Semantic Compositionality through Recursive Matrix-Vector Spaces

Sonu Agarwal* and Viveka Kulharia*

E-mail: sonuagr@iitk.ac.in; vivkul@iitk.ac.in

Abstract

Word-vectors have been existing for a long time now, which characterizes corresponding word uniquely such that we can relate one word to other using some operations on respective vectors. Even though it has been very successful in its attempt, it couldn't be significantly expanded to other areas of NLP like getting the semantics of a sentence using the vectors related to the words composing it. Recently, it has become one of the hottest fields of machine learning as compositional functions have been learnt which perform quite well. We would explore some of the methods to learn such functions.

Introduction

In this work, we would get into an emergent field of Recursive Neural Networks (RNN) to get the semantics of sentences using their binarized parse trees. Besides learning vectors, word-matrices will also be learnt along with other parameters which would capture the power of the words to modify its neighbors.[1]

There aren't many datasets where semantic labeling of only single sentences has been done

*To whom correspondence should be addressed still it would be tested on two SemEval datasets where the task is to find the relation between two selected nominals of a sentence.

Here we focus on classifying semantic relationship between a preselected pair of nominals present in a sentence into 19 different categories[2] in case of one dataset and 13 categories in case of other. Some of these categories are cause-effect, product-producer and topic-message along with the direction. The learning is done using two methods: without any external features and with external features namely POS, WordNet and NER.

Table 1: All nine classes with examples (src: SemEval 2010 Task 8)

Relationship	Sentence with labeled nouns for which to predict relationships
Cause-Effect(e2,e1)	"He had chest pains and headaches$_{e1}$ from mold$_{e2}$ in the bedrooms."
Entity-Origin(e1,e2)	"Lawyers in Detroit also worked overtime as several lawsuits$_{e1}$ ensued from angry and injured fans$_{e2}$."
Message-Topic(e1,e2)	"The Pulitzer Committee issues an official citation$_{e1}$ explaining the reasons$_{e2}$ for the award."
Product-Producer(e2,e1)	"The court$_{e1}$ decided the objection by making the instalment order$_{e2}$ as sought."
Entity-Destination(e1,e2)	"The solute was placed inside a beaker and 5 mL of the solvent$_{e1}$ was pipetted into a 25 mL glass flask$_{e2}$ for each trial."
Member-Collection(e1,e2)	"The fifty essays$_{e1}$ collected in this volume$_{e2}$ testify to most of the prominent themes from Professor Quispel's scholarly career."
Instrument-Agency(e2,e1)	"The author$_{e1}$ of a keygen uses a disassembler$_{e2}$ to look at the raw assembly code."
Component-Whole(e1,e2)	"The system as described above has its greatest application in an arrayed configuration$_{e1}$ of antenna elements$_{e2}$."
Content-Container(e1,e2)	"The bomb$_{e1}$ was in a suitcase$_{e2}$ loaded in Frankfurt and transferred to the doomed Boeing 747 in London."

Related Works

Composition of word-Vectors to get the semantic meaning of the containing phrase got boost due to Lapata et al. (2010). They attempted to get the semantics of small phrases by trying various compositional functions over the word vectors. The major breakthrough of this paper was that it worked well for smaller phrases and the process didn't require manual assigning of features to vectors. They were aware of the limitations of one function to be used for composition of all kinds of phrases as the way humans compose isn't that perfect. They also admitted that by choosing vectors and that too of same dimension, complexity of sentences was reduced and was restricted but it gave some computational efficiency. Same was the reason stated for not using matrices for representation of adjectives as modifiers even though Clark et al. (2008) had already stated it to be a good representation scheme. They didn't maintain similar knowledge of syntax across all functions as they chose different composition functions, but they tried to have the order of words taken into account. The functions they chose were like[3]:

$$p = Au + Bv \tag{1}$$

The equation 1 has the matrix A and B signifying the contribution of u and v to phrase p. Now it can be seen that though it works well, as was tested, with smaller phrases, it can't be expanded well to larger sentences.

The paper that we used in this project (Socher et al., 2012 : MVRNN) solved the problem of choosing a function by learning a nonlinear function over a binarized parse tree as a RNN. Matrices was also learned for each word which instead of signifying the contribution of a word to a phrase signified how it affected its neighbors. And as RNN was used, sentence of arbitrary length could be covered.

Further research by author Richard Socher (Socher et al., 2013) brings two major changes on the table. One is the use of Stanford Sentiment Treebank, which is a corpus of movie reviews' snippets with each node of their parse tree manually annotated. It drastically improves the accuracy of any model trained on it as the model can learn its parameters better as it exactly knows the sentiments of the phrases composing a sentence at every node up to the root. Earlier the model had to tune its parameters by just looking at the root node, which won't be that effective. The other change was the reduction of the number of parameters to be learnt by using just one function powerful enough to synthesis a phrase vector using the constituent vectors. Tensor was used for this purpose and thus the model has constant number of parameters to be learnt.

