Cross-Domain Aspect Extraction using Transformers Augmented with Knowledge Graphs

FROM:CIKM-2022

ADVISOR : JIA-LING, KOH

SPEAKER: FAN-JI-YU

DATE:2022/03/07

Content

Introduction

- Aspect Based Sentiment Analysis
- Input
- Problem Formulate

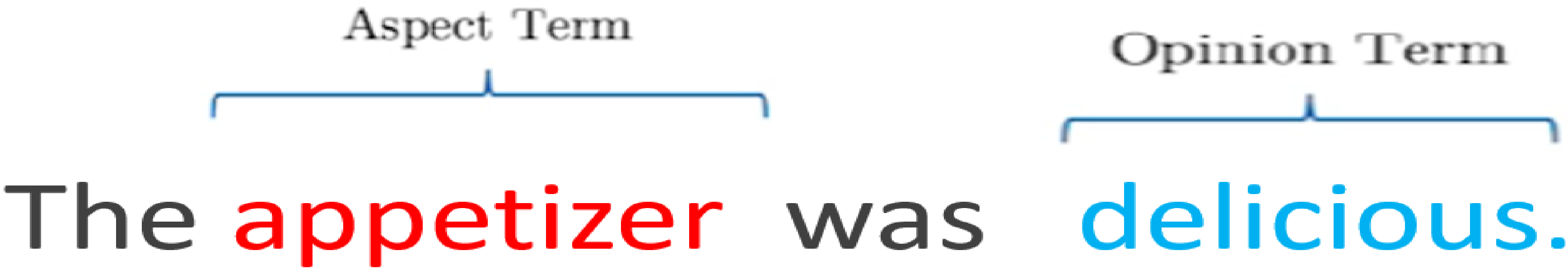
Method

Exprement

Conclusion

Aspect Based Sentiment Analysis(ABSA)

A fundamental task of the **fine-grained** sentiment analysis is **Aspect** and **Opinion** Term extraction.

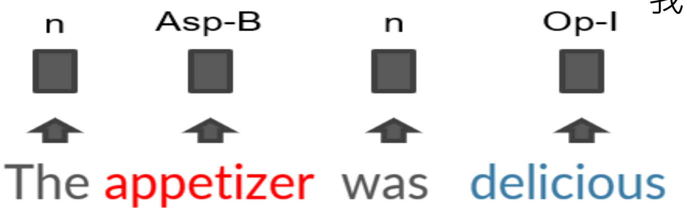


Introduction: (Input)

- Asp & Op -> Sequence tagging problem

  - Input Sequence : X = {x1, x2, .., xn}
  - Goal :Correctly predict a label Y = {y1, y2, …, yn} $yi \in \{\boldsymbol{BA}, \boldsymbol{IA}, BO, IO, N\}$

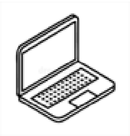
我們這邊專注在Aspect



Introduction: (Problem Formulate)

- Asp & Op -> Sequence tagging problem
  Same-domain(**Impressive results**)
  Cross-domain (**lack of extensibility and robustness**) :

- $D^{S} = \{(x_j^S\ y_j^S)\}$, $D^{T} = \{(x_i^T\ )\}$
- Predict label of **target** domain y_i^T

Source 

unrelated

Target 

Laptop:[processor, hardware] → Restaurant:[food, appetizer]

