Sparse Single-Hidden Layer Feedforward Network for Mapping Natural Language Questions to SQL Queries

Issam Hadj Laradji, Lahouari Ghouti, Faisal Saleh, and Musab A. AlTurki

Department of Information and Computer Science,
King Fahd University of Petroleum and Minerals, Dhahran 3126, Saudi Arabia
issam.laradji@gmail.com, {lahouari,musab}@kfupm.edu.sa,
faisal86@icloud.com

Abstract. Mapping natural language (NL) statements into SQL queries allows users to interact with systems through everyday language. Semantic parsing has seen a growing interest over the past decades. In this paper, we extend single hidden layer feedforward network (SLFN) by adding the Kullback-Liebler (KL) divergence parameter to its objective function. We refer to this algorithm as Sparse SLFN (S-SLFN) which can learn whether an SQL query answers a particular NL question. With Bag of Words (BoW) representing the questions and the queries, the algorithm, by enforcing sparsity, is meant to retain robust features representing informative relationships and structure of the data. Experimental results show that S-SLFN outperforms SLFN and other algorithms for the GeoQueries dataset by a respectable margin.

Keywords: Single-hidden Layer Feedforward Network (SLFN), Sparsity, Semantic Parsing.

1 Introduction

Powerful consumer handheld devices became increasingly dominant over the past years, underscoring the need to simplify complex tasks for users not well-acquainted with technology. One such task is to retrieve data records corresponding to the user's query. To simplify the task is to allow users to ask for information in everyday language. Therefore, a system should be able to map the received natural language (NL) statement into SQL queries to fetch the right records.

Some early methods for semantic parsing adopted formal rules for mapping NL statements to machine instructions. Jones et al. [1] developed tree transducers for the mapping, with a variational Bayesian inference algorithm providing elegant solutions to the problem. Further, Jones et al. [2] presented an approach that makes use of synchronous context free graph grammars. It constructs an intermediate graph-structured meaning representation, which, with the application of synchronous hyperedge replacement grammars, can be translated into either its respective machine instruction or natural language statement.

S. Wermter et al. (Eds.): ICANN 2014, LNCS 8681, pp. 241–248, 2014.

Other learning algorithms employing probability functions have emerged as well. Poon and Domingos [3] developed a deep network whose inputs represent the dependency trees of given sentences, and whose hidden features represent clusters of meaning expressions, realizing a novel unsupervised approach to semantic parsing. Involving Support Vector Machines, Giordani and Moschitti [4] provided an interesting perspective to semantic parsing by constructing custom kernels and applying them to datasets containing NL statements with their matching SQL statements.

The work in this paper is inspired by the unsupervised feature extraction algorithm, Sparse Auto-encoders (SAE) [5]. From image pixels, SAE would extract new robust features representing interesting structural information of the pixels, meant to hold essential information of the image. However, when facing datasets containing only few question-SQL matching pairs, the features extracted from learning to reconstruct them would not likely be robust.

Classifiers, on the other hand, can leverage samples of non-matching question-query pair representing the combined features of questions and their non-matching queries, to efficiently construct the decision function. Furthermore, what advantages SAE has in extracting features can be added to single-hidden layer feedforward networks (SLFN). Therefore, we developed a sparse SLFN containing the Kullback-Liebler (KL) divergence parameter in its objective function. By injecting sparsity, the hidden layer would learn robust hidden features representing superior structural information that would otherwise foster correct mapping of natural language (NL) statements to their respective SQL queries.

While the first step of the approach involves Bag of Words (BoW) extraction of terms and their occurrences as features, KL divergence ensures that the relationships between these features developed in the hidden layer are informative by discarding those that do not contribute much in building the decision function.

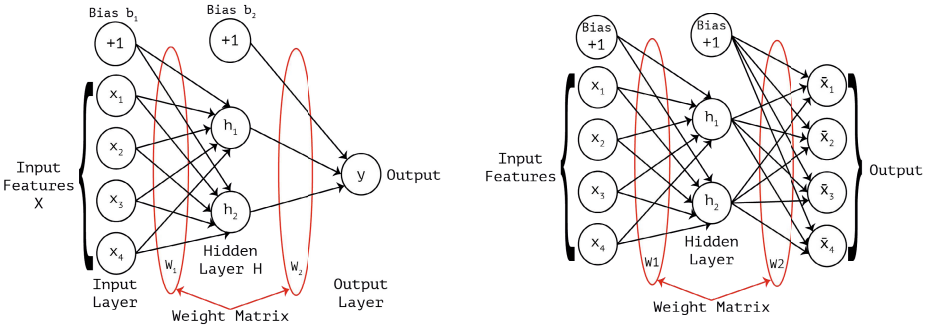
Results on the GeoQueries showed that S-SLFN outperformed single hidden layer feedforward network [6], Logistic Regression [7], and Support Vector Machines [8], by at least 2% AUC. The process involves the classifier training on the training dataset - containing the correct and incorrect pairs of NL statement and SQL query - and then predicting the SQL queries closest in answering the NL statements in the testing set.

