Sequential short text classification using recurrent neural networks

Ji Young Lee

Introduction

- Short text classification has applications such as question answering systems and sentiment analysis
- Many existing systems do not incorporate the context of short texts
- Our system classifies short texts:
  - using recurrent neural networks (RNNs)
  - based on context

MR dataset

- Classify movie reviews into bipolar sentiment:
  - POSITIVE, or NEGATIVE
- 10662 samples (no official train/test split)

Sample	Class
travels a fascinating arc from hope and euphoria to reality and disillusionment .	POSITIVE
an easy watch , except for the annoying demeanour of its lead character .	NEGATIVE

TREC dataset

- Classify questions into 6 semantic classes:
  - DESCRIPTION, ENTITY, NUMERICAL VALUE, HUMAN, LOCATION, ABBREVIATION
- 5952 samples (500 samples as test set)

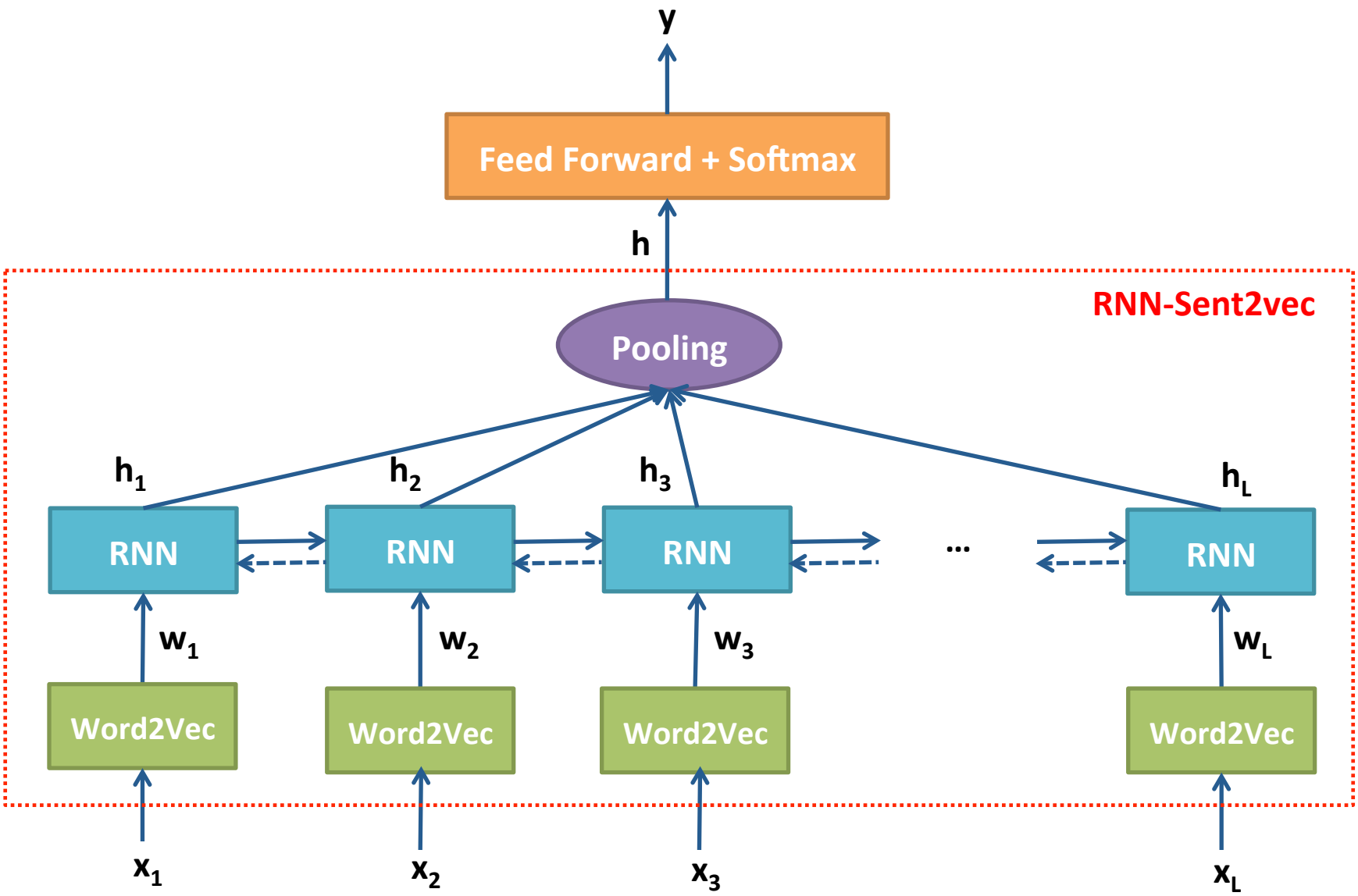
Sample	Class
Why do heavier objects travel downhill faster ?	DESCRIPTION
What is the birthstone of October ?	ENTITY
What is Susan B. Anthony 's birthday ?	NUMERICAL

