Semantic Relation Classification for Concepts of Wordnet with Deep Learning

Khuyagbaatar Batsuren[1], Altangerel Chagnaa[2], Amarsanaa Ganbold[2] and Zoljargal Munkhjargal[2]
[1]Department of Information Engineering and Computer Science
University of Trento, Trento, Italy
k.batsuren@unitn.it
[2]Department of Information and Computer Science
National University of Mongolia, Ulaanbaatar, Mongolia
{altangerelc; amarsanaag; zoljargalm}@gmail.com

***Abstract*—Machines can currently perform no better results than humans for many simple tasks such as Named-Entity Recognition, Question Answering and other tasks. To reach over the human performances, big structured knowledge bases(KB) are needed for machines to train. This structured KB is usually and manually built by humans. These KBs store millions of facts about the world. Yet despite their seemingly huge size, these repositories are still far from complete. Enriching or completing existing KBs are very tedious for humans and crowd workers, and should be very large and virtually unbound. Therefore, Semantic Relation Classification for Concepts of Wordnet has been very active and challenging task. For this task, the state-of-art methods using deep learning used no gloss information of concepts. From being motivated by it, in this article we have proposed a new deep architecture which takes a label and gloss in concepts as inputs. The experimental results showed that our method has a very big capability to compete and improve the state-of-art performance. In future work, we will extract semantic relations from a text data through world wide websites and then enrich existing KBs by using crowdsourcing techniques to verify extracted semantic relations.**

Keywords—semantic relation extraction; automated knowledge base construction; deep learning;

I. INTRODUCTION

Recently, machines can perform no better results than humans for many simple tasks such as Named-Entity Recognition, Question Answering and other tasks. To reach over the human performances, big structured knowledge bases(KB) are needed for machines to train. Structured KBs including Wikipedia, Freebase[3], YAGO[5], Microsoft's Satori, Wordnet[6], and Google's Knowledge Graph are usually and manually built by humans or crowd workers and should be very large and virtually unbound. These KBs store millions of facts about the world. Yet despite their seemingly huge size, these repositories are still far from complete. Enriching or completing existing KBs are very tedious for humans and crowd workers, and virtually unbound.

One successful example of those structured KBs is Universal Knowledge Core which includes hundreds of thousands of concepts of the real-world entities. It consists of three components: concept core, domain core and natural language core. While constructing concept and domain core, it requires people to describe concepts and relations between them. According to [8], they first collected concepts from various sources including GeoNames, TGN, and WordNet while building Space domain. Then, they build explicitly semantic relations between concepts. As I mentioned earlier, building semantic relations is very tedious work for human and recognizing semantic relations for machines is still open issue while building concept core of a domain.

The use of word representations pre-trained in an unsupervised fashion from lots of text has become a key "secret sauce" for the success of many NLP systems in recent years. The word representations computed by using Neural Networks (NN) already captured semantic and syntactic features.

Based on those word embeddings, in this report we proposed deep architecture to recognize semantic relation between concepts.

The remainder of the paper is organized as follows: Section 2 contains more details about our proposed method: Deep Neural Network for Semantic Relation Classification for Concepts of WordNet. Subsequently, how the our neural are training and how we made big text corpora and leveraging it are explained. In section 3, the experimental analysis and the related results are provided. Last section addresses our conclusion.

II. PROPOSED METHOD

The purpose of our method is to recognize a semantic label between two concepts. Vector representations of words of label and gloss in concepts are taken as inputs. As input we will try to pre-process our features as little as possible and the use a multi-layer neural network architecture, trained in end-to-end fashion. The architecture takes a sequence of words or whole sentence and learns several layers of feature extraction that process the inputs. The features in the deep neural network are automatically trained by backpropagation.

The deep neural network of our proposed method is summarized in Fig. 1 and 2. In Fig. 1, first layer named sentence approach transform each sequence of words or each sentence to global features which is using a Sentence approach explained details in subsection A. Second layer extracts

features from a combination of all global features. The following layers are standard NN layers.

