文本匹配相关方向打卡点总结

原创 夕小瑶 夕小瑶的卖萌屋 2019-10-18

来自专辑
卖萌屋@自然语言处理

Motivation

前不久小夕在知乎上写了一个回答《NLP有哪些独立研究方向》[1]，于是有不少小伙伴来问分类和匹配的参考资料了，鉴于文本分类的资料已经超级多了，就不写啦（不过分类相关的tricks可以看之前写的这篇文章《文本分类重要tricks总结》）。匹配问题由于场景比较多，相关的文章不多，所以本文就致力于总结一下文本匹配问题上可以打卡的相关资料啦。

文本匹配是一个很宽泛的概念，只要目的是研究两段文本之间的关系，基本都可以把这个问题看作是文本匹配问题。由于在不同的场景下对”匹配“的定义可能非常不同，因此文本匹配并不是一个完整独立的研究方向。不过有相当多的NLP任务可以建模成文本匹配问题，当它们建模成文本匹配问题时，当然会发现模型结构、训练方法等是高度高度相似的，却又有着微妙的不同。所以这个问题虽然跑个baseline简单，但是把具体的匹配问题中做好却并不容易（尤其是在有BERT之前）。

下面就来具体说说可以打卡的内容。

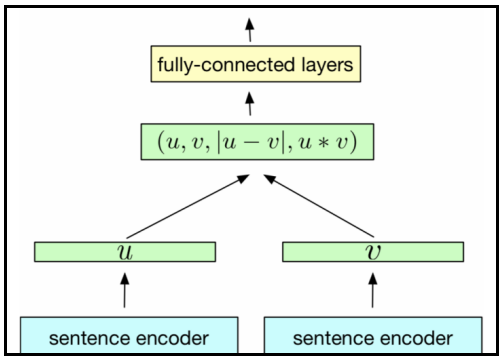
PS：订阅号后台回复**「文本匹配」**可领取小夕打包好的论文大礼包噢～（包括正文中的papers）

本文目录

打卡的baseline模型

无论具体的匹配问题是什么，有一些很好实现的baseline是可以不管三七二十一的直接跑一下的。

我自己最喜欢用的baseline是SiameseCNN这种结构的模型，毕竟从头手撸一遍非常快的，跑的又很快，效果又不错，训练又比较稳定，受超参数的影响相对较小。



模型大体结构如图所示，这里一般没必要实现的太花哨，一般就用一层CNN来分别encoding一下需要匹配的textA和textB，然后max pooling一下或再concat一个mean pooling得到两个文本的向量表示vecA和vecB（上图中的u和v）。

这之后可以直接套用一些公式如cosine距离、L1距离、欧式距离等得到两个文本的相似度，不过我们做文本匹配并不一定是希望判断这两个文本是否相似，除了相似关系外，还可以有问答关系、对话回复关系、文本蕴含关系等，因此更通用的做法是基于u和v构建用于建模两者匹配关系的特征向量，然后**用额外的模型（比如MLP）来学习通用的文本关系函数映射。**

这个特征向量可以像上图一样包括vec1, vec, |vec1-vec2|, vec1*vec2，也可以包括一些更加fancy的features，比如小夕常加的max(vec1, vec2)^2等，在一些匹配场景下有奇效。当然啦，更加靠谱的还是根据实际匹配场景的(bad)case来精心构造features。

如果对LSTM有执念，完全可以用lstm替代cnn来当sentence encoder，也就是使用SiameseLSTM结构，同样这里的encoder可以搭配各种预训练模型强化一下文本的向量表示。

燃鹅，其实有了BERT之后，我就更喜欢拿BERT来当baseline了┐(￣▽￣"")┌，毕竟连代码都不用写了，更方便（经常baseline跑了一下发现问题解决了）。

打卡的任务场景和数据集

一、相似度计算&复述识别（textual similarity&paraphrase identification）

这个可以说是文本匹配最典型最经典的场景了，也就是判断两段文本是不是表达了同样的语义，即是否构成复述（paraphrase）关系。有的数据集是给出相似度等级，等级越高越相似（这种更合理一些），有的是直接给出0/1匹配标签。这一类场景一般建模成分类问题。

