# 相关性：BERT模型及其推理

王树森

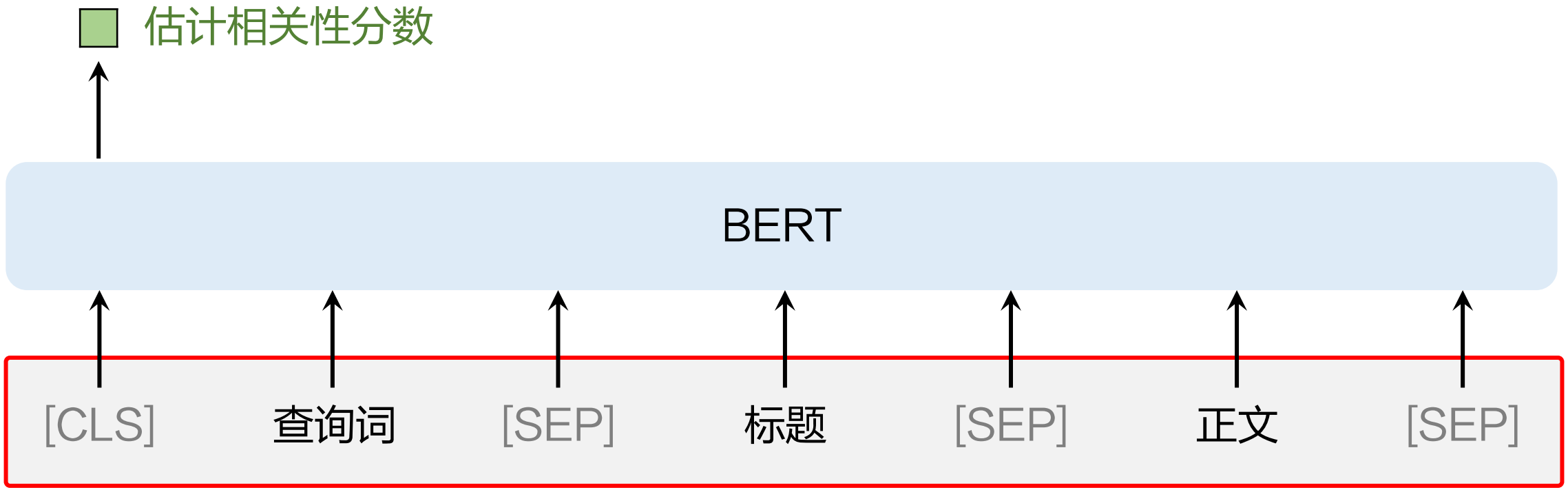
http://wangshusen.github.io/

# 相关性模型

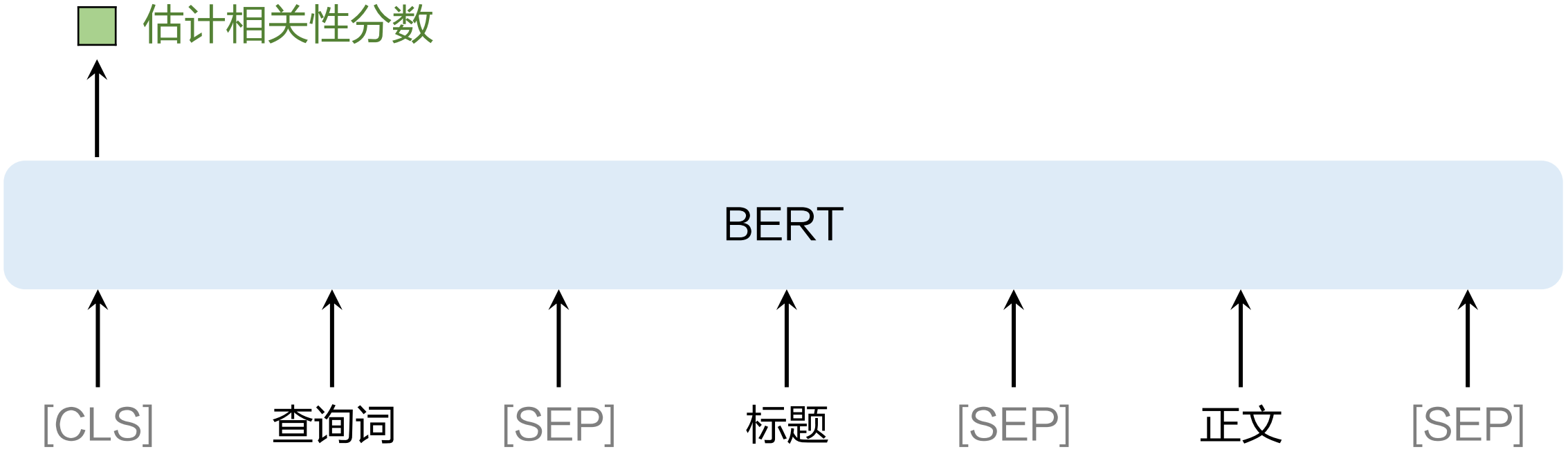
- 现代搜索引擎普遍使用 BERT 模型计算 $q$ 和 $d$ 的相关性。
- 交叉 BERT 模型（单塔）准确性好，但是推理代价大，通常用于链路下游（精排、粗排）。
- 双塔 BERT 模型不够准确，但是推理代价小，常用于链路上游（粗排、召回海选）。
- 训练相关性 BERT 模型的 4 个步骤：预训练、后预训练、微调、蒸馏。（下节课）

