**文章编号：**

# 知识图谱中实体相似度计算研究*

李阳[1]

（1.华东理工大学计算机科学与工程系，上海，200237）

**摘要**：实体相似度的计算有诸多应用，例如电商平台的相似商品推荐，医疗疗效分析中的相似病人组等。在知识图谱的实体相似度计算中，给出了每个实体的属性值，并对部分实体进行相似度的标注，要求能得到其他实体之间的相似度。本文把该问题归结为监督学习问题，提出一种通用的实体相似度计算方法，通过清洗噪声数据，对数值、列表以及常文本等不同数据类型进行预处理，使用 SVM，Logistic 回归等分类模型、Random Forest 等集成学习模型以及排序学习模型进行建模，得到了较好的结果。

**关键词**：实体相似度；监督学习；分类模型；集成学习

**中图分类号**：TP391　　**文献标识码**：A

# A Research on Entities Similarity Calculation in Knowledge Graph

Li Yang[1]

(1. Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China)

**Abstract**: Entities similarity is useful in many areas, such as recommending similar merchandises in E-commerce platforms, analyzing similar patients in healthcare, etc. During the calculation of entity similarity in knowledge graph, the attributes of every entity and portion of entity pairs' similarity are given. We are supposed to calculate similarity of other entity pairs. This article defines this task as a supervised learning problem. We propose a general method to calculate entity similarity, firstly preprocess the data and then use classification model, including SVM, Logistic Regression, and integrated learning model, such as Random Forest, and Learning to rank model. After comparing the different methods, the best result is achieved.

**Key words:** Entity Similarity; Supervised Learning; Classification Model; Ensemble Learning

## 1 引言

知识图谱（knowledge graph）是时下非常热门的研究领域。它本质上是一种语义网络，其结点代表实体（entity）或者概念（concept），边代表实体/概念之间的各种语义关系。对于一个包含众多实体的知识库，我们除了关注实体本身的信息外，还需要关注实体与实体之间的关联信息。其中面临的一个问题就是：给定两个实体，如何判断它们之间是否相似，以及相似的程度有多高。

实体间的相似是指实体之间在深层语义上的相似，而非只关注表层信息的传统相似度。例如“刘德华”和“张学友”都是香港歌手，他们之间有很多共同的属性；相对的，“刘德华”和“马德华”虽然在名字上很相似，但他们之间却没有太多的属性共性，所以 “张学友”比“马德华”对于“刘德华”来说更相似。同样的，“相宜本草”和“佰草集”都是化

* **收稿日期：**　　**定稿日期：**

**基金项目**：心血管疾病与肿瘤疾病中西医临床大数据处理分析与应用研究（2015AA020107）

**作者简介**：李阳（1992--），男，硕士研究生，主要研究领域为自然语言处理与大数据挖掘。

妆品品牌，所以它们之间的相似度就要比“相宜本草”和“薰衣草”要高。

判断实体之间相似度需要首先理解实体的语义信息，传统的字符相似度方法不可行。在数据已经结构化存储的知识库中，实体的属性可以作为相似度判断的主要依据。然而，实体的属性多种多样，如何判断何为重要属性和如何计算属性上的相似度成为了解决问题的关键。

百度知识图谱竞赛提供了实体数据和训练数据。实体数据给出了实体所包含的所有属性，训练数据给出了部分实体对的相似度打分。其中实体属性有多种不同的数据类型。我们提出了一种通用的实体相似度计算方法，计算各个对应属性之间的相似度作为特征，使用 SVM，Logistic 回归等分类模型、RandomForest 等集成学习模型以及排序学习模型进行建模，使用监督学习的方法。对比不同方法的效果之后，又引入了一种文档主题生成模型（LDA）对文本型属性特征进行优化，最终得到了较好的结果。