代表性数据集：

- *SemEval STS Task：*从2012年开始每年都举办的经典NLP比赛。这个评测将两段文本的相似度程度表示为0.0~5.0，越靠近0.0表示这两段文本越不相关，越靠近5.0表示越相似。使用皮尔逊相关系数（Pearson Correlation）来作为评测指标。链接[2]
- *Quora Question Pairs (QQP)：*这个数据集是Quora发布的。相比STS，这个数据集规模明显大，包含400K个question-question pairs，标签为0/1，代表两个问句的意思是否相同。既然建模成了分类任务，自然可以使用准确率acc和f1这种常用的分类评价指标啦。（知乎什么时候release一个HuQP数据集(￣∇￣)）链接[3]
- *MSRP/MRPC：*这是一个更标准的复述识别数据集。在QQP数据集中文本都是来自用户提问的问题，而MRPC里的句子则是来源于新闻语料。不过MRPC规模则要小得多，只有5800个样本（毕竟是2005年release的数据集，而且人工标注，所以可以理解┐(￣▽￣"")┌）。跟QQP一样，MRPC一般也用acc或f1这种分类指标评估。链接[4]
- *PPDB：*这个paraphrase数据集是通过一种ranking方法来远程监督[]做出来的，所以规模比较大。文本粒度包含lexical level（单词对）、phrase level（短语对）和syntactic level（带句法分析标签）。而且不仅包含英文语料，还有法语、德语、西班牙语等15种语言（为什么没有中文！）。语料库规模从S号、M号一直到XXXL号让用户选择性下载也是很搞笑了，其中短语级就有7000多万，句子级则有2亿多。由于语料规模太大，标注质量还可以，因此甚至可以拿来训练词向量[5]。链接[6]

二、问答匹配（answer selection）

问答匹配问题虽然可以跟复述识别一样强行建模成分类问题，但是实际场景往往是从若干候选中找出正确答案，而且相关的数据集也往往通过一个匹配正例+若干负例的方式构建，因此往往建模成ranking问题。

在学习方法上，不仅可以使用分类的方法来做（在ranking问题中叫**pointwise learning**），还可以使用其他learning-to-rank的学习方法，如**pairwise learning**（"同question的一对正负样本"作为一个训练样本）和**listwise learning**（"同question的全部样本排好序"作为一个训练样本）。因此，相应的评价指标也多使用**MAP、MRR**这种ranking相关的指标。

> 注意：这并不代表pointwise matching这种分类做法就一定表现更弱，详情见相关papers

代表性数据集如：

- *TrecQA：*包含56k的问答对（但是只有1K多的问题，负样本超级多），不过原始的数据集略dirty，包含一些无答案样本和只有正样本以及只有负样本的样本（什么鬼句子），所以做research的话注意一下，有些paper是用的clean版本（滤掉上述三类样本），有的是原始版本，一个数据集强行变成了两个track。链接[7]
- *WikiQA：*这也是个小数据集，是微软从bing搜索query和wiki中构建的。包含10K的问答对（1K多的问题），样本正负比总算正常了些。链接[8]，paper[9]
- *QNLI：*总算有大规模数据集了，这个是从SQuAD数据集改造出来的，把context中包含answer span的句子作为匹配正例，其他作为匹配负例，于是就有了接近600K的问答对（包含接近100K的问题）。链接[10]

三、对话匹配（response selection）

对话匹配可以看作进阶版的问答匹配，主要有两方面升级。

一方面，对话匹配在问答匹配的基础上引入了历史轮对话，在历史轮的限制下，一些本来可以作为回复的候选会因此变得不合理。比如，历史轮提到过你18岁了，那么对于query"你今天在家做什么呢"，你就不能回复"我在家带孙子"了。

> ps：一个价值五毛钱的例子(¬_¬)

另一方面，对于一个query，对话回复空间要远比问题答案空间大得多，对于问答类query，正确答案往往非常有限，甚至只有一个，但是对话类query却往往有一大串合理的回复，甚至有一大堆的万能回复比如"哦"，"好吧"，"哈哈哈"。很多时候的回复跟query在lexical level上基本没有交集，因此对话匹配模型更难训一些，数据质量稍差就难以收敛。因此做够了问答匹配，来做做对话匹配还是比较意思滴。

该问题一般使用**Recall_n@k**（在n个候选中，合理回复出现在前k个位置就算召回成功）作为评价指标，有时也会像问答匹配一样使用**MAP、MRR**等指标。

