PPT顺序不太对，我按照论文顺序做了调整

Assertion-based QA with Question-Aware Open Information Extraction

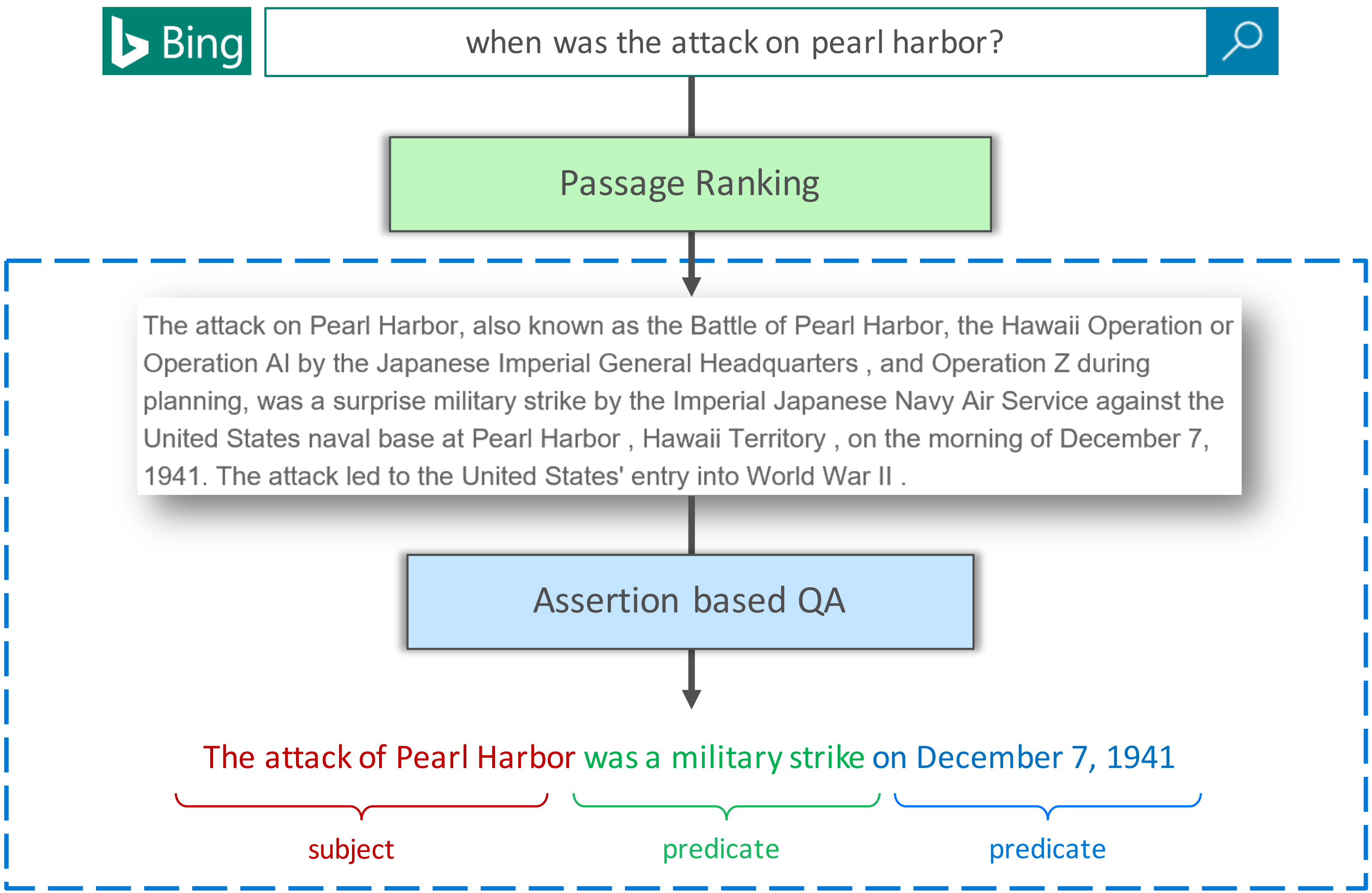
Zhao Yan[1], Duyu Tang[2], Nan Duan[2], Shujie Liu[2], Wendi Wang[2], Daxin Jiang[2], Ming Zhou[2], Zhoujun Li[1]