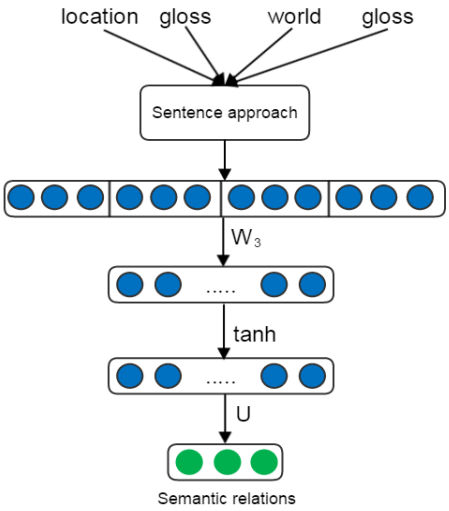

Fig. 1. Our proposed deep neural network.

A. *Sentence Approach*

The Sentence approach is first introduced in [2], which is shown in Fig. 2. It successively takes the complete sentence, passes it through the lookup table layer (1), produces local features around each word of the sentence thanks to convolutional layers, combines these feature into a global feature vector which can then be fed to standard affine layers (4).

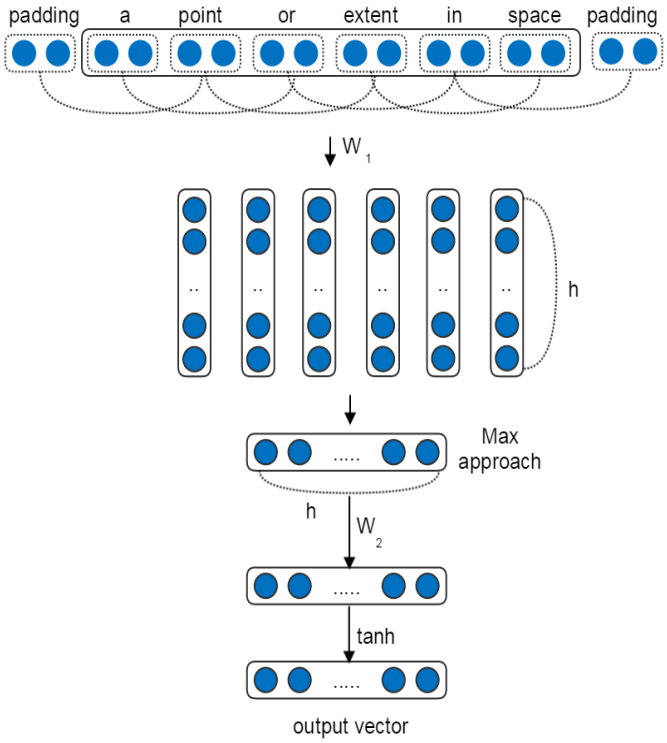

Fig. 2. Neural network for the Sentence approach[5].

All word vectors are n-dimensional and saved in a large matrix $L \in \mathbb{R}^{nXV}$ where V is the size of the vocabulary and n is the word vector size to be chosen by the user. Each word has an unique index, that is representing a column number of L matrix. The input sentence or sequence of words is represented as $\{s_1, s_2, ..., s_N\}$ of N words and thus is transformed into a series of vectors $\{x_1, x_2, ..., x_N\}$ by applying L matrix each of its words. The feature window is not well defined for words near the beginning or the end of sentence. Therefore, the special beginning and end tokens are utilized, which we define as <s> and </s> respectively.

We define the vector corresponding to the begin padding as x_s and for the end padding as x_e. Hence, we will get the following sequence of vectors:

$$(x_s, x_1, x_2, ..., x_N, x_e).$$

For the sentence, we have the following windows that we will give as input to the neural network of the sentence approach as shown in Fig. 2: (In this example, we assumed a size of a window is three.)

$$([x_s, x_1, x_2], [x_1, x_2, x_3], [x_2, x_3, x_4], \ldots, [x_{N-1}, x_N, x_e]).$$

Then, as mentioned before, Max approach extracts global features from local features on each window of words. In next layer, global features are processed with simple neural layer. As the result, the global feature vector representing the gloss or label of the concept is extracted.

These global feature vectors are utilized in our proposed method to classify semantic relations between concepts of WordNet.