DSTC4 dataset

- 35 dialogs (31,034 utterances) between a guide and a tourist (6 dialogs as test set)
- Classify each utterance into 88 speech acts (4 categories x 22 attributes)
  - **4 categories**: QUESTION, RESPONSE, INIATIVE, FOLLOW
  - **22 attributes**: ACKNOWLEDGE, CANCEL, CLOSING, COMMIT, CONFIRM, ENOUGH, EXPLAIN, HOW_MUCH, HOW_TO, INFO, NEGATIVE, OPENING, POSITIVE, PREFERENCE, RECOMMEND, THANK, WHAT, WHEN, WHERE, WHICH, WHO

Sample	Class
T: Can you recommend a hotel?	QST:RECOMMEND
G: Alright.	RES:POSITIVE
G: On Orchard Road there are a number of hotels.	FOL:INFO
T: Okay.	FOL:ACKNOWLEDGE

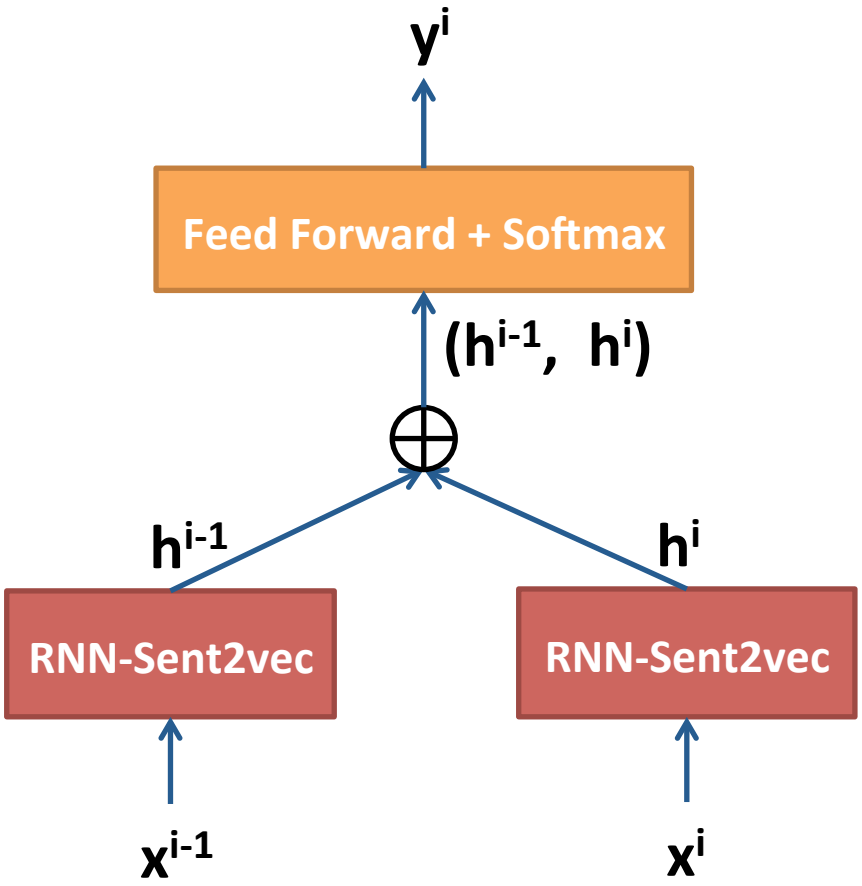
RNN-based classification system



Model options and hyperparameters

- **Word2Vec:**
  - pre-trained word vectors from word2vec
  - random initialization
- **RNN:**
  - LSTM or GRU
  - mono- or bi-directional
  - output dimension: 50, 100, 200, 300, or 500
- **Pooling:**
  - last
  - mean
  - max
  - content-based
- **Regularization:**
  - dropout
  - weight decay
  - maxnorm

With bigram context



Training/Experimental Setup

- Stochastic Gradient Descent
- Adadelta update rule
- Early stopping based on validation set

- Train/Validation/Test split:
  - Test set: official split (if not given, randomly select 10%)
  - Validation set: randomly select 10% of non-test set

Results compared with baselines

Model	MR	TREC	DSTC4
RNN	81.4	91.2	**66.7**
CNN	**81.5**	93.6	-
SVM_S	-	**95.0**	-
SVM	-	-	56.3
LR	-	-	56.0
MV-RNN	79.0	-	-
Sent-Parser	79.5	-	-
NBSVM	79.4	-	-
Tree-CRF	77.3	-	-

Baselines

- MR, TREC: results in the literature, as summarized by (Kim, 2014)