1. Beihang University
2. Microsoft

Presenter: **Xiaocheng Feng** on behalf of Duyu Tang

Document-based QA

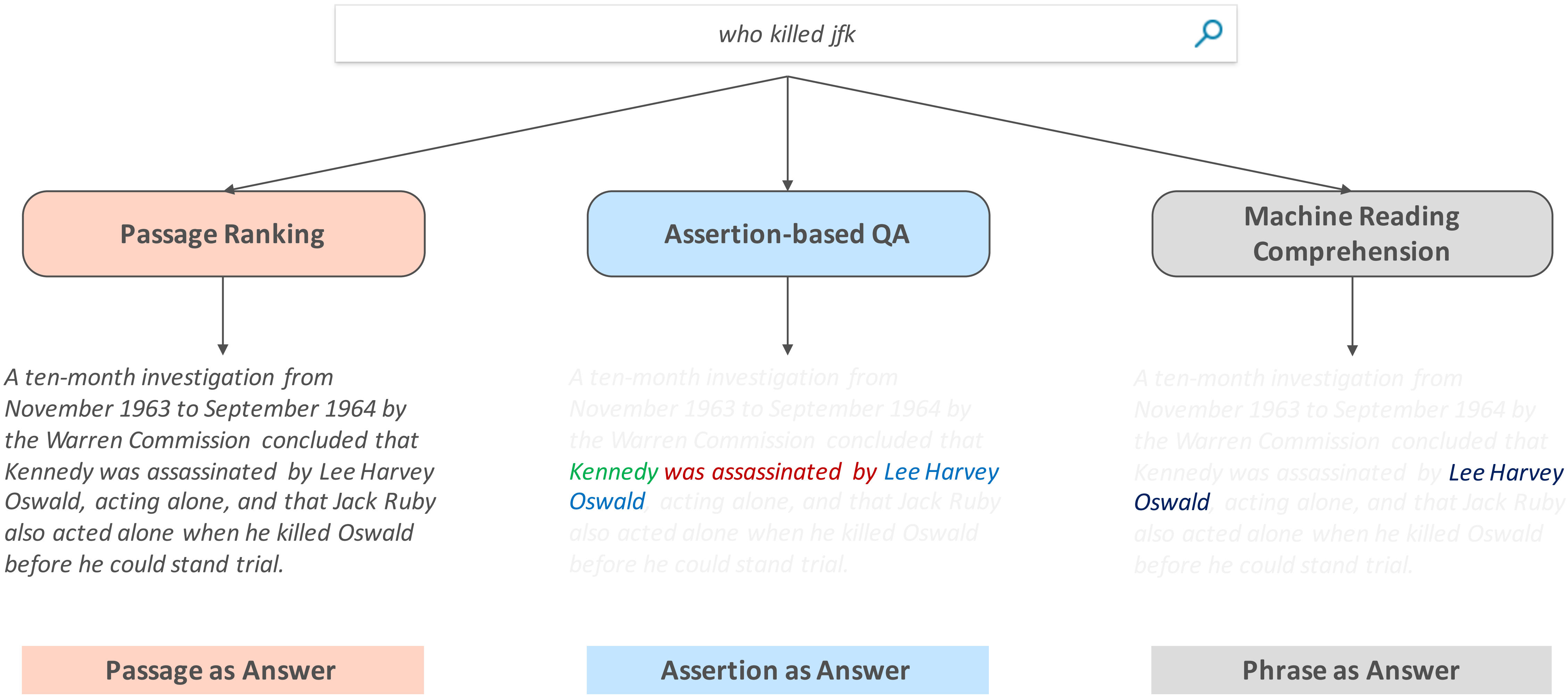
知识问答：从互联网上找到最相关的文档
PBQA：从文档里找出最相关的段落
SBQA：从段落里找到最相关的句子
MRC：从段落里找出可以当答案的词组
ABQA：从段落里找出SPO断言