Methodology

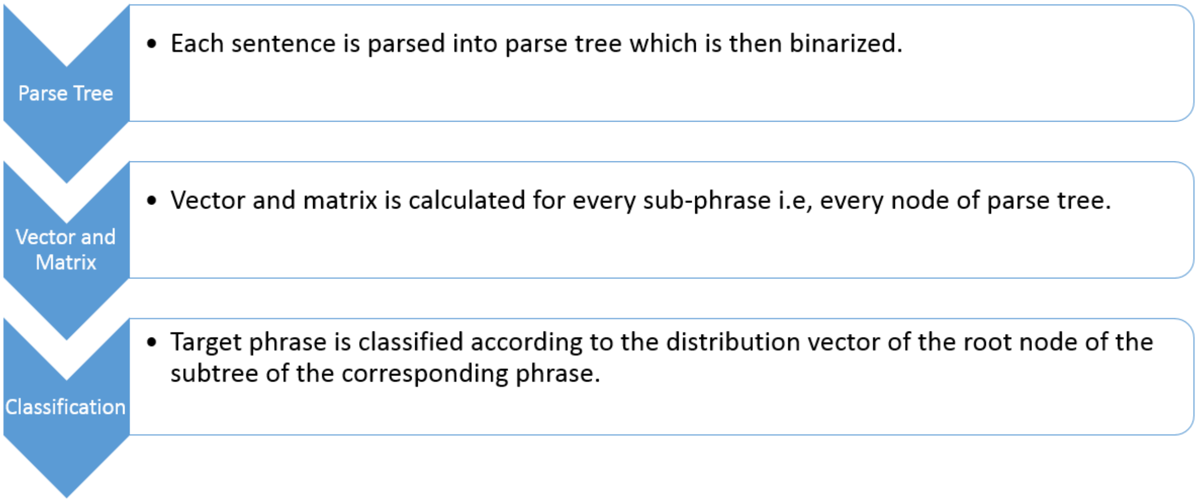

Parse Tree
• Each sentence is parsed into parse tree which is then binarized.
Vector and Matrix
• Vector and matrix is calculated for every sub-phrase i.e, every node of parse tree.
Classification
• Target phrase is classified according to the distribution vector of the root node of the subtree of the corresponding phrase.

Figure 1: Basic flowchart of our method

- In a general parse tree, a node can have more than two children. But in this method we can combine only two nodes at a time as the function requires two argument which is also easier to handle. So we binarize the general parse tree so that we are able to combine children vectors to obtain vectors for longer phrases. Refer figure 2 to see an example of binarized parse tree.

- Value of vector and matrix of each word depends on its own meaning and its capability to modify other words respectively. For example, in the phrase “too good” word “too” doesn’t have much of its own meaning but modifies good in a significant way. So, its vector will be very close to zero and its matrix will assume significant value.

- The dimension of the target vector will be same as the number of classes. During the training, we provide the system with the target vector corresponding to each phrase to be classified and same is calculated during testing to classify phrases.

