教育部高等学校大学计算机课程教学指导委员会

中国大学生计算机设计大赛

软件开发类作品文档简要要求

作品编号：202301040039

作品名称：基于集成学习的返乡人群预测系统

作　　者：李骏杰，孙航，左文吉

版本编号：1.0

填写日期：2023 年 4 月 29 日

填写说明：

1、本文档适用于**所有**涉及软件开发的作品，包括：软件应用与开发、大数据、人工智能、物联网应用；

2、正文一律用五号宋体，一级标题为二号黑体，其他级别标题如有需要，可根据需要设置；

3、本文档为简要文档，不宜长篇大论，简明扼要为上；

4、提交文档时，以 PDF 格式提交本文档；

5、本文档内容是正式参赛内容组成部分，务必真实填写。如不属实，将导致奖项等级降低甚至终止本作品参加比赛。

一、需求分析

1.1 产品开发背景

每逢农历新年前夕，离乡的打工人都会返家过年，全国会出现大规模的高交通运输压力及堵塞的现象，简称为春运，更有地表最大规模人口迁徙之称。从2022 年 1 月 10 日起计算，2022 年人口迁移的态势每日都高于 2021 年。以 1 月 17 日的春运首日为例，全国迁徙规模指数为 345.187，与去年同期的 279.758 相比有一定上升，但仍然远低于前年；1 月 29 日，全国迁徙规模指数为 510.917，与去年同期的 276.778 相比升幅较大。

在全国，广东是唯一一个有着规模双向流动的省份，可见在吸引大量的人口聚集的同时，也是外出务工最多的省份之一。在 1 月 25 日至 29 日期间，广东省位列全国热门迁出地（出发地）及迁入地（目的地）的榜首。数据显示，广东的迁出人口数量占全国迁出人口总量的 17.17%至 19.90%，同期排入前三的省份包括江苏、浙江、四川;广东的迁入人口数量亦占全国迁入人口总量的约十分之一，同期排入前三的'省份包括四川、安徽、湖南。

除了地区经济发展因素等，人口在地区间流动的差异还受到疫情等因素影响。2022 年，北京的迁徙规模不及常年，在年初的大数据中，北京在前十名迁出、迁入的城市中仅出现 2 次，且排名相对靠后。

并且由于近两年新冠疫情带来的影响，导致农民工返乡情况的不确定因素增大，因此，对于农民工春节返乡情况的预测是十分必要的。

1.2 产品应用场景分析

现如今伴随着时代的不断发展，国家对于城市的建设也越来越成熟。在这样的情况之下，我国绝大多数城市地区的发展都已经进入了饱和的状态。而多数在城市务工的底层劳动人员，也都面临着失业的困境。

春节前后是各大运营商营销活动的重要节点，更是一年当中农村促销各类电信业务的黄金时期。为了更好支撑春节期间电信业务市场业务拓展，有效支撑春节返乡营销保障专项工作。本文提出一种基于集成学习的返乡人群预测系统，对

返乡用户进行挖掘识别，从而为一线营销提供依据，进而提前抢占市场先机，提高电信用户市场保有率。

二、集成学习算法介绍

2.1 集成学习简介

集成学习是训练多个机器学习模型并将其输出组合在一起的过程。组织以不同的模型为基础，致力构建一个最优的预测模型。组合各种不同的机器学习模型可以提高整体模型的稳定性，从而获得更准确的预测结果。集成学习模型通常比单个模型更可靠，因此，它们经常在许多机器学习竞赛中获胜。工程师可以使用多种技术来创建集成学习模型。而简单的集成学习技术包括平均不同模型的输出结果，同时还开发了更复杂的方法和算法，专门用于将许多基础学习者/模型的预测结果组合在一起。

