Multi-task Learning for Semantic Relations Discovery

Georgios Balikas[1], Gaël Dias[2], Massih-Reza Amini[3], and Houssam Akhmouch[2,4]

[1] Kelkoo Group, Grenoble, France
[2] Normandy University, CNRS GREYC, France
[3] University of Grenoble Alps, CNRS LIG, France
[4] Crédit Agricole Brie Picardie, France

Abstract. Identifying the semantic relations that hold between words is of crucial importance for reasoning purposes. Within this context, different methodologies have been proposed that either exclusively focus on a single lexical relation (two-class problem) or learn specific classifiers capable of identifying multiple semantic relations (multi-class problem). In this paper, we propose another way to look at the problem that relies on the multi-task learning paradigm. Preliminary results based on simple learning strategies and state-of-the-art distributional feature representations show that concurrent learning can lead to improvements.

Keywords: Co-Hyponymy · Hypernymy · Multi-task Learning · Neural Networks

1 Introduction

Semantic relations embody a large number of symmetric and asymmetric linguistic phenomena such as co-hyponymy (bike $\leftrightarrow$ scooter) or hypernymy (bike $\rightarrow$ tandem), and their automatic identification is of crucial importance for reasoning purposes. Most approaches focus on modeling a single semantic relation and consist in deciding whether a given relation r holds between a pair of words (x,y) or not (i.e. two-class problem). Another research direction consists in dealing with multiple semantic relations and can be defined as deciding which semantic relation r_i (if any) holds between a pair of words (x, y) (i.e. multi-class problem). In this paper, we propose another way to look at the problem based on the idea that learning semantic relations concurrently may lead to performance improvements when compared to a set of two-class classifiers. Within this context, we propose to study both co-hyponymy and hypernymy based on the findings of [12] that show that learning term embeddings that take into account co-hyponymy similarity improves supervised hypernymy identification. As a consequence, we define a multi-task learning strategy using a hard parameter sharing neural network model that takes as input a learning word pair (x, y) encoded as the concatenation[5] of both word embeddings. The intuition behind our experiment is that if the tasks are correlated, the neural network should improve its generalization ability by taking into account the shared information. Preliminary results over the gold standard dataset ROOT9 [7] show that classification improvements can be obtained.

[5] Best configuration reported in [9] for standard non path-based supervised learning.

2 Methodology and Setups

Multi-task learning architecture. Concurrent learning of tasks that have cognitive similarities is often beneficial. We may hypothesize that recognizing related semantic relations concurrently can benefit classification models across tasks. To test this hypothesis, we propose to use a multi-task learning algorithm that relies on hard parameter sharing. The idea is that the shared parameters can benefit the performance of all tasks learned concurrently if the tasks are related. In particular, we propose an architecture based on a feed-forward neural network to perform the classification step illustrated in Figure 1. The input of the network is the concatenation of the word embeddings of the word pairs followed by a series of non-linear hidden layers. Then, a number of softmax layers gives the network predictions. Here, a softmax layer corresponds to a task, and concurrently learning M tasks requires M separate output softmax layers. The efficiency of hard parameter sharing architectures relies on the fact that the first layers that are shared are tuned by back-propagating the classification errors of every task. That way, the architecture uses the datasets of all tasks, instead of just one at a time.

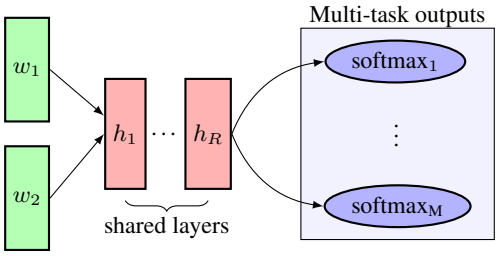

Fig. 1. Multi-task learning architecture.

Learning setup. To concurrently learn co-hyponymy and hypernymy as two classification tasks, we implemented the multi-task architecture shown in Figure 1 using Keras [2] and defined 2 fully-connected hidden layers (i.e. h_1, h_2, $R = 2$) of 50 neurons each as well as 2 softmax layers. The word embeddings are initialized with the 300-dimensional representations of ConceptNet [10]. The activation function of the hidden layers is the sigmoid function and the weights of the layers are initialized with a uniform distribution scaled as described in [3]. As for the learning process, we use the Root Mean Square Propagation (RMSprop) optimization method with learning rate set to 0.001 and the default value for $\rho = 0.9$. For every task, we use the binary cross-entropy loss function and the network is trained with batches of 32 examples[6].

Dataset and lexical split. In order to perform our experiments, we use the ROOT9 dataset [7] that contains 9,600 word pairs. The word pairs are equally distributed among three classes (hypernymy, co-hyponymy and random) and involve adjectives, nouns and verbs. Here, we exclusively focus on nouns and keep all hypernyms, co-hyponyms and random pairs that can be represented by ConceptNet embeddings. Following a classical learning procedure, the dataset must be split into train, validation and test subtests. The

[6] Code is available at `https://github.com/balikasg/multitask-learning`.

standard procedure is random splitting. However, [4] point out that using distributional representations in the context of supervised learning tends to perform lexical memorization. In this case, the model mostly learns independent properties of single terms in pairs. To overcome this situation and prevent the model from overfitting, [4] suggest to split the train and test sets such that each one contains a distinct vocabulary. This procedure is called lexical split. Here, we propose to apply lexical split as defined in [4]. So, lexical repetition exists in the train and validation subsets, but the test set is exclusive in terms of vocabulary. Note that all subsets are available for replicability[7].

