**! 這不是GNN**

**待補充:**

1. **Embedding**
2. **Pooling**
3. **Spatial**
4. **Feature map**

**Knowledge Graph 介紹**

1.常用來處理NLP task

(1) question answering

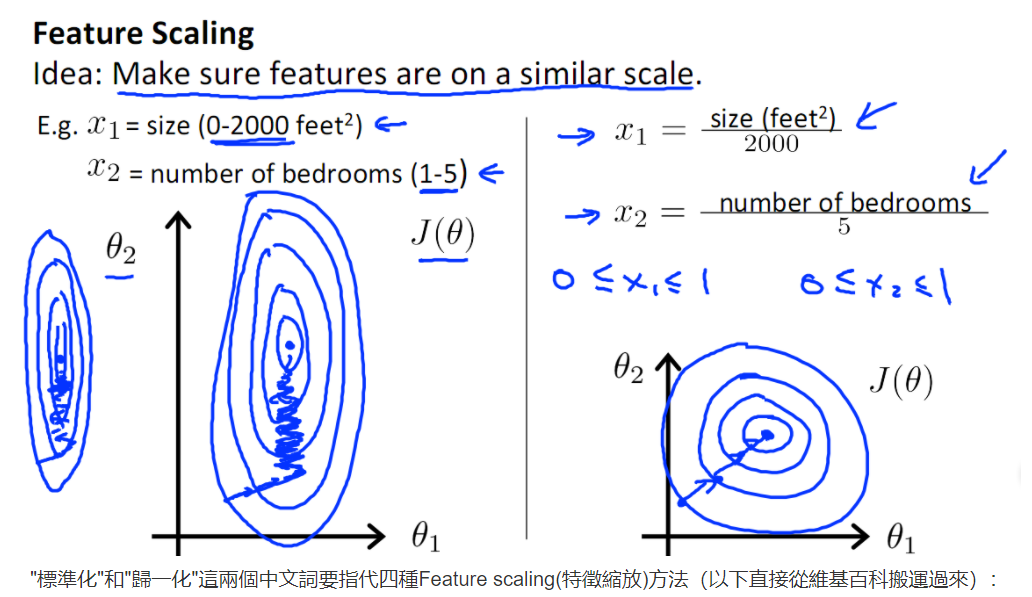
(2) dialog system

(3) relation extraction

(4) recom-mender systems

2.它的特性

(1) 將事實關聯都集合起來 scaling (做一個特徵縮放) 是一件非常困難的事情



(2)

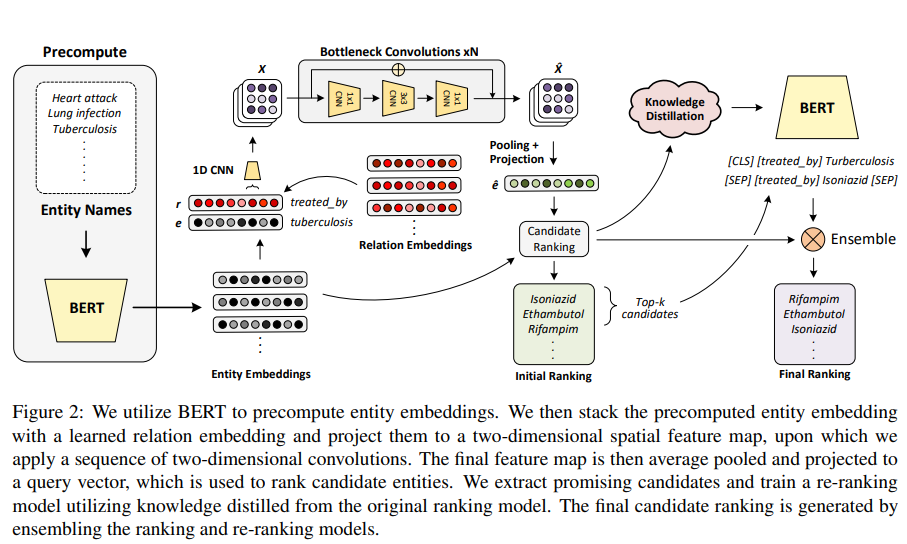
BLABLABLA DATASET不太重要先不理

**本篇論文主要做的三件事**

1.使用 深度卷積 的模型架構 來讓 文字EMBEDDING 更有效率

2.re-ranking，將評分model 從 大 model 到 小 model (兩個MODEL )

3.將兩個KG Dataset 丟到 KGC

**關於Model 的完整架構** 

步驟

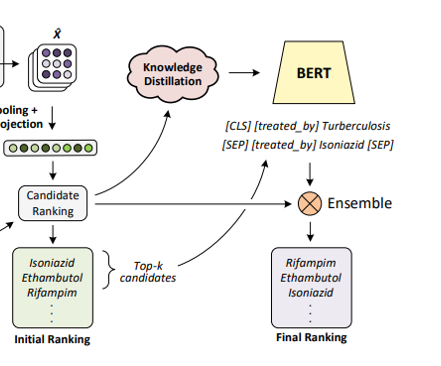
1. 優化BERT
2. 使用學習過的關係EMBEDDING堆積預處理的EMBEDDING
3. 根據”連續的”(Sequence) 二維(two dimensional) 卷積(convolution) 將其投影到二維的two-dimensional spatial feature map
4. 最後的feature map 再 平均 pooling 還有 投影到 一個 query vector，並將此用來 排名候選人 (rank candidate entities)

