An End-to-end Approach to Learning Semantic Frames with Feedforward Neural Network

Yukun Feng, Yipei Xu and Dong Yu*
College of Information Science
Beijing Language and Culture University
No.15 Xueyuan Rd., Beijing, China, 100083
{fengyukun,xuyipei,yudong}@blcu.edu.cn

Abstract

We present an end-to-end method for learning verb-specific semantic frames with feedforward neural network (FNN). Previous works in this area mainly adopt a multi-step procedure including part-of-speech tagging, dependency parsing and so on. On the contrary, our method uses a FNN model that maps verb-specific sentences directly to semantic frames. The simple model gets good results on annotated data and has a good generalization ability. Finally we get 0.82 F-score on 63 verbs and 0.73 F-score on 407 verbs.

1 Introduction

Lexical items usually have particular requirements for their semantic roles. Semantic frames are the structures of the linked semantic roles near the lexical items. A semantic frame specifies its characteristic interactions with things necessarily or typically associated with it (Alan, 2001). It is valuable to build such resources. These resources can be effectively used in many natural language processing (NLP) tasks, such as question answering (Narayanan and Harabagiu, 2004) and machine translation (Boas, 2002).

Current semantic frame resources, such as FrameNet (Baker et al., 1998), PropBank (Palmer et al., 2005) and VerbNet (Schuler, 2005), have been manually created. These resources have promising applications, but they are time-consuming and expensive. El Maarouf and Baisa (2013) used a bootstrapping model to classify the patterns of verbs from Pattern Dictionary of English[1] (PDEV). El Maarouf et al. (2014) used a Support Vector Machine (SVM) model to classify the patterns of PDEV . The above supervised approaches are most closely related to ours since PDEV is also used in our experiment. But the models above are tested only on 25 verbs and they are not end-to-end. Popescu used Finite State Automata (FSA) to learn the pattern of semantic frames (Popescu, 2013). But the generalization ability of this rule-based method may be weak. Recently, some unsupervised studies have focused on acquiring semantic frames from raw corpora (Materna, 2012; Materna, 2013; Kawahara et al., 2014b; Kawahara et al., 2014a). Materna used LDA-Frame for identifying semantic frames based on Latent Dirichlet Allocation (LDA) and the Dirichlet Process. Kawahara et al. used Chinese Restaurant Process to induce semantic frames from a syntactically annotated corpus. These unsupervised approaches have a different goal compared with supervised approaches. They aim at identifying the semantic frames by clustering the parsed sentences but they do not learn from semantic frames that have been built. These unsupervised approaches are also under a pipeline framework and not end-to-end.

One related resource to our work is Corpus Pattern Analysis (CPA) frames (Hanks, 2012). CPA proposes a heuristic procedure to obtain semantic frames. Most current supervised and unsupervised approaches are under similar pipeline procedure. The procedure can be summarized as follows with an example sentence "The old music deeply moved

*The corresponding author.

[1] http://pdev.org.uk/

Proceedings of NAACL-HLT 2016, pages 1–7,
San Diego, California, June 12-17, 2016.

the old man":

step 1 Identify the arguments near "moved", which can be expressed as (subject:music, object:man)

step 2 Attach meanings to above arguments, which can be expressed as (subject:Entity, object:Human)

step 3 Clustering or classifying the arguments to get semantic frames.

However, step 1 and 2 are proved to be difficult in SemEval-2015 task 15 [2] (Feng et al., 2015; Mills and Levow, 2015).

This paper presents an end-to-end approach by directly learning semantic frames from verb-specific sentences. One key component of our model is well pre-trained word vectors. These vectors capture fine-grained semantic and syntactic regularities (Mikolov et al., 2013) and make our model have a good generalization ability. Another key component is FNN model. A supervised signal allows FNN to learn the semantic frames directly. As a result, this simple model achieves good results. On the instances resources of PDEV, we got 0.82 F-score on 63 verbs and 0.73 on 407 verbs.

The contributions of this paper are summarized as follows:

- Semantic frames can be learned with neural network in an end-to-end map and we also analysed our method in detail.
- We showed the power of pre-trained vectors and simple neural network for the learning of semantic frames. It is helpful in developing a more powerful approach.
- We evaluate the learned semantic frames on annotated data precisely and got good results with not much training data.