代表性数据集：

- *UDC：*Ubuntu Dialogue Corpus是对话匹配任务最最经典的数据集，包含1000K的多轮对话（对话session），每个session平均有8轮对话，不仅规模大而且质量很高，所以近些年的对话匹配工作基本都在这上面玩。链接[11]，paper[12]
- *Douban Conversation Corpus：*硬要给UDC挑毛病的话，就是UDC是在ubuntu技术论坛这种限定域上做出来的数据集，所以对话topic是非常专的。所以@吴俣 大佬release了这个开放域对话匹配的数据集，而且由于是中文的，所以case study的过程非常享受。链接[13]，paper[14]

四、自然语言推理/文本蕴含识别（Natural Language Inference/Textual Entailment）

NLI，或者说RTE任务的目的就是判断文本A与文本B是否构成语义上的推理/蕴含关系：即，给定一个描述「前提」的句子A和一个描述「假设」的句子B，若句子A描述的前提下，若句子B为真，那么就说文本A蕴含了B，或者说A可以推理出B；若

B为假，就说文本A与B互相矛盾；若无法根据A得出B是真还是假，则说A与B互相独立。

显然该任务可以看作是一个3-way classification的任务，自然可以使用分类任务的训练方法和相关评价指标。当然也有一些早期的数据集只判断文本蕴含与否，这里就不贴这些数据集了。

代表性数据集：

- *SNLI：Stanford Natural Language Inference*数据集是NLP深度学习时代的标志性数据集之一，2015年的时候发布的，57万样本纯手写和手工标注，可以说业界良心了，成为了当时NLP领域非常稀有的深度学习方法试验场。链接[15]，paper[16]
- *MNLI：Multi-Genre Natural Language Inference*数据集跟SNLI类似，可以看做SNLI的升级版，包含了不同风格的文本（口语和书面语），包含433k的句子对，链接[17]
- *XNLI：*全称是*Cross-lingual Natural Language Inference*。看名字也能猜到这个是个多语言的数据集，XNLI是在MNLI的基础上将一些样本翻译成了另外14种语言（包括中文）。链接[18]

五、信息检索中的匹配

除上述4个场景之外，还有query-title匹配、query-document匹配等信息检索场景下的文本匹配问题。不过，信息检索场景下，一般先通过检索方法召回相关项，再对相关项进行rerank。对这类问题来说，**更重要的是ranking**，而不是非黑即白或单纯的selection。ranking问题就不能仅仅依赖文本这一个维度的feature了，而且相对来说判断两个文本的语义匹配的有多深以及关系有多微妙就没那么重要了。

从纯文本维度上来说，q-a、q-r匹配和NLI相关的方法在理论上当然可以套用在query-title问题上；而query-doc问题则更多的是一个检索问题了，传统的检索模型如TFIDF、BM25等虽然是词项（term）level的文本匹配，但是配合下查询扩展，大部分case下已经可以取得看起来不错的效果了。如果非要考虑语义层次的匹配，也可以使用LSA、LDA等主题模型的传统方法。当然啦，强行上深度学习方法也是没问题的，例如做一下query理解，甚至直接进行query-doc的匹配（只要你舍得砸资源部署），相关工作如

DSSM：CIKM2013 | Learning Deep Structured Semantic Models for Web Search using Clickthrough Data

CDSSM：WWW2014 | Learning Semantic Representations Using Convolutional Neural Networks for Web Search

HCAN：EMNLP2019 | Bridging the Gap between Relevance Matching and Semantic Matching for Short Text Similarity Modeling

六、机器阅读理解问题

同时，还有一些不那么直观的文本匹配任务，例如机器阅读理解（MRC）。这是一个在文本段中找答案片段的问题，换个角度来说就可以建模成带上下文的问答匹配问题（虽然候选有点多┐(￣▽￣"")┌）。代表性数据集如SQuAD系列、MS MARCO、CoQA、NewsQA，分别cover了很多典型的NLP问题：MRC任务建模问题、多文档问题、多轮交互问题、推理问题。因此做匹配的话，相关的代表性工作如BiDAF、DrQA等最好打卡一下的。

BiDAF：ICLR2017 | Bidirectional Attention Flow for Machine Comprehension

DrQA：ACL2017 | Reading Wikipedia to Answer Open-Domain Questions

PS：

上述各个场景的模型其实差不太多，甚至一些方法直接在多个匹配场景上进行实验，近两年的paper也大多claim自己是一个非常general的匹配框架/模型。因此下面介绍打卡paper的时候就不区分场景啦，而是分成基于表示和基于交互来介绍打卡点。