出于多种原因，机器学习模型可能会彼此不同。不同的机器学习模型可以对总体数据的不同样本进行操作，可以使用不同的建模技术，并且使用不同的假设。

想象一下，如果你加入由不同专业人员组成的团队，那么肯定会有一些你知道和不知道的技术，假设你正在和其他成员一起讨论一个技术主题。他们也像你一样，只对自己的专业有所了解，而对其他专业技术一无所知。但是，如果最终能将这些技术知识组合在一起，将会对更多领域有更准确的猜测，这是集成学习的原理，也就是结合不同个体模型（团队成员）的预测以提高准确性，并最大程度地减少错误。

所有的模型都有一定的误差。一个模型的误差将不同于另一个模型产生的误差，因为模型本身由于上述原因而不同。当检查所有的错误时，它们不会聚集在某一个答案周围，而是广泛分布。不正确的猜测基本上分散在所有可能的错误答案上，并相互抵消。与此同时，来自不同模型的正确猜测将聚集在正确的答案周围。当使用集成训练方法时，可以找到更可靠的正确答案。

2.2 集成学习的分类

常见的集成学习框架有三种：

1.Bagging

并行：各个基模型之间不存在强依赖关系，代表是随机森林算法。每个基模

型基于对训练集进行有放回抽样得到子训练集（0.632 采样法）进行训练。使用投票法综合基模型的预测结果，票数最多的类别为预测类别。

2.Boosting

串行：基模型之间存在强依赖关系，必须串行生成。每个基模型都会在前一个基模型学习的基础上进行学习。综合方式为加权法。

3.Stacking

串行。先用全部数据训练好基模型，然后基模型对每个训练样本进行的预测，其预测值将作为训练样本的特征值，得到新的训练样本，然后基于新的训练样本进行训练得到下一个基模型，得到最终预测结果。

2.3 原理分析

2.3.1 个体学习器

集成学习的第一个问题就是如何得到若干个个体学习器。这里有两种选择。第一种就是所有的个体学习器都是一个种类的，或者说是同质的（homogeneous），同质集成中的个体学习器也称为“基学习器”（base learner），相应的学习算法称为“基学习算法”（base learning algorithm）。比如都是决策树个体学习器，或者都是神经网络个体学习器。第二种是所有的个体学习器不全是一个种类的，或者说是异质的（heterogeneous）。比如我们有一个分类问题，对训练集采用支持向量机个体学习器，逻辑回归个体学习器和朴素贝叶斯个体学习器来学习，再通过某种结合策略来确定最终的分类强学习器。这时个体学习器一般不称为基学习器，而称作“组件学习器”（component leaner）或直接称为个体学习器。

弱学习器（weak learner）：指泛化性能略优于随机猜测的学习器：例如在二分类问题上精度略高于 50%的分类器。

集成学习的直觉是结合多个个体的能力，获得远超个体的集体能力优势。这种直觉在实际上对于“弱学习器”是非常符合的。故很多集成学习的研究也都是针对弱学习器，而基学习器有时也被直接称为弱学习器。一般经验中，如果把好坏不一的东西掺杂在一起，那么最终结果很可能是整体效果比最坏的东西要好一些，但又比最好的那个要坏一些，那么这种情况下不如就让最好的单独去工作，

而不要参与混合。

根据个体学习器生成方式的不同，目前集成学习方法大致可分为两大类，第一个是个体学习器之间存在强依赖关系，一系列个体学习器基本都需要串行生成的序列化方法，代表算法是 boosting 系列算法，第二个是个体学习器之间不存在强依赖关系，一系列个体学习器可以并行生成，代表算法是 bagging 和随机森林（Random Forest）系列算法。

2.3.2 方差与偏差

偏差：描述样本拟合出的模型的预测结果的期望与样本真实结果的差距，要想偏差表现的好，就需要复杂化模型，增加模型的参数，但这样容易过拟合。

方差：描述样本上训练出来的模型在测试集上的表现，要想方差表现的好，需要简化模型，减少模型的复杂度，但这样容易欠拟合。

在集成学习框架中，通过计算模型的期望和方差，我们可以得到模型整体的期望和方差。为了简化模型，我们假设基模型的期望为μ，方差为σ^2，模型的权重为γ，两两模型间的相关系数为ρ。因为集成学习是加法模型，那么有：