B. *Feedforward function for Our architecture*

As mentioned above, each concept's gloss and label vectors are given as input to multi-layer neural network, which has one hidden layer. This hidden layer has dimensionality H1 . The first hidden layer extracts a combination features from local features on concepts. The hidden layer is also used as a combination of local features for a SoftMax classifier which will return a probability for R semantic relation between concepts. For instance the feed-forward equations are as follows.

$$z^{(1)} = W^{(1)} \begin{bmatrix} l_1 \\ g_1 \\ l_2 \\ g_2 \\ R \end{bmatrix} + b^{(1)} \quad (1)$$

$$a^{(1)} = f(z^{(1)}) \quad (2)$$

$$h = a^{(1)} \quad (3)$$

where the model parameters $W^{(1)} \in \mathbb{R}^{H_1 \times 4C}$, $b^{(1)} \in \mathbb{R}^{H_1 \times 1}$ where C is the feature vector size of a gloss or a label, r is the number of classes and the model function f. The final prediction, h, is the probabilities of each class for CNER task.

The nonlinearity function f can be either the sigmoid function or the hyperbolic tanh function. For our model, the tanh function is used. One useful property of tanh is that its

derivative can be expressed in terms of the function value itself:

$$\frac{d}{dx}\tanh x = 1 - \tanh^2 x \quad (4)$$

Now, let us summarize this whole procedure, the network and its inputs by the following notation. The training inputs consist of a set of (window, label) pairs $\left(e_1^{(i)}, R^{(i)}, e_2^{(i)}\right)$ with i ranging from 1 to m, the number of all pairs(e_1, R, e_2) in the training corpus. Each $x^{(i)} = [l_1^{(i)}, g_1^{(i)}, l_2^{(i)}, g_2^{(i)}, R^{(i)}]$ and $y^{(i)} \in \{0,1\}^{r\times 1}$. The θ parameters are defined to hold all network parameters: $\theta = (L, W^{(1)}, \mathrm{b}^{(1)})$. Using these notations, our entire neural network is defined in the one function as follows:

$$h_\theta\left(x^{(i)}\right) = f\left(W^{(1)}x^{(i)} + b^{(1)}\right) \quad (5)$$

Our neural network is trained by maximizing a likelihood over the training data, using stochastic gradient descent(SGD). We want to maximize the following log-likelihood, called the cost function, with respect to θ which is all trainable parameters of the network.

$$J(\theta) = \frac{1}{m}\sum_1^m \left[h_\theta\left(x^{(i)}\right)_{y^{(i)}} - \log\left(\sum_{t=1}^r e^{h_\theta\left(x^{(i)}\right)_t}\right)\right] + \frac{2C}{m}\|\theta\|_2^2 \quad (6)$$

where m is the number of all training instances. As shown in (6), this cost function is simply the sum of all individual costs over m training examples, plus a regularization term, which we try to maximize. A regularization is used to avoid the overfitting.

2.3.1 Stochastic Gradient Maximizing (6) with stochastic gradient is achieved by iteratively selecting a random example (x, y) and making a gradient step:

$$\theta \leftarrow \theta + \lambda\frac{\partial J(\theta)}{\partial\theta} \quad (7)$$

where λ is a chosen learning rate. Our neural network described in Figure 1 is a succession of layers that correspond to the successive composition of functions. The neural network is finally composed with the cost function(6). Thus, an analytical formulation of the derivative can be computed, by applying the differentiation chain rule through the network, and through the cost function(9).

2.3.2 Random initializations In the beginning of the training we initialize these parameters randomly. One effective strategy for random initialization is to randomly select values for $W^{(1)}$ uniformly in the range $[-\epsilon_{init}, \epsilon_{init}]$. The one effective strategy (Manning, 2012) for choosing ϵ_{init} is to base it on the number of units feeding into the layer and the number of units of this current layer as follows:

$$\epsilon_{init} = \frac{\sqrt{6}}{\sqrt{fanIn + fanOut}} \quad (8)$$

where $fanIn = 4\text{x}C$ and $fanOut = H_1$ in our case. This range of values ensures that the parameters are kept small and makes the learning more efficient.