The focus of this talk

3 Types of PassageQA Approaches

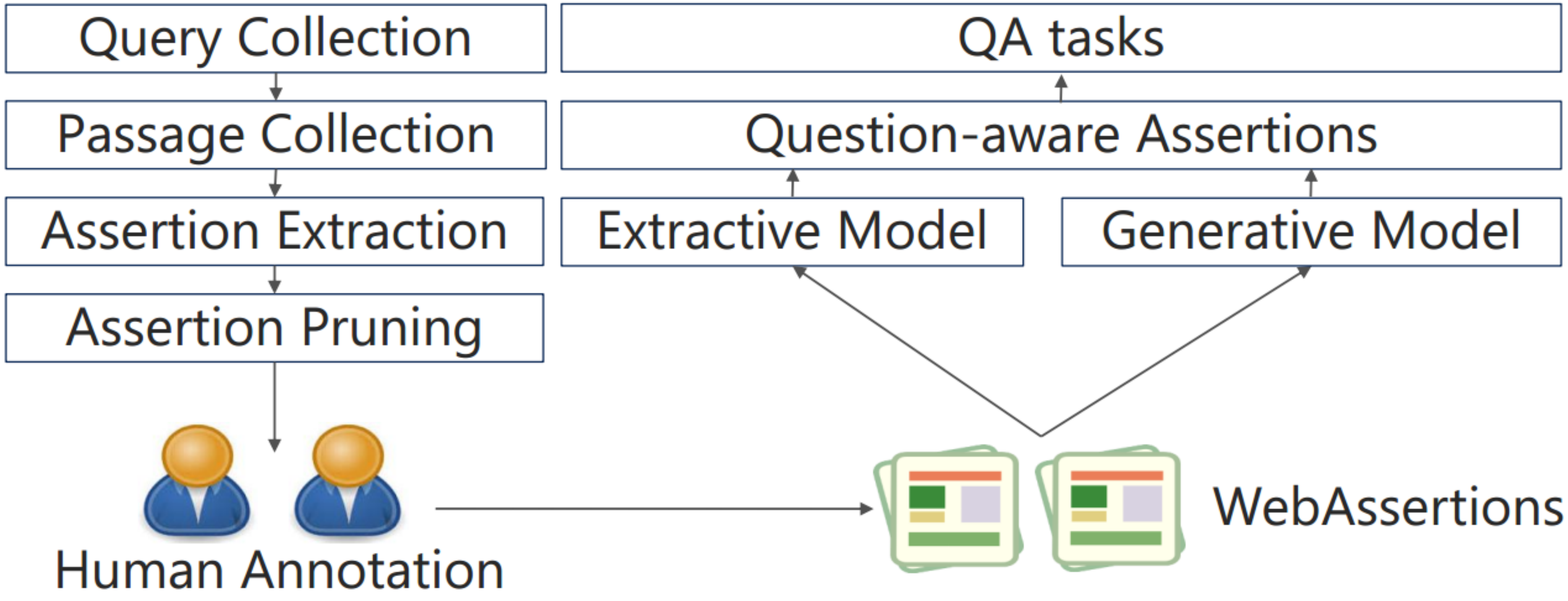
这个例子不太好，不一定是有这个句子
不过目前这篇文章的断言只能从一个句子里面去抽取，没有实现跨句抽取



The big picture

这张图指的是ABQA数据集的构建及ABQA在其它QA任务中的应用

来自搜索引擎的日志



第二个实验，ABQA+PBQA

第一个实验，测试ABQA本身

Dataset: WebAssertion

数据集构造：1.利用开源的OIE工具：ClausIE算法——基于依存树的人工语法模版来检测和抽取基于结构的断言；
2.is–a规则

每个问题–段落对能平均产生6.41个断言，
每个问题由平均6.00个词组成，
每个段落由平均39.33个词组成，
每个断言由8.62个词组成

- An assertion is annotated as 1 if
  1. it correctly answers the question and
  2. meantime has a complete meaning

Statistics of WebAssertions	
# of question-passage	55,960
# of question-passage-assertion	358,427
Avg. assertions / question-passage	6.41
Avg. Words / question	6.00
Avg. Words / passage	39.33
Ave. Words / assertion	8.62

Question	when will shanghai disney open
Passage	the Disney empire's latest outpost, Shanghai Disneyland, will open in late 2015, reports the associated press.
Label	**Assertion**
0	<the Disney empire's latest outpost; is; Shanghai Disneyland >
0	<the Disney empire's latest outpost; will open; in late 2015>
0	<the associated press; reports; the Disney empire's latest outpost will open in late 2015>
1	<Shanghai Disneyland; will open; in late 2015 >