Feature Augments

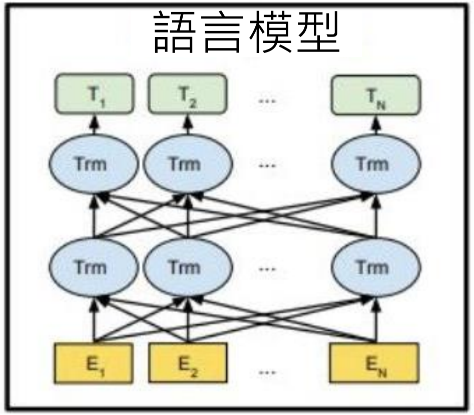
我們今天把特徵加入提升模型效果

這邊有很成兩種特徵Augment

1. **Early Augment: 提示模型訓練加強訓練**
2. Late Augment :提示結果加強權重

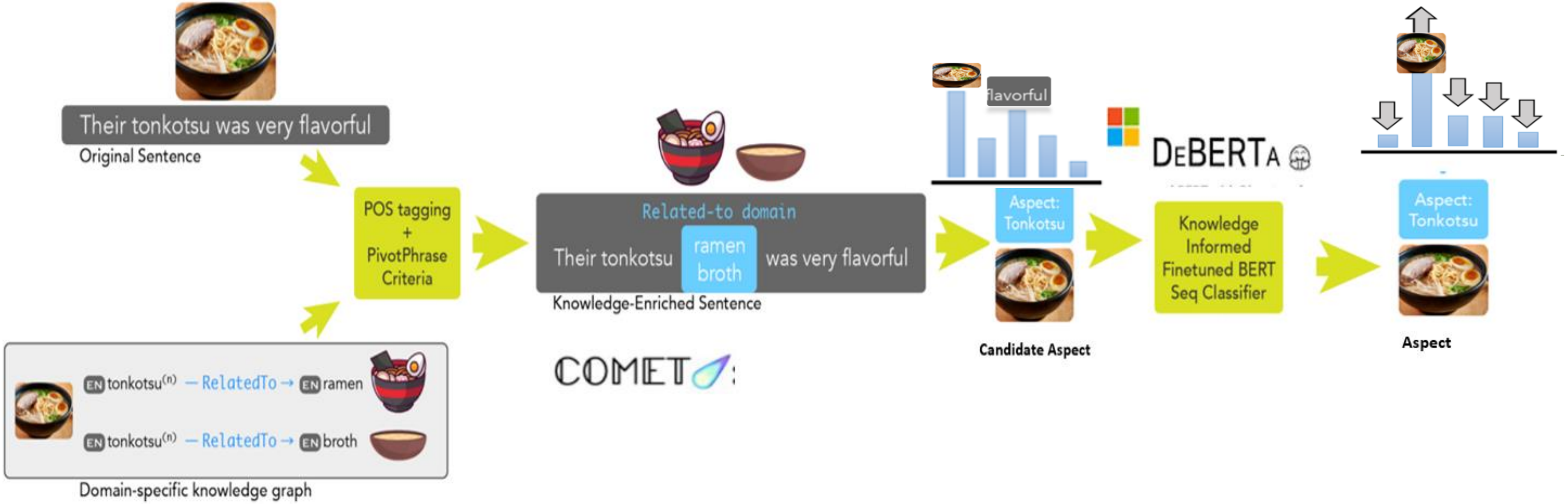
Late Augment

output sequence



Early Augment

input sequence


Their tonkotsu was very flavorful
Original Sentence
POS tagging
+
PivotPhrase
Criteria
EN tonkotsu(n) — RelatedTo → EN ramen
EN tonkotsu(n) — RelatedTo → EN broth
Domain-specific knowledge graph
Related-to domain
Their tonkotsu
ramen
broth
was very flavorful
Knowledge-Enriched Sentence
COMET
flavorful
Aspect:
Tonkotsu
Candidate Aspect
DeBERTa
Knowledge
Informed
Finetuned BERT
Seq Classifier
Aspect:
Tonkotsu
Aspect

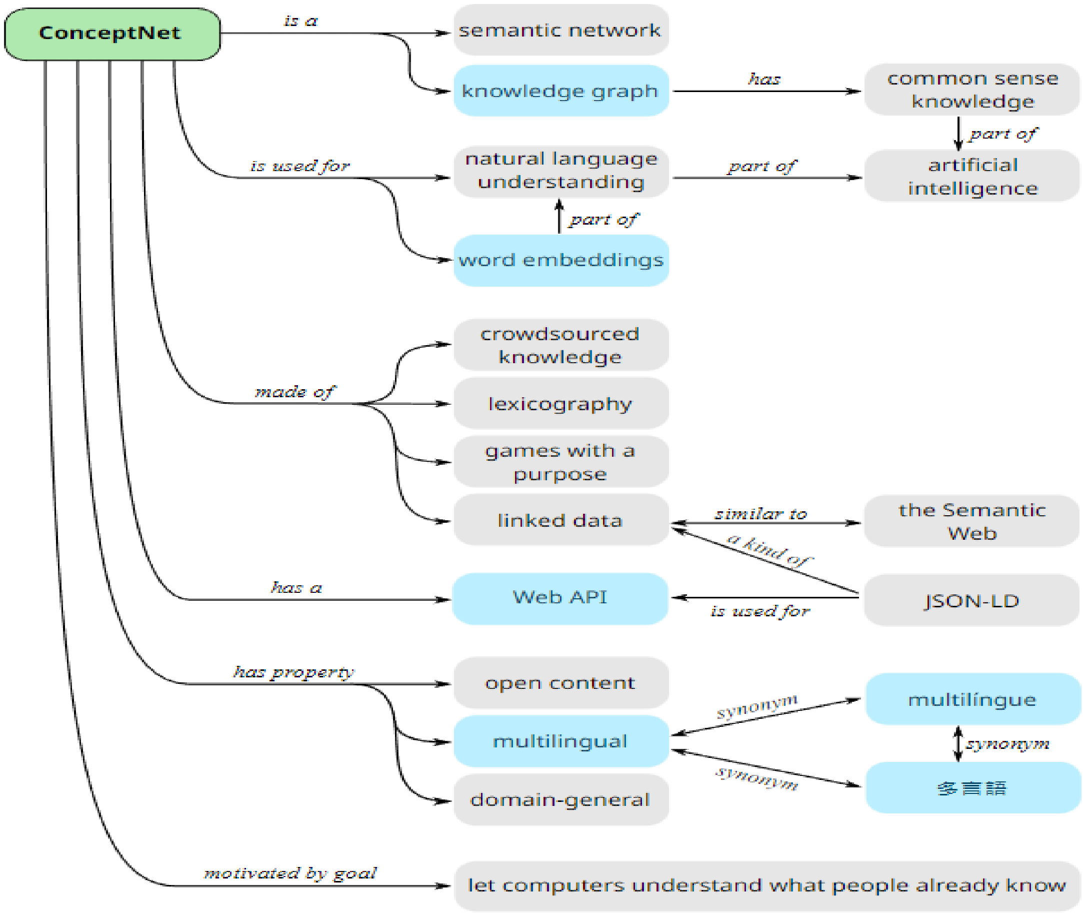
Structure

Content

- Introduction
- Method
  - Domain-Specific KG Preparation
  - Determining when to Inject Knowledge
  - Knowledge Injection Mechanisms
- Exprement
- Conclusion

