Injecting Relational Structural Representation in Neural Networks for Question Similarity

UNIVERSITY OF TRENTO - Italy
Department of Information Engineering and Computer Science

Antonio Uva, Daniele Bonadiman
[1]DISI, University of Trento 38123 Povo (TN), Italy
{auva87,d.bonadiman}@gmail.com

Alessandro Moschitti
Amazon, Manhattan Beach, CA, USA, 90266
amosch@amazon.com

Motivation

- Recent years have seen an exponential growth of forums, leading to the development of **Community Question Answering** (cQA) technology.
- cQA systems respond to new user questions using the answers to similar previously asked questions in the forums.
- **Syntactic information** is essential to achieve high accuracy in question reranking and question duplicate detection tasks (Nakov et al., 2016)
- **Neural Networks** (NNs) are every effective to manage lexical variability.
- Unfortunately, effectively using syntactic information in NNs is still an open problem
- We need an approach that aims at injecting syntactic information in NNs, still keeping them simple.

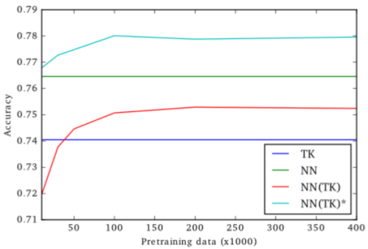
Injecting Structures in NNs

- **NNs for Question Similarity.** We experimented with two encoders:
  - CNN model proposed by (Severyn and Moschitti, 2016); and
  - a Bidirectional LSTM (BiLSTM) encoder.
- **Injecting structural information in the network**. Weak supervision technique

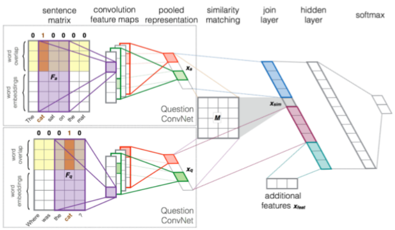
(i) An SVM using Tree Kernels (TKs) is trained on the GS data

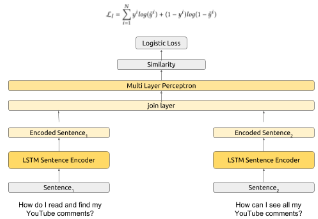
(ii) The SVM model classifies an additional unlabeled set producing automatic data

(iii) A neural network is trained on the gold and the automatic data.



NNs Models for Question Similarity





- CNN model (Severyn and Moschitti, 2016)
- Word overlap embeddings for modeling **relational info**.
- Similarity matrix to compute question-question sim score.
- Joint layer concat question repr., word overlap and sim. score.

- BiLSTM
- Use two LSTM sentence encoders.
- Minimize train log loss.
- Concatenate question repr. and feed to MLP.

Experiments

Mode	Automatic data	GS data	DEV	TEST
TK-10k	-	10k	0.7405	0.7337
CNN-10k	-	10k	0.7646	0.7569
LSTM-10k	-	10k	0.7521	0.7450
CNN(CNN-10k)*	50k	10k	0.7601	0.7598
CNN(TK-10k)*	50k	10k	0.7748	0.7652
LSTM(TK-10k)*	50k	10k	0.7706	0.7505
CNN(TK-10k)*	375k	10k	**0.7796**	**0.7728**
Voting(TK+CNN)	-	10K	0.7838	0.7792

Table 1: Accuracy on Quora dataset

Model	Automatic data	DEV	TEST
CNN	-	0.7000	0.7514
TK	-	0.7340	0.7686
CNN(TK)	50k	0.5580	0.5428
CNN(TK)*	50k	0.7160	**0.7814**
CNN(TK)	93k	0.7000	0.6957
CNN(TK)*	93k	**0.7380**	0.7614

Table 2: Accuracy on QL dataset

Results and Conclusions

- CNN(TK)* improves **0.4%** and **3%** Accuracy points on QL dev. and test sets, respectively over the TKs
- CNN(TK-10K)* improves **1%** Accuracy points on Quora dev. and test sets wrt the CNN. baseline.
- CNN(TK-10K)* model reaches almosts the same performance of a voting model combining CNN and TK.
- This seems to show that NN learned the combination of **lexical info** of the CNN model and **syntactic info** of TKs.