基于深度学习的自动文本摘要

叶茂青，王珺，游小艳，徐嘉鸿，刘宇轩，李凌峰

Abstract

自动文本摘要指利用计算机程序自动地总结出文档的主要内容，自动文本摘要主要可分为抽取式摘要，即直接从原文中抽取一些句子组成摘要，和生成式摘要，即尝试理解原文的意思并生成出摘要，本文尝试使用不同的句向量生成方式和方法进行抽取式摘要，并对不同方法的效果和优缺点进行分析。

1. 句向量的生成

1.1. 词向量

Word2vec[6]，为一群用来产生词向量的相关模型。这些模型为浅层双层的神经网络，用来训练以重新建构语言学之词文本。网络以词表现，并且需猜测相邻位置的输入词，在 word2vec 中词袋模型假设下，词的顺序是不重要的。

训练完成之后，word2vec 模型可用来映射每个词到一个向量，可用来表示词对词之间的关系。该向量为神经网络之隐藏层。

Word2vec 依赖 skip-grams 或连续词袋（CBOW）来建立神经词嵌入。

1.2. 句向量

一种简单的句向量生成就是通过拼接或者平均词向量的方式实现 [4]，但这种方式生成的句向量对于各个词的权重都是一致的，不符合语言的特性，相应的人们也提出了加权平均的方式，比如通过 TF-IDF 评估一个词语对于一个文档的重要程度，一个词语 TF-IDF 值与它在文档中出现频数成正比，与它在语料库中出现的频率成反比。TF-IDF 由 TF 词频 (Term Frequency) 和 IDF 逆向文件频率 (Inverse Document Frequency) 相乘而得。但这种方式本质上无法理解上下文的语义，同一个词在不同的语境意思可能不一样，但是却会被表示成同样的句向量。也有人尝试用 word2vec 的方法进行扩展，使用与 skip-grams 类似的模型进行训练，如 Skip-thought vectors[3]，但是这种方法过于直接，句子包含多种组合，通过一句话推知上下文不符合常识。因此目前效果更好的句向量生成方法通常采用有监督学习的方法，如 InferSent[1]，它使用 Sentence Natural Language Inference（NLI）数据集（包含 570k 对标有 3 种类别的句子：中性，矛盾和包含）来在句子编码器之上训练分类器。两个句子都使用相同的编码器进行编码，而分类器则是根据两个句嵌入构建的对表征进行训练。通过这种方法获取的句向量可以更有效的泛化到下游任务。而在 2018 年，语言模型开始向多任务训练转变，如 BERT[2], 这可以让模型更充分的了解上下文的含义，生成更好的嵌入。

1.3. 选取的方案

我们尝试采用了简单的平均和用 TFIDF 加权平均 word2vec 的方式获取的词向量，同时采取有监督学习获取句向量中比较经典的 InferSent 模型进行句向量的获取。

2. 摘要生成

2.1. 生成式摘要

生成式摘要，即尝试理解原文的意思并生成出摘要，这种方式生成的摘要可以生成出原文不存在的句子，更贴近人类进行摘要的过程，也是目前热门的研究方向。利用深度学习解决生成式摘要最早的尝试在 2015 年，Rush 等人 [9]尝试用 seq2seq 模型生成文本摘要，2016 年，Rush 等人 [7]又在原来的模型上做出改进，使用 RNN 网络代替了原来的 FNN，并引入了 Large Vocabulary Trick，Vocabulary expansion，Feature-rich Encoder，Switching Generator/Pointer 等机制，提升

了模型的表现。但此时的模型生成的摘要带有大量的UNK，可读性差，See 等人 [10]之后提出了将 pointer-network[11]引入进生成式摘要中，模型可以从原文中复制词汇，极大的改善了摘要的可读性。我们尝试使用pointer-generator networks[10]进行生成式摘要的训练，并复现了论文的结果，最后生成的部分摘要虽然可以与人工摘要不分上下，但大部分摘要都存在着语法问题，或是事实错误的问题。查找了更多文献后，我们并没有找到解决方案，且目前也没有良好的损失函数可以评判生成式摘要在语法，摘要准确度上的损失。考虑到训练时长（单 Titan X 需要 3 天的时间才能让网络收敛）和修改代码的难度，我们放弃了这一方案。