ConceptNet

要使用Concpet NET



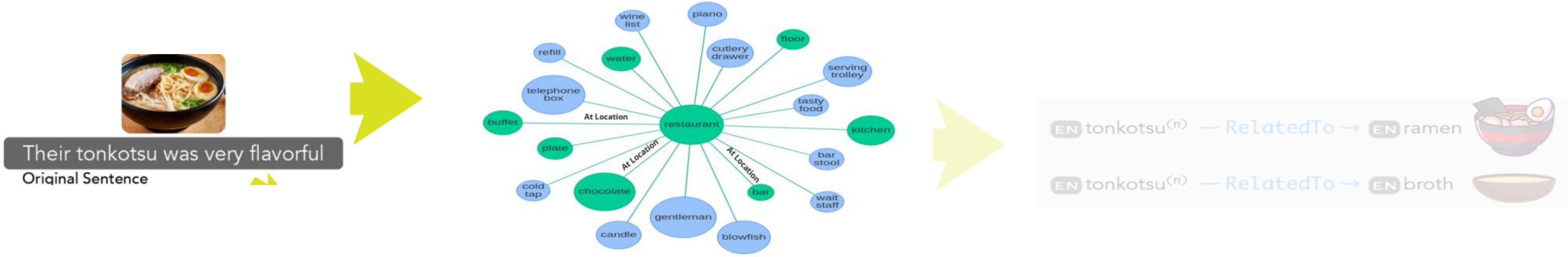
Domain-Specific KG Preparation ConceptNet

- Querying a **subgraph** from **ConceptNet** which is related to target **domain.**

- 並運TF-IDF 計算 Target domain 跟 seed **(node in side the subgraph) 重要性分數.**

  $S_d = \{s_1, s_2, \ldots, s_k\}$. Find **Top K seed where k =7**

- **Subgraph**的Maximum Distance is 2 (h=2)



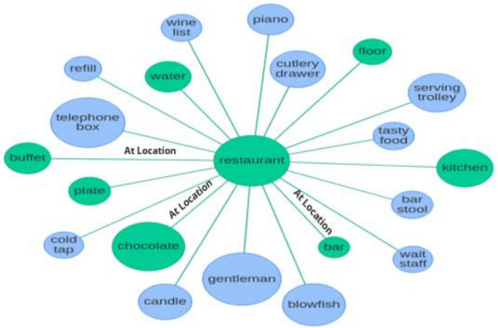
Domain-Specific KG Preparation
Cosine similarity

SubGraph 中的每個node 都會Pre-compute的 embedding

- 運用Cosine similarity 計算 node i 跟 j 之間的相似性
- 消除掉跟Target Domain 低度相關性的點 $P_{\min} < 0.2$

$$r_{i,j} = \frac{\mathbf{e}_i \cdot \mathbf{e}_j}{\|\mathbf{e}_i\|\|\mathbf{e}_j\|}$$

$$P_{\min} = \min_{\forall i \in \{1,..,h\}} r_{s,i}$$



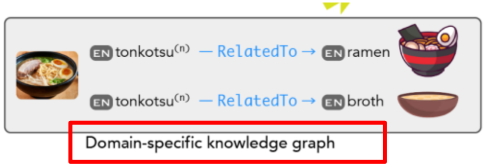
Domain-Specific KG Preparation

COMET

因為Corpus每條評論都是單獨Subgraph

這邊我們使用 Commet 去自動增強KG 之間的連接性

到這裡我們才完成Domain-Specific KG 的準備

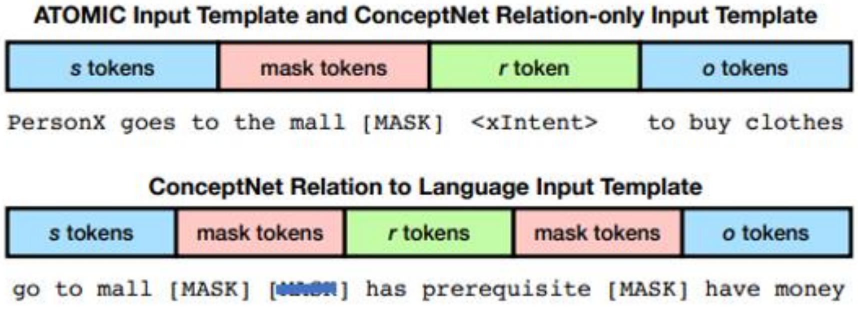


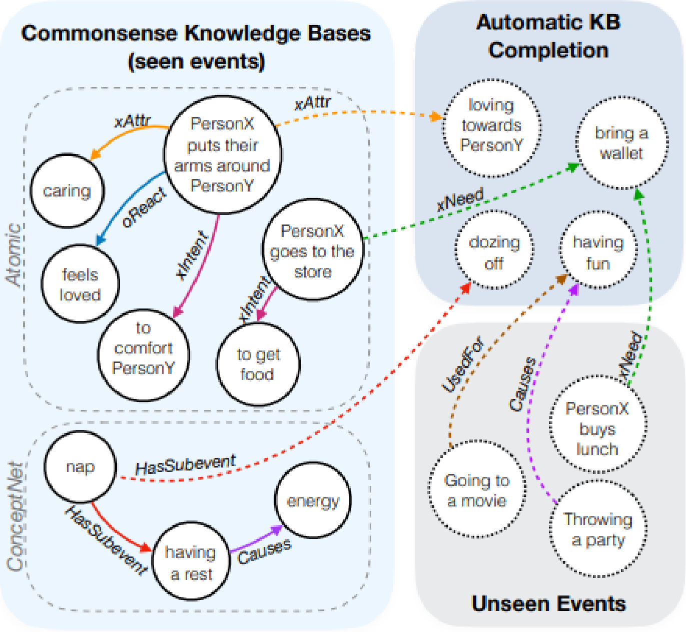
COMET

使用GPT預訓練模型，我們給他Head 以及 relation

讓他去生成可能的tail 產生 Knowledge Graph

Tuple (H,R,**T**)