# 交叉 BERT 模型

# 交叉BERT模型



# 交叉BERT模型




- token embedding：表征 token 本身
- position embedding：位置编码，表征token的序
- segment embedding：用于区分查询词、标题、正文

每个token被表征为 3 个向量，取加和作为 token 的表征。

# 字粒度 vs 字词混合粒度

- 字粒度：将每个汉字/字符作为一个 token。
  - 词表较小（几千），只包含汉字、字母、常用字符。
  - 优点：实现简单，无需做分词。

# 字粒度 vs 字词混合粒度

- 字粒度：将每个汉字/字符作为一个 token。
- 字词混合粒度：做分词，将分词结果作为 tokens。
  - 词表较大（几万、十几万），包含汉字、字母、常用符号、常用中文词语、常用英文单词。
  - 与字粒度相比，字词混合粒度得到的序列长度更短（即 token 数量更少）。
  - 参见 WoBERT （https://github.com/ZhuiyiTechnology/WoBERT）

# 字粒度 vs 字词混合粒度

- 字粒度：将每个汉字/字符作为一个 token。
- 字词混合粒度：做分词，将分词结果作为 tokens。
- 序列更短（token 数量更少）有什么好处？
  - BERT 的计算量是 token 数量的超线性函数。
  - 为了控制推理成本，会限定 token 数量，例如 128 或 256。
  - 如果文档超出 token 数量上限，会被截断，或者做抽取式摘要。
  - 使用字词混合粒度，token 数量更少，推理成本降低。（字粒度需要 256 token，字词混合粒度只需要 128 token。）

# 交叉BERT模型的推理降本

- 对每个 $(q, d)$ 二元组计算相关性分数 score，代价很大。
- 用 Redis 这样的 KV 数据库缓存 $\langle q, d, \text{score} \rangle$。
  - $(q, d)$ 作为 key，相关性分数 (score) 作为 value。
  - 如果命中缓存，则避免计算。
  - 如果超出内存上限，按照 least recently used (LRU) 清理缓存。