2 Model Description

2.1 Overview

Our model gets verb-specific semantic frames directly from verb-specific sentences. A running example of learning semantic frames is shown in Figure 1. Our FNN model can be regarded as a continuous function

$$c = f(x). \tag{1}$$

Here $x \in \mathbb{R}^n$ represents the vector space of the sentence and c represents the index of the semantic frame. Instead of a multi-step , FNN model directly maps the sentence into semantic frame. In the training phrase "The old music deeply moved the old man" is mapped into vector space and "Entity move Human" is learned from the vector space. In the testing phrase, an example result of FNN model can roughly expressed as "Entity move Human" = f("The fast melody moved the beautiful girl") = which is an end-to-end map.

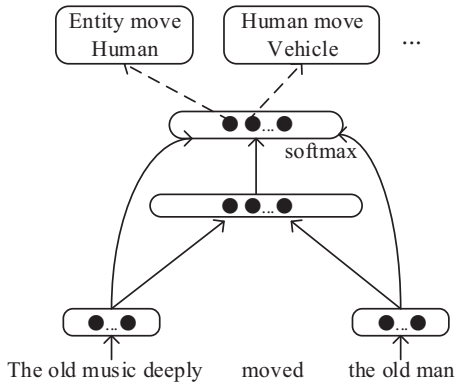

Figure 1: Model architecture for an example of learning semantic frames directly from verb-specific sentence. The sentence is divided into two windows. "The old music deeply" is in the left window and "the old man" is in the right window. The target verb "moved" is not used in the input. The input is connected to output layer. Each unit of output layer corresponds to one semantic frame of the target verb.

2.2 Feedforward Neural Network

Denote $C_{i:j}$ as the concatenation of word vectors in a sentence. Here i and j are word indexes in the sentence. The input layer is divided into two windows (padded with zero vector where necessary), which are called left window and right window. The input for FNN is represented as

$$x = C_{v-lw:v-1} \oplus C_{v+1:v+rw}, \tag{2}$$

where v denotes the index of target verb in the sentence, $\oplus$ is the concatenation operator, lw is the

[2] http://alt.qcri.org/semeval2015/task15/

length of left window and rw is the length of right window. Both lw and rw are hyperparameters. The target verb can be ignored by the input layer because the arguments of it lie on the left and right windows. W, U and V respectively represent the weight matrix between input and hidden layer, hidden and output layer and input and output layer. d and b respectively represent the bias vector on hidden and output layer. We use hyperbolic function as our activation function in hidden layer. Using matrix-vector notation, the net input of softmax layer of FNN can be expressed as:

$$a = \lambda(U\text{tanh}(Wx + d) + b) + (1 - \lambda)Vx. \quad (3)$$

Here λ controls the relative weight of the two items in the above formula. FNN will have three layers when λ is set to 1 and two layers without bias when λ is set to 0. Then a softmax function is followed for classification:

$$p_i = \frac{e^{a_i}}{\sum_i e^{a_i}}. \quad (4)$$

Here p_i represents the probability of the semantic frame i given x. The cost we minimize during training is the negative log likelihood of the model plus the L2 regularization term. The cost can be expressed as:

$$L = -\sum_{m=1}^{M} \text{log}p_{t_m} + \beta\text{R}(U, W, V). \quad (5)$$

Here M is number of training samples and t_m is the index of the correct semantic frame for the m'th sample. R is a weight decay penalty applied to the weights of the model and β is the hyperparameter controlling the weight of the regularization term in the cost function.

2.3 Model Analysis

We extend equation 1 as

$$c = f(w_{v-lw}, ...w_i..., w_{v+rw}). \quad (6)$$

w_i is the i'th word vector in the input vector space above. Note f is a continuous function and similar words are likely to have similar word vectors. That is to say, if $c_1 = f(w_{v-lw}, ...w_i..., w_{v+rw})$ we usually have $c_1 = f(w_{v-lw}, ...sw_i..., w_{v+rw})$ with w_i similar to sw_i. One obvious example but roughly expressed is if "Entity move Human" = f("The","old","music","the","old","man"), then it will have "Entity move Human" = f("The","fast","melody","the","beautiful","girl"). Because "music" and "melody" can be regarded as similar words, which is also the case for "man" and "girl". Since one of the critical factors for semantic frame is semantic information in specific unit (e.g., subject and object), the pre-trained vectors can easily capture what this task needs. Thus pre-trained vectors can have a good generalization ability for semantic frame learning. In the training phrase, FNN can learn to capture the key words which have more impact on the target verb. This will be shown later in the experiment. Because the input of FNN is a window with fixed length, this would cause a limited ability of capturing long-distance key words. Despite this weakness of this model, it still got good results.

