结合N-gram与doc2vec和多任务卷积神经网络的文本分类方法

大数据精准营销中搜狗用户画像挖掘

杨毅[1]，邓齐林[2]，徐俊[3]
[1,2]南京大学计算机科学与技术系，[3]国防科技大学计算机学院
Email: 549596590@qq.com, qldeng@qq.com, 309925258@qq.com

2016年12月20号

Abstract

尽管缺少语义信息，基于N-gram与SVM的文本分类方法至今仍然是分类准确率很高的模型。对于短文本，N-gram配合Naive Bayes就具有很高的性能。对于长文本，最近提出的基于神经网络模型的doc2vec和卷积神经网络(cnn)具有很好的性能。而搜索引擎的用户搜狗记录是由若干短文本构成的长文本，我们提出结合N-gram与doc2vec和多任务卷积神经网络的文本分类模型。实验结果表明，提出的模型在此任务上有很好的表现。

1 简介

文本分类是指按照预先定义的主题类别，为文档集合中的每个文档确定一个类别。文本分类在信息检索、Web文档自动分类、数字图书馆、自动文摘、分类新闻组、文本过滤、单词语义辨析等领域取得了广泛的应用。由于其巨大的商业价值，文本分类得到了广泛的研究。众多的统计方法和机器学习方法被应用于文本分类，传统的以TF-IDF为代表的词频特征配合Naive Bayes, Logistic Regression, 和SVM等分类器在一段很长的时间里占据着主要地位，主要由于其拥有模型简单，而分类准确率很高的优点。另外存在一些基于潜在语义分析的模型，主要是LSI和LDA，也得到了广泛的研究，但其分类准确率一直很难超过TF-IDF+SVM。Wang等人提出

结合Naive Bayes特征与SVM分类器，这一简单的模型却在很多数据集上取得了当时最好的性能[8]。

尽管词频特征拥有了不错的性能，但是其忽略了上下文信息，没有考虑词的先后顺序和相似性，另一方面，词频特征无法考虑句子的语义信息。为了解决这些问题，词向量(word2vec)[6]和句子向量(doc2vec)[3]被提出。顾名思义，词向量是指为每个词语分配一个向量，词向量能够考虑词语之间的语义信息，使得相关词在向量空间内相似性更高。同理，句向量是为每个句子分配一个向量，使得相似的句子在向量空间具有更高的相似性。词向量最早于2000年由Bengio教授等人提出[1]，经过一系列的发展，如今已经有很多方法用于生成词，主要的一类是基于神经网络模型。

词向量为每个词学到一个向量，一个句子由若干具有先后顺序的词构成，那么，使用词向量特征，一个句子可以表示为一个矩阵。如果直接拿这个矩阵作为特征，势必会造成输入维度过高的问题(假设词向量长度为100，句子长度为100，那么句子的特征将会是10000维)。于是，借鉴图片识别的卷积神经网络模型，Kim等人提出了用于文本分类的1维卷积神经网络模型[2]，该模型能够考虑词之间的语义信息，同时，通过卷积操作，可以考虑词语的上下文信息。然而，当上下文关系较弱，甚至存在切断问题时(主要存在于上下两个句子的无关时，比如用户两次搜索记录之间，第一个搜索记录与第二个搜索记录之间不存在上下文关系), Kim等人提出的卷积神经网络将不再适用。

句向量模型是在词向量基础之上发展而来的，具有两种学习算法PV-DM和PV-DBOW，其中PV-DM是利用句向量和上下文单词预测中心单词，而PV-DBOW模型只使用句向量预测句子中的单词。在许多数据集上，PV-DBOW的分类模型好于PV-DM模型[4, 7]，我们认为这主要是因为，PV-DM在训练句向量的过程中同时加入了词向量信息，拥有较强的上文本依赖关系假设，而实际应用中，词语的上下文关系可能较弱。然而，PV-DBOW模型完全不考虑词语的上下关系，也存在改进的空间。