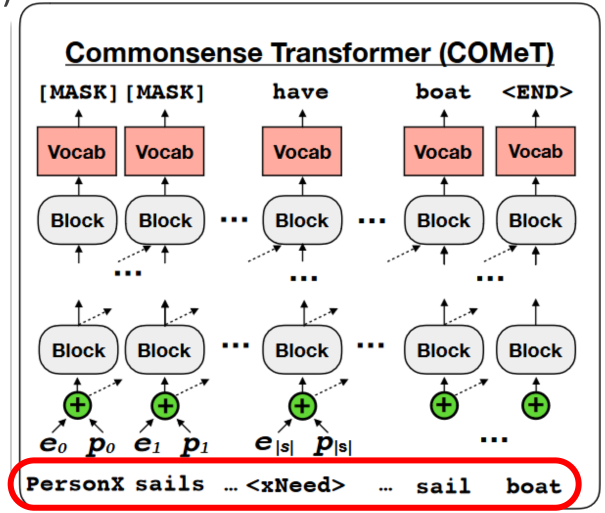


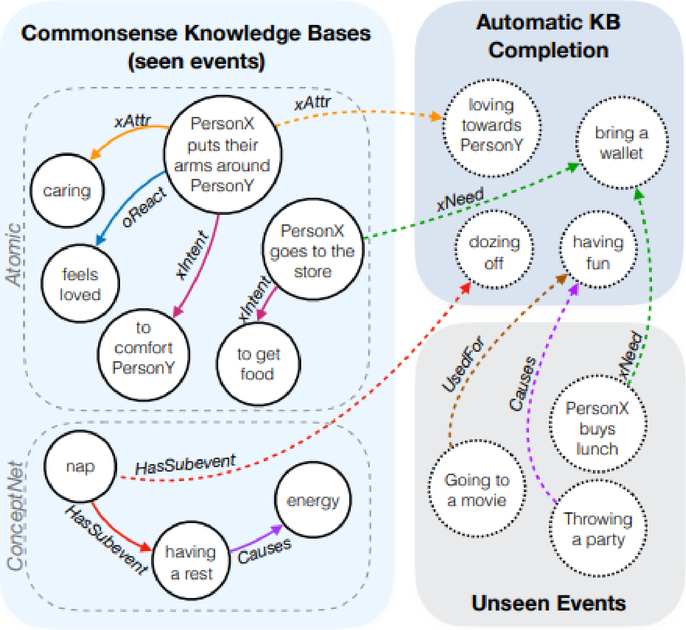
COMET

使用GPT預訓練模型，我們給他Head 以及 relation

讓他去生成可能的tail 產生 Knowledge Graph

Tuple (H,R,T)





Content

- Introduction
- Method
  - Domain-Specific KG Preparation
  - Determining when to Inject Knowledge
  - Knowledge Injection Mechanisms
- Exprement
- Conclusion

Determining when to Inject Knowledge

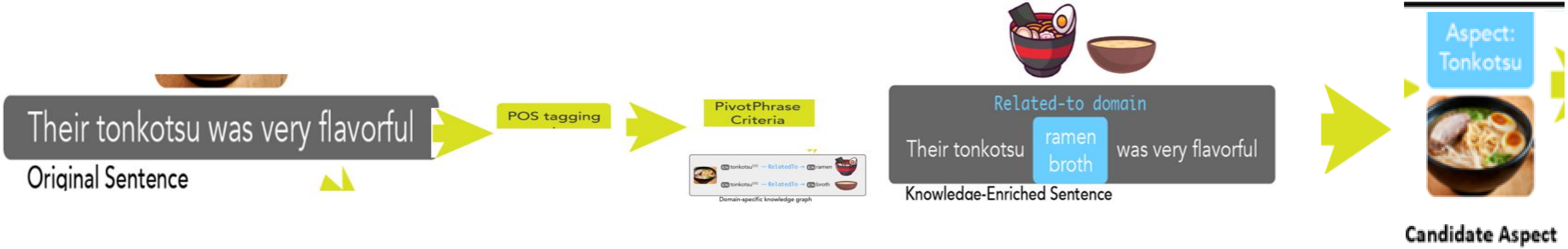
1. 把原語句做**POS tagging**，篩選出是 名詞或是名詞片語的地方並標記出來

