Semantic Compositionality through Recursive Matrix-Vector Spaces

Poster presented by: Sonu Agarwal, Viveka Kulharia

Mentor: Prof. Amitabha Mukerjee

1. INTRODUCTION

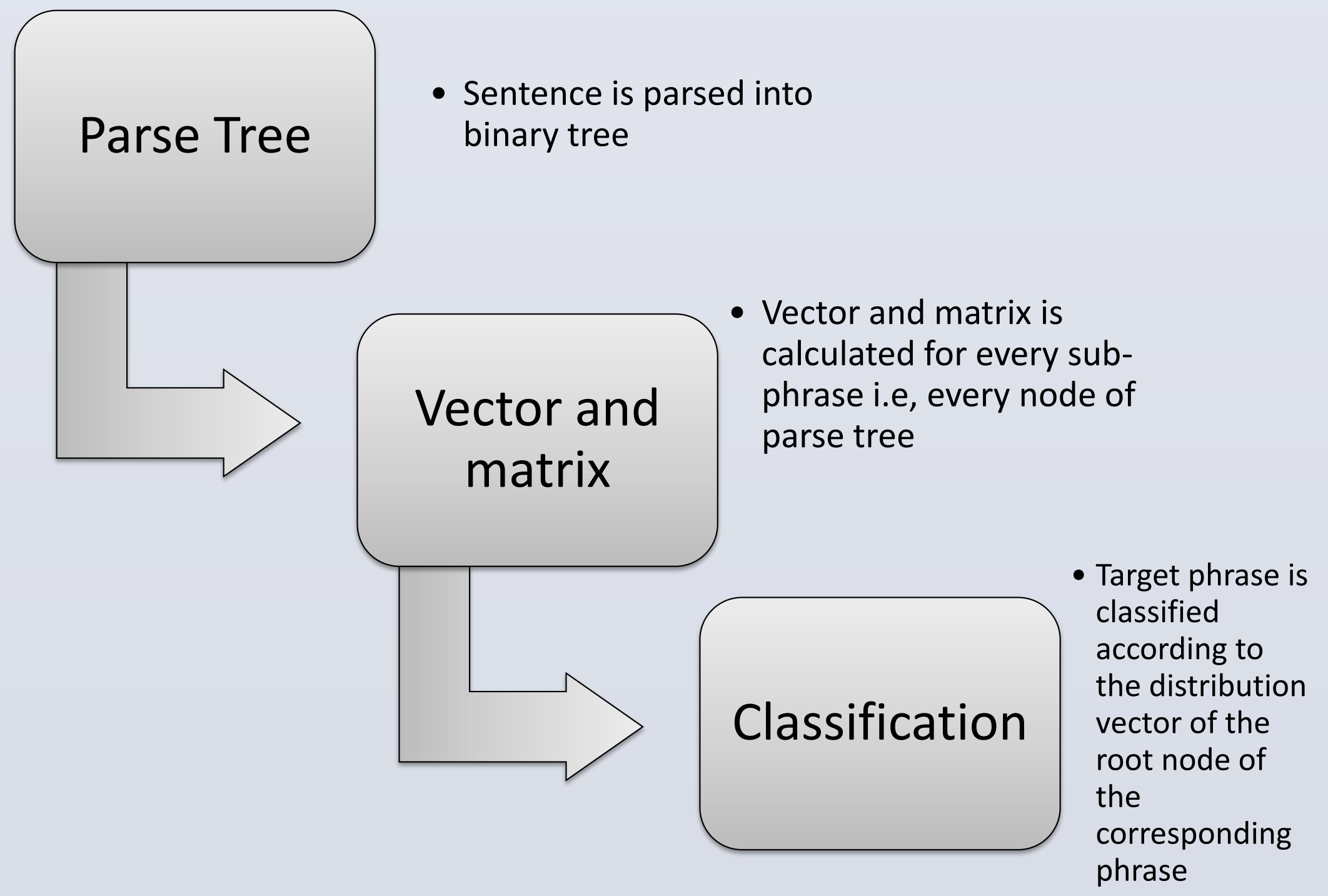
Single-word vector-based model has been very successful at learning the lexical information but are not able to capture the compositional meaning of longer phrases or sentences.

