell Labs vector space model [10]. With advances in machine learning, especially in discriminative classification techniques, in the last decade, researchers have been able to apply off-the-shelf classification algorithms. Typically word n-grams are used as features after preprocessing with generic entities, such as dates, locations, or phone numbers. Because of the very large dimensions of the input space, large margin classifiers such as SVMs [4] or Boosting [3] were found to be very good candidates.

Deep learning methods have first been used for semantic utterance classification by Sarikaya et al. [6]. They have experimented with Deep Belief Networks (DBNs), which are stacks of Restricted Boltzmann Machines (RBMs) followed by fine tuning. RBM is a two-layer network, which can be trained reasonably efficiently in an unsupervised fashion.

In our earlier work, we have investigated the use of deep learning methods, namely Deep Convex Networks (DCNs) [1] and Kernel DCNs (K-DCNs) [7], for semantic utterance classification using lexical, named entity, and query click features. DCN is shown to be superior to DBN, not only in terms of accuracy, but also in training scalability and efficiency [11, 12]. K-DCNs allow the use of kernel functions during training, combining the power of kernel based methods and deep learning. While both approaches resulted in performances better than a Boosting-based baseline, K-DCNs have shown significantly bigger performance gains due to the use of query click features. Similar pattern has been observed comparing Boosting vs. SVM classification for the same task.

4 Query Click Logs

Query Click Logs (QCL) are logs of unstructured text including both the users queries sent to a search engine and the links that the users clicked on from the list of sites returned by that search engine. A common representation of such data is a bi-partite query-click graph as shown in 2, where one set of nodes represents queries, and the other set of nodes represents URLs, and an edge is placed between two nodes representing a query q and a URL u, if at least one user who typed q clicked on u.

Traditionally, the edge of the click graph is weighted based on the raw click frequency (number of clicks) from a query to a URL. Some of the challenges in extracting useful information from QCL is that the feature space is very high dimensional (there are thousands of url clicks linked to many queries), and there are millions of queries logged daily.

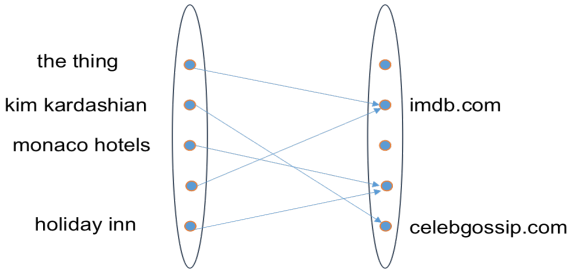

Figure 1: Depiction of Query Click Logs represented as a bi-partite graph of queries and URLs. Edges typically include frequencies from query to URL.

5 Zero-Shot Semantic Learning

Traditional SUC systems rely on a large set of labelled examples (X_r, C_r) to learn a good classifier f. Thus it suffers from bootstrapping issues and it makes scaling to a large number of classes costly. In this section we are interested in the problem of learning f with only unlabelled examples X_r. This is a form of zero-shot learning [13, 14]. We will present such a method for semantic utterance classification.

Semantic utterance classification relies on the inductive principle that there is an underlying semantic connection between utterances and classes. All the utterances belonging to a class share some form of similarity to each other. Traditionally, SUC systems are trained with labelled data to learn this relation. However, this overlooks the fact that a lot of the semantics of language can be discovered without labelled data. What's more, the name of semantic classes are not chosen randomly. They are often chosen because they describe the essence of the class. These two facts can be used easily by humans to classify without task-specific labels. For instance, it is easy to see that the utterance *the particle has exploded* belongs more to the class *physics* than *outdoors*. This is the very human ability that we wish to replicate here.

We propose a framework called zero-shot semantic learning (ZSL). It learns to perform SUC with only a set of unlabelled examples $\mathcal{X} = \{X_1, \ldots, X_N\}$ and the set of class names $\mathcal{C} = \{C_1, \ldots, C_M\}$. Furthermore, the names of the classes must belong to the same language as the inputs $\mathcal{X}$. This framework has the form

$$P(C_r|X_r) = \frac{1}{Z} e^{-\|P(H|X_r) - P(H|C_r)\|^2} \tag{2}$$

where $Z = \sum_C e^{-\|P(H|X_r) - P(H|C)\|^2}$. $P(H|X)$ is a probability distribution over different meanings of the input X. It is used to recover the meaning of the utterance X_r. The distribution of meanings according to a class $P(H|C_r)$ is given by the distribution of meanings of the class name. For example, given a class C_r with the name *physics* the distribution is found by using the class name as an utterance $P(H|C_r) = p(H|X = \{\text{physics}\})$. Intuitively, Equation 2 finds the class name which has the closest semantic meaning to the utterance. This framework will classify properly if

- The semantics of the language are properly captured by $P(H|X)$. Meaning that utterances are clustered according to their meaning.
- The class name C_r describes the semantic core of the class well. The best class name would have a meaning $P(H|C_r)$ that is the mean of the meaning of all its utterances $E_{X_r|C_r}[P(H|X_r)]$.