针对卷积神经网络模型和句向量的PV-DBOW模型存在的问题，我们提出使用N-gram特征表征短的上下文关系，而忽略长的上下文关系。然后，在N-gram特征的基础之上，使用卷积核大小为1的1维卷积神经网络。对于doc2vec，在N-gram特征的基础上，使用PV-DBOW模型训练句向量。这样一来，即解决了卷积神经网络的切断问题，又解决了PV-DBOW模型完全没有考虑上下文关系的问题。另外，考虑到不同任务之间的相关性(比如年龄和学历的相关性)，我们使用了多任务的卷积神经网络。实验结果表明，我们的模型具有非常好的性能。最终在大数据精准营销中搜狗用户画像挖掘比赛中取得了复赛第一名的成绩（fox团队）[9]。

2 背景

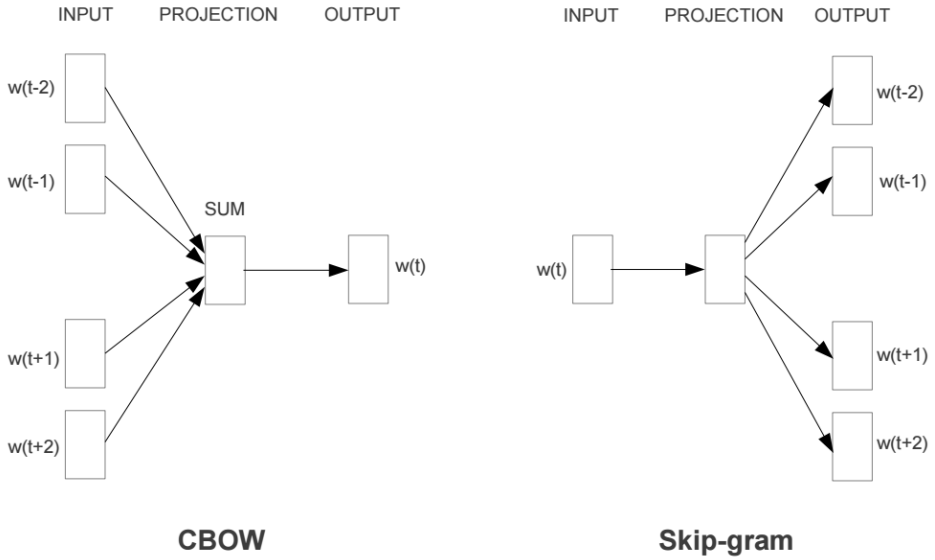

Figure 1: 词向量的两种模型。左图为CBOW模型，右图为Skip-gram模型。

2.1 词向量

这里我们主要介绍Mikolov等人提出的Word2vec模型[6]。词向量是语言模型的产物，所谓语言模型，是给定一个文本集合，最大化文本中的文本出现的似然概率。最大化文本出现的似然概率，可以在某种程度上等价于给定句子的上下文，最大化中心词出现的概率，这也是word2vec两种算法中CBOW模型的基本思想。另一种算法Skip-gram模型与CBOW模型思想刚好相反，使用窗口内的中心词，预测窗口内的其他词。

图1是两种模型的结构示意图，无论是哪种算法，都是由一层映射层和一层softmax层构成。CBOW模型是将窗口内的其他词对应的向量求和，作为分类器的输入，预测中心词。Skip-gram模型是使用中心词的向量作为分类器的输入，预测窗口内的其他词。两种模型都采用BP算法训练，训练过程中同时调整网络权重和词向量矩阵。由于一般词语数量很大，会导致输出节点太多，可以使用层次softmax和负采样两种方式进行优化。层次softmax是使用哈弗曼编码替代one-hot形式的编码，使得输出节点个数降为$log(N)$（N是文本集合中不同词语的个数）；负采样是每次预测中心词时，随机采样k个负样本，一起构成训练块，这样输出节点个数就只有（k+1）个，k一般取5-20，负采样可以看做对输出概率期望的近似，有理论可以证明负采样的期望近似于全采样。

