A Model of Zero-Shot Learning of Spoken Language Understanding

Majid Yazdani
Computer Science Department
University of Geneva
majid.yazdani@unige.ch

James Henderson
Xerox Research Center Europe
james.henderson@xrce.xerox.com

Abstract

When building spoken dialogue systems for a new domain, a major bottleneck is developing a spoken language understanding (SLU) module that handles the new domain's terminology and semantic concepts. We propose a statistical SLU model that generalises to both previously unseen input words and previously unseen output classes by leveraging unlabelled data. After mapping the utterance into a vector space, the model exploits the structure of the output labels by mapping each label to a hyperplane that separates utterances with and without that label. Both these mappings are initialised with unsupervised word embeddings, so they can be computed even for words or concepts which were not in the SLU training data.

1 Introduction

Spoken Language Understanding (SLU) in dialogue systems is the task of taking the utterance output by a speech recognizer and assigning it a semantic label that represents the dialogue actions of that utterance accompanied with their associated attributes and values. For example, the utterance ”I would like Chinese food” is labelled with *inform(food=Chinese)*, in which *inform* is the dialogue action that provides the value of the attribute *food* that is *Chinese*.

Dialogue systems often use hand-crafted grammars for SLU, such as Phoenix (Ward, 1994), which are expensive to develop, and expensive to extend or adapt to new attributes and values. Statistical SLU models are usually trained on the data obtained from a specific domain and location, using a structured output classifier that can be discriminative (Pradhan et al., 2004; Kate and Mooney, 2006; Henderson et al., 2012) or generative (Schwartz et al., 1996; He and Young, 2005). Gathering and annotating SLU data is costly and time consuming and therefore SLU datasets are small compare to the number of possible labels.

Because training sets for a new domain are small, or non-existent, learning is often an instance of Zero-shot or One-shot learning problems (Palatucci et al., 2009; L. Fei-Fei; Fergus, 2006), in which zero or few examples of some output classes are available during the training. For example, in the restaurant reservation domain, not all possible combinations of foods and dialogue actions may be included in the training set. The general idea to solve this type of problems is to map the input and class labels to a semantic space of usually lower dimension in which similar classes are represented by closer points in the space (Palatucci et al., 2009; Weston et al., 2011; Weston et al., 2010). Usually unsupervised knowledge sources are used to form semantic codes of the labels that helps us to generalize to unseen labels.

On the other hand, there are also different ways to express the same meaning, and similarly, most of them can not be included in the training set. For instance, the system may have seen ”Please give me the telephone number” in training, but the user might ask ”Please give me the phone” at test time. This problem, feature sparsity, is a common issue in many NLP tasks. Decomposition of input feature parameters using vector-matrix multiplication (Bengio et al., 2003; Collobert et al., 2011; Collobert and Weston, 2008) has addressed this sparsity issue successfully in previous work. In this way, by sharing the word representations and composition matrices, we can overcome feature sparsity by producing similar representations for similar utterances.

In order to represent words and concepts we use word embeddings, which are a form of vector space model. Word embeddings have proven to be effective models of semantic representation

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 244–249,
Lisbon, Portugal, 17-21 September 2015. ©2015 Association for Computational Linguistics.

of words in various NLP tasks (Baroni et al., 2014; Yazdani and Popescu-Belis, 2013; Collobert et al., 2011; Collobert and Weston, 2008; Huang et al., 2012; Mikolov et al., 2013b). In addition to parameter sharing, these representations enable us to leverage large scale unlabelled data. Because word embeddings trained on unlabeled data reflect the similarity between words, they help the model generalize from the words in the original training corpora to the words in the new extended domain, and help generalize from small amounts of data in the extended domain.

The contribution of this paper is to build a representation learning classifier for the SLU task that can generalize to unseen words and labels. For every utterance we learn how to compose the word vectors to form the semantics of that utterance for this task of language understanding. Furthermore, we learn how to compose the semantics of each label from the semantics of the words used to name that label. This enables us to generalize to unseen labels.