2.2. 抽取式摘要

抽取式摘要从原文中抽取出一定句子作为摘要，由于是从原文中摘取的句子，这种方式生成的摘要不存在语法错误或事实错误的问题，但抽取的句子拼接在一起可能不通顺，将句子按原文的顺序排布一定程度上可以让摘要更加通顺，但是并没有从根本上解决这个问题。抽取式摘要的方式有很多，聚类，MMR, 基于图排序的 LexRank，Textrank，使用整数线性规划的 ILP 算法等，我们采用了聚类，MMR，Textrank 这三种方法进行了尝试。

2.3. 聚类

聚类算法是一种无监督学习的算法，给定一组数据点，可以用聚类算法将每个数据点分到特定的组中，假设我们从文本中抽取出三句摘要，通过句嵌入方法得到句子的嵌入后，我们可以使用聚类算法取得三个聚类中心，将其看作文章的三个信息点，计算每句话的嵌入与聚类中心的距离，对三个嵌入中心各选取一个距离最近的句子就可以得到摘取出的三句。

2.4. Textrank

Textrank 算法基于著名的网页排序算法 PageRank 改动而来，PageRank 算法的基本思想在于，一个页面如果被多个页面引用，说明该页面的重要性高，如果一个页面被重要性高的页面引用，相应的该页面的重要性也会提升。Textrank 算法首先把所有文章整合成文本数据，接着把文本分割成单个句子，将句子转换成词向量从而构建节点连接图，用句子之间的余弦相似度作为边的权重并用矩阵存储起来，再将相似矩阵转换为以句子为节点、余弦相似度为边的图结构，计算句子的 TextRank 值，进而选出重要性高的 N 句作为摘要。

2.5. MMR

Maximal Marginal Relevance(MMR), 不同于聚类方法和 Textrank 方法选取全文中最重要的句子，MMR 算法可以均衡的考虑文章摘要的重要性和多样性，MMR 算法的计算公式如下

$$MMR\ Score(i) = \lambda * score(i) - (1-\lambda) * max_j S(i,j) \tag{1}$$

$$score(i) = S(i, doc) \tag{2}$$

λ 的值越大，代表得分更注重摘要的准确度，λ 的值越小，代表得分更注重摘要的多样性。S 函数计算文本与句子或句子与句子之间的相似度，相似度计算可以使用欧氏距离或者余弦相似度进行计算。

3. 结果对比及分析

3.1. 评测指标

目前自动文本摘要的评测方法主要使用ROUGE[5]，ROUGE 方法的基本思想是将模型生成的摘要与参考摘要两者的 n-gram 进行对比，从而计算召回率，准确率等数据，ROUGE 的评价指标在自动文摘上主要使用 ROUGE-N 和 ROUGE-L 两个，ROUGE-N 计算生成的摘要与标注摘要的 n-gram 召回率，通常用 ROUGE-1/2 来评估，ROUGE-L 计算两个文本单元之间的最长公共序列。ROUGE 评测方法作为一个 2003 年提出的方法，目前看来有很大的缺点，使用 n-gram 模型进行统计十分呆板，没有考虑到生成摘要在语法，事实等方面上的准确性，但目前来说还没有一个更好的评价方法可以取代 ROUGE 方法。以下的结果均使用 python 的 rouge 库进行计算，相比起论文中普遍使用的通过 perl 写的 rouge 评测工具，由于在语句处理上有差异，两者在得分上会有不同。

3.2. 模型效果对比

ORACLE 指从文章取出 3 句，在可能的组合中，ROUGE 得分最大的组合，LEAD3 选用文章的前三句，是常用且十分有效的 Baseline，RANDOM 采用随机选

Models	$ROUGE-1$	$ROUGE-2$	$ROUGE-L$
ORACLE	56.61	29.57	38.48
LEAD3	40.62	17.05	24.86
RANDOM	29.15	8.78	17.62
平均词向量 + 聚类	31.47	10.25	18.89
TFIDF 加权平均词向量 + 聚类	32.28	10.85	19.48
InferSent（Glove）+ 聚类	33.05	11.42	19.33
InferSent（FastText）+ 聚类	32.51	11.03	19.23
平均词向量 +Textrank	31.39	10.12	18.18
InferSent+MMR	40.63	17.06	24.87

取 3 句的方式。InferSent 模型有使用 Glove 进行嵌入的模型，也有使用 FastText 进行嵌入的模型，经过测试，使用 Glove 嵌入的模型在效果上更优，所以之后使用的 InferSent 都是使用 Glove 进行训练的模型。