Most of the heavy lifting in this framework is performed by $P(H|X)$ whose job is to put related utterances close in the latent space. There are a wide array of models that can provide us with $p(H|X)$. This includes LSA, PCA, and other well known unsupervised learning algorithms. In this paper we will focus on using deep learning to obtain our latent meaning representation. In this context, we will be learning an embedding which is able to disentangle factors of variations in the meaning of a document.

We obtain embeddings by training deep neural networks using the Query Click Logs. The Query Click Logs associate unstructured query texts with a website. The associated website was the one

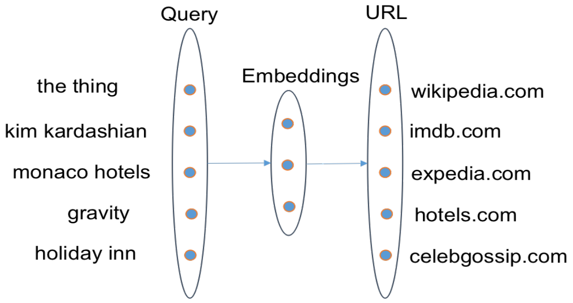

Figure 2: Depiction of the deep network from queries to URLs using the query click logs bi-partite graph.

Method	Restaurant	Hotel	Flight	Events	Transportation
ZSL with Bag-of-words	0.616	0.641	0.683	0.559	0.5
ZSL with $p(url\|query)$ (LR)	0.779	0.821	0.457	0.677	0.472
ZSL with $p(url\|query)$ (DNN)	0.838	0.862	0.46	0.631	0.503
ZSL with DNN Embedding	**0.858**	**0.935**	**0.870**	**0.727**	**0.667**
Representative URL heuristic (DNN)	0.798	0.892	0.769	0.707	0.577

Table 1: *Comparison of several zero-shot semantic learning methods for 5 semantic classes. Our proposed zero-shot learning system with DNN embeddings outperforms other approaches.*

that was selected by a user following the query. We make the mild hypothesis that the website clicked following a query reveals the meaning or intent behind a query. The queries which have similar meaning or intent will map to the same website. For example, it is easy to see that queries associated with the website *imdb.com* share a semantic connection to movies. We train the network with the query as input and the website as the output (see Figure 2). This learning scheme is inspired by the neural language models [15] who learn word embeddings by learning to predict the next word in a sentence. The idea is that the last hidden layer of the network has to learn an embedding space which is helpful to classification. And in order to do this, it will map similar inputs in terms of the classification task close in the embedding space.

We train deep neural networks with softmax output units and rectified linear hidden units. The inputs X_r are queries represented in bag-of-words format. The labels Y_r are the index of the website that was clicked. We train the network to minimize the negative log-likelihood of the data

$$\mathcal{L}(X, Y) = -\log P(Y_r|X_r)$$

The network has the form

$$P(Y = i|X_r) = \frac{e^{W_i^{n+1} H^n(X_r) + b_i^{n+1}}}{\sum_j e^{W_j^{n+1} H^n(X_r) + b_j^{n+1}}}$$

The latent representation function H^n is composed on n hidden layers

$$\begin{aligned} H^n(X_r) &= \max(0, W^n H^{n-1}(X_r) + b^n) \\ H^1(X_r) &= \max(0, W^1 X_r + b^1) \end{aligned}$$

We have a set of weight matrices W and biases b for each layer giving us the parameters $\theta = \{W^1, b^1, \ldots, W^{n+1}, b^{n+1}\}$ for the full network. Though rectified linear units are not smooth, research [16, 17] has shown that they can greatly improve the speed of learning of the network. We train the network using stochastic gradient descent with minibatches.

The meaning representation $P(H|X)$ is found at the last embedding layer $H^n(X_r)$. The optimal number of layers to use is not known in advance and is found through cross-validation.