模型的总体期望：

$$E(F) = E(\sum_{m}^{M} r_m f_m)$$

$$= \sum_{m}^{M} r_m E(f_m)$$

模型总体方差（利用协方差的性质，协方差与方差的关系）：

$$Var(F) = Var(\sum_{m}^{M} r_m f_m)$$

$$= \sum_{m}^{M} Var(r_m f_m) + \sum_{m \neq n}^{M} Cov(r_m f_m, r_n f_n)$$

$$= \sum_{m}^{M} r_m^2 Var(f_m) + \sum_{m \neq n}^{M} \rho r_m r_n \sqrt{Var(f_m)} \sqrt{Var(f_n)}$$

$$= mr^2\sigma^2 + m(m-1)\rho r^2\sigma^2$$

$$= mr^2\sigma^2(1-\rho) + m^2 r^2 \sigma^2 \rho$$

模型的准确度可由偏差和方差共同决定：

$$Error = bias^2 + var + \xi$$

2.3.3 Bagging

对于 Bagging 来说，每个基模型的权重等于 1/m 且期望近似相等，所以可以得到：

$$E(F) = \sum_{m}^{M} r_m E(f_m)$$

$$= m\frac{1}{m}\mu$$

$$= \mu$$

$$Var(F) = mr^2\sigma^2(1-\rho) + m^2r^2\sigma^2\rho$$

$$= m\frac{1}{m^2}\sigma^2(1-\rho) + m^2\frac{1}{m^2}\sigma^2\rho$$

$$= \frac{\sigma^2(1-\rho)}{m} + \sigma^2\rho$$

通过上式可以得到：整体模型的期望等于基模型的期望，这也就意味着整体模型的偏差和基模型的偏差近似。整体模型的方差小于等于基模型的方差，当且仅当相关性为 1 时取等号，随着基模型数量增多，整体模型的方差减少，从而防止过拟合的能力增强，模型的准确度得到提高。但是，模型的准确度一定会无限逼近于 1 吗？并不一定，当基模型数增加到一定程度时，方差公式第一项的改变对整体方差的作用很小，防止过拟合的能力达到极限，这便是准确度的极限了。所以这就是为什么 Bagging 中的基模型一定要为强模型，如果 Bagging 使用弱模型则会导致整体模型的偏差变大，而准确度降低。

随机森林是经典的基于 Bagging 框架的模型，并在此基础上通过引入特征采样和样本采样来降低基模型间的相关性，在公式中显著降低方差公式中的第二项，略微升高第一项，从而使得整体降低模型整体方差。