3 Experiments

3.1 Task and Datasets

SemEval-2015 Task 15 is a CPA (Hanks, 2012) dictionary entry building task. The task has three subtasks. Two related subtasks are summarized as follows [3]:

- CPA parsing. This task requires identifying syntactic arguments and their semantic type of the target verb. The result of this task followed by our example sentence can be "The old [subject/Entity music] deeply moved the old [object/Human man]". The syntactic arguments in the example are "subject" and "object" respectively labelled on the word "music" and "man". Their semantic types are "Entity" and "Human". Thus a pattern of the target verb "move" can be "[subject/Entity] move [object/Human]".

- CPA Clustering. The result of the first task give the patterns of the sentences. This task aims at clustering the most similar sentences according to the found patterns. Two sentences which belong to the similar pattern are more likely in the same cluster.

[3] Subtask 3 is CPA Automatic Lexicography. Since we have nothing to do with this task, we don't make a introduction.

Datasets Statistics					B-cubed or micro-average F-score of Methods			
	Verb number	Training data	Testing data	Semantic frame number	FNN	SEB	DULUTH	BOB90
MTDSEM	4	136.5	159	4.86	0.7	0.59	0.52	**0.74**
	3	1546.33	214.67					
PDEV1	407	373.49	158.32	6.53	0.73	0.63	-	-
PDEV2	63	1421.22	606.60	9.60	0.82	0.64	-	-

Table 1: Summary statistics for the datasets (left) and results of our FNN model against other methods (right). On the right side, MTDSEM is evaluated by B-cubed F-score for clustering. On PDEV1 and PDEV2, FNN model is evaluated by micro-average F-score. SEB is always evaluated by B-cubed F-score as the base score. DULUTH and BOB90 are Participant teams in 2015.

SemEval-2015 Task 15 has two datasets which are called Microcheck dataset and Wingspread dataset. The dataset of SemEval-2015 Task 15 was derived from PDEV (Baisa et al.,). That is to say, all the sentences in SemEval-2015 Task 15 are from PDEV. These datasets have a lot of verbs and have many sentences for each verb. Each sentence of each verb corresponds to one index of the semantic frames. Note that the semantic frames are verb-specific and each verb has a close set of its own semantic frames. Thus in our experiment we build one model for each verb. Our task is to classify each sentence directly into one semantic frame which is different from CPA clustering, but we will also test our model with clustering metric against other systems. We only remove punctuation for these datasets. To test our model we split these datasets into training data and testing data. Summary statistics of the these datasets are in Table 1. In Table 1, Figure 2 and Table 3, Verb number is the number of verbs, Training data and Testing data represent the average number of sentences for each verb and Semantic frame number is the average number of semantic frames for each verb. Details of creating the datasets are as follows:

- **MTDSEM**: Microcheck test dataset of SemEval-2015 Task 15. For each verb in MTDSEM we select training sentences from PDEV that doesn't appear in MTDSEM.
- **PDEV1**: For each verb, we filter PDEV with the number of sentences not less than 100 and the number of semantic frames not less than 2. Then we split the filtered data into training data and testing data, respectively accounted for 70% and 30% for each semantic frame of each verb.
- **PDEV2**: Same with PDEV1, but with the difference of threshold number of sentences set to 700. PDEV2 ensures that the model has relatively enough training data.
- **MTTSEM**: Microcheck train dataset and test dataset of SemEval-2015 Task 15. We split MTTSEM as above to get training data and testing data for each verb. The summary statistic of this dataset is separately shown in Table 3.

We use the publicly available word2vec vectors that were trained through GloVe model (Pennington et al., 2014) on Wikipedia and Gigaword. The vectors have dimensionality of 300. The word vectors not in pre-trained vectors are set to zero.

3.2 Experimental Setup

We build one model for each verb. Training is done by stochastic gradient descent with shuffled minibatches and we keep the word vectors static only update other parameters. In our experiments we keep all the same hyperparameters for each verb. we set learning rate to 0.1, lw and rw to 5, minibatch size to 5, L2 regularization parameter β to 0.0001, the number of hidden unit to 30 and λ to 0. Because of limited training data, we do not use early stopping. Training will stop when the zero-one loss is zero over training data for each verb. The official evaluation method used B-cubed definition of Precision and Recall (Bagga and Baldwin, 1998) for CPA clustering. The final score is the average of B-cubed F-scores over all verbs. Since our task can be regarded as a supervised classification, we also use the micro-average F-score to evaluate our results.