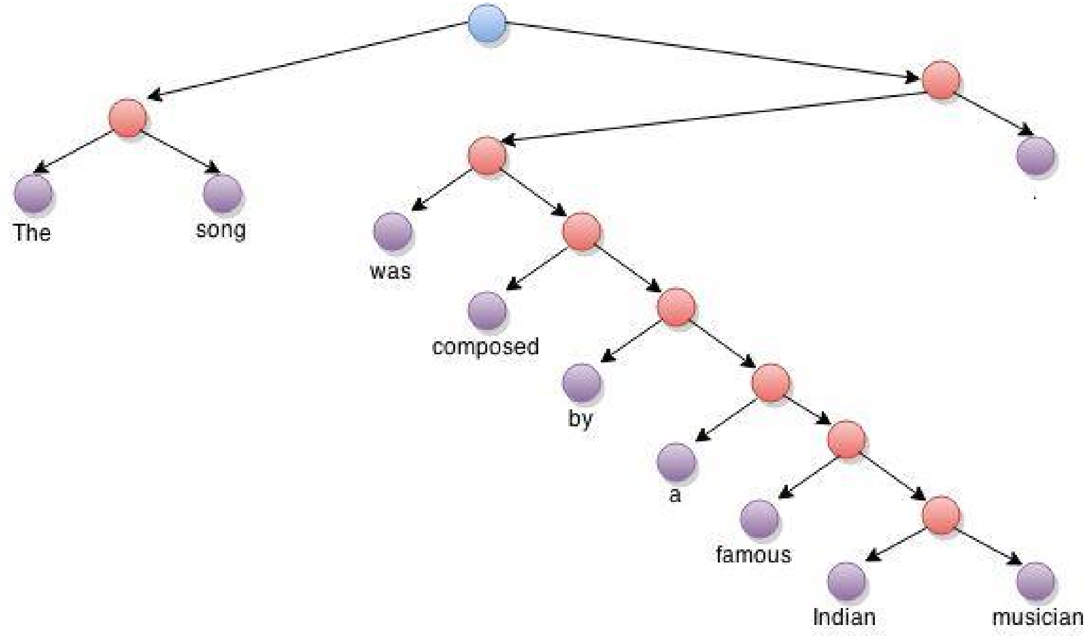


Figure 2: Binarized Parse Tree (Constructed using draw.io)

Recursive Matrix-Vector Model

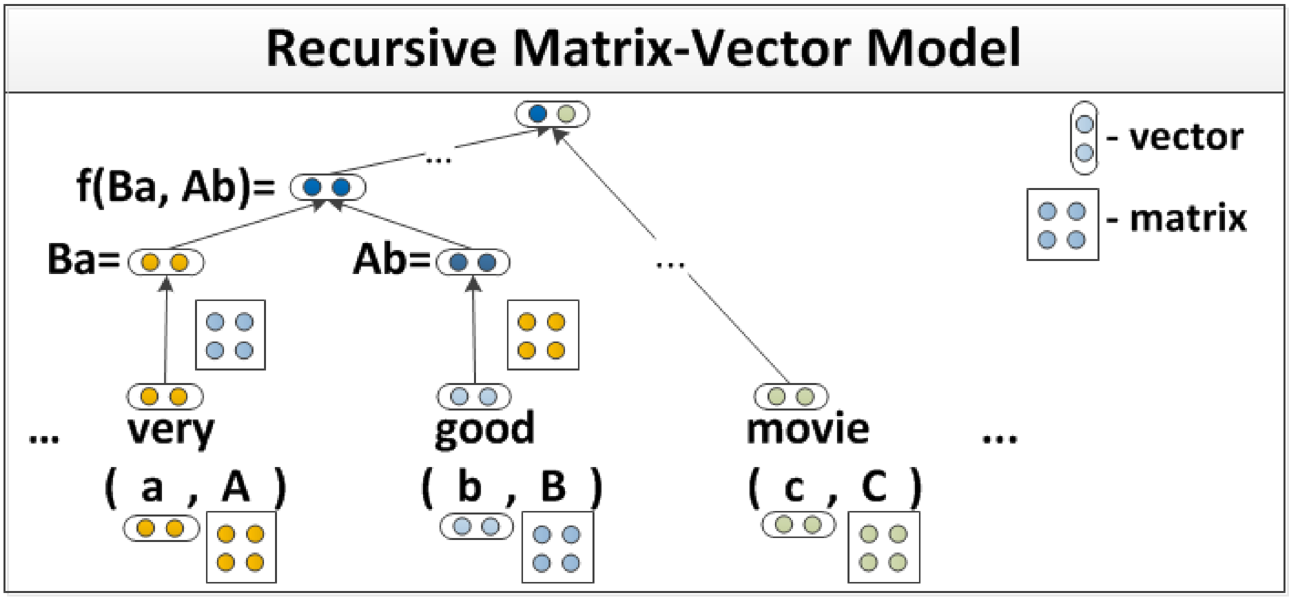


Figure 3: RNN which learns semantic vector representation of phrases (src: ref^1)

Initialization

The vectors and matrices are initiallized in following way[1]:

- Initialize all the word vectors with 50-dimensional word-vectors pre-trained on Wikipedia text by Collobert and Weston (2008)
- Initialize matrices as $X = I + \epsilon$, where I is the 50x50 identity matrix and ϵ is Gaussian noise.

Composition

To combine the two child-subphrases, the parent vector p and parent matrix P are obtained using nonlinear function $g()$ and $R^{n\times 2n}$ matrices W and W_M are learnt by the model.[1]

$$p = f_{A,B}(a,b) = f(Ba, Ab) = g(W \begin{bmatrix} Ba \\ Ab \end{bmatrix}) \tag{2}$$

$$P = f_M(A,B) = W_M \begin{bmatrix} A \\ B \end{bmatrix} \tag{3}$$

Training

We train vector representations by adding to every node a softmax classifier to predict relationship classes or class distribution over sentiment.[1]

$$d(p) = \text{softmax}(W^{\text{label}} p) \tag{4}$$

where $W^{\text{label}} \in R^{K\times N}$ is a weight matrix to transform the N dimensional vector p to K dimensional vector where K is the number of labels.