2.3.4 Boosting

对于 Boosting 来说，由于基模型共用同一套训练集，所以基模型间具有强

相关性，故模型间的相关系数近似等于 1，针对 Boosting 化简公式为：

$$E(F) = \sum_{m}^{M} r_m E(f_m)$$

$$Var(F) = mr^2\sigma^2(1-\rho) + m^2r^2\sigma^2\rho$$

$$= m\frac{1}{m^2}\sigma^2(1-1) + m^2\frac{1}{m^2}\sigma^2 1$$

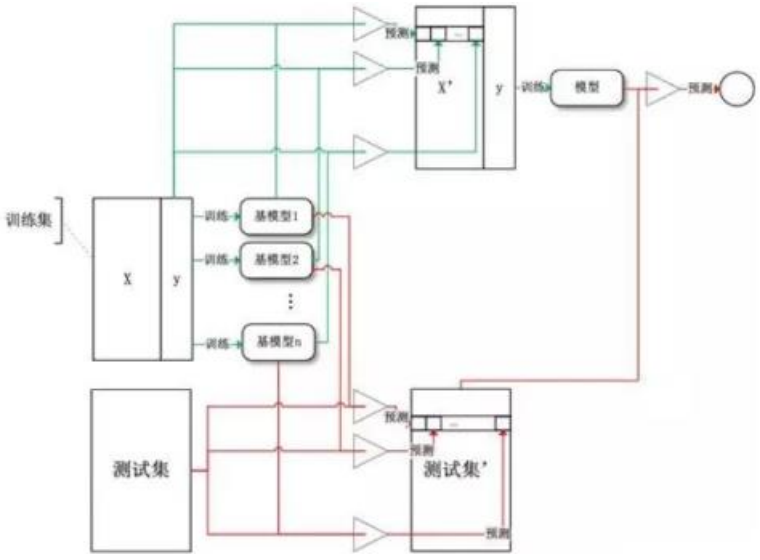
$$= \sigma^2$$

通过上式可以得到：整体模型的方差等于基模型的方差，如果基模型不是弱模型，其方差相对较大，这将导致整体模型的方差很大，即无法达到防止过拟合的效果。因此，Boosting 框架中的基模型必须为弱模型。

此外 Boosting 框架中采用基于贪心策略的前向加法，整体模型的期望由基模型的期望累加而成，所以随着基模型数的增多，整体模型的期望值增加，整体模型的准确度提高。

基于 Boosting 框架的 GBDT 模型中基模型也为树模型，同随机森林一样，也可以对特征进行随机抽样来使基模型间的相关性降低，从而达到减少方差的效果。

2.3.5 Stacking



将训练好的所有基模型对训练基进行预测，第 j 个基模型对第 i 个训练样本的预测值将作为新的训练集中第 i 个样本的第 j 个特征值，最后基于新的训练集

进行训练。同理，预测的过程也要先经过所有基模型的预测形成新的测试集，最后再对测试集进行预测。Stacking 算法分为 2 层，第一层是用不同的算法形成 T 个弱分类器，同时产生一个与原数据集大小相同的新数据集，利用这个新数据集和一个新算法构成第二层的分类器。

Stacking 就像是 Bagging 的升级版，Bagging 中的融合各个基础分类器是相同权重，而 Stacking 中则不同，Stacking 中第二层学习的过程就是为了寻找合适的权重或者合适的组合方式。

三、算法构建

3.1 情景分析

春节返乡期间，外地人员返乡，由于地市的变更会带来手机话费、流量资费的提高。因此返乡期间是运营商进行用户营销以及策反的一个绝佳时机。通过构建返乡模型，从而实现目标用户的精准营销与挖掘。同时模型预测输出的返乡用户，也将作为一线人员进行返乡用户营销的有利支撑。

3.2 数据准备

3.2.1 数据选取

以目标用户前几个月的通话行为信息建立宽表，然后探索返乡用户的通话行为特征。首先，用 2021 年 10、11、12 月的目标用户的通话行为特征，并用 2022 年春节是否返乡来确定正负样本进行建模，选择较优的模型。然后，使用此模型对 2022 年 8、9、10 月的目标用户的通话特征进行预测。此次建模共获取训练目标用户数 248 万，预测用户规模 280 万。主要选取以下几个维度：

（1）用户基本属性信息：本地网信息、套餐名称、归属网格、付费类型等；

（2）用户交往圈信息：交往号码，交往号码归属本地网等；

（3）用户通话信息：通话时长、通话次数、节假日通话次数、主叫次数等。

3.2.2 数据处理

通过数据选取步骤，得到数据建模宽表，由于样本数据中常常包含许多含有噪声、不完整，甚至不一致的数据，对数据挖掘所涉及的数据对象必须进行预处理。数据预处理主要包括：数据筛选、数据变量转换、缺失值处理、坏数据处理、数据归一化等。结合业务相关经验与知识，对本文所涉及的缺失数据较少的字段进行相关插补，对离群点数据做删除处理。

本文使用一个简易的模型，将数据切割成 60 份，对每一份数据单独作为验证集，如果验证集的 AUC 几乎接近于 0.5，则验证集中的数据大多数为干扰数据。

```
gbc = GradientBoostingClassifier()
gbc_test_preds = model_train(gbc, "GradientBoostingClassifier", 60)
```