The remainder of the paper is organized as follows. Section 2 provides technical background; section 3 presents the proposed approach; section 4 explains the experimentation results and analysis; section 5 concludes the paper.

2 Technical Background

2.1 Single-Hidden Layer Feedforward Neural Network

Assume a single hidden layer feedforward network SLFN with L hidden neurons and n training samples. Consider a matrix $X \in R^n \times R^m$ defining the input vectors as $\{x_i | x_i \in R^m\}$for $i = 0, 1, ..., n$ where m is the number of input features representing a vector x_i, the bias vectors $b_1 \in R^L$ and $b_2 \in R$, and the target vector $Y \in R^n$ defined as y_i for $i = 1, 2, ..., n$ where y_i is the respective output of



(a) Single-layer Feed-Forward Network.

(b) Sparse auto-encoder.

Fig. 1. Neural Networks

x_i. Let us also consider the matrices $W_1 \in R^m \times R^L$, and $W_2 \in R^L$ representing the outgoing weights of the input layer and the hidden layer, respectively. Then, the output of SLFN is,

$$f(x) = W_2\, g(X \cdot W_1 + b_1) + b_2 \tag{1}$$

where $g(x) : R \to R$ is the activation function (e.g. sigmoid and hyperbolic tanh).

The objective function set as cross-entropy is defined as follows,

$$J(W, b; x, y) = -f(x) \ln\ y - (1 - f(x)) \ln(1 - y) \tag{2}$$

Taking the gradient of eq. (2) with respect to the parameters would allow updating the parameters as follows,

$$\begin{aligned} W_i &:= W_i - \alpha \left[\frac{1}{n} \Delta W_i \right] \\ b_i &:= b_i - \alpha \left[\frac{1}{n} \Delta b_i \right] \end{aligned} \tag{3}$$

where α is the learning rate, ΔW_i is the weight change in terms of the objective function derivative with respect to weight i, and Δb_i is the bias change in terms of the objective function derivative with respect to the bias unit i. An SLFN network is shown in Fig. 1 (a).

2.2 Sparse Auto-encoders

Illustrated in Fig. 1 (b), Sparse Auto-encoders aim to extract a robust representation - retained in the hidden layer - of the data by learning to reconstruct the input features. In addition to the objective function defined for SLFN, Sparse

Auto-encoders include a penalty term called the Kullback-Liebler (KL) divergence,

$$KL(p||\hat{p}) = p\ log\frac{p}{\hat{p}_j} + (1-p)\ log\frac{1-p}{1-\hat{p}_j} \tag{4}$$

where the value ρ is manually set to control the values of $\hat{\rho}$ - the average activation values of a hidden node over the data samples - by adding a penalty cost whenever $\hat{\rho}$ differs from ρ. This controls the degree of sparsity of the features retained in the hidden layer.

3 Proposed Approach

This section provides details of the proposed approach, whose scheme is illustrated in Fig. 2.

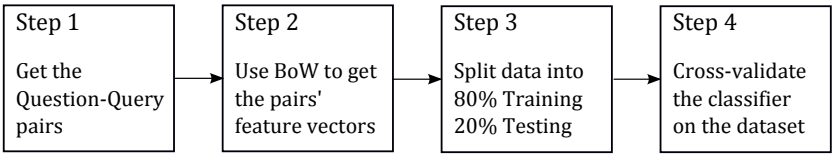


Fig. 2. Flow diagram of proposed procedure

3.1 Feature Representation

Given a set of samples $(x_i, y_i), i = 1, 2, ..., N$ where $x \in \mathbb{R}^{(n+m)}$ and $y_i = \{0, 1\}$, n and m are the number of features representing question q_i and SQL query s_i, respectively. For q_i and s_i that match, $y_i = 1$; otherwise, $y_i = 0$. We say q_i matches s_i when s_i returns results relevant to q_i. It is worth noting that for datasets containing no non-matching question-query pairs (negative sample), by the Cartesian product, we create negative samples by concatenating the feature vector of each question with each wrong query.

These features are extracted using Bag of Words (BoW) [9], where for each unique term, except for the stop words - like 'a', and 'the', the number of occurrences are counted and given in the feature vector.