$t(x) \in R^{K\times 1}$ is the required vector at node x after applying softmax classifier. $t(x)$ is a 0-1 vector such that the element denoting the correct label as 1 and rest all are 0. The cross entropy error between $d(x)$ and $t(x)$ is then computed as[4]:

$$E(x) = -\sum_{k=1}^{K} t_k(x)\log(d_k(x)) \tag{5}$$

Now, the objective functon is defined as the sum of $E(x)$ over all training data[4]:

$$J(\theta) = \frac{1}{N}\sum_{x} E(x) + \frac{\lambda}{2}||x||^2 \tag{6}$$

Where λ is a regularization parameter and $(W, W_M, W^{\text{label}}, L, L_M)$ is the set of model parameters that should be learned. L and L_M are set of word vectors and word matrices respectively.[1]

Classification

The task is to get relation between a pair of nominals. So, only the lexicons between the nominals in the sentence syntax are to be considered:

- We first find the smallest subtree spanning the nominals whose relation we want to classify.
- We then select the root node of the subtree and classify the relationship using its vector.
- The vector of the root node is found by applying MV-RNN recursively from bottom to top along the subtree.

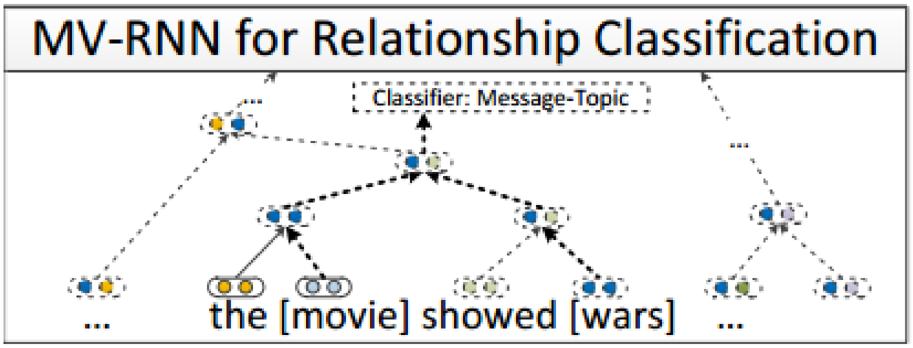

MV-RNN for Relationship Classification
Classifier: Message-Topic
...
the [movie] showed [wars]
...

Figure 4: MV -RNN learns vectors in the path connecting two words (src: ref[1])