Lai等人在不同的任务、数据集上对不同方法训练的词向量性能做了大量的比较[5]，他们的实验结果表明，在文本集合不太大的情况下，使

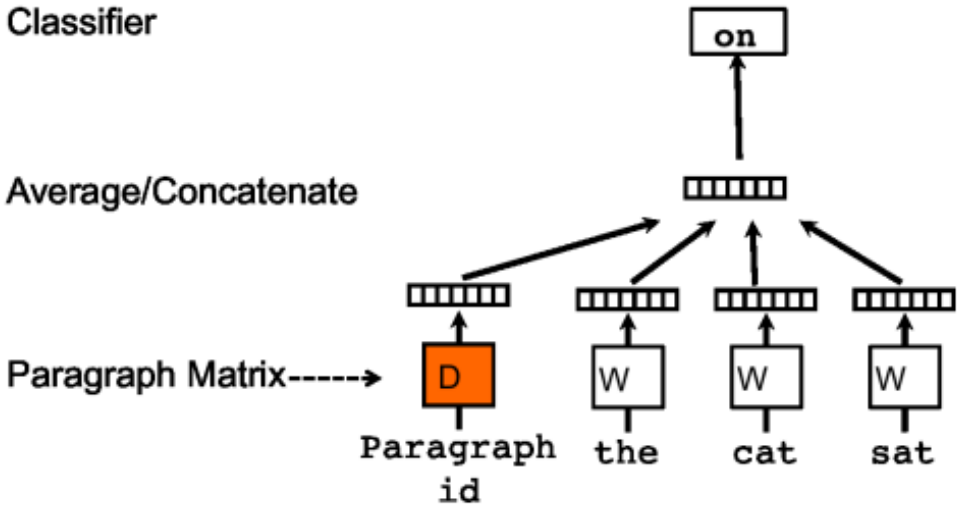

Figure 2: 句向量分布式存储模型(PV-DM)，使用上下文单词对应的向量和句向量的平均或拼接作为分类器的输入，预测中心词。

用Skip-gram模型训练的词向量，在文本分类任务上具有不错的性能。

2.2 句向量

句向量[3]是在词向量的基础上发展而来的，句子可以看做是一个分布在句子每个位置中的词语，在这种假设下，句向量与词向量的CBOW模型对应，在预测中心词时，除了上下文单词，还加入了句向量，这种模型叫做句向量分布式存储模型(PV-DM)。图2是PV-DM模型的结构示意图。包含一个映射层、一个平均池化层或者拼接方式的flatten层、一个作为分类器的softmax层。PV-DM可以看作是某种类型的N-gram模型，因为学到的句向量中包含了每个词的上下文关系。

与PV-DM模型依赖于单词较强的上下文关系不同，句向量的另一种模型PV-DBOW模型在预测中心词时，不依赖上下文单词，只使用了句向量作为分类器的输入。图3是这种模型的示意图。PV-DBOW模型可以看做是分布式的词袋模型，因为训练的是句子与句中每个词单独的对应关系，句向量包含了句中每个词的信息。