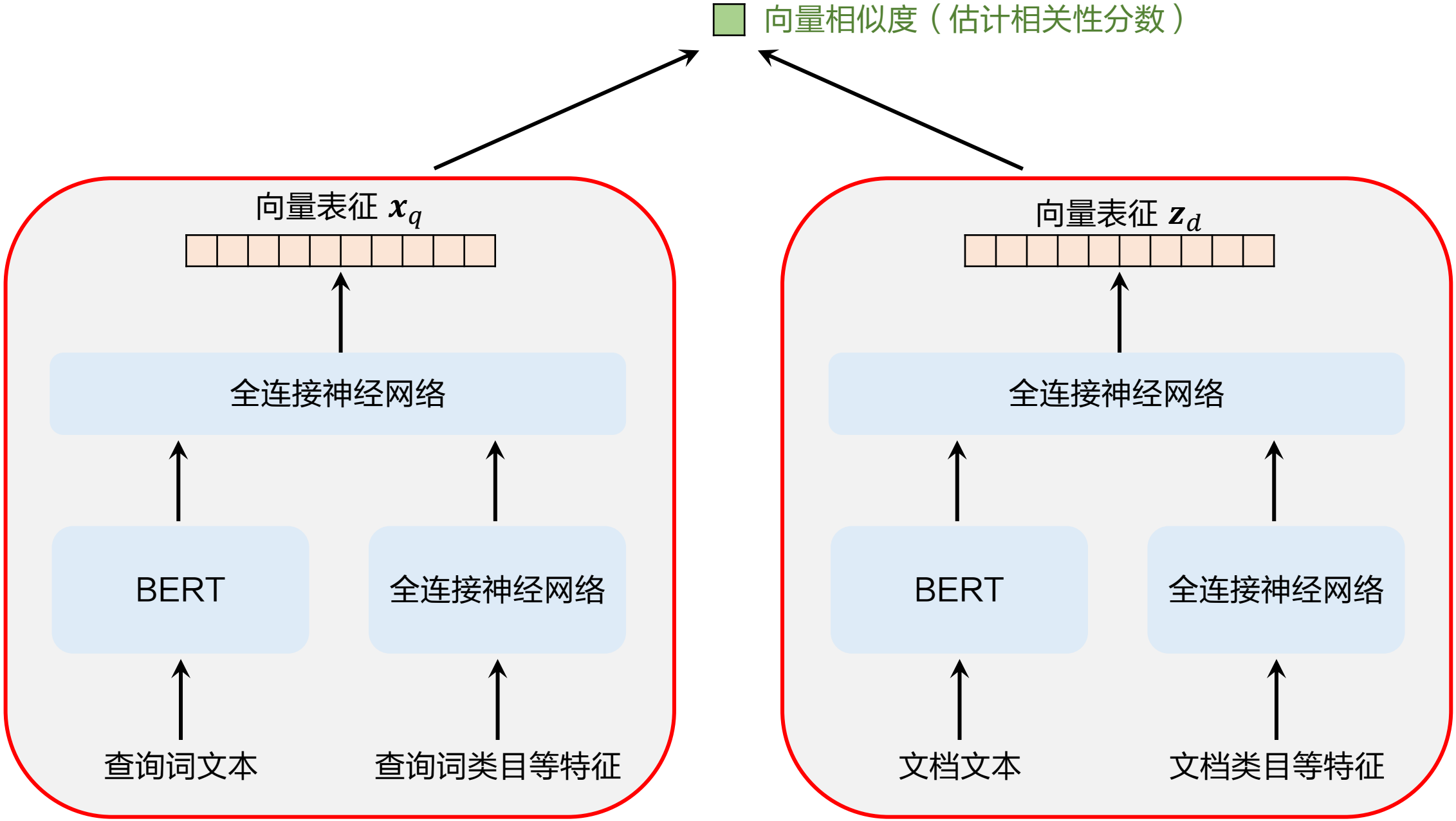
# 交叉BERT模型的推理降本

- 对每个 $(q, d)$ 二元组计算相关性分数 score，代价很大。
- 用 Redis 这样的 KV 数据库缓存 $\langle q, d, \text{score} \rangle$。
- 模型量化技术，例如将 float32 转化成 int8。
  - 训练后量化 (post-training quantization，PTQ)。
  - 训练中量化 (quantization-aware training，QAT)。