本文的主要贡献为：

1）提出了一种用于计算知识图谱中实体相似度的通用方法，可以处理实体的各种类型的属性值，包括数值型、列表型和文本型。

2）在文本型数据的相似度计算中使用了语义模型，增加实体在语义上的相似度。

3）在实体的相似度计算中，使用集成学习的方法，提高分类的正确率。

## 2 相关工作

目前，国内外在知识图谱中对实体相似度的计算有很多研究，主要分为两个方向：一个是相似实体推荐，另一个是知识推理[2]。

相似实体推荐在诸多领域有着广泛应用。在电商和搜索引擎中，推荐系统（Recommender System）扮演着举足轻重的角色，它能向用户推荐有用的对象[1]，基于用户相似度构建用户群体可以使推荐结果应用于群体中的所有用户，基于对象相似度构建对象群体可以使推荐结果包含多个同类对象供用户挑选。在医疗领域中，2012 年，IBM 的 Jimeng Sun 等人提出了一种有监督的学习方法，使用基于广义马氏距离的复合距离集成方法来评估病人之间的相似度[3]，病人数据使用关键的临床指标来表示。这样，医生的诊断可以利用相似病人之间的信息来辅助决策。但是样本数据都是单一的数值类型，而在知识图谱中的实体数据中，实体的属性类型多样，有数值型、列表型以及文本型等。而文本型数据的相似度计算是目前研究的重点。

对于计算文本相似度的方法，目前主要分为两个方向。一种是基于统计的方法，另一种是基于语义分析的方法[4]。

基于统计方法的相似度计算通常采用向量空间模型[4] (vector space model, VSM)进行文本的表示，将文本表示为特征词集合，将这些特征词作为最基本的元素，然后统计文本中这些特征词的词频得到特征词，通过计算在特征词向量空间上的相似度来代表文本的相似度。使用 VSM 模型，关键是计算词的权重，通常使用 TFIDF[4]向量来计算。VSM 模型使用特征词在文本的统计特性，能有效地对文本进行表示。但是它并没有考虑文本中的语义信息，而且产生的特征向量维度较高，并具有很高的稀疏性，影响了计算的效能。

由于 VSM 的局限性就产生了一种基于语义的分析方法。语义分析是指从词语间的语义关系，考虑词语的相似性，即近义词等。Sussna M 通过分析词义网中的节点密度、深度和链接关系提出了一种基于词义网边的词语相似度计算方法[5]。还有基于语义词典 WordNet

的方法[6-9]，WordNet 中使用同义词集合作为基础构建单位。在一个同义词集合中的词所代表的意思是相近的，有些情况下这些词之间可以相互交换。另外，还有一种文档主题生成模型，LDA 模型[10]，它能挖掘文本中深层的语义信息，虽然也是基于统计的方法，但却能够有效的降低维度，提高效率。

最近关于实体相似度的研究中，李荣等[11] 提出一种综合的概念相似度计算方法，在计算概念相似度时，不仅考虑概念本身的语义，而且考虑概念的属性和上下文结构，进行本体映射。刘杰[13] 提出一种通过对特征语义进行分析，定义不同实体特征相似度的计算模型和权值计算模型，实现特征权值的自动计算，用于处理本体映射问题。薛咏等[13]使用一种混合式的相似度算法计算实体相似度，将结构语义与元素级相似度相结合。但是上述的三种方法都是解决异构本体语义一致性与本体复用问题，将多个指向现实世界的实体映射在一起。而本文的任务是计算不同实体之间的相似程度，而且目前并没有相关的完整的计算方法，所以本文结合语义模型，提出了一种用于计算知识图谱中实体相似度的通用方法。

## 3 算法设计

本文解决的问题是在实体集合$E=\{e_1,e_2,\dots,e_n\}$中，对于分类 C，类标签为[0,1,2,3,4]，给定实体相似度种子集合$S=\{<e_i,e_j>|e_i,e_j\in E,<e_i,e_j>\in C\}$以及待分类集合$D=\{<e_x,e_y>|e_x,e_y\in E\}$，求出集合 D 中每个元素即实体对所属的分类。