由于句向量是继承于词向量，词向量的哈弗曼编码和负采样，句向量中也同时可以使用。

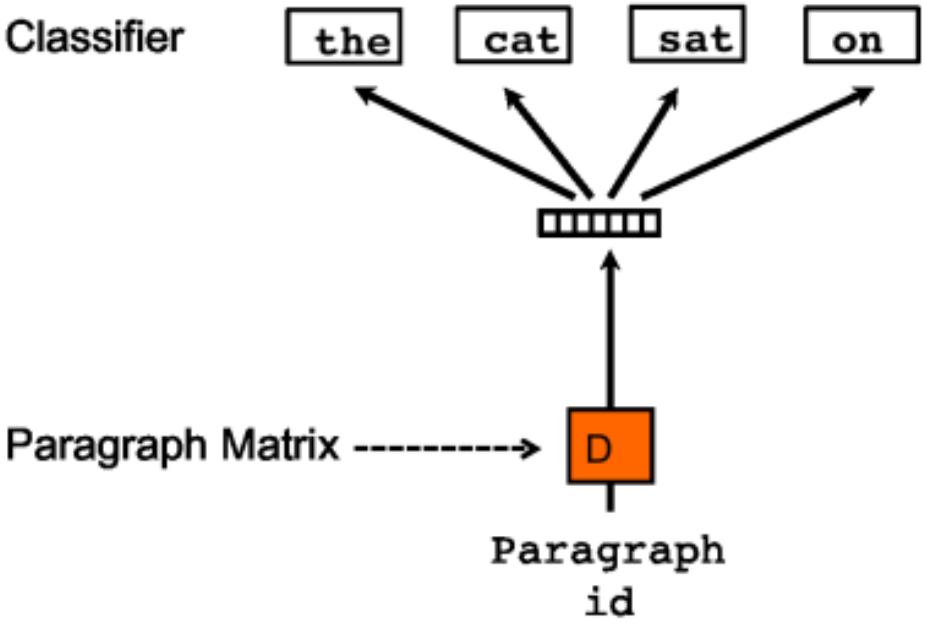

Figure 3: 句向量分布式词袋模型(PV-DBOW)，使用句向量为分类器的输入，预测中心词。

2.3 卷积神经网络

与句向量只使用一个向量表示句子不同，Kim等人提出的卷积神经网络[2]使用句子中的词语对应的词向量拼接成矩阵作为句子的特征。由于每个词的词向量是一个整体，无法采取用于图片的卷积神经网络所采取的2维的卷积操作，用于文本分类的卷积神经网络只在时间尺度上进行卷积，而不切分词向量。

图4是Kim等人提出的卷积神经网络的示意图。他们使用多个尺度不同的卷积核进行卷积操作后，使用最大池化(Max pooling)操作，然后连接全连接层和用于分类的softmax层。这样的结构，卷积操作类似于N-gram，不同的是，使用了词向量之后可以考虑词语之间的相似性（比如近义词）。