步驟後做的事情

再 train 一個 ranking model (叫做 re-ranking model)，從原本的model取得優化過的知識

最後的結果

由 ranking model 以及 re-ranking model 共同所決定



以上再講這邊這幾件事情

**接著來講重要的entity-ranking and entity-reranking 分別怎麼實作**

1. Entity-ranking的training
   1. 我們把剩餘的connections 加上去每個bottleneck block，進而改善神經網路
   2. 使用線性代數 Matrix的好處，讓大家快速運算，(A fully connected layer followed by a PReLU nonlin-earity) (如何計算分數, scoring 的部分)
2. Training
   1. 1vsALL training strategy with binary cross-entropy loss function
   2. 加上一層sigmoid去降低每一個enitity的機率
   3. Adam Optimizer

(1) decoupled weight de-cay regularization

(2) label smoothing

* 1. 1vsALL training strategy with binary cross-entropy loss function
  2. 加上一層sigmoid去降低每一個enitity的機率
  3. Adam Optimizer

(1) decoupled weight de-cay regularization

(2) label smoothing

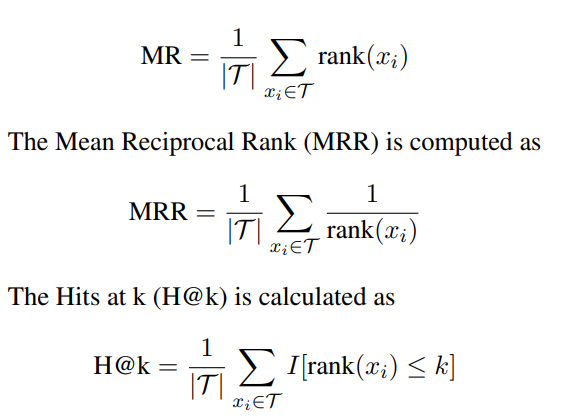
從entity-reranking 那裏開始!

1. Entity re-ranking的training
2. Student re-ranking network 設計成一個triplet classification (ei, rj, ek) model，並全部傳入“完整的”triplet fact， (在這之中不放Incomplete fact )

這樣會導致三個綁再一起的elements能夠互相學習

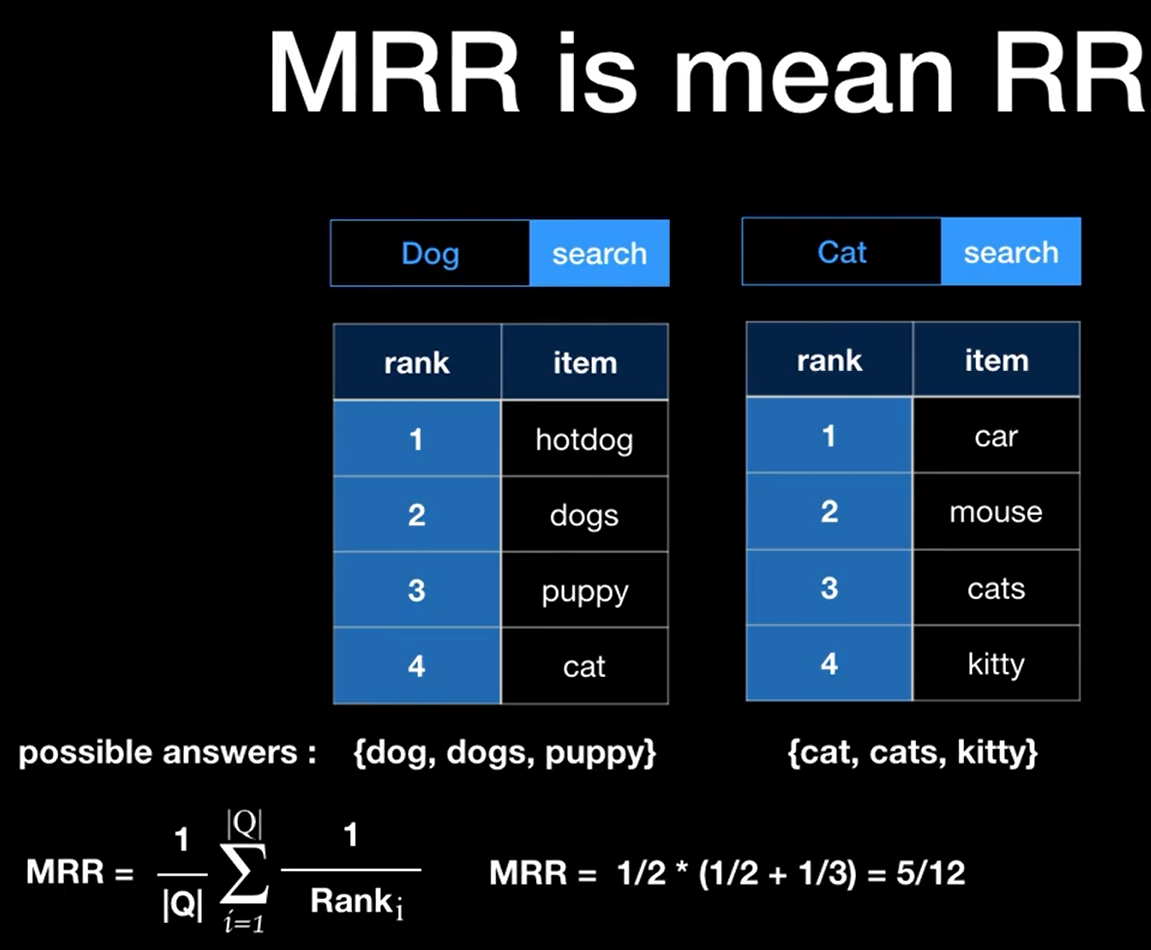
2.

評分方式



MRR 越高越好，越高代表答對率越高

MR 是MRR的倒數，因此越低代表答對率越高



MR 越低越好，因為它是MRR的倒數，這篇論文在做的事情就是

算出來，多一個取平均而已