6 Zero-Shot Semantic Clustering

In the previous section, we have discussed a novel way to use unlabelled examples to perform zero-shot SUC. However, it is not clear how to use the wealth of unlabelled data in the case where labelled examples are available. The embeddings described in Section 5 could be useful and it has been shown [18, 19] that using unsupervised learning algorithms like the restricted Boltzmann machine [20] can help leverage this additional data. These unsupervised algorithms can be used to initialize the parameters of a DNN or to extract features/embeddings. Effectively, these methods replace the task of learning $P(C|X)$ to learning a density model of the data $P(X)$. The hypothesis is that $P(C|X)$ shares structure with $P(X)$. Thus the features learned from $P(X)$ will be useful to model $P(C|X)$. In other words, we assume that learning features from $P(X)$ will be a good proxy to learn features for $P(Y|X)$. However, it is not clear how of a proxy that is for a given task. In this section, we propose a more reasoned proxy task to learn features for semantic utterance classification. It can be thought of as a zero-shot clustering method.

We consider that the quality of a proxy $\hat{f}$ for a function f is measured by the error $E_X[\|f(X) - \hat{f}(X)\|^2]$. A good proxy should have a small error. It is easy to see gradient-based learning with $\hat{f}$ approximates learning with f. This explains why bootstrapping a classifier with the objective $\hat{f}$ may be useful. This framework imposes several restrictions over the function $\hat{f}$. If $f : X \to Y$ then we must have $\hat{f} : X \to Y$. The proxy should be defined over the same input and output space. The restriction over the input space is easy to satistify. It is satisfied by the various pretraining methods like restricted Boltzmann machines [20] and regularized auto-encoders [21, 22]. The restriction over the output is not satisfied by these methods and thus they cannot be measured as proxies under this definition. In general finding a function satisfying these restrictions is hard, but we have already introduced the building blocks for such a function in the context of semantic utterance classification.

Zero-shot semantic learning can be used to define a good proxy task. As we will show in Section 7 the classification results with ZSL are good and thus $E_X[\|f(X) - \hat{f}(X)\|^2]$ is comparatively small. ZSL relies on learning embeddings on the Query Click Logs that cluster together utterances that have the same meaning. These embeddings do not have any pressure to cluster according to the SUC classes. We would like these embeddings to cluster not only according to meaning, but also to cluster according to the final SUC classes. In order to do this we can use ZSL as a proxy to quantify the quality of a clustering over classes. One possibility is to maximize the likelihood $P(C_r|X_r)$ of ZSL directly, but this requires labelled data. Instead we define this quality measure as the entropy over the predicted semantic classes

$$\begin{aligned} H(P(C_r|X_r)) &= E[I(P(C_r|X_r))] \\ &= E[-\sum_i P(C_r = i|X_r)\log P(C_r = i|X_r)]. \end{aligned} \tag{3}$$

The entropy tell us the uncertainty we have over the class. The more certain we are of the class, the better the clustering given by the embedding $P(H|X)$. The better the proxy function $\hat{f}$ the better this measure ($\|H(f(X)) - H(\hat{f}(X))\|^2 \leq K\|f(X) - \hat{f}(X)\|^2$ by Lipschitz continuity). Another key property is that this measure marginalizes over possible classes and so does not require labelled data.

Zero-shot semantic clustering (ZSC) leverages this measure to learn an embedding that clusters according to the semantic classes *without any labelled data*. It relies on jointly learning an embedding space by predicting the clicks and optimizing the clustering measure given by Equation 3. To our knowledge, ZSC is the first direct zero-shot clustering method. The objective has the form

$$\mathcal{L}(X, Y) = -\log P(Y|X) + \lambda H(P(C|X)). \tag{4}$$

The variable X is the input, Y is the website that was clicked, C is a semantic class. The functions $\log P(Y|X)$ and $\log P(C|X)$ are predicted by a deep neural network as described in the previous section. Both functions use the same embedding provided by the last hidden layer of the network. The term $H(P(C|X))$ can be thought of as a regularization that encourages the embedding to cluster according to the classes. It is a force in the embedding space that makes the examples congregate around the position of class names in the embedding space. The hyper-parameter λ controls the strength of that force in the overall objective. We find this value by cross-validation.

7 Experiments

In this section, we evaluate the proposed zero-shot semantic learning and clustering methods proposed in the previous sections.

Restaurant	Hotel	Flight	Events	Transportation
steakhouse	suites	airline	festivals	distributing
diner	hyatt	airfaire	upcoming	dfw
seafood	resorts	plane	fireworks	petroleum
tavern	ramada	baggage	happening	hospitality

Table 2: *Nearest neighbours in the embedding space. Each column displays the 5 nearest neighbours of the word at the top. We can see that the embedding captures the semantics of the words.*

We have gathered a month of query click log data from a search engine for learning the embeddings. We restricted the websites to the the 1000 most popular websites in this log. The words in the bag-of-words vocabulary are the 9521 found in the supervised SUC task we will use. All queries containing only unknown words were filtered out. We found that using a list of stop-words improved the results. After these restrictions, the dataset comprises 620,474 different queries.