3 模型

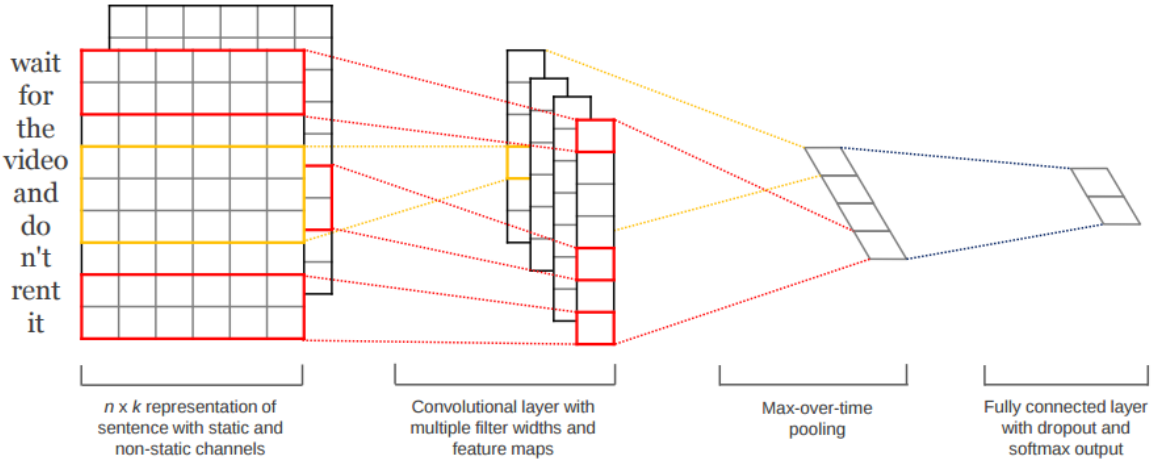

Figure 4: 用于文本分类的卷积神经网络模型，只在时间尺度上进行卷积操作。

3.1 概要

搜狗赛题的特点是用户的搜索记录由若干没有顺序的短文本构成。如果直接采取分类能力较强的PV-DBOW模型训练句向量，完全忽略了短文本中的上下文关系，将会带来了一定程度上的信息损失。而卷积神经网络用于本赛题时，尺度大于1的卷积核存在切断问题，即两个搜索记录之间不存在上下文关系，这样势必会影响整个网络的性能。我们希望，能够在假设单词之间相互独立的同时，考虑短文中的上下文关系。我们使用了N-gram这个即传统又简单的模型表征短文中的上下文关系，在N-gram的基础之上，我们使用PV-DBOW模型训练句向量(这一思想Li等人也提出过[4])，或者使用卷积核大小为1的卷积神经网络。

3.2 N-gram特征提取

大量的研究表明，简单的one-hot特征(0-1特征）虽然损失了大量的信息，但是分类能力却非常好[8]。我们认为这种现象主要是由于one-hot特征相比于词频特征，在训练分类器时，不容易过拟合，使得训练过程可以持续更久，泛化能力更强。于是，对于N-gram特征，我们使用的是one-hot形式。针对搜狗赛题，我们将每条搜索记录分词后提取uni-gram和bi-gram特征，然后，将单个用户的所有搜索记录合并，取one-hot形式的特征。下面是一个例子:

- 搜索记录集合： 微微一笑很倾城 南京明天天气 上海天气
- 特征词: 微微一笑，很，倾城， 微微一笑_很， 很_倾城， 南京，明

天，天气， 南京_明天，明天_天气 上海 上海_天气

注意这里天气搜索了两次，但是取one-hot形式之后，只算1次。

3.3 句向量模型

在提取完one-hot形式的N-gram特征之后，我们在此基础上使用PV-DBOW模型为每个用户训练一个句向量。由于DBOW模型在训练时，句中单词没有上下信息，因此，训练句向量时，每隔几次迭代我们都会打乱文本顺序和每个文本中的单词顺序。

在训练好句向量模型之后，我们使用了神经网络模型作为分类器，句向量作为输入，用户的年龄、学历或性别作为输出。另外，我们实验还发现使用不同尺寸的句向量进行集成可以提高最终的分类准确率。

3.4 多任务卷积神经网络

由于我们采用了N-gram表征文本的上下文信息，在训练词向量时，似乎没有上下文信息可用。然而，我们认为同一用户的所有特征词(N-gram特征)之间是相互相关的，而不考虑特征词的上下关系和远近关系。要实现这种模型，理论上只需要使用窗口无限大的Skip-gram模型训练词向量即可。实际由于训练时间问题，我们使用了窗口大小为30的Skip-gram模型为N-gram特征词训练词向量。

在获取到N-gram的词向量之后，一个用户的所有特征词可以构成一个矩阵，由于没有了上下文信息，因此，我们只需要使用尺寸为1的卷积核进行卷积操作，在进行完卷积操作之后，我们使用了平均池化(Average Pooling)取代Kim等人用的最大池化，之后，使用了一个全连接层，作为年龄、学历和性别三种不同任务的同享特征层。再往后，就是每个任务单独的全连接层和softmax输出层。为了防止过拟合，我们在网络的不同位置加入了Dropout层。具体的网络架构可以见图5。

在网络底层使用共享层，高层根据任务分开训练，这种多任务方式的神经网络模型，能够使用不同种类监督信息来引导底层特征的学习。尤其是本赛题中，学历和年龄的强相关性使得同时训练能够提高两者的分类准率。另外，由于不同于情感分析中起情感作用的只是少数词，本赛题中能够起到分类作用的词较多，因而使用平均池化可以获得更好的性能。