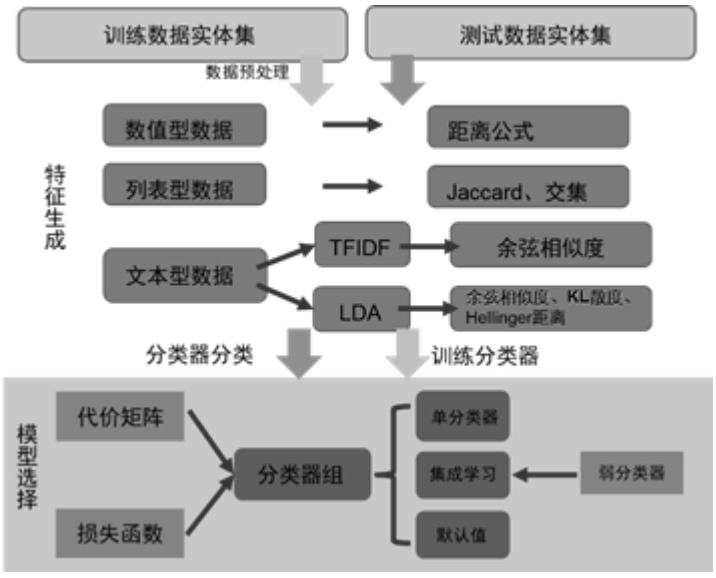


图 1 算法整体框架图

由于集合 S 和 D 中的每个实体对都同属一个 type（总共有六种 type：Movie，MusicRecording，ShowSeries，SoftwareApplication，TVSeries，VideoGame），并且 S 中没有相似度为 0 的实体对，所以我们做出以下假设：

**假设 1**　不同类别（type）的实体对相似度为 0。

相似度为 0 代表实体之间很不相似。对于不同类别的实体，其属性的维度不同，共同的属性也很少，而且在实际应用中计算不同类别实体的相似度也没有意义，所以我们假设不同类别的实体之间相似度为 0。

**假设 2**　实体与自身的相似度为 4。

**假设 3** 两个实体的相似度满足对称性，即实体 1 和实体 2 的相似度等于实体 2 和实体 1 的相似度。

根据以上假设，我们可以把问题简化为：在同 type 的实体集合$E' = \{e_1, e_2, \ldots, e_n\}$中，对于分类 C，类标签为[1,2,3,4]，给定实体相似度种子集合$S = \{< e_i, e_j > | e_i, e_j \in E', < e_i, e_j > \in C\}$以及待分类集合$D = \{< e_x, e_y > | e_x, e_y \in E'\}$，求出集合 D 中每个元素即实体对所属的分类。

本文算法的整体框架如图 1 所示。下面针对算法的各个步骤进行详细解释。

### 3.1 数据预处理

通过对训练数据的观察，我们发现了一些噪声数据。在训练集中存在这样的实体对：实体 A 和 B 的相似度为 c，但是 B 和 A 的相似度却不是 c，不符合我们的假设 3；实体 A 与它自身之间的相似度不是 4，不符合我们的假设 2；两个实体 A 和 B 同属一个系列，应当很相似，然而它们之间的相似度却不高，比如夺宝奇兵 3 和夺宝奇兵 4 的相似度是 1，爱情手册 2 和爱情手册 3 的相似度却是 2，不符合实际情况。这些噪声数据的存在会影响分类器的效果。图 2 为噪声数据的数量和分布情况。通过统计我们发现噪声数据数量不多，共占训练集样本总数的 0.9%，为了提高模型的精度，将这些噪声数据剔除。

另外，对于列表型的实体属性存在表达不统一的情况。例如，”inLanguage”属性，有些实体属性值为汉语普通话、普通话、简体中文、国语、汉语、普通话国语、中文等，它们都表示同一种语言，但是说法完全不相同，这种不一致性将会影响对二者相似度的评判。为了后续的处理，我们对数据进行归一化。将上述的这些表达方式都改成汉语，还有其它的类似改动。