# 交叉BERT模型的推理降本

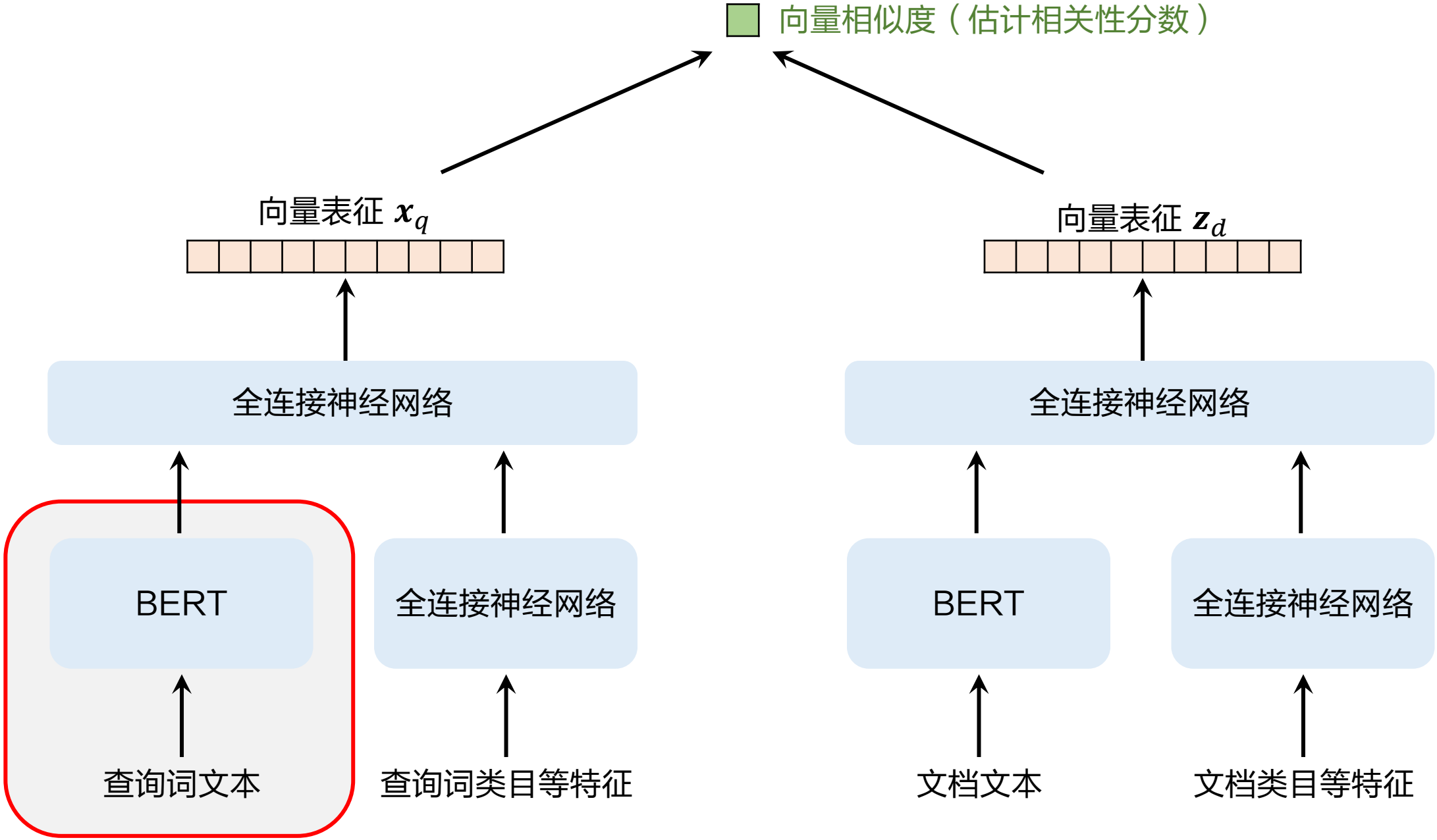
- 对每个 $(q, d)$ 二元组计算相关性分数 score，代价很大。
- 用 Redis 这样的 KV 数据库缓存 $\langle q, d, \text{score} \rangle$。
- 模型量化技术，例如将 float32 转化成 int8。
- 使用文本摘要降低 token 数量。
  - 如果文档长度超出上限，则用摘要替换文档。
  - 在文档发布时计算摘要。可以是抽取式，也可以是生成式。
  - 如果摘要效果好，可以将 token 数量上限降低，比如从 128 降低到 96。