3.5 集成

集成不同的模型往往可以更好的分类准确率[7]，因此，我们集成了上述的句向量与多任务卷积神经网络两种模型。我们采用的集成策略很简单，每

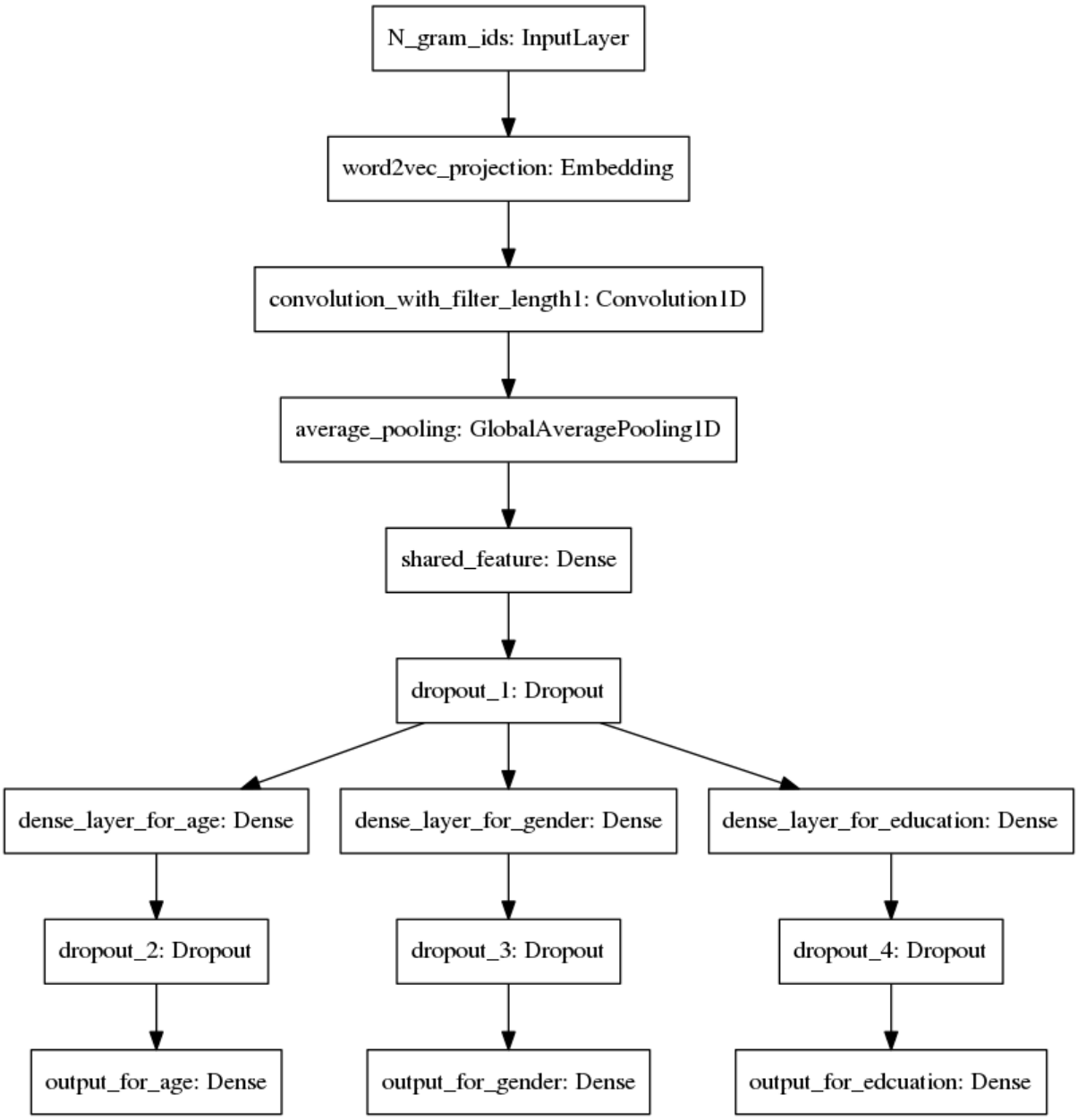

Figure 5: N-gram作为输入的多任务卷积神经网络模型。

个模型为每个类别预测一个概率值之后，将概率值取log，然后采用不同模型的线性组合作为最终结果。

4 实验结果

4.1 实验数据

实验只使用了搜狗赛题[9]中的复赛数据集，包含10万条训练数据和10万条测试数据。本赛题由三个任务构成，即推测用户的年龄、性别和学历。训练集中年龄包含7个类别的数据:0(未知年龄); 1(0-18岁); 2(19-23岁); 3(24-30岁); 4(31-40岁); 5(41-50岁); 6(51-999岁)。性别包含3个类别:0(未知);1(男性);2(女性)。学历包含7个类别:0(未知学历); 1(博士); 2(硕士); 3(大学生); 4(高中); 5(初中); 6(小学)。要求根据用于一个月内的搜索记录建立模型，预测出测试集中每个用户的年龄、性别和学历，预测时预测类别不能是0(未知)。

4.2 实验参数设置

4.2.1 句向量

我们使用了两组不同的参数来训练句向量，它们差别只在于句向量维度，一个使用了100维，另一个使用了200维。其他参数及学习过程如下:

- 负采样:5
- 单词最小出现次数:2
- 学习率最大0.2，最小0.1，每个学习率迭代5次，学习率每次下降0.025

学习完句向量之后，需要使用神经网络模型作为分类器，对于每个任务，我们都丢弃了未知类别的数据。对于维度为100的句向量，使用的是四层神经网络，第一隐藏层100个神经元，第二隐藏层50个神经元。对于维度为200的句向量，使用的也四层神经网络，第一隐藏层200个神经元，第二隐藏层60个神经元。

4.2.2 卷积神经网络模型

我们首先对每个用户的特征词进行切断和补齐，使其长度为600。之后使用了200个长度为1的卷积核，共享全连接层使用了200个神经元。每个任务单独的全连接层使用了100个神经元。所有的Dropout都设置为0.3。对于卷积神经网络，我们使用了包含未知类别的数据，只是将其权重设置为0。