In this work we use the word2vec software of Mikolov et al. (2013a)[1] to induce unsupervised word embeddings that are used to initialize word embedding parameters. For this, we use an English Wikipedia dump as our unlabelled training corpus, which is a diverse broad-coverage corpus. It has been shown (Baroni et al., 2014; Mikolov et al., 2013b) that these embeddings capture lexical similarities even when they are trained on a diverse corpus like Wikipedia. We test our models on a restaurant booking domain. We investigate domain adaptation by adding new attribute types (e.g. *goodformeal*) and new attribute values (e.g. *Hayes Valley* as a restaurant location). Our experiments indicate that our model has better performance compared to a hand-crafted system as well as a SVM baseline.

2 SLU Datasets

The dialogue utterances used to build the SLU dataset were collected during a trial of online dialogue policy adaptation for a restaurant reservation system based in San Francisco. The trial began with (*area*, *pricerange* and *food*), and adapted the Interaction Manager online to handle the additional attribute types *near*, *allowedforkids*, and *goodformeal* (Gašic et al., 2014). User utterances from these trials were transcribed and annotated with dialogue acts by an expert, and afterwards edited by another expert[2]. Each user utterance was annotated with a set of labels, where each label consists of an act type (e.g. inform, request), an attribute type (e.g. foodtype, pricerange), and an attribute value (e.g. Chinese, Cheap).

The dataset is separated into four subsets, SFCore, SF1Ext, SF2Ext and SF3Ext, each with an increasing set of attribute types, as specified in Table 1. This table also gives the total number of utterances in each data set. For our first experiment, we split each dataset into about 15% for the testing set and 85% for the training set. For our second experiment we use each extended subset for testing and its preceding subsets for training.

Ontology	Attribute types (# of values)	# of utterances
SFCore	*food*(59), *area*(155), *pricerange*(3)	1103
SF1Ext	SFCore + *near*(39)	1810
SF2Ext	SF1Ext + *allowedforkids*(2)	1571
SF3Ext	SF2Ext +*goodformeal*(4)	1518

Table 1: Domains for San Francisco (SF) restaurants expanding in complexity

3 A Dialogue Act Representation Learning Classifier

The SLU model is run on each hypothesis output by the ASR component, and tries to predict the correct set of dialogue act labels for each hypothesis. This problem is in general an instance of multi-label classification, because a single utterance can have multiple dialogue act labels. Also, these labels are structured, since each label consist of an act type, an attribute type, and an attribute value. Each label component also has canonical text associated with it, which is the text used to name the label component (e.g. “Chinese” as a value).

The number of possible dialogue acts grows rapidly as the domain is extended with new attribute types and values, making this task one of multi-label classification with a very large number of labels. One natural approach to this task is to train one binary classifier for each possible label, to decide whether or not to include it in the output. In our case, this requires training a large number of classifiers, and it is impossible to generalize to

[1] https://code.google.com/p/word2vec/

[2] This data is publically available from https://sites.google.com/site/parlanceprojectofficial/home/datarepository

dialogue acts that include attributes or values that were not in the training set since there won't be any parameter sharing among label classifiers.

In our alternative approach, we build the representation of the utterance and the representation of the label from their constituent words, then we check if these representations match or not. In the following we explain in details this representation learning model.

3.1 Utterance Representation Learning

In this section we explain how to build the utterance representation from its constituent words. In addition to words, we use bigrams, since they have been shown previously to be effective features for this task (Henderson et al., 2012). Following the success in transfer learning from parsing to understanding tasks (Henderson et al., 2013; Socher et al., 2013), we use dependency parse bigrams in our features as well. We learn to build a local representation at each word position in the utterance by using the word representation, adjacent word representations, and the head word representation. Let $\phi(w)$ be a d dimensional vector representing the word w, and $\phi(U_i)$ be a h dimensional vector which is the local representation at word position i. We compute the local representation as follows:

$$\phi(U_i) = \sigma(\phi(w_i)W_{word} + \phi(w_h)W_{parse_{R_k}} + \phi(w_j)W_{previous} + \phi(w_k)W_{next}) \quad (1)$$

in which w_h is the head word with the dependency relation R_k to w_i, and w_j and w_k are the previous and next words. W_{word} is a $d \times h$ matrix that transforms the word embedding to hidden representation inputs. $W_{parse_{R_k}}$ is a $d \times h$ matrix for the relation R_k that similarly transforms the head word embedding (so W_{parse} is a tensor), and $W_{previous}$ and W_{next} similarly transform the previous and next words' embeddings. Figure 1 depicts this representation building at each word.