测试结果为：

KFlod	AUC
0	0.9034
1	0.9112
2	0.9045
3	0.9006
4	0.9013
5	0.8986
6	0.9009
7	0.9116
8	0.9245
9	0.8902
10	0.8995
11	0.9047
12	0.9291
13	0.8980
14	0.9279
15	0.8995
16	0.9080
17	0.8942
18	0.9179
19	0.9044
20	0.8937
21	0.9138
22	0.9024
23	0.9091
24	0.8937
25	0.9173
26	0.9047
27	0.9010
28	0.9047
29	0.9144
30	0.8984
31	0.9079
32	0.9240
33	0.8899
34	0.8780
35	0.8942
36	0.9112
37	0.9221
38	0.9273
39	0.9137
40	0.9206

41	0.9053
42	0.9033
43	0.9193
44	0.9141
45	0.9087
46	0.8957
47	0.9142
48	0.9179
49	0.9128
50	0.5608
51	0.5328
52	0.5020
53	0.5067
54	0.4888
55	0.5065
56	0.5163
57	0.5019
58	0.5235
59	0.4842

很明显可以看出，50~59 的数据为干扰数据，所以剔除干扰数据：

```
train = train[:50000]
label = label[:50000]
```

3.2.3 特征选择

在有限的条件下，本次模型主要选取通话特征维度进行建模，并设计以下宽表字段：

字段	描述
opp_nbr	目标用户号码
opp_latn_id	目标用户地市
fam_nbr	农村家庭号码
fam_latn_id	家庭号码所在地
avg_all_cnts	平均通话次数
avg_all_dur	平均通话时长
avg_all_days	平均通话天数
avg_holidays_cnts	节假日平均通话次数
avg_weekday_cnts	工作日平均通话次数
weekholi_cnt_ratio	工作日和节假日通话次数占比
avg_holidays_dur	节假日平均通话时长
avg_weekday_dur	工作日平均通话时长
weekholi_dur_ratio	工作日和节假日通话时长占比
avg_day_time_cnts	平均白天通话次数
avg_night_time_cnts	平均夜晚通话次数
daynight_cnt_ratio	白天和夜晚通话次数占比
avg_day_time_dur	平均白天通话时长
avg_night_time_dur	平均夜晚通话时长
daynight_dur_ratio	白天和夜晚通话时长占比
if_fanxiang	是否返乡

3.3 建立模型

建立返乡模型，最终是预测用户是否返乡，是一个二分类的问题，因此此次建模应当使用分类算法。通过利用 python 软件中的 sklearn 库，对数据进行建模。在建模过程中选取了 GradientBoostingClassifier、HistGradientBoostingClassifier、XGBClassifier、LGBMClassifier、CatBoostClassifier 算法作为基础模型对数据进行了训练，并经过反复迭代训练，直至样本数据模型效果达到最优。

```
gbc = GradientBoostingClassifier(
    n_estimators=50,
    learning_rate=0.1,
    max_depth=5
)
hgbc = HistGradientBoostingClassifier(
    max_iter=100,
    max_depth=5
)
xgbc = XGBClassifier(
    objective='binary:logistic',
    eval_metric='auc',
    n_estimators=100,
    max_depth=6,
    learning_rate=0.1
)
gbm = LGBMClassifier(
    objective='binary',
    boosting_type='gbdt',
    num_leaves=2 ** 6,
    max_depth=8,
    colsample_bytree=0.8,
    subsample_freq=1,
    max_bin=255,
    learning_rate=0.05,
    n_estimators=100,
    metrics='auc'
)
cbc = CatBoostClassifier(
    iterations=210,
    depth=6,
    learning_rate=0.03,
    l2_leaf_reg=1,
    loss_function='Logloss',
    verbose=0
)
```

通过 StackingClassifier 将 6 个模型进行 Stack，Stack 模型用 LogisticRegression。

```
estimators = [
    ('gbc', gbc),
    ('hgbc', hgbc),
    ('xgbc', xgbc),
    ('gbm', gbm),
    ('cbc', cbc)
]
clf = StackingClassifier(
    estimators=estimators,
    final_estimator=LogisticRegression()
)
```