We introduce a recursive neural network model (RNN) that learns compositional vector representations of phrases or sentences of arbitrary length or syntactic type. We assign a vector and a matrix to each node in the parse tree. Vector contains the inherent meaning of the word and matrix captures how it changes the meaning of its neighboring words or phrases. A representation for a longer phrase is computed in a bottom-up manner by recursively combining children words according to the syntactic structure in the parse tree.

Other previous methods like (Mitchell and Lapata, 2010) just add matrix product of vectors and do not use parsed tree. Further research by author Richard Socher (Socher et al., 2013) uses Recursive Neural Tensor Network.

Here we focus on classifying semantic relationship between the pair of nominals in a sentence into 19 different categories. Some of these categories are cause-effect, product-producer and topic-message along with the direction.

2. METHODOLOGY



Parse Tree

- Sentence is parsed into binary tree

Vector and matrix

- Vector and matrix is calculated for every sub-phrase i.e, every node of parse tree

Classification

- Target phrase is classified according to the distribution vector of the root node of the corresponding phrase

2.1 Binarized Parse Tree

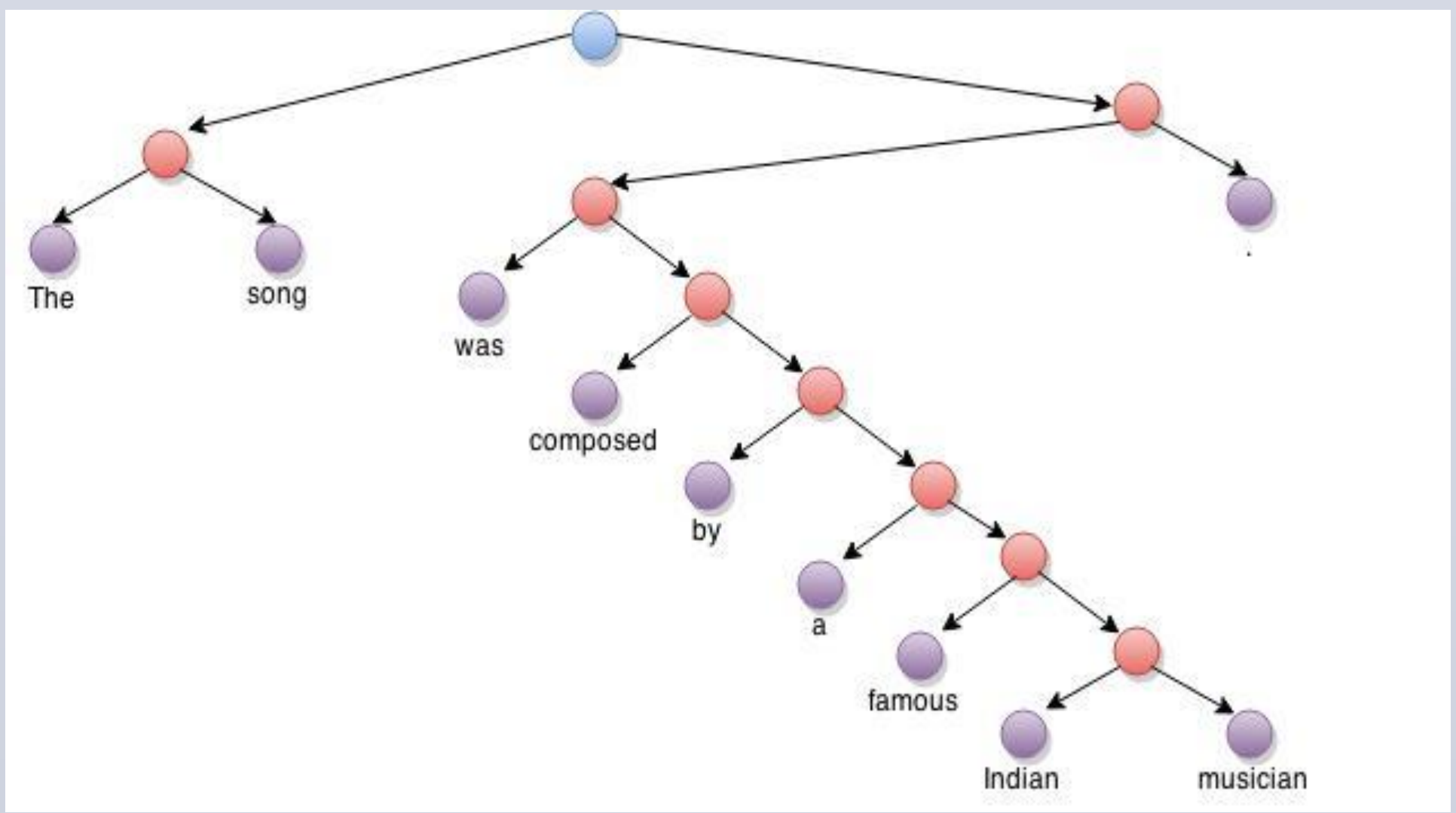


Fig 1: Binarized parse tree (Constructed using draw.io)