do not answer the question

A bad assertion

所以只有断言4可以被标注label 1：精准地回答了问题

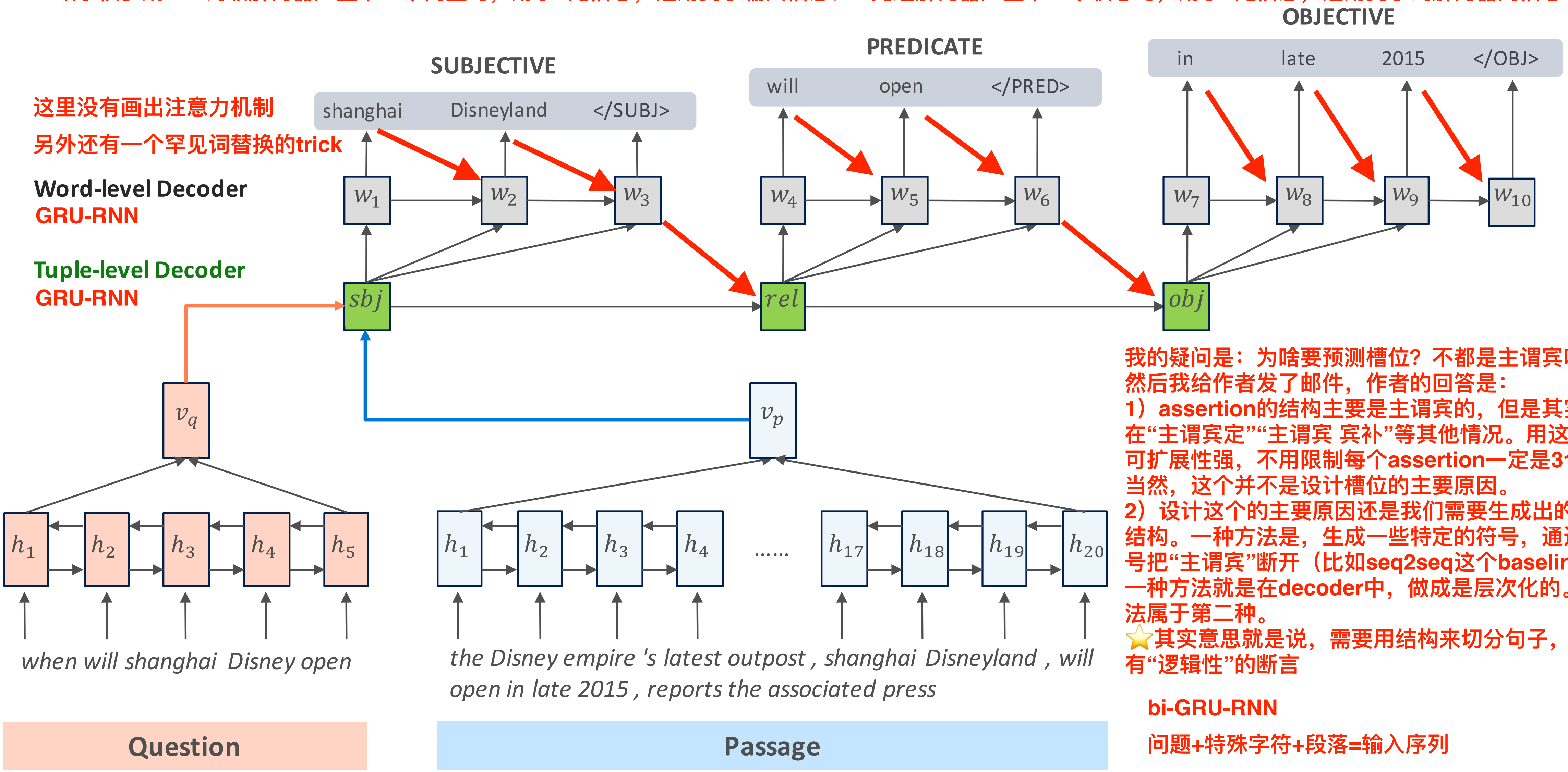
An annotated example

ABQA: A Generative Approach with Hierarchical Decoder

PPT缺了很多线：1.词级解码器产生下一个向量时，用了3处信息，还用到了输出信息；2.元组解码器产生下一个状态时，用了2处信息，还用到了词解码器的信息

这里没有画出注意力机制

另外还有一个罕见词替换的trick



我的疑问是：为啥要预测槽位？不都是主谓宾嘛？
然后我给作者发了邮件，作者的回答是：
1）assertion的结构主要是主谓宾的，但是其实也存在“主谓宾定”“主谓宾 宾补”等其他情况。用这样的方法，可扩展性强，不用限制每个assertion一定是3个部分。当然，这个并不是设计槽位的主要原因。
2）设计这个的主要原因还是我们需要生成出的内容具有结构。一种方法是，生成一些特定的符号，通过特定的符号把“主谓宾”断开（比如seq2seq这个baseline）。还有一种方法就是在decoder中，做成是层次化的。我们的方法属于第二种。
⭐其实意思就是说，需要用结构来切分句子，以便生成有“逻辑性”的断言

bi-GRU-RNN

问题+特殊字符+段落=输入序列

ABQA: A Ranking based Approach

the Disney empire 's latest outpost , shanghai Disneyland , will open in late 2015 , reports the associated press .

