CS 563: Deep Learning for Sentiment Analysis

Asif Ekbal
AI-NLP-ML Research Group
Dept of Computer Sc & Engineering
IIT Patna, Patna
Email: asif.ekbal@gmail.com, asif@iitp.ac.in

Outline

- Sentiment Analysis: Introduction, Background
- Fine-grained Sentiment Analysis: Aspect based Sentiment Analysis (*ABSA*)
- Deep Learning Approaches to Sentiment Analysis
    - Convolutional Neural Network (CNN)
    - Recurrent Neural Network (RNN)
    - Long Short Term Memory (LSTM)
    - Attention Mechanism
    - Memory Network

- Inter-aspect dependency
- Target-specific representation
- Context-dependent target
- Multilinguality and Cross-linguality
- Takeaways

Sentiment Analysis

Sentiment analysis aims to identify the orientation of opinion in a piece of text

Why do we need Sentiment Analysis?

- *What others think* has always been an important piece of information
- *Overwhelming amount of information* on one topic: Manually reading or analysing all data is very inefficient

- *Biased/Fake* reviews
- An example
  - Mr. X needs to buy a phone. He was browsing amazon.in and found 1000 reviews for a particular phone.

Scenario 1:

- Let there are ***850 negative***, ***100 positive*** and ***50 neutral*** reviews
- Sentiment → **Negative**.

Scenario 2:

- Let there are ***420 negative***, ***480 positive*** and ***100 neutral*** reviews.
- Sentiment → **Positive**

What if all the 100 positive reviews are at the top?

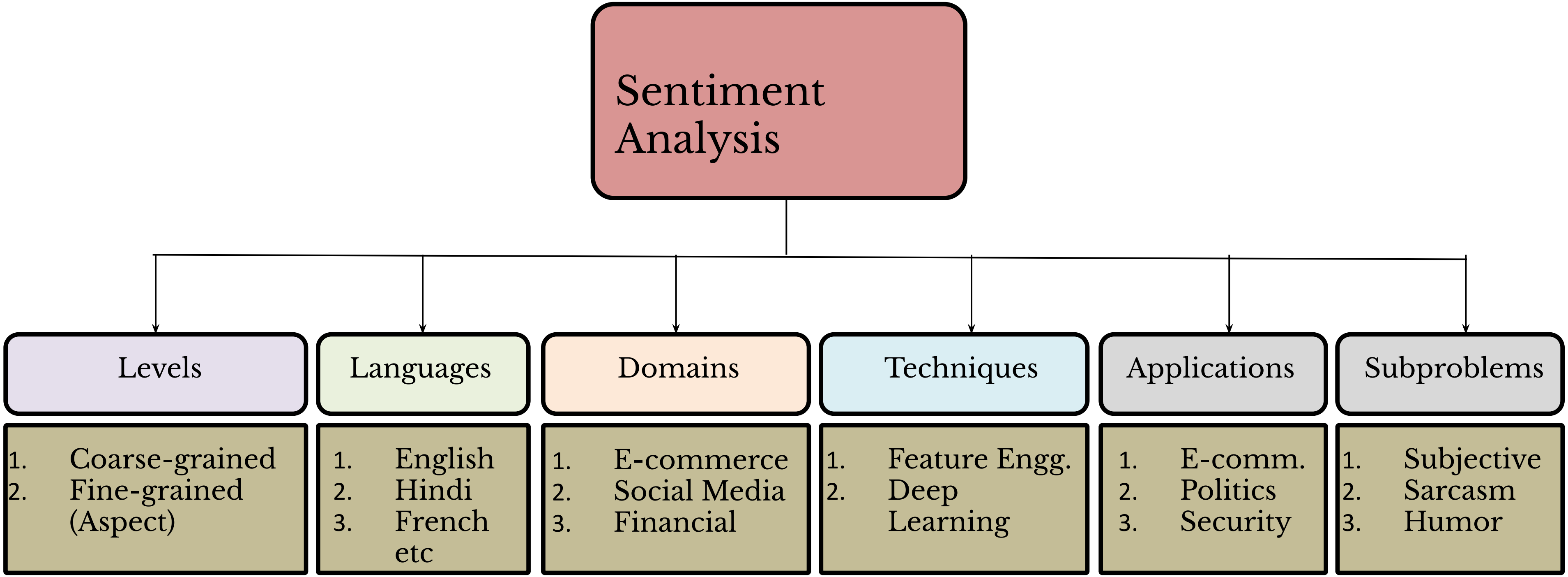
What if few of the reviews (e.g. 100) are fake?

Challenges

- Similar lexical features but different sentiments
    - **This movie is not good**
    - **No movie can be better than this**
- Different styles of writing but same sentiment
    - **It's an extremely useless phone**
    - **I have wasted my money on this phone**
    - **I could have bought iphone instead of this**
- Product name, even, may appear in different forms
    - G-phone, Google-phone etc.
- Sentiment lexicons are not sufficient for sentiment analysis

  "*The food is very **cheap** here.*" vs "*The service is very **cheap** here.*"
- Reviews may not be genuine

Sentiment Analysis: A broader view



Sentiment Analysis: Subproblems

Subproblems	Text	Remarks
Subjectivity	*This movie is **awesome**.*	Positive
	*This movie is **pathetic**.*	Negative
	*This movie is **3-hours long**.*	Neutral
Thwarting	***Impressive** story, **good** acting, however, it **didn't meet** my expectation.*	Small portion at the end dictates its sentiment.
Sarcasm	*This movie is **awesome to put you to sleep**.*	Criticism in a humorous way.
Humble Bragging	*My life is **miserable**, I have to **sign 300 autographs per day**.*	Draw attention to something of which someone is proud.
Discourse-based SA	*This movie is a classic, **although**, I don't like 'sci-fi'.*	Sentiment is altered due to connectives.
Sense-based SA	*Shane Warne is a **deadly** spinner. (Positive)*	Different sense leads to different sentiments.
	*The campus has **deadly** snakes. (Negative)*	
Sentiment Intensity	*Movie was **ok**.*	Weak positive sentiment.
	*Movie was **good**.*	Mild positive sentiment.
	*Movie was **awesome**.*	Strong positive sentiment.

Sentiment Analysis: Granularity

- Based on the granularity of analysis, we can categorize it as:
  - Coarse-grained Sentiment Analysis (**Document-level or Sentence-level**)
  - Fine-grained Sentiment Analysis (**Phrase-level or Aspect-level**)

- **Aspect Based Sentiment analysis (ABSA):** Sentiment towards an aspect (or opinion-target or feature)

  Its battery is awesome but camera is very poor.

  इसकी बैटरी शानदार है, लेकिन कैमरा बहुत ही ख़राब है।

  (*Isakee baiTaree shaanadaara hai, lekin kaimaraa bahut hee kharaab hai..*)

  Positive about the ***battery*** but ***negative*** about the ***camera***

Aspect Term Extraction

Given a set of sentences with pre-identified entities (e.g., restaurants), identify the aspect terms present in the sentence and return a list containing all the distinct aspect terms

" I liked the *service* and the *staff* , but not the *food* " →

service, staff , food

"*Ambience* and *music* funky, which I enjoy" →

Ambience, music

"Awesome *form factor* and great *battery life*" →

form factor, battery life

Polarity Identification

For a given set of aspect terms within a sentence, determine whether the polarity of each aspect term is *positive, negative, neutral* or *conflict* (i.e., both positive and negative)

"I liked the *service* and the *staff*, but not the *food*"

→ { **service:** *Positive*, **staff:** *Positive*, **food:** *Negative* }

"I did add a *SSD drive* and *memory*"

→ { **SSD drive**: *Neutral*, **memory:** *Neutral* }

"The *RAM memory* is good but should have splurged for 8Mb instead of 4Mb"

→ { **RAM memory:** *Conflict* }

Aspect Based Sentiment Analysis: Few examples

*The **speed**, the **design**.. it is lightyears ahead of any PC I have ever owned.*	Speed, Design	Positive, Positive
***Tech support** would not fix the problem unless I bought your plan for $150 plus.*	Tech support	Negative
*Certainly not the best **sushi** in New York, however, it is always fresh, and the **place** is very clean, sterile.*	Sushi, Place	Conflict, Positive
It was very expensive for what you get.	Implicit aspect (price), however, no textual presence implies no aspect term	
I enjoy having Apple products.	-No Aspect term-	

Informed Decision: Coarse-grained vs Fine-grained SA

The zoom is excellent, but the LCD is blurry.
Great value for the price.
Although the display is poor the picture quality is amazing.
Batteries drain pretty quickly.
I love this camera but for short battery life is definitely a pain.
It is a good camera for the price.

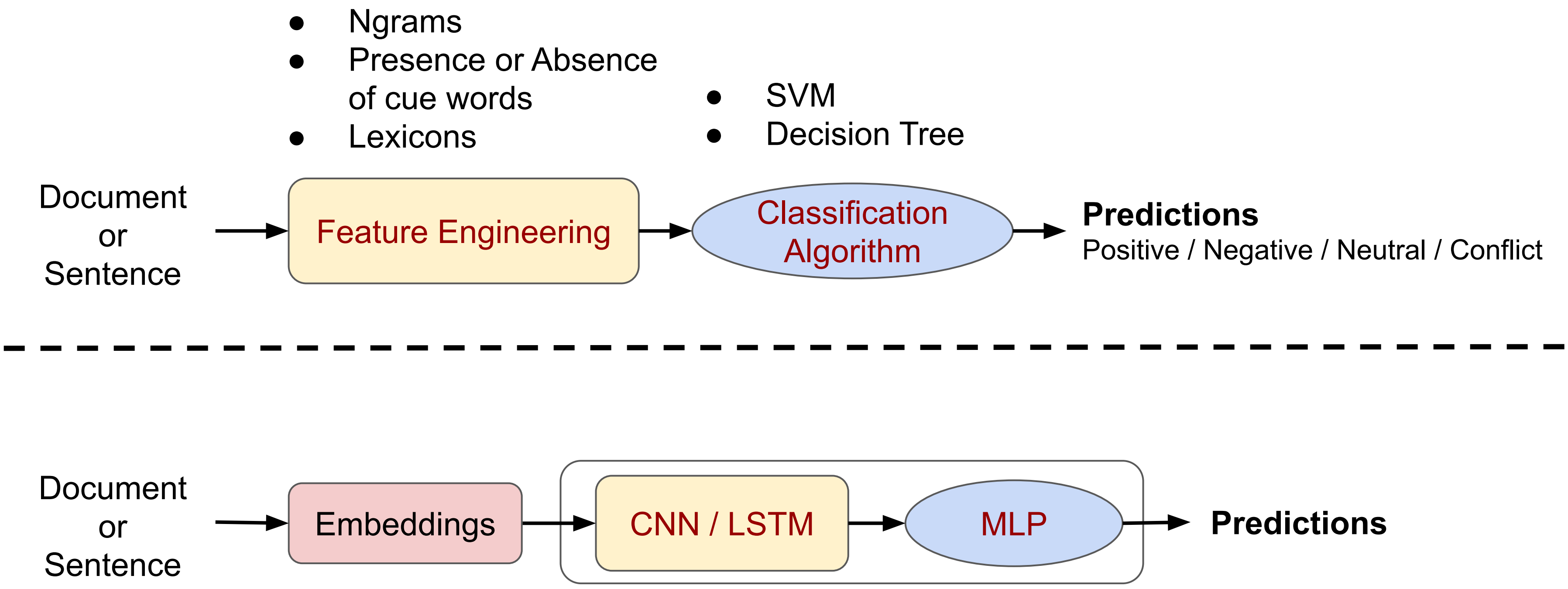
Sentence-level SA

- Product X: 3.5 ★

Aspect-level SA

- Price: 4.5 ★
- Zoom: 5.0 ★
- Picture Quality: 4.5 ★
- Battery: 1.5 ★
- Display: 1.0 ★

Traditional ML vs. DL pipeline



Convolutional Neural Network

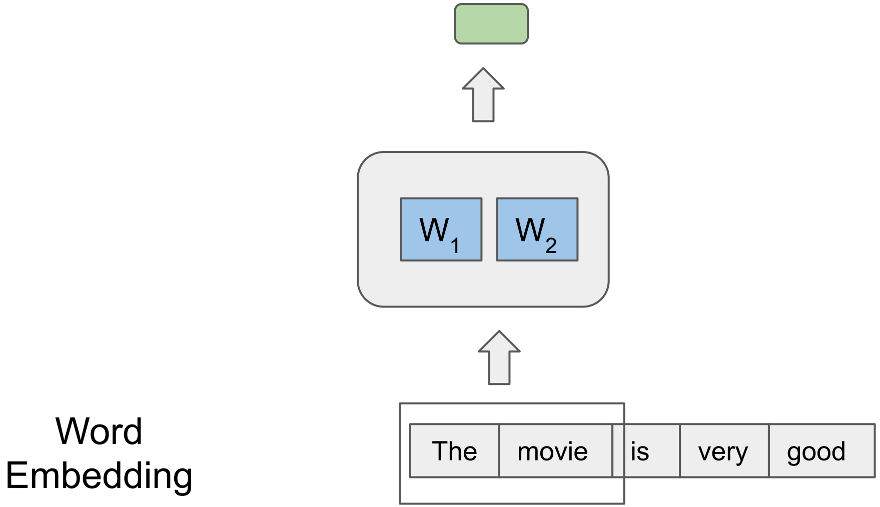
What is CNN?

- Convolutional neural network (CNN, or ConvNet) is a type of feed-forward artificial neural network where the individual neurons are tiled in such a way that they respond to overlapping regions in the input field. (wikipedia)

- *CNNs are good at learning features from the data*
    - Local connectivity
    - Share weights/parameters across spatial positions

CNNs for Sentiment Analysis

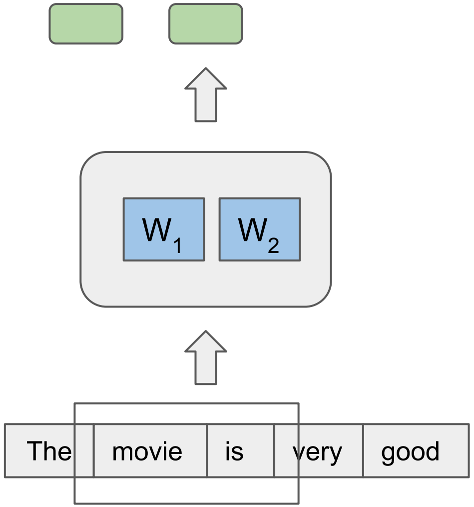
Use Feedforward neural network on consecutive *ngram* words



Simple Linear layer looking at two words at a time

CNNs for Sentiment Analysis

Use Feedforward neural network on consecutive *ngram* words

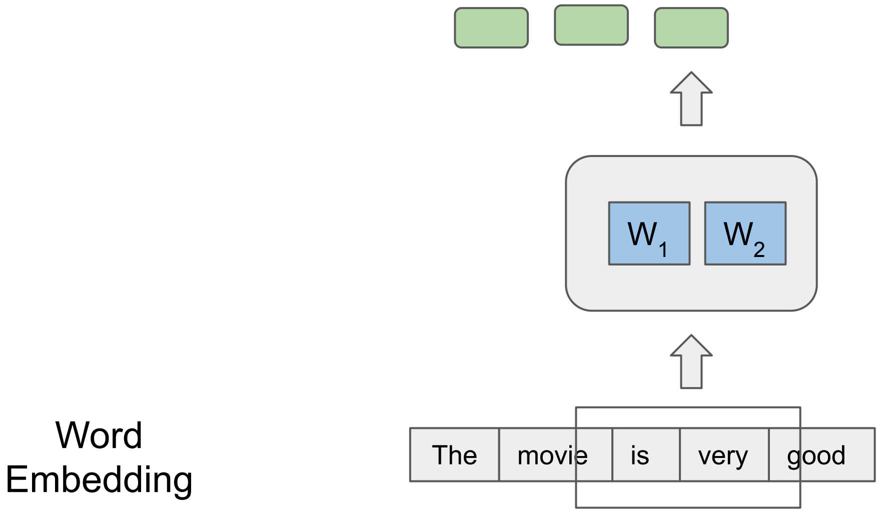


Word Embedding

Simple Linear layer looking at two words at a time

CNNs for Sentiment Analysis

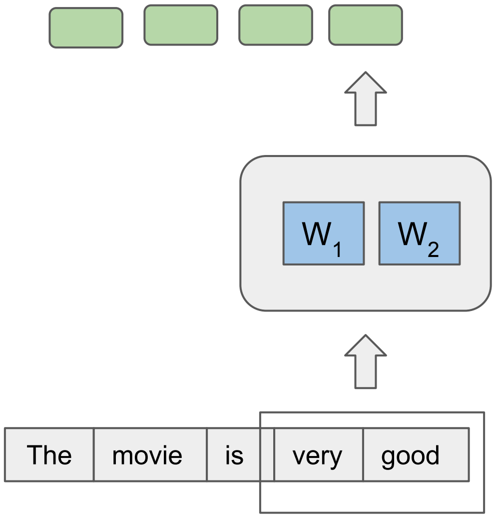
Use Feedforward neural network on consecutive *ngram* words



Simple Linear layer looking at two words at a time

CNNs for Sentiment Analysis

Use Feedforward neural network on consecutive *ngram* words

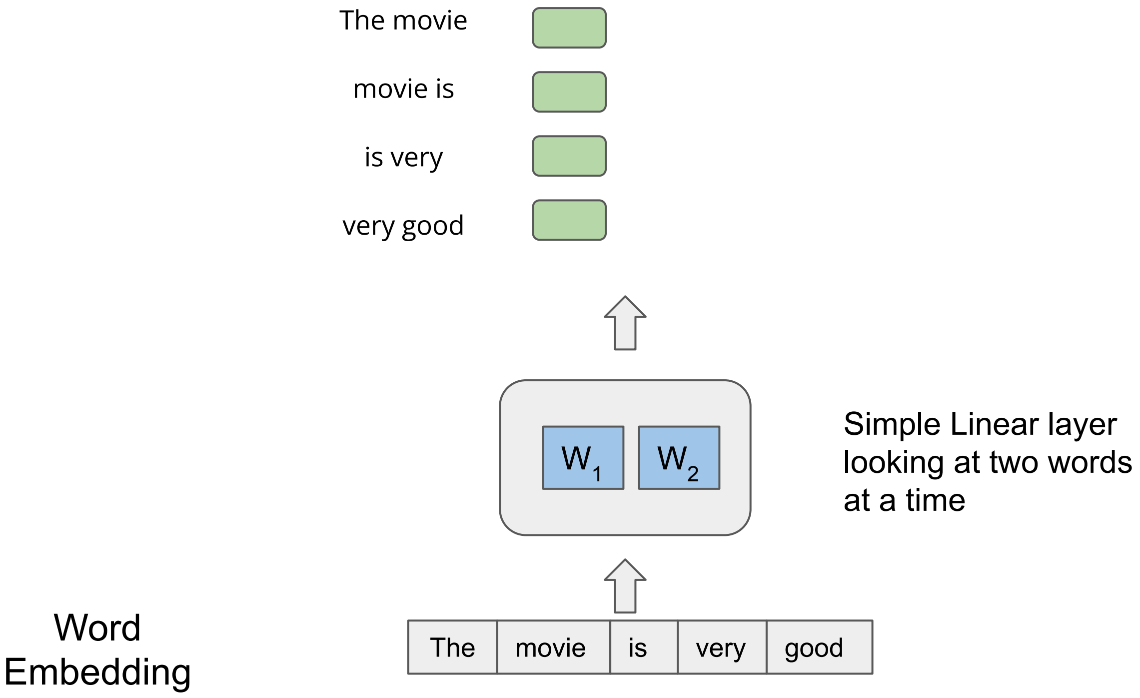


Simple Linear layer looking at two words at a time

Word Embedding

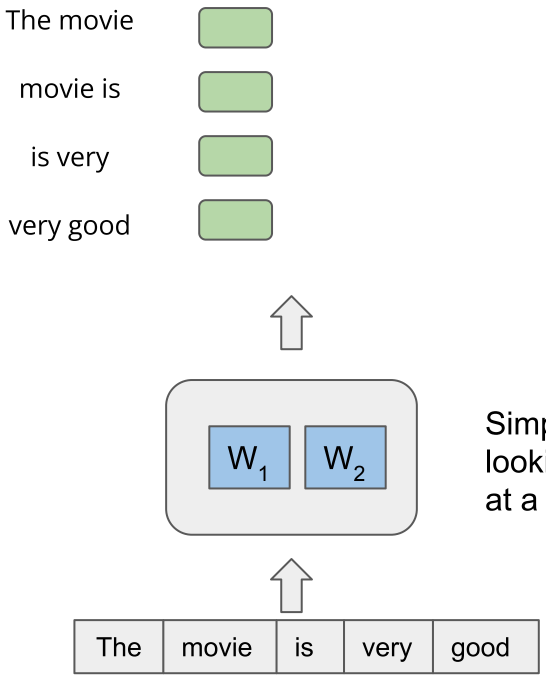
CNNs for Sentiment Analysis

Use Feedforward neural network on consecutive *ngram* words



CNNs for Sentiment Analysis

Use Feedforward neural network on consecutive *ngram* words



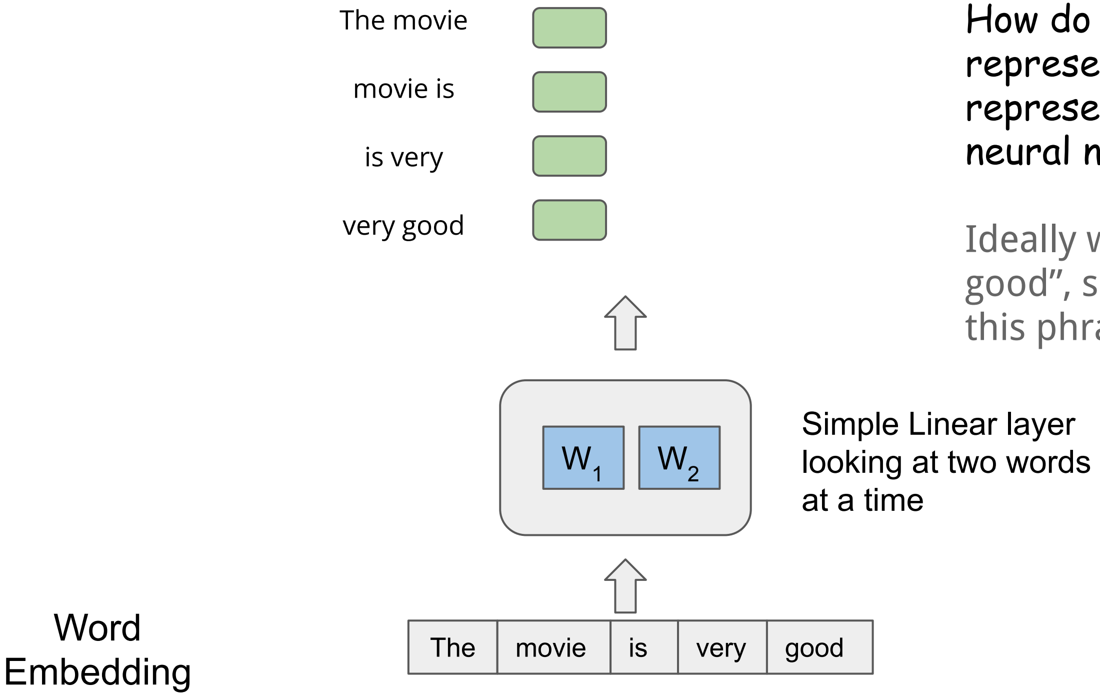
How do we go from variable length representation to a fixed length representation so that the feed-forward neural network can handle?