Although word representations are automatically trained by the backpropagation during the training process for our neural network, those embeddings do not carry good syntactic and semantic information. Recently, several language models (LM) which produce word embeddings carrying more semantic and syntactic features have been proposed in [1] [2] [4] [12] [14] [15] [16] [17] [18][19].

III. Experimental Results and Discussion

Experiments are conducted on both WordNet [6] to predict whether some relations hold using other facts in the database. This can be seen as common sense reasoning over known facts or link prediction in relationship networks. For instance, if somebody was born in *London*, then their nationality would be *British*. If a *German Shepard* is a *dog*, it is also a *vertebrate*.

We first describe the datasets, then compare the above models and conclude with several analyses of important modeling decisions, such as whether to use entity vectors or word vectors. The WordNet relationships we consider are *has instance, type of, member meronym, member holonym, part of, has part, subordinate instance of, domain region, synset domain region, similar to, domain topic*.

For WordNet we use 112,581 relational triplets for training. In total, there are 38,696 unique entities in 11 different relations. One important difference to previous work is our dataset generation which filters trivial test triplets. We filter out tuples from the testing set if either or both of their two entities also appear in the training set in a different relation or order. For instance, if (e_1,*similar to*,e_2) appears in training set, we delete (e_2,*similar to*,e_1) and (e_1,*type of*,e_2), etc from the testing set. In the case of synsets containing multiple words, we pick the first, most frequent one.

Our proposed method achieves an accuracy of 71.2% with semantically initialized word vectors and 62.2% with randomly initialized ones.

IV. Conclusion

In this article we proposed the deep neural network for Semantic Relation Classification for Concepts of WordNet. In experimental results we showed that our architecture is very good to take advantages of pre-trained word representations and competitive with the state-of-the-art performances. In future work, we will work in more details of experimental works and try to extend it.

References

[1] Alexandrescu, A. and Kirchhoff, K. Factored neural language models. Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL(NAACL), 1-4, 2006.

[2] Luong, M. et al. Better word representations with recursive neural networks for morphology. Conference on Computational Natural Language Learning(CoNLL), 104-113, 2013.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD), 2008.

[4] Bengio, Y. et al. A neural probabilistic language model. Journal of Machine Learning Research, 3, 1137–1155, 2003

[5] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. The Journal of Machine Learning Research, 12, 2493-2537.

[6] G.A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 1995.

[7] F.M. Suchanek, G. Kasneci, and G.Weikum. Yago: a core of semantic knowledge. In Proceedings of the 16th international conference on World Wide Web, 2007.

[8] Giunchiglia, Fausto and Maltese, Vincenzo and Farazi, Feroz and Dutta, Biswanath. GeoWordNet: a resource for geo-spatial applications. Technical Report DISI-09-071, Department of Information Engineering and Computer Science, University of Trento. Proc. of the 7th Extended Semantic Web Conference (ESWC) 2010.

[9] J. Turian, L. Ratinov, and Y. Bengio. Word representations: a simple and general method for semisupervised learning. In Proceedings of ACL, pages 384–394, 2010.

[10] A. Bordes, X. Glorot, J. Weston, and Y. Bengio. Joint Learning of Words and Meaning Representations for Open-Text Semantic Parsing. AISTATS, 2012.

[11] A. Bordes, J. Weston, R. Collobert, and Y. Bengio. Learning structured embeddings of knowledge bases. In AAAI, 2011.

[12] R. Collobert and J. Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In ICML, 2008.

[13] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng. Improving Word Representations via Global Context and Multiple Word Prototypes. In ACL, 2012.

[14] Mikolov, T. and Zweig, G. Context dependent recurrent neural network language model. In Proceedings of Spoken Language Technologies(SLT), 234-239, 2012.

[15] Mikolov, T. et al. Recurrent neural network based language model. In INTERSPEECH, 1045-1048, 2010.

[16] Mikolov, T. et al. (2011)Extensions of recurrent neural network language model. In ICASSP, 5528-5531.

[17] Mikolov, T. et al. (2013a) Linguistic regularities in continuous space word representations. In NAACL-HLT, 746–751.

[18] Mikolov, T. et al. (2013b) Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS. 3111-3119.

[19] Mikolov, T. et al. (2013c) Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013, Available at: http://arxiv.org/abs/1301.3781.