Rule-based Open IE

词级别：公共词数量、IBM model 1训练的相似度特征
词组级别：基于语义和翻译的特征
句级别：基于两个CNN计算相似度，基于RNN和GRU的向量表示——最后4位隐藏状态向量和双向向量；CNN、RNN和bi-GRU的参数都是预先训练好的

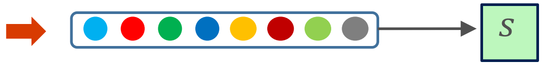
the Disney empire 's latest outpost	is	shanghai Disneyland
the Disney empire 's latest outpost	will open	in late 2015
the associated press	reports	the Disney empire 's latest outpost will open in late 2015
Shanghai Disneyland	will open	in late 2015

Assertion Ranking **用了基于决策森林的开源算法LambdaMART**

features at different levels

- Word level
- Phrase level
- Sentence level
- …

1.	Shanghai Disneyland	will open	in late 2015
2.	the Disney empire 's latest outpost	will open	in late 2015
3.	the associated press	reports	the Disney empire 's latest outpost will open in late 2015
4.	the Disney empire 's latest outpost	is	shanghai Disneyland



Features

这里的语料与论文中提到的不一致：含义就是基于翻译法计算了词、词组之间的相似度

论文中对于前两组特征描述的非常少，基本上就是引用别人的模型来计算一个特征

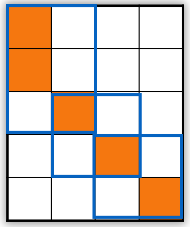
- **Word-to-Word translation model**

12M <question, related question> pairs from WikiAnswers (English)

17M <question, related question> pairs from Baidu Zhidao (Chinese)

Question-Question Pairs

Word alignment



- **Word embedding**

10M sentences from English Wikipedia (English)

10M sentences from Chinese Wikipedia (Chinese)

Wikipedia Sentences

Word embedding

- **Paraphrasing**

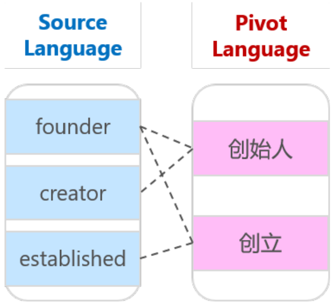
43.8M paraphrase pairs (English)

11.3M paraphrase pairs (Chinese)

不是应该是词组嘛?