Simple Linear layer looking at two words at a time

Word Embedding

CNNs for Sentiment Analysis

Use Feedforward neural network on consecutive *ngram* words

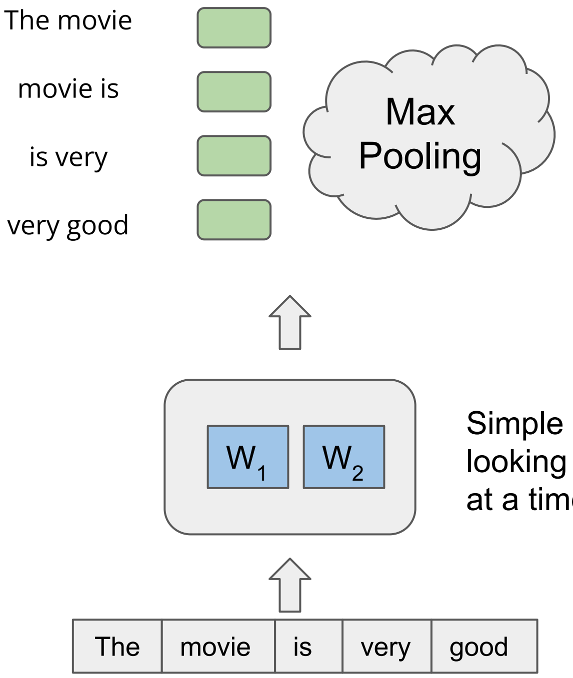


How do we go from variable length representation to a fixed length representation so that the feed-forward neural network can handle?

Ideally we have to choose the phrase "very good", so weightage have to be given to this phrase

CNNs for Sentiment Analysis

Use Feedforward neural network on consecutive *ngram* words



Simple Linear layer looking at two words at a time

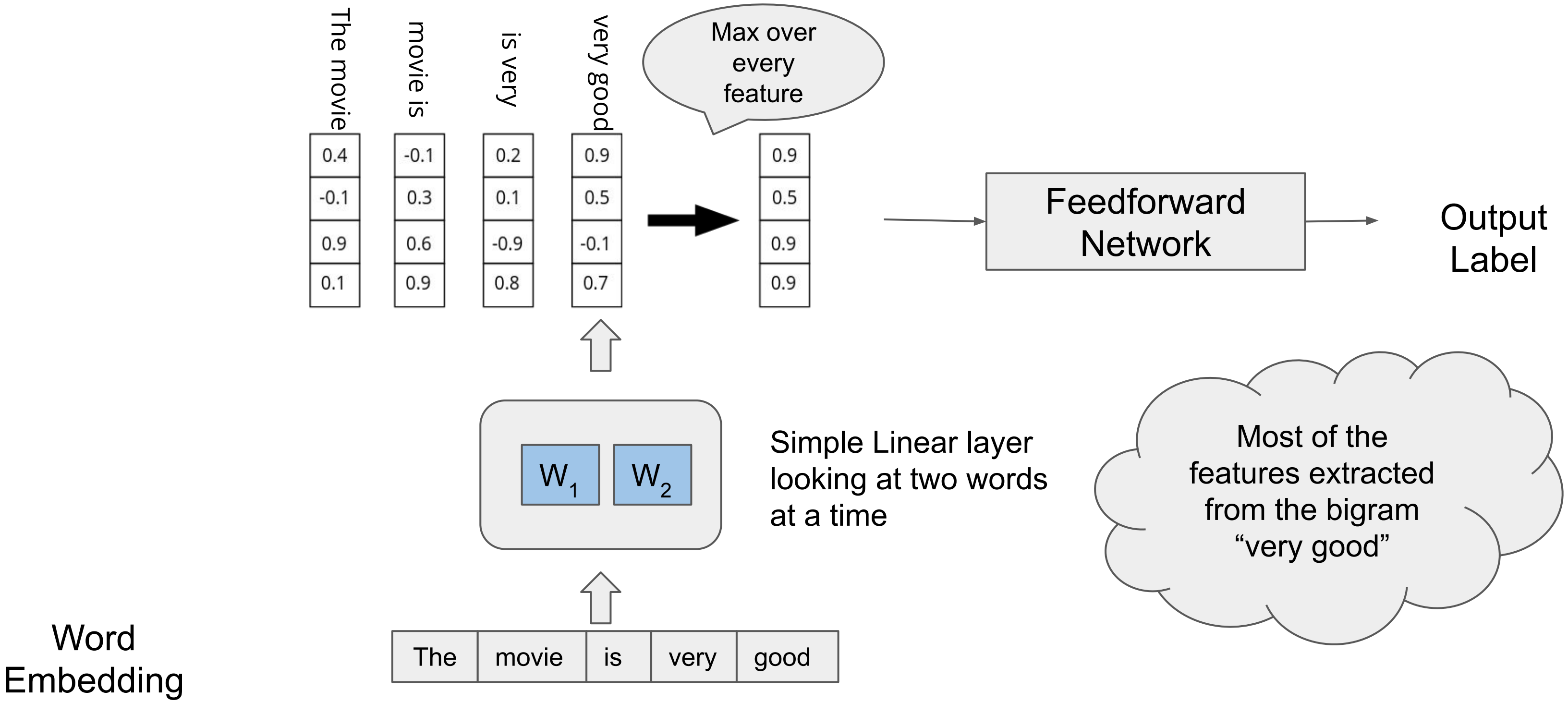
Word Embedding

How do we go from variable length representation to a fixed length representation so that the feed-forward neural network can handle?

Ideally we have to choose the phrase "very good", so weightage have to be given to this phrase

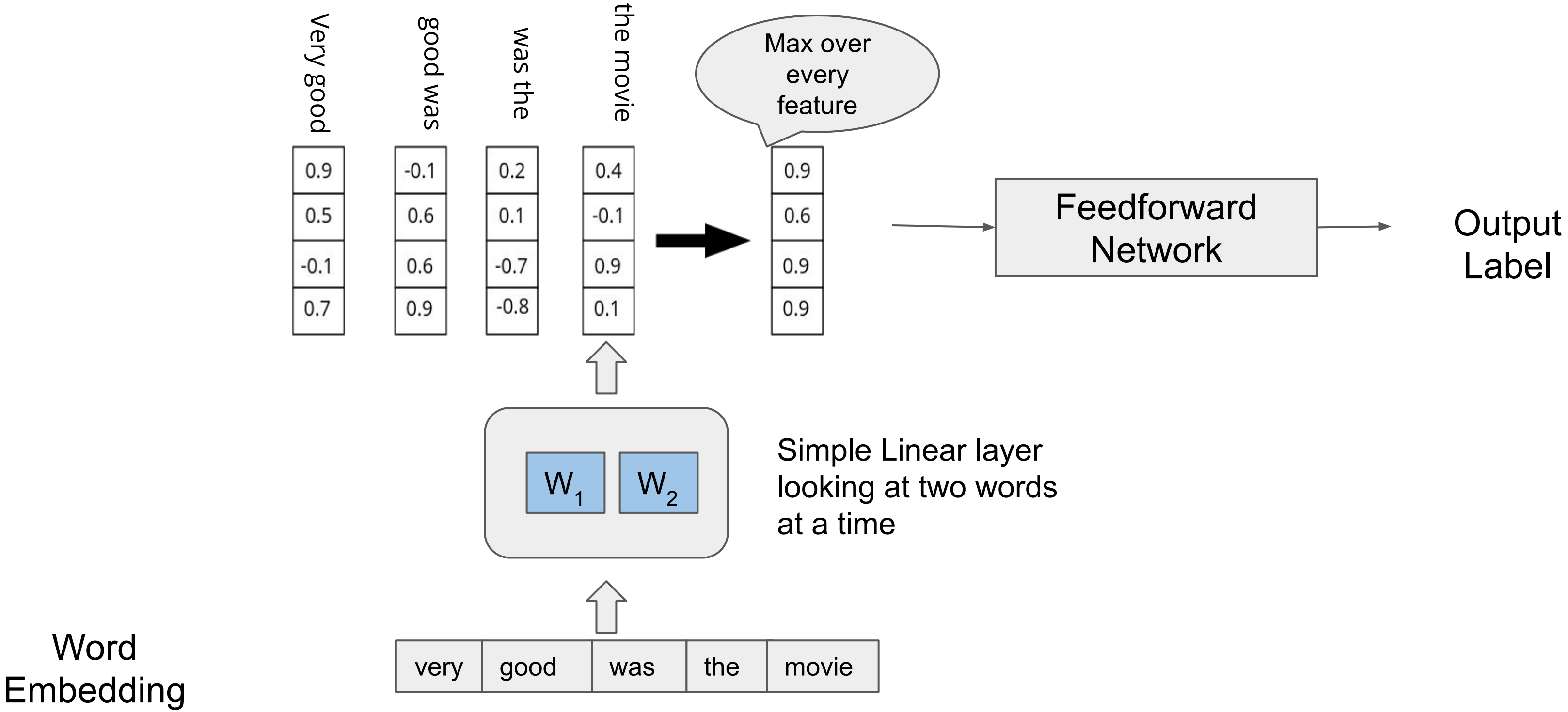
CNNs for Sentiment Analysis

Use Feedforward neural network on consecutive *ngram* words



CNNs for Sentiment Analysis

Use Feedforward neural network on consecutive *ngram* words



A Hybrid Deep Learning Architecture for Sentiment Analysis [Akhtar et al. 2016]

- CNN based hybrid architecture for sentiment analysis
  - Replace a weak classifier (**softmax regression)** with a stronger classifier (**SVM**) at the output layer
- Assist CNN with optimized feature set obtained through GA based multiobjective optimization

- For each aspect, look for the sentiment marker near the aspect term itself
  - Define context as +/- few words (e.g., 3) in the neighbourhood, i.e., 3 prev tokens and 3 next tokens
    - ***Tech support*** *would not fix the problem*
      - [null, null, null, ***Tech_support***, would, not, fix]
    - *The entire **place** is very clean*
      - [null, the, entire, ***place***, is, very, clean]

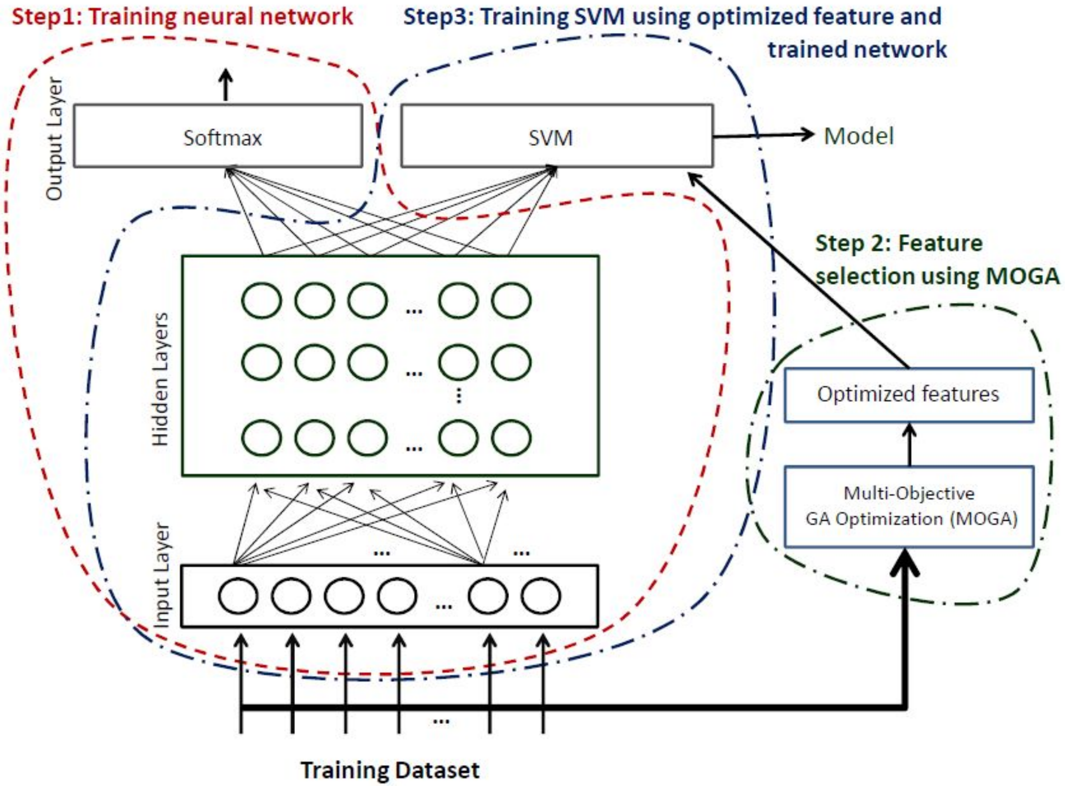
Md Shad Akhtar, Ayush Kumar, Asif Ekbal, Pushpak Bhattacharyya. 2016. A Hybrid Deep Learning Architecture for Sentiment Analysis. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 482–493, Osaka, Japan, December 11-17 2016.

Classification Model

1. Training of a typical convolutional neural network (CNN)
   - Obtain weight matrix

2. A multi-objective GA based optimization technique (NSGA-II) for extracting the optimized set of features
   - Two objectives
     - *Accuracy* (maximize)
     - *Num of features* (minimize)

3. Training of SVM utilizing the network trained in first step and optimized features



Datasets

- **Hindi**
  - Twitter (SAIL - 2015): 1.6K sentences
  - Product/Service reviews: 5.4K sentences
    - Aspect based sentiment analysis
    - Sentence based sentiment analysis
  - Movie reviews: 2.1K sentences
    - Sentence based sentiment analysis

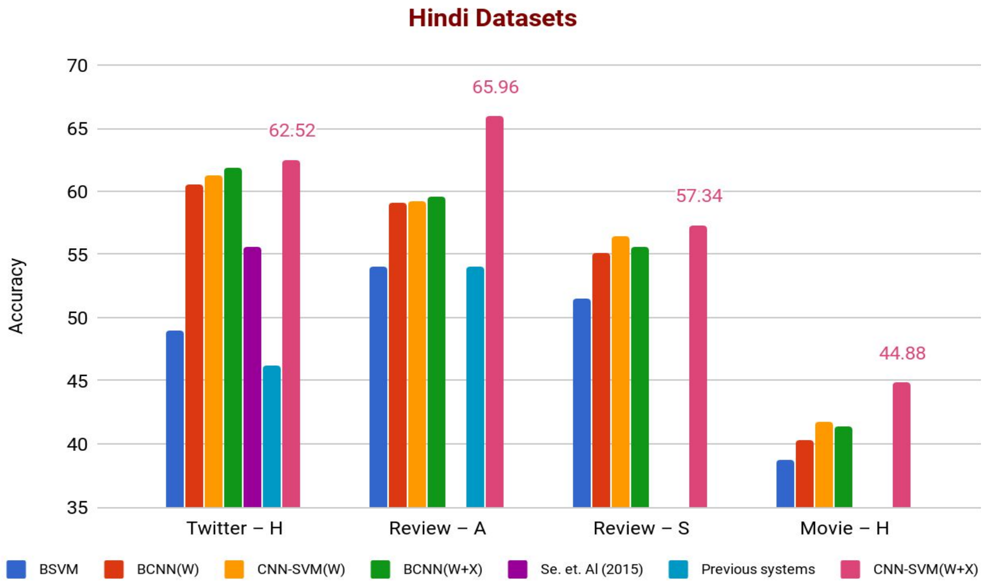
- **English**
  - Twitter (SemEval - 2015): 10.2K sentences
    - Generic tweets
    - Sarcastic tweets
  - Product/Service reviews (SemEval - 2014): 7.6K sentences
    - Aspect based sentiment analysis

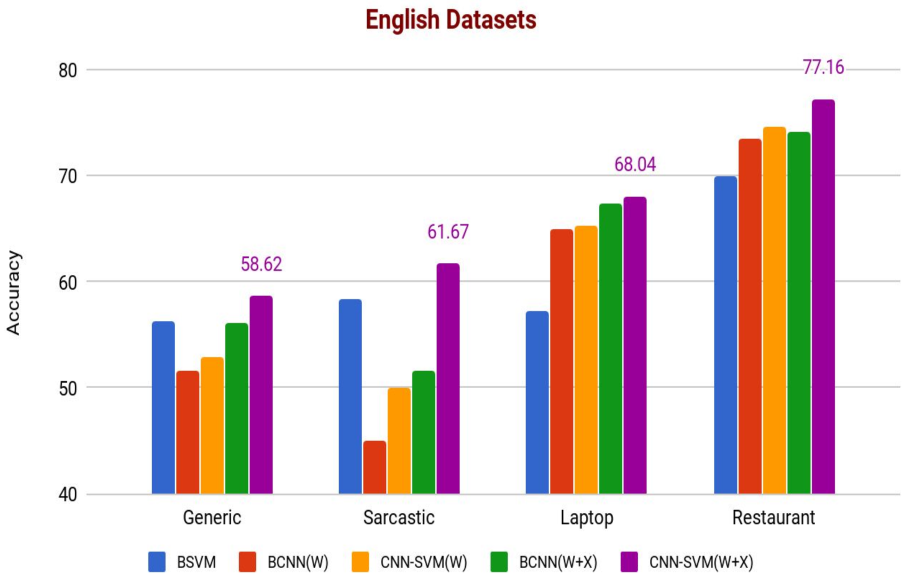
Hindi product and movie reviews datasets are available at: http://www.iitp.ac.in/~ai-nlp-ml/resources.html

Feature set

Language	Dataset	Optimized Features (NSGA-II)
Hindi	Twitter	Emoticons, Punctuation, SentiWordNet
	Review – A, Review – S	Semantic Orientation (SO)
	Movie	Semantic Orientation (SO), SentiWordNet
English	Twitter	Hashtag, Emoticons, Punctuation, BingLiu, NRC
	Review – A	BingLiu, MPQA

Evaluation





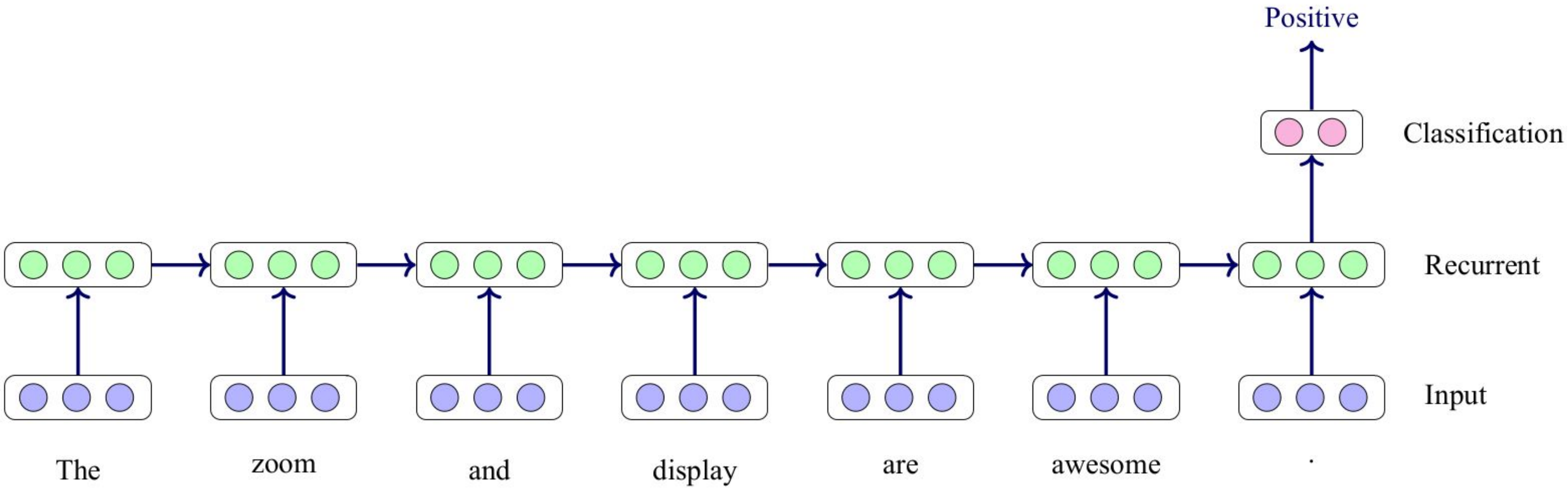
- B_{SVM} : SVM based model
- $B_{CNN(W)}$: CNN based model with word vectors as input
- $B_{CNN(W+X)}$: CNN based model with word vectors and optimized feature set as input.
- ***CNN-SVM***$_{(W)}$: SVM on top of CNN with word vectors as input.
- ***CNN-SVM***$_{(W+X)}$: SVM on top of CNN with word vectors and optimized feature set as input.

Recurrent Neural Network

RNN - Example

Sentiment Classification:

- Given a sentence X, find the associated sentiment polarity of the sentence



RNN/LSTM based Aspect Sentiment Classification

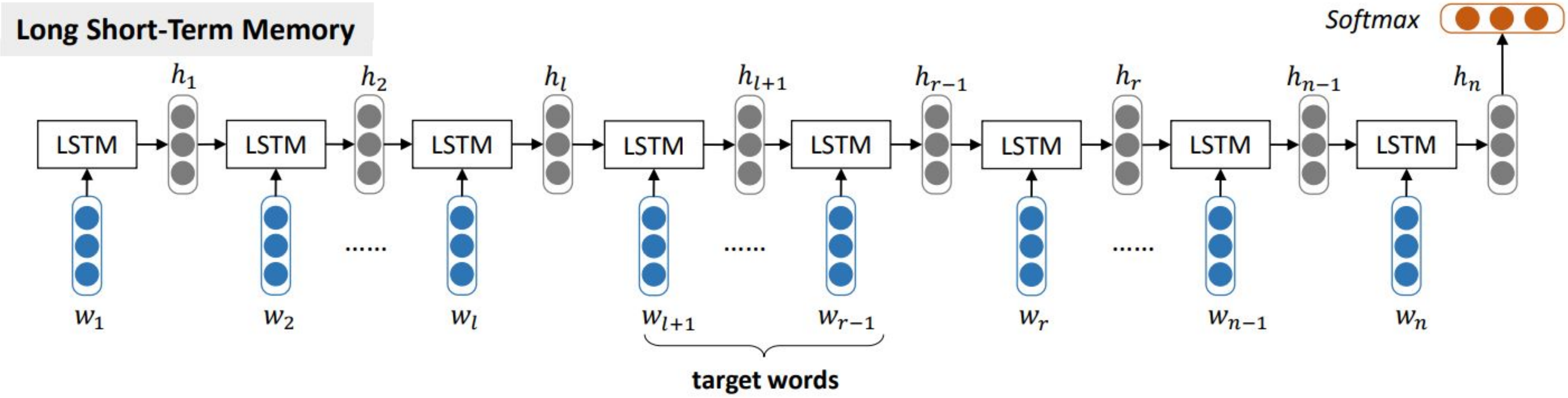
Effective LSTMs for Target-Dependent Sentiment Classification [Tang et al. 2016]

- Long Short-Term Memory (LSTM)
  - Models the semantic representation of a sentence without considering the target word being evaluated
- Target-Dependent Long Short-Term Memory (TD-LSTM)
  - Extend LSTM by considering the target word
- Target-Connection Long Short-Term Memory (TC-LSTM)
  - Semantic relatedness of target with its context words are incorporated

Duyu Tang, Bing Qin, Xiaocheng Feng, Ting Liu. 2016. Effective LSTMs for Target-Dependent Sentiment Classification. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 3298–3307, Osaka, Japan, December 11-17 2016.