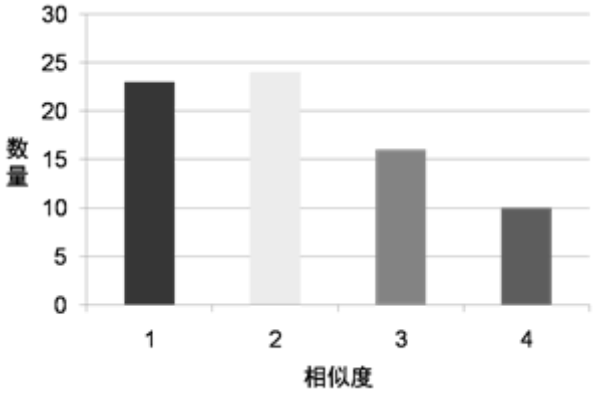


图 2 噪声数据分布

### 3.2 特征生成

实体属性值的数据类型有三种：数值型，列表型，文本型。对于不同类型的数据使用不同的相似度计算方法来度量它们之间的相似性。

$$D(d_i, d_j) = \frac{|d_j - d_i|}{Max(d_n) - Min(d_n)} \quad (1)$$

对于数值类型的属性值 d，我们使用公式（1）来计算两个实体在该属性上的相似度。通过该式可以看出，D 的值域为[0,1]，而且$d_i$和$d_j$之间相差的越大，D 值就越大，表示它们

之间的相似度越小。

列表型的属性表示其值是某集合中的一个或多个元素，比如电影实体的演员属性，其值是全体演员集合中的多个元素。列表型的数据可以作为集合进行处理。对于列表型的属性值，我们使用两种指标来衡量它们的相似性。一种是计算交集的个数，交集个数越大表示它们之间越相似。另外一种是 Jaccard 相似度[14]，计算公式如公式（2）。

$$Jaccard(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

对于两个集合 A 和 B ，Jaccard 相似度的值域是[0,1]，值越大表示它们之间的相似性越高。

对于文本型的属性，其值是一段文字信息。文本型的数据中包含很多潜在的语料信息，对它们的很好利用将会在很大程度上反映两个实体之间的相似性。

我们首先使用了基于向量空间模型的 TFIDF 方法。TFIDF 是一种统计方法，用来评估一个字词对于一个文档的重要程度。TFIDF 是词频(TF)*逆文档频率(IDF)。所以这里需要统计单位是词。先使用中文分词工具对文本数据进行分词后，计算得到每个文本数据的 TDIDF 向量。得到向量之后，使用余弦相似度[15]来衡量它们之间的相似性。那么对于 A，B 两个 n 维的 TDIDF 向量，它们之间的余弦相似度可通过公式（3）计算。

$$CosSim = \frac{\sum_{i=1}^{n}(A_i \times B_i)}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \tag{3}$$