Bilingual Sentence Pairs

Phrase table extraction

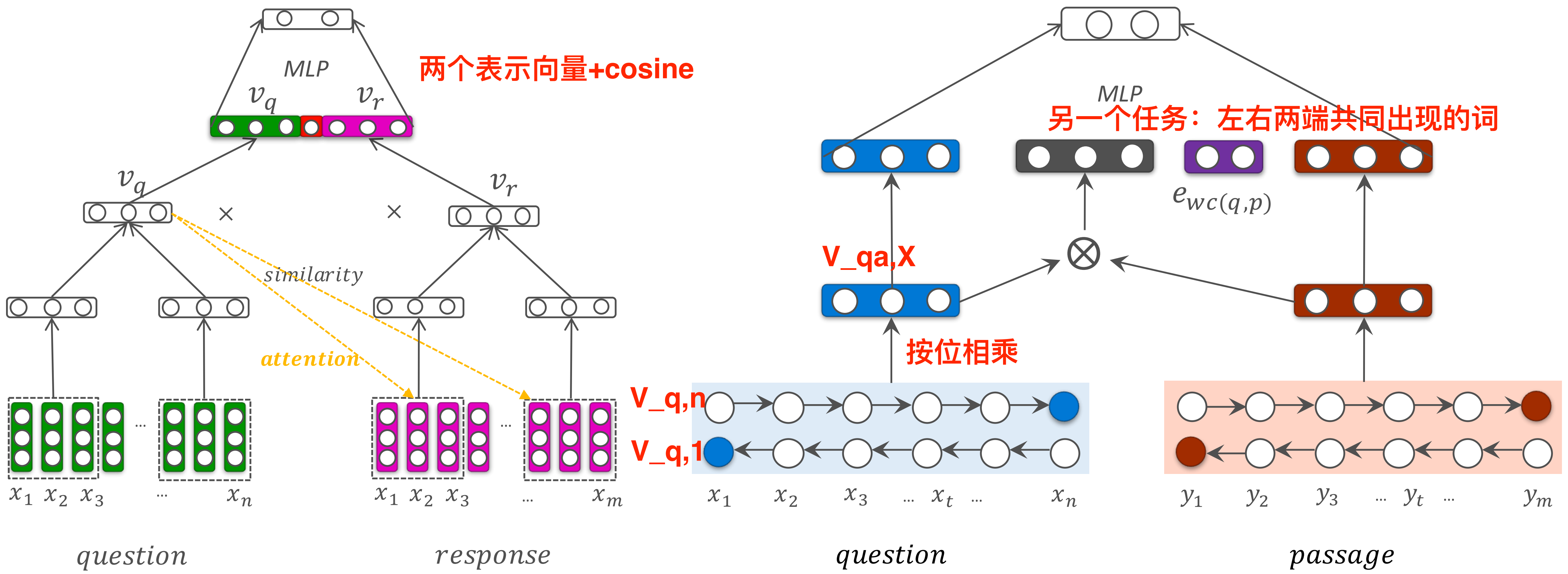


$$p(s_j|s_i) = \sum_t p(t|s_i) \cdot p(s_j|t)$$

founder ||| creator ||| 0.01214179
founder ||| founded ||| 0.00694428
founder ||| start ||| 0.00280628
founder ||| set up ||| 0.00088949
founder ||| established ||| 0.00065527
founder ||| pioneer ||| 0.00047020
...

Features (cont.) 这里的语料与论文中提到的也不一致，特征构建也不一致...

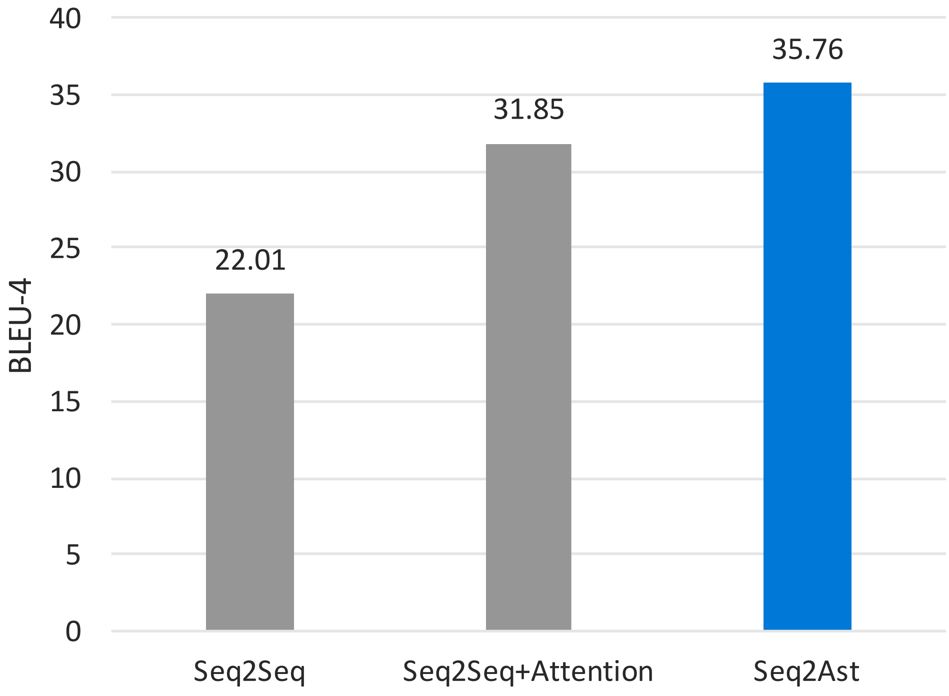
- **Compute relevance between questions and responses**
  - Use **10M** <Question, Answer> pairs as training data to handle question queries
  - Use **10M** <Sentence, Next Sentence> pairs as training data to handle non-question queries



Evaluation on ABQA

BLEU-4:针对文本生成任务的评价指标，
http://blog.csdn.net/qq_21190081/article/details/53115580
训练、开发和测试：8:1:1