2. 把這些tagging 完成的語句跟剛剛的**Domain-Specific KG** 去**Compare**

3. 被**POS tagging的**文字**必須完全跟Domain裡符合(由左至右比較)**

4.**符合後**插入原來的語句中獲得 **Knowledge-Enriched Sentence**

幫助找出**可能為 Aspect的Token** ，並在放入語言模型後增進 **aspect classification**



Content

Introduction

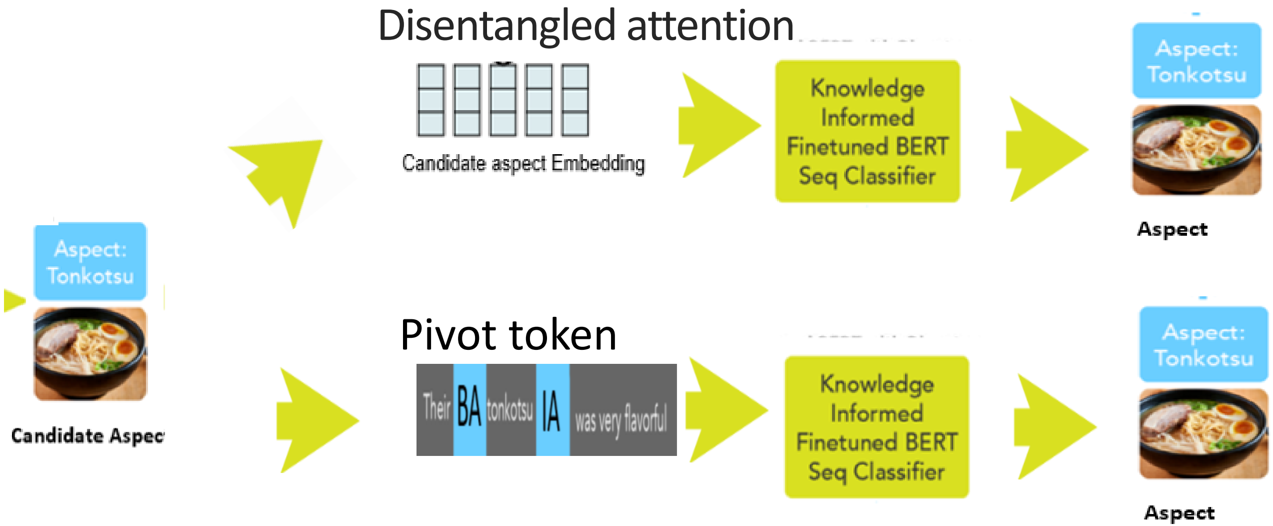
Method

- Domain-Specific KG Preparation
- Determining when to Inject Knowledge
- **Knowledge Injection Mechanisms**
  - **Pivot token**
  - Embedding

Exprement

Conclusion

Knowledge Injection Mechanisms



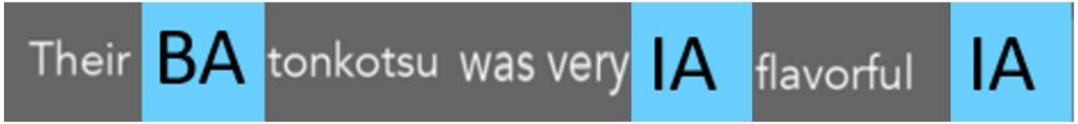
Knowledge Injection Mechanisms(Pivot token)

pivot token:

是一種特殊的 token，用於向模型指示前面的 token 很有可能被標記為 aspect

Stochastic Insertion:

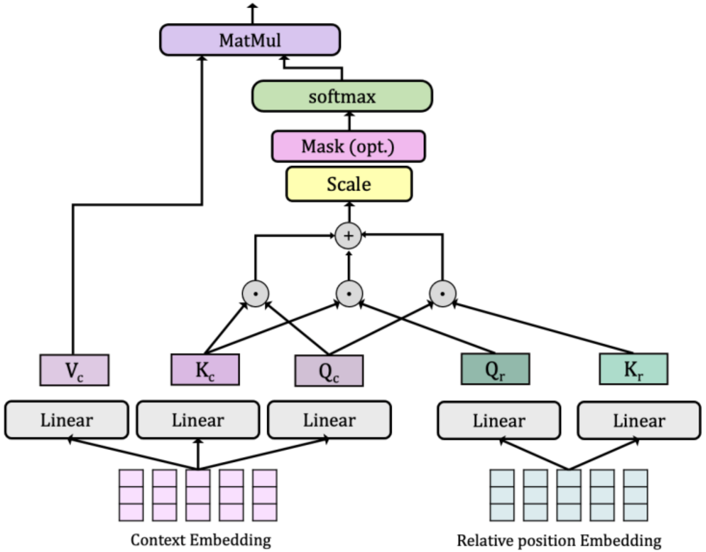
幫助模型可以有寬容度去學習，學習到太單一的Domain 並且還可以讓模型學習到語言模型的隱藏資訊。(在訓練的時候適當加入一些雜訊是好的



Knowledge Injection Mechanisms(Disentangled Attention)

這邊我先跟大家我講一個小笑話幫助大家對這個模型可以有更多記憶點
有一天大家看到Bert 都發現你好像變壯了，因為我是新**De(的)bert**阿!!(機器人笑)

- Deberta 是一個成功的NLU模型



Knowledge Injection Mechanisms(Disentangled Attention)

今天input Sequence 會轉換成兩種表示法

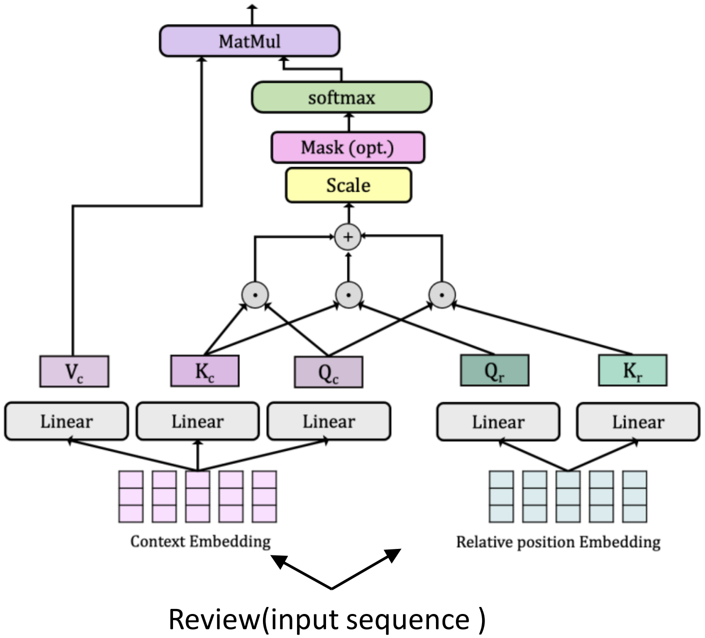
$\boldsymbol{c_i}$ (Content Embedding) & $\boldsymbol{p_i}$ (Position Embedding)