通过上式可以看出余弦相似度值域为[0,1]，而且值越大相似性也越高。

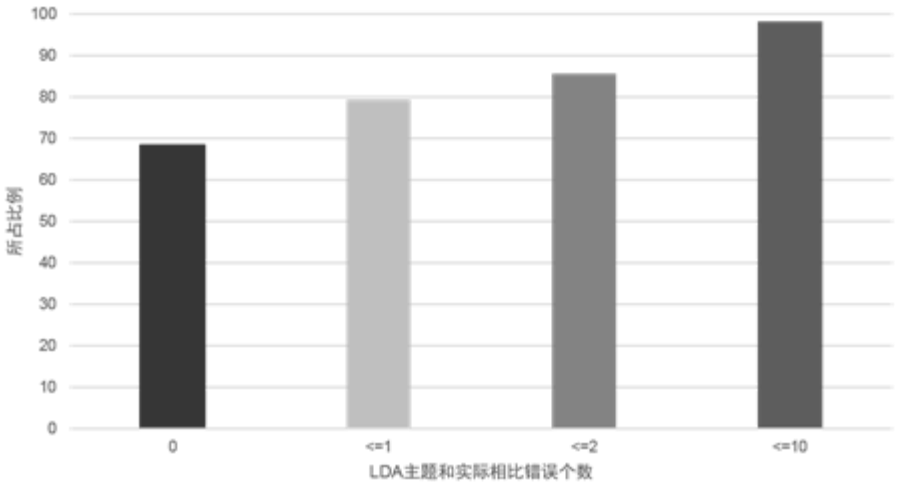


图 3 LDA 主题预测结果

TFIDF 通过统计的方法来用向量表示文本文档，它作用的对象是文档中的词。如果文档太长，将导致 TFIDF 向量维度特别高，不利于计算。而且，在文本中通常会包含一些潜在的重要信息，如文档的主题，表示主题的词不一定在文档中出现，但是它能反映文本的一个重要特性。所以我们引入了 LDA(Latent Dirichlet Allocation)模型，它能有效的降低维度。它是一种文档主题生成模型，包含词、主题和文档三层结构。每一个文档代表了一些主题所构成的一个概率分布，而每一个主题又代表了很多单词所构成的一个概率分布。

对于有描述（description）属性的每一个类别实体集合构造一个训练语料，即不同的类别有不同的语料，并且产生不同的主题。这也合乎实际，例如 Movie 类别，它包含一些子类别：动作、科幻、恐怖、爱情等等；而 SoftwareApplication 包含：办公、娱乐、影音等等。对每一个类别，得到属于该类别的所有实体的 description 属性值，进行分词然后通过 LDA 模型得到每一个语料属于 n 个主题的概率分布，即一个 n 维的向量。

得到 n 维的 LDA 主题模型向量之后，就可以计算两个实体之间在 description 属性上基于 LDA 模型的相似性。对于 LDA 向量我们使用两种计算相似度的方法来衡量它们之间的相似性：余弦相似度和 Hellinger 距离。在概率论和统计理论中，Hellinger 距离被用来度量两个概率分布的相似度。对于两个离散度概率分布 P=(p1,p2,...,pn)和 Q=（q1,q2,...,qn)，它们的 Hellinger 距离可以定义为公式（4）。

$$H(P,Q)=\frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{k}(\sqrt{p_i}-\sqrt{q_i})^2} \tag{4}$$

上式可以被看作是两个离散概率分布平方根向量的欧式距离，计算出来的值越小表示两个分布越相似。

### 3.3 模型选择

特征生成之后，需要选择分类模型。因为有训练数据，所以可以使用传统的分类方法，有 SVM、Logistic regression、LDA( Linear Discriminant Analysis，与之前提到的 LDA 主题模型是不相同的)，以及集成学习[16]分类方法：RandomForest、AdaBoost、GradientBoosting、Bagging。引入集成学习是因为单一分类器的分类效果不是很理想。而集成学习可以利用单个分类器分类效果一般的特点，将多个分类器的分类结果综合，从而得到一个更好的结果。

集成学习包含两种主要的分类思想：Bagging 和 Boosting。Bagging 的基本思路就是从样本集合中随机生成它的 n 个大小相同的子集，然后使用一种分类算法来用这 n 个子集来训练 n 个分类器，当对测试数据进行预测时，用这 n 个分类器来进行投票，票数最多的那个类将会作为这个集成分类器的最终分类结果。而 Boosting 的方法就稍微有些复杂，它也同样是产生多个分类器，但是后一个分类器依赖于前一个。即前一个被分错的样本的权值会增加，使得下一个分类器尽量不要将其分错。这样就产生了一个连续的分类器，而且每一个分类器都会得到一个权值，当进行预测时，将所有的分类器预测结果乘以相应权值，这样就得到了最后的预测结果。在前面提到的集成学习算法中，RandomForest、AdaBoost、GradientBoosting 就属于 Boosting 分类方法。