4.2.3 集成

在预测时，我们使用两个句向量模型的预测结果，与卷积神经网络的预测结果取Log之后进行线性组合。两个句向量模型的预测结果在线性组合中的权重都为2，卷积神经网络的预测结果权重为1。

4.3 对比方法

我们对比了传统的TF-IDF+SVM，Wang等人提出的NBSVM[8]，单独的多任务卷积神经网络(MultiTask-CNN)，Unigram特征配合维度为100句向量模型(PV-DBOW_100_uni-gram)，Uni-gram+Bi-gram特征配合维度为100的句向量模型(PV-DBOW_100_uni+bi-gram)，Unigram特征配合维度为200句向量模型(PV-DBOW_200_uni-gram)，Uni-gram+Bi-gram特征配合维度为200的句向量模型(PV-DBOW_200_uni+bi-gram)与集成句向量和卷积神经网络的集成模型(Ensemble)的分类准确率。

4.4 实验结果

由于没有测试集的label，我们所有的实验都是在训练集上进行5折交叉验证的结果。表2列出了10次实验的平均结果。

Table 1: 不同方法在训练集上5折交叉验证的准确率

模型	年龄	性别	学历
TF-IDF+SVM	0.593	0.838	0.628
NBSVM	0.593	0.837	0.621
CNN	0.620	0.842	0.643
PV-DBOW_100_uni-gram	0.611	0.838	0.633
PV-DBOW_100_uni+bi-gram	0.623	0.848	0.647
PV-DBOW_200_uni-gram	0.611	0.839	0.634
PV-DBOW_200_uni+bi-gram	0.624	0.848	0.648
Ensemble	**0.631**	**0.853**	**0.654**

4.5 分析

实验结果表明集成的模型具有最好的分类准确率。同时，在单独的方法里，Uni-gram + Bi-gram特征配合 PV-DBOW模型取得了最好的分类准确率。PV-DBOW模型学到的实际上也是词袋特征，之所以能比TF-IDF和NBSVM取得更好的分类准确率，我们认为主要是因为PV-DBOW模