$$A_{i,j}^{c2c} = Q_i^c K_j^{c\,T}, A_{i,j}^{c2p} = Q_i^c K_{\delta(i,j)}^{p\,T}$$

$$A_{i,j}^{p2c} = K_j^c Q_{\delta(j,i)}^{p\,T}$$

$$A_{i,j} = A_{i,j}^{c2c} + A_{i,j}^{c2p} + A_{i,j}^{p2c}$$

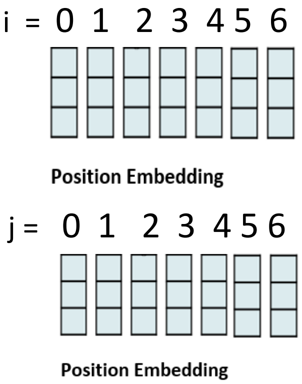
$$H = \left(\frac{A}{\sqrt{3d}}\right)V^c$$

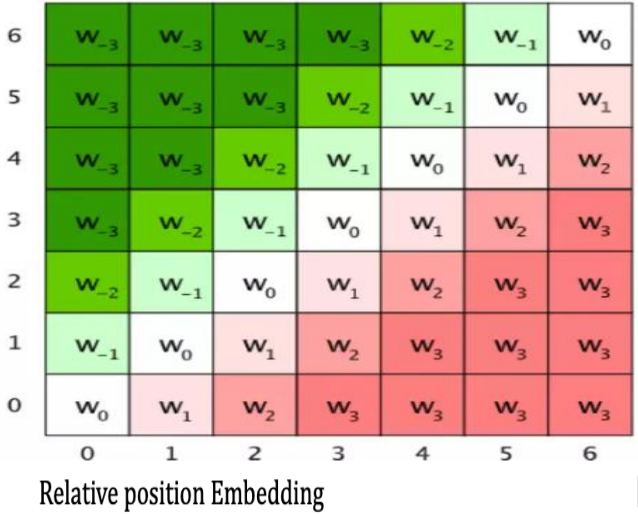


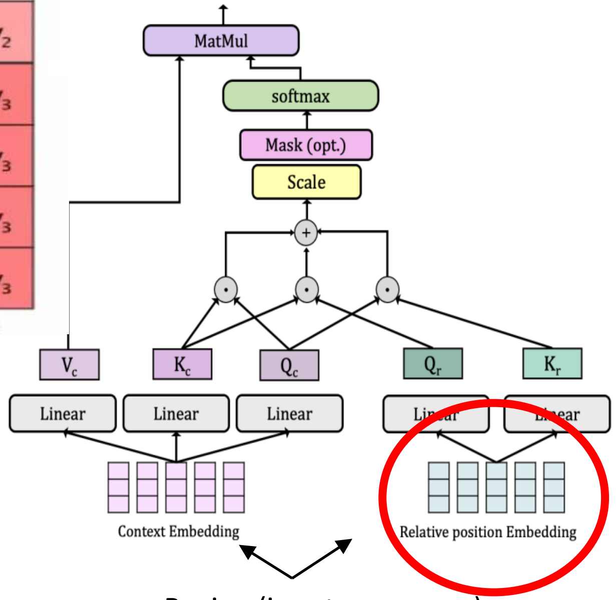
$\delta(i, j) \in [0, 2k)$ is the relative distance between I & j

Knowledge Injection Mechanisms(Disentangled Attention)

Window掃過不可以大於WINDOW





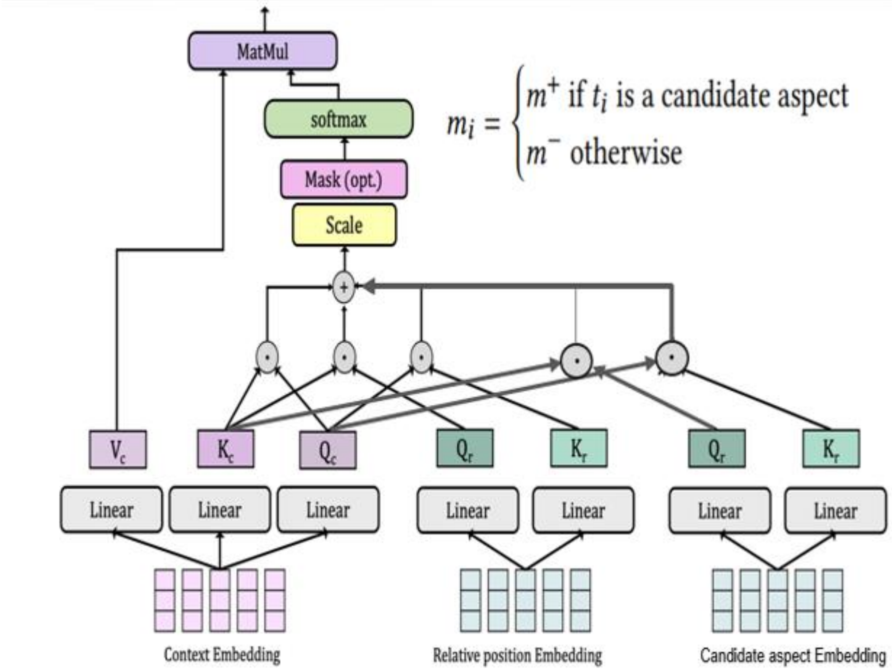


why is 2k ?