# 双塔 BERT 模型



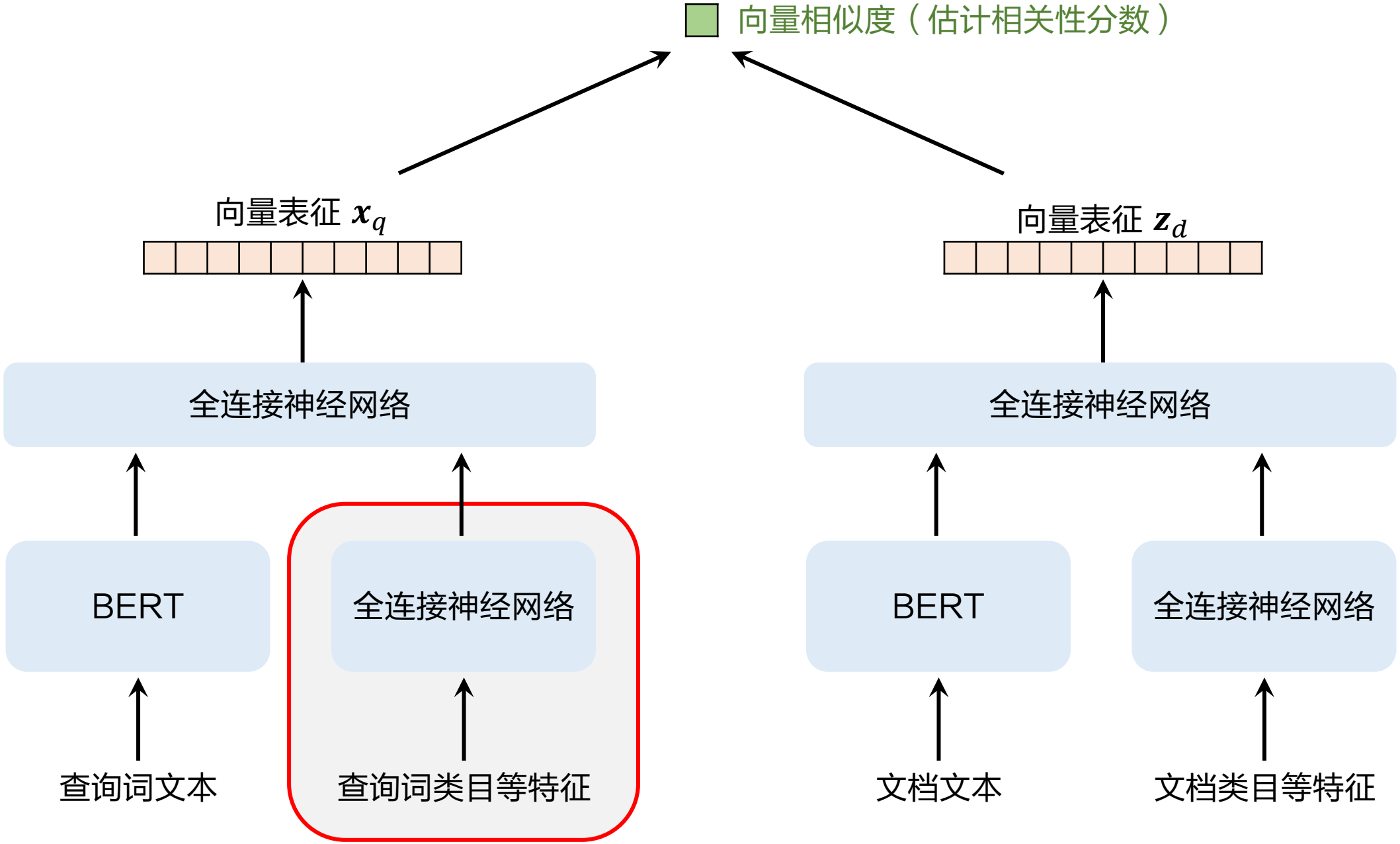

交叉BERT模型

双塔BERT模型



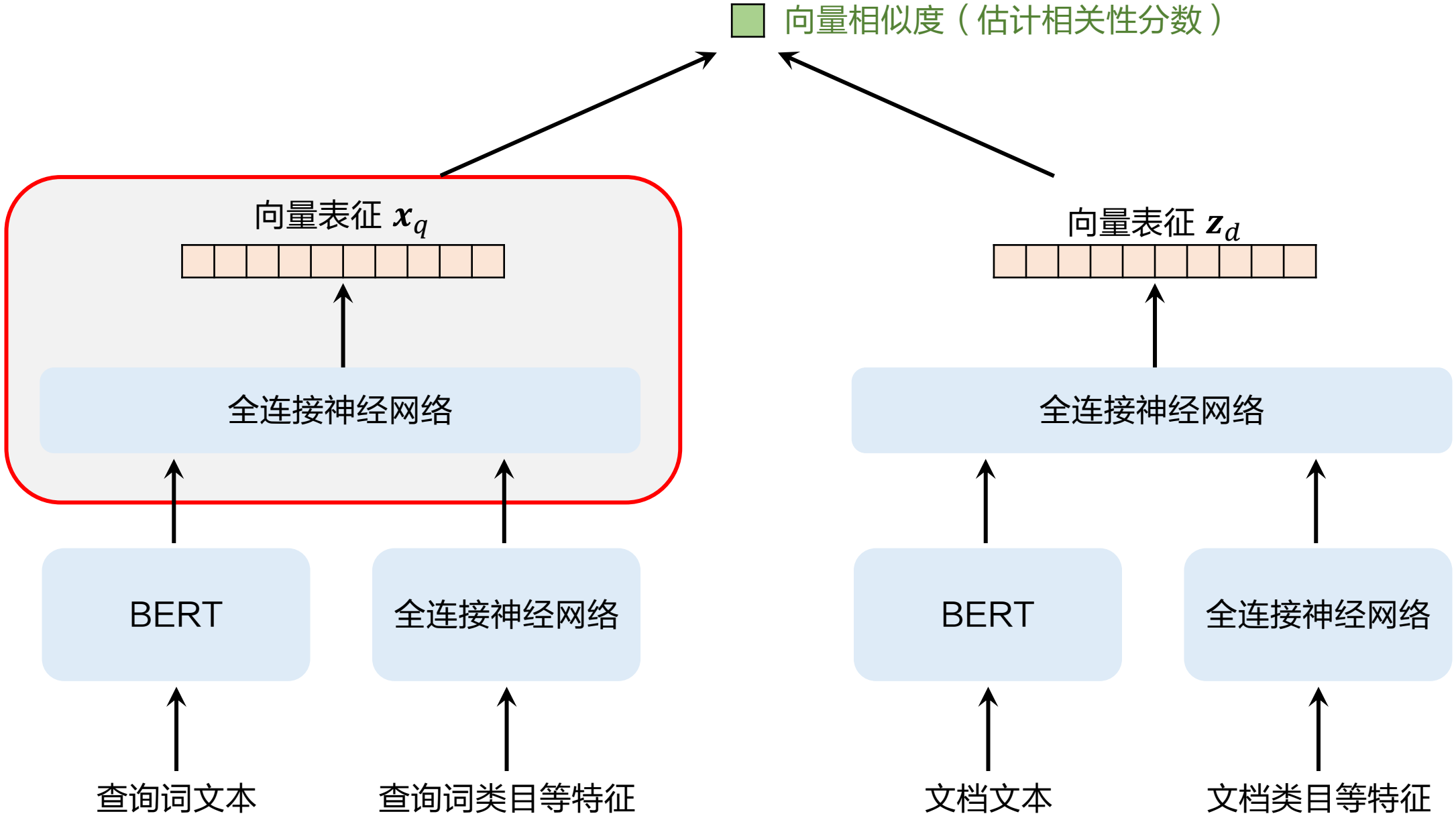

交叉BERT模型

双塔BERT模型



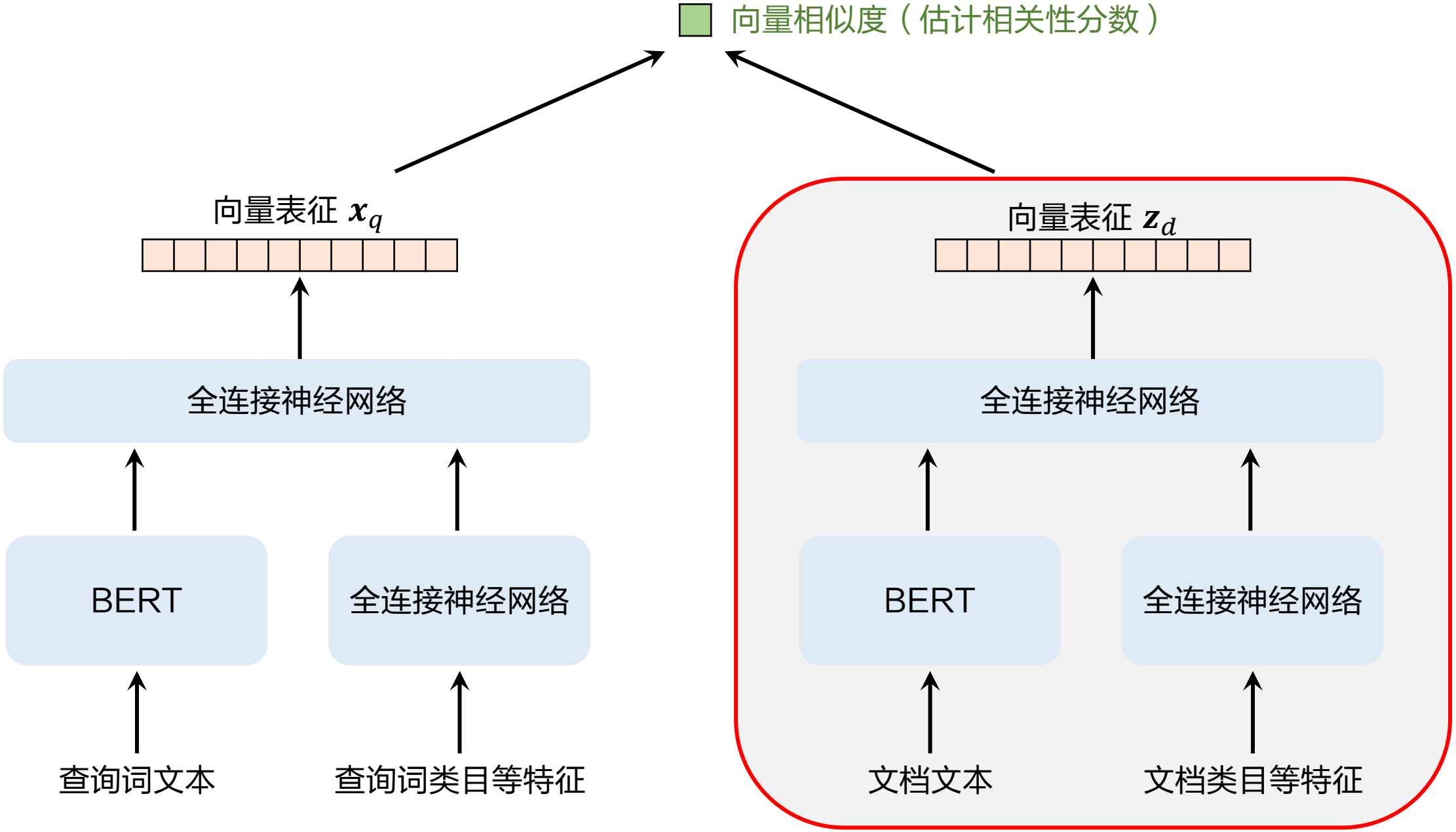

交叉BERT模型

双塔BERT模型



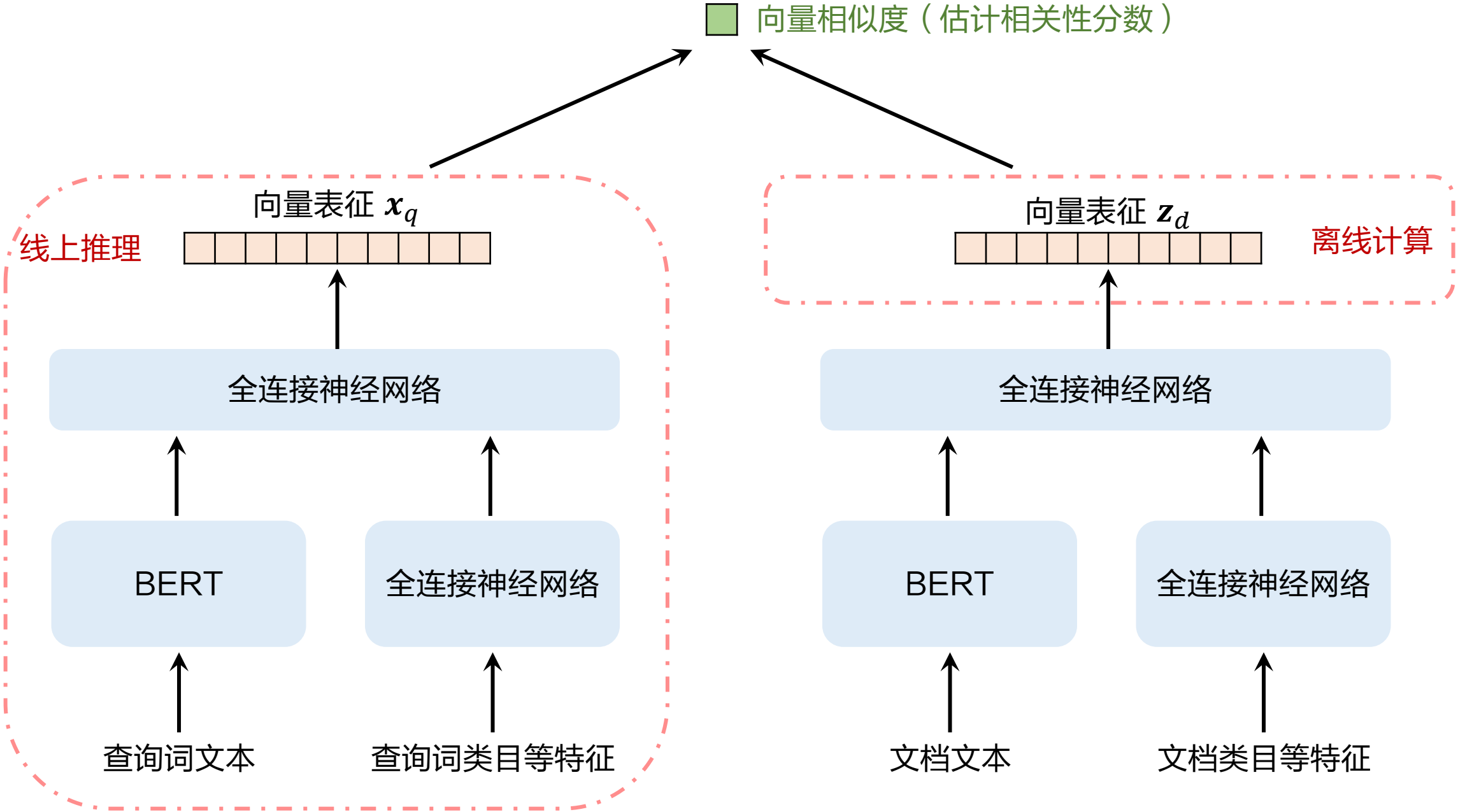

交叉BERT模型

双塔BERT模型




交叉BERT模型

双塔BERT模型




交叉BERT模型

双塔BERT模型