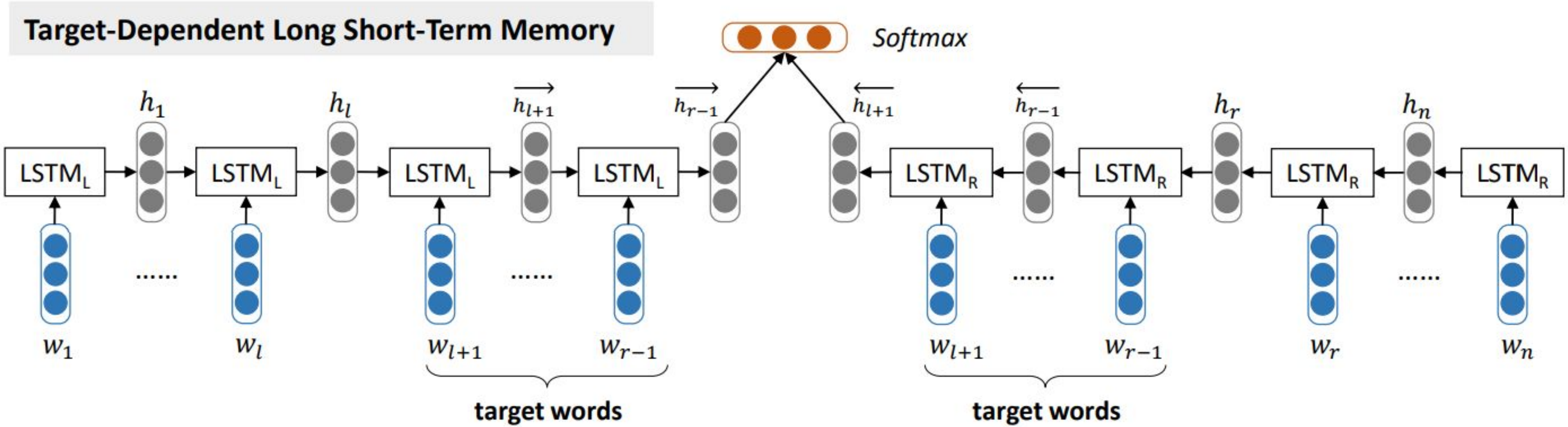
Simple LSTM [Tang et al. 2016]

- Models the semantic representation of a sentence without considering the target word being evaluated
  - No discrimination between the following two instances
    - Its ***battery*** is awesome but camera is poor.
    - Its battery is awesome but ***camera*** is poor.



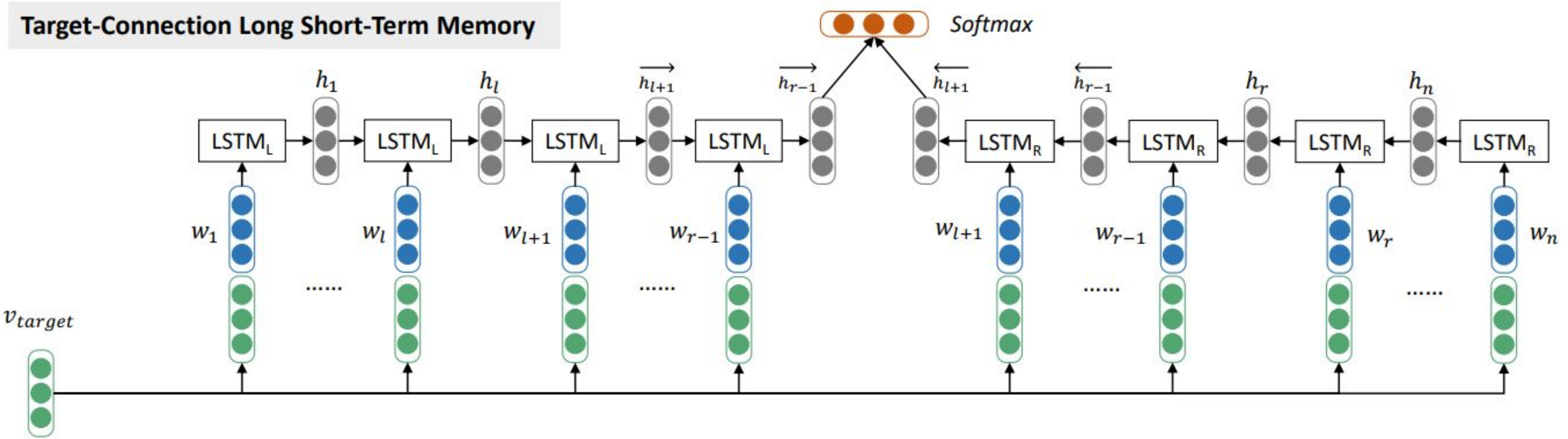
Target-Dependent Long Short-Term Memory (TD-LSTM) [Tang et al. 2016]

- Considers the target word
  - Its ***battery*** is awesome but camera is poor.
    - $LSTM_L$(Its ***battery***) + $LSTM_R$(***battery*** is awesome but camera is poor.)
  - Its battery is awesome but ***camera*** is poor.
    - $LSTM_L$(Its battery is awesome but ***camera***) + $LSTM_R$(***camera*** is poor.)



Target-Connection Long Short-Term Memory (TC-LSTM) [Tang et al. 2016]

- Relationship between the word and the target is incorporated
  - Its ***battery*** is awesome but camera is poor.
    - LSTM_{L1}(Its, ***battery***) → LSTM_{L2}(battery, ***battery***)
    - LSTM_{R7}(battery, ***battery***) ← LSTM_{R6}(is, ***battery***) ← LSTM_{R5}(awesome, ***battery***) ← LSTM_{R4}(but, ***battery***) ← LSTM_{R3}(camera, ***battery***) ← LSTM_{R2}(is, ***battery***) ← LSTM_{R1}(poor, ***battery***)



Experiments

- Dataset
  - Dong et al., 2014
    - Train: 6,248 sentences
    - Test: 692 sentences
    - Sentiment distribution: 25% → Positive, 25% → Negative, 50% → Neutral

Method	Accuracy	Macro-F1
LSTM	0.665	0.647
TD-LSTM	0.708	0.690
TC-LSTM	0.715	0.695

Solving Data Sparsity for Aspect based Sentiment Analysis using Cross-linguality and Multi-linguality [Akhtar et al. 2018]

- Low-resource languages usually suffer in performance due to the *non-availability of sufficient* training data instances
- Low-resource languages (e.g. Hindi, Bengali etc.) usually suffer due to the non-availability of sufficient data instances

- **Problem:** Data Sparsity in word representation (i.e. *absence of representation of a word*) is another problem
- Out-of-vocabulary (OOV) words in a word embedding model pose a serious challenge to the underlying learning algorithm

Shad Akhtar, Palaash Sawant, Sukanta Sen, Asif Ekbal, and Pushpak Bhattacharyya (2018). Solving Data Sparsity for Aspect based Sentiment Analysis using Cross-linguality and Multi-linguality. In Proceedings of the 16th Annual Conference of the NAACL:HLT-2018, June 2018, New Orleans, LAUSA, pages 572–582.

Solution to the OOV problem

- Solution: Minimize the effect of data sparsity problem in a resource-scarce language scenario by leveraging the information of resource-rich languages

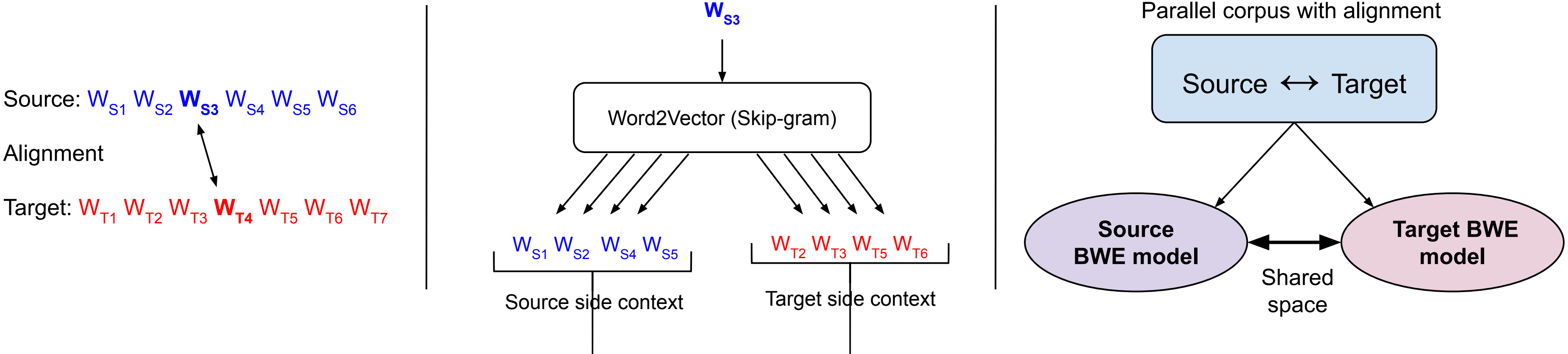
  - **How?**
    - Word embedding space of two languages may not be same
    - Therefore, cannot use the two embeddings in the similar context

  - Project the embeddings of two languages into a shared space
    - Bi-lingual embeddings (Luong et al., 2015)

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015, ***Bilingual Word Representations with Monolingual Quality in Mind***. In *NAACL Workshop on Vector Space Modeling for NLP*.

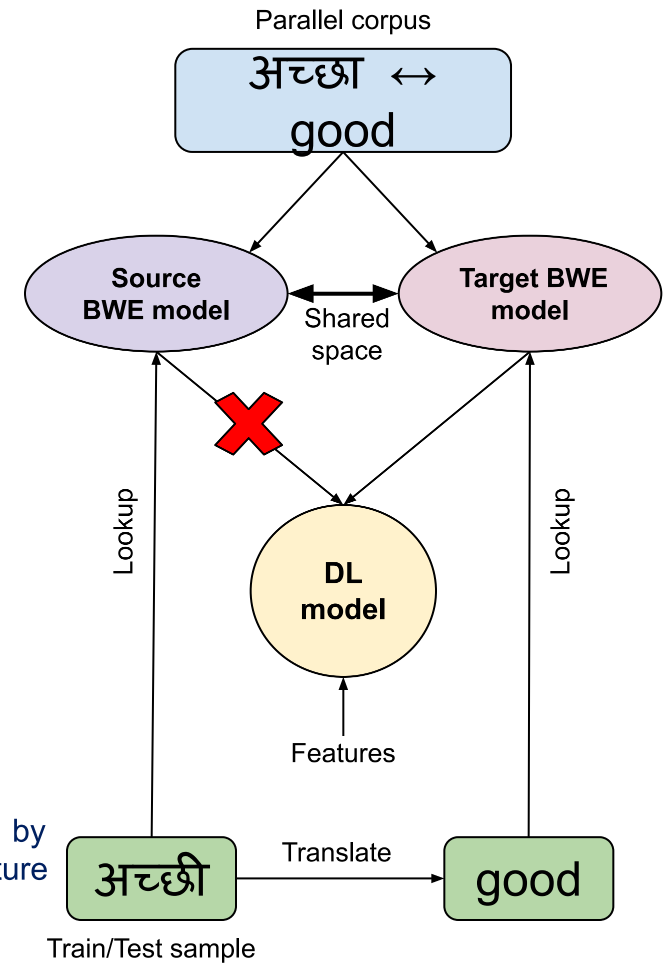
Bi-lingual Word Embeddings (BWE): (Loung et al., 2015)

- **Bi-lingual word embeddings aims to *bridge the language divergence* in the vector space**
  - Requires a *parallel corpus* and *alignment information* among parallel sentences
  - Utilize existing word2vec skip-gram model (Mikolov et al., 2013)
  - For each word, the authors defined its context to include the neighbouring words from both the source and target languages



Proposed Approach

- Utilize bilingual word embeddings for a pair of languages (e.g., Hindi-English, French-English)
- Extract word representations for all the words in a sentence from the source (e.g., Hindi) bilingual word embedding
- For all the OOV words, translate into the target language, and perform another lookup in target bilingual embedding
- Spelling varation: Two differently spelled words in Hindi such as *‘किबनशन | kambineshana’* and *‘कंबीनशन | kaMbIneshana’* translate to an English word “combination”



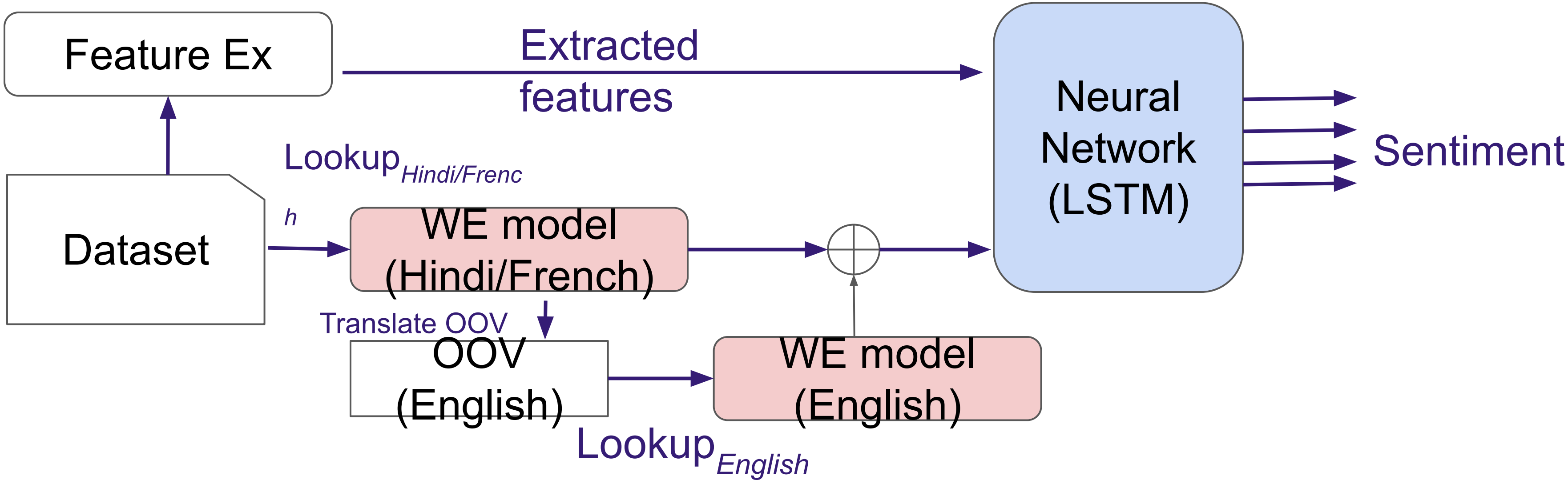
- Further, leverage the effectiveness of English side resources by translating a word into English and then extracting its feature representation
  - Bing Liu, MPQA, SentiWordNet and Semantic Orientation

Two setups

- Multi-lingual Setup
  - ***Train*** *and* ***Test*** *on* ***Source*** *language (i.e., Hindi or French)*
  - Utilize bi-lingual embeddings for OOV words
  - Utilize English-side lexicons for the feature extraction

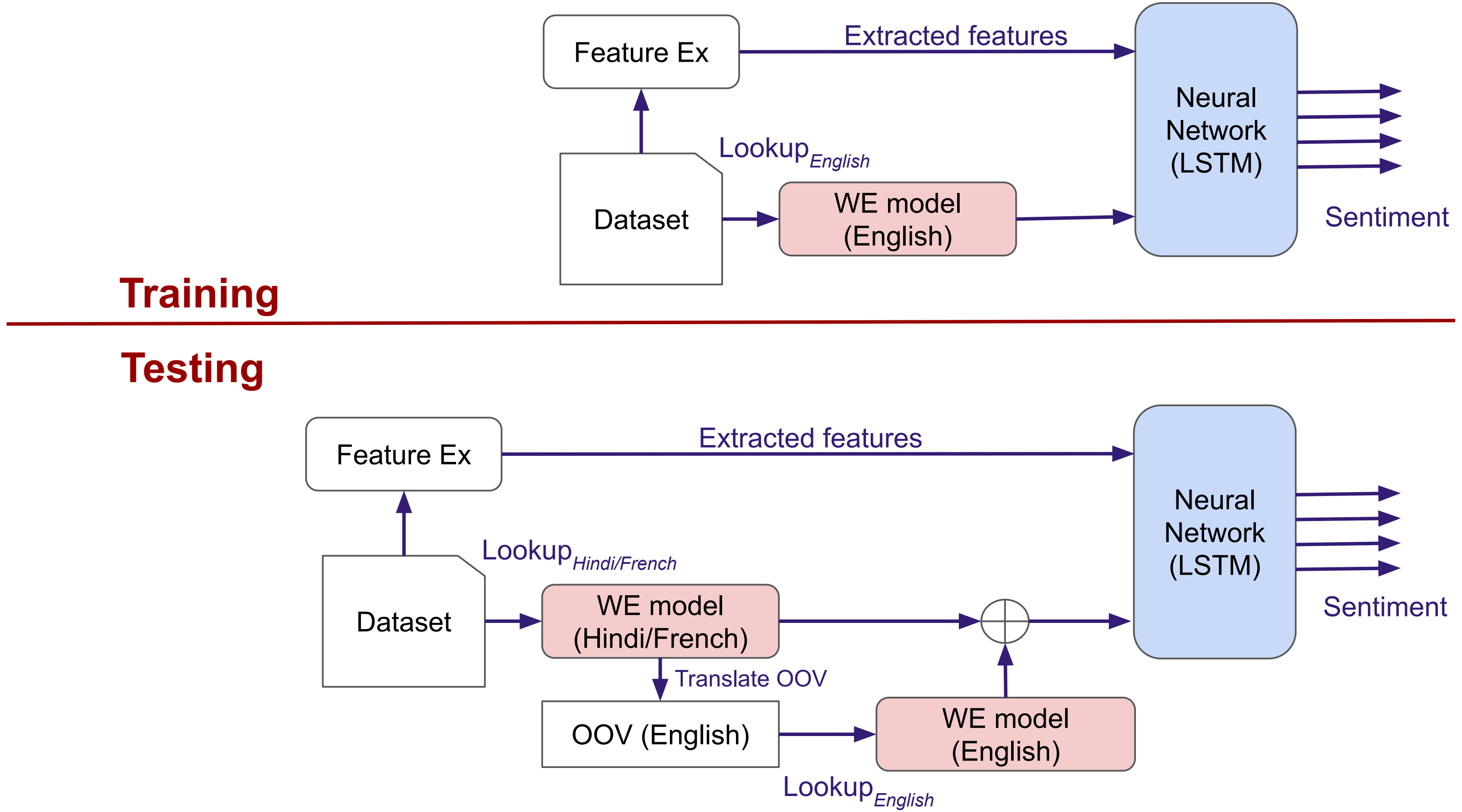
- Cross-lingual Setup
  - ***Train*** *on* ***Target*** *language (i.e., English) and* ***Test*** *on* ***Source*** *language (i.e., Hindi or French)*
  - Utilize bi-lingual embeddings for OOV words
  - Utilize English-side lexicons for the feature extraction

Multi-lingual Setup



Training and Testing scenarios

Cross-lingual Setup



Hybrid Architecture

Three architectures based on the position of the fusion of hand-crafted features

A1. Early fusion:

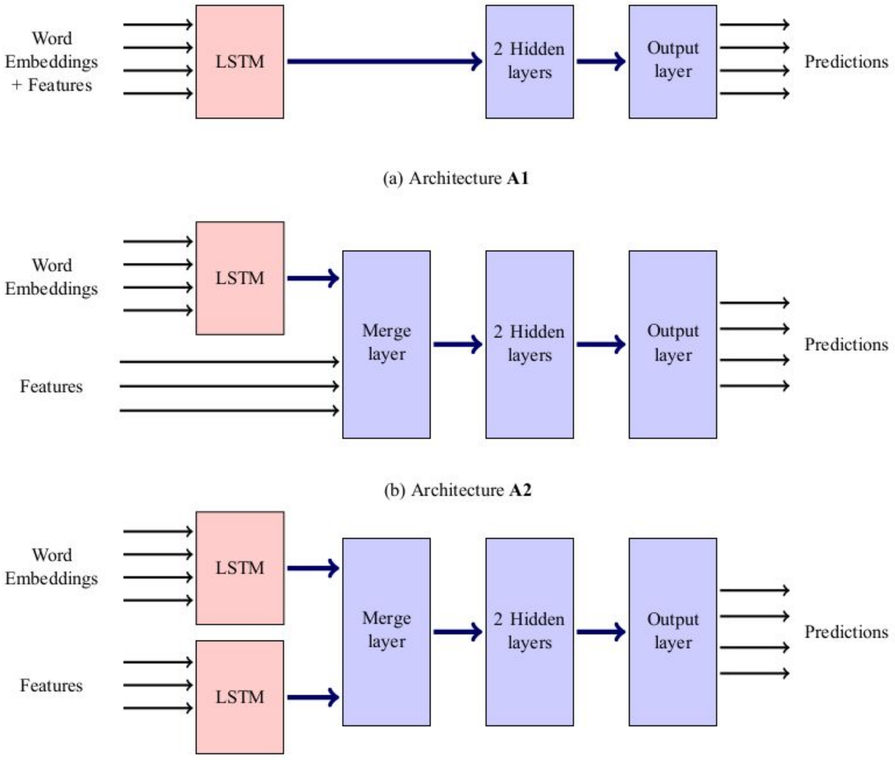
- LSTM(*WE* + *Feat*)

A2. Delayed fusion:

- LSTM(*WE*) + *Feat*

A3. Delayed fusion with sequential feature representation

- LSTM(*WE*) + LSTM(*Feat*)