Recursive Matrix-Vector Model

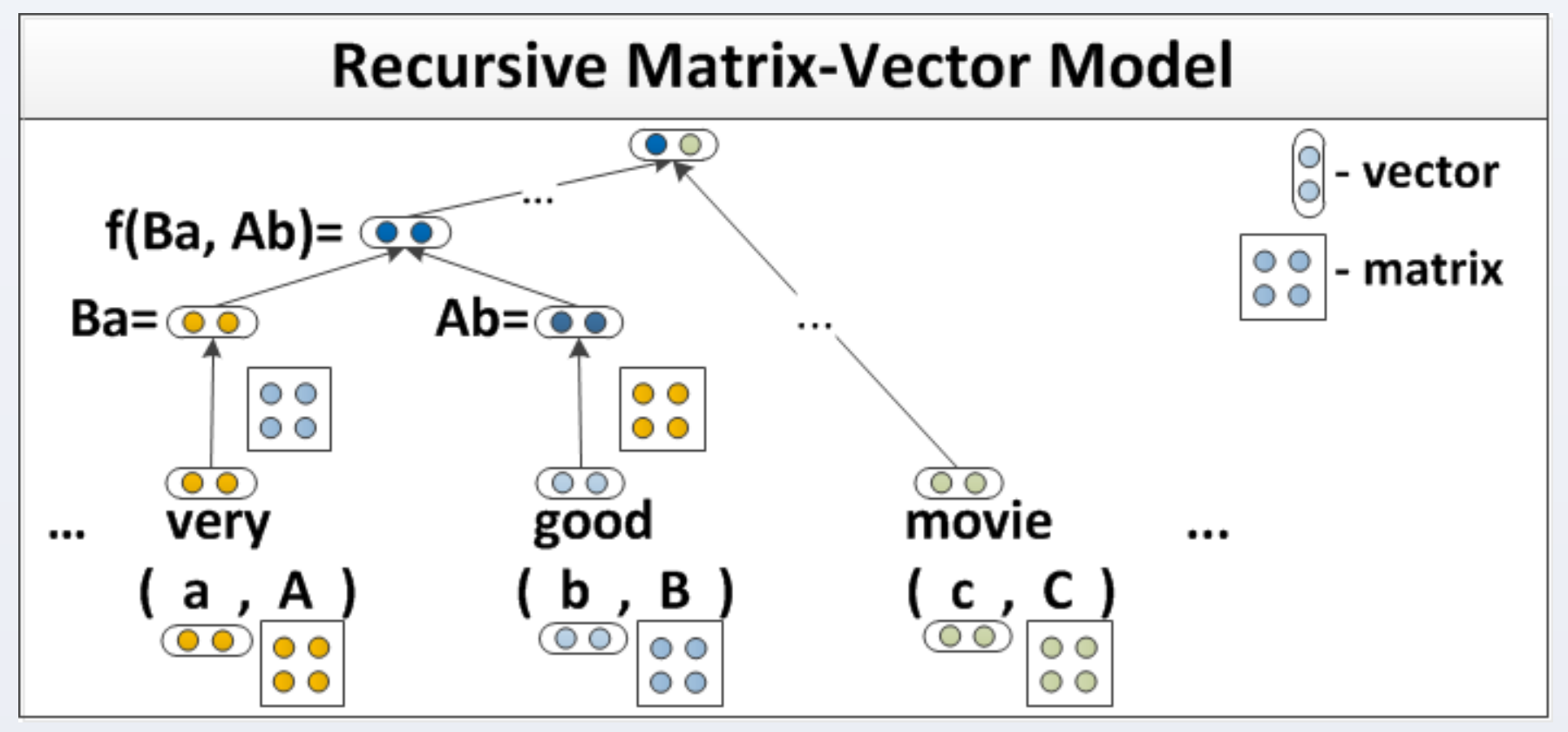


Fig 2: RNN which learns semantic vector representation of phrases (src: Ref[1])

Initialization :

- Initialize all the word vectors with pre-trained 50-dimensional word-vectors
- Initialize matrices as $X = I + \varepsilon$, where I is the identity matrix and ε is Gaussian noise

Composition:

$$p = f_{A,B}(a,b) = f(Ba, Ab) = g\left(W\begin{bmatrix} Ba \\ Ab \end{bmatrix}\right)$$

$$P = f_M(A,B) = W_M\begin{bmatrix} A \\ B \end{bmatrix}$$

Training

We train vector representations by adding on top of each parent node a softmax classifier to predict a class distribution over sentiment or relationship classes.

$$d(p) = soft\max\left(W^{label}p\right)$$

where $W^{label} \in R^{K\times n}$ is a weight matrix. If there are K labels, then $d \in R^K$ is a K-dimensional multinomial distribution.

We denote $t(x) \in R^{K\times 1}$ as the target distribution vector at node x. t(x) has a 0-1 encoding: the entry at the t(x) is 1, and the remaining entries are 0. We then compute the cross entropy error between d(x) and t(x):

$$E(x) = -\sum_{k=1}^{K} t_k(x)\log d_k(x)$$

and define an objective function as the sum of E(x) over all training data:

$$J(\theta) = \frac{1}{N}\sum_x E(x) + \frac{\lambda}{2}[\![\theta]\!]^2$$

where $\theta = (W, W_M, W^{label}, L, L_M)$ is the set of our model parameters that should be learned. λ is a vector of regularization parameters.

L and L_M are set of word vectors and word matrices respectively.

2.2 Classification of Semantic Relationship

- We first find the path in the parse tree between the two words whose relation we want to classify.
- We then select the highest node of the path and classify the relationship using that node's vector as features.
- We apply MV-RNN model to the subtree spanned by the two words.