# 总结

# 交叉BERT模型

- 交叉 BERT 模型（单塔）准确性高，计算量大，适用于精排、粗排。
- 字词混合粒度分词降低序列长度（即 token 数量）。
- 用 KV 内存数据库缓存 $\langle q, d, \text{score} \rangle$，可以避免大部分计算。
- 用模型量化技术，把 float32 转化成 int8，降低推理成本。
- 设置较小的 token 数量上限，将长文档替换成摘要。

# 双塔BERT模型

- 双塔 BERT 模型准确性低，计算量小，适用于粗排、召回海选。
- 事先离线计算每篇文档 $d$ 的向量表征 $\boldsymbol{z}_d$，将 $(d, \boldsymbol{z}_d)$ 存入哈希表。
- 线上计算 $(q, d)$ 的相关性时，给定候选文档 $d$，从哈希表中读取它的向量表征 $\boldsymbol{z}_d$。
- 线上计算查询词 $q$ 的向量表征 $\boldsymbol{x}_q$，然后计算内积 $\langle \boldsymbol{x}_q, \boldsymbol{z}_d \rangle$，作为 $(q, d)$ 相关性分数。

# Thank You!

http://wangshusen.github.io/

# Anchor Query

- 给定文档 $d$，生成相关的查询词 $q_1, \cdots, q_k$。
  - 构造一个 $(q, d)$ 数据集，$q$ 与 $d$ 相关性高。
  - 训练一个生成模型（比如 Transformer），输入 $d$，生成 $q$。
  - 做后处理，用相关性模型排除与 $d$ 相关性不够高的 $q$。

# Anchor Query

- 给定文档 $d$，生成相关的查询词 $q_1, \cdots, q_k$。
- 与 $d$ 高相关的查询词 $\{q\}$ 叫做 anchor query，作为文档内容输入相关性 BERT 模型。
- 可以将 anchor query 看做文档的关键词或极简的摘要，起到“打标签”的作用。
- 如果 anchor query 质量高，可以让 BERT 更准确预测相关性。
- 对交叉 BERT 模型的提升有限；对双塔 BERT 模型提升较大。