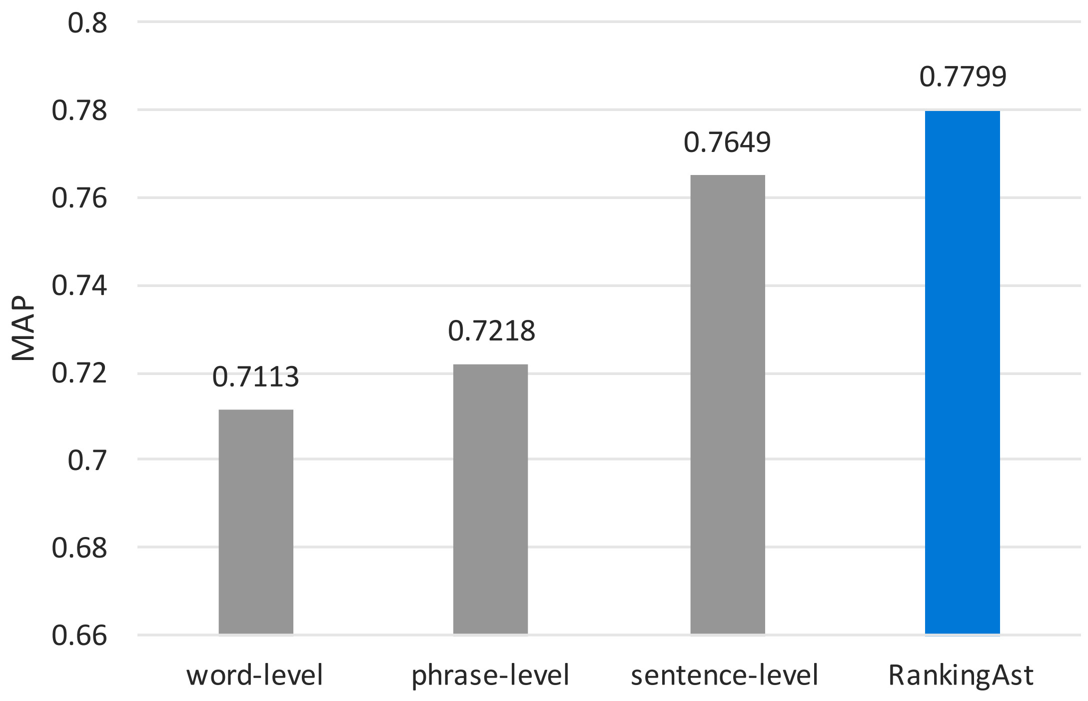
- Evaluation on Generation
  - Our generative model is **Seq2Ast**