对于不同的类别，训练不同的分类器，最终得到该类别下分类效果最好的分类器，然后对测试集进行相应的预测。选择分类效果比较好的分类器需要对训练集进行交叉验证，然后从这 7 个分类器中选择。对分类器的选择取决于评价函数，不同的评价函数将会选择出不同的分类器。在这个问题中，我们使用的评价指标是公式（5）。

$$\mathrm{D}=\sqrt{\sum_{i=1}^{n}(Sc_i-Sm_i)^2} \tag{5}$$

其中$Sc_i$表示预测的相似度，$Sm_i$表示实际的相似度。D 值越小，表示分类的性能越好。另外我们还使用了正确率来作为辅助的评价指标。当模型选择完成后，使用训练全集来训练该分类器，之后进行预测。

### 3.4 算法总结

我们的问题是给出测试集中实体对的相似度。算法主要分为两大模块：特征生成和分

类器训练。首先，得到原始的实体数据，预处理后生成特征，通过对数值型、列表型以及文本型数据的分别处理，得到了多种衡量相似度的指标作为特征。然后，使用种子实体对训练分类器，进而完成对测试数据的分类。

本文使用的实体相似度计算的数据由百度知识图谱竞赛提供，所有的实体共有六种type。不同 type 的实体，属性会有不同，所以特征维度也会有差异。但是同 type 下实体的特征维度是相同的，所以我们需要对不同 type 的实体集合，分别训练不同的分类器进行分类。

表 1 原数据描述

| 类别 | 属性 | 实体数量 |
|---|---|---|
| Movie | id, name, description, inLanguage, datePublish, url, country, actor, director, editor | 4779 |
| MusicRecording | id, name, url, datePublish, duration, genre, byArtist, inAlbum | 1638 |
| ShowSeries | id, name, description, url, host, genre | 665 |
| SoftwareApplication | id, name, description, inLanguage, url, platform, fileSize, operatingSystem | 1017 |
| TVSeries | id, name, description, url, actor, numberOfEpisodes, director, | 2257 |
| VideoGame | id, name, url, publisher, alias,genre | 1107 |

## 4 实验分析

实体的 type 属性共有六种取值：Movie，MusicRecording，ShowSeries，SoftwareApplication，TVSeries，VideoGame。相同 type 下的实体它们的属性列表是相同的，不同 type 下的实体属性列表是不同的，各 type 的属性如表 1 所示。在每一个实体的属性中，都有 id，name 和 url。

表 2 各 type 下的最优分类器

| 类别 | 最优分类器 |
|---|---|
| Movie | AdaBoost |
| MusicRecording | LDA |
| ShowSeries | Logistic regression |
| SoftwareApplication | GradientBoosting |
| TVSeries | AdaBoost |
| VideoGame | Bagging |

在选择模型的过程中，需要使用交叉验证来得出每个分类器在训练集上的分类结果，然后比较各个分类器的性能。图 4 就是 6 个 type 在各个分类器上的分类结果，它们都是在训练集上采用 5 重交叉验证得出的结果。图中的 loss 代表 D 值，横坐标表示使用的分类算法，从左到右依次是 SVM、Logistic regression、RandomForest、AdaBoost、GradientBoosting、Bagging 、LDA，纵坐标表示它们的 loss 值和正确率，其中正确率以%为单位。对于集成学习方法，它们内部使用的弱分类器都是决策树。从图中不难看出，单一分类器的分类效果比较差，所以我们才引入了集成学习的分类方法，它将多个弱分类器结合在一起，使用不同的决策规则，如投票或者是平均的方法，来集成各个分类器的分类结果，从而提高分类的效果。但是使用集成学习之后发现分类效果提高的并不明显，但是总的来说要比单一