3.3 Experimental Results

Table 1 shows the results on MTDSEM with supervised and unsupervised approaches. SemEval-2015 Task 15 baseline (SEB) clusters all sentences together for each verb. That is to say, SEB assigns the

Verb-specific Sentences	Verb-specific Semantic Frames
Mary resisted the temptation to answer her back and after a moment's silence	[[Human 1]] answer ([[Human 2]]) back [[Human 1]]
Pamala Klein would seem to have a lot to answer for.	[[Human]] have a lot to answer for [NO OBJ]
and I will answer for her safety	[[Human]] answer [NO OBJ] for [[Eventuality]]
he cannot answer for Labour party policies	[[Human]] answer [NO OBJ] for [[Eventuality]]
it is fiction and can not be made real by acting it out	[[Human]] act [[Event or Human Role or Emotion]] out
You should try to build up a network of people you trust	[[Human]] build ([[Entity]]) up

Table 2: Example results of our FNN model mapping verb-specific sentences to semantic frames on PDEV.

same cluster to all the sentences and is evaluated by B-cubed F-score for clustering. So its score depends on the distribution of semantic frames. The higher the score is, the more concentrated the distribution of semantic frames is. SEB to get higher score usually indicates other methods are more likely to get high scores, so we use it as a base score. DULUTH (Pedersen, 2015) treated this task as an unsupervised word sense discrimination or induction problem. The number of semantic frames was predicted on the basis of the best value for the clustering criterion function. BOB90 [4] used a supervised approach to tackle the clustering problem (Baisa et al., 2015) and get the best score on MTDSEM. An example result of FNN model on PDEV is shown in Table 2

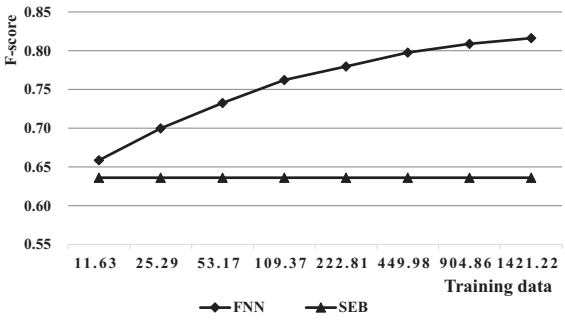

Figure 2: Results of FNN on PDEV2. The testing data is fixed at 606.60. The training data increases two times at each step. Y-axis represents B-cubed F-score for SEB and micro-average F-score for FNN.

4 Discussions

4.1 Large vs. Small Training Data

MTDSEM is divided into two parts to report on the left part of Table 1. One part has larger training data while the other part has little. Our FNN model gets a relatively lower score, mainly because the part of training data is too small. FNN got 0.88 B-cubed F-score on the larger training data part and 0.57 on the other part. In order to show the real power of our model, PDEV1 and PDEV2 were made which have much more training data than MTDSEM and more verbs to test. It shows a better result on hundreds of verbs. We also made Figure 2 to show the performance of FNN model when the training data size increases. As a result, our method can perform really well on sufficient training data.

4.2 The Direct Connection

Our FNN model has a direct connection from input to output layer controlled by λ in the second term of the equation 3. It is designed to speed up the convergence of training (Bengio et al., 2006), since the direct connection allows the model fast learning from the input. But In our experiments the number of epoch before the convergence of training is very close between FNN with two layers and FNN with three layers. On the contrary, we observed that FNN with two layers where λ is set to zero got a slightly better F-score than FNN where λ is set to 0.5 and 1. This may suggest FNN with two layers is good enough on PDEV.