(a) Architecture **A1**

(b) Architecture **A2**

(c) Architecture **A3**

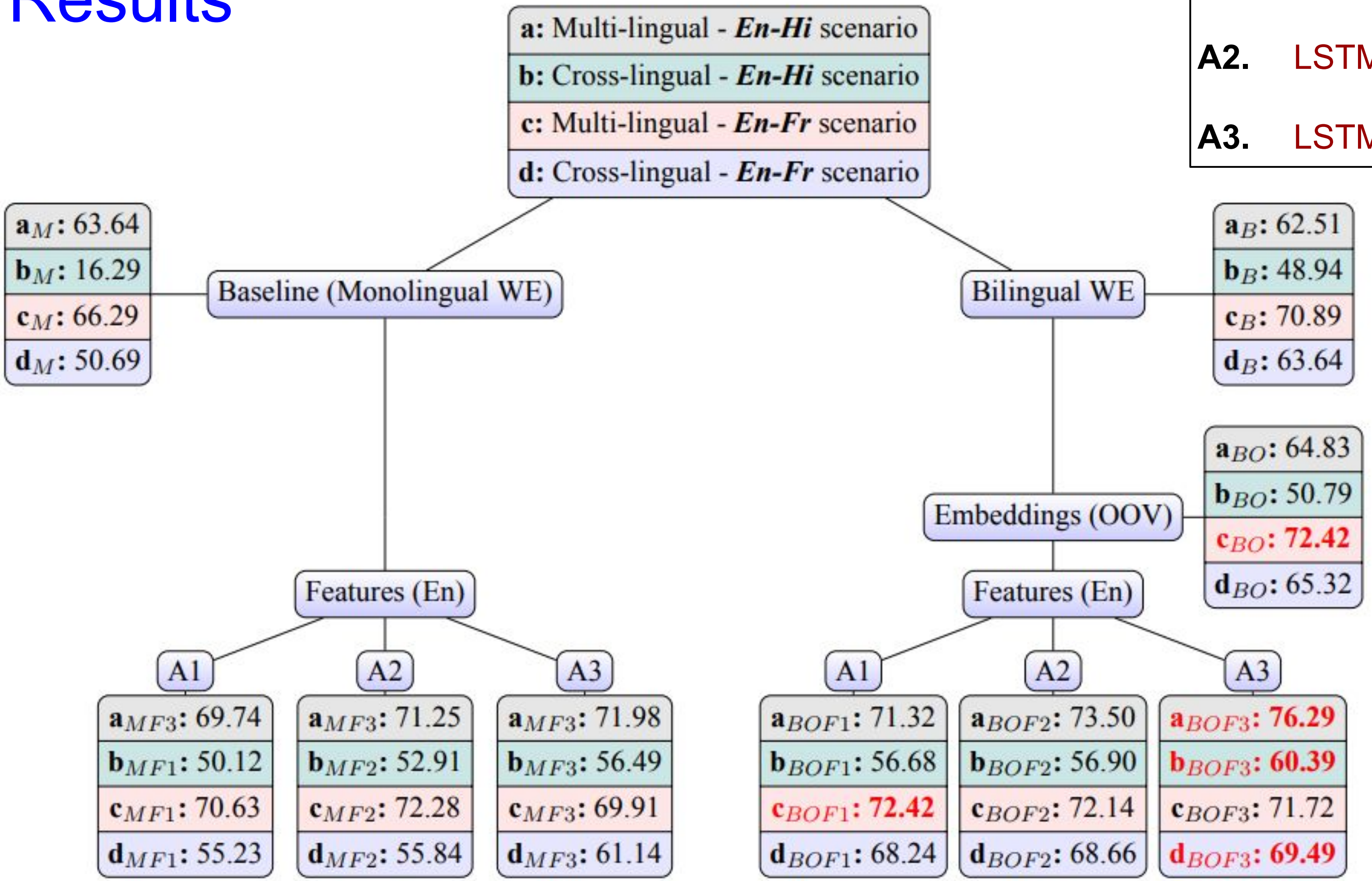
Dataset and Experimental setups

- Aspect Based Sentiment Analysis

Language pairs	Datasets	Review Sentences	Aspect terms
English - Hindi	**English - SemEval 2014** (Pontiki et al., 2014)	3845	3012
	Hindi (Akhtar et al., 2016)	5417	4509
English - French	**English - SemEval 2016** (Pontiki et al., 2016)	3365	2676
	French - SemEval 2016 (Pontiki et al., 2016)	2429	3482

- **Setups**
  - Multi-lingual Setup
    - *Train and Test on Source language (i.e., Hindi or French)*
    - Utilize English-side lexicons for the feature extraction
  - Cross-lingual Setup
    - *Train on Target language (i.e., English) and Test on Source language (i.e., Hindi or French)*
    - Utilize English-side lexicons for the feature extraction

Results



A1. LSTM(WE+Feat)

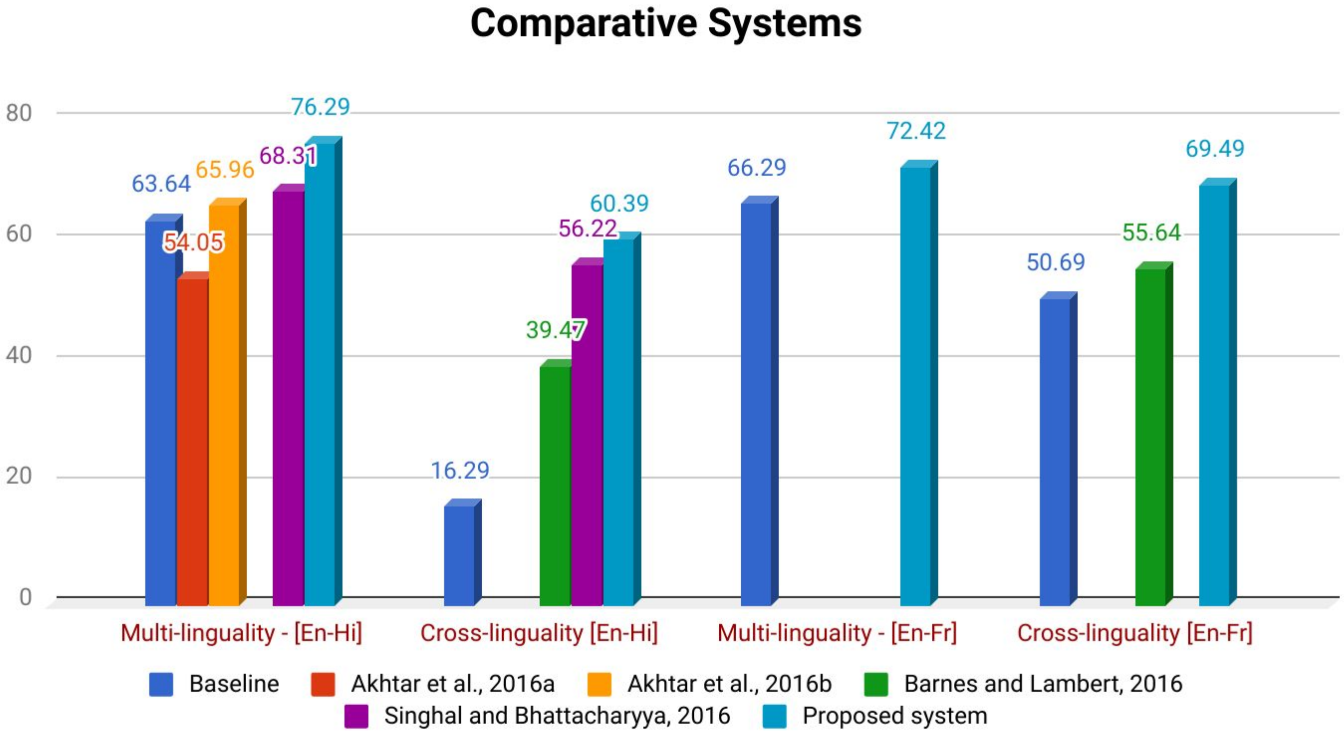
A2. LSTM(WE) + Feat

A3. LSTM(WE) + LSTM(Feat)

*Accuracy values

F: Feature
1,2,3: Architecture

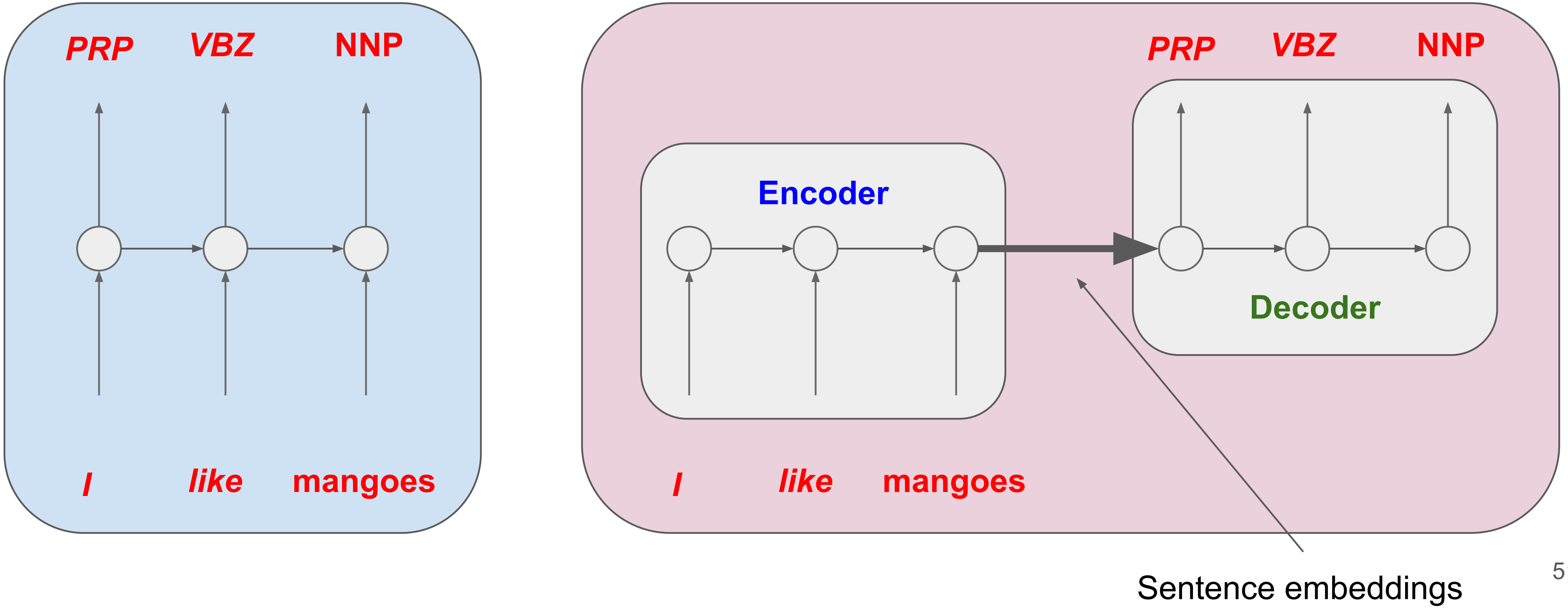
Comparative Analysis



Attention Mechanism

Sequence labeling v/s Sequence transformation

- PoS Tagging



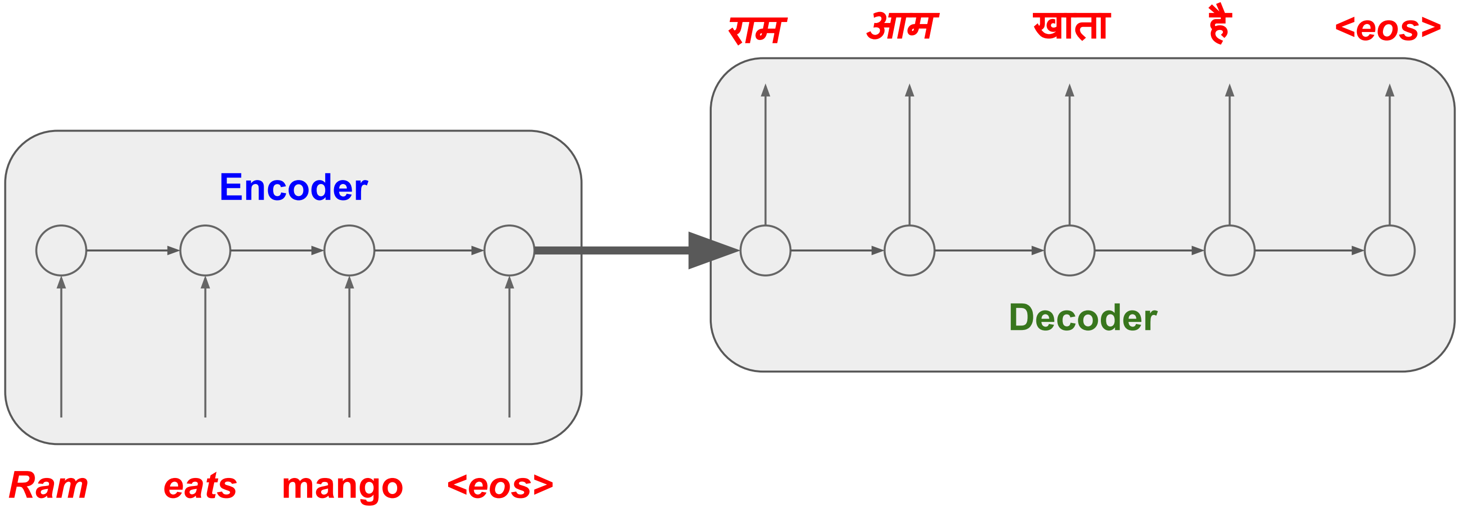
Why is sequence transformation required?

- For many application length of i/p and o/p are not necessarily the same
  - E.g. *Machine Translation, Summarization, Question Answering* etc.

- For many applications length of o/p is not known
- Non-monotone mapping: Reordering of words
- PoS tagging, Named Entity Recognition etc. do not require these capabilities

Encode-Decode paradigm

- English-Hindi Machine Translation
  - Source sentence: 3 words
  - Target sentence: 4 words
  - Second word of the source sentence maps to 3rd & 4th words of the target sentence
  - Third word of the source sentence maps to 2nd word of the target sentence



Problems with Encode-Decode paradigm

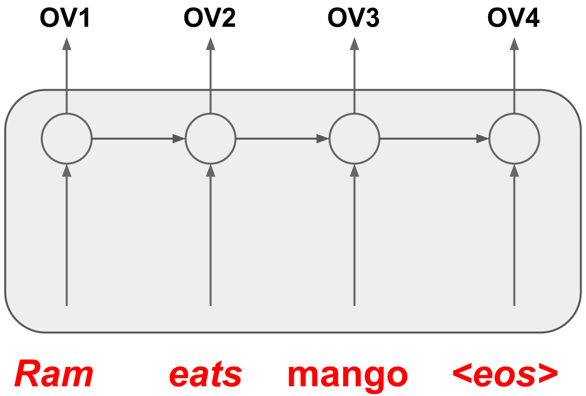
- Encoding transforms the entire sentence into a single vector
- Decoding process uses this sentence representation for predicting the output
  - Quality of prediction depends upon the quality of sentence embeddings
- After few time steps decoding process may not properly use the sentence representation due to long-term dependency

Solutions

- To improve the quality of predictions we can
  - Improve the quality of sentence embeddings **‘OR’**
  - Present the source sentence representation for prediction at each time step **‘OR’**
  - Present the **RELEVANT** source sentence representation for prediction at each time step
    - *Encode* - *Attend* - *Decode* (***Attention mechanism***)

Attention Mechanism

- Represent the source sentence by the set of **output vectors** from the encoder
- Each **output vector** (OV) at time *t* is a contextual representation of the input at time *t*

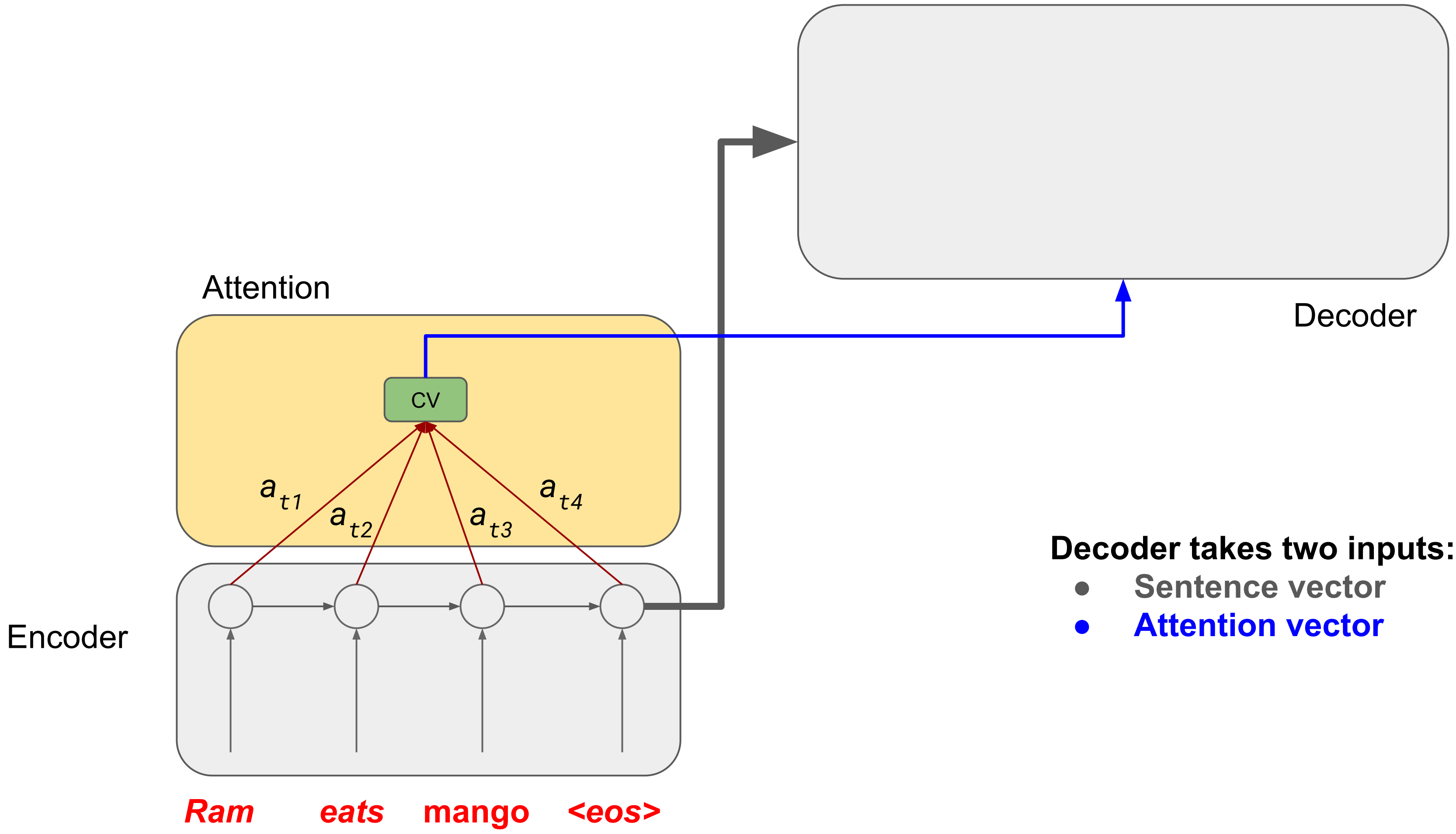


Attention Mechanism

- Each of these output vectors (OVs) may not be equally relevant during decoding process at time *t*
- Weighted average of the output vectors can resolve the relevancy
  - Assign more weights to an output vector that needs more ***attention*** during decoding at time *t*
- The weighted average ***context vector (CV)*** will be the input to decoder along with the sentence representation
  - $CV_i = \sum a_{ij} \cdot OV_j$