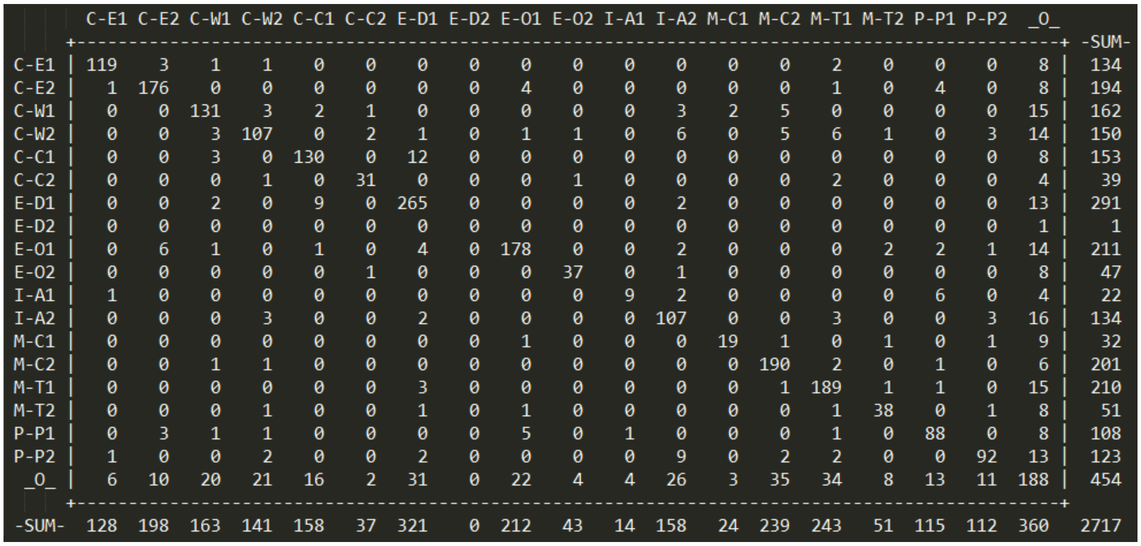
Results

Dataset 1

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
```

Figure 5: Classification corresponding to 19 classes using external features

For classification using external features,

F1 score ignoring directionality = 83.16

F1 score considering directionality = 82.51

Accuracy ignoring directionality = 77.62 %

Accuracy considering directionality = 77.07 %

And For classification without using any external features,

F1 score ignoring directionality = 80.99

F1 score considering directionality = 79.73

Accuracy ignoring directionality = 75.49 %

Accuracy considering directionality = 74.42 %

Few individual examples

The table 2 consists of some sentences with their correct labelling and labels provided by the model with and without external features. Here are the possible explanations of these individual results:

- Sentence 1 is correctly classified with or without any external features.
- Sentence 2 is wrongly classified without using any external features but with features, its correctly classified. It can be observed that there is very small difference between Cause-effect and Origin-Entity, so its not that big error.
- Sentence 3 is wrongly classified on using external features but without external features, its classified correctly. The possible reason is that using external features, most of the movie names may have been replaced with "drama" due to NER and most of them may have been of the relationship "Message-Topic", so due to overfitting, it strongly learnt this and gave wrong classification.

- Sentence 4 is wrongly classified whether external features are used or not. It may be noted that Entity-Origin is not too different from Product-Producer.

Table 2: Few individual examples (src: SemEval 2010 Task 8)

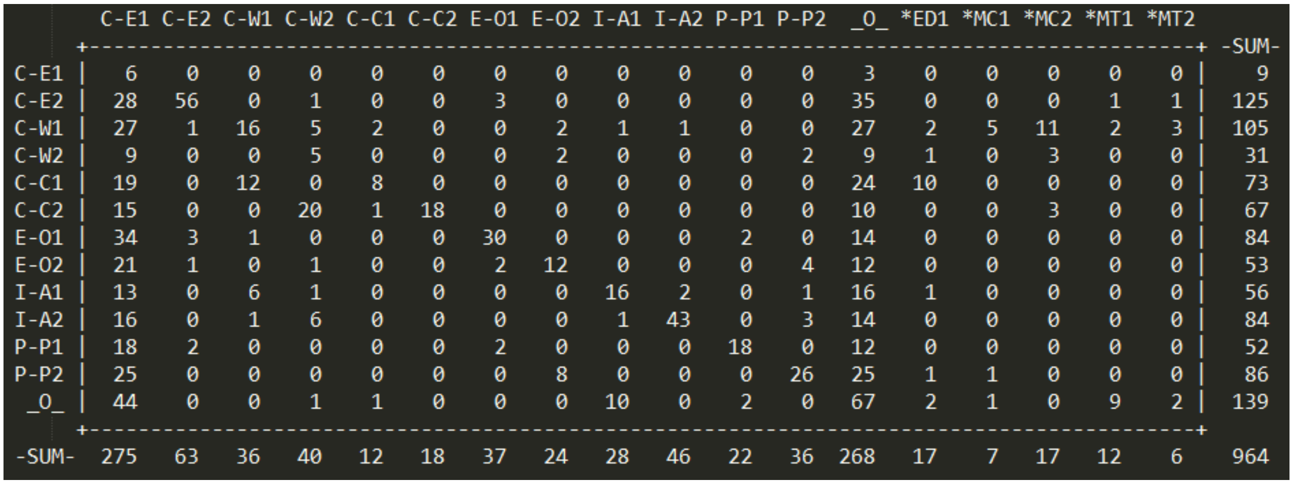
Sentence	Correct Relation	Without External Features	With External Features
“Many of his literary pieces$_{e1}$ narrate and mention stories$_{e2}$ that took place in Lipa.”	Message-Topic(e1,e2)	Message-Topic(e1,e2)	Message-Topic(e1,e2)
“Avian influenza$_{e1}$ is an infectious disease of birds caused by type A strains of the influenza virus$_{e2}$.”	Cause-Effect(e2,e1)	Entity-Origin(e1,e2)	Cause-Effect(e2,e1)
“He has built a formidable reputation writing powerful dramas$_{e1}$ for the stage and screen, based on *real events*$_{e2}$ or socially vital issues.”	Other	Other	Message-Topic(e1,e2)
“Essentially, the blisters$_{e1}$ that appear in the mouth are caused by the herpes simplex virus$_{e2}$ type 1, HSV-1 for short.”	Product-Producer(e1,e2)	Entity-Origin(e1,e2)	Other

Dataset 2

We also used a different dataset from ”SemEval 2007 Task 4” to perform test and checked the domain adaptivity of the presented technique and corresponding code. It initially contained 14 relationship classes[5]. “Theme-Tool” was also present as a class which wasn’t learnt by our previous model, so we changed it to “Other”, hence we got 13 classes. The class “Part-Whole” was changed to “Component-Whole”, while the classes “Origin-Entity(e1,e2)” and “Origin-Entity(e2,e1)” was change to “Entity-Origin(e2,e1)” and “Entity-Origin(e1,e2)” respectively. The modified dataset has data for only 6 of the 9 classes which are: Cause-Effect, Component-Whole, Content-Container, Entity-Origin, Instrument-Agency and Product-Producer. The dataset was trimmed to match the semEval 2010 task 8 as it was our learning model. There were also some erroneous sentences in the dataset which couldn't be parsed and some others which lacked the labelling of their correspopnding relationship. They needed to be removed from the dataset. The number of such sentenced were around 8 in training set and around 5 in test set uniformly spread in the set.

We ran test on the new test data using the same training model as was used in the previous case.

Results obtained for the new dataset are as follows:



```
        C-E1 C-E2 C-W1 C-W2 C-C1 C-C2 E-O1 E-O2 I-A1 I-A2 P-P1 P-P2  _O_ *ED1 *MC1 *MC2 *MT1 *MT2
      +--------------------------------------------------------------------------------------------+ -SUM-
 C-E1 |    6    0    0    0    0    0    0    0    0    0    0    0    3    0    0    0    0    0 |    9
 C-E2 |   28   56    0    1    0    0    3    0    0    0    0    0   35    0    0    0    1    1 |  125
 C-W1 |   27    1   16    5    2    0    0    2    1    1    0    0   27    2    5   11    2    3 |  105
 C-W2 |    9    0    0    5    0    0    0    2    0    0    0    2    9    1    0    3    0    0 |   31
 C-C1 |   19    0   12    0    8    0    0    0    0    0    0    0   24   10    0    0    0    0 |   73
 C-C2 |   15    0    0   20    1   18    0    0    0    0    0    0   10    0    0    3    0    0 |   67
 E-O1 |   34    3    1    0    0    0   30    0    0    0    2    0   14    0    0    0    0    0 |   84
 E-O2 |   21    1    0    1    0    0    2   12    0    0    0    4   12    0    0    0    0    0 |   53
 I-A1 |   13    0    6    1    0    0    0    0   16    2    0    1   16    1    0    0    0    0 |   56
 I-A2 |   16    0    1    6    0    0    0    0    1   43    0    3   14    0    0    0    0    0 |   84
 P-P1 |   18    2    0    0    0    0    2    0    0    0   18    0   12    0    0    0    0    0 |   52
 P-P2 |   25    0    0    0    0    0    0    8    0    0    0   26   25    1    1    0    0    0 |   86
  _O_ |   44    0    0    1    1    0    0    0   10    0    2    0   67    2    1    0    9    2 |  139
      +--------------------------------------------------------------------------------------------+
 -SUM-   275   63   36   40   12   18   37   24   28   46   22   36  268   17    7   17   12    6    964
```

Figure 6: Relationship classification of new data without external features

Note that although test data belongs to only 6*2 of the 9*2 classes, some of the data are still being classified as one of the remaining 3*2 classes.

Without using external features,

F1 score ignoring directionality = 40.08

F1 score considering directionality = 34.75

Accuracy ignoring directionality = 37.34 %

Accuracy considering directionality = 33.30 %

Analysis

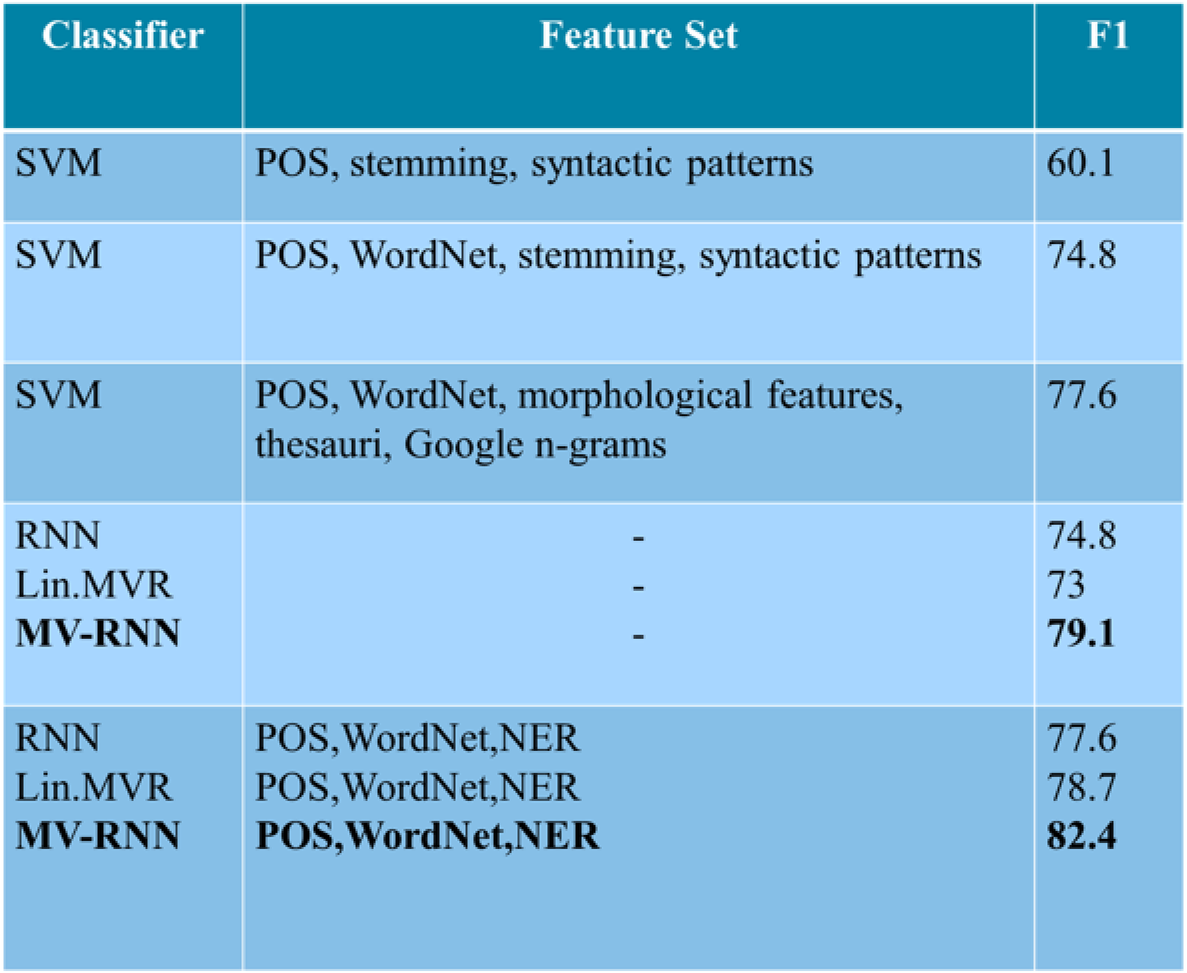
We explored the reasons for the low F1 score and came up with few possible explanations of the same:

- Dataset for different task: The dataset that we used was intended for a slightly different task. In that task, system would be given the information about the relationship about the pair of nominals and task was only to decide the direction of the relationship. For example, the system would be made to know that two given nominals have "Cause-Effect" relation among themselves, the system would have to propose which one of them is the cause and which one is effect. So the 7 semantic relations present in the raw dataset were not exhaustive and possibly overlapping[5] i.e. in the dataset strict classification of relationship wasn't done, and focus was on getting the correct directionality. So, our method was bound to fail on it as it may be possible that relationship of nominals in similar sentence can be classified into two different classifiers just to ask for the directionality given the classification. It turned out to be the correct reason as on training on the new dataset, the F1 score drastically decreased which imply that the model couldn't learn much from the training set due to poor classification.

