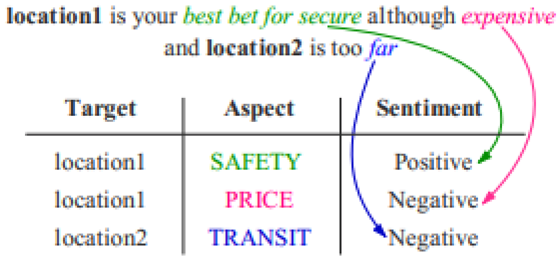
since 1995

Day 123: NLP Papers Summary – Context-Aware Embedding For Targeted Aspect-Based Sentiment Analysis

By Ryan 2nd May 2020 No Comments

Objective and Contribution

Proposed context-aware embeddings to refine the embeddings of targets and aspects using highly correlative words. A lot of previous work uses context-independent vectors to construct targets and aspects embeddings, which lead to loss of semantic information and failed to capture the interconnection between the specific target, its aspect, and its context. This approach has led to SOTA results in targeted aspect-based sentiment analysis (T/ The goal of TABSA is that given an input sentence, we want to extract the sentiment of the aspect that belongs to a target. Figure below showcase the TABSA task:



location1 is your *best bet for secure* although *expensive* and **location2** is too *far*

Target	Aspect	Sentiment
location1	SAFETY	Positive
location1	PRICE	Negative
location2	TRANSIT	Negative

Figure 1: Example of TABSA task. Highly correlative words and corresponding aspects are in the same color. Entity names are masked by **location1** and **location2**.

The contributions are as follows:

1. Construct context-aware embeddings for targets by using sparse coefficient vectors to identify words that are highly correlated to the targets and refining the target embeddings accordingly
2. Fine-tuned aspect embeddings to be as close to the highly correlated target embeddings
3. Achieved SOTA results on SentiHood and SemEval 2015

Methodology

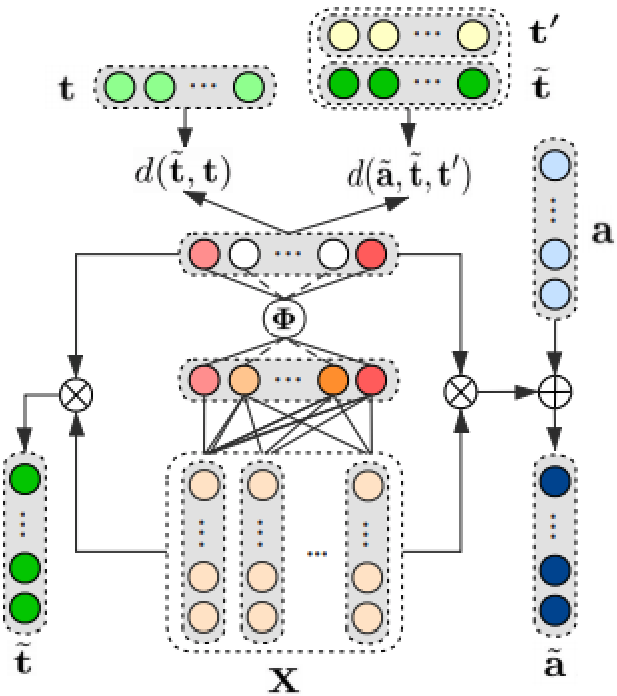


Figure 2: The framework of our refinement model. ⊗ is element-wise product, ⊕ is vector addition, Φ is step function.

The model framework has the following steps:

1. Sentence embedding matrix X is feed into the fully connected layer and step function to create sparse coefficient vector u'.
2. The hidden output of u' is used to refine the target and aspect embeddings
3. Compute the squared Euclidean function and train the model to minimise the distance to obtain the final refined embeddings for target and aspect

TARGET REPRESENTATION

The refined target embeddings can be computed by multiplying the sentence word embeddings X with the sparse coefficient vector u'. The sparse coefficient vector showcase the importance of different words in the context using a step function. For each target, we compute the context-aware target embedding by iteratively minimising the squared Euclidean distance between the target and the highly correlative words in the sentence.

ASPECT REPRESENTATION

We refined the aspect embeddings by using the sparse coefficient vectors of highly correlated words. The argument behind this is that aspects words usually contain important information and context information often has a high connection to the aspects. Again, for each aspect, we compute the context-aware aspect embeddings by minimising the squared Euclidean distance between aspect embeddings, context-aware target embeddings, and irrelevant embeddings. This would fine-tuned our aspect embeddings to move closer to our highly correlated target embeddings and move away from irrelevant ones.

Experiments and Results

There are two evaluation datasets: SentiHood and SemEval 2015 Task 12.