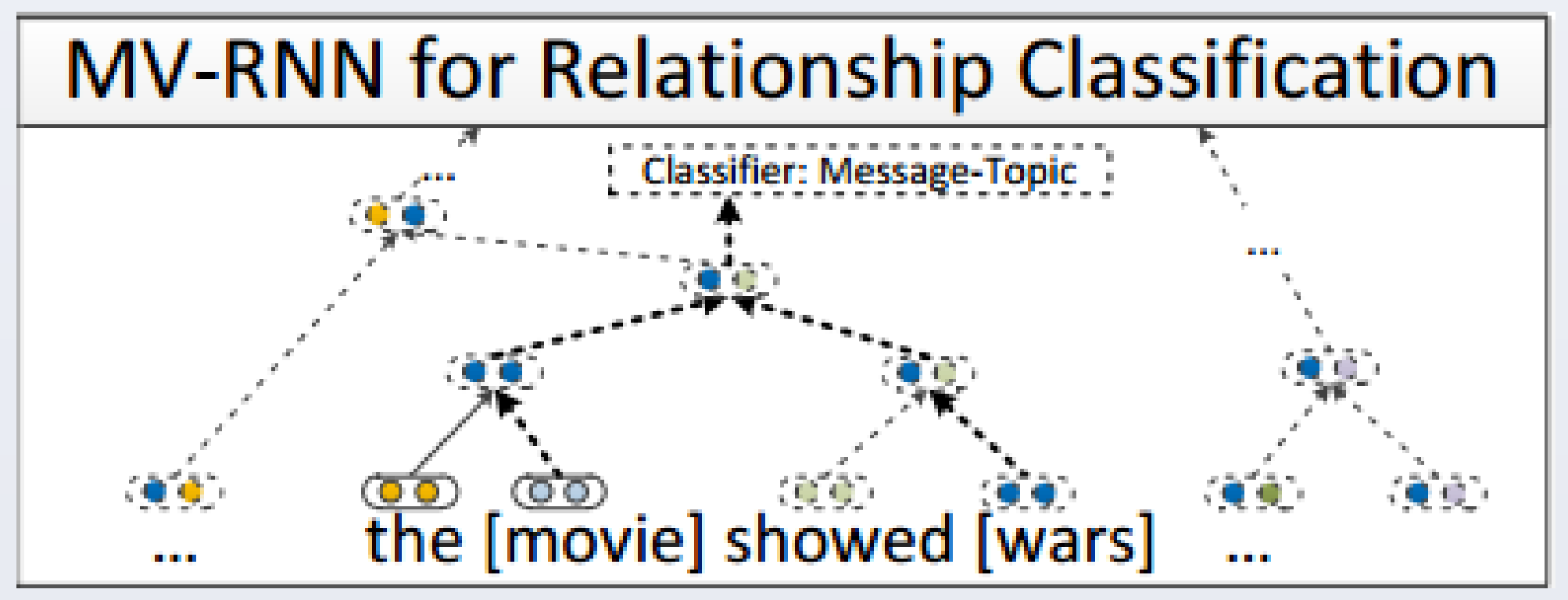

MV-RNN for Relationship Classification
Classifier: Message-Topic
...
...
the [movie] showed [wars]
...
...