3.2 Label Representation Learning

One standard way to address the problem of multi-label classification is building binary classifiers for each possible label. Large margin classifiers have been shown to be an effective tool for this task (Pradhan et al., 2004; Kate and Mooney, 2006). We use the same idea of binary classifiers to learn one hyperplane per label, which separates the utterances with this label from all other utterances, with a large margin. In the standard way of building the classifier, each label's hyperplane is independent of other labels. To extend this model to a zero-shot learning classifier, we use parameter sharing among label hyperplanes so that similar labels have similar hyperplanes.

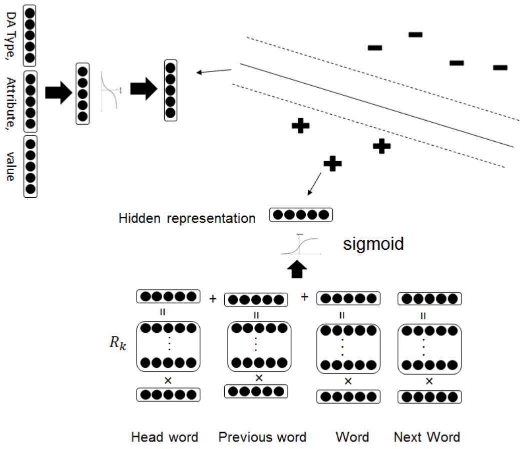

Figure 1: The multi-label classifier

We exploit the structure of labels by assuming that each hyperplane representation is a composition of representations of the label's constituent components, namely dialogue action, attribute and attribute value. We learn the composition function and the constituent representations while training the classifiers, using the labelled SLU data. The constituent representations are initialised as the word embeddings for the label constituent's name string, such as "inform", "food" and "Chinese", where these embeddings are trained on the unlabelled data. Figure 1 depicts the classifier model.

We define the hyperplane of the label $a_j(att_k = val_l)$ with its normal vector W_{a_j,att_k,val_l} as:

$$W_{a_j,att_k,val_l} = \sigma([\phi(a_j), \phi(att_k), \phi(val_l)]W_{ih})W_{ho}$$

where $\phi(\cdot)$ is the same mapping to d dimensional word vectors that is used above in the utterance representation, W_{ih} is a $3d \times h$ matrix and W_{ho} is a $h \times h$ matrix. The score of each local representation vector $\phi(U_i)$ is its distance from this label hyperplane, which is computed as the dot product of the local vector $\phi(U_i)$ with the normal vector W_{a_j,att_k,val_l}.

We sum these local scores for each position i to build the whole utterance score: $\sum_i \phi(U_i)W^T_{a_j,att_k,val_l}$. Alternatively we can think of this computation as summing the local vectors to get a whole-utterance representation $\phi(U) = \sum_i \phi(U_i)$ and then doing the dot product. The pooling method (sum) used in the model is (intentionally) over-simplistic. We did not want to distract from the main contribution of the paper, and our dataset did not justify any more complex solution since utterances are short. It can be replaced by more powerful approaches if it is needed.

To train a large margin classifier, we train all the parameters such that the score of an utterance is bigger than a margin for its labels and less than the negative margin for all other labels. Thus, the loss function is as follows:

$$\min_{\theta} \frac{\lambda}{2}\theta^2+\sum_{U} \max(0, 1-y\sum_{i} \phi(U_i)W^T_{a_j,att_k,val_l}) \tag{2}$$

where θ is all the parameters of the model, namely $\phi(w_i)$ (word embeddings), W_{word}, W_{Parse}, $W_{previous}$, W_{next}, W_{ih}, and W_{ho}. y is either 1 or -1 depending whether the input U has that label or not.