3.3. 结果分析

经过我们的测试，TFIDF 加权平均词向量一定程度上可以改善模型的表现，换用更好的句向量嵌入模型也可以让模型效果更好，而使用聚类方法和基于图的 Textrank 算法在表现上并没有太大差别。聚类中使用欧氏距离和余弦距离（代码中我们让向量归一化然后计算欧氏距离，两者在数学上是等价的）对模型的效果影响不大。聚类算法和 Textrank 算法表现不佳的原因一定程度上是因为这类算法只注重摘要的准确度而忽略了多样性。而 MMR 算法可以通过 λ 的值控制摘要在准确度和多样性之间的取舍，以下是我们使用不同的 λ 值做出的结果。

$lambda$	$ROUGE-1$	$ROUGE-2$	$ROUGE-L$
0.2	22.05	5.50	13.58
0.5	40.63	17.06	24.87
0.8	31.44	10.72	16.99

在 λ 值偏大时，MMR 算法更注重摘要的精确度，最后的 ROUGE 得分也与只注重摘要精确度的聚类算法和 Textrank 算法类似，而当 λ 值偏小时，模型偏向与摘要的多样性，模型的效果有明显的下降，当 λ 值选择恰当时，MMR 算法的表现会明显优于聚类算法和 Textrank 算法。

此外，在对聚类结果进行可视化的过程中，我们发现了聚类算法可能存在的问题，那就是文本中的极端数据可能会影响聚类的结果，下图是我们使用 PCA 降维后对聚类结果可视化的结果。可见由于文本中部分句子的嵌入有着严重的偏离，使用 K-means 聚类所得的聚类中心会向着这些样本偏移，从而造成选取的句子不属于文章的主要信息。虽然降维后的显示不能实际表示嵌入空间中各向量之间的位置，距离关系，但我们认为在高维空间上，这种问题仍然可能发生。我们考虑过使用其他的聚类算法进行尝试，但其他的聚类方法都难以直接套用在自动文摘任务上，如 DBSCAN 只会对样本分簇而不会给出聚类中心，无法评判句子的得分，Affinity propagation 算法无法设定聚类数目，不能进行对比。

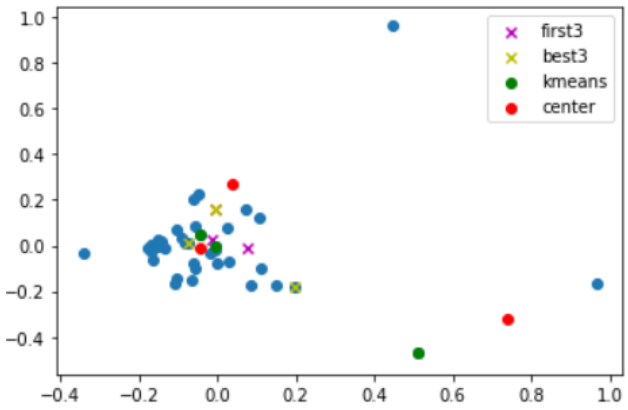

图 1. 聚类可视化

4. 总结

随着深度学习的发展，自动文本摘要的研究方向从抽取式摘要向生成式摘要转变，受限于时间和显卡性能，我们并没有使用生成式摘要的模型，而是尽可能

的选用无监督学习的方式进行抽取式的摘要。从最后模型的结果不难看出，自动文本摘要要求对文章信息的精确把握，而这对嵌入模型和抽取模型都有着一定的要求，近年自动文本摘要的 STOA 模型也普遍使用了 bert 进行嵌入，利用深度学习建立良好的语言模型并不是易事，这是自动文本摘要受限的原因之一。

自动文本摘要发展缓慢的另外一个原因，就是评测指标的问题，ROUGE 提出这么久之后，依然没有一个更好的评价模型可以用于自动文本摘要领域，但今年也可以看到有研究人员开始重新思考自动文本摘要评价的指标，如 Peyrard 和 Maxime 为文本摘要提出了四个评价角度 [8]。如果之后的研究可以设计出适合于自动文本摘要的目标函数，相信自动文本摘要领域会有更好的发展。

参考文献

[1] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. arXiv preprint arXiv:1705.02364, 2017. 1

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018. 1

[3] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In Advances in neural information processing systems, pages 3294–3302, 2015. 1

[4] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In International conference on machine learning, pages 1188–1196, 2014. 1

[5] Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, pages 150–157, 2003. 2

[6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013. 1

[7] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv preprint arXiv:1602.06023, 2016. 1

[8] Maxime Peyrard. A simple theoretical model of importance for summarization. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1059–1073, 2019. 4

[9] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685, 2015. 1

[10] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368, 2017. 2

[11] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In Advances in Neural Information Processing Systems, pages 2692–2700, 2015. 2