We evaluate the performance of the methods for SUC on the dataset gathered by [1]. It was compiled from utterances by users of a spoken dialog system. There are 16,000 training utterances, 2000 utterances for validation and 2000 utterances for testing. Each utterance is labelled with one of 25 domains.

The hyper-parameters of the models are tuned on the validation set. The learning rate parameter of gradient descent is found by grid search with $\{0.1, 0.01, 0.001\}$. The number of layers is between 1 and 3. The number of hidden units is kept constant through layers and is found by sampling a random number from 300 to 800 units. We found that it was helpful to regularize the networks using dropout [23]. We sample the dropout rate randomly between 0% dropout and 20%. The λ of the zero-shot clustering method is found through grid-search with $\{0.1, 0.01, 0.001\}$. The models are trained on a cluster of computers with double quad-core Intel(R) Xeon(R) CPUs with 2.33GHz and 8Gb of RAM. Training either the ZSL or ZSC method on the QCL data requires 4 hours of computation time.

We can evaluate qualitatively the performance of the embedding method described in Section 5 for zero-shot semantic learning by looking at the nearest neighbours of words in the embedding space. Table 2 shows the nearest neighbours of specific words in the embedding space learned by a network with 2 layers and 727 hidden units. The neighbours of the selected words all share the semantic domain of the word. This validates the hypothesis that there is a semantic link between the query texts and the websites.

We evaluate the zero-shot semantic learning system by measuring its effectiveness for classification. Our results are given in Table 1. The performance is measured using the AUC (Area under the curve of the precision-recall curve). We compare against various means of obtaining the meaning representation $P(H|X)$. We compare with using the bag-of-words representation (denoted *ZSL with Bag-of-words*), the distribution of websites predicted for a query by a logistic regression (*ZSL with* $p(url|query)$ *(LR)*) and a deep neural network (*ZSL with* $p(url|query)$ *(DNN)*). We also compare with a sensible heuristic method denoted *Representative URL heuristic*. We associate each domain with a website (i.e. *flights* with *expedia.com*). The probability of belonging to a semantic class is given by the probability that the utterance is associated with the website for that semantic class. Table 1 shows that the proposed method of zero-shot semantic learning with DNN embeddings performs best on each considered class. These results provide experimental evidence for the validity of the hypothesis behind ZSL.

We also compare the zero-shot learning system with a supervised SUC system. We compare ZSL with a linear SVM. The task is identify utterances of the *restaurant* semantic class. Figure 3 shows the performance of the linear SVM as the number of training examples increases. The performance of ZSL is shown as a straight line. Predictably, the SVM which is trained with labels achieves better results. However, we observe that the ZSL system does get within 90% of the performance of the SVM.

We evaluate the zero-shot semantic method as a feature extraction method for a supervised model. In our experiments, we have used the embeddings as additional features for a linear SVM. We compare with the state-of-the-art method and other approaches in Table 3. The state-of-the-art method is the Kernel DCN on QCL features with 5.94% test error. However, we train using the more scalable

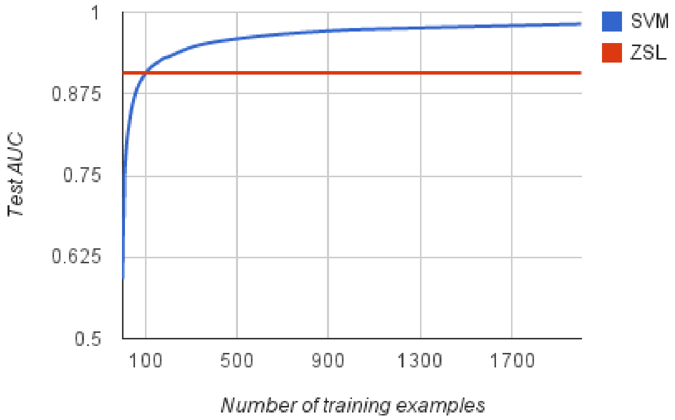

Figure 3: Comparison between the proposed method and an SVM trained with increasing amount of examples. ZSL achieves with 90% of the performance of the SVM.