To optimize this large margin classifier we perform stochastic gradient descent by using the adagrad algorithm on this primal loss function, similarly to Pegasos SVM (Shalev-Shwartz et al., 2007), but here we backpropagate the errors to the representations to train the word embeddings and composition functions. In each iteration of the stochastic training algorithm, we randomly select an utterance and its labels as positive examples and choose randomly another utterance with a different label as a negative example. When choosing the negative sample randomly, we sample utterances with the same dialogue act but different attribute or value with 4 times higher probability than utterances with a different dialogue act. This biased negative sampling speeds up the training process since it provides more difficult training examples to the learner.

The model is able to address the adaptivity issues because the utterance and the dialogue act representations are in the same space using the same shared parameters $\phi(w)$, which are initialised with unsupervised word embeddings. It has been shown that such word embeddings capture word similarities and hence the classifier is no longer ignorant about any new attribute type or attribute value. Also, there is parameter sharing between dialogue acts because these word/label embeddings are shared, and the matrices for the composition of these representations are the same across all dialogue acts. This can help overcome sparsity in the SLU training set by transferring learning between similar situations and similar dialogue act triples. For example, if the training set does not contain any examples of the act ”request(postcode)”, but many examples of ”request(phone)”, sharing the parameters can help with the recognition of ”request(postcode)” in utterances similar to ”request(phone)”. Moreover, the SLU model is to some extent robust against paraphrasing in the input utterance because it maps the utterance to a semantic space, and uses parse bigrams. More sophisticated vector-space semantic representations of the utterance are an area for future work, but should be largely orthogonal to the contribution of this paper.

To find the set of compatible dialogue acts for a given utterance, we should check all possible dialogue acts. This can severely slow down SLU. To avoid testing all possible dialogue combinations, we build three different classifiers: The first one recognises the act types in the utterance, the second one recognises the attribute types for each of the chosen act types, and the third classifier recognises the full dialogue acts as we described above, but only for the chosen pairs of act types and attribute types.

4 SLU Experiments

In the first experiment, we measure SLU performance trained on all available data, by building a dataset that is the union of all the above datasets. This measures the performance of SLU when there is a small amount of data for an extended domain. This dataset, similarly to `SF3Ext`, has 6 main attribute types. Table 2 shows the performance of this model. We report as baselines the performance of the Phoenix system (hand crafted for this domain) and a binary linear SVM trained on the same data. The hidden layers have size *h*=*d*=50. For this experiment, we split each dataset into about 15% for the testing set and 85% for the training set.

System	Outputs	Precision	Recall	F-core
Phoenix	516	84.10	41.65	55.71
SVM	690	65.03	52.45	58.06
Our	932	90.24	81.15	85.45

Table 2: Performance on union of data (SFCore+SF1Ext+SF2Ext+SF3Ext)

Our SLU model can adapt well to the extended domain with more attribute types. We observe

model, train set	Test set		
	SF1Ext	SF2Ext	SF3Ext
	P—R—F	P—R—F	P—R—F
Our SFcore	73.36—66.11—69.54	74.61—59.73—66.34	72.54—53.86—61.81
SVM SFcore	50.66— 38.7— 43.87	49.64—34.70— 40.84	48.99—30.91—37.90
Our SF1Ext		83.18—66.08—73.65	78.32—59.98—67.93
SVM SF1Ext		58.72—41.71—48.77	53.25—34.88—42.15
Our SF2Ext			84.12—67.78—75.07
SVM SF2Ext			59.27—42.80—49.70

Table 3: SLU performance: trained on a smaller domain and tested on more inclusive domains.

particularly that the recall is almost twice as high as the hand-crafted baseline. This shows that our SLU can recognise most of the dialogue acts in an utterance, where the rule-based Phoenix system and a classifier without composed output cannot. Overall there are 1042 dialogue acts in the test set. SLU recall is very important in the overall dialogue system performance, as the effect of a missed dialogue act is hard to handle for the Interaction Manager. Both hand-crafted and our system show relatively high precision.