$\delta\,(i, j) \in [0, 2k)$ is the relative distance between I & j

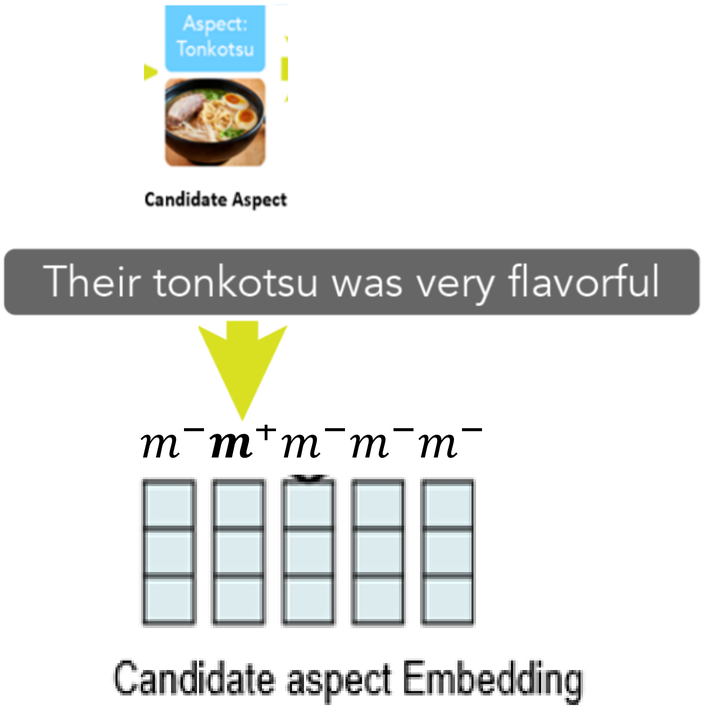
Review(input sequence)

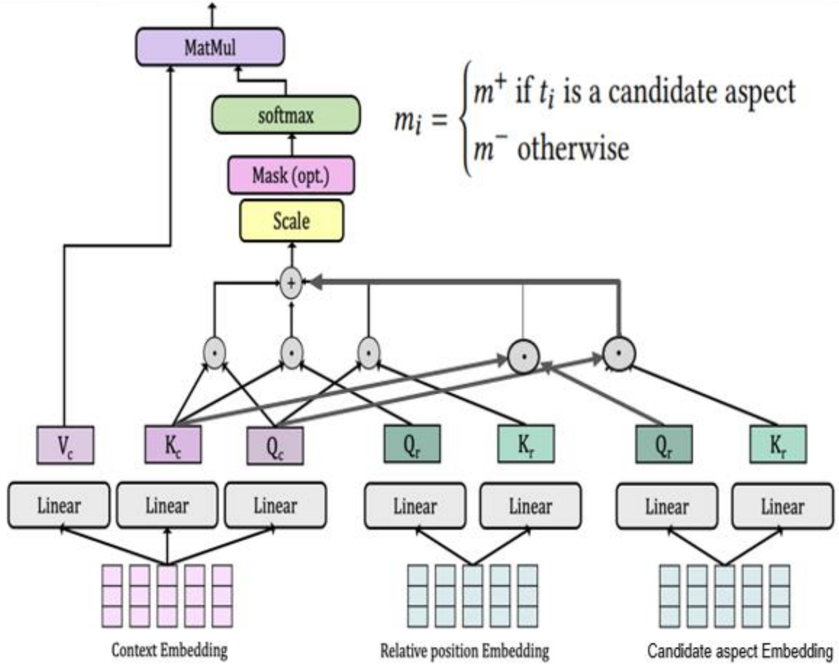
$$
\begin{aligned}
A_{i,j}^{c2c} &= Q_i^c K_j^{c\,T}, \; A_{i,j}^{c2p} = Q_i^c K_{\delta(i,j)}^{p\,T} \\
A_{i,j}^{p2c} &= K_j^c Q_{\delta(j,i)}^{p\,T} \\
A_{i,j}^{c2m} &= Q_i^c K_j^{m\,T}, \; A_{i,j}^{m2c} = Q_i^m K_j^{c\,T} \\
\hat{A}_{i,j} &= A_{i,j}^{c2c} + A_{i,j}^{c2p} + A_{i,j}^{p2c} \\
&\quad + A_{i,j}^{c2m} + A_{i,j}^{m2c} \\
\hat{H} &= \left(\frac{\hat{A}}{\sqrt{5d}}\right) V^c
\end{aligned}
\tag{4}
$$



Knowledge Injection Mechanisms(Disentangled Attention)

Allows for finer-grained control over the attention patterns exhibited in the model





Knowledge Injection Mechanisms(Disentangled Attention)

Allows for finer-grained control over the attention patterns exhibited in the model