MODELS COMPARISON

1. *LSTM-Final*. BiLSTM that only uses the final hidden states
2. *LSTM-Loc*. BiLSTM that uses the hidden states where the location target is
3. *SenticLSTM*. BiLSTM that uses external knowledge

4. *Delayed-Memory*. Delayed memory mechanism
5. *RE+SenticLSTM*. Our refined embeddings + SenticLSTM
6. *RE+Delayed-Memory*. Our refined embeddings + Delayed-Memory

RESULTS

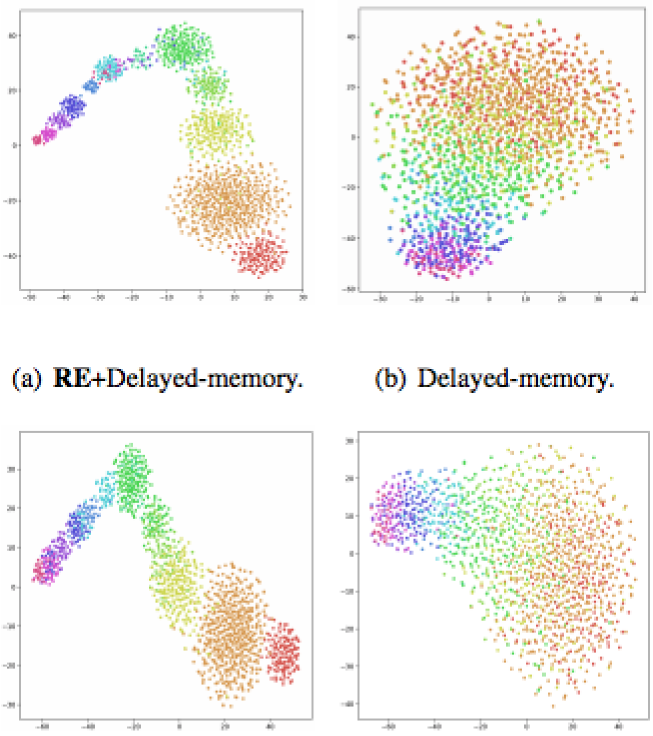
Model	Aspect Detection			Sentiment Classification	
	Acc. (%)	F1 (%)	AUC (%)	Acc. (%)	AUC (%)
LSTM-Final	—	68.9	89.8	82.0	85.4
LSTM-Loc	—	69.3	89.7	81.9	83.9
SenticLSTM	67.4	78.2	—	89.3	—
RE+SenticLSTM† **(ours)**	73.8	79.3	—	**93.0**	—
Delayed-memory	73.5	78.5	94.4	91.0	94.8
RE+Delayed-memory† **(ours)**	**76.4**	**81.0**	**96.8**	92.8	**96.2**

Table 1: Experimental results on SentiHood. † denotes average score over 10 runs, and best scores are in bold.

Model	Aspect Detection			Sentiment Classification	
	Acc. (%)	F1 (%)	AUC (%)	Acc. (%)	AUC (%)
SenticLSTM	67.3	76.4	—	76.5	—
RE+SenticLSTM†**(ours)**	71.2	78.6	89.5	76.8	82.3
Delayed-memory	70.3	77.4	90.8	76.4	83.6
RE+Delayed-memory†**(ours)**	**71.6**	**79.1**	**91.8**	**77.2**	**84.6**

Table 2: Experimental results on Semeval 2015.

For SentiHood, our proposed methods added on top of SenticLSTM and Delayed-Memory achieved better performance than the original models in both aspect detection and sentiment classification. Our context-aware embeddings has allowed models to better capture aspects and sentiment information as we are able to better model the interconnection between the target, its aspects and the context. For SemEval 2015, we showcase similar results with our proposed methods outperforming the original models. Below is the figure that visualise our proposed context-aware embeddings vs the original aspect embeddings using TSNE. As shown, there are more separation between different aspects using our context-aware embeddings, showcasing its ability to distinguish between different aspects in the context as well as its ability to capture the common traits of specific aspects.



(a) **RE**+Delayed-memory. (b) Delayed-memory.

(c) **RE**+SenticLSTM. (d) SenticLSTM.

Figure 3: The visualization of intermediate embeddings learned by embedding-based models. Different colors represent different aspects.

Conclusion and Future Work

By selecting and using highly correlated words to refine targets and aspects embeddings, we are able to extract the interconnection between specific target, its aspect and its context, to generate a better meaningful embedding. Future work involves exploring this method for other similar NLP tasks.

Source: https://www.aclweb.org/anthology/P19-1462.pdf

Ryan

Data Scientist

Previous Post

Day 122: NLP Papers Summary - Applying BERT to Document Retrieval with Birch

Next Post

Day 124: NLP Papers Summary - TLDR: Extreme Summarization of Scientific Documents

Data Science Natural Language Processing NLP Papers Summary

Day 365: NLP Papers Summary – A Survey on Knowledge Graph Embedding

Ryan
30th December 2020

Data Science Implementation Natural Language Processing

Day 364: Ryan's PhD Journey – OpenKE-PyTorch Library Analysis + code snippets for 11 KE models

Ryan
29th December 2020

Data Science Ryan's PhD Journey

Day 363: Ryan's PhD Journey – Literature Review – Knowledge Acquisition – 1st Passes XIV

Ryan
28th December 2020