注意：虽然基于表示的文本匹配方法（一般为Siamese网络结构）与基于交互的匹配方法（一般使用花式的attention完成交

互）纷争数年，不过最终文本匹配问题还是被BERT及其后辈们终结了。因此下面两节请带着缅怀历史的心情来打卡，**不必纠结paper的细节，大体知道剧情就好。**

打卡的Siamese结构（基于表示）

这种结构就是本文开头提到的，首先对两段文本分别进行encoding进而得到各自的向量表示，然后通过相似度计算函数或相关结构来得到最终的匹配关系。

在baseline阶段提到的SiameseCNN和SiameseLSTM的基础上，这个方向往下做无非就是两个方向：

1. 加强encoder，得到更好的文本表示
2. 加强相似度计算的函数建模

对于第一个方向，无非就是使用更深更强大的Encoder，代表性打卡工作如

InferSent：EMNLP2017 | Supervised Learning of Universal Sentence Representations from Natural Language Inference Data

> ps：虽然这篇paper的真正目的是迁移学习

SSE：EMNLP2017 | Shortcut-Stacked Sentence Encoders for Multi-Domain Inference

对于第二个方向，则是使用更花哨的相似度计算函数或更花哨的用于学习相似度函数的网络结构，可打卡的工作如

SiamCNN：ASRU2015 | Applying deep learning to answer selection: A study and an open task

SiamLSTM：AAAI2016 | Siamese Recurrent Architectures for Learning Sentence Similarity

Multi-view：2016 EMNLP | Multi-view Response Selection for Human-Computer Conversation

显而易见，这个方向可玩性不强（虽然容易work但是paper写出来不够炫酷），所以不要问为什么只更新到了2017年，因为2016年attention就遍地开花了，自然大家基本都跑去赶潮做花式交互结构了。

打卡的花式attention结构（基于交互）

顾名思义，这种思路就是首先通过attention为代表的结构来对两段文本进行不同粒度的交互（词级、短语级等），然后将各个粒度的匹配结果通过一种结构来聚合起来，作为一个超级特征向量进而得到最终的匹配关系。

显然这种思路下，除了让文本对的交互更花哨以外，就是考虑让模型变得更深（从而建模更高level的匹配关系）。

不过个人经验来说，这种思路下虽然可以玩的花样很多，一些论文argue的点也看似有一些道理，不过实际很多模型都是在廖廖一两个数据集上疯(暴)狂(力)改(搜)进(索)各种structure才把分数刷上去的，导致这种structure看似在某个场景甚至仅仅是某些数据集上work，实际上这个structure可能仅仅迎合了特定数据分布或特定场景的一些特性，导致很多工作放到一个新场景下就效果翻车了，甚至努力调参都调不动太多。

因此在BERT之前这类论文提出的模型虽然看起来高大上，不过可能换个数据集后还不如稍微调调参拍拍脑袋的SiameseCNN好用。所以在刷这类论文时，千万不要被蜜汁花哨的模型结构迷惑了双眼噢~相关工作很多，从中挑选了几篇比较有代表性或比较有信息量或容易阅读的。

MatchCNN：AAAI2016 | Text Matching as Image Recognition

DecAtt：EMNLP2016 | A Decomposable Attention Model for Natural Language Inference

CompAgg：ICLR2017 | A COMPARE-AGGREGATE MODEL FOR MATCHING TEXT SEQUENCES

ESIM：ACL2017 | Enhanced LSTM for Natural Language Inference

2018 COLING | Neural Network Models for Paraphrase Identification, Semantic Textual Similarity, Natural Language Inference, and Question Answering

> ps：这篇paper其实可以看做是对前面各模型的实验和分析大总结

DAM：ACL2018 | Multi-Turn Response Selection for Chatbots with Deep Attention Matching Network

HCAN：EMNLP2019 | Bridging the Gap between Relevance Matching and Semantic Matching for Short Text Similarity Modeling

此外，这里尤其要注意一下模型对称性的问题，像文本相似度计算/q-q匹配/title-title匹配这类场景下的匹配是对称的，即match(a,b)=match(b,a)，但是模型不对称后，就会让模型自己额外的学习这个先验知识，除非数据集很大，或者已经预训练过了，否则效果很容易翻车。当然了，也有一些tricks可以强行使用不对称模型，即在这类场景下对每个样本都跑一遍match(a,b)和match(b,a)然后取平均，不过相比天然对称的模型效果如何就要看各位炼丹师的水平啦

打卡的学习方法

pointwise/pairwise/listwise learning这三种方法已经资料满天飞了，这里就不赘述了。这里给还不熟悉的小伙伴们推荐一篇文章[19]