3 Results

For comparative evaluation, we implement two baseline systems: (1) Majority Baseline and (2) Logistic Regression that has shown positive results in [8] over parts of the ROOT9 dataset. As for evaluation metrics, we report two measures: (1) Accuracy and (2) Macro-average F_1 measure (MaF_1). Accuracy captures the number of correct predictions over the total predictions, while MaF_1 evaluates how the model performs across the different relations as it averages the F_1 measures of each relation without weighting the number of examples in each case. Preliminary results of our architecture are illustrated in Table 1.

	Co-hyponym vs. Random		Hypernym vs. Random		Average Results	
Algorithm	Accuracy	MaF_1	Accuracy	MaF_1	Accuracy	MaF_1
Majority Baseline	0.761	0.432	0.698	0.411	0.730	0.422
Logistic Regression	**0.900**	0.841	0.818	0.748	0.859	0.795
Multitask learning	0.895	**0.849**	**0.841**	**0.803**	**0.868**	**0.826**

Table 1. ROOT9. ConceptNet embeddings. Accuracy and Macro F_1 scores.

The multi-task paradigm shows that improved accuracy and MaF_1 scores can be achieved on average reaching respectively values of 86.8% and 82.6%, thus showing improvements of 0.9% and 3.1% over the best baseline (i.e. logistic regression). In this case, the best improvements are obtained for the classification of hypernym pairs with benefits of 2.3% in terms of accuracy and 5.5% in terms of MaF_1, indeed suggesting that there exists a learning link between hypernymy and co-hyponymy. In parallel, the results for co-hyponymy classification are equivalent to a classical supervised strategy using logistic regression. So, the results seem to put in advance the fact that we can expect an improvement for hypernymy classification but not for co-hyponymy in a multi-task environment, suggesting a positive influence of co-hyponymy learning towards hypernymy but not the opposite. Note that these results seem to confirm the findings of [12] for another learning scenario.

4 Conclusions and Future Directions

In this paper, we proposed to study the concurrent learning of co-hyponymy and hypernymy using a hard parameter sharing multi-task architecture and state-of-the-art distributional input representations (concatenation of ConceptNet embeddings). Obtained

[7] Data are available at `https://github.com/balikasg/multitask-learning`.

results show that concurrent learning can lead to improvements, thus justifying our initial hypothesis. In particular, we have shown that hypernymy classification can gain from concurrent learning of co-hyponymy. Based on these preliminary findings, a vast amount of improvements can now be introduced into the framework to increase overall performance. First, we aim at studying the interaction between more semantic relations such as synonymy and meronymy. Then, with respect to the input features, we intend to study the potential benefits from dedicated embeddings such as hypervec [6] and dual embeddings [5]. Moreover, we deeply believe that the LSTM path-based features introduced in [9] and some well-defined word pairs similarity measures [8] can lead to classification improvements by complementing the information present in distributional semantic spaces. Moreover, it is clear that more complex architectures such as convolutional neural networks may improve the learning process as it is proposed in [1] for similar tasks. Finally, we plan to make the original task more difficult by including the detection of the direction of the asymmetric relations and adding noisy pairs as in [11].

References

1. Attia, M., Maharjan, S., Samih, Y., Kallmeyer, L., Solorio, T.: Cogalex-v shared task: Ghhh - detecting semantic relations via word embeddings. In: Workshop on Cognitive Aspects of the Lexicon. pp. 86–91 (2016)
2. Chollet, F.: Keras. `https://keras.io` (2015)
3. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: 13th International Conference on Artificial Intelligence and Statistics. pp. 249–256 (2010)
4. Levy, O., Remus, S., Biemann, C., Dagan, I.: Do supervised distributional methods really learn lexical inference relations? In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 970–976 (2015)
5. Nalisnick, E., Mitra, B., Craswell, N., Caruana, R.: Improving document ranking with dual word embeddings. In: 25th International Conference on World Wide Web. pp. 83–84 (2016)
6. Nguyen, K.A., Köper, M., Schulte im Walde, S., Vu, N.T.: Hierarchical embeddings for hypernymy detection and directionality. In: Conference on Empirical Methods in Natural Language Processing. pp. 233–243 (2017)
7. Santus, E., Lenci, A., Chiu, T., Lu, Q., Huang, C.: Nine features in a random forest to learn taxonomical semantic relations. In: 10th International Conference on Language Resources and Evaluation. pp. 4557–4564 (2016)
8. Santus, E., Shwartz, V., Schlechtweg, D.: Hypernyms under siege: Linguistically-motivated artillery for hypernymy detection. In: 15th Conference of the European Chapter of the Association for Computational Linguistics. pp. 65–75 (2017)
9. Shwartz, V., Goldberg, Y., Dagan, I.: Improving hypernymy detection with an integrated path-based and distributional method. In: 54th Annual Meeting of the Association for Computational Linguistics. pp. 2389–2398 (2016)
10. Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: An open multilingual graph of general knowledge. In: 31st Conference on Artificial Intelligence. pp. 4444–4451 (2017)
11. Vylomova, E., Rimell, L., Cohn, T., Baldwin, T.: Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. In: 54th Annual Meeting of the Association for Computational Linguistics. pp. 1671–1682 (2016)
12. Yu, Z., Wang, H., Lin, X., Wang, M.: Learning term embeddings for hypernymy identification. In: 24th International Joint Conference on Artificial Intelligence. pp. 1390–1397 (2015)