4.3 The Ability of Capturing Key Words

FNN have the ability to capture the key words of the target verb. To show this, we test our FNN model on MTTSEM with different preprocessing shown in Table 3. We only remove the punctuation of MTTSEM1 which is same as before. MTTSEM2 only contains the gold annotations of syntactic arguments provided by CPA parsing. Note that MTTSEM2 only contains the key words for each target verb and ignore those unimportant words in the sentences. MTTSEM3 is same as MTTSEM2 but with the difference of the arguments for each tar-

[4] BOB90 did not submit an article

get verb provided by Stanford Parser (De Marneffe et al., 2006). Dependents that have the following relations to the target verb are extracted as arguments:

nsubj, xsubj, dobj, iobj, ccomp, xcomp, prep_*

As a result, FNN reasonably gets the best score on MTTSEM2 and FNN also gets a good score on MTTSEM1 but much lower score on MTTSEM3. This shows that FNN would have the ability to capture the key words of target verb. The result on MTTSEM1 and MTTSEM3 shows that our FNN model captures the key words more effectively than the parser for this task.

	MTTSEM1 (verb-specific sentences)	MTTSEM2 (gold annotations)	MTTSEM3 (automatic annotations)
Verb number	28		
Training data	111.25		
Testing data	46.39		
FNN	0.76	**0.82**	0.67
SEB	0.62		

Table 3: Results on MTTSEM with different preprocessing.

5 Conclusion

This paper has described an end-to-end approach to obtain verb-specific semantic frames. We evaluated our method on annotated data. But we do not identify the semantic roles for target verbs and the verb-specific model suffers not enough training data. A promising work is to merge these semantic frames over multiple verbs which can greatly increase the training data size. Also, convolutional layer can be applied on the input vector to extract features around verb and more powerful neural network can be used to model the verb.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful suggestions and comments.

References

Keith Alan. 2001. In *Natural Language Semantics*, page 251. Blackwell Publishers Ltd, Oxford.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 79–85. Association for Computational Linguistics.

Vít Baisa, Ismaïl El Maarouf, Pavel Rychlỳ, and Adam Rambousek. Software and data for corpus pattern analysis.

Vıt Baisa, Jane Bradbury, Silvie Cinková, Ismaıl El Maarouf, Adam Kilgarriff, and Octavian Popescu. 2015. Semeval-2015 task 15: A corpus pattern analysis dictionary-entry-building task.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics.

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. 2006. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer.

Hans Christian Boas. 2002. Bilingual framenet dictionaries for machine translation. In *LREC*.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.

Ismaıl El Maarouf and Vıt Baisa. 2013. Automatic classification of patterns from the pattern dictionary of english verbs. In *Joint Symposium on Semantic Processing.*, page 95.

Ismail El Maarouf, Jane Bradbury, Vít Baisa, and Patrick Hanks. 2014. Disambiguating verbs by collocation: Corpus lexicography meets natural language processing. In *LREC*, pages 1001–1006.

Yukun Feng, Qiao Deng, and Dong Yu. 2015. Blcunlp: Corpus pattern analysis for verbs based on dependency chain. *Proceedings of SemEval*.

Patrick Hanks. 2012. How people use words to make meanings: Semantic types meet valencies. *Input, Process and Product: Developments in Teaching and Language Corpora*, pages 54–69.

Daisuke Kawahara, Daniel Peterson, and Martha Palmer. 2014a. A step-wise usage-based method for inducing polysemy-aware verb classes. In *ACL (1)*, pages 1030–1040.

Daisuke Kawahara, Daniel Peterson, Octavian Popescu, Martha Palmer, and Fondazione Bruno Kessler. 2014b. Inducing example-based semantic frames from a massive amount of verb uses. In *EACL*, pages 58–67.

Jiří Materna. 2012. Lda-frames: An unsupervised approach to generating semantic frames. In *Computational Linguistics and Intelligent Text Processing*, pages 376–387. Springer.

Jirí Materna. 2013. Parameter estimation for lda-frames. In *HLT-NAACL*, pages 482–486.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

Chad Mills and Gina-Anne Levow. 2015. Cmills: Adapting semantic role labeling features to dependency parsing. *SemEval-2015*, page 433.

Srini Narayanan and Sanda Harabagiu. 2004. Question answering based on semantic structures. In *Proceedings of the 20th international conference on Computational Linguistics*, page 693. Association for Computational Linguistics.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Ted Pedersen. 2015. Duluth: Word sense discrimination in the service of lexicography. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 438–442, Denver, Colorado, June. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Octavian Popescu. 2013. Learning corpus patterns using finite state automata. In *Proceedings of the 10th International Conference on Computational Semantics*, pages 191–203.

Karin Kipper Schuler. 2005. Verbnet: A broad-coverage, comprehensive verb lexicon.