- DSTC4: SVM & LR with n-grams and a few hand-written features such as number of question marks, utterance lengths, whether speaker changed

	DSTC4	
RNN output dim	Non-sequential	Sequential
50	49.1	55.7
100	55.0	59.7
200	53.1	59.2
300	50.1	58.6
500	55.3	**66.7**

	MR		TREC	
pooling	mono-dir	Bi-dir	Mono-dir	bi-dir
last	80.6	**81.4**	90.8	90.0
mean	78.1	79.0	90.6	**91.2**
max	79.1	80.1	90.8	90.2
content	79.4	79.8	**91.2**	90.6

Conclusion

- **Non-sequential classification:** comparable to other baselines
- **Sequential classification**: RNN-based architecture with bigram context beats the baselines using traditional methods
- **Future work:**
  - Doing controlled test for different hyperparameters
  - Verifying the performance on more datasets
  - Trying different pre-trained word vectors, and making them static
  - More complex mechanism for incorporating context: trigram context or hierarchical RNN
- **References:**

  - Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014)
  - Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014)

Appendix: Content-based Pooling

To combine $h_1, \ldots, h_L$ into a single vector h, take a convex combination

$$h = \sum_{i=1}^{L} \alpha_i h_i$$

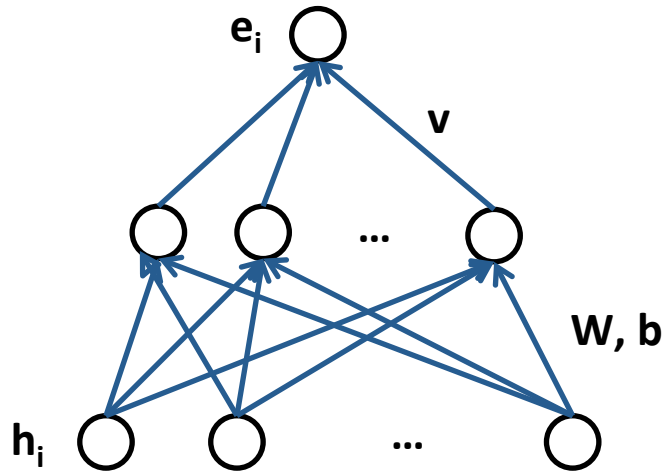
where α_i is determined by

- First, computing e_i as the scalar output of feed forward neural network with 1 hidden layer from each h_i

$$e_i = v^T tanh(W h_i + b)$$

- Then, taking the softmax of $(e_1, \ldots, e_L)$

$$\alpha_i = \frac{exp(e_i)}{\sum_{i=1}^{L} exp(e_i)}$$



(Motivated by attention-based mechanism for machine translations)