- Another reason could also be that we used the previous training model. But it doesn't seem to be a valid reason as we tried to train on the new training data, modified to meet our requirement already mentioned above, and we were able to run the code for the same but the corresponding results were non-acceptable (F1 score was below 10) leading us to believe that either there was some bug in the code or data was very much irrelevant. We tried to analyze both of these scenario and it seems that data is too irrelevant for us.

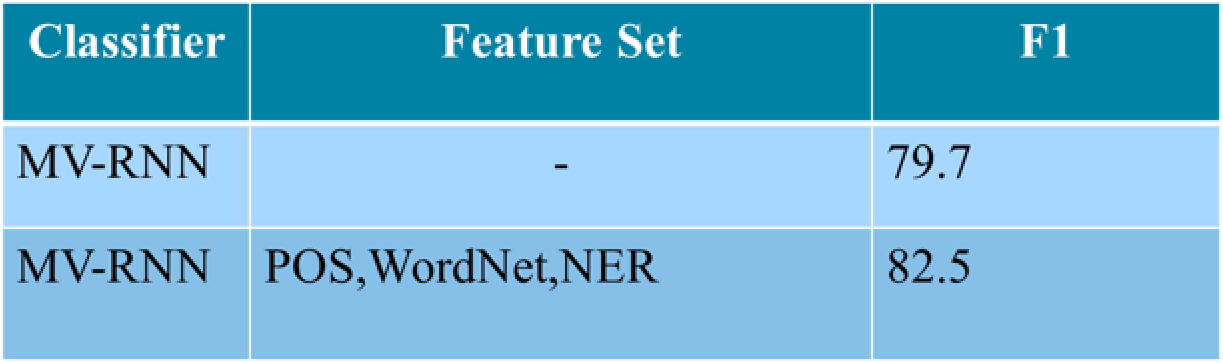
- Another reason could for poor F1 score may be that the presented technique might not be domain adaptive[6]. Our model was trained on properly annotated training set with a particular writing style, but it can be possible that the text style of the new dataset may be completely different in the way words are used. As it turns out, it may be a possible reason but not as major as that mentioned in first point as when trained on the new dataset, it gave even poorer result(F1 score was below 10).

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 Lin.MVR **MV-RNN**	- - -	74.8 73 **79.1**
RNN Lin.MVR **MV-RNN**	POS,WordNet,NER POS,WordNet,NER **POS,WordNet,NER**	77.6 78.7 **82.4**

Figure 7: Result comparison with other methods (src: all results are from ref[1])



Classifier	Feature Set	F1
MV-RNN	-	79.7
MV-RNN	POS,WordNet,NER	82.5

Figure 8: The results we obtained with and without features

Our results of MV-RNN with and without the external features are slightly better than those presented in the reference paper.

Improvement in the result of MV-RNN from other methods is 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.

  RNN inherently maintains the order by being recursive and using matrices which changes the meaning of the neighboring words.

- Consider only fixed number of neighbours around each word.

  In RNN, effect of a word isn't limited to a fixed number of words in neighborhood as while composing, the effect of modifying matrix of each word travels the hierarchical structure .

- The features used are manually developed which won't necessarily capture all the features of the word.

  RNN automatically learns the features from raw input.

Future Works

- It can be used to automatically assign relationship between words which can be used to create a knowledge graph. e.g. “The book is on the table” and “The table contains the book” both have same directional relation (Content-Container) from book to table. A huge knowledge graph can be created by expanding the number of relationships.