Content

- Introduction
- Method
- Exprement
  - Baseline
  - Analysis
- Conclusion

Baseline

Model	(L, R)	(L, D)	(R, L)	(R, D)	(D, L)	(D, R)	**Mean**
KG-only	56.0	27.3	40.5	28.1	39.5	56.2	41.3
BERT	45.1 (3.6)	42.2 (0.5)	44.6 (1.9)	38.1 (1.3)	47.0 (2.2)	51.9 (2.2)	44.8 (2.0)
DeBERTa	54.3 (1.7)	40.5 (1.4)	47.5 (2.3)	39.6 (1.6)	47.1 (2.1)	54.5 (2.2)	47.3 (1.9)
DeBERTa-MA	61.5 (1.4)	40.2 (1.1)	43.4 (2.5)	38.0 (1.8)	47.2 (1.1)	62.0 (0.7)	48.7 (1.4)
DeBERTa-PT	66.0 (1.8)	41.0 (1.2)	49.7 (1.3)	38.5 (0.8)	52.5 (1.6)	64.9 (0.8)	52.1 (1.3)
BERT-PT	**66.4** (1.1)	**42.3** (1.1)	**49.9** (1.4)	39.5 (1.8)	**55.3** (1.4)	**65.8** (0.7)	**53.2** (1.3)

Pivot better than modified attention ?

Model	(L, R)	(L, D)	(R, L)	(R, D)	(D, L)	(D, R)	Mean
KG-only	56.0	27.3	40.5	28.1	39.5	56.2	41.3
BERT	45.1 (3.6)	42.2 (0.5)	44.6 (1.9)	38.1 (1.3)	47.0 (2.2)	51.9 (2.2)	44.8 (2.0)
DeBERTa	54.3 (1.7)	40.5 (1.4)	47.5 (2.3)	39.6 (1.6)	47.1 (2.1)	54.5 (2.2)	47.3 (1.9)
DeBERTa-MA	61.5 (1.4)	40.2 (1.1)	43.4 (2.5)	38.0 (1.8)	47.2 (1.1)	62.0 (0.7)	48.7 (1.4)
DeBERTa-PT	66.0 (1.8)	41.0 (1.2)	49.7 (1.3)	38.5 (0.8)	52.5 (1.6)	64.9 (0.8)	52.1 (1.3)
BERT-PT	**66.4** (1.1)	**42.3** (1.1)	**49.9** (1.4)	39.5 (1.8)	**55.3** (1.4)	**65.8** (0.7)	**53.2** (1.3)

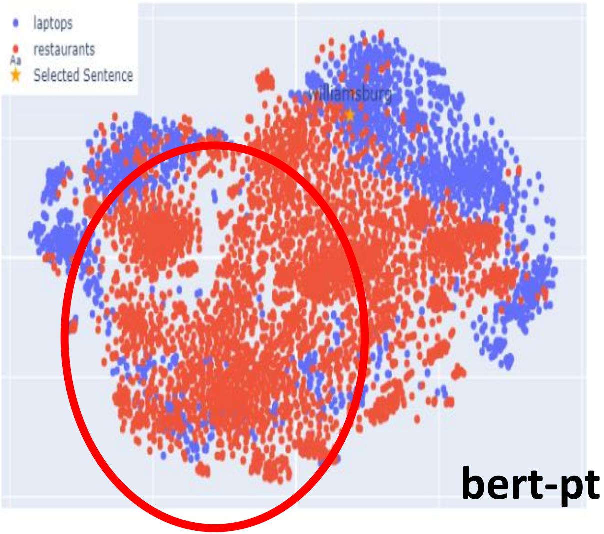
1. 在Knowledge inject的時候，會造成相對位置的訊息被干擾了
2. 因為在Pivot 的提示性還是大於新增候選的表示法還要高

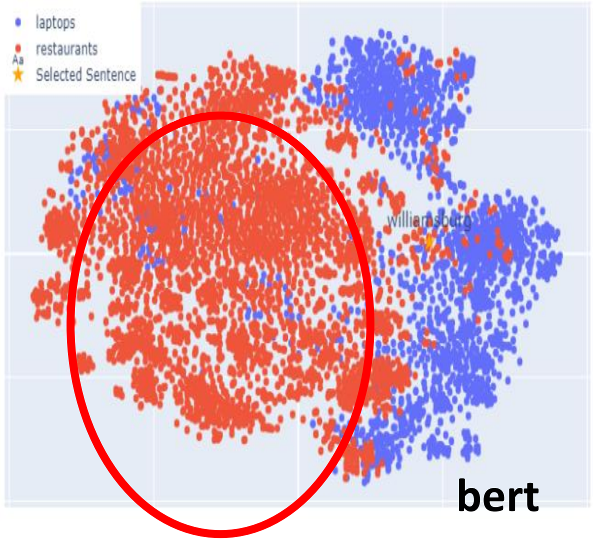
Stochastic & Deterministic insertion

Model	Method	(L, R)	(L, D)	(R, L)	(R, D)	(D, L)	(D, R)
BERT-PT	S	**66.4**	42.3	49.9	39.5	**55.3**	**65.8**
BERT-PT	D	47.3	**44.0**	**52.5**	41.8	48.8	52.8
DeBERTa-PT	S	66.0	41.0	49.7	38.5	52.5	64.9
DeBERTa-PT	D	57.9	43.5	51.8	**43.2**	50.1	56.8
DeBERTa-MA	S	61.5	40.2	43.4	38.0	47.2	62.0
DeBERTa-MA	D	51.6	41.7	44.1	38.5	47.1	55.4

在跨Domain的時候具有顯著的提升或是效果差異小

Analysis





這張圖顯示 Bert-PT 可以有效幫助，銜接了 Domain之間的差距並且Domain之間有更多重疊

Conclusion

- New approach how to **auto constructing** and **determining inject** domain-specific KGs
- 並且運用了潛在的前後文關係來幫助 Cross Domain