To illustrate this, given feature vectors representing q_i and s_i, as in Fig. 3. We join the feature vectors together to form the question-query pair qs_i.

The benefit of BoW lies in their ability to compactly, and effectively summarize the features defining the statements.

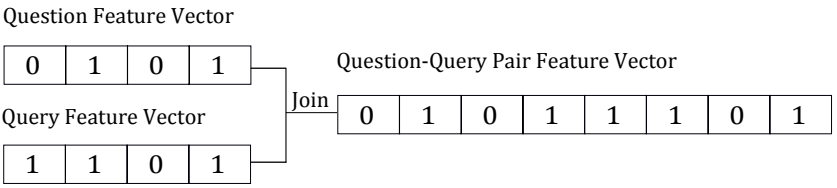


Fig. 3. Joining the feature vectors to form the question-query pair

3.2 Sparse Single-Layer Feed-Forward Network

We developed a Sparse Single-layer Feed-Forward Network (S-SLFN) that extends SLFN by adding the sparsity term - Kullback-Liebler (KL) divergence - to the objective function given in eq. (2). The term works by discouraging redundant and uninformative hidden activations. In other words, on training, the algorithm discards hidden activations whose penalty cost, determined by KL, supersedes their contributions to constructing the decision boundary. To put this under mathematical formulation, the modified objective function is the combination of $J(W, b; x, y)$ in eq. (2) and $KL(p||\hat{p})$ in eq. (4), which is,

$$J_{Sparse}(W, b) = J(W, b; x, y) + KL(p||\hat{p}) \tag{5}$$

The value for ρ is set arbitrarily. ρ penalizes the objective function when the average hidden activation values of a hidden node over the data samples - given as $\hat{\rho}$ - is different from ρ. The Kullback-Liebler (KL) divergence measures the distance between the two distributions, ρ and $\hat{\rho}$. It is asymmetric in the sense that a $\hat{\rho}$ larger than ρ is penalized with higher cost than if it were smaller, even when the difference is equal. This provides a favorable outcome, as having lower penalty for smaller $\hat{\rho}$ would serve the main objective of the algorithm: to extract a sparse set of hidden features.

Since the meaning of a phrase is not necessarily reflected by the meaning of the individual constituent words, the sparsity term would provide more information about a statement than word occurrences. It allows for the extraction of robust features describing the essence of a statement while eliminating possibly noisy information from redundant words.

4 Experimentation

4.1 Experimental Setup

We ran the experiments in a machine with 3.6 GHz quad-core CPU and 32 GB RAM operating a 64-bit Windows 7. For a fair, reliable assessment, we split the data into 80% training set and 20% testing set in a stratified manner. As such, both sets have the same ratio of positive samples to negative samples. We repeated the cross-validation five times and took the average of their scores, based on the metric Area Under the Receiver Operating Characteristic (ROC) curve (AUC). The reason for such metric is that, the datasets are imbalanced, as they contain negative samples (non-matching question-SQL query pairs) that highly outnumber matching pairs. After all, AUC is a popular metric for imbalanced datasets [10].

For the benchmark, we evaluated the following algorithms with the described settings (unless specified otherwise),

1. Logistic Regression with stochastic gradient descent and iterations till convergence.

2. Three different Support Vector Machines (SVM): SVM with linear kernel, SVM with polynomial kernel of degree 3, and SVM with Radial Basis Function (RBF) kernel.
3. Single-hidden layer feedforward neural network (SLFN) with 25 hidden neurons, stochastic gradient descent and iterations till convergence
4. Sparse (SLFN) with 25 hidden neurons, stochastic gradient descent, iterations till convergence, and ρ set to 0.12 for GeoQueries, and 0.1 for RestQueries.

While the choice of ρ is subjective, we found that these values has lead to better results than otherwise.

For each question-query pair from the testing set, the classifier outputs the probability that the pair matches. The AUC metric then evaluates the performance of the computed probabilities.

4.2 Experimental Design

We evaluated the learning algorithms on two datasets: GeoQueries[1] and RestQueries[1]. Table 1 reports the statistics of the two datasets including the number of samples, and the number of extracted BoW features. It is noteworthy to mention that, while the original datasets contained only positive samples (matching pairs), by the Cartesian product explained in section 3.1, we introduced a large number of negative samples to help classifiers develop robust decision functions.

Table 1. Generated dataset statistics

Dataset	**Questions**	**Queries**	**Positive samples**	**Negative samples**	**BoW features**
GeoQueries	149	80	164	11756	89
RestQueries	126	77	852	8850	21

Below we show an example of a matching question-query pair from the GeoQueries dataset.