特征筛选思路：先将模型训练好，然后对验证集进行测试得到基础 AUC，之后循环遍历所有特征，在验证集上对单个特征进行 mask 后，得到 mask 后的 AUC，评估两个 AUC 的差值，差值越大，则说明特征重要性越高。

最后的训练结果为：

KFold	AUC
0	0.9069
1	0.9091
2	0.9156
3	0.9059
4	0.9075
5	0.9083
6	0.9007
7	0.9173
8	0.9160
9	0.9146
overall	0.9100

返乡预测模型固化下来后，对 2022 年目标用户进行预测，并在某地市进行试点，抽取预测返乡用户数 15 万，最终的正确率达到 91.17%。

四、项目总结

4.1 商业推广

本文利用集成学习技术，将用户的通话行为作为特征，预测用户是否返乡，经过数据验证，模型能够较为有效地预测哪些是返乡用户。在具体的商用方面，该模型能够满足绝大多数电商机构的需要。伴随着科学技术的发展和社会的进步，人们的消费意识逐渐提升，电商行业也得以迅猛发展。返乡预测系统是个需求量巨大的行业，各大厂都在做一个值得信赖的返乡预测系统。

通过对用户行为数据的分析，我们可以了解到很多用户相关的信息，包括用户的有效来源、用户对于产品的认知程度、用户对于产品的喜好方向，能够帮助我们对用户有更多的认识。同时，在用户行为中，也蕴含着极高的商业价值，而这里的商业价值指的是，通过对用户行为数据的分析，识别其中蕴含的成交可能性，将用户行为数据直接转化成为线索，利用线索中蕴含的用户意向数据、用户基本信息数据，可以分析出用户的意向、用户的职业等关键信息。研究这些信息，抓住用户心理，对用户进行进一步的营销，进行销售跟进或开展免费试用等营销活动，促使用户成交，达成提升营收的目的。

当我们将这些用户行为中隐藏的商业价值全部挖掘出来以后，我们就挖掘到了一批极具成交潜力的用户。这些用户数据经过一定的数据清洗处理，过滤掉不可联系的用户（没有手机号、邮箱，或者可以推送消息的终端），也就形成了线索，可以送达业务一线，由对应的团队进一步地培育、跟进。根据培育、跟进的数据结果反馈，我们可以再调整策略，力求最大化开发用户行为的价值。

4.2 社会价值

可以说自然界的所有生物都具有一定程度的预测功能。比如即将下雨，蚂蚁可以通过周围环境准确判断并且搬家躲避；比如地震前，就有许多动物表现异常，也是求生存的一种本能；豹子通过预测猎物的强弱大小，奔跑的路径、方向和速度，适时采取拦截的方式获取猎物，以取得最大的成功率。

而我们人类做任何事情更是运用预测的方法，所以有“凡事预则立，不预则

废”的说法，凡事未雨绸缪是人生规避风险的一种方法。中国人自古更讲求谋略，凡事谋在先，则能知己知彼，百战百胜，也就是凡事要计划在先，不做无准备的事。比如我们到某地，这个简单的过程就包含了很多预测，我们预测哪条路能够到达目的地，需要采取哪种方法可以到达目的地，甚至还要预测大概耗时多久，路上是否顺畅以及会发生什么事情等等。

返乡预测系统的社会价值，即是指对人类生存和发展来说，它是必需的、有益的。随着发展的社会化和社会的现代化，返乡预测系统的社会价值问题就更为明显了。应该强调，对社会进步的评价不仅是科学技术的，更是社会的。因此，社会评价标准实际往往影响人类活动的价值取向和选择。社会所需要的是对人类活动及其社会化的适宜发展，并要求发展与社会经济相适应的适用技术。社会需要决定着人类活动的发展，并形成对人类活动发展的社会价值观念。

项目成果将为返乡预测应用技术提升精准度服务，在实际应用层面深入理解提问者需求，为公众提供更具规范性的返乡预测结论。本项目将作为原子能力助力返乡预测应用技术精准度提升，探索下一代人类活动预测技术，具有广泛的技术和公益价值。在科普宣传和应用场景两大类需求场景下为抗疫人员提供有效的数据工具，助力政府、企业、机构的抗疫、防控和宣传工作。

五、参考文献

[1] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), 1746–1751.