In the next experiment, we measure how well the new SLU model performs in an extended domain without any training examples from that extended domain. We train a SLU model on each subset, and test it on each of the more inclusive subsets. Table 3 shows the results.

Not surprisingly, the performance is better if SLU is trained on a similar domain to the test domain, and adding more attribute types and values decreases the performance more. But our SLU can generalise very well to the extended domain, achieving much better generalisation that the SVM model.

4.1 Conclusion

In this paper, we describe a new SLU model that is designed for improved domain adaptation. The multi-label classification problem of dialogue act recognition is addressed with a classifier that learns to build an utterance representation and a dialogue act representation, and decides whether or not they are compatible. The dialogue act representation is a vector composition of its constituent labels' embeddings, and is trained as the hyperplane of a large margin binary classifier for that dialogue act. The utterance representation is trained as a composition of word embeddings. Since the utterance and the dialogue act representations are both built using unsupervised word embeddings and share these embedding parameters, the model can address the issues of domain adaptation. Word embeddings capture word similarities, and hence the classifier is able to generalise from known attribute types or values to similar novel attribute types or values. We tested this SLU model on datasets where the number of attribute types and values is increased, and show much better results than the baselines, especially in recall. The model succeeds in both adapting to an extended domain using relatively few training examples and in recognising novel attribute types and values.

Acknowledgments

The research leading to this work was funded by the EC FP7 programme FP7/2011-14 under grant agreement no. 287615 (PARLANCE), and Hasler foundation project no. 15019, Deep Neural Network Dependency Parser for Context-aware Representation Learning. The authors also would like to thank Dr.Helen Hastie for her help in annotating the dataset.

References

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 238–247.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167.

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

M Gašic, D Kim, P Tsiakoulis, C Breslin, M Henderson, M Szummer, B Thomson, and S Young. 2014. Incremental on-line adaptation of pomdp-based dialogue managers to extended domains.

Yulan He and Steve Young. 2005. Semantic processing using the hidden vector state model. *Computer Speech and Language*, 19:85–106.

Matthew Henderson, Milica Gašić, Blaise Thomson, Pirros Tsiakoulis, Kai Yu, and Steve Young. 2012. Discriminative Spoken Language Understanding Using Word Confusion Networks. In *Spoken Language Technology Workshop, 2012.*

James Henderson, Paola Merlo, Ivan Titov, and Gabriele Musillo. 2013. Multilingual joint parsing of syntactic and semantic dependencies with a latent variable model. *Comput. Linguist.*, 39(4):949–998.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving Word Representations via Global Context and Multiple Word Prototypes. In *Annual Meeting of the Association for Computational Linguistics (ACL).*

Rohit J. Kate and Raymond J. Mooney. 2006. Using string-kernels for learning semantic parsers. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 913–920.

R.; Perona L. Fei-Fei; Fergus. 2006. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 28:594–611, April.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781.*

Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013).*

Mark Palatucci, Dean Pomerleau, Geoffrey E. Hinton, and Tom M. Mitchell. 2009. Zero-shot learning with semantic output codes. In *Advances in Neural Information Processing Systems 22*, pages 1410–1418.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, and James H. Martin. 2004. Shallow semantic parsing using support vector machines.

Richard Schwartz, Scott Miller, David Stallard, and John Makhoul. 1996. Language understanding using hidden understanding models. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 2, pages 997–1000. IEEE.

Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. 2007. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th International Conference on Machine Learning*, pages 807–814.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

Wayne Ward. 1994. Extracting information in spontaneous speech. In *ICSLP*.

Jason Weston, Samy Bengio, and Nicolas Usunier. 2010. Large scale image annotation: Learning to rank with joint word-image embeddings. *Mach. Learn.*, 81.

Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. Wsabie: Scaling up to large vocabulary image annotation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, pages 2764–2770.

Majid Yazdani and Andrei Popescu-Belis. 2013. Computing text semantic relatedness using the contents and links of a hypertext encyclopedia. *Artif. Intell.*, 194:176–202.