Compare to sequence-to-sequence learning methods

Evaluation on ABQA MAP：平均准确率

- Evaluation on Ranking
  - Our ranking model is **RankingAst**



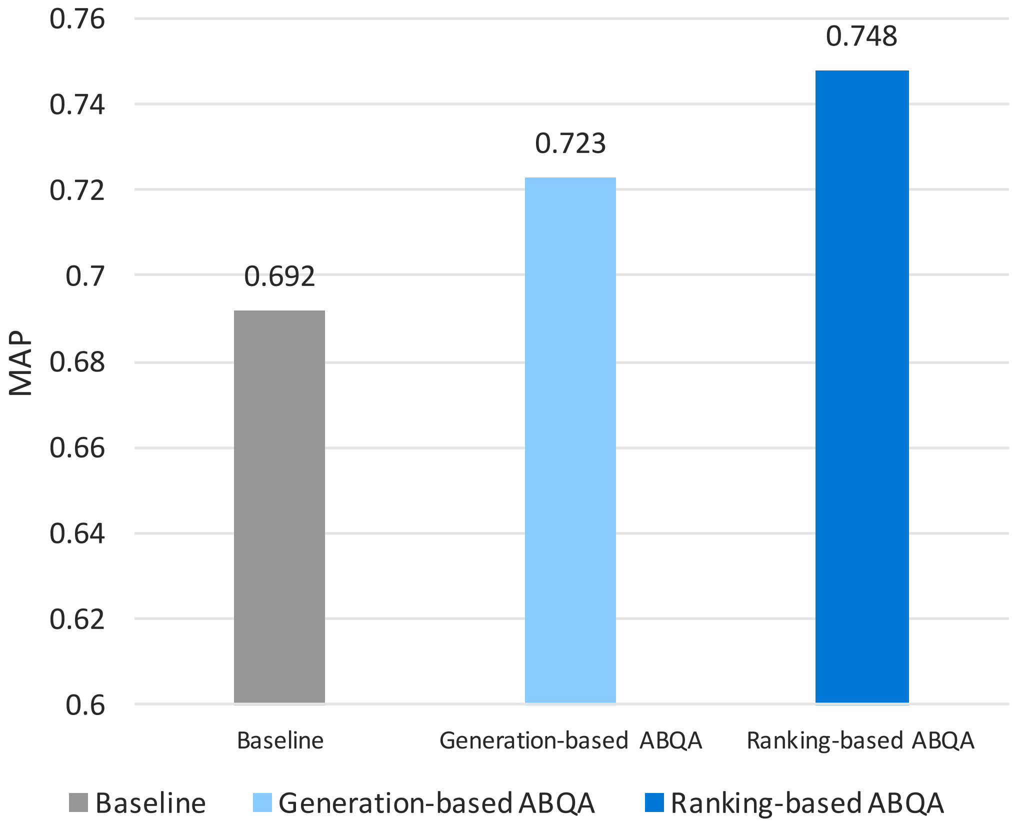
Effects of features at different levels

Evaluation

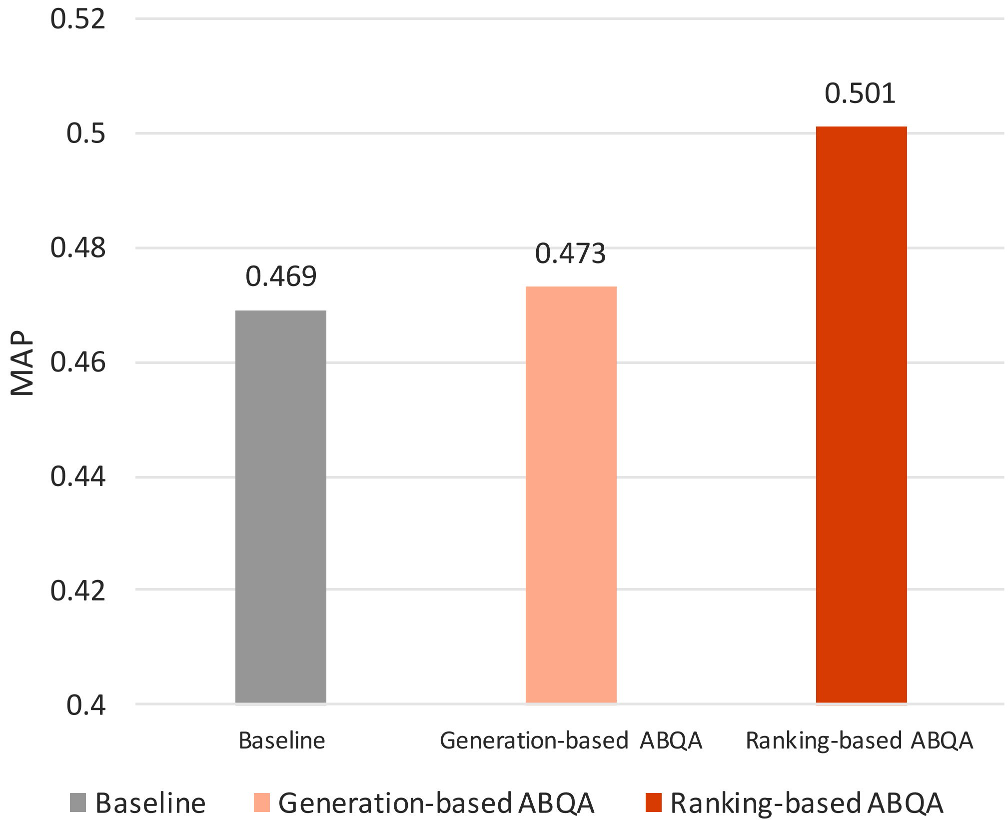
端到端训练：用了基于决策森林的开源算法LambdaMART
ABQA的结果只是作为一个特征加入到PBQA中——用CNN编码成连续向量

- Integrate ABQA into PassageQA task

Wiki这个数据集就是用于句子选择任务的，MARCO这个数据本用于机器阅读，但在这里也可以实现段落排序



WikiQA Dataset



MS MARCO Dataset

这张是完整的比对表，可以看到抽取法效果好一些，并且在MARCO数据集上有所表现。
不过整体来看，这个ABQA特征并不是特别出色

	Methods	WikiQA		MARCO	
		MAP	**MRR**	**MAP**	**MRR**
Published Models					
(1)	CNN+Cnt (Yang, Yih, and Meek 2015)	65.20%	66.52%	-	-
(2)	LSTM+Att+Cnt (Miao, Yu, and Blunsom 2015)	68.55%	70.41%	-	-
(3)	ABCNN (Yin et al. 2016)	69.21%	71.08%	46.91%	47.67%
(4)	Dual-QA (Tang et al. 2017)	68.44%	70.02%	48.36%	49.11%
(5)	IARNN-Occam (Wang, Liu, and Zhao 2016)	73.41%	74.18%	-	-
(6)	conv-RNN (Wang, Jiang, and Yang 2017)	**74.27%**	75.04%	-	-
(7)	CNN+CH (Tymoshenko, Bonadiman, and Moschitti 2016)	73.69%	**75.88%**	-	-
Our Models					
(8)	Baseline	69.89%	71.33%	45.97%	46.62%
(9)	Baseline+RndAst	69.17%	70.12%	46.62%	47.27%
(10)	Baseline+MaxAst	71.82%	72.81%	49.37%	50.05%
(11)	Baseline+ExtAst	72.33%	73.52%	**50.07%**	**50.76%**
(12)	Baseline+Seq2Ast	72.26%	73.35%	47.44%	48.10%

Table 8: Evaluation of answer selection task on WikiQA and MARCO datasets.

Thanks!