[2] Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A Convolutional Neural Network for Modelling Sentences. Acl, 655–665.

[3] Armand Joulin, Edouard Grave , Piotr Bojanowski, omas Mikolov. Bag of Tricks for Efficient Text Classification. arXiv, 2016.

[4] Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio. Neural Machine Translation By Jointly Learning To Align And Translate. ICLR, 2015.

[5] Sanjeev Arora, Yingyu Liang, Tengyu Ma. A Simple but Tough-to-Beat Baseline for Sentence Embeddings,2016.

[6] C Zhou, C Sun, Z Liu, F Lau. A C-LSTM neural network for text classification. arXiv, 2015.

[7] X Zhang, J Zhao, Y LeCun. Character-level convolutional networks for text classification. NIPS, 2015.

[8] S Lai, L Xu, K Liu, J Zhao. Recurrent convolutional neural networks for text classification. AAAI, 2015.

[9] Z Lin, M Feng, CN Santos, M Yu, B Xiang. A structured self-attentive sentence embedding. arXiv, 2017.

[10] J Howard, S Ruder. Universal language model fine-tuning for text classification. arXiv, 2018.

[11] J Devlin, MW Chang, K Lee, K Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv, 2018.

[12] J Wang, Z Wang, D Zhang, J Yan. Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification. IJCAI, 2017.

[13] J Chen, Y Hu, J Liu, Y Xiao, H Jiang. Deep short text classification with knowledge powered attention. AAAI,2019.

[14] H Ren, L Yang, E Xun. A sequence to sequence learning for Chinese grammatical error correction. NLPCC,2018.

[15] Y Hong, X Yu, N He, N Liu, J Liu. FASPell: A Fast, Adaptable, Simple, Powerful Chinese Spell Checker Based On DAE-Decoder Paradigm. EMNLP, 2019.

[16] I Antonellis, H Garcia-Molina. Simrank++ query rewriting through link analysis of the clickgraph. WWW'08.

[17] G Grigonytė, J Cordeiro, G Dias, R Moraliyski. Paraphrase alignment for synonym evidence discovery. COLING,2010.

[18] X Wei, F Peng, H Tseng, Y Lu, B Dumoulin. Context sensitive synonym discovery for web search queries. CIKM,2009.

[19] S Zhao, H Wang, T Liu. Paraphrasing with search engine query logs. COLING, 2010.

[20] X Ma, X Luo, S Huang, Y Guo. Multi-Distribution Characteristics Based Chinese Entity Synonym Extraction from The Web. IJISA, 2019.

[21] H Fei, S Tan, P Li. Hierarchical multi-task word embedding learning for synonym prediction. KDD,2019.

[22] M Qu, X Ren, J Han. Automatic synonym discovery with knowledge bases. KDD,2017.

[23] J Shen, R Lyu, X Ren, M Vanni, B Sadler. Mining Entity Synonyms with Efficient Neural Set Generation. AAAI,2019.

[24] A Vaswani, N Shazeer, N Parmar. Attention is all you need. NIPS, 2017.

[25] Berant J, Chou A, Frostig R, et al. Semantic Parsing on Freebase from Question-Answer Pairs. EMNLP,2013.

[26] Cai Q, Yates A. Large-scale Semantic Parsing via Schema Matching and Lexicon Extension. ACL,2013.

[27] Bordes A, Chopra S, Weston J. Question answering with subgraph embeddings. arXiv,2014.

[28] Dong L, Wei F, Zhou M, et al. Question Answering over Freebase with Multi-Column Convolutional Neural Networks. ACL,2015.

[29] E Malmi, S Krause, S Rothe, D Mirylenka. Encode, Tag, Realize: High-Precision

Text Editing. arXiv, 2019.