的分类器好。

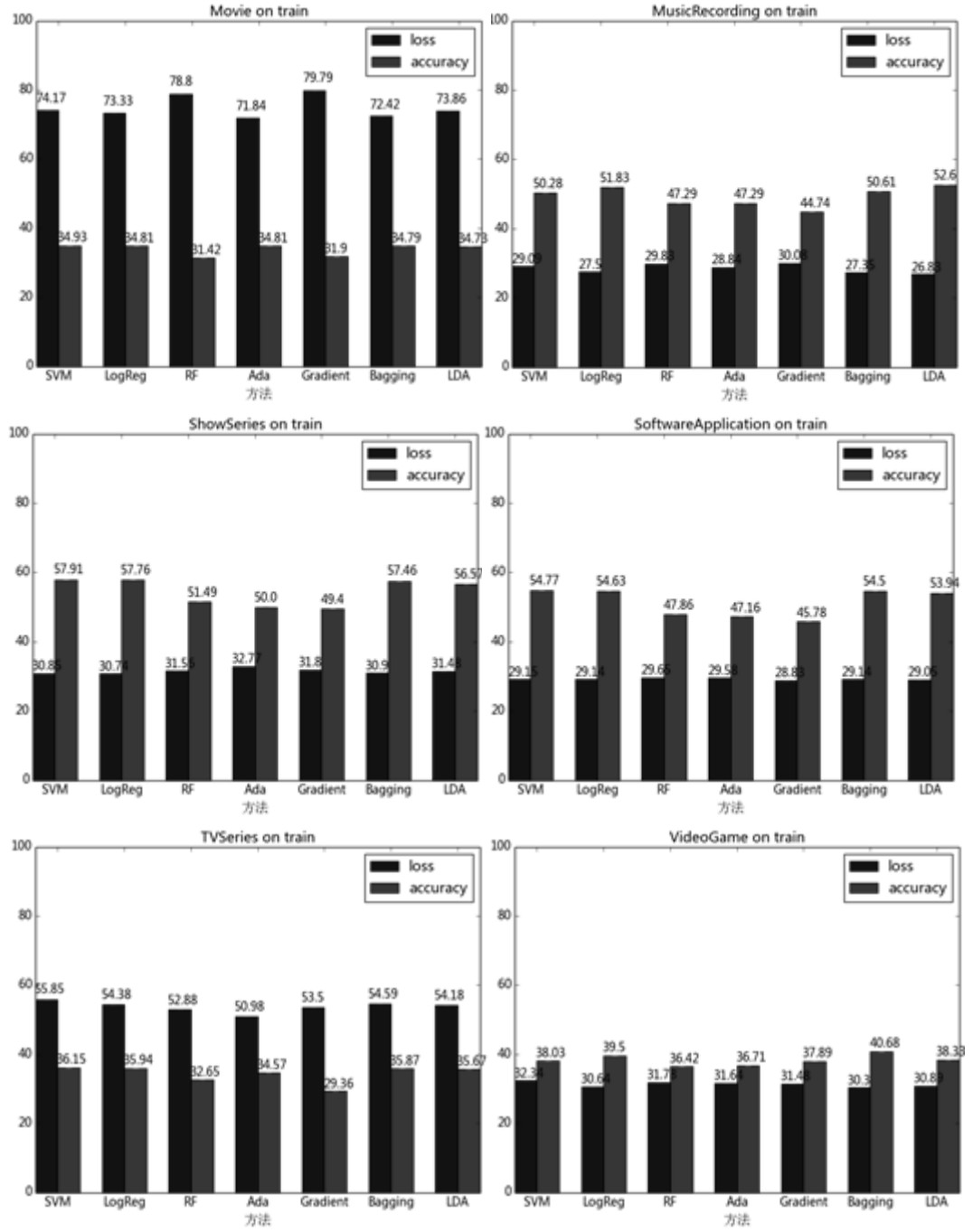


图 4 训练集上的分类结果

在训练集上得出结果之后，选择分类效果最好的分类器来进行最后的训练，完成预测。通过图 4 我们选出了每一个 type 下分类最好的分类器，结果如表 2 所示，这里选择依据的评价函数是 D 值，因为最终对测试集的预测进行评价的函数是 D 值。模型选择完成之后，需要在训练全集上使用最优的分类器来进行训练预测。最后在测试集上的分类结果如表 3