模型	年龄	性别	学历
TF-IDF+SVM	0.593	0.838	0.628
NBSVM	0.593	0.837	0.621
原始CNN	0.615	0.842	0.637
多任务CNN	0.620	0.842	0.643
原始PV-DBOW	0.611	0.839	0.634
加入N-gram的PV-DBOW	0.624	0.848	0.648
Ensemble	**0.631**	**0.853**	**0.654**

型学到的句向量维度较低，能够使用拟合能力更好的神经网络模型，而不会造成参数过多出现过拟合现象。相比于此，TF-IDF和NBSVM的输入维度太高，只能使用线性的SVM分类器，拟合能力较弱。

卷积神经网络模型的性能介于PV-DBOW模型与NBSVM之间。不同于词袋模型，卷积神经网络模型由于在卷积操作之后直接使用了平均池化，实际上这种池化丢失了词语之间的相关性，直接为所有词语使用卷积操作提取一个特征后求了平均。我们分析，可能是这种信息丢失造成了卷积神经网络在本数据上性能不如PV-DBOW。然而，卷积神经网络使用word2vec作为输入，考虑了词语之间的相关性，这点是PV-DBOW模型没有考虑的。

通过以上分析，我们可以发现，卷积神经网络和句向量分别丢失了不同的信息，这也是我们集成这两种模型的主要原因。事实上，我们也曾将NBSVM, TFIDF+SVM集成到模型中，但提升总体上非常小。

5 总结与商业应用

本文通过分析句向量和卷积神经网络两种模型在文本分类上的优缺点，提出在one-hot形式的N-gram特征上训练PV-DBOW模型和卷积神经网络，并进行集成的方法。实验结果表明，提出的模型在分类准确率方面，高于其他比较模型。

在用于商业上时，本方法适用于一般的文本分类，因此，在信息检索、Web文档自动分类、数字图书馆、分类新闻组、文本过滤等众多应用中都能使用。针对搜狗用户画像挖掘这一具体应用，由于存在很强的数据漂移现象(用户搜索热词随着时间的变化不断变化)，我们推荐每隔一段时间(10天-30天为周期)，训练一次模型。如果采取分批预测的方法，本文提出的模型已经够了。如果采取在线预测的方法，我们建议使用doc2vec的infer功能(为新文本推测句向量)[3]，另一方面，对于卷积神经网络，直接忽略系统中不存在的词，当然对于在线预测模式，我们也推荐

每隔一段时间重新训练一遍模型。

致谢

本次比赛取得的成绩很大程度上受益于南京大学计算机科学与技术系机器人智能与神经计算实验室申富饶教授的指导，和沈少锋等学长的大力支持。对于比赛过程中给予我们帮助的老师和同学，我们都饱含感激之心。

参考文献

[1] Bengio, Yoshua, et al. "A neural probabilistic language model." journal of machine learning research 3.Feb (2003): 1137-1155.

[2] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

[3] Le, Quoc V., and Tomas Mikolov. "Distributed Representations of Sentences and Documents." ICML. Vol. 14. 2014.

[4] Li, Bofang, et al. "Learning Document Embeddings by Predicting N-grams for Sentiment Classification of Long Movie Reviews." arXiv preprint arXiv:1512.08183 (2015).

[5] Lai, Siwei, et al. "How to generate a good word embedding?." (2015).

[6] Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. "Linguistic Regularities in Continuous Space Word Representations." HLT-NAACL. Vol. 13. 2013.

[7] Mesnil, Grégoire, et al. "Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews." arXiv preprint arXiv:1412.5335 (2014).

[8] Wang, Sida, and Christopher D. Manning. "Baselines and bigrams: Simple, good sentiment and topic classification." Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2. Association for Computational Linguistics, 2012.

[9] `http://www.wid.org.cn/data/science/player/competition/detail/rank/239/8`