- It should be tried to get the semantics of a paragraph containing sentences. Weighted average of vectors of the sentences with their length as weight can be used to get the vector corresponding to the paragraph. It would be trained best on movie review with each sentence labeled but presently no such dataset is available. Treebank exists but it was created on movie review snippets, and so may not be that effective as here the aim is to classify the paragraph, not just sentences and so to learn better, its desirable to have classification of reviews along with its composing sentences.

Challenges

- For testing, a dataset of parsable sentences is needed which don’t always exist in reality, and hence the dataset need to be customized, so tough to use it on very big datasets.

- There are a lot of parameters to learn, so takes a lot of time to train. And for better training, a large amount of training data is required.

- The classification of relation between a pair of nominals is done based onlt on the lexicons in between them which may not always give correct classification as relation of two nominals may be dependent on lexicons not between them in the sentence syntax.

- The training is supervised and the data is manually annotated which is bound to have human errors and fine-grained sentiment analysis can’t be easily achieved as people usually mark the sentences discretely.

- One major drawback of this model is that it can't learn current world-knowledge unless trained explicitly as it's tough to get period of time from the data as sentence meaning may change with time.

- Its accuracy can be increased by annotating the semantics of each node of the parse tree as done by Stanford Treebank but then it raises the question: Till what extent we need to manually annotate, and it can't be done manually for all the data. Moreover, a lot of small phrases have neutral semantics so it doesn't help a lot[7].

Conclusion

- The introduction of matrix-vector representations with a recursive neural network is the main idea which sets this model apart from other existing models.

- The model introduced in this work not only learns word vector representation of a word but also that how it modifies other words by learning the corresponding matrix.

- This MV-RNN model is good not only theoretically but also presents us with much better performance in terms of accuracy and F1 score on large dataset than other existing methods.

References

(1) Socher, R.; Huval, B.; Manning, C. D.; Ng, A. Y. Semantic compositionality through recursive matrix-vector spaces. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. 2012; pp 1201–1211.

(2) SemEval-2010. http://semeval2.fbk.eu/semeval2.php?location=tasks#T11, 2010; [Online; accessed 18-April-2015].

(3) Mitchell, J.; Lapata, M. *Cognitive science* **2010**, *34*, 1388–1429.

(4) Hashimoto, K.; Miwa, M.; Tsuruoka, Y.; Chikayama, T. Simple Customization of Recursive Neural Networks for Semantic Relation Classification. EMNLP. 2013; pp 1372–1376.

(5) SemEval-2007. `http://nlp.cs.swarthmore.edu/semeval/tasks/task04/description.shtml`, 2007; [Online; accessed 18-April-2015].

(6) Qiu, Q.; Patel, V. M.; Turaga, P.; Chellappa, R. *Computer Vision–ECCV 2012*; 2012; pp 631–645.

(7) Socher, R.; Perelygin, A.; Wu, J. Y.; Chuang, J.; Manning, C. D.; Ng, A. Y.; Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. Proceedings of the conference on empirical methods in natural language processing (EMNLP). 2013; p 1642.