所示。

从表 3 中可以看出，使用表 2 中的分类器序列来对 6 个 type 进行训练，即最优分类器，然后对相应的测试集进行预测，结果是最好的，而且明显要优于对所有 type 使用同一种分类算法的方案。而且对于单一分类器，它们最终的分类结果比集成学习的分类算法差。所以引入集成学习来处理这个问题是正确的选择。

表 3 不同的集成方法的实验结果

| 分类器 | D 值 |
|---|---|
| **最优分类器** | **30.82** |
| 都使用 SVM | 33.11 |
| 都使用 Logistic regression | 32.03 |
| 都使用 RandomForest | 32.98 |
| 都使用 AdaBoost | 31.77 |
| 都使用 GradientBoosting | 31.76 |
| 都使用 Bagging | 31.18 |
| 都使用 LDA | 32.28 |

## 5. 结束语

在知识图谱中的实体相似度计算中，本文提出了一种通用的基于语义的实体相似度计算方法。利用实体各种类型的属性值之间的相似度构建特征，采用集成学习的方法，不断地优化分类效果，最终我们得到的相似度计算结果在百度知识图谱竞赛中获得了第一名。但是我们预测出来的结果正确率却在 40%左右。正确率低的原因是特征值之间的区分度不大，所以后续的工作中需要挖掘出具有区分意义的语义特征，提高文本相似度的计算正确率。

## 参考文献


[1] Ricci F, Shapira B. Recommender systems handbook [M]. Springer, 2011.

[2] Y. Chen, J. Yang, D. Xu, Z. Zhou, and D. Tang. Inference analysis and adaptive training for belief rule based systems [J]. Expert Systems with Applications, vol. 38, no. 10, pp. 12 845-12 860, Sep. 2011.

[3] Sun J, Wang F, Hu J, et al. Supervised patient similarity measure of heterogeneous patient records[J]. ACM SIGKDD Explorations Newsletter, 2012, 14(1): 16-24.

[4] 华秀丽，朱巧明，李培峰.语义分析与词频统计相结合的中文文本相似度量方法研究[J].计算机应用研究,2012,29(3):833-836.

[5] Salton G, Mcgill M J. Introduction to modern information retrieval [M].New York：McGraw-Hill，1983.

[6] Sussna M. Word sense disambiguation for free-text indexing using a massive semantic network[C]. Proceedings of the 2nd International Conference on Information and Knowledge Managemen（CIKM’93），Washington DC，US，1993：67-74.

[7] Bouras C, Tsogkas V. A clustering technique for news articles using WordNet[J]. Knowledge-Based Systems,2012,36(6):115-128.

[8] Abdalgader K, Skabar A. Unsupervised similarity-based word sense disambiguation using context vectors and sentential word importance [J]. ACM Trans. on Speech and Language Processing,2012,9(1):1-21.

[9] Martinez S, Sanchez D, Valls A. Semantic adaptive microaggregation of categorical micro data [J]. Computer Security,2012,31(5):653-672.
[10] Huang H B,Liu Z Z, Zhang W M, et al. Research on calculating semantic similarity based on HOM[J]. Systems Engineering and Electronics,2009,31(7):1750-1754.
[11] 李荣，杨冬，刘磊. 基于本体的概念相似度计算方法研究[J]. 计算机研究与发展，2011, 48(S3):312-317.
[12] 刘杰. 一种基于自动特征权值的实体相似度计算方法[J]. 重庆科技学院学报:自然科学版，2014, 16(3):157-160.
[13] 薛咏，冯博琴，武艳芳. ABox 推理计算实体相似度[J]. 西安交通大学学报, 2015, 49(09):70-76.
[14] Jaccard, Paul. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin de la Société Vaudoise des Sciences Naturelles 37: 547–579.
[15] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases [J]. SIGMOD 1993.
[16] Hang Li. Learning to Rank for Information Retrieval and Natural Language Processing [M]. Morgan & Claypool, 2011.

作者联系方式：李阳，上海市徐汇区梅陇路 130 号，200237，18801951931，marine1ly@163.com

李阳（1992 一），硕士研究生，主要研究领域为自然语言处理。
E-mail：marine1ly@163.com