where a_{ij} = `weight of the` j^{th} `OV`

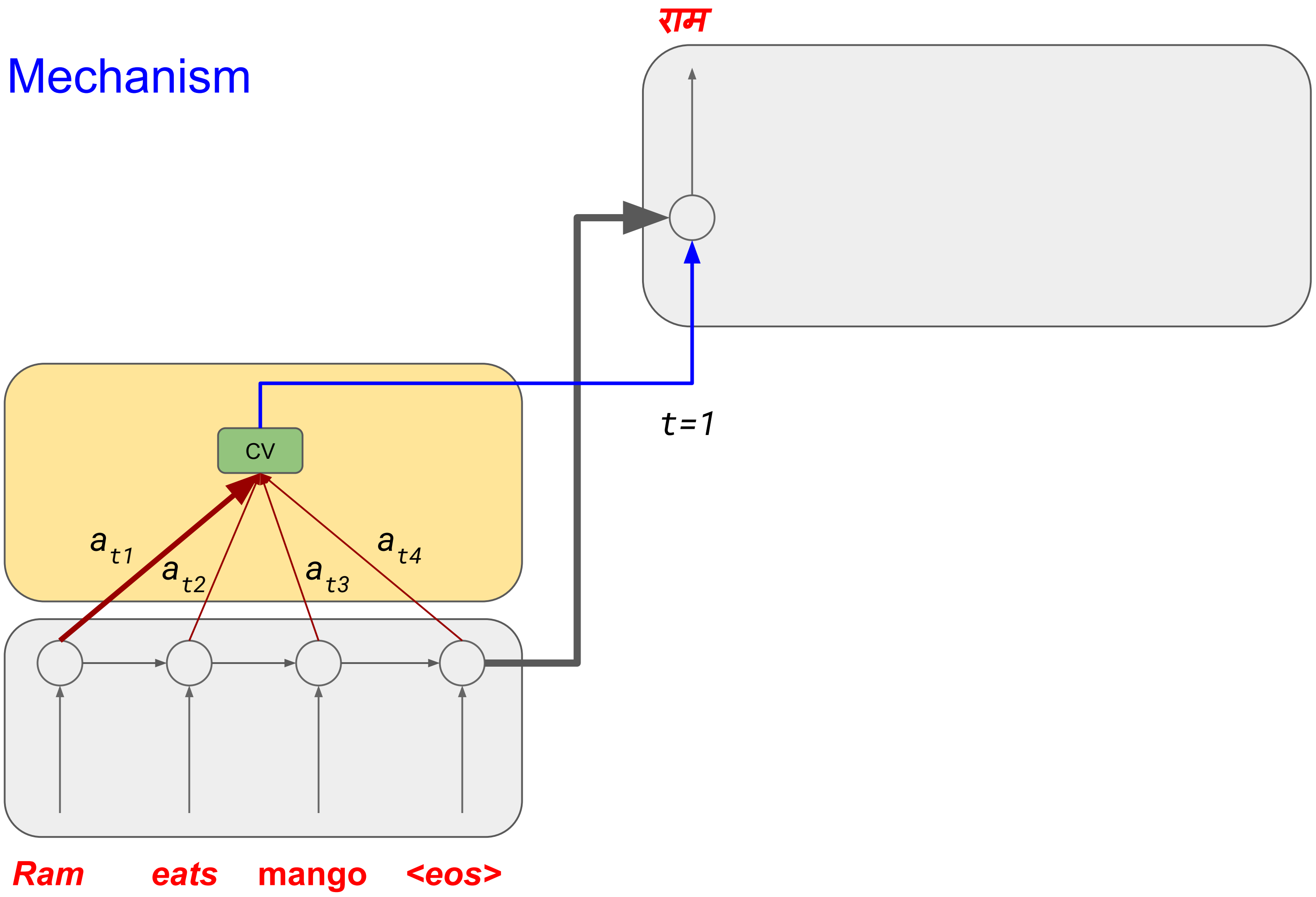
Attention Mechanism



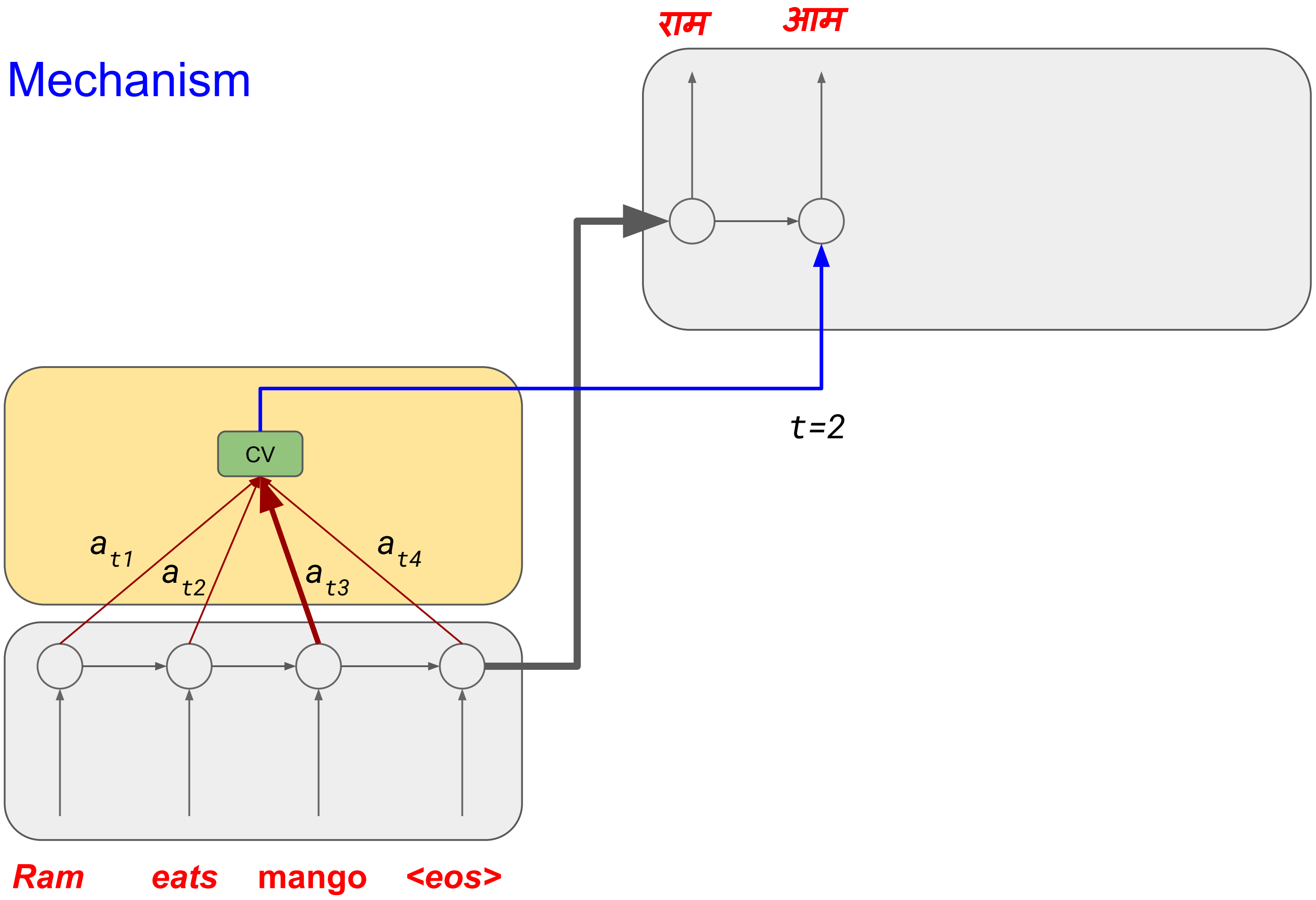
Decoder takes two inputs:

- **Sentence vector**
- **Attention vector**

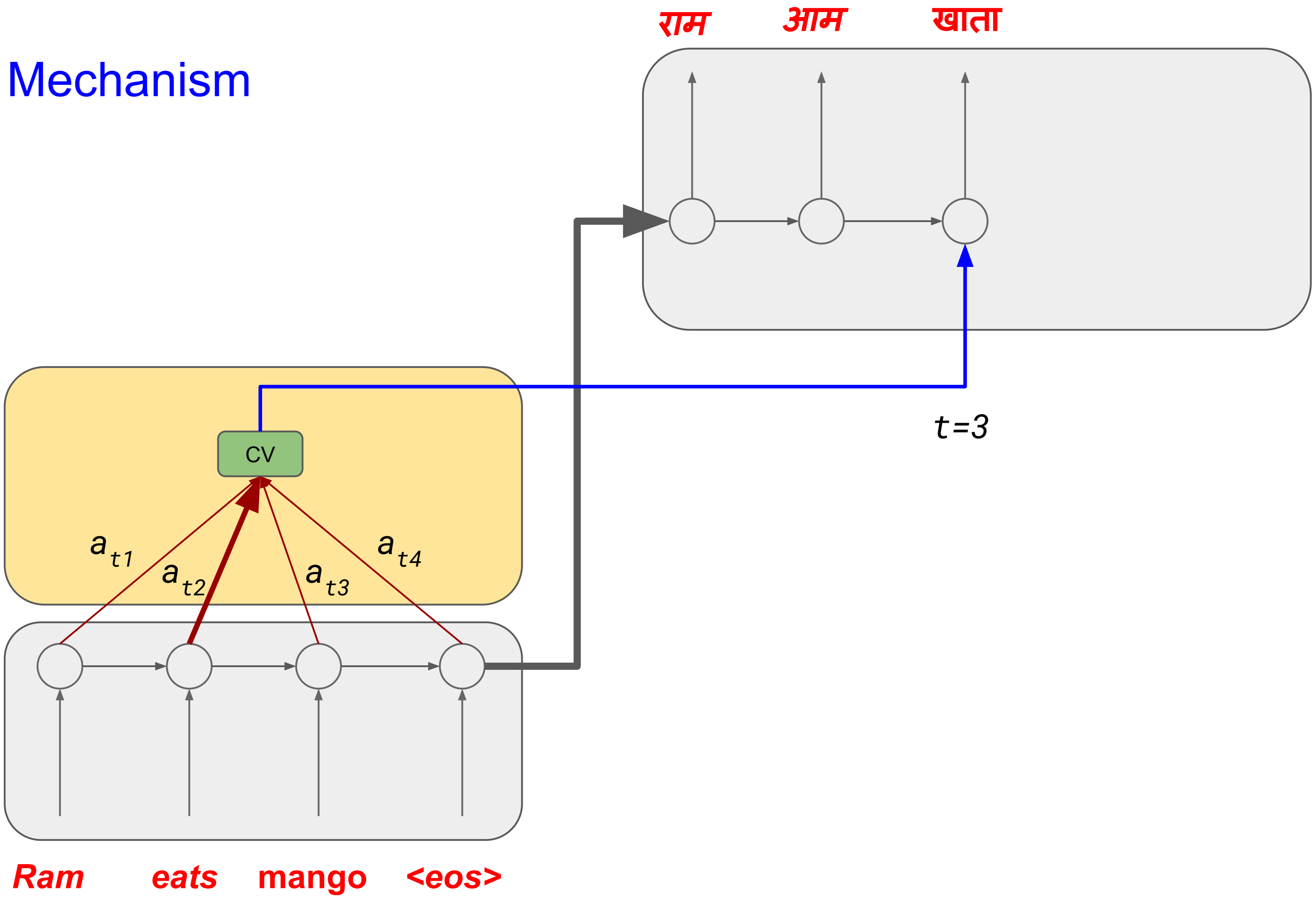
Attention Mechanism



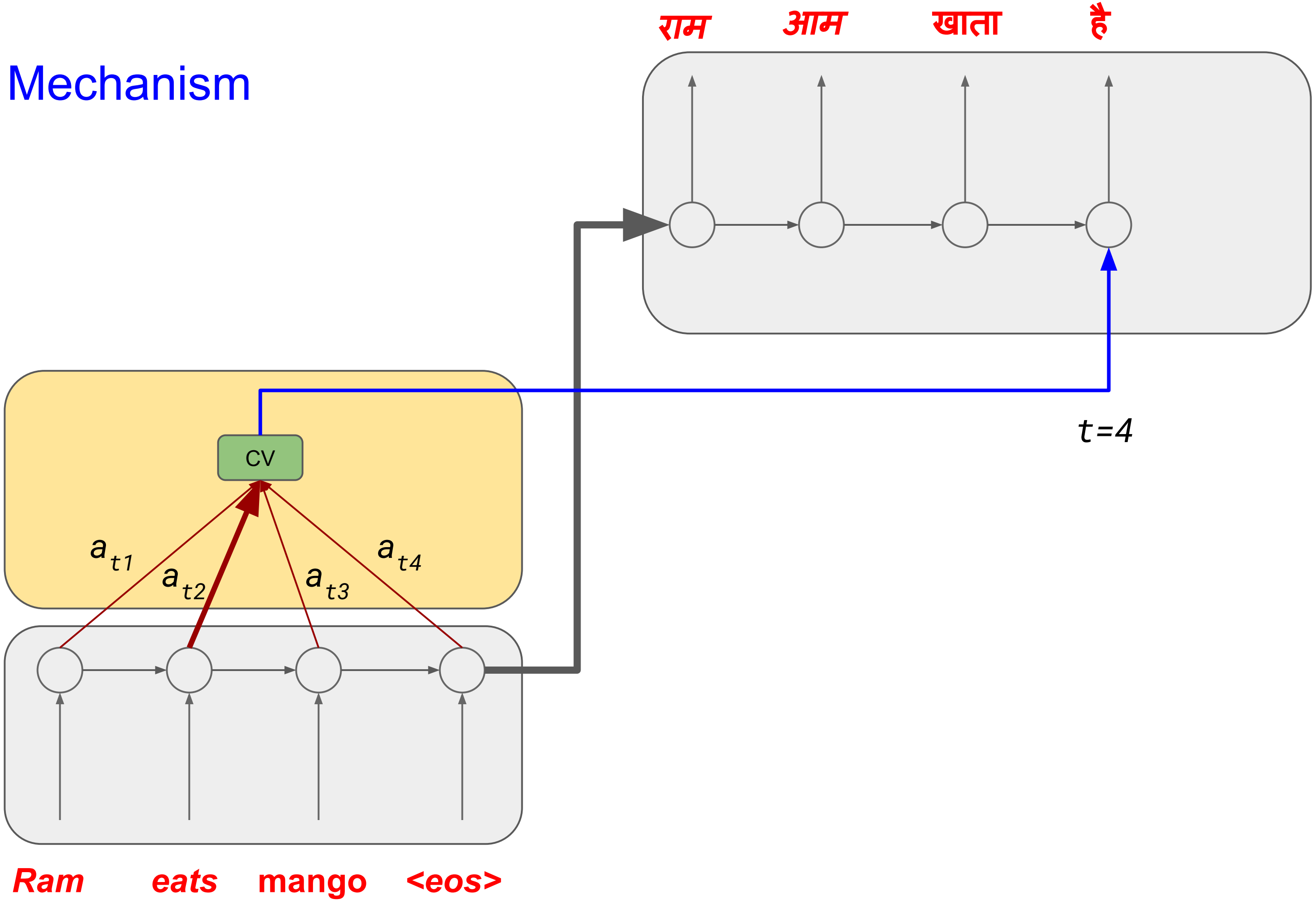
Attention Mechanism



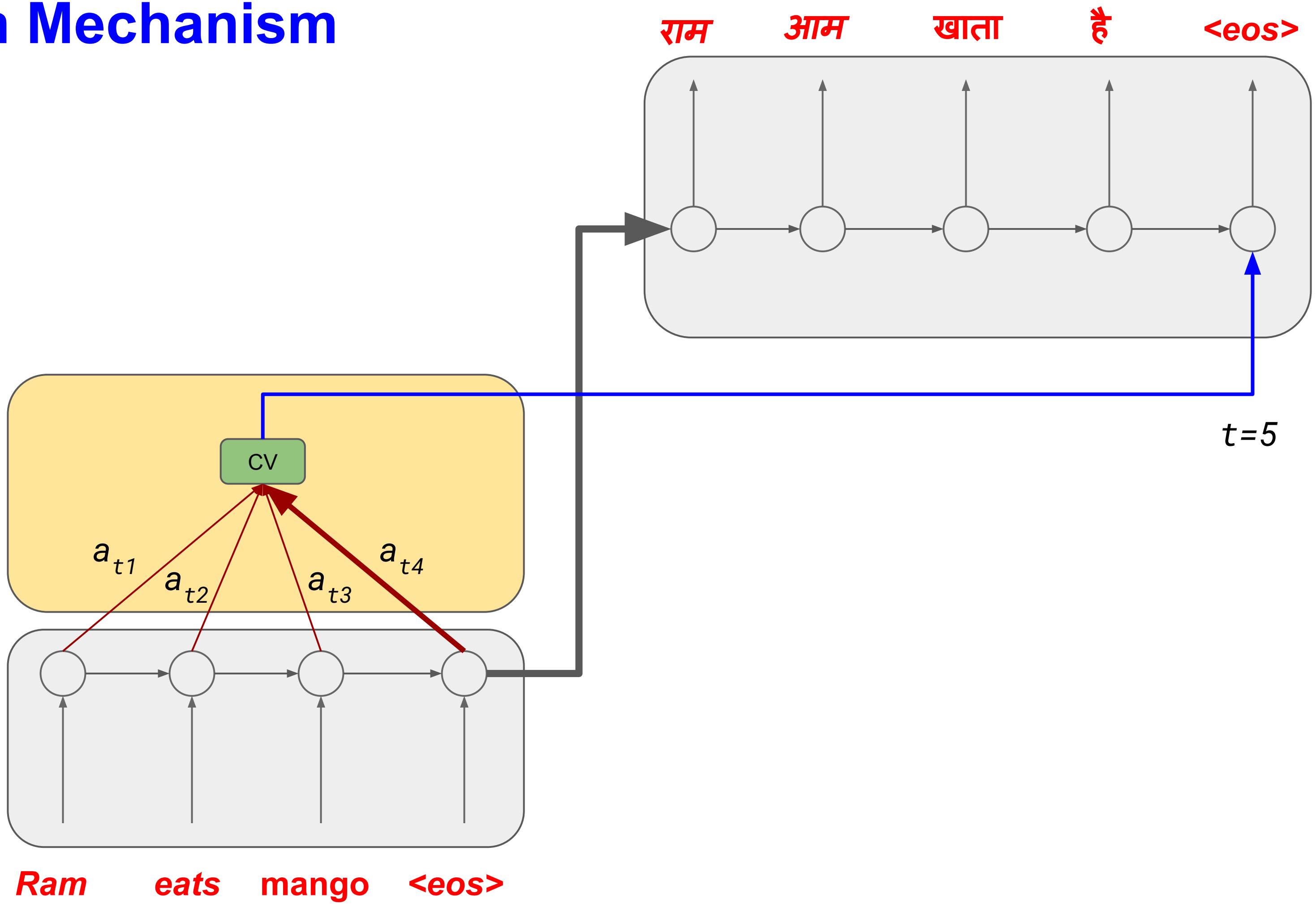
Attention Mechanism



Attention Mechanism



Attention Mechanism



Attention Mechanism for Classification

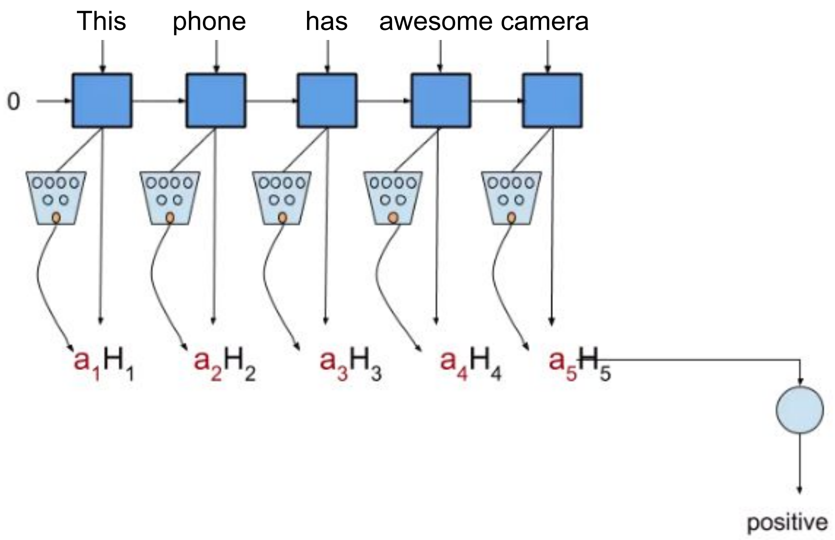
Attention mechanism for classification

- Every word in a sentence is not equally important for any task
  - Sentiment Classification: *Adjectives* are more important than prepositions or conjunctions
    - This phone has awesome camera. → Word '*awesome*' is the most important word in the whole sentence considering the *positive* sentiment
- Why not weight each word in a sentence according to its importance?
- Attention mechanism is the solution
  - Compute attention weights (a_i) by building a small fully-connected neural network on top of each encoded state
  - A single-unit final layer corresponds to the attention weight

$$y_i = tanh(W.\ H_i)$$

$$a_i = \exp(y_i) \,/\, \Sigma_j \exp(y_j)$$

$$h_i = a_i.\ H_i$$



Attention for Aspect Sentiment Classification

- Attend the important word considering the target
  - Its ***battery*** is awesome but ***camera*** is poor
    - For target ***battery,*** awesome will have highest weight
    - For target ***camera,*** poor will have highest weight

- Through attention mechanism, the network can learn the association of *awesome* for ***battery*** and *poor* for ***camera*** in aspect sentiment classification

Attention-based LSTM for Aspect-level Sentiment Classification [Wang et al. 2016]

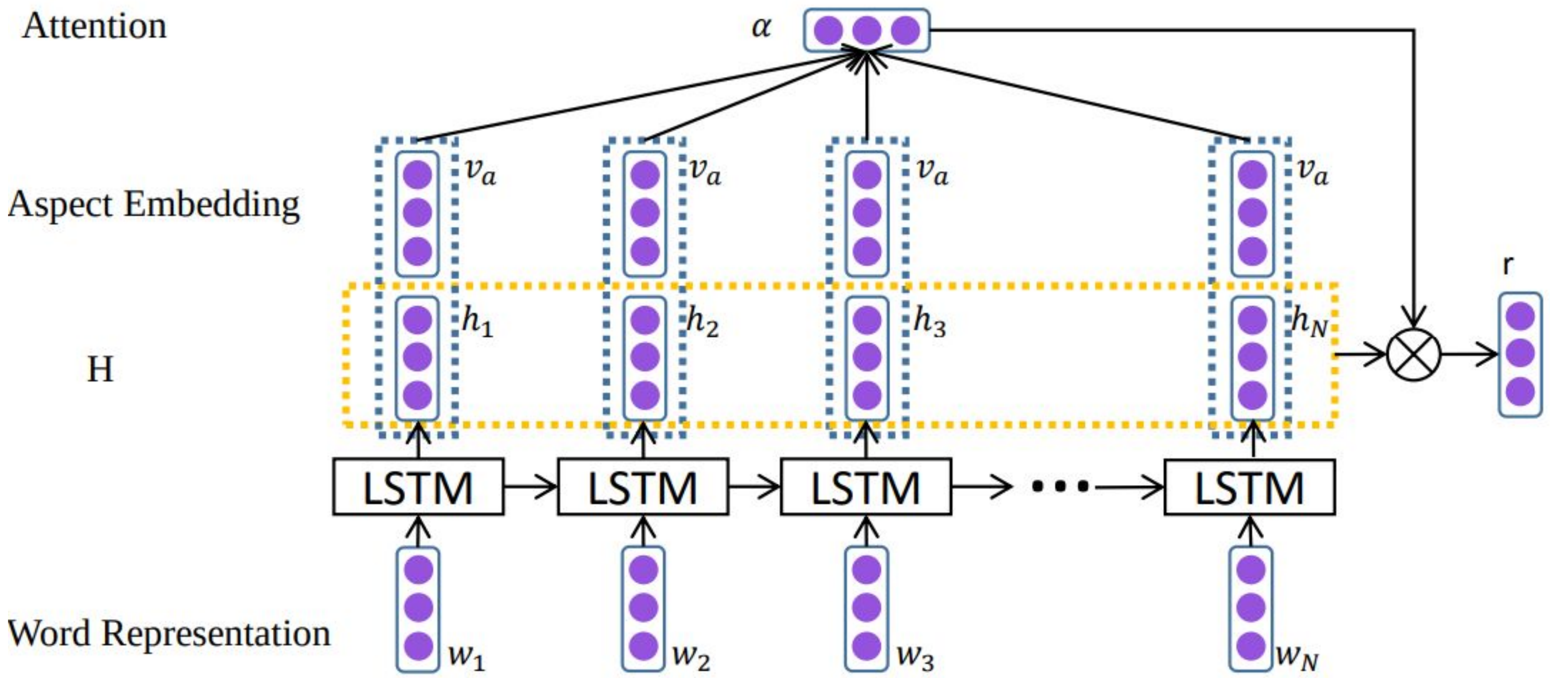
- Incorporation of only target information is not sufficient
- Application of attention mechanism can extract the association of important word for an aspect
- **Two architectures**
  - Attention-based LSTM (AT-LSTM)
    - Relationship between the word and the target is incorporated at the attention layer
  - Attention-based LSTM with Aspect Embedding (ATAE-LSTM)
    - Relationship between the word and the target is incorporated at the input and attention layer both

Yequan Wang, Minlie Huang, Li Zhao and Xiaoyan Zhu. 2016. Attention-based LSTM for Aspect-level Sentiment Classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 606–615, Austin, Texas, November 1-5, 2016.