- **Question:** what is the capital of the state with the largest population?
- **Query:** select distinct state.capital from state where state.population=(select max(state.population) from state)
- **Extracted BoW terms:** what, capital, state, largest, population, select, distinct, capital, max

The feature vector representing the question-query pair above will contain '1's in the indices corresponding to the extracted BoW terms and '0' s for the rest of the BoW terms. The construction of this feature vector is explained in section 3.1.

[1] Available at: `http://disi.unitn.it/~agiordani/corpora.htm`

4.3 Results and Analysis

We trained the algorithms on the GeoQuery dataset to evaluate the hypothesis that S-SLFN achieves the better performance. Table 2 shows that S-SLFN has indeed topped the benchmark with a solid improvement of 2% over the next best achieving algorithm - SVM with RBF kernel.

This suggests that the Kullback-Liebler (KL) divergence term retained better features of the training samples, whereas SLFN and SVM retained weaker features that are possibly redundant and uninformative.

Finally, testing the algorithms on the RestQuery dataset have shown another favorable achievement of S-SLFN. As illustrated in Table 2, while S-SLFN did not see a vast improvement over SLFN, it still maintained its first position as the best performing classifier.

Why S-SLFN did not improve much over SLFN can be attributed to the fact that only few BoW features are extracted from RestQueries dataset (Table 1). Because of the limited number of hidden features that can be extracted from the dataset, S-SLFN and SLFN would more or less retain similar features.

Table 2. Comparison between algorithms using the AUC performance metric

Algorithm	**GeoQueries**	**RestQueries**
Logistic Regression (LR)	0.84 ± 0.019	0.80 ± 0.015
SVM with Linear Kernel	0.71 ± 0.035	0.47 ± 0.076
SVM with polynomial kernel	0.80 ± 0.026	0.54 ± 0.032
SVM with RBF kernel	0.91 ± 0.021	0.49 ± 0.086
Single-layer Feed-Forward Network (SLFN)	0.89 ± 0.038	0.82 ± 0.010
Sparse SLFN (S-SLFN)	**0.93** ± 0.020	**0.83** ± 0.009

For RestQueries, each SVM achieved low, yet highly unstable AUC results - as illustrated by the high standard deviation given in Table 2, unlike Logistic Regression, SLFN and S-SLFN. This suggests that SVM is not efficient for RestQueries' type of data, as SVM labeled almost all samples as non-matching pairs, favoring the majority class (negative samples) over the minority (positive samples).

5 Conclusion

We developed a sparse single-hidden feedforward network, a supervised learning algorithm for semantic parsing, specifically for mapping Natural language questions to formal SQL queries. The Kullback-Liebler (KL) divergence parameter in the objective function allows for learning robust features for the hidden layer.

Experimental results have justified the efficacy of S-SLFN over the standard SLFN when the number of BoW is large. For future work, it would be interesting to apply S-SLFN for other problems under semantic parsing, and for online (real-time) mapping of natural questions to SQL queries.

Acknowledgment. The authors would like to thank King Fahd University of Petroleum and Minerals (KFUPM) for supporting this work.

References

1. Jones, B.K., Johnson, M., Goldwater, S.: Semantic parsing with bayesian tree transducers. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers, vol. 1, pp. 488–496. Association for Computational Linguistics (2012)
2. Jones, B., Andreas, J., Bauer, D., Hermann, K.M., Knight, K.: Semantics-based machine translation with hyperedge replacement grammars. In: COLING, pp. 1359–1376 (2012)
3. Poon, H., Domingos, P.: Deep learning for semantic parsing
4. Giordani, A., Moschitti, A.: Semantic mapping between natural language questions and sql queries via syntactic pairing. In: Horacek, H., Métais, E., Muñoz, R., Wolska, M. (eds.) NLDB 2009. LNCS, vol. 5723, pp. 207–221. Springer, Heidelberg (2010)
5. Ng, A.: Sparse autoencoder
6. Sanger, T.D.: Optimal unsupervised learning in a single-layer linear feedforward neural network. Neural Networks 2(6), 459–473 (1989)
7. Hosmer Jr., D.W., Lemeshow, S., Sturdivant, R.X.: Applied logistic regression. Wiley. com (2013)
8. Hearst, M.A., Dumais, S., Osman, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intelligent Systems and their Applications 13(4), 18–28 (1998)
9. Wallach, H.M.: Topic modeling: beyond bag-of-words. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 977–984. ACM (2006)
10. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering 21(9), 1263–1284 (2009)