Features	Kernel DCN	SVM
Raw	9.52%	10.09%
QCL features [24]	**5.94%**	6.36%
DNN urls		6.88%
DNN embeddings		6.2%
ZSC embeddings		**5.73%**

Table 3: *Classification results with various additional features. The ZSC embeddings produce the best results for the linear SVM.*

linear SVM which leads to 6.36% with the same input features. Using the embeddings learned from the QCL data as described in Section 5 yields 6.2% errors. Comparatively, using ZSC embeddings reduces the error to 5.73%. The ZSC are the best features for the linear SVM.

8 Conclusion

We have introduced two novels methods of zero-shot learning for SUC. Zero-shot semantic learning allows classification without labels with applications to SUC problems with large number of classes. Zero-shot semantic clustering is a method for feature extraction for traditional SUC systems. Both approaches exploit unlabelled data with deep learning and our experiments have shown the effectiveness of both methods.

References

[1] G. Tur, L. Deng, D. Hakkani-Tür, and X. He, "Towards deeper understanding deep convex networks for semantic utterance classification," in *In Proceedings of the ICASSP*, Kyoto, Japan, March 2012.

[2] G. Tur and R. De Mori, Eds., *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, John Wiley and Sons, New York, NY, 2011.

[3] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2/3, pp. 135–168, 2000.

[4] P. Haffner, G. Tur, and J. Wright, "Optimizing SVMs for complex call classification," in *Proceedings of the ICASSP*, Hong Kong, April 2003.

[5] S. Yaman, L. Deng, D. Yu, Y.-Y. Wang, and A. Acero, "An integrative and discriminative technique for spoken utterance classification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 6, pp. 1207–1214, 2008.

[6] R. Sarikaya, G. E. Hinton, and B. Ramabhadran, "Deep belief nets for natural language call-routing," in *Proceedings of the ICASSP*, Prague, Czech Republic, 2011.

[7] L. Deng, G. Tur, X. He, and D. Hakkani-Tür, "Use of kernel deep convex networks and end-to-end learning for spoken language understanding," in *In Proooceedings of the IEEE SLT Workshop*, Miami, FL, December 2012.

[8] G. Tur and L. Deng, *Intent Determination and Spoken Utterance Classification, Chpater 3 in Book: Spoken Language Understanding*, John Wiley and Sons, New York, NY, 2011.

[9] A. L. Gorin, G. Riccardi, and J. H. Wright, "How May I Help You?," *Speech Communication*, vol. 23, pp. 113–127, 1997.

[10] J. Chu-Carroll and B. Carpenter, "Vector-based natural language call routing," *Computational Linguistics*, vol. 25, no. 3, pp. 361–388, 1999.

[11] L. Deng and D. Yu, "Deep convex nets: A scalable architecture for speech pattern classification," in *Proceedings of the Interspeech*, Florence, Italy, 2011.

[12] L. Deng, D. Yu, and J. Platt, "Scalable stacking and learning for building deep architectures," in *Proc. ICASSP*, Kyoto, Japan, 2012.

[13] Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio, "Zero-data learning of new tasks," in *AAAI Conference on Artificial Intelligence*, 2008.

[14] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell, "Zero-shot learning with semantic output codes," in *Advances in neural information processing systems*, 2009, pp. 1410–1418.

[15] Yoshua Bengio, "Neural net language models," *Scholarpedia*, vol. 3, no. 1, pp. 3881, 2008.

[16] Xavier Glorot, Antoine Bordes, and Yoshua Bengio, "Deep sparse rectifier neural networks," in *JMLR W&CP: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, Apr. 2011.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25 (NIPS'2012)*. 2012.

[18] Xavier Glorot, Antoine Bordes, and Yoshua Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," In *Proceedings of theTwenty-eight International Conference on Machine Learning (ICML'11)* [25], pp. 97–110.

[19] Hugo Larochelle and Yoshua Bengio, "Classification using discriminative restricted Boltzmann machines," in *Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML'08)*, William W. Cohen, Andrew McCallum, and Sam T. Roweis, Eds. 2008, pp. 536–543, ACM.

[20] Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[21] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, Dec. 2010.

[22] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," In *Proceedings of theTwenty-eight International Conference on Machine Learning (ICML'11)* [25].

[23] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," Tech. Rep., arXiv:1207.0580, 2012.

[24] D. Hakkani-Tür, L. Heck, and G. Tur, "Exploiting query click logs for utterance domain detection in spoken language understanding," in *Proceedings of the ICASSP*, Prague, Czech Republic, 2011.

[25] "Proceedings of the twenty-eight international conference on machine learning (icml'11)," in *Proceedings of theTwenty-eight International Conference on Machine Learning (ICML'11)*, -1.