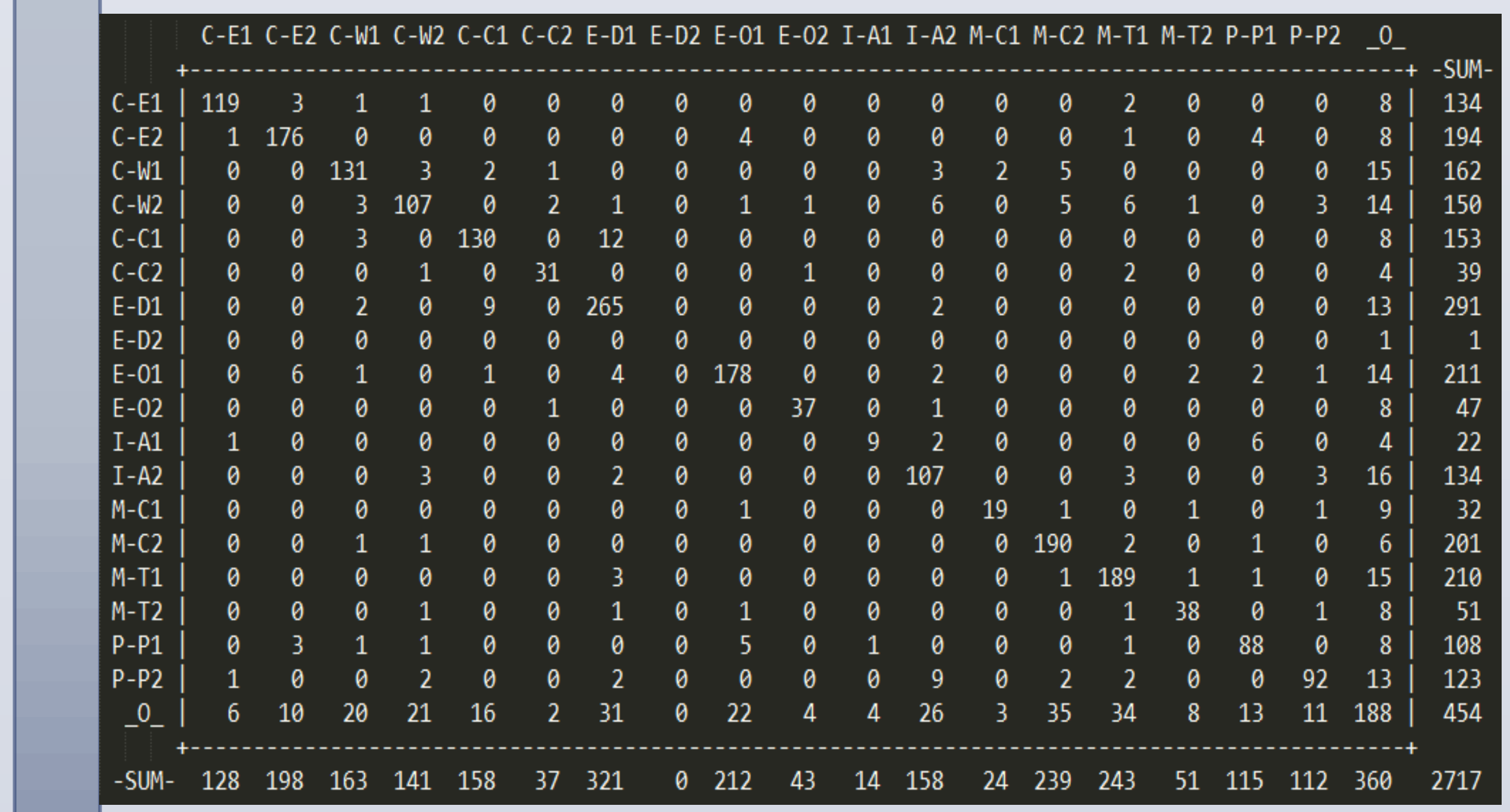
Fig 3: MV-RNN learns vectors in the path connecting two words (src: Ref[1])

Results

We performed experimentation on the following dataset:

- SemEval 2010 Task 8

There are 9 ordered relationships (with two directions) and an undirected other class, resulting in 19 classes. Among the relationships are: message-topic, cause-effect, instrument-agency. A pair is counted as correct if the order of the words in the relationship is correct.



```
        C-E1 C-E2 C-W1 C-W2 C-C1 C-C2 E-D1 E-D2 E-O1 E-O2 I-A1 I-A2 M-C1 M-C2 M-T1 M-T2 P-P1 P-P2  _O_
      +-----------------------------------------------------------------------------------------------+ -SUM-
 C-E1 | 119    3    1    1    0    0    0    0    0    0    0    0    0    0    2    0    0    0    8 |  134
 C-E2 |   1  176    0    0    0    0    0    0    4    0    0    0    0    0    1    0    4    0    8 |  194
 C-W1 |   0    0  131    3    2    1    0    0    0    0    0    3    2    5    0    0    0    0   15 |  162
 C-W2 |   0    0    3  107    0    2    1    0    1    1    0    6    0    5    6    1    0    3   14 |  150
 C-C1 |   0    0    3    0  130    0   12    0    0    0    0    0    0    0    0    0    0    0    8 |  153
 C-C2 |   0    0    0    1    0   31    0    0    0    1    0    0    0    0    2    0    0    0    4 |   39
 E-D1 |   0    0    2    0    9    0  265    0    0    0    0    2    0    0    0    0    0    0   13 |  291
 E-D2 |   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1 |    1
 E-O1 |   0    6    1    0    1    0    4    0  178    0    0    2    0    0    0    2    2    1   14 |  211
 E-O2 |   0    0    0    0    0    1    0    0    0   37    0    1    0    0    0    0    0    0    8 |   47
 I-A1 |   1    0    0    0    0    0    0    0    0    0    9    2    0    0    0    0    6    0    4 |   22
 I-A2 |   0    0    0    3    0    0    2    0    0    0    0  107    0    0    3    0    0    3   16 |  134
 M-C1 |   0    0    0    0    0    0    0    0    1    0    0    0   19    1    0    1    0    1    9 |   32
 M-C2 |   0    0    1    1    0    0    0    0    0    0    0    0    0  190    2    0    1    0    6 |  201
 M-T1 |   0    0    0    0    0    0    3    0    0    0    0    0    0    1  189    1    1    0   15 |  210
 M-T2 |   0    0    0    1    0    0    1    0    1    0    0    0    0    0    1   38    0    1    8 |   51
 P-P1 |   0    3    1    1    0    0    0    0    5    0    1    0    0    0    1    0   88    0    8 |  108
 P-P2 |   1    0    0    2    0    0    2    0    0    0    0    9    0    2    2    0    0   92   13 |  123
  _O_ |   6   10   20   21   16    2   31    0   22    4    4   26    3   35   34    8   13   11  188 |  454
      +-----------------------------------------------------------------------------------------------+
 -SUM-  128  198  163  141  158   37  321    0  212   43   14  158   24  239  243   51  115  112  360   2717
```

Accuracy (calculated for the above confusion matrix) = 2094/2717 = 77.07%

F1 Score = 82.51%

We also used a different dataset modified according to our code requirement from "SemEval 2007 Task 4" to perform test and used the previous trained model.

F1 score for this experimentation was obtained to be 40.08 % ignoring directionality.

Comparison with other methods

Classifier	Feature Set	F1
SVM	POS, stemming, syntactic patterns	60.1
SVM	POS, WordNet, stemming, syntactic patterns	74.8
SVM	POS, WordNet, morphological features, thesauri, Google n-grams	77.6
RNN	-	74.8
Lin.MVR	-	73
MV-RNN	-	79.1
RNN	POS,WordNet,NER	77.6
Lin.MVR	POS,WordNet,NER	78.7
MV-RNN	POS,WordNet,NER	**82.5**

Table 1:Result comparison with other methods (src: other results from ref[1])

Improvement in the result is also due to some common drawbacks in other methods. For example:

- Many methods represent text in terms of unordered list of words while sentiments depend not just on the word meanings but also how they are ordered.
- The features used are manually developed which won't necessarily capture all the features of the word.

Conclusion

- Our model builds on a syntactically plausible parse tree and can handle compositional phenomena.
- The main novelty of our model is the combination of matrix-vector representations with a recursive neural network.
- It can learn both the meaning vectors of a word and how that word modifies its neighbors (via its matrix).
- The MV-RNN combines attractive theoretical properties with good performance on large, noisy datasets.

References

- Richard Socher, Brody Huval, Christopher D. Manning and Andrew Y. Ng. "*Semantic Compositionality through Recursive Matrix-Vector Space*". Conference on Empirical Methods in Natural Language Processing (EMNLP 2012, Oral)
- J. Mitchell and M. Lapata." *Composition in distributional models of semantics*" Cognitive Science,34(2010):1388–1429
- Kazuma Hashimoto, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. " *Simple customization of recursive neural networks for semantic relation classification*". 2013 In EMNLP.