打卡的pretrain models

虽然经过若干年的炼丹，靠model structure已经可以在非常多的文本匹配任务场景取得不错的效果了，但是实验证明，还是没法跟海量语料上pretrain的模型比的，先上一张图，问答数据集TrecQA上的实验结果：

Model	TrecQA		TwitterURL	Quora
	MAP	MRR	macro-F1	Acc
InferSent	0.521	0.559	0.797	0.866
DecAtt	0.660	0.712	0.785	0.845
ESIM_{seq}	0.771	0.795	**0.822**	0.850
ESIM_{tree}	0.698	0.734	-	0.755
$\text{ESIM}_{seq+tree}$	0.749	0.768	-	0.854
PWIM	0.739	0.795	0.809	0.834
State-of-the-Art Models				
Rao et al. (2016)	**0.780**	0.834	-	-
Gong et al. (2018)	-	-	-	**0.891**
BERT	0.838	0.887	0.852	0.892
Our Approach				
RM	0.756	0.812	0.790	0.842
SM	0.663	0.725	0.708	0.817
HCAN	0.774	**0.843**	0.817	0.853

其中HCAN是EMNLP2019新提出的模型，虽然已经吊打了ESIM、DecAtt等老一代花哨模型，但是可以看到还是被BERT吊打了，更不必说跟XLNet、ERNIE2.0和RoBERTa等近期模型去对比了。所以**真正大一统文本匹配任务的话，目前来看还是离不开大型预训练模型的。**

当然啦，非要用传统的匹配模型的话，至少还有ELMo可以拿来强行续命【手动狗头】

打卡的开源工具

虽然文本匹配baseline容易构造，不过要在具体场景搭建一个完整的系统还是工作量比较大的，借助一些好用的开源工具可以大大提升开发效率。

MatchZoo[20]：一个通用文本匹配工具包，囊括了非常多代表性的数据集、匹配模型和场景，接口友好，非常适合拿来跑baseline。

AnyQ[21]：一个面向FAQ集和的问答系统框架，插件和配置机制做的很赞，集成了一堆代表性的匹配模型和一些检索模型，完整涵盖了Question Analysis、Retrieval、Matching和Re-Rank这4个做问答系统的全部必备环节。

DGU[22]：一个bert-based通用对话理解工具，提供了一套simple but effective的对话任务解决方案，一键刷爆各个对话任务（包括多轮对话匹配）的SOTA也是一个神奇的体验了。

PS：订阅号后台回复**「文本匹配」**可领取小夕打包好的论文大礼包噢～（包括正文中的papers）

参考文献（正文中贴了的就不在下面写啦）

[1] https://www.zhihu.com/question/335289475/answer/811315108

[2] http://ixa2.si.ehu.es/stswiki/index.php/STSbenchmark

[3] https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs

[4] https://www.microsoft.com/en-us/download/details.aspx?id=52398

[5] 2015TACL | From Paraphrase Database to Compositional Paraphrase Model and Back

[6] http://paraphrase.org/#/download

[7] https://trec.nist.gov/data/qa.html

[8] https://www.microsoft.com/en-us/download/details.aspx?id=52419

[9] Yang Y, Yih W, Meek C. Wikiqa: A challenge dataset for open-domain question answering[C]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015: 2013-2018

[10] https://firebasestorage.googleapis.com/v0/b/mtl-sentence-representations.appspot.com/o/data%2FQNLIv2.zip?alt=media&token=6fdcf570-0fc5-4631-8456-9505272d1601

[11] http://dataset.cs.mcgill.ca/ubuntu-corpus-1.0/

[12] Lowe R, Pow N, Serban I, et al. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems[J]. arXiv preprint arXiv:1506.08909, 2015.

[13] https://archive.org/details/DoubanConversaionCorpus

[14] Wu Y, Wu W, Xing C, et al. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots[J]. arXiv preprint arXiv:1612.01627, 2016.

[15] https://nlp.stanford.edu/projects/snli/

[16] Bowman S R, Angeli G, Potts C, et al. A large annotated corpus for learning natural language inference[J]. arXiv preprint arXiv:1508.05326, 2015

[17] http://www.nyu.edu/projects/bowman/multinli

[18] https://www.nyu.edu/projects/bowman/xnli

[19] https://zhuanlan.zhihu.com/p/26539920

[20] https://github.com/NTMC-Community/MatchZoo.

[21] https://github.com/baidu/AnyQ

[22] https://github.com/PaddlePaddle/models