Attention-based LSTM (AT-LSTM)

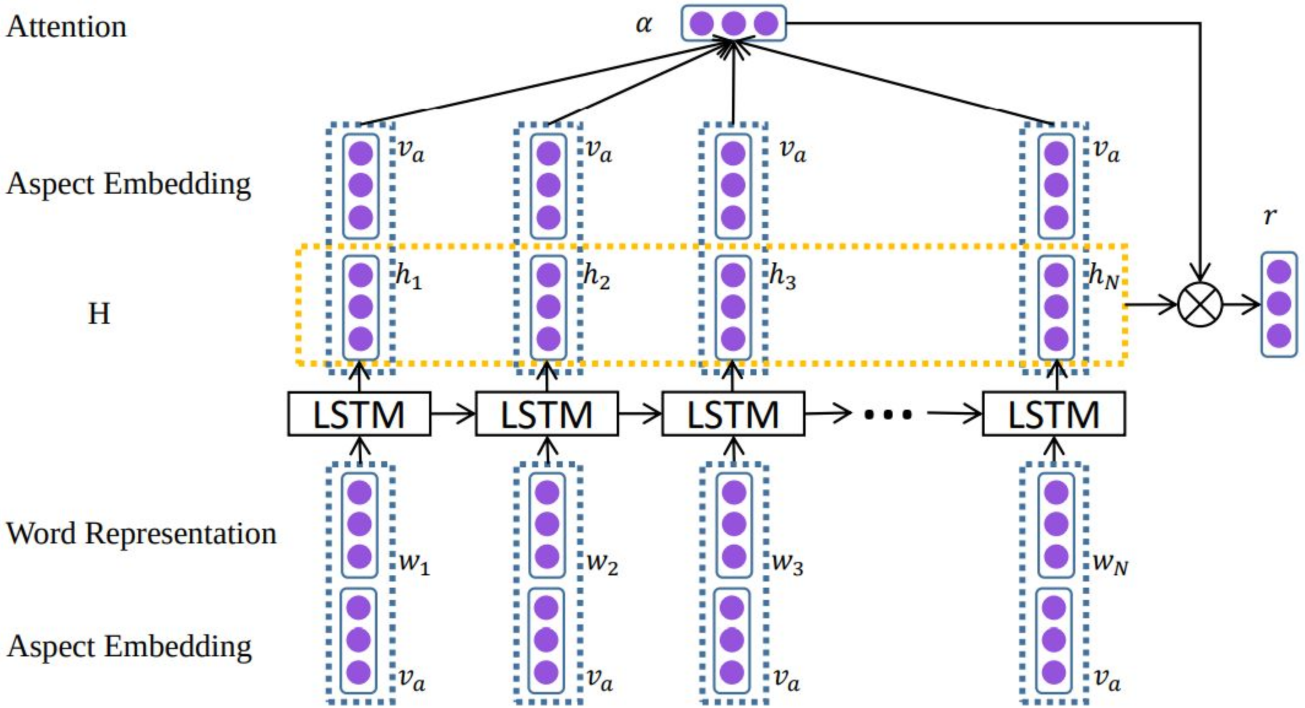
Aspect a = ***battery life***
Aspect Embedding v_a = (emb$_{battery}$ + emb$_{\text{life}}$) / 2

- Relationship between the word and the target is incorporated at the attention layer



Attention-based LSTM with Aspect Embedding (ATAE-LSTM)

- Relationship between the word and the target is incorporated at the input layer and the attention layer



Datasets

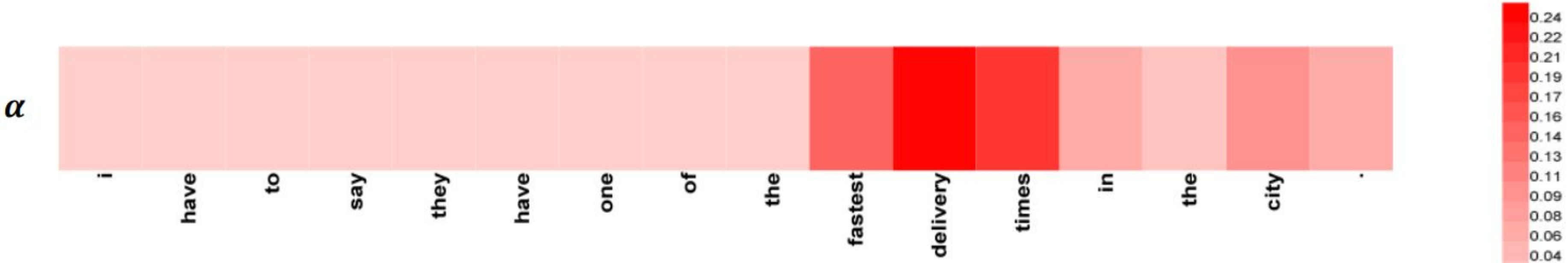
- SemEval-2014 [Pontiki et al., 2014]

Aspect	Positive		Negative		Neutral	
	Train	Test	Train	Test	Train	Test
Food	867	302	209	69	90	31
Price	179	51	115	28	10	1
Service	324	101	218	63	20	3
Ambience	263	76	98	21	23	8
Misc	546	127	199	41	357	51
Total	2179	657	839	222	500	94

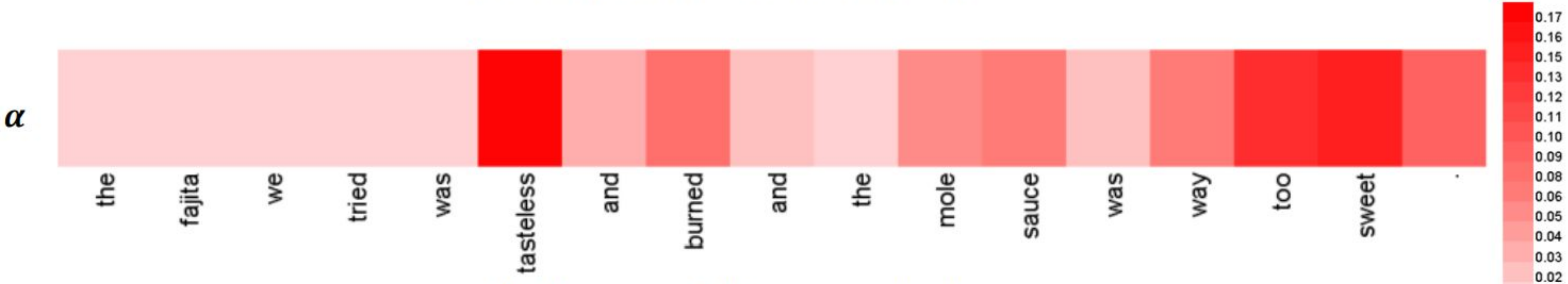
Experimental Results

Method	Pos/Neg/Neu	Pos/Neg
LSTM	82.0	88.3
TD-LSTM	82.6	89.1
TC-LSTM	81.9	89.2
AT-LSTM	83.1	89.6
ATAE-LSTM	84.0	89.9

Attention weights: Heatmaps



(a) the aspect of this sentence: *service*



(b) the aspect of this sentence: *food*

Aspect-level Sentiment Classification with HEAT (HiErarchical ATtention) Network [Cheng et al. 2017]

- **Introduced HiErarchical ATtention (HEAT) network**
  - Aspect attention (*with respect to the aspect category)*
  - Sentiment attention (*with respect to the aspect target*)
- **Aspect attention**
  - pays attention to the aspect information, i.e., aspect terms, under the direction of the target aspect category
- **Sentiment attention**
  - aims to capture the sentiment feature of the text under the direction of the target aspect category and the extracted aspect information

Jiajun Cheng, Shenglin Zhao, Jiani Zhang, Irwin King, Xin Zhang, Hui Wang. 2017. Aspect-level Sentiment Classification with HEAT (HiErarchical ATtention) Network. In Proceedings of the CIKM’17, November 6-10, 2017, Singapore.

A motivating example

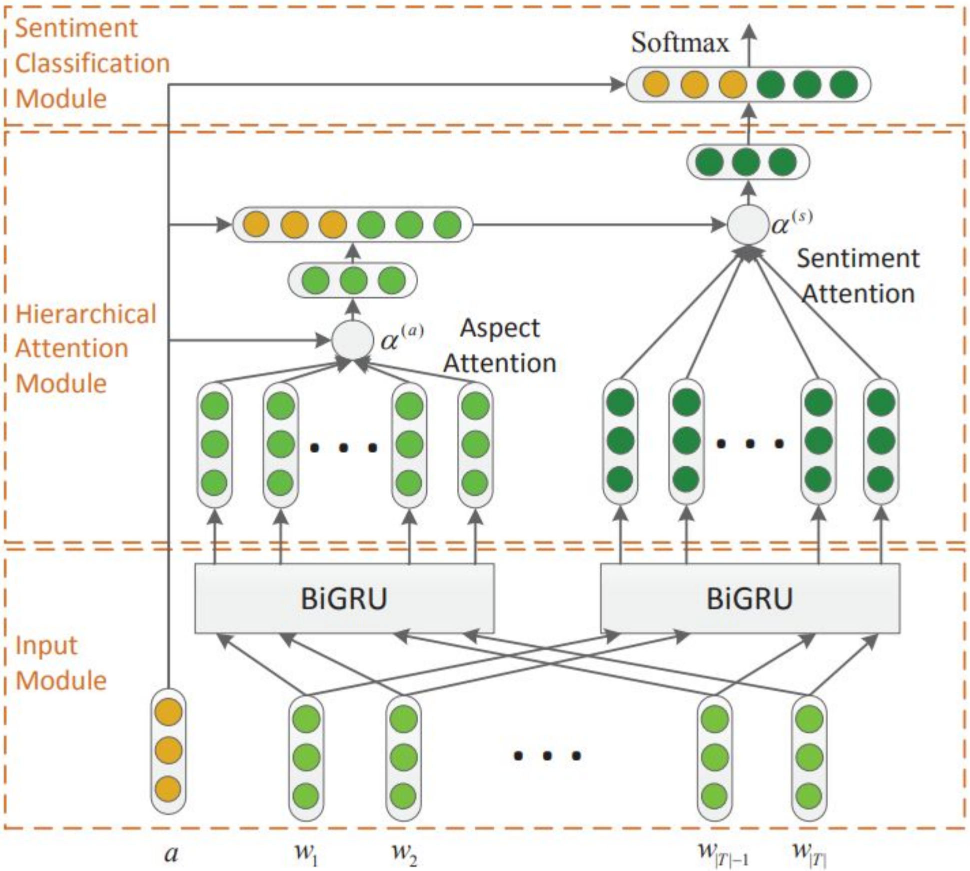
The tastes are great, but the service is dreadful

- Both sentiment-bearing words, ***great*** and ***dreadful*** can be used for both aspect category, ***food*** and ***service***
- Given aspect category *food*, model can attend to both *great* and *dreadful*- *Confusing !*
- **Remedy**
  - Leveraging aspect term to bridge the gap
  - Given aspect category ***food,*** much easier to find aspect term ***tastes*** than to discriminate which sentiment word is corresponding to the aspect (*through aspect attention)*
  - Under the guidance of aspect term ***tastes***, we can easily choose the sentiment word ***great*** and decide the sentiment polarity on the aspect

HEAT for Aspect Sentiment Classification

- Aspect attention aims to pay attention to the aspect information, i.e. aspect terms (***taste***), under the direction of the target aspect (***food***)

- Sentiment attention aims to capture the sentiment feature of the text (***great***) under the direction of the target aspect (***food)*** and the extracted aspect information (***tastes***)



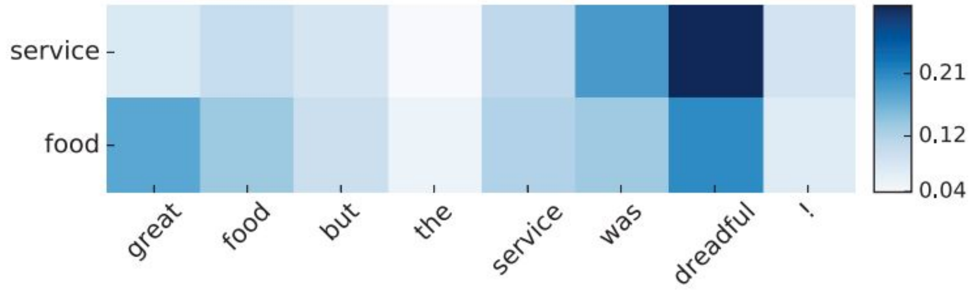
Experiments

- Dataset
  - SemEval-2014 [Pontiki et al., 2014] → Restaurant
  - SemEval-2015 [Pontiki et al., 2015] → Restaurant and Laptop
  - SemEval-2016 [Pontiki et al., 2016] → Restaurant

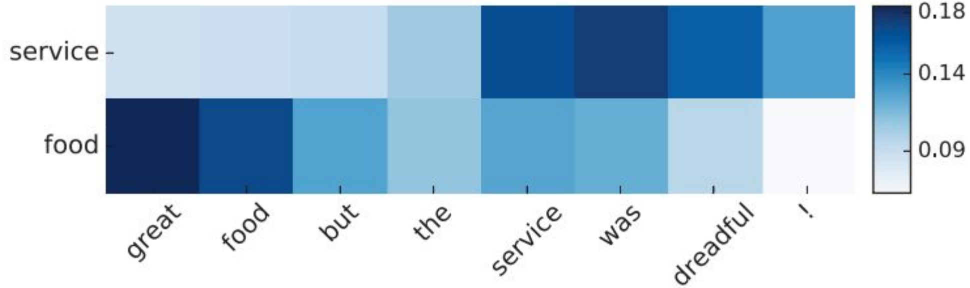
HEAT: Standard Attention (Softmax)
HEATB: Bernoulli Attention (Sigmoid)

Method	Restaurant 14		Restaurant 15		Restaurant 16		Laptop 15	
	Pos/Neg	Pos/Neg/Neu	Pos/Neg	Pos/Neg/Neu	Pos/Neg	Pos/Neg/Neu	Pos/Neg	Pos/Neg/Neu
AT-LSTM	89.6	83.1	81.0	77.2	87.6	83.0	86.3	82.1
ATAE-LSTM	89.9	84.0	80.9	77.4	87.2	82.7	85.8	82.3
AT-BiGru	90.4	84.3	82.8	79.2	90.4	86.7	87.0	84.3
HEAT-GRU	89.6	84.3	81.2	79.1	89.7	85.5	87.8	84.5
HEATB-GRU	89.4	84.0	81.8	79.6	89.2	85.4	87.3	84.2
HEAT-BiGRU	91.3	85.1	83.0	80.1	90.8	87.1	87.9	84.9
HEATB-BiGRU	91.1	84.9	83.4	80.5	91.1	87.5	88.0	85.1

Attention Analysis



(a) Result of AT-BiGRU.



(b) Result of HEATB-BiGRU.

1. AT-BiGRU gets confused to locate sentiment word for aspect food in Figure 4(a)

Given aspect food, both "*great*" and "*dreadful"* obtain high scores

2. In Figure 4(b) HEATB-BiGRU solves the problem well

Expression "*service was dreadful*!" gets higher scores than other words given aspect ***service***
Expression "*great food*" achieves the top scores given aspect ***food***

Interactive Attention Networks for Aspect-Level Sentiment Classification [Ma et al. 2017]

- Previous approaches incorporated the target information (i.e. *aspect*) for modelling the target-specific contexts
  - Generated target-specific representations
- Studies ignored the separate modeling of target with respect to context
- BUT, coordination of targets and contexts could be useful
  - Example, "*The picture quality is clear-cut but the battery life is too short*"

    When *short* is collocated with *battery life*, sentiment class is *negative*

  - BUT, for *Short fat noodle spoon, relatively deep some curva*

    When *short* is collocated with *spoon*, sentiment tends to be *neutral*

Dehong Ma, Sujian Li, Xiaodong Zhang, Houfeng Wang. 2017. Interactive Attention Networks for Aspect-Level Sentiment Classification. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17).

Interactive Attention Networks for Aspect-Level Sentiment Classification [Ma et al. 2017]

- Now, the issue

  How to simultaneously model the target and context precisely?

- First, target and context can determine representations of each other

  For example, when we see the target "***picture quality***", context word "***clear-cut***" is naturally associated with the target and the vice-versa

  We argue that targets and contexts can be modeled separately but learned from their interaction

- Second, different constituents of a target aspect and context offer different information

  For example, it is easy to know that "*picture*" plays a more important role in the representation of the target "*picture quality*" (described by *clear-cut*)

Interactive Attention Networks for Aspect-Level Sentiment Classification [Ma et al. 2017]

- Both *targets* and *contexts* deserve special treatment and need to be learned their own representations via *interactive learning*
- **Why interactive?**
  - Interactively learn attentions in the contexts and targets, and generate the representations for targets and contexts separately

- **Steps of IAN**

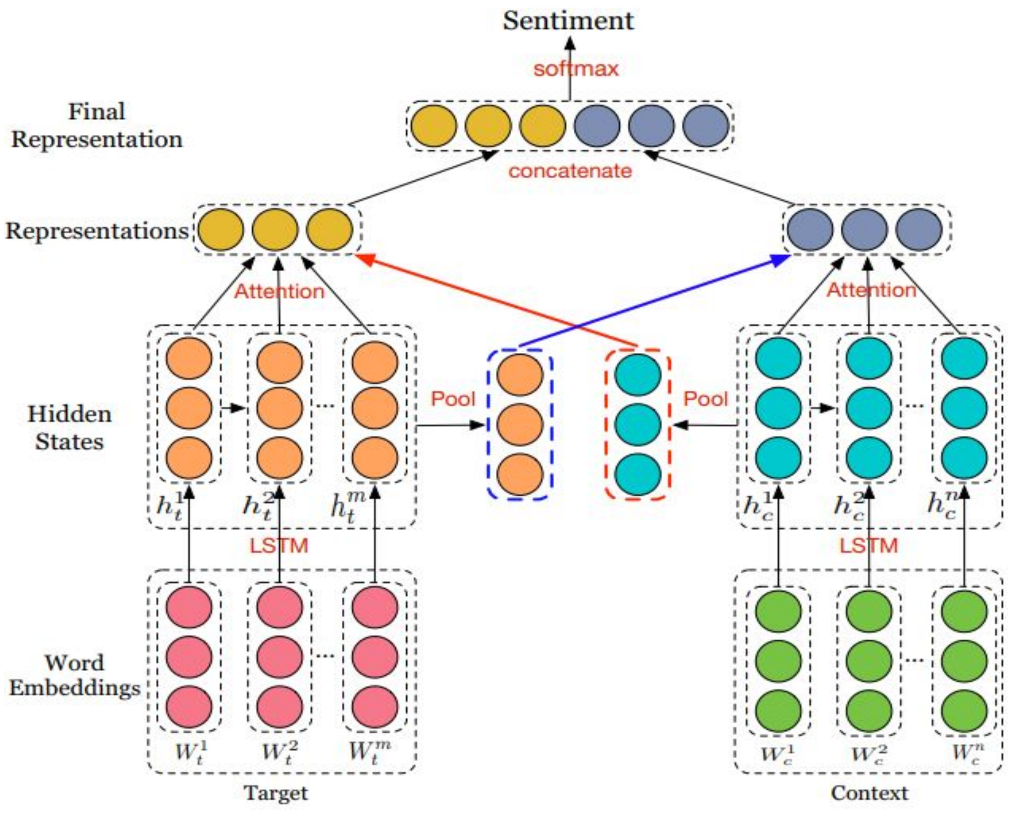
  - Utilizes the attention mechanism associated with a target to get important information from the context and compute context representation for sentiment classification

  - Makes use of the interactive information from context to supervise the modeling of the target which is helpful to judging sentiment

  - Finally, with both target representation and context representation concatenated, IAN predicts the sentiment polarity for the target within its context

Interactive Attention Networks (IAN)

- IAN learns the attentions for the contexts and targets separately
  - Generates the separate representations for targets and contexts via interaction with each other



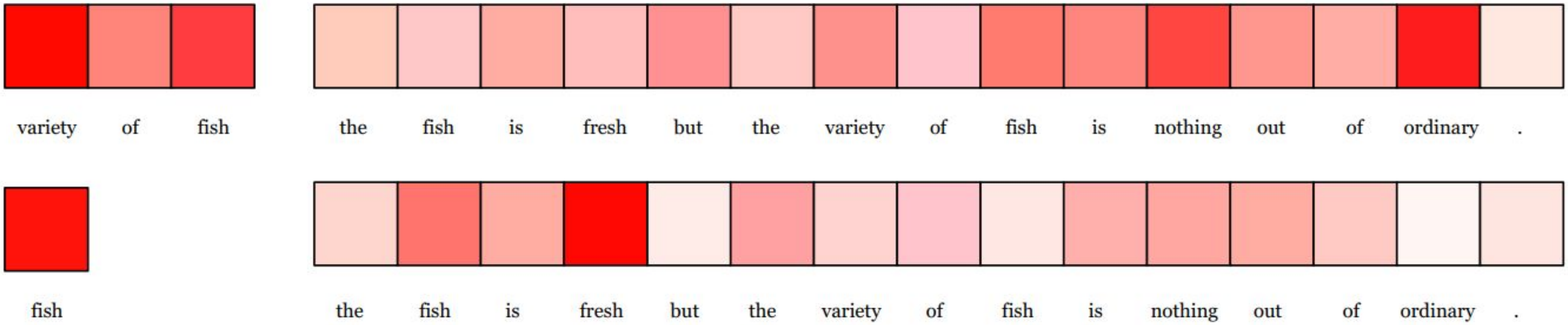
Experimental Results

- Dataset: SemEval-2014 [Pontiki et al., 2014]
    - Restaurant and Laptop

Method	Restaurant	Laptop
Majority	0.535	0.650
LSTM	0.743	0.665
TD-LSTM	0.756	0.681
AE-LSTM	0.762	0.689
ATAE-LSTM	0.772	0.687
IAN	**0.786**	**0.721**

Attention weights: Heatmap

“The ***fish*** is fresh but the ***variety of fish*** is nothing out of ordinary.



(a) weight for target

(b) weight for context

Effective Attention Modeling for Aspect-Level Sentiment Classification [He et al. 2018]

- Improved the *effectiveness of attention mechanism* to capture the *importance of each context* word towards a *target* by modeling their semantic associations
    - Proposed a method for ***target representation*** that better captures the semantic meaning of the ***opinion target***
    - Introduced an ***attention model*** that incorporates ***syntactic information*** obtained from a ***dependency parser***

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Effective Attention Modeling for Aspect-Level Sentiment Classification. In Proceedings of the 27th International Conference on Computational Linguistics, pages 1121–1131 Santa Fe, New Mexico, USA, August 20-26.

Target Representation

- While computing attention, simple averaging may not capture the real semantics of the target well
  - E.g., “*hot dog*” → Averaging of vectors may not represent it closer to the cluster of food items
- Represent the target as a weighted summation of aspect embeddings
- For aspect embedding matrix $T \in R^{K \times d}$, the target representation is computed as follows

$$\mathbf{t}_s = \mathbf{T}^\top \cdot \mathbf{q}_t$$

$$\mathbf{q}_t = softmax(\mathbf{W}_t \cdot \mathbf{c}_s + \mathbf{b}_t)$$

$$\mathbf{c}_s = Average(\frac{1}{m}\sum_{i=1}^{m}\mathbf{e}_{a_i}, \frac{1}{n}\sum_{j=1}^{n}\mathbf{e}_{w_j})$$

where ***K*** is number of predefined aspects (e.g., food, price, service, ambience and misc), ***m*** is the length of target, ***n*** is the length of sentence and ***e*** stands for embedding

Syntactic Information

- *Opinion words* that are closer to the *target* in the *dependency tree* are more relevant for determining its sentiment
- Attention model selectively attends to a small window of context words based on their location
- For a context window ws,

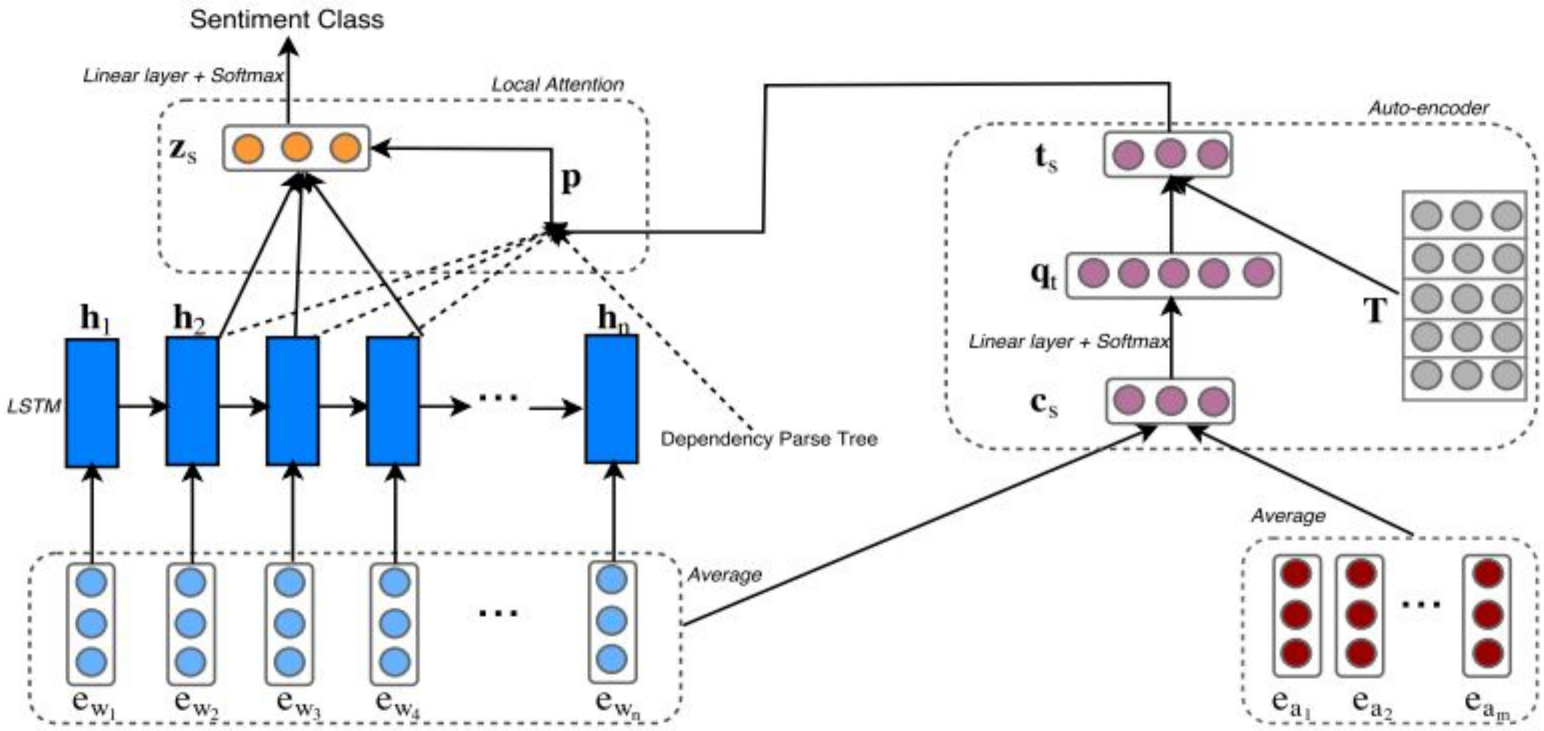
$$p_i = \frac{d_i}{\sum_j d_j}$$

$$d_i = \begin{cases} \frac{1}{2^{(l_i-1)}} \cdot exp(f_{score}(\mathbf{h}_i, \mathbf{t}_s)), & \text{if } l_i \in [1, ws] \\ 0, & \text{otherwise} \end{cases}$$

$$f_{score}(\mathbf{h}_i, \mathbf{t}_s) = tanh(\mathbf{h}_i^T \cdot \mathbf{W}_a \cdot \mathbf{t}_s)$$

where t_s is the target representation, l_i is the distance from the target in the dependency tree

Architecture



Experiments

- Dataset
  - SemEval-2014 [Pontiki et al., 2014] → Restaurant and Laptop
  - SemEval-2015 [Pontiki et al., 2015] → Restaurant
  - SemEval-2016 [Pontiki et al., 2016] → Restaurant

Method	Restaurant 14		Laptop 14		Restaurant 15		Restaurant 16	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
SVM	80.16	NA	70.49	NA	NA	NA	NA	NA
LSTM	75.23	64.21	66.79	64.02	75.28	54.1	81.94	58.11
LSTM+Attm	76.83	66.48	68.07	65.27	77.38	60.52	82.73	59.12
TDLSTM	75.37	64.51	68.25	65.96	76.39	58.7	82.16	54.21
TDLSTM+Attn	75.66	65.23	67.82	64.37	77.1	59.46	83.11	57.53
ATAE-LSTM	78.6	67.02	68.88	65.93	78.48	62.84	83.77	61.71
MemNet	76.87	66.4	68.91	63.95	77.89	59.52	83.04	57.91
LSTM+Attn+TarRep	78.95	68.67	70.69	66.59	80.05	**68.73**	84.24	**68.62**
LSTM+SynAttn	80.45	71.26	**72.57**	69.13	80.28	65.46	83.39	66.83
LSTM+SynAttn+TarRep	**80.63**	**71.32**	71.94	**69.23**	**81.67**	66.05	**84.61**	67.45

Hierarchical Attention based Position-aware Network for Aspect-level Sentiment Analysis [Li et al. 2018]

- Introduces **position embeddings** to learn the **position-aware representations** of sentences and generate the **target-specific representations** of contextual words
- Position of a target aspect in a sentence provides useful evidence
  - "*I bought a mobile phone, its camera is wonderful but the battery life is a bit short*"
  - In context window approach: For "*battery life*", both *"wonderful" and "short"* are likely to be considered as its adjunct word
  - If we encode the position information into the representation of each word effectively, we would have more confidence in concluding that the "**short**" is the adjunct word of "**battery life**" and predict the **sentiment** as **negative**
- Encode the position information into the representation of each word effectively

Lishuang Li, Yang Liu and AnQiao Zhou. 2018. Hierarchical Attention Based Position-aware Network for Aspect-level Sentiment Analysis. In Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL2018), pages 181-189 Brussels, Belgium, Oct 31-Nov 1.

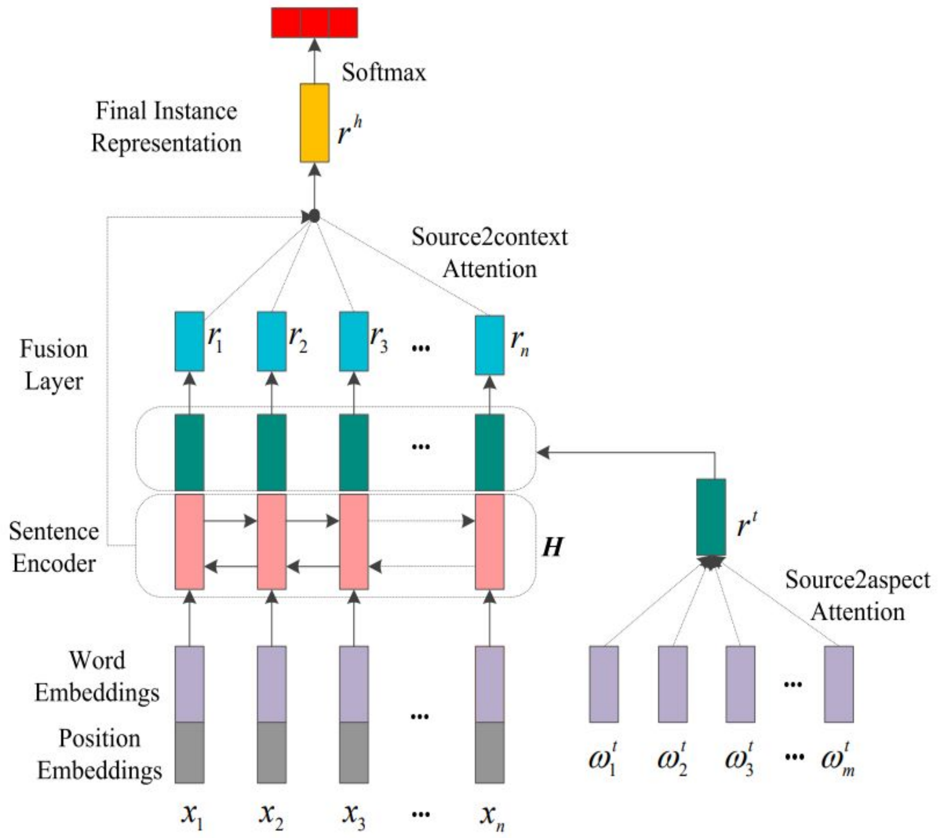
Hierarchical Attention Based Position-aware Network (HAPN)

- Position embeddings of word w_i

where
$$\begin{cases} i-k & i<k \\ i-k-m & n \geq i > k+m \\ 0 & k+m \geq i \geq k \end{cases}$$

k: Index of first word of target
m: Length of the target
n: Length of the sentence

- Source2aspect Attention
  - capture the most important clues in the target words
- Source2context Attention
  - capture the most indicative sentiment words in the context
  - generates weighted-sum embedding for sentence representation



Hierarchical Attention: More details

- Source2Aspect attention
  - Similar to self-attention
  - Generates the representation of aspect
  - Subsequently, *aspect-specific representation of each word = aspect representation + encoded position-aware representation*
  - Position-aware encoding: corresponds to the output of Bi-GRU that has input as *position embedding + word embedding*

- Source2Context attention
  - Captures the most indicative sentiment words in the context
  - Generates the weighted sum embeddings as the final sentence representation

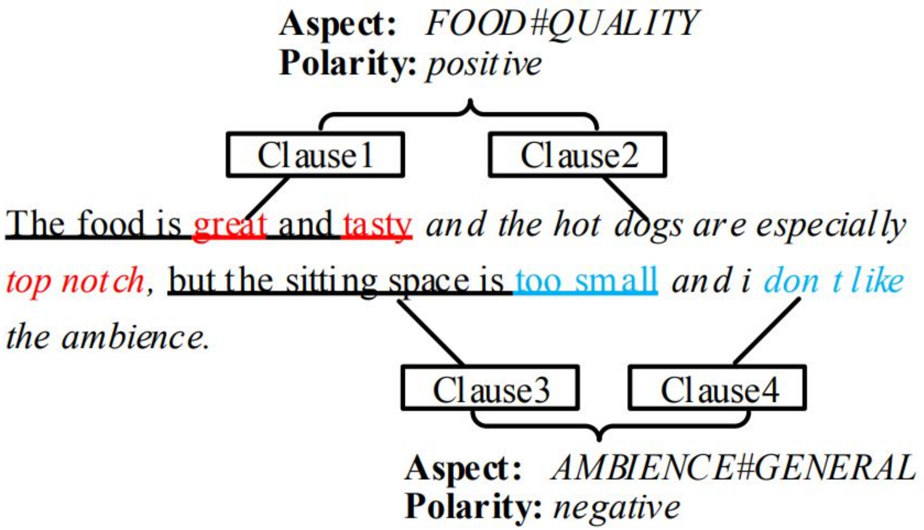
Experiments

- Dataset
  - SemEval-2014 [Pontiki et al., 2014]
    - Restaurant and Laptop
- Position embedding: position embedding lookup table is initialized randomly and tuned in the training phase
- BiGRU-PW
  - Weights the word embeddings of each word in the sentence based on the distance from the target
- BiGRU-PE
  - Concatenates the word embeddings and the position embeddings of each word

Method	Restaurant	Laptop
Majority	65.00	53.45
Bi-LSTM	78.57	70.53
Bi-GRU	80.27	73.35
Bi-GRU-PW	79.55	71.94
Bi-GRU-PE	80.89	76.02
TDLSTM	75.63	68.13
MemNet	79.98	70.33
IAN	78.60	72.10
HAPN	**82.33**	**77.27**

Aspect Sentiment Classification with both Word-level and Clause-level Attention Networks [Wang et al. 2018]

- Highlight the need for incorporating the importance of both words and clauses inside a sentence



Jingjing Wang, Jie Li, Shoushan Li, Yangyang Kang, Min Zhang, Luo Si, Guodong Zhou. 2018. Aspect Sentiment Classification with both Word-level and Clause-level Attention Networks. In Proceedings of IJCAI-2018, 4439-4445.

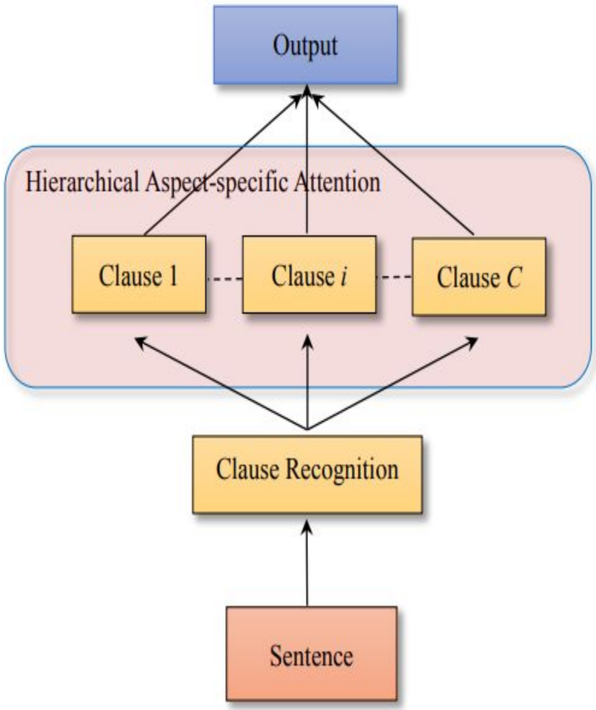
Motivation

- For a specific aspect, importance degrees of different words are different
  - Words such as “great”, “tasty” contribute much in implying the positive sentiment polarity for the aspect **FOOD#QUALITY**; BUT
  - Words such as “is”, “and” don’t contribute

- For a particular aspect, the importance degrees of different clauses are different
  - the first and second clauses have much stronger information in assisting the prediction of the sentiment polarity for the aspect **FOOD#QUALITY**
  - In contrast, the third and fourth clauses are more relevant to the aspect **AMBIENCE#GENERAL**

Proposed Approach

- Clause Recognition
  - Sentence-level discourse segmentation to segment a sentence into several clauses
- Hierarchical Attention
  - ***Word-level attention***: BiLSTM layers to encode all clauses and employed a word-level attention layer to capture the i*mportance degrees of words in each clause*
  - ***Clause-level attention***: BiLSTM layer to encode the output from the former layers and propose a clause-level attention layer to capture the importance degrees of all the clauses inside a sentence



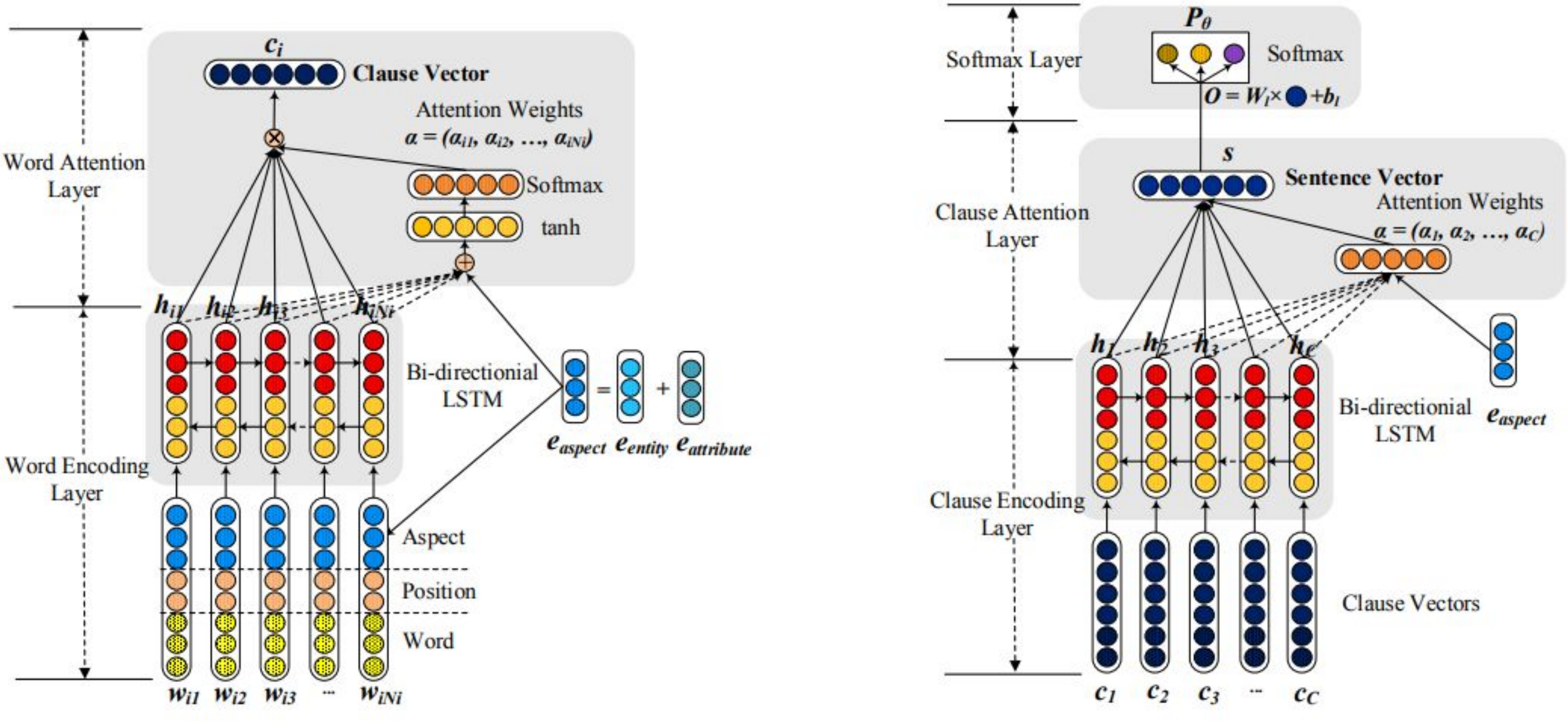
Clause Recognition

- Similar to *discourse segmentation*
  - Breaks a given text into non-overlapping segments called elementary discourse units (EDUs)

- Adopted Rhetorical Structure Theory (RST) [MANN, 1988]

[The food is great and tasty][A] [and the hot dogs are especially top notch,][B] [but the sitting space is too small][C] [and i don't like the ambience.][D]

Architecture



(a) Word-level Aspect-specific Attention Module

(b) Clause-level Aspect-specific Attention Module

Experiments

- Dataset
  - SemEval-2015 [Pontiki et al., 2015] → Restaurant and Laptop

Method	Restaurant		Laptop	
	Accuracy	Macro-F1	Accuracy	Macro-F1
Majority	0.537	0.233	0.570	0.242
LSTM	0.735	0.617	0.734	0.608
TC-LSTM	0.747	0.634	0.745	0.622
ATAE-LSTM	0.752	0.641	0.747	0.637
IAN	0.755	0.639	0.753	0.625
Hierarchical BiLSTM	0.763	0.647	0.767	0.632
Word-level Attn	0.789	0.662	0.785	0.646
Clause-level Attn	0.783	0.659	0.779	0.647
Word & Clause-level Attn	**0.809**	**0.685**	**0.816**	**0.667**

Memory Network for Aspect Sentiment Classification

Memory Network

- Introduced by [Weston et al. 2014]
- Core idea
  - Inference with a long-term memory component, which could be read, written to, and jointly learned with the goal of using it for prediction
- Formally,
  - A memory m → Array of objects/vectors
  - *Four components*
    - **Input feature map (*I*)**→ converts input (x) to internal feature representation
      - $I(x)$
    - **Generalization (*G*)**→ updates old memories with new input. Network compresses and generalizes its memories at this stage for some intended future use
      - $m_i = G(m_i, I(x), m), \ \forall i.$
    - **Output feature map (*O*)**→ generates an output representation given a new input and the current memory state,
      - $o = O(I(x), m)$
    - **Response (*R*)**→ outputs a response based on the output representation
      - $r = R(o).$

Some more details: Memory Network

- **Input**: Any kind of operations possible (NER, PoS tagging, Coreference etc. on text)
- **Generalization**
  - Its main task is to store the current input in a slot of the memory
  - Update the old stored values based on the new evidence
  - Memory can be stored with topic or entity if the input is very big (Freebase, Wikipedia etc)
- **Output**: Typically responsible for reading from memory and performing inference, e.g., calculating what are the relevant memories to perform a good response
- **Response:** Produces the final response given O

Example in a QA setup, *O finds relevant memories, and then R produces the actual wording of the answer, e.g., R could be an RNN that is conditioned on the output of O.*

Why is Memory Network for ASC?

- Conventional neural models like LSTM captures context information in an *implicit way*, and are *incapable of explicitly exhibiting important context clues of an aspect*
    - Only a small subset of context words actually needed in determining the sentiment polarity
- Example: *great food but the service was dreadful!*

  "**dreadful**" is an important clue for the aspect "**service**" but "**great**" is not needed

- Standard LSTM works in a sequential way
    - Manipulates each context word with the same operation
    - AND hence, it cannot explicitly reveal the importance of each context word

Why is Memory Network for ASC?

- *What could be the desirable solution then?*
  - Should be capable of explicitly capturing the importance of context words
  - Use the information to build up features for the sentence after given an aspect word
- *What a human will do?*
  - will selectively focus on parts of the contexts, and
  - acquire information where it is needed to build up an internal representation towards an aspect in his/her mind

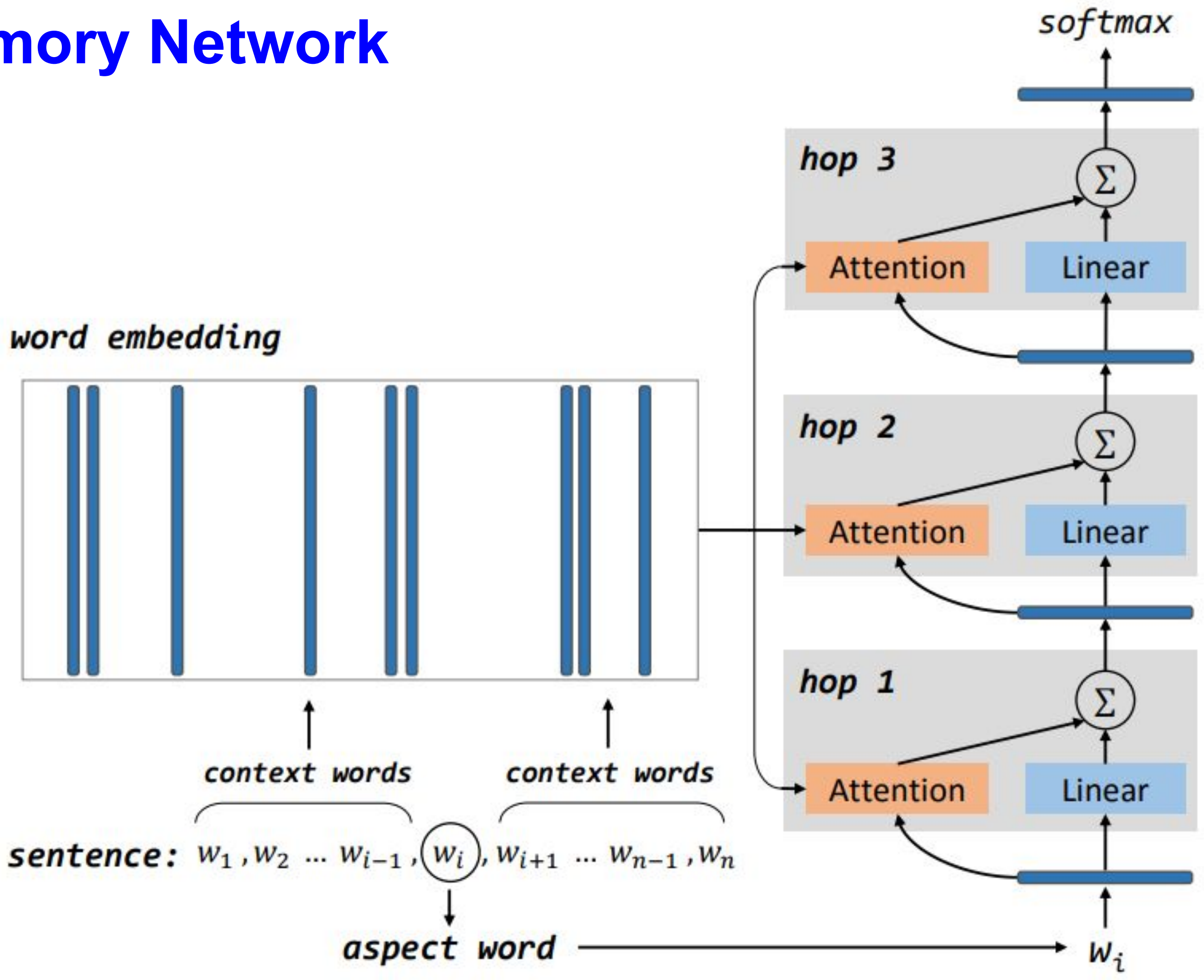
Equivalent to store an object in memory and then search for a reasonable match

Aspect Level Sentiment Classification with Deep Memory Network [Tang et al. 2016]

- Explicitly captures the *importance of each context word* when inferring the *sentiment polarity of an aspect*
- Utilized multiple computational layers with ***shared parameters (hops)***, each of which is a *neural attention model* over an external memory
- Each layer is a content- and location- based attention model, which *first learns the importance/weight of each context word* and then *utilizes this information to calculate the continuous text representation*

Duyu Tang, Bing Qin, Ting Liu. 2016. Aspect Level Sentiment Classification with Deep Memory Network. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 214–224, Austin, Texas, November 1-5, 2016.

Deep Memory Network



Attention: Content and Location

- **Content**
  - Determines the most attended context word with respect to the target aspect
  - Model could adaptively assign an importance score to each piece of memory according to its semantic relatedness with the aspect
- **Location**
  - Sentiment-bearing word closer to the aspect is more important
  - Distance of the word from the target is therefore very important
  - Introduced four strategies to include the location information
  - Memory content is updated based on the location attention (i.e. *how far is the memory element from the target aspect?*)

Multiple hops: Learning multiple levels of abstractions

- Single attention layer is essentially a weighted average compositional function
  - Not powerful enough to handle the sophisticated computationality like *negation*, *intensification* and contrary in language
- Multiple computational layers allow the deep memory network to learn representations of text with multiple levels of abstraction
- Each layer/hop retrieves important context words, and transforms the representation at previous level into a representation at a higher, slightly more abstract level

Experiments

- Dataset
  - SemEval-2014 [Pontiki et al., 2014]

Method	Laptop	Restaurant
Majority	53.45	65.00
Feature+SVM	72.10	80.89
LSTM	66.45	74.28
TD-LSTM	68.13	75.63
TD-LSTM + ATTENTION	66.24	74.31
MemNet(1)	67.66	76.10
MemNet(2)	71.14	78.61
MemNet(3)	71.74	79.06
MemNet(4)	72.21	79.87
MemNet(5)	71.89	80.14
MemNet(6)	72.21	80.05
MemNet(7)	72.37	80.32
MemNet(8)	72.0	80.14
MemNet(9)	72.21	80.95

Target-Sensitive Memory Networks for Aspect Sentiment Classification [Wang et al. 2018]

- In Memory Network, attention mechanism plays a crucial role in detecting the sentiment context for the given target
- However, sentiment polarity of the (detected in memory networks) context is dependent on the given target and it cannot be inferred from the context alone
  - Sentiment contexts for both these sentences are "*high*", i.e. the attention mechanism will have higher weights for "*high*" in both the cases
    - *The **price** is high.* → Negative
    - *The **screen resolution** is high. → Positive*

- Incorporate target information to infer **(price, high) as negative** and **(screen resolution, high) as positive**

Shuai Wang, Sahisnu Mazumder, Bing Liu, Mianwei Zhou, Yi Chang. 2018. Target-Sensitive Memory Networks for Aspect Sentiment Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 957–967 Melbourne, Australia, July 15 - 20, 2018.

Six variants of TMNs

1. **Non-linear Projection (NP)**
   - α_i , c_i and v_t ⇒ Attention score, Context and vector of target (t)
   - Interaction between target and context

$$s = W \cdot tanh(\sum_i \alpha_i c_i + v_t)$$

2. **Contextual Non-linear Projection (CNP)**

$$s = W \sum_i \alpha_i \cdot tanh(c_i + v_t)$$

3. **Interaction Term (IT)**:
   - It measures the sentiment-oriented interaction effect between targets and contexts, i.e., Target-Context-Sentiment (TCS)

$$s = \sum_i \alpha_i (W_s c_i + w_I \langle d_i, d_t \rangle)$$

$$d_i = D x_i, d_t = D t$$

D = Embedding matrix that captures the sentiment interactions

Six variants of TMNs

4. **Coupled Interaction (CI):**
   - Additionally captures the global correlation between context and different sentiment classes

$$s = \sum_i \alpha_i (W_s c_i + W_I \langle d_i, d_t \rangle e_i)$$

5. Joint Coupled Interaction (JCI):
   - Simplification of CI model

$$s = \sum_i \alpha_i (W_s c_i + W_I \langle d_i, d_t \rangle c_i)$$

6. Joint Projected Interaction (JPI)
   - First component, captures target-independent sentiment effect
   - Second component, TCS interaction

$$s = \sum_i \alpha_i W_J \tanh(W_1 c_i) + \sum_i \alpha_i W_J \langle d_i, d_t \rangle \tanh(W_2 c_i)$$

Experiments

- Dataset: SemEval-2014 [Pontiki et al., 2014]
    - Restaurant and Laptop

Method	Restaurant		Laptop	
	1-hop	3-hop	1-hop	3-hop
AE-LSTM	66.45	-	62.45	-
ATAE-LSTM	65.41	-	59.41	-
NP	64.62	65.98	62.63	67.79
CNP	65.58	66.87	64.38	64.85
IT	65.37	68.64	63.07	66.23
CI	66.78	68.49	63.65	66.79
JCI	66.21	68.84	64.19	67.23
JPI	66.58	67.86	64.53	64.16

IARM: Inter-Aspect Relation Modeling with Memory Networks in Aspect-Based Sentiment Analysis [Majumder et al. 2018]

- Incorporates the neighboring aspects related information for sentiment classification of the target aspect (i.e. *there is a dependency between different aspect terms*)
  - Example 1: "*The **menu** is very limited - I think we counted 4 or 5 **entries**.*"
    - Non-trivial to predict the sentiment for aspect "***entries***", unless the other aspect "**menu**" is considered
    - Negative sentiment of **"menu**" induces "**entries**" to have the same sentiment
  - Example 2: "***Food** is usually very good, though I wonder about freshness of **raw vegetables***"
    - No clear sentiment marker for "***raw vegetables***"
    - The ***positive*** sentiment of "***food***", due to the word "***good***", and the presence of conjunction "***though***" determines the sentiment of "***raw vegetables***" to be *negative*

Navonil Majumder, Soujanya Poria, Alexander Gelbukh, Md Shad Akhtar, Erik Cambria , Asif Ekbal. 2018. IARM: Inter-Aspect Relation Modeling with Memory Networks in Aspect-Based Sentiment Analysis. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3402–3411 Brussels, Belgium, October 31 - November 4, 2018.

Method: Key Steps

- **Input Representation**
  - Input sentences and aspect-terms are represented using pre-trained Glove word embeddings
  - For multi-worded aspect-terms, we take the mean of constituent word embeddings as aspect representation
- **Aspect-Aware Sentence Representation**
  - Embedding of each word in a sentence is concatenated with the given aspect representation
  - Modified sequence of words is fed to a GRU for context propagation
  - Attention layer to obtain the aspect-aware sentence representation (for all the aspects in a sentence)

Method: Key Steps

- **Inter-Aspect Dependency Modeling**
    - Match the target-aspect-aware sentence representation with aspect-aware sentence representation of the other aspects
    - More refined sentence representation after a certain number of iterations of the memory network
    - Softmax layer for final classification

Inter-Aspect Relation Modeling (IARM)

- Aspect-Aware Sentence Representation (AASR) [Wang et al. 2016]

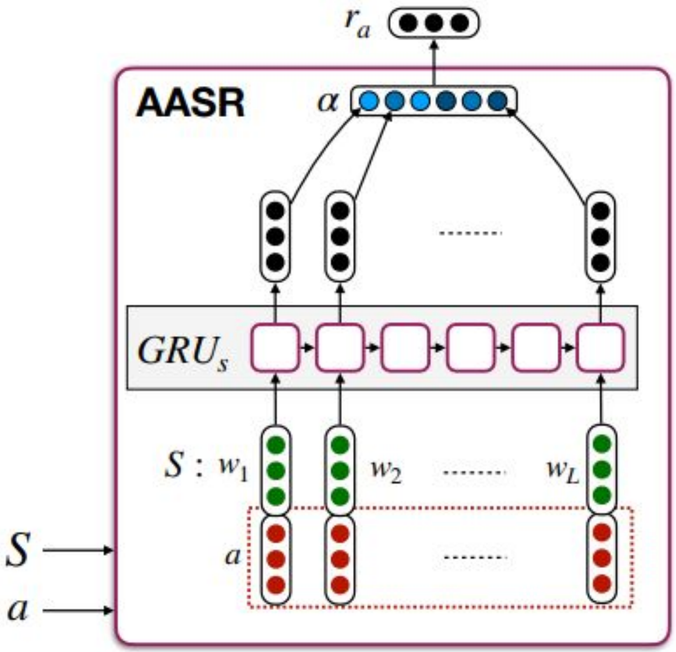
$$S_{a_i} = [w_1 \oplus a_i, w_2 \oplus a_i, \ldots, w_L \oplus a_i] \in \mathbb{R}^{L\times 2D}$$

$$R_{ai} = GRU\,(S_{ai})$$

$$\alpha = softmax(R_{ai}\ W_s + b)$$

$$r_{ai} = \alpha^T R_{ai}$$

$$R = [r_{a_1}, r_{a_2}, \ldots, r_{a_M}]$$



Aspect-aware Sentence Representation

Inter-Aspect Relation Modeling (IARM)

- Inter-Aspect Dependency Modeling
  - Models the dependency of the target aspect with the other aspects in the sentence

$$Q = GRU_a(R)$$

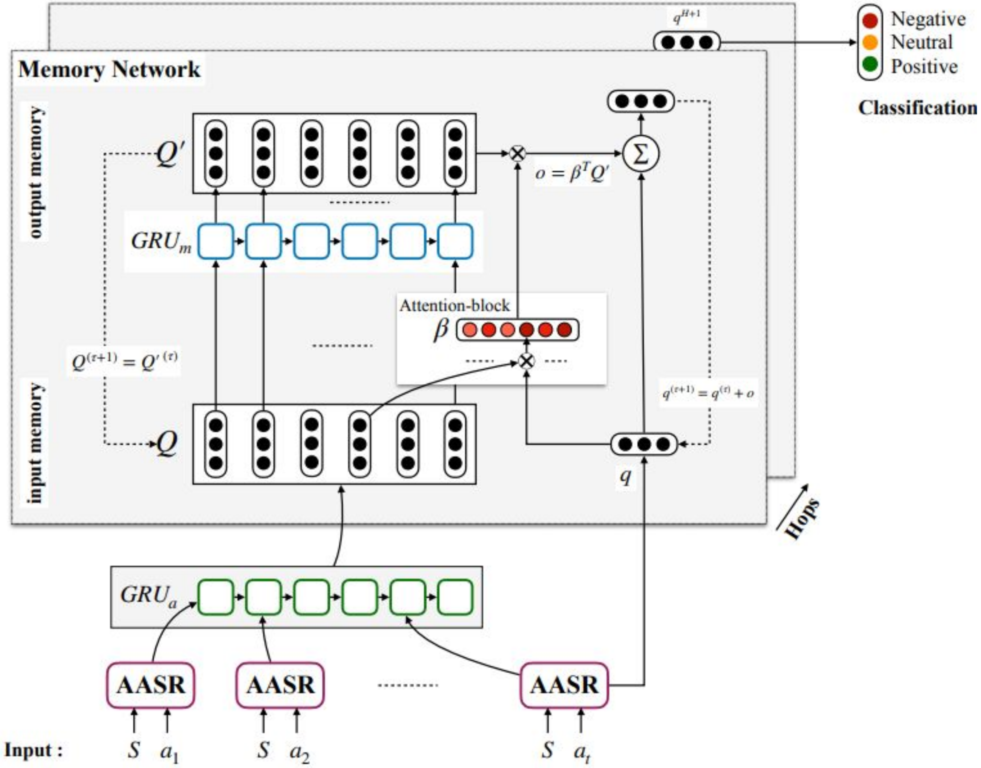
$$q = \tanh(r_{a_t} W_T + b_T)$$

$$z = qQ^T,$$

$$\beta = softmax(z)$$

$$Q' = GRU_m(Q)$$

$$o = \beta^T Q'$$



Experimental Results

- Dataset: SemEval-2014 [Pontiki et al., 2014]
    - Restaurant and Laptop

Method	Restaurant	Laptop
Majority	0.535	0.650
LSTM	0.743	0.665
TD-LSTM	0.756	0.681
AE-LSTM	0.762	0.689
ATAE-LSTM	0.772	0.687
IAN	0.786	0.721
IARM	**0.800**	**0.738**

Attention weights: Heatmaps

Example 1:

*“I recommend any of their **salmon dishes**.”*

(a) Attention weight for aspect “*salmon dishes*” for IAN.

(b) Attention weight for aspect “*salmon dishes*” for IARM.

(a) Attention weights for aspect “*cosi sandwiches*” for IAN.

(b) Attention weights for aspect “*cosi sandwiches*” for IARM.

(c) Attention weights for aspect “*coffee*” for IARM.

Example 2:

*“**Coffee** is a better deal than overpriced **cosi sandwiches**.”*

Summary and Takeaways

- Summary
    - Presented the background of ABSA
    - Presented the state-of-the-art deep learning models like LSTM, LSTM with attention, GRU, Memory networks etc. for aspect classification
- Takeaways
    - LSTM with target-specific attention helps obtaining good accuracy for ASC
    - Encoding position of the apect term in the sentence helps for better classification
    - Interactive attention (aspect-aware as well as context-aware representations) can better disambiguate the classification
    - Hierarchical attention (attention at aspect level to find most matching aspect term + attention to find the best sentiment bearing words) is useful
    - Memory network could be employed to model the inter-aspect relations
    - Cross-lingual embedding representation is important to perform multi-lingual and cross-lingual SA involving low-resource languages

Future Works

- Sentiment intensity prediction in ABSA
- ABSA in multi-modal scenario
- Effective solutions to ABSA in low-resource scenario
    - Cross-lingual embedding
    - Injecting external knowledge base into deep neural network
    - Transfer learning and domain adaptation

References

- Zeiler, Matthew D. and Fergus, Rob (2014). Visualizing and Understanding Convolutional Networks. Computer Vision -- ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I. Springer International Publishing
- Sepp Hochreiter and Jurgen Schmidhuber. 1997. ¨ Long short-term memory. Neural computation, 9(8):1735–1780.
- Duyu Tang, Bing Qin, Xiaocheng Feng, Ting Liu. 2016. Effective LSTMs for Target-Dependent Sentiment Classification. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 3298–3307, Osaka, Japan, December 11-17 2016.
- Md Shad Akhtar, Palaash Sawant, Sukanta Sen, Asif Ekbal, and Pushpak Bhattacharyya (2018). Solving Data Sparsity for Aspect based Sentiment Analysis using Cross-linguality and Multi-linguality. In Proceedings of the 16th Annual Conference of the NAACL:HLT-2018, June 2018, New Orleans, LA, USA, pages 572–582.
- Yequan Wang, Minlie Huang, Li Zhao and Xiaoyan Zhu. 2016. Attention-based LSTM for Aspect-level Sentiment Classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 606–615, Austin, Texas, November 1-5, 2016.
- Jiajun Cheng, Shenglin Zhao, Jiani Zhang, Irwin King, Xin Zhang, Hui Wang. 2017. Aspect-level Sentiment Classification with HEAT (HiErarchical ATtention) Network. In Proceedings of the CIKM-17, November 6-10, 2017, Singapore.
- Dehong Ma, Sujian Li, Xiaodong Zhang, Houfeng Wang. 2017. Interactive Attention Networks for Aspect-Level Sentiment Classification. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17).
- Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Effective Attention Modeling for Aspect-Level Sentiment Classification. In Proceedings of the 27th International Conference on Computational Linguistics, pages 1121–1131 Santa Fe, New Mexico, USA, August 20-26.
- Lishuang Li, Yang Liu and AnQiao Zhou. 2018. Hierarchical Attention Based Position-aware Network for Aspect-level Sentiment Analysis. In Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL2018), pages 181-189 Brussels, Belgium, Oct 31-Nov 1.
- Jingjing Wang, Jie Li, Shoushan Li, Yangyang Kang, Min Zhang, Luo Si, Guodong Zhou. 2018. Aspect Sentiment Classification with both Word-level and Clause-level Attention Networks. In Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL2018), pages 181-189 Brussels, Belgium, Oct 31-Nov 1.
- XINYI WANG, GUANGLUAN XU, JINGYUAN ZHANG, XIAN SUN, LEI WANG, AND TINGLEI HUANG. 2018. Syntax-Directed Hybrid Attention Network for Aspect-Level Sentiment Analysis. IEEE Acess, Jan. 2019.
- Duyu Tang, Bing Qin, Ting Liu. 2016. Attention-based LSTM for Aspect-level Sentiment Classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 214–224, Austin, Texas, November 1-5, 2016.
- Cheng Li, Xiaoxiao Guo, Qiaozhu Mei. 2017. Deep Memory Networks for Attitude Identification. In Proceedings of the WSDM 2017, February 06-10, 2017, Cambridge, United Kingdom.
- Shuai Wang, Sahisnu Mazumder, Bing Liu, Mianwei Zhou, Yi Chang. 2018. Target-Sensitive Memory Networks for Aspect Sentiment Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 957–967 Melbourne, Australia, July 15 - 20, 2018.
- Navonil Majumder, Soujanya Poria, Alexander Gelbukh, Md Shad Akhtar, Erik Cambria , Asif Ekbal. 2018. IARM: Inter-Aspect Relation Modeling with Memory Networks in Aspect-Based Sentiment Analysis. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3402–3411 Brussels, Belgium, October 31 - November 4, 2018.

References

- Md. Shad Akhtar, Tarun Garg, Asif Ekbal (2020). **Multi-task learning for aspect term extraction and aspect sentiment classification.** Neurocomputing 398: 247-256 (2020)
- Md. Shad Akhtar, Palaash Sawant, Sukanta Sen, Asif Ekbal, Pushpak Bhattacharyya **(2019). Improving Word Embedding Coverage in Less-Resourced Languages Through Multi-Linguality and Cross-Linguality: A Case Study with Aspect-Based Sentiment Analysis.** ACM Trans. Asian Low Resour. Lang. Inf. Process. 18(2): 15:1-15:22 (2019)
- Md. Shad Akhtar, Abhishek Kumar, Asif Ekbal, Chris Biemann, Pushpak Bhattacharyya **(2019). Language-Agnostic Model for Aspect-Based Sentiment Analysis.** IWCS (1) 2019: 154-164
- Md. Shad Akhtar, Palaash Sawant, Sukanta Sen, Asif Ekbal, Pushpak Bhattacharyya **(2018). Solving Data Sparsity for Aspect Based Sentiment Analysis Using Cross-Linguality and Multi-Linguality.** NAACL-HLT 2018: 572-582
- Md. Shad Akhtar, Deepak Kumar Gupta, Asif Ekbal, Pushpak Bhattacharyya (2017). **Feature selection and ensemble construction: A two-step method for aspect based sentiment analysis.** Knowl. Based Syst. 125: 116-135 (2017)
- Md. Shad Akhtar, Asif Ekbal, Pushpak Bhattacharyya **(2016). Aspect based Sentiment Analysis in Hindi: Resource Creation and Evaluation.** LREC 2016
- **Md. Shad Akhtar,** Asif Ekbal, Pushpak Bhattacharyya **(2016) .Aspect Based Sentiment Analysis: Category Detection and Sentiment Classification for Hindi.** CICLing (2) 